

# 2 Content Management Interoperability 3 Services Version 1.0

## 4 CMIS Specification

### 5 Committee Draft 04

6 23 September 2009

#### 7 Specification URIs:

##### 8 This Version:

9 <http://docs.oasis-open.org/cmisis/CMIS/v1.0/CD04/CMIS Specification.doc> (Authoritative)

10 <http://docs.oasis-open.org/cmisis/CMIS/v1.0/CD04/CMIS Specification.html>

11 <http://docs.oasis-open.org/cmisis/CMIS/v1.0/CD04/CMIS Specification.pdf>

##### 12 Previous Version:

13 N/A

##### 14 Latest Version:

15 <http://docs.oasis-open.org/cmisis/CMIS/v1.0/CMIS Specification.doc> (Authoritative)

16 <http://docs.oasis-open.org/cmisis/CMIS/v1.0/CMIS Specification.html>

17 <http://docs.oasis-open.org/cmisis/CMIS/v1.0/CMIS Specification.pdf>

##### 18 Technical Committee:

19 OASIS Content Management Interoperability Services (CMIS) TC

##### 20 Chair(s):

21 David Choy, EMC

##### 22 Editor(s):

23 Al Brown, IBM

24 Ethan Gur-Esh, Microsoft

25 Ryan McVeigh, Oracle

26 Florian Muller, OpenText

27

28

##### 29 Related work:

30

##### 31 Declared XML Namespace(s):

32 <http://docs.oasis-open.org/ns/cmisis/core/200908/>

33 <http://docs.oasis-open.org/ns/cmisis/restatom/200908/>

34 <http://docs.oasis-open.org/ns/cmisis/messaging/200908/>

35

36

37**Abstract:**

38 The Content Management Interoperability Services (CMIS) standard defines a domain model and  
39 Web Services and Restful AtomPub bindings that can be used by applications to work with one or  
40 more Content Management repositories/systems.

41

42 The CMIS interface is designed to be layered on top of existing Content Management systems  
43 and their existing programmatic interfaces. It is not intended to prescribe how specific features  
44 should be implemented within those CM systems, not to exhaustively expose all of the CM  
45 system's capabilities through the CMIS interfaces. Rather, it is intended to define a  
46 generic/universal set of capabilities provided by a CM system and a set of services for working  
47 with those capabilities.

48

49**Status:**

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

53 Technical Committee members should send comments on this specification to the Technical  
54 Committee's email list. Others should send comments to the Technical Committee by using the  
55 "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-](http://www.oasis-open.org/committees/CMIS/)  
56 [open.org/committees/CMIS/](http://www.oasis-open.org/committees/CMIS/).

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

61 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/CMIS/)  
62 [open.org/committees/CMIS/](http://www.oasis-open.org/committees/CMIS/).

---

## 63 Notices

64 Copyright © OASIS® 2009. All Rights Reserved.

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

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

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

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

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

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

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

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

107

---

## 108 Table of Contents

1091	Introduction.....	10
110	1.1 Terminology.....	10
111	1.2 Normative References.....	10
112	1.3 Non-Normative References.....	10
1132	Domain Model.....	11
114	2.1 Data Model.....	11
115	2.1.1 Repository.....	11
116	2.1.1.1 Optional Capabilities.....	11
117	2.1.1.2 Implementation Information.....	14
118	2.1.2 Object.....	14
119	2.1.2.1 Property.....	15
120	2.1.3 Object-Type.....	16
121	2.1.3.1 Object-Type Hierarchy and Inheritance.....	16
122	2.1.3.2 Object-Type Attributes.....	17
123	2.1.3.3 Object-Type Property Definitions.....	19
124	2.1.4 Document Object.....	23
125	2.1.4.1 Content Stream.....	23
126	2.1.4.2 Renditions.....	24
127	2.1.4.3 Document Object-Type Definition.....	25
128	2.1.5 Folder Object.....	32
129	2.1.5.1 File-able Objects.....	33
130	2.1.5.2 Folder Hierarchy.....	34
131	2.1.5.3 Paths.....	35
132	2.1.5.4 Folder Object-Type Definition.....	35
133	2.1.6 Relationship Object.....	40
134	2.1.6.1 Relationship Object-Type Definition.....	41
135	2.1.7 Policy Object.....	45
136	2.1.7.1 Policy Object-Type Definition.....	46
137	2.1.8 Access Control.....	49
138	2.1.8.1 ACL, ACE, Principal, and Permission.....	49
139	2.1.8.2 CMIS Permissions.....	50
140	2.1.8.3 ACL Capabilities.....	50
141	2.1.9 Versioning.....	59
142	2.1.9.1 Version Series.....	59
143	2.1.9.2 Latest Version.....	59
144	2.1.9.3 Major Versions.....	59
145	2.1.9.4 Services that modify Version Series.....	60
146	2.1.9.5 Versioning Properties on Document Objects.....	61
147	2.1.9.6 Document Creation and Initial Versioning State.....	62
148	2.1.9.7 Version Specific/Independent membership in Folders.....	62
149	2.1.9.8 Version Specific/Independent membership in Relationships.....	63
150	2.1.9.9 Versioning visibility in Query Services.....	63
151	2.1.10 Query.....	63
152	2.1.10.1 Relational View Projection of the CMIS Data Model.....	64

153	2.1.10.2 Query Language Definition.....	65
154	2.1.10.3 Escaping.....	74
155	2.1.11 Change Log.....	74
156	2.1.11.1 Completeness of the Change Log.....	75
157	2.1.11.2 Change Log Token.....	75
158	2.1.11.3 Change Event.....	75
159	2.2 Services .....	76
160	2.2.1 Common Service Elements.....	76
161	2.2.1.1 Paging .....	76
162	2.2.1.2 Retrieving additional information on objects in CMIS service calls.....	76
163	2.2.1.3 Change Tokens.....	79
164	2.2.1.4 Exceptions.....	79
165	2.2.1.5 ACLs.....	83
166	2.2.2 Repository Services.....	83
167	2.2.2.1 getRepositories.....	83
168	2.2.2.2 getRepositoryInfo.....	83
169	2.2.2.3 getTypeChildren.....	85
170	2.2.2.4 getTypeDescendants.....	85
171	2.2.2.5 getTypeDefinition.....	86
172	2.2.3 Navigation Services .....	86
173	2.2.3.1 getChildren.....	86
174	2.2.3.2 getDescendants.....	87
175	2.2.3.3 getFolderTree.....	89
176	2.2.3.4 getFolderParent.....	90
177	2.2.3.5 getObjectParents.....	90
178	2.2.3.6 getCheckedOutDocs.....	91
179	2.2.4 Object Services.....	92
180	2.2.4.1 createDocument.....	92
181	2.2.4.2 createDocumentFromSource.....	93
182	2.2.4.3 createFolder.....	95
183	2.2.4.4 createRelationship.....	95
184	2.2.4.5 createPolicy.....	96
185	2.2.4.6 getAllowableActions .....	97
186	2.2.4.7 getObject.....	98
187	2.2.4.8 getProperties.....	98
188	2.2.4.9 getObjectByPath.....	99
189	2.2.4.10 getContentStream.....	99
190	2.2.4.11 getRenditions.....	100
191	2.2.4.12 updateProperties.....	100
192	2.2.4.13 moveObject.....	101
193	2.2.4.14 deleteObject.....	102
194	2.2.4.15 deleteTree.....	102
195	2.2.4.16 setContentStream.....	103
196	2.2.4.17 deleteContentStream .....	104
197	2.2.5 Multi-filing Services.....	104
198	2.2.5.1 addObjectToFolder.....	105
199	2.2.5.2 removeObjectFromFolder.....	105
200	2.2.6 Discovery Services.....	105

201	2.2.6.1 query.....	105
202	2.2.6.2 getContentChanges.....	106
203	2.2.7 Versioning Services.....	108
204	2.2.7.1 checkOut.....	108
205	2.2.7.2 cancelCheckOut.....	108
206	2.2.7.3 checkIn.....	109
207	2.2.7.4 getObjectOfLatestVersion.....	110
208	2.2.7.5 getPropertiesOfLatestVersion.....	110
209	2.2.7.6 getAllVersions.....	111
210	2.2.8 Relationship Services.....	112
211	2.2.8.1 getObjectRelationships.....	112
212	2.2.9 Policy Services .....	113
213	2.2.9.1 applyPolicy.....	113
214	2.2.9.2 removePolicy.....	113
215	2.2.9.3 getAppliedPolicies.....	113
216	2.2.10 ACL Services.....	114
217	2.2.10.1 getACL.....	114
218	2.2.10.2 applyACL.....	115
219	3 Restful AtomPub Binding.....	116
220	3.1 Overview.....	116
221	3.1.1 Namespaces.....	116
222	3.1.2 Authentication.....	116
223	3.1.3 Response Formats.....	116
224	3.1.4 Optional Arguments.....	117
225	3.1.5 Errors and Exceptions.....	117
226	3.1.6 Renditions.....	117
227	3.1.7 Content Streams.....	117
228	3.1.8 Paging of Feeds.....	117
229	3.1.9 Services not Exposed.....	117
230	3.2 HTTP .....	118
231	3.2.1 Entity Tag.....	118
232	3.2.2 HTTP Range .....	118
233	3.2.3 HTTP OPTIONS Method .....	118
234	3.2.4 HTTP Status Codes.....	118
235	3.2.4.1 Common CMIS Exceptions.....	118
236	3.2.4.2 Other Exceptions.....	118
237	3.2.4.3 Notable HTTP Status Codes.....	118
238	3.3 Media Types.....	119
239	3.3.1 CMIS Atom.....	119
240	3.3.2 CMIS Query.....	120
241	3.3.3 CMIS Allowable Actions.....	120
242	3.3.4 CMIS Tree.....	121
243	3.3.5 CMIS ACL.....	126
244	3.4 Atom Extensions for CMIS.....	126
245	3.4.1 Atom Element Extensions.....	126
246	3.4.1.1 AtomPub Workspace.....	126

247	3.4.1.2 Atom Feed.....	127
248	3.4.1.3 Atom Entry.....	127
249	3.4.2 Attributes.....	127
250	3.4.2.1 cmisra:id.....	127
251	3.4.2.2 cmisra:renditionKind.....	128
252	3.4.3 CMIS Link Relations .....	128
253	3.4.3.1 Existing Link Relations.....	129
254	3.4.3.2 Hierarchy Navigation Internet Draft Link Relations.....	130
255	3.4.3.3 Versioning Internet Draft Link Relations.....	131
256	3.4.3.4 CMIS Specific Link Relations.....	131
257	3.5 Atom Resources.....	133
258	3.5.1 Feeds .....	133
259	3.5.2 Entries.....	133
260	3.5.2.1 Hierarchical Atom Entries.....	134
261	3.6 AtomPub Service Document (Repository).....	136
262	3.6.1 URI Templates .....	137
263	3.6.1.1 Object By Id .....	139
264	3.6.1.2 Object By Path.....	139
265	3.6.1.3 Query.....	140
266	3.6.1.4 Type By Id.....	141
267	3.6.2 HTTP Methods.....	142
268	3.6.2.1 GET.....	142
269	3.7 Service Collections.....	142
270	3.7.1 Root Folder Collection.....	142
271	3.7.2 Query Collection.....	142
272	3.7.2.1 POST.....	143
273	3.7.3 Checked Out Collection.....	145
274	3.7.3.1 GET.....	145
275	3.7.3.2 POST.....	146
276	3.7.4 Unfiled Collection.....	149
277	3.7.4.1 GET.....	150
278	3.7.4.2 POST.....	150
279	3.7.5 Types Children Collection.....	153
280	3.7.5.1 GET.....	154
281	3.8 Collections.....	154
282	3.8.1 Relationships Collection.....	154
283	3.8.1.1 GET.....	155
284	3.8.1.2 POST.....	155
285	3.8.2 Folder Children Collection.....	157
286	3.8.2.1 GET.....	158
287	3.8.2.2 POST.....	159
288	3.8.3 Policies Collection.....	167
289	3.8.3.1 GET.....	167
290	3.8.3.2 POST.....	167
291	3.9 Feeds.....	170
292	3.9.1 Object Parents Feed.....	170
293	3.9.1.1 GET.....	172

294	3.9.2 Changes.....	172
295	3.9.2.1 GET.....	177
296	3.9.3 Folder Descendants.....	178
297	3.9.3.1 GET.....	184
298	3.9.3.2 DELETE.....	184
299	3.9.4 Folder Tree.....	184
300	3.9.4.1 GET.....	187
301	3.9.4.2 DELETE.....	187
302	3.9.5 AllVersions Feed.....	188
303	3.9.5.1 GET.....	189
304	3.9.5.2 DELETE.....	190
305	3.9.6 Type Descendants Feed.....	190
306	3.9.6.1 GET.....	198
307	3.10 Resources.....	198
308	3.10.1 Type Entry.....	198
309	3.10.1.1 GET.....	198
310	3.10.2 Document Entry.....	200
311	3.10.2.1 GET.....	201
312	3.10.2.2 PUT.....	203
313	3.10.2.3 DELETE.....	203
314	3.10.3 Document Private Working Copy (PWC) Entry.....	203
315	3.10.3.1 GET.....	204
316	3.10.3.2 PUT.....	205
317	3.10.3.3 DELETE.....	206
318	3.10.4 Folder Entry.....	206
319	3.10.4.1 GET.....	207
320	3.10.4.2 PUT.....	208
321	3.10.4.3 DELETE.....	208
322	3.10.5 Relationship Entry.....	208
323	3.10.5.1 GET.....	209
324	3.10.5.2 PUT.....	210
325	3.10.5.3 DELETE.....	211
326	3.10.6 Policy Entry.....	211
327	3.10.6.1 GET.....	211
328	3.10.6.2 PUT.....	213
329	3.10.6.3 DELETE.....	213
330	3.10.7 Content Stream.....	213
331	3.10.7.1 GET.....	213
332	3.10.7.2 PUT.....	213
333	3.10.7.3 DELETE.....	214
334	3.10.8 ACL Resource.....	214
335	3.10.8.1 GET.....	214
336	Web Services Binding.....	216
337	4.1 Overview.....	216
338	4.1.1 WS-I.....	216
339	4.1.2 Authentication.....	216
340	4.1.3 Content Transfer.....	216



341	4.1.4 Reporting Errors.....	216
342	4.2 Web Services Binding Mapping.....	216
343	4.3 Additions to Part I.....	216
344	4.3.1 updateProperties and checkIn Semantics.....	216
345	4.3.2 Content Ranges.....	217
346	4.3.3 Extensions.....	217
347	4.3.4 Web Services Specific Structures.....	217
348	4.3.4.1 cmisFaultType and cmisFault.....	217
349	4.3.4.2 cmisRepositoryEntryType.....	217
350	4.3.4.3 cmisTypeContainer.....	217
351	4.3.4.4 cmisTypeDefinitionListType.....	217
352	4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and cmisObjectInFolderContainerType.....	217
353	4.3.4.6 cmisObjectListType and cmisObjectInFolderListType.....	218
354	4.3.4.7 cmisContentStreamType.....	218
355	4.3.4.8 cmisACLType.....	218
356	4.3.4.9 cmisExtensionType.....	218
357	5 IANA Considerations.....	219
358	5.1 Content-Type Registration.....	219
359	5.1.1 CMIS Query.....	219
360	5.1.2 CMIS AllowableActions.....	219
361	5.1.3 CMIS Tree.....	220
362	5.1.4 CMIS Atom.....	221
363	5.1.5 CMIS ACL.....	222
364	6 Conformance.....	224
365	A. Acknowledgements.....	226
366	B. Non-Normative Text.....	229
367	C. Revision History.....	230
368		
369		

---

# 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.

## 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
- [RFC4918] L. Dusseault, *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*, June 2007
- [RFC3986] T. Berners-Lee, R. Fielding, L. Masinter, *Unified Resource Identifier*, January 2005
- [CMISDM] OASIS, Committee Draft 0.63, “Content Management Interoperability Services (CMIS) Domain Model”, March 2009

## 1.3 Non-Normative References

---

## 402 2 Domain Model

### 403 2.1 Data Model

404 CMIS provides an interface for an application to access a **Repository**. To do so, CMIS specifies a core  
405 data model that defines the *persistent* information entities that are managed by the repository, and  
406 specifies a set of basic services that an application can use to access and manipulate these entities. In  
407 accordance with the CMIS objectives, this data model does not cover *all* the concepts that a full-function  
408 ECM repository typically supports. Specifically, transient entities (such as programming interface objects),  
409 administrative entities (such as user profiles), and extended concepts (such as compound or virtual  
410 document, work flow and business process, event and subscription) are not included.

411 However, when an application connects to a CMIS service endpoint, the same endpoint MAY provide  
412 access to more than one CMIS repository. (How an application obtains a CMIS service endpoint is  
413 outside the scope of CMIS. How the application connects to the endpoint is a part of the protocol that the  
414 application uses.) An application MUST use the CMIS “Get Repositories” service (*getRepositories*) to  
415 obtain a list of repositories that are available at that endpoint. The Repository Identity MUST uniquely  
416 identify an available repository at this service endpoint. Both the repository name and the repository  
417 identity are opaque to CMIS. Aside from the “Get Repositories” service, all other CMIS services are  
418 single-repository-scoped, and require a Repository Identity as an input parameter. In other words, except  
419 for the “Get Repositories” service, multi-repository and inter-repository operations are not supported by  
420 CMIS.

#### 421 2.1.1 Repository

422 The repository itself is described by the CMIS “Get Repository Information” service. The service output is  
423 fully described in section 2.2.2.2 *getRepositoryInfo*.

##### 424 2.1.1.1 Optional Capabilities

425 Commercial ECM repositories vary in their designs. Moreover, some repositories are designed for a  
426 specific application domain and may not provide certain capabilities that are not needed for their targeted  
427 domain. Thus, a repository implementation may not necessarily be able to support all CMIS capabilities. A  
428 few CMIS capabilities are therefore “optional” for a repository to be compliant. A repository’s support for  
429 each of these optional capabilities is discoverable using the *getRepositoryInfo* service. The following is  
430 the list of these optional capabilities. All capabilities are “Boolean” (i.e. the Repository either supports the  
431 capability entirely or not at all) unless otherwise noted.

432

##### 433 Navigation Capabilities:

###### 434 **capabilityGetDescendants**

435 Ability for an application to enumerate the descendants of a folder via the *getDescendants*  
436 service.

437 See section: [2.2.3.2 getDescendants](#)

438

###### 439 **capabilityGetFolderTree**

440 Ability for an application to retrieve the folder tree via the *getFolderTree* service.

441 See section: [2.2.3.3 getFolderTree](#)

442

#### 443**Object Capabilities:**

444 **capabilityContentStreamUpdatability** (**enumCapabilityContentStreamUpdates**)

445 Indicates the support a repository has for updating a document's content stream. Valid values  
446 are:

- 447 • **none**: The content stream may never be updated.
- 448 • **anytime**: The content stream may be updated any time.
- 449 • **pwconly**: The content stream may be updated only when checked out. The abbreviation  
450 PWC is described in section 2.1.8.3.2 .

451 See Section: 2.1.4.1 Content Stream

452

453 **capabilityChanges** (**enumCapabilityChanges**)

454 Indicates what level of changes (if any) the repository exposes via the "change log" service. Valid  
455 values are:

- 456 • **none**: The repository does not support the change log feature.
- 457 • **objectidonly**: The change log can return only the ObjectIDs for changed objects in  
458 the repository and an indication of the type of change, not details of the actual change.
- 459 • **properties**: The change log can return properties and the ObjectID for the changed  
460 objects
- 461 • **all**: The change log can return the ObjectIDs for changed objects in the repository and  
462 more information about the actual change

463 See Section: 2.1.11 Change Log

464

465 **capabilityRenditions** (**enumCapabilityRendition**)

466 Indicates whether or not the repository exposes renditions of document objects.

- 467 • **none**: The repository does not expose renditions at all.
- 468 • **read**: Renditions are provided by the repository and readable by the client.

469

#### 470**Filing Capabilities:**

471 **capabilityMultifiling**

472 Ability for an application to file a document or other file-able object in more than one folder

473 See Section: 2.1.5 Folder Object

474

475 **capabilityUnfiling**

476 Ability for an application to leave a document or other file-able object not filed in any folder

477 See Section: 2.1.5 Folder Object

478

479 **capabilityVersionSpecificFiling**

480 Ability for an application to file individual versions (i.e., not all versions) of a document in a folder

481 See Section: 2.1.8.3.2

482

#### 483**Versioning Capabilities:**

484     **capabilityPWCUpdatable**  
485         Ability for an application to update the "Private Working Copy" of a checked-out document  
486         See Section: 2.1.8.3.2  
487  
488     **capabilityPWCSearchable**  
489         Ability of the Repository to include the "Private Working Copy" of checked-out documents in  
490         query search scope; otherwise PWC's are not searchable  
491         See Section: 2.1.8.3.2  
492  
493     **capabilityAllVersionsSearchable**  
494         Ability of the Repository to include non-latest versions of document in query search scope;  
495         otherwise only the latest version of each document is searchable  
496         See Section: 2.1.8.3.2  
497  
498**Query Capabilities:**  
499     **capabilityQuery (enumCapabilityQuery)**  
500         Indicates the types of queries that the Repository has the ability to fulfill. Query support levels are:  
501         

- **none:** No queries of any kind can be fulfilled.

502         

- **metadataonly:** Only queries that filter based on object properties can be fulfilled.

503             Specifically, the CONTAINS() predicate function is not supported.  
504         

- **fulltextonly:** Only queries that filter based on the full-text content of documents can be

505             fulfilled. Specifically, only the CONTAINS() predicate function can be included in the  
506             WHERE clause.  
507         

- **bothseparate:** The repository can fulfill queries that filter EITHER on the full-text content

508             of documents OR on their properties, but NOT if both types of filters are included in the  
509             same query.  
510         

- **bothcombined:** The repository can fulfill queries that filter on both the full-text content of

511             documents and their properties in the same query.  
512         See Section: 2.1.10 Query  
513  
514     **capabilityJoin (enumCapabilityJoin)**  
515         Indicates the types of JOIN keywords that the Repository can fulfill in queries. Support levels are:  
516         

- **none:** The repository cannot fulfill any queries that include any JOIN clauses.

517         

- **inneronly:** The repository can fulfill queries that include an INNER JOIN clause, but

518             cannot fulfill queries that include other types of JOIN clauses.  
519         

- **innerandouter:** The repository can fulfill queries that include any type of JOIN clause

520             defined by the CMIS query grammar.  
521         See Section: 2.1.10 Query  
522  
523**ACL Capabilities:**  
524     **capabilityACL (enumCapabilityACL)**  
525         Indicates the level of support for ACLs by the repository  
526         

- **none:** The repository does not support ACL services

- 527       • **discover**: The repository supports discovery of ACLs (getACL and other services)
- 528       • **manage**: The repository supports discovery of ACLs AND applying ACLs (getACL and
- 529           applyACL services)
- 530       See Section: 2.8 Access Control

## 5312.1.1.2 Implementation Information

532The “Get Repository Information” service MUST also return implementation information including vendor  
533name, product name, product version, version of CMIS that it supports, the root folder ID (see section  
5342.1.5.2 Folder Hierarchy), and MAY include other implementation-specific information. The version of  
535CMIS that the repository supports MUST be expressed as a Decimal that matches the specification  
536version.

## 5372.1.2 Object

538The entities managed by CMIS are modeled as typed **Objects**. There are four base types of objects:  
539**Document Objects**, **Folder Objects**, **Relationship Objects**, and **Policy Objects**.

- 540       • A *document object* represents a standalone information asset. Document objects are the
- 541           elementary entities managed by a CMIS repository.
- 542       • A *folder object* represents a logical container for a collection of “file-able” objects, which include
- 543           folder objects and document objects. Folder objects are used to organize file-able objects.
- 544           Whether or not an object is file-able is specified in its [object-type definition](#).
- 545       • A *relationship object* represents an instance of directional relationship between two objects. The
- 546           support for relationship objects is optional, and may be discovered via the “Get Type Children”
- 547           service.
- 548       • A *policy* object represents an administrative policy, which may be “applied” to one or more
- 549           “controllablePolicy” objects. Whether or not an object is controllable is specified in its object-type
- 550           definition. The support for policy objects is optional, and may be discovered via the “Get Type
- 551           Children” service.

552Additional object-types MAY be defined in a repository as subtypes of these base types. CMIS services  
553are provided for the discovery of object-types that are defined in a repository. However, object-type  
554management services, such as the creation, modification, and deletion of an object-type, are outside the  
555scope of CMIS.

556Every CMIS object has an opaque and immutable **Object Identity** (ID), which is assigned by the  
557repository when the object is created. An ID uniquely identifies an object within a repository regardless of  
558the type of the object. Repositories SHOULD assign IDs that are “permanent” – that is, they remain  
559unchanged during the lifespan of the identified objects, and they are never reused or reassigned after the  
560objects are deleted from the repository.

561Every CMIS object has a set of named, but not explicitly ordered, **Properties**. (However, a Repository  
562SHOULD always return object properties in a consistent order.) Within an object, each property is  
563uniquely identified by its property definition id.

564In addition, a document object MAY have a **Content-Stream**, which may be used to hold a raw digital  
565asset such as an image or a word-processing document. A repository MUST specify, in each object-type  
566definition, whether document objects of that type MAY, MUST, or MUST NOT have a content-stream. A  
567document MAY also have one or more **Renditions** associated with it. A rendition can be a thumbnail or  
568an alternate representation of the content stream.

569 Document or folder objects MAY have one **Access Control List** (ACL), which controls access to the  
570 document or folder. A policy object may also control access to the document or folder. An ACL  
571 represents a list of **Access Control Entries** (ACEs). An ACE in turn represents one or more permissions  
572 being granted to a **principal** (a user, group, role, or something similar).

573 The notion of localization of the objects in the data model is entirely repository specific.

## 574 2.1.2.1 Property

575 A property MAY hold zero, one, or more typed data value(s). Each property MAY be *single-valued* or  
576 *multi-valued*. A single-valued property contains a single data value, whereas a multi-valued property  
577 contains an ordered list of data values of the same type. The ordering of values in a multi-valued property  
578 MAY be preserved by the repository.

579 If a value is not provided for a property, the property is in a “*value not set*” state. There is no “null” value  
580 for a property. Through protocol binding, a property is either not set, or is set to a particular value or a list  
581 of values.

582 A multi-valued property is either set or not set in its entirety. An individual value of a multi-valued property  
583 MUST NOT be in an individual “value not set” state and hold a position in the list of values. An empty list  
584 of values MUST NOT be allowed.

585 Every property is typed. The Property-type defines the data type of the data value(s) held by the property.  
586 CMIS specifies the following Property-types. They include the following data types defined by “XML  
587 Schema Part 2: Datatypes Second Edition” (W3C Recommendation, 28 October 2004,  
588 <http://www.w3.org/TR/xmlschema-2/>):

- 589 • `string` (xsd:string)
- 590 • `boolean` (xsd:boolean)
- 591 • `decimal` (see section 2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions)
- 592 • `integer` (xsd:integer)
- 593 • `datetime` (xsd:dateTime and see section 2.1.3.3.5 Attributes specific to Decimal Object-Type  
594 Property Definitions)
- 595 • `uri` (xsd:anyURI)

596

597

598 In addition, the following Property-Types are also specified by CMIS:

- 599 • `id`
  - 600 • `html`
- 601 Individual protocol bindings MAY override or re-specify these property types.

602

603 All properties MUST supply a String **queryName** attribute which is used for query and filter operations on  
604 object-types. This is an opaque String with limitations. This string SHOULD NOT contain any characters  
605 that negatively interact with the BNF grammar.

606

607 The string MUST NOT contain:

- 608 • whitespace “ ”,
- 609 • comma “,”
- 610 • double quotes “”
- 611 • single quotes “”

- 612 • backslash “\”
- 613 • the period “.” character or,
- 614 • the open “(” or close “)” parenthesis characters.

615

### 6162.1.2.1.1 ID Property

617An ID property holds a system-generated, read-only identifier, such as an Object ID, an Object-Type ID,  
618etc. (The ID Property-Type is NOT defined by xsd:id.) The lexical representation of an ID is an opaque  
619string. As such, an ID cannot be assumed to be interpretable syntactically or assumed to be to be collate-  
620able with other IDs, and can only be used in its entirety as a single atomic value. When used in a query  
621predicate, an ID can only participate in an “equal” or a “not equal” comparison with a string literal or with  
622another ID.

623While all CMIS identities share the same Property-Type, they do not necessarily share the same address  
624space. Unless explicitly specified, ID properties NEED NOT maintain a referential integrity constraint.  
625Therefore, storing the ID of one object in another object NEED NOT constrain the behavior of either  
626object. A repository MAY, however, support referential constraint underneath CMIS if the effect on CMIS  
627services remains consistent with an allowable behavior of the CMIS model. For example, a repository  
628MAY return an exception when a CMIS service call violates an underlying referential constraint  
629maintained by the repository. In that case, an error message SHOULD be returned to the application to  
630describe the cause of exception and suggest a remedial action. The content of such messages is outside  
631the scope of CMIS.

### 6322.1.2.1.2 HTML Property

633An HTML property holds a document or fragment of Hypertext Markup Language (HTML) content. HTML  
634properties are not guaranteed to be validated in any way. The validation behavior is entirely repository  
635specific.

### 6362.1.3 Object-Type

637An **Object-Type** defines a fixed and non-hierarchical set of properties (“schema”) that all objects of that  
638type have. This schema is used by a repository to validate objects and enforce constraints, and is also  
639used by a user to compose object-type-based (structured) queries.

640All CMIS objects are strongly typed. If a property not specified in an object’s object-type definition is  
641supplied by an application, an exception SHOULD be thrown.

642Each object-type is uniquely identified within a repository by a system-assigned and immutable **Object-**  
643**Type Identifier**, which is of type ID.

644A CMIS repository MUST expose exactly one collection of Object-Types via the “Repository” services  
645(*getTypeChildren*, *getTypeDescendants*, *getTypeDefinition*).

646While a repository MAY define additional object-types beyond the [CMIS Base Object-Types](#), these  
647Object-Types MUST NOT extend or alter the behavior or semantics of a CMIS service (for example, by  
648adding new services). A repository MAY attach additional constraints to an object-type underneath CMIS,  
649provided that the effect visible through the CMIS interface is consistent with the allowable behavior of  
650CMIS.

### 6512.1.3.1 Object-Type Hierarchy and Inheritance

652**Hierarchy** and **Inheritance** for Object-Types are supported by CMIS in the following manner:



- 653 • A CMIS repository MUST have these base types:
  - 654 ○ *cmis:document* object-type
  - 655 ○ *cmis:folder* object-type
- 656 • A CMIS repository MAY have these base types:
  - 657 ○ *cmis:relationship* object-type
  - 658 ○ *cmis:policy* object-type
- 659 • Additional base types MUST NOT exist. Additional object-types MAY be defined as sub-types or
 660 descendant types of these four base types.
- 661 • A **Base Type** does not have a parent type.
- 662 • A non-base type has one and only one parent type. An object-type's **Parent Type** is a part of the
 663 object-type definition.
- 664 • An object-type definition includes a set of *object-type attributes* (e.g. Fileable, Queryable, etc.)
 665 and a property schema that will apply to Objects of that type.
  - 666 ○ There is no inheritance of object-type attributes from a parent object-type to its sub-types.
- 667 • The properties of a CMIS base type MUST be inherited by its descendant types.
- 668 • A **Child Type** whose immediate parent is NOT its base type SHOULD inherit all the property
 669 definitions that are specified for its parent type. In addition, it MAY have its own property
 670 definitions.
  - 671 ○ If a property is NOT inherited by a subtype, the exhibited behavior for query MUST be as
 672 if the value of this property is "not set" for all objects of this sub-type.
- 673 • The scope of a query on a given object-type is automatically expanded to include all the
 674 **Descendant Types** of the given object-type with the attribute *includedInSuperTypeQuery*
 675 equals TRUE. This was added for synthetic types as well as to support different type hierarchies
 676 that are not necessarily the same as CMIS. Only the properties of the given object-type,
 677 including inherited ones, MUST be used in the query. Properties defined for its descendant types
 678 MAY NOT be used in the query, and CAN NOT be returned by the query.
  - 679 ○ If a property of its parent type is not inherited by this type, the property MUST still appear
 680 as a column in the corresponding virtual table in the relational view, but this column MUST
 681 contain a NULL value for all objects of this type. (See section 2.1.10 Query.)

### 6822.1.3.2 Object-Type Attributes

#### 6832.1.3.2.1 Attributes common to ALL Object-Type Definitions

684All **Object-Type Definitions** MUST contain the following **attributes**:

685	<b>id</b>	ID
686	This opaque attribute uniquely identifies this object-type in the repository.	
687		
688	<b>localName</b>	String (optional)
689	This attribute represents the underlying repository's name for the object-type. This field is	
690	opaque and has no uniqueness constraint imposed by this specification.	
691		
692	<b>localNamespace</b>	String (optional)

693 This attribute allows repositories to represent the internal namespace of the underlying  
694 repository's name for the object-type.

695

696 **queryName** String

697 Used for query and filter operations on object-types. This is an opaque String with limitations.  
698 This string SHOULD NOT contain any characters that negatively interact with the BNF grammar.

699

700 The string MUST NOT contain:

701     • whitespace “ ”,  
702     • comma “,”  
703     • double quotes “”  
704     • single quotes “’”  
705     • backslash “\”  
706     • the period “.” character or,  
707     • the open “(” or close “)” parenthesis characters.

708

709 **displayName** String (optional)

710 Used for presentation by application.

711

712 **baseId** Enum

713 A value that indicates whether the base type for this Object-Type is the Document, Folder,  
714 Relationship, or Policy base type.

715

716 **parentId** ID

717 The ID of the Object-Type's immediate parent type.  
718 It MUST be “not set” for a base type.

719

720 **description** String (optional)

721 Description of this object-type, such as the nature of content, or its intended use. Used for  
722 presentation by application.

723

724 **creatable** Boolean

725 Indicates whether new objects of this type MAY be created. If the value of this attribute is FALSE,  
726 the repository MAY contain objects of this type already, but MUST NOT allow new objects of this  
727 type to be created.

728

729 **fileable** Boolean

730 Indicates whether or not objects of this type are [file-able](#).

731

732 **queryable** Boolean

733 Indicates whether or not this object-type can appear in the FROM clause of a query statement. A  
734 non-queryable object-type is not visible through the relational view that is used for query, and  
735 CAN NOT appear in the FROM clause of a query statement.

736		
737	<b>controllablePolicy</b>	Boolean
738	Indicates whether or not objects of this type are controllable via policies. Policy objects can only	
739	be applied to controllablePolicy objects.	
740		
741	<b>controllableACL</b>	Boolean
742	This attribute indicates whether or not objects of this type are controllable by ACL's. Only objects	
743	that are controllableACL can have an ACL.	
744		
745	<b>fulltextIndexed</b>	Boolean
746	Indicates whether objects of this type are indexed for full-text search for querying via the	
747	CONTAINS() query predicate.	
748		
749	<b>includedInSupertypeQuery</b>	Boolean
750	Indicates whether this type and its subtypes appear in a query of this type's ancestor types.	
751	For example: if Invoice is a sub-type of cmis:document, if this is TRUE on Invoice then for a query	
752	on cmis:document, instances of Invoice will be returned if they match.	
753	If this attribute is FALSE, no instances of Invoice will be returned even if they match the query.	

### 7542.1.3.3 Object-Type Property Definitions

755 Besides these object-type attributes, an object-type definition SHOULD contain inherited property  
756 definitions and zero or more additional property definitions. All the properties of an object, including  
757 inherited properties, MUST be retrievable through the "get" services, and MAY appear in the SELECT  
758 clause of a query.

#### 7592.1.3.3.1 Property Types

760 Property types are defined in section 2.1.2.1 Property.

### 7612.1.3.3.2 Attributes common to ALL Object-Type Property Definitions

762 All **Object-Type Property Definitions** MUST contain the following **attributes**:

763	<b>id</b>	ID
764	This opaque attribute uniquely identifies the property in the repository. If two Object-Types each	
765	contain property definitions with the same ID, those property definitions are the same.	
766		
767	<b>localName</b>	String (optional)
768	This attribute represents the underlying repository's name for the property. This field is opaque	
769	and has no uniqueness constraint imposed by this specification.	
770		
771	<b>localNamespace</b>	String (optional)
772	This attribute allows repositories to represent the internal namespace of the underlying	
773	repository's name for the property.	
774		
775	<b>queryName</b>	String

776 Used for query operations on properties. This is an opaque String with limitations. Please see  
777 `queryName` in Object-Type Attributes for the limitations on what characters are not allowed.

778

779 **displayName** String (optional)  
780 Used for presentation by application.

781

782 **description** String (optional)  
783 This is an optional attribute containing a description of the property

784

785 **propertyType** Enum  
786 This attribute indicates the type of this property. It MUST be one of the allowed property types.  
787 (See section 2.1.2.1 Property.)

788

789 **cardinality** Enum  
790 Indicates whether the property can have “zero or one” or “zero or more” values.  
791 Values:

792 

- **single**: Property can have zero or one values (if property is not required), or exactly one

  
793 value (if property is required)

794 

- **multi**: Property can have zero or more values (if property is not required), or one or more

  
795 values (if property is required).

796 Repositories SHOULD preserve the ordering of values in a multi-valued property. That is, the  
797 order in which the values of a multi-valued property are returned in get operations SHOULD be  
798 the same as the order in which they were supplied during previous create/update operation.

799

800 **updatability** Enum  
801 Indicates under what circumstances the value of this property MAY be updated.  
802 Values:

803 

- **readonly**: The value of this property MUST NOT ever be set directly by an application. It

  
804 is a system property that is either maintained or computed by the repository.

805 

- The value of a `readOnly` property MAY be indirectly modified by other repository

  
806 interactions (for example, calling “updateProperties” on an object will change the

807 object’s last modified date, even though that property cannot be directly set via an

808 `updateProperties()` service call.)

809 

- **readwrite**: The property value can be modified using the *updateProperties* service.

810 

- **whencheckedout**: The property value MUST only be update-able using a “private

  
811 `working copy` Document.

812 

- I.e. the update is either made on a “private working copy” object or made using a

  
813 “check in” service.

814 

- **oncreate**: The property value MUST only be update-able during the Create operation

  
815 on that Object.

816

817 **inherited** Boolean  
818 Indicates whether the property definition is inherited from the parent-type when TRUE or it is  
819 explicitly defined for this object-type when FALSE.

820

821     **required**             Boolean

822         If TRUE, then the value of this property MUST never be set to the “not set” state when an object

823         of this type is created/updated. If not provided during a create or update operation, the repository

824         MUST provide a value for this property.

825         If a value is not provided, then the default value defined for the property MUST be set. If no

826         default value is provided and no default value is defined, the repository MUST throw an

827         exception.

828         A property definition SHOULD never state that a property has a “required” value of TRUE and an

829         updatability value of “readonly”.

830

831     **queryable**             Boolean

832         Indicates whether or not the property MAY appear in the WHERE clause of a CMIS query

833         statement.

834         This attribute MUST have a value of FALSE if the Object-type’s attribute for “Queryable” is set to

835         FALSE.

836

837     **orderable**             Boolean

838         Indicates whether the property can appear in the ORDER BY clause of a CMIS query statement

839         or an ORDERBY parameter.

840         This property MUST be FALSE for any property whose cardinality is “multi”.

841

842     **choices**             <PropertyChoiceType list> (multi-valued)

843         Indicates an explicit ordered set of values allowed for this property.

844         If this attribute is “not set”, then any valid value for this property based on its type may be used.

845         Each choice includes a displayName and a value. The displayName is used for presentation

846         purpose. The value will be stored in the property when selected.

847         Choices MAY be hierarchically presented.

848

849     **openChoice**           Boolean

850         This attribute is only applicable to properties that provide a value for the “Choices” attribute.

851         If FALSE, then the data value for the property MUST only be one of the values specified in the

852         “Choices” attribute. If TRUE, then values other than those included in the “Choices” attribute may

853         be set for the property.

854

855     **defaultValue**        <PropertyType>

856         The value that the repository MUST set for the property if a value is not provided by an

857         application when the object is created.

858         If no default value is specified and an application creates an object of this type without setting a

859         value for the property, the repository MUST attempt to store a “value not set” state for the

860         property value. If this occurs for a property that is defined to be required, then the creation

861         attempt MUST throw an exception.

862         The attributes on the default value element are the same as the attributes on the property

863         definition.

### 8642.1.3.3.3 Attributes specific to Integer Object-Type Property Definitions

865The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is  
866“Integer”, in addition to the common attributes specified above. A repository MAY provide additional  
867guidance on what values can be accepted. If the following attributes are not present the repository  
868behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

869     **minValue**             Integer

870         The minimum value allowed for this property.

871         If an application tries to set the value of this property to a value lower than **minValue**, the  
872         repository MUST throw a **constraint** exception.

873

874     **maxValue**             Integer

875         The maximum value allowed for this property.

876         If an application tries to set the value of this property to a value higher than **maxValue**, the  
877         repository MUST throw a **constraint** exception.

878

### 8792.1.3.3.4 Attributes specific to DateTime Object-Type Property Definitions

880The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is  
881“Decimal”, in addition to the common attributes specified above. A repository MAY provide additional  
882guidance on what values can be accepted. If the following attributes are not present the repository  
883behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

884     **resolution**           String Enumeration

885         This is the precision in bits supported for values of this property. Valid values for this attribute are:

- 886             • Year: Year resolution is persisted
- 887             • Date: Date resolution is persisted
- 888             • Time: Time resolution is persisted

889

### 8902.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions

891The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is  
892“Decimal”, in addition to the common attributes specified above. A repository MAY provide additional  
893guidance on what values can be accepted. If the following attributes are not present the repository  
894behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

895     **precision**           Integer Enumeration

896         This is the precision in bits supported for values of this property. Valid values for this attribute are:

- 897             • 32: 32-bit precision (“single” as specified in IEEE-754-1985).
- 898             • 64: 64-bit precision (“double” as specified in IEEE-754-1985.)

899

900     **minValue**           Decimal

901         The minimum value allowed for this property.

902         If an application tries to set the value of this property to a value lower than **minValue**, the  
903         repository MUST throw a **constraint** exception.

904

905     **maxValue**           Decimal

906 The maximum value allowed for this property.  
907 If an application tries to set the value of this property to a value higher than `maxValue`, the  
908 repository MUST throw a **constraint** exception.

### 9092.1.3.3.6 Attributes specific to String Object-Type Property Definitions

910The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is  
911“String”, in addition to the common attributes specified above. A repository MAY provide additional  
912guidance on what values can be accepted. If the following attributes are not present the repository  
913behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

914 **maxLength** Integer

915 The maximum length (in characters) allowed for a value of this property.  
916 If an application attempts to set the value of this property to a string larger than the specified  
917 maximum length, the repository MUST throw a **constraint** exception.

### 9182.1.4 Document Object

919Document objects are the elementary information entities managed by the repository.

920Depending on its Object-type definition, a Document Object may be:

- 921 • **Version-able:** Can be acted upon via the Versioning Services (for example: [checkOut](#), [checkIn](#)).
- 922 • **File-able:** Can be filed in zero, one, or more than one folder via the Multi-filing services.
- 923 • **Query-able:** Can be located via the Discovery Services (*query*).
- 924 • **Controllable-Policy:** Can have Policies applied to it (see section 2.1.7 Policy Object.)
- 925 • **Controllable-ACL:** Can have an ACL applied to it (see section 2.8 Access Control)

926Additionally, whether a Document object MUST, MAY or MUST NOT have a content-stream is specified  
927in its object-type definition. A Document Object MAY be associated with zero or more renditions.

928Note: When a document is versioned, each version of the document is a separate document object. Thus,  
929for document objects, an object ID actually identifies a specific version of a document.

### 9302.1.4.1 Content Stream

931A content-stream is a binary stream. Its maximum length is repository-specific. Each content-stream has  
932a **MIME Media Type**, as defined by RFC2045 and RFC2046. A content-stream’s attributes are  
933represented as properties of the content-stream’s containing document object. There is no MIME-type-  
934specific attribute or name directly associated with the content-stream outside of the document object.

935CMIS provides basic CRUD services for content-stream, using the ID of a content-stream’s containing  
936document object for identification. A content stream also has a `streamId` which is used for access to the  
937stream. The “Set Content-Stream” service (*setContentStream*) either creates a new content-stream for a  
938document object or replaces an existing content-stream. The “Get Content-Stream” service  
939(*getContentStream*) retrieves a content-stream. The “Delete Content-Stream” service  
940(*deleteContentStream*) deletes a content-stream from a document object. In addition, the  
941“CreateDocument” and “Check-in” services MAY also take a content-stream as an optional input. A  
942content stream MUST be specified if required by the type definition. These are the only services that

943operate on content-stream. The “Get Properties” and “Query” services, for example, do not return a  
944content-stream.

945“Set Content-Stream” and “Delete Content-Stream” services are considered modifications to a content-  
946stream’s containing document object, and SHOULD therefore change the object’s *LastModificationDate*  
947property upon successful completion.

948The ability to set or delete a content stream is controlled by the  
949capabilityContentStreamUpdatability capability.

## 9502.1.4.2 Renditions

951Some ECM repositories provide a facility to retrieve alternative representations of a document. These  
952alternative representations are known as renditions. This could apply to a preview case which would  
953enable the client to preview the content of a document without needing to download the full content.  
954Previews are generally reduced fidelity representations such as thumbnails. Renditions can take on any  
955general form, such as a PDF version of a word document.

956A CMIS repository MAY expose zero or more renditions for a document or folder in addition to a  
957document’s content stream. CMIS provides no capability to create or update renditions accessed through  
958the rendition services. Renditions are specific to the version of the document and may differ between  
959document versions. Each rendition consists of a set of rendition attributes and a rendition stream.  
960Rendition attributes are not object properties, and are not queryable. They can be retrieved using the  
961getRenditions service. A rendition stream can be retrieved using the getContentStream service with the  
962rendition’s streamId parameter.

### 9632.1.4.2.1 Rendition Attributes

964A rendition has the following attributes:

965	<b>streamId</b>	ID
966	Identifies the rendition stream.	
967		
968	<b>contentType</b>	String
969	The MIME type of the rendition stream.	
970		
971	<b>length</b>	Integer (optional)
972	The length of the rendition stream in bytes.	
973		
974	<b>title</b>	String (optional)
975	Human readable information about the rendition.	
976		
977	<b>kind</b>	String
978	A categorization String associated with the rendition.	
979		
980	<b>height</b>	Integer (optional)
981	Typically used for ‘image’ renditions (expressed as pixels). SHOULD be present if kind =	
982	cmis:thumbnail.	
983		
984	<b>width</b>	Integer (optional)



985 Typically used for 'image' renditions (expressed as pixels). SHOULD be present if kind =  
986 `cmis:thumbnail`.

987

988 **renditionDocumentId** ID (optional)

989 If specified, then the rendition can also be accessed as a document object in the CMIS services. If  
990 not set, then the rendition can only be accessed via the rendition services. Referential integrity of  
991 this ID is repository-specific.

## 9922.1.4.2.2 Rendition Kind

993A Rendition may be categorized via its `kind`. The repository is responsible for assigning kinds to  
994Renditions, including custom kinds. A repository kind does not necessarily identify a single Rendition for  
995a given Object.

996CMIS defines the following kind:

- 997
- **cmis:thumbnail**: A rendition whose purpose is to provide an image preview of the document  
998 without requiring the client to download the full document content stream. Thumbnails are  
999 generally reduced fidelity representations.

## 10002.1.4.3 Document Object-Type Definition

1001This section describes the definition of the Document Object-Type's attribute values and property  
1002definitions which must be present on Document instance objects. All attributes and property definitions  
1003are listed by their ID.

### 10042.1.4.3.1 Attributes specific to Document Object-Types

1005The following Object **attributes** MUST only apply to Object-Type definitions whose `baseId` is the  
1006`cmis:document` Object-Type, in addition to the common attributes specified above:

1007 **versionable** Boolean

1008 Indicates whether or not objects of this type are version-able. (See section 2.1.8.3.2 .)

1009

1010 **contentStreamAllowed** Enum

1011 A value that indicates whether a content-stream MAY, MUST, or MUST NOT be included in  
1012 objects of this type. Values:

- 1013
- **notallowed**: A content-stream MUST NOT be included

1014

  - **allowed**: A content-stream MAY be included

1015

  - **required**: A content-stream MUST be included (i.e. MUST be included when the object is  
1016 created, and MUST NOT be deleted.)

### 10172.1.4.3.2 Attribute Values

1018The Document Object-Type MUST have the following attribute values.

1019Notes:

- 1020
- A value of `<repository-specific>` indicates that the value of the property MAY be set to any valid  
1021 value for the attribute type.

1022

  - Unless explicitly stated otherwise, all values specified in the list MUST be followed for the Object-  
1023 Type definition.

1024

1025     **id**  
1026         Value: cmis:document  
1027  
1028     **localName**  
1029         Value: <repository-specific>  
1030  
1031     **localNamespace**  
1032         Value: <repository-specific>  
1033  
1034     **queryName**  
1035         Value: cmis:document  
1036  
1037     **displayName**  
1038         Value: <repository-specific>  
1039  
1040     **baseId**  
1041         Value: cmis:document  
1042  
1043     **parentId**  
1044         Value: Not set  
1045  
1046     **description**  
1047         Value: <repository-specific>  
1048  
1049     **creatable**  
1050         Value: <repository-specific>  
1051  
1052     **fileable**  
1053         Value: TRUE  
1054  
1055     **queryable**  
1056         Value: SHOULD be TRUE  
1057  
1058     **controllablePolicy**  
1059         Value: <repository-specific>  
1060  
1061     **includedInSupertypeQuery**  
1062         Value: <repository-specific>  
1063  
1064     **versionable**  
1065         Value: <repository-specific>  
1066

1067     **contentStreamAllowed**  
1068         Value: <repository-specific>  
1069  
1070     **controllableACL**  
1071         Value: <repository-specific>  
1072  
1073     **fulltextIndexed**  
1074         Value: <repository-specific>

### 10752.1.4.3.3 Property Definitions

1076The Document base Object-Type MUST have the following property definitions, and MAY include  
1077additional property definitions. Any attributes not specified for the property definition are repository  
1078specific. For all property definitions on base types, the query name MUST be the same as the property  
1079ID. The repository MUST have the following property definitions on the Document Type:

1080		
1081	<b>cmis:name</b>	Name of the object
1082	Inherited:	False
1083	Property Type:	String
1084	Cardinality:	Single
1085		
1086	<b>cmis:objectId</b>	Id of the object
1087	Required:	False
1088	Inherited:	False
1089	Property Type:	ID
1090	Cardinality:	Single
1091	Updatability:	Read Only
1092	Choices:	Not Applicable
1093	Open Choice:	Not Applicable
1094	MUST be set on the object	
1095		
1096	<b>cmis:baseTypeId</b>	Id of the base object-type for the object
1097	Required:	False
1098	Inherited:	False
1099	Property Type:	ID
1100	Cardinality:	Single
1101	Updatability:	Read Only
1102	Choices:	Not Applicable
1103	Open Choice:	Not Applicable
1104	MUST be set on the object	
1105		
1106	<b>cmis:objectTypeId</b>	Id of the object's type
1107	Required:	True
1108	Inherited:	False

1109	Property Type:	ID
1110	Cardinality:	Single
1111	Updatability:	oncreate
1112	Choices:	Not Applicable
1113	Open Choice:	Not Applicable
1114	MUST be set on the object	
1115		
1116	<b>cmis:createdBy</b>	User who created the object.
1117	Required:	False
1118	Inherited:	False
1119	Property Type:	String
1120	Cardinality:	Single
1121	Updatability:	Read Only
1122	Choices:	Not Applicable
1123	Open Choice:	Not Applicable
1124	Queryable:	True
1125	Orderable:	True
1126	MUST be set on the object	
1127		
1128	<b>cmis:creationDate</b>	DateTime when the object was created.
1129	Required:	False
1130	Inherited:	False
1131	Property Type:	DateTime
1132	Cardinality:	Single
1133	Updatability:	Read Only
1134	Choices:	Not Applicable
1135	Open Choice:	Not Applicable
1136	Queryable:	True
1137	Orderable:	True
1138	MUST be set on the object	
1139		
1140	<b>cmis:lastModifiedBy</b>	User who last modified the object.
1141	Required:	False
1142	Inherited:	False
1143	Property Type:	String
1144	Cardinality:	Single
1145	Updatability:	Read Only
1146	Choices:	Not Applicable
1147	Open Choice:	Not Applicable
1148	Queryable:	True
1149	Orderable:	True
1150	MUST be set on the object	

1151		
1152	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
1153	Required:	False
1154	Inherited:	False
1155	Property Type:	DateTime
1156	Cardinality:	Single
1157	Updatability:	Read Only
1158	Choices:	Not Applicable
1159	Open Choice:	Not Applicable
1160	Queryable:	True
1161	Orderable:	True
1162	MUST be set on the object	
1163		
1164	<b>cmis:changeToken</b>	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1165		
1166	Required:	False
1167	Inherited:	False
1168	Property Type:	String
1169	Cardinality:	Single
1170	Updatability:	Read Only
1171	Choices:	Not Applicable
1172	Open Choice:	Not Applicable
1173	MUST be set on the object	
1174		
1175	<b>cmis:isImmutable</b>	TRUE if the repository MUST throw an error at any attempt to update or delete the object.
1176		
1177	Required:	False
1178	Inherited:	False
1179	Property Type:	Boolean
1180	Cardinality:	Single
1181	Updatability:	Read Only
1182	Choices:	Not Applicable
1183	Open Choice:	Not Applicable
1184	MUST be set on the object	
1185		
1186	<b>cmis:isLatestVersion</b>	See section 2.1.8.3.2 .
1187	Required:	False
1188	Inherited:	False
1189	Property Type:	Boolean
1190	Cardinality:	Single
1191	Updatability:	Read Only
1192	Choices:	Not Applicable

1193	Open Choice:	Not Applicable
1194	MUST be set on the object	
1195		
1196	<b>cmis:isMajorVersion</b>	See section 2.1.8.3.2 .
1197	Required:	False
1198	Inherited:	False
1199	Property Type:	Boolean
1200	Cardinality:	Single
1201	Updatability:	Read Only
1202	Choices:	Not Applicable
1203	Open Choice:	Not Applicable
1204	MUST be set on the object	
1205		
1206	<b>cmis:isLatestMajorVersion</b>	See section 2.1.8.3.2 .
1207	Required:	False
1208	Inherited:	False
1209	Property Type:	Boolean
1210	Cardinality:	Single
1211	Updatability:	Read Only
1212	Choices:	Not Applicable
1213	Open Choice:	Not Applicable
1214	MUST be set on the object	
1215		
1216	<b>cmis:versionLabel</b>	See section 2.1.8.3.2 .
1217	Required:	False
1218	Inherited:	False
1219	Property Type:	String
1220	Updatability:	Read Only
1221	Choices:	Not Applicable
1222	Open Choice:	Not Applicable
1223	MUST be set on the object	
1224		
1225	<b>cmis:versionSeriesId</b>	See section 2.1.8.3.2 .
1226	Required:	False
1227	Inherited:	False
1228	Property Type:	ID
1229	Cardinality:	Single
1230	Updatability:	Read Only
1231	Choices:	Not Applicable
1232	Open Choice:	Not Applicable
1233	MUST be set on the object	
1234		

1235	<b>cmis:isVersionSeriesCheckedOut</b>	See section 2.1.8.3.2 .
1236	Required:	False
1237	Inherited:	False
1238	Property Type:	Boolean
1239	Cardinality:	Single
1240	Updatability:	Read Only
1241	Choices:	Not Applicable
1242	Open Choice:	Not Applicable
1243	MUST be set on the object	
1244		
1245	<b>cmis:versionSeriesCheckedOutBy</b>	See section 2.1.8.3.2 .
1246	Required:	False
1247	Inherited:	False
1248	Property Type:	String
1249	Cardinality:	Single
1250	Updatability:	Read Only
1251	Choices:	Not Applicable
1252	Open Choice:	Not Applicable
1253		
1254	<b>cmis:versionSeriesCheckedOutId</b>	See section 2.1.8.3.2 .
1255	Required:	False
1256	Inherited:	False
1257	Property Type:	ID
1258	Cardinality:	Single
1259	Updatability:	Read Only
1260	Choices:	Not Applicable
1261	Open Choice:	Not Applicable
1262		
1263	<b>cmis:checkinComment</b>	See section 2.1.8.3.2 .
1264	Required:	False
1265	Inherited:	False
1266	Property Type:	String
1267	Cardinality:	Single
1268	Updatability:	Read Only
1269	Choices:	Not Applicable
1270	Open Choice:	Not Applicable
1271		
1272	<b>cmis:contentStreamLength</b>	Length of the content stream (in bytes).
1273	Required:	False
1274	Inherited:	False
1275	Property Type:	Integer
1276	Cardinality:	Single

1277	Updatability:	Read Only
1278	Choices:	Not Applicable
1279	Open Choice:	Not Applicable
1280	MUST be set if the document has a content stream	
1281		
1282	<b>cmis:contentStreamMimeType</b>	MIME type of the Content Stream
1283	Required:	False
1284	Inherited:	False
1285	Property Type:	String
1286	Cardinality:	Single
1287	Updatability:	Read Only
1288	Choices:	Not Applicable
1289	Open Choice:	Not Applicable
1290	MUST be set if the document has a content stream	
1291		
1292	<b>cmis:contentStreamFileName</b>	File name of the Content Stream
1293	Required:	False
1294	Inherited:	False
1295	Property Type:	String
1296	Cardinality:	Single
1297	MUST be set if the document has a content stream	
1298		
1299	<b>cmis:contentStreamId</b>	Id of the stream
1300	Required:	False
1301	Inherited:	False
1302	Property Type:	ID
1303	Cardinality:	Single
1304	Updatability:	Read Only
1305	Choices:	Not Applicable
1306	Open Choice:	Not Applicable

## 13072.1.5 Folder Object

1308A folder object serves as the anchor for a collection of *file-able* objects. The folder object has an *implicit*  
1309hierarchical relationship with each object in its collection, with the anchor folder object being the **Parent**  
1310object and each object in the collection being a **Child** object. This implicit relationship has specific  
1311containment semantics which MUST be maintained by the repository with implicit referential integrity.  
1312(That is, there will never be a dangling parent-relationship or a dangling child-relationship. Furthermore,  
1313object A is a parent of object B if and only if object B is a child of object A.) This system-maintained  
1314implicit relationship is distinct from an *explicit* relationship which is instantiated by an application-  
1315maintained Relationship Object. (See section 2.1.6 Relationship Object.)

1316A folder object does not have a content-stream and is not version-able. A folder object MAY be  
1317associated with zero or more renditions (see section 2.1.4.2 Renditions).



### 13182.1.5.1 File-able Objects

1319A *file-able* object is one that MAY be “filed” into a folder. That is, it MAY be a child object of a folder  
1320object. The following list defines whether the base CMIS Object-types are file-able:

- 1321     **cmis:folder**
- 1322         MUST be file-able
- 1323
- 1324     **cmis:document**
- 1325         MUST be file-able
- 1326
- 1327     **cmis:relationship**
- 1328         MUST NOT be file-able
- 1329
- 1330     **cmis:policy**
- 1331         MAY be file-able

### 13322.1.5.1.1 Document Version Series and Filing

1333Since document objects are versionable, a document object’s membership in a folder MAY be version-  
1334specific or version-independent. That is, the folder membership MAY be restricted to that particular  
1335version of the document or MAY apply to all versions of the document. Whether or not a repository  
1336supports version-specific filing is discoverable via the “Get Repository Information” service  
1337(*getRepositoryInfo*).

1338When the child objects of a folder are retrieved, a specific version of a document MAY be returned. If the  
1339repository supports version-specific filing, the specific version filed in that folder is returned. If the  
1340repository does not support version-specific filing, the latest version of the document is returned.

1341Likewise, this version sensitivity in child-binding also affects the behavior of parent retrieval for a  
1342document object, as well as the scope of the IN\_FOLDER() and IN\_TREE() function calls in a query. For  
1343non-versionable fileable objects, their membership in a folder does not have version sensitivity.

### 13442.1.5.1.2 Filing Restrictions by Object-Type

1345A folder collection’s membership MAY be restricted by object-type. Each folder object has a multi-valued  
1346*AllowedChildObjectTypeId*s property, which specifies that only objects of these types are allowed to be its  
1347children. If this property is “not set”, then objects of any file-able type MAY be filed in the Folder. It is  
1348repository-specific if subtypes of the types listed in the *AllowedChildObjectTypeId*s property MAY be filed  
1349in the folder.

1350Because of these filing constraints, when a new folder object is created, an existing folder object MUST  
1351be specified as its parent.

1352When a non-file-able object is created, a parent folder MUST NOT be specified.

1353When a file-able object is deleted, it is removed from any folder collection in which the object is a  
1354member. In other words, when an object is deleted, all implicit parent-child relationships with the deleted  
1355object as a child cease to exist.

1357CMIS imposes the following constraints on folder objects:

- 1363With these constraints, the folder objects in a CMIS repository necessarily form a strict hierarchy, with the  
1364Root Folder being the root of the hierarchy.

1368A non-folder object does not have any descendant object. Thus, a **Folder Graph** that consists of all  
1369fileable objects as nodes, and all the implicit folder containment relationships as directed edges from  
1370parent to child, is a directed acyclic graph, possibly with some disconnected (orphan) nodes. It follows  
1371that the tree rooted at any given folder object is also a directed acyclic graph, although a non-folder object  
1372in the tree MAY have ancestors that are not ancestors of the rooted folder.



1374Folder objects are handled using [the basic CRUD services for objects](#), and the folder graph is traversed  
1375using the [Navigation Services](#).

1376The **Root Folder** is a special folder such that it cannot be created, deleted, or moved using CMIS  
1377services. Otherwise, it behaves like any other folder object.

## 13782.1.5.3 Paths

1379A folder hierarchy MAY be represented in a canonical notation such as path. For CMIS, a path is  
1380represented by:

- 1381 • '/' for the root folder
- 1382 • All paths start with the root folder.
- 1383 • A set of the folder and object path segments separated by '/' in order of closest to the root.
- 1384 • Folder and object path segments are specified by pathSegment tokens which can be retrieved by  
1385 all services that take an `includePathSegments` parameter.
- 1386 • A pathSegment token MUST not include a '/' character.
  - 1387 ○ It is repository specific how a repository chooses the value for pathSegment.
  - 1388 Repositories might choose to use `cmis:name` or content stream filename for
  - 1389 pathSegment token.
- 1390 • The pathSegment token for each item MUST uniquely identify the item in the folder.

1391

1392E.g., if folder A is under the root, and folder B is under A, then the path would be `/A/B`.

1393A path for an object may be calculated by taking the item's parent folder `cmis:path` property and  
1394appending the "/" character and the object's pathSegment. This constructed path may be given as input  
1395to the `getObjectByPath` service for object by path retrieval.

1396The `getObjectParents` service returns `relativePathSegment` tokens. These tokens are the  
1397pathSegment of the input object relative to the parent folders.

## 13982.1.5.4 Folder Object-Type Definition

1399This section describes the definition of the Folder Object-Type's attribute values and property definitions  
1400which must be present on Folder instance objects. All attributes and property definitions are listed by  
1401their ID.

### 14022.1.5.4.1 Attribute Values

1403The Folder Object-Type MUST have the following attribute values.

1404Notes:

- 1405 • A value of `<repository-specific>` indicates that the value of the property MAY be set to any valid  
1406 value for the attribute type.
- 1407 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the  
1408 Object-Type definition.

1409

1410 **id**

1411 Value: `cmis:folder`

1412

1413 **localName**

1414 Value: `<repository-specific>`

1415

1416     **localNamespace**  
1417         Value: <repository-specific>  
1418  
1419     **queryName**  
1420         Value: cmis:folder  
1421  
1422     **displayName**  
1423         Value: <repository-specific>  
1424  
1425     **baseId**  
1426         Value: cmis:folder  
1427  
1428     **parentId**  
1429         Value: Not set  
1430  
1431     **description**  
1432         Value: <repository-specific>  
1433  
1434     **creatable**  
1435         Value: <repository-specific>  
1436  
1437     **fileable**  
1438         Value: TRUE  
1439  
1440     **queryable**  
1441         Value: SHOULD be TRUE  
1442  
1443     **controllablePolicy**  
1444         Value: <repository-specific>  
1445  
1446     **includedInSupertypeQuery**  
1447         Value: <repository-specific>  
1448  
1449     **controllableACL**  
1450         Value: <repository-specific>  
1451  
1452     **fulltextIndexed**  
1453         Value: <repository-specific>  
1454

## 14552.1.5.4.2 Property Definitions

1456The Folder base Object-Type MUST have the following property definitions, and MAY include additional  
1457property definitions. Any attributes not specified for the Property Definition are repository specific. For all  
1458property definitions on base types, the query name MUST be the same as the property ID. The repository  
1459MUST have the following property definitions on the Folder Type:

1460	<b>cmis:name</b>	Name of the object
1461	Inherited:	False
1462	Property Type:	String
1463	Cardinality:	Single
1464	Required:	True
1465		
1466	<b>cmis:objectId</b>	Id of the object
1467	Required:	False
1468	Inherited:	False
1469	Property Type:	ID
1470	Cardinality:	Single
1471	Updatability:	Read Only
1472	Choices:	Not Applicable
1473	Open Choice:	Not Applicable
1474	MUST be set on the object	
1475		
1476	<b>cmis:baseTypeid</b>	Id of the base object-type for the object
1477	Required:	False
1478	Inherited:	False
1479	Property Type:	ID
1480	Cardinality:	Single
1481	Updatability:	Read Only
1482	Choices:	Not Applicable
1483	Open Choice:	Not Applicable
1484	MUST be set on the object	
1485		
1486	<b>cmis:objectTypeid</b>	Id of the object's type
1487	Required:	False
1488	Inherited:	False
1489	Property Type:	ID
1490	Cardinality:	Single
1491	Updatability:	oncreate
1492	Choices:	Not Applicable
1493	Open Choice:	Not Applicable
1494	MUST be set on the object	
1495		
1496	<b>cmis:createdBy</b>	User who created the object.

1497	Required:	False
1498	Inherited:	False
1499	Property Type:	String
1500	Cardinality:	Single
1501	Updatability:	Read Only
1502	Choices:	Not Applicable
1503	Open Choice:	Not Applicable
1504	Queryable:	True
1505	Orderable:	True
1506	MUST be set on the object	
1507		
1508	<b>cmis:creationDate</b>	DateTime when the object was created.
1509	Required:	False
1510	Inherited:	False
1511	Property Type:	DateTime
1512	Cardinality:	Single
1513	Updatability:	Read Only
1514	Choices:	Not Applicable
1515	Open Choice:	Not Applicable
1516	Queryable:	True
1517	Orderable:	True
1518	MUST be set on the object	
1519		
1520	<b>cmis:lastModifiedBy</b>	User who last modified the object.
1521	Required:	False
1522	Inherited:	False
1523	Property Type:	String
1524	Cardinality:	Single
1525	Updatability:	Read Only
1526	Choices:	Not Applicable
1527	Open Choice:	Not Applicable
1528	Queryable:	True
1529	Orderable:	True
1530	MUST be set on the object	
1531		
1532	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
1533	Required:	False
1534	Inherited:	False
1535	Property Type:	DateTime
1536	Cardinality:	Single
1537	Updatability:	Read Only
1538	Choices:	Not Applicable

1539	Open Choice:	Not Applicable
1540	Queryable:	True
1541	Orderable:	True
1542	MUST be set on the object	
1543		
1544	<b>cmis:changeToken</b>	Token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1545		
1546	Required:	False
1547	Inherited:	False
1548	Property Type:	String
1549	Cardinality:	Single
1550	Updatability:	Read Only
1551	Choices:	Not Applicable
1552	Open Choice:	Not Applicable
1553	MUST be set on the object	
1554		
1555	<b>cmis:parentId</b>	ID of the parent folder of the folder.
1556	Required:	False
1557	Inherited:	False
1558	Property Type:	ID
1559	Cardinality:	Single
1560	Updatability:	Read Only
1561	Choices:	Not Applicable
1562	Open Choice:	Not Applicable
1563	MUST be set on the object	
1564		
1565	<b>cmis:path</b>	The fully qualified path to this folder. See section 2.1.5.3
1566		Paths.
1567	Required:	False
1568	Inherited:	False
1569	Property Type:	String
1570	Cardinality:	Single
1571	Updatability:	Read Only
1572	Choices:	Not Applicable
1573	Open Choice:	Not Applicable
1574	MUST be set on the object	
1575		
1576	<b>cmis:allowedChildObjectTypes</b>	Id's of the set of Object-types that can be created, moved or
1577		filed into this folder.
1578	Required:	False
1579	Inherited:	False
1580	Property Type:	ID

1581	Cardinality:	Multi
1582	Updatability:	Read Only
1583	Choices:	Not Applicable
1584	Open Choice:	Not Applicable

## 15852.1.6 Relationship Object

1586A relationship object is semantically a *dependent* object. A relationship object MUST NOT have a content-  
1587stream, and MUST NOT be versionable, MUST NOT be queryable, and MUST NOT be fileable, although  
1588it MAY be controllable.

1589If a repository does not support relationship objects, the relationship base object-type SHOULD NOT be  
1590returned by a “Get Types” service call.

1591A **Relationship Object** instantiates an explicit, binary, directional, non-invasive, and typed relationship  
1592between a **Source Object** and a **Target Object**. The source object and the target object MUST both be  
1593independent objects, such as a document object, a folder object, or a policy object. Whether a policy  
1594object is allowed to be the source or target object of a relationship object is repository-specific.

1595The relationship instantiated by a relationship object is *explicit* since it is explicitly represented by an  
1596object and is explicitly managed by application.

1597This relationship is *non-invasive* in the sense that creating or removing this relationship SHOULD NOT  
1598modify either the source or the target object. That is, it SHOULD NOT require an update capability (or  
1599permission) on either object; SHOULD NOT affect the versioning state of either object; and SHOULD  
1600NOT change their “Last Modification Date”.

1601Explicit relationships can be used to create an arbitrary relationship graph among independent objects.  
1602Such a relationship graph is only structural in nature. No inheritance or transitive properties are attached  
1603to a relationship graph.

1604The notion of a source object and a target object of a relationship is used solely to indicate the direction of  
1605the relationship. No semantics or implementation bias is implied by this terminology.

1606The binding of a relationship object to a source document object or to a target document object MAY be  
1607either version-specific or version-independent. This version sensitivity is repository-specific, and is largely  
1608transparent to CMIS. An independent object MAY participate in any number of explicit relationships, as  
1609the source object for some and as the target object for others. Multiple relationships MAY exist between  
1610the same pair of source and target objects.

1611Referential integrity, either between the source object and the target object, or between the relationship  
1612object and the source or target object, is repository-specific. Therefore, creating an explicit relationship  
1613between two objects MAY impose a constraint on any of the three objects, and removing a relationship or  
1614deleting either the source or the target object MAY be restricted by such a constraint. If the source or the  
1615target object of a relationship is deleted, the repository MAY automatically delete the relationship object.

1616Like all CMIS objects, relationship objects are typed. Typing relationship allows them to be grouped,  
1617identified, and traversed by type id, and for properties to be defined for individual relationship types.

1618Additionally, a relationship object-type MAY specify that only Objects of a specific Object-Type can  
1619participate as the source object or target object for relationship objects of that type. If no such constraints  
1620are specified, then an independent object of any type MAY be the source or the target of a relationship  
1621object of that type.



1622When a relationship object is created, the source object ID and the target object ID MUST reference valid  
1623non-relationship CMIS objects.

1624When a relationship object is retrieved, its source object or target object MAY no longer exist, since  
1625referential integrity MAY not be maintained by a repository.

1626In addition to object CRUD services, a “Get Relationships” service (*getObjectRelationships*) may be used  
1627to return a set of relationship objects in which a given independent object is identified as the source or the  
1628target object, according to the binding semantics maintained by the repository (i.e., either a version-  
1629specific or a version-independent binding as described above).

## 16302.1.6.1 Relationship Object-Type Definition

1631This section describes the definition of the Relationship Object-Type’s attribute values and property  
1632definitions which must be present on Relationship instance objects. All attributes and property definitions  
1633are listed by their ID.

### 16342.1.6.1.1 Attributes specific to Relationship Object-Types

1635The following Object **attributes** MUST only apply to Object-Type definitions whose baseId is the  
1636cmis:relationship Object-Type, in addition to the common attributes specified above:

1637     **allowedSourceTypes**                             ID (multi-valued)

1638         A list of object-type IDs, indicating that the source object of a relationship object of this type  
1639         MUST only be one of the types listed.

1640         If this attribute is “not set”, then the source object MAY be of any type.

1641

1642     **allowedTargetTypes**                             ID (multi-valued)

1643         A list of object-type IDs, indicating that the target object of a relationship object of this type MUST  
1644         only be one of the types listed.

1645         If this attribute is “not set”, then the target object MAY be of any type.

### 16462.1.6.1.2 Attribute Values

1647The Relationship Object-Type MUST have the following attribute values.

1648Notes:

1649     • A value of <repository-specific> indicates that the value of the property MAY be set to any valid  
1650     value for the attribute type.

1651     • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the  
1652     Object-Type definition.

1653

1654     **id**

1655         Value: cmis:relationship

1656

1657     **localName**

1658         Value: <repository-specific>

1659

1660     **localNamespace**

1661         Value: <repository-specific>

1662

1663     **queryName**  
1664         Value: cmis:relationship  
1665  
1666     **displayName**  
1667         Value: <repository-specific>  
1668  
1669     **baseId**  
1670         Value: cmis:relationship  
1671  
1672     **parentId**  
1673         Value: Not set  
1674  
1675     **description**  
1676         Value: <repository-specific>  
1677  
1678     **creatable**  
1679         Value: <repository-specific>  
1680  
1681     **fileable**  
1682         Value: FALSE  
1683  
1684     **queryable**  
1685         Value: <repository-specific>  
1686  
1687     **includedInSupertypeQuery**  
1688         Value: <repository-specific>  
1689  
1690     **controllablePolicy**  
1691         Value: <repository-specific>  
1692  
1693     **allowedSourceTypes**  
1694         Value: <repository-specific>  
1695  
1696     **allowedTargetTypes**  
1697         Value: <repository-specific>  
1698  
1699     **controllableACL**  
1700         Value: <repository-specific>  
1701  
1702     **fulltextIndexed**  
1703         Value: <repository-specific>  
1704

### 17052.1.6.1.3 Property Definitions

1706The Relationship base Object-Type MUST have the following property definitions, and MAY include  
1707additional property definitions. Any attributes not specified by the Property Definitions are repository  
1708specific. For all property definitions on base types, the query name MUST be the same as the property  
1709ID. The repository MUST have the following property definitions on the Relationship Type:

1710		
1711	<b>cmis:name</b>	Name of the object
1712	Inherited:	False
1713	Property Type:	String
1714	Cardinality:	Single
1715		
1716	<b>cmis:objectId</b>	Id of the object
1717	Required:	False
1718	Inherited:	False
1719	Property Type:	ID
1720	Cardinality:	Single
1721	Updatability:	Read Only
1722	Choices:	Not Applicable
1723	Open Choice:	Not Applicable
1724	MUST be set on the object	
1725		
1726	<b>cmis:baseTypeId</b>	Id of the base object-type for the object
1727	Required:	False
1728	Inherited:	False
1729	Property Type:	ID
1730	Cardinality:	Single
1731	Updatability:	Read Only
1732	Choices:	Not Applicable
1733	Open Choice:	Not Applicable
1734	MUST be set on the object	
1735		
1736	<b>cmis:objectTypeId</b>	Id of the object's type
1737	Required:	False
1738	Inherited:	False
1739	Property Type:	ID
1740	Cardinality:	Single
1741	Updatability:	oncreate
1742	Choices:	Not Applicable
1743	Open Choice:	Not Applicable
1744	MUST be set on the object	
1745		
1746	<b>cmis:createdBy</b>	User who created the object.

1747	Required:	False
1748	Inherited:	False
1749	Property Type:	String
1750	Cardinality:	Single
1751	Updatability:	Read Only
1752	Choices:	Not Applicable
1753	Open Choice:	Not Applicable
1754	MUST be set on the object	
1755		
1756	<b>cmis:creationDate</b>	DateTime when the object was created.
1757	Required:	False
1758	Inherited:	False
1759	Property Type:	DateTime
1760	Cardinality:	Single
1761	Updatability:	Read Only
1762	Choices:	Not Applicable
1763	Open Choice:	Not Applicable
1764	MUST be set on the object	
1765		
1766	<b>cmis:lastModifiedBy</b>	User who last modified the object.
1767	Required:	False
1768	Inherited:	False
1769	Property Type:	String
1770	Cardinality:	Single
1771	Updatability:	Read Only
1772	Choices:	Not Applicable
1773	Open Choice:	Not Applicable
1774	MUST be set on the object	
1775		
1776	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
1777	Required:	False
1778	Inherited:	False
1779	Property Type:	DateTime
1780	Cardinality:	Single
1781	Updatability:	Read Only
1782	Choices:	Not Applicable
1783	Open Choice:	Not Applicable
1784	MUST be set on the object	
1785		
1786	<b>cmis:changeToken</b>	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1787		
1788	Required:	False

1789	Inherited:	False
1790	Property Type:	String
1791	Cardinality:	Single
1792	Updatability:	Read Only
1793	Choices:	Not Applicable
1794	Open Choice:	Not Applicable
1795		
1796	<b>cmis:sourceId</b>	ID of the source object of the relationship.
1797	Required:	True
1798	Inherited:	False
1799	Property Type:	ID
1800	Cardinality:	Single
1801	Choices:	Not Applicable
1802	Open Choice:	Not Applicable
1803		
1804	<b>cmis:targetId</b>	ID of the target object of the relationship.
1805	Required:	True
1806	Inherited:	False
1807	Property Type:	ID
1808	Cardinality:	Single
1809	Choices:	Not Applicable
1810	Open Choice:	Not Applicable

## 18112.1.7 Policy Object

1812A policy object represents an administrative policy that can be enforced by a repository, such as a  
1813retention management policy. CMIS 1.0 does not specify what kinds of administrative policies that are  
1814specifically supported, nor attempts to model administrative policy of any particular kind. Only a base  
1815object-type is specified for policy objects. Each policy object holds the text of an administrative policy as a  
1816repository-specific string, which is opaque to CMIS and which may be used to support policies of various  
1817kinds. A repository may create subtypes of this base type to support different kinds of administrative  
1818policies more specifically. If a repository does not support policy objects, the policy base object-type  
1819SHOULD NOT be returned by a “Get Types” service call. This is an extension point for repositories that  
1820want to expose other capabilities via CMIS that are not supported directly in CMIS 1.0.

1821Aside from allowing an application to create and maintain policy objects, CMIS allows an application to  
1822“apply” a policy to an object, and to remove an applied policy from an object. An object to which a policy  
1823may be applied is called a *controllable* object. A policy MAY be applied to multiple controllable objects.  
1824Conversely, a repository MAY allow multiple policies applied to a controllable object. (A repository may,  
1825for example, impose constraints such as only one policy of each kind can be applied to an object.)  
1826Whether or not an object is controllable is specified by the object’s type definition. Applying a policy to an  
1827object is to place the object under the control of that policy (while the object may also be under the control  
1828of other policies at the same time), and removing an applied policy from one of its controlled objects is to  
1829remove the corresponding control from that object. This control may change the state of the object, may  
1830impose certain constraints on service calls operating on this object, or may cause certain management  
1831actions to take place. The effect of this control, when this effect takes place, and how this control interacts  
1832with other controls, are repository-specific. Only directly/explicitly applied policies are covered by CMIS  
18331.0. Indirectly applying policy to an object, e.g. through inheritance, is outside the scope of CMIS 1.0.

1834A policy object does not have a content-stream and is not versionable. It may be fileable, queryable or  
 1835controllable. Policy objects are handled using the basic CRUD services for objects. If a policy is updated,  
 1836the change may alter the corresponding control on objects that the policy is currently applied to. If a  
 1837controlled object is deleted, all the policies applied to that object, if there are any, are removed from that  
 1838object. A policy object that is currently applied to one or more controllable objects CAN NOT be deleted.  
 1839That is, there is an implicit referential constraint from a controlled object to its controlling policy object(s).  
 1840Besides the basic CRUD services, the “Apply Policy” (*applyPolicy*) and the “Remove Policy”  
 1841(*removePolicy*) services may be used to apply a policy object to a controllable object and respectively to  
 1842remove an applied policy from one of its controlled objects. In addition, the “Get Applied Policies”  
 1843(*getAppliedPolicies*) service may be used to obtain the policy objects that are currently applied to a  
 1844controllable object.

## 18452.1.7.1 Policy Object-Type Definition

1846This section describes the definition of the Policy Object-Type’s attribute values and property definitions  
 1847which must be present on Policy instance objects. All attributes and property definitions are listed by their  
 1848ID.

### 18492.1.7.1.1 Attribute Values

1850The Policy Object-Type MUST have the following attribute values.

1851Notes:

- 1852 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid  
 1853 value for the attribute type.
- 1854 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the  
 1855 Object-Type definition.

1856

1857 **id**

1858 Value: cmis:policy

1859

1860 **localName**

1861 Value: <repository-specific>

1862

1863 **localNamespace**

1864 Value: <repository-specific>

1865

1866 **queryName**

1867 Value: cmis:policy

1868

1869 **displayName**

1870 Value: <repository-specific>

1871

1872 **baseId**

1873 Value: cmis:policy

1874

1875 **parentId**

1876 Value: Not set

1877

1878 **description**  
1879 Value: <repository-specific>  
1880  
1881 **creatable**  
1882 Value: <repository-specific>  
1883  
1884 **fileable**  
1885 Value: <repository-specific>  
1886  
1887 **queryable**  
1888 Value: <repository-specific>  
1889  
1890 **includedInSupertypeQuery**  
1891 Value: <repository-specific>  
1892  
1893 **controllablePolicy**  
1894 Value: <repository-specific>  
1895  
1896 **controllableACL**  
1897 Value: <repository-specific>  
1898  
1899 **fulltextIndexed**  
1900 Value: <repository-specific>  
1901

## 19022.1.7.1.2 Property Definitions

1903The Policy base Object-Type MUST have the following property definitions, and MAY include additional  
1904property definitions. Any attributes not specified by the Property Definitions are repository specific. For  
1905all property definitions on base types, the query name MUST be the same as the property ID. The  
1906repository MUST have the following property definitions on the Policy Type:

1907		
1908	<b>cmis:name</b>	Name of the object
1909	Inherited:	False
1910	Property Type:	String
1911	Cardinality:	Single
1912		
1913	<b>cmis:objectId</b>	Id of the object
1914	Required:	False
1915	Inherited:	False
1916	Property Type:	ID
1917	Cardinality:	Single
1918	Updatability:	Read Only
1919	Choices:	Not Applicable

1920	Open Choice:	Not Applicable
1921		
1922	<b>cmis:baseTypeId</b>	Id of the base object-type for the object
1923	Required:	False
1924	Inherited:	False
1925	Property Type:	ID
1926	Cardinality:	Single
1927	Updatability:	Read Only
1928	Choices:	Not Applicable
1929	Open Choice:	Not Applicable
1930		
1931	<b>cmis:objectTypeId</b>	Id of the object's type
1932	Required:	False
1933	Inherited:	False
1934	Property Type:	ID
1935	Cardinality:	Single
1936	Updatability:	oncreate
1937	Choices:	Not Applicable
1938	Open Choice:	Not Applicable
1939		
1940	<b>cmis:createdBy</b>	User who created the object.
1941	Required:	False
1942	Inherited:	False
1943	Property Type:	String
1944	Cardinality:	Single
1945	Updatability:	Read Only
1946	Choices:	Not Applicable
1947	Open Choice:	Not Applicable
1948		
1949	<b>cmis:creationDate</b>	DateTime when the object was created.
1950	Required:	False
1951	Inherited:	False
1952	Property Type:	DateTime
1953	Cardinality:	Single
1954	Updatability:	Read Only
1955	Choices:	Not Applicable
1956	Open Choice:	Not Applicable
1957		
1958	<b>cmis:lastModifiedBy</b>	User who last modified the object.
1959	Required:	False
1960	Inherited:	False
1961	Property Type:	String



1962	Cardinality:	Single
1963	Updatability:	Read Only
1964	Choices:	Not Applicable
1965	Open Choice:	Not Applicable
1966		
1967	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
1968	Required:	False
1969	Inherited:	False
1970	Property Type:	DateTime
1971	Cardinality:	Single
1972	Updatability:	Read Only
1973	Choices:	Not Applicable
1974	Open Choice:	Not Applicable
1975		
1976	<b>cmis:changeToken</b>	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1977		
1978	Required:	False
1979	Inherited:	False
1980	Property Type:	String
1981	Cardinality:	Single
1982	Updatability:	Read Only
1983	Choices:	Not Applicable
1984	Open Choice:	Not Applicable
1985		
1986	<b>cmis:policyText</b>	User-friendly description of the policy
1987	Required:	True
1988	Inherited:	False
1989	Property Type:	String
1990	Cardinality:	Single
1991	Choices:	Not Applicable
1992	Open Choice:	Not Applicable

## 19932.1.8 Access Control

1994A repository can support either a base set of CMIS-defined permissions and/or its own set of repository  
1995specific permissions.

1996The getACL service allows the requestor to specify that the result be expressed using only the CMIS  
1997defined permissions. Without this restriction, the response may include, or be solely expressed in  
1998repository specific permissions. The applyACL service permits either CMIS permissions or repository  
1999permissions, or a combination of both, to be used.

### 20002.1.8.1 ACL, ACE, Principal, and Permission

2001An **ACL** is a list of **Access Control Entries** (ACEs) and MAY hold zero or more ACEs. If an ACL has no  
2002ACEs, the behavior is the same as if the ACL is not set.

2003An **ACE** holds:

- 2004 • one **Principal**: A principal represents a user management object, e.g. a user, group, or role.  
2005 It holds one **String** with the *principalid*.
- 2006 • One or more **Strings** with the names of the *permissions*.
- 2007 • a **Boolean** flag *direct*, which indicates if TRUE the ACE is directly assigned to the object. If  
2008 FALSE, that the ACE is somehow derived.

## 20092.1.8.2 CMIS Permissions

2010There are three basic permissions predefined by CMIS:

- 2011 • **cmis:read**: to be used to express “permission to read”. A Repository SHOULD express the  
2012 permission for reading properties AND reading content with this permission.
- 2013 • **cmis:write**: to be used to express “permission to write”. SHOULD be used to express permission  
2014 to write properties and content of an object. MAY include other basic CMIS permissions.
- 2015 • **cmis:all**: SHOULD be used to express all the permissions of a repository. SHOULD include all  
2016 other basic CMIS permissions.

2017How these basic permissions can be mapped to the allowable actions is repository specific. However, the  
2018actual repository semantics for the basic permissions with regard to allowable actions can be discovered  
2019by the *mappings* parameter returned by *getRepositoryInfo* (see below).

2020Repositories MAY extend this set with repository-specific permissions.

## 20212.1.8.3 ACL Capabilities

2022Whether a repository supports ACLs at all, may be discovered via *capabilityACL* returned by  
2023*getRepositoryInfo* (see section 2.1.1.1 Optional Capabilities). If *capabilityACL* is *none*, ACLs are not  
2024supported by the repository.

2025If *capabilityACL* is *discover* or *manage*, additional information about the repositories permission model  
2026and how changes to ACL are handled, can be discovered via the *getRepositoryInfo* service:

- 2027 • **<Array> Enum propagation**: specifies, how non-direct ACEs can be handled by the repository  
2028 using the following values (see section 2.2.10.2 *applyACL*):
  - 2029 ○ **objectonly** indicates, that the repository is able to apply ACEs to a document or folder,  
2030 without changing the ACLs of other objects.
  - 2031 ○ **propagate**: indicates that the ACEs is to be applied to the given object and all inheriting  
2032 objects.
  - 2033 ○ **repositorydetermined** indicates, that the repository has its own mechanism of  
2034 computing how changing an ACL for an object influences the non-direct ACEs of other  
2035 objects.
- 2036 • **<Array> PermissionDefinition repositoryPermissions**: is a list with names and descriptions of  
2037 the supported permissions.
- 2038 • **<Array> PermissionMapping mappings**: contains a list with mappings for the basic CMIS  
2039 permissions to allowed actions.

## 20402.1.8.3.1 Supported Permissions

2041The list of permission definitions returned by *getRepositoryInfo* lists all the *permissions* a repository  
2042supports. This list also includes the CMIS permissions if supported by the repository.

2043A *PermissionDefinition* holds:

- 2044 • **String permission**: the (technical) name of the permission (unique within the list of permission  
2045 definitions).

- 2046 • **(Optional) String description:** an optional description of the permission that should be used as  
2047 the permission's name to be presented to the user.

### 2048**2.1.8.3.2 AllowableActions & Permission Mapping**

2049CMIS provides a mechanism called “AllowableActions” which allows an application to discover the set of  
2050service operations that can currently be performed on a particular object, without having to actually invoke  
2051the service.

2052The set of allowable actions on an object at a point in time are affected not only by CMIS ACLs, but also  
2053by other factors such as:

- 2054 • Constraints inherent in the CMIS Domain Model based on the object's base type or current  
2055 versioning state.
- 2056 • Policies or other control mechanisms that are opaque to CMIS.

2057

2058CMIS defines several services that applications can use at run-time to discover the AllowableActions for  
2059an object.

2060If a Repository supports ACLs, then the repository **MUST** provide a mapping table that defines how the  
2061permissions supported by the repository interact with the CMIS allowable actions, i.e. which permissions  
2062are necessary for a principal to have on one or more objects in order to potentially perform each action,  
2063subject to the other constraints on allowable actions above.

2064This section defines both the allowable actions as well as how those actions are presented in the  
2065PermissionMapping table.

2066The Permission Mapping table contains a set of (*key, permissions*) pairs:

- 2067 • **String Key:** Because several allowable actions may require permissions on more than one object  
2068 – for example, moving a document from one folder to another may require permissions on the  
2069 document and each of the folders – the mapping table is defined in terms of permission “keys”,  
2070 where each key combines the name of the allowable action as the object for which the principal  
2071 needs the required permission.
  - 2072 ○ For example – the `canMoveObject.Source` key indicates the permissions that the  
2073 principal must have on the “source folder” to move an object from that folder into another  
2074 folder.
- 2075 • **<Array> String permissions:** The names of one or more permissions that the principal **MUST**  
2076 have. If more than one permission is specified, then the principal **MUST** be allowed to perform the  
2077 operation if they have **ANY** of the listed permissions.

2078The list below defines all mapping keys, as well as a permissions mapping that repositories **SHOULD**  
2079use. Repositories **MAY** require additional permissions.

2080For convenience, the list below groups all mapping entries by the underlying Allowable Actions, and  
2081includes descriptive information. For each Allowable Action the following information is given:

2082	Description:	The description and name of the service the AllowableAction enables.
2083	Base Object:	The base object-types for which the allowable action <b>MAY</b> be <b>TRUE</b> .
2084	Operand:	The object the permission applies to.
2085	Key:	The permission mapping key.
2086	Permissions:	The permission values.

2087

#### 2088**Navigation Services:**

##### 2089 **canGetDescendants**

2090	Description:	Can get the descendants of the folder ( <code>getDescendants</code> )
2091	Base Object:	<code>cmis:folder</code>

2092	Operand:	cmis:folder
2093	Key:	canGetDescendants.Folder
2094	Permission:	Read
2095		
2096	<b>canGetFolderTree</b>	
2097	Description:	Can get the sub-folder tree of the folder ( <code>getFolderTree</code> )
2098	Base Object:	cmis:folder
2099	Operand:	cmis:folder
2100	Key:	canGetFolderTree.Folder
2101	Permission:	Read
2102		
2103	<b>canGetChildren</b>	
2104	Description:	Can get the children of the folder ( <code>getChildren</code> )
2105	Base Object:	cmis:folder
2106	Operand:	cmis:folder
2107	Key:	canGetChildren.Folder
2108	Permission:	Read
2109		
2110	<b>canGetFolderParent</b>	
2111	Description:	Can get the parent/ancestor folder(s) of the folder ( <code>getFolderParent</code> )
2112	Base Object:	cmis:folder
2113	Operand:	cmis:folder
2114	Key:	canGetFolderParent.Folder
2115	Permission:	Read
2116		
2117	<b>canGetObjectParents</b>	
2118	Description:	Can get the parent folders of the object. ( <code>getObjectParents</code> )
2119	Base Object:	cmis:document, cmis:folder, cmis:policy
2120	Operand	Object
2121	Key:	canGetObjectParents.Object
2122	Permission:	Read
2123		
2124	<b>Object Services:</b>	
2125	<b>canCreateDocument</b>	
2126	Description:	Can create a cmis:document Object in the folder ( <code>createDocument</code> )
2127	Base Object:	cmis:folder
2128	Operand:	Folder
2129	Key:	canCreateDocument.Folder
2130	Permission:	Read
2131		
2132	<b>canCreateFolder</b>	

2133	Description:	Can create a cmis:folder Object as a child of the specified folder
2134	(createFolder)	
2135	Base Object:	cmis:folder
2136	Operand:	Folder
2137	Key:	canCreateFolder.Folder
2138	Permission:	Read
2139		
2140	<b>canCreateRelationship</b>	
2141	Description:	Can create a Relationship in which this Object is a source
2142	(createRelationship)	
2143	Base Object:	cmis:document, cmis:folder
2144	Operand:	Object
2145	Key:	canCreateRelationship.Source
2146	Permission:	Read
2147		
2148	<b>canCreateRelationship</b>	
2149	Description:	Can create a Relationship in which this Object is a target
2150	(createRelationship)	
2151	Base Object:	cmis:document, cmis:folder
2152	Operand:	Object
2153	Key:	canCreateRelationship.Target
2154	Permission:	Read
2155		
2156	<b>canGetProperties</b>	
2157	Description:	Can read the properties of this object (getProperties)
2158	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2159	Operand:	Object
2160	Key:	canGetProperties.Object
2161	Permission:	Read
2162		
2163	<b>canGetRenditions</b>	
2164	Description:	Can retrieve the renditions of this object (getRenditions)
2165	Base Object:	cmis:document
2166	Operand:	Object
2167	Key:	canGetRenditions.Object
2168	Permission:	Read
2169		
2170	<b>canGetContentStream</b>	
2171	Description:	Can get the content stream for the Document object
2172	(getContentStream)	
2173	Base Object:	cmis:document
2174	Operand:	Object

2175	Key:	canGetContentStream.Object
2176	Permission:	Read
2177		
2178	<b>canUpdateProperties</b>	
2179	Description:	Can update the properties of this object ( <code>updateProperties</code> )
2180	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2181	Operand:	Object
2182	Key:	canUpdateProperties.Object
2183	Permission:	Write
2184		
2185	<b>canMoveObject</b>	
2186	Description:	Can move the object ( <code>moveObject</code> )
2187	Base Object:	cmis:document, cmis:folder, cmis:policy
2188	Operand:	Object
2189	Key:	canMoveObject.Object
2190	Permission:	Write
2191		
2192	<b>canMoveObject</b>	
2193	Description:	Can move an object into this folder ( <code>moveObject</code> )
2194	Base Object:	cmis:folder
2195	Operand:	Folder
2196	Key:	canMoveObject.Target
2197	Permission:	Read
2198		
2199	<b>canMoveObject</b>	
2200	Description:	Can move an object from this folder ( <code>moveObject</code> )
2201	Base Object:	cmis:folder
2202	Operand:	Folder
2203	Key:	canMoveObject.Source
2204	Permission:	Read
2205		
2206	<b>canDeleteObject</b>	
2207	Description:	Can delete this object ( <code>deleteObject</code> )
2208	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2209	Operand:	Object
2210	Key:	canDelete.Object
2211	Permission:	Write
2212		
2213	<b>canDeleteObject</b>	
2214	Description:	Can delete an object that is a child of this folder ( <code>deleteObject</code> )
2215	Base Object:	cmis:folder

2216	Operand:	Folder
2217	Key:	canDelete.Folder
2218	Permission:	Read
2219		
2220	<b>canSetContentStream</b>	
2221	Description:	Can set the content stream for the Document object
2222		(setContentStream)
2223	Base Object:	cmis:document
2224	Operand:	Object
2225	Key:	canSetContentStream.Document
2226	Permission:	Write
2227		
2228	<b>canDeleteContentStream</b>	
2229	Base Object:	cmis:document
2230	Action:	Can delete the content stream for the Document object
2231		(deleteContentStream)
2232	Operand:	Object
2233	Key:	canDeleteContentStream.Document
2234	Permission:	Write
2235		
2236	<b>canDeleteTree</b>	
2237	Base Object:	cmis:folder
2238	Action:	Can delete the folder and all contained objects (deleteTree)
2239	Operand:	Object
2240	Key:	canDeleteTree.Folder
2241	Permission:	Write
2242		
2243	<b>Filing Services:</b>	
2244	<b>canAddObjectToFolder</b>	
2245	Description:	Can file the document in a folder (addObjectToFolder)
2246	Base Object:	cmis:document, cmis:policy
2247	Operand:	Object
2248	Key:	canAddObjectToFolder.Object
2249	Permission:	Read
2250		
2251	<b>canAddObjectToFolder</b>	
2252	Description:	Can file a document in the specified folder (addObjectToFolder)
2253	Base Object:	cmis:document, cmis:policy
2254	Operand:	Object
2255	Key:	canAddObjectToFolder.Folder
2256	Permission:	Read
2257		

2258	<b>canRemoveObjectFromFolder</b>	
2259	Description:	Can unfile the specified document from a folder
2260	(removeObjectFromFolder)	
2261	Base Object:	cmis:document, cmis:policy
2262	Operand:	Object
2263	Key:	canRemoveObjectFromFolder.Object
2264	Permission:	Read
2265		
2266	<b>canRemoveObjectFromFolder</b>	
2267	Description:	Can unfile a document from the specified folder
2268	(removeObjectFromFolder)	
2269	Base Object:	cmis:document, cmis:policy
2270	Operand:	Object
2271	Key:	canRemoveObjectFromFolder.Folder
2272	Permission:	Read
2273		
2274	<b>Versioning Services:</b>	
2275	<b>canCheckOut</b>	
2276	Description:	Can check out the Document object (checkOut)
2277	Base Object:	cmis:document
2278	Operand:	Object
2279	Key:	canCheckOut.Document
2280	Permission:	Write
2281		
2282	<b>canCancelCheckOut</b>	
2283	Description:	Can cancel the check out the Document object (cancelCheckOut)
2284	Base Object:	cmis:document
2285	Operand:	Object
2286	Key:	canCancelCheckOut.Document
2287	Permission:	Write
2288		
2289	<b>canCheckIn</b>	
2290	Description:	Can check in the Document object (checkIn)
2291	Base Object:	cmis:document
2292	Operand:	Object
2293	Key:	canCheckIn.Document
2294	Permission:	Write
2295		
2296	<b>canGetAllVersions</b>	



2297	Description:	Can get the version series for the Document object ( <code>getAllVersions</code> )
2298	Base Object:	cmis:document
2299	Operand:	Object
2300	Key:	canGetAllVersions.Document
2301	Permission:	Read

2302

#### 2303 Relationship Services:

##### 2304 **canGetObjectRelationships**

2305	Description:	Can get the relationship in which this object is a source/target ( <code>getObjectRelationships</code> )
2306		
2307	Base Object:	cmis:document, cmis:folder, cmis:policy
2308	Operand:	Object
2309	Key:	canGetObjectRelationships.Object
2310	Permission:	Read

2311

#### 2312 Policy Services:

##### 2313 **canApplyPolicy**

2314	Description:	Can apply a policy to the Object ( <code>applyPolicy</code> )
2315	Base Object:	cmis:document, cmis:folder
2316	Operand:	Object
2317	Key:	canApplyPolicy.Object
2318	Permission:	Read

2319

##### 2320 **canApplyPolicy**

2321	Description:	Can apply the specified policy to an Object ( <code>applyPolicy</code> )
2322	Base Object:	cmis:policy
2323	Operand:	Object
2324	Key:	canApplyPolicy.Policy
2325	Permission:	Read

2326

##### 2327 **canRemovePolicy**

2328	Description:	Can remove a policy from the specified Object ( <code>removePolicy</code> )
2329	Base Object:	cmis:document, cmis:folder
2330	Operand:	Object
2331	Key:	canRemovePolicy.Object
2332	Permission:	Read

2333

##### 2334 **canRemovePolicy**

2335	Description:	Can remove the specified policy from an Object ( <code>removePolicy</code> )
2336	Base Object:	cmis:document, cmis:folder
2337	Operand:	cmis:policy

2338	Key:	canRemovePolicy.Policy
2339	Permission:	Read
2340		
2341	<b>canGetAppliedPolicies</b>	
2342	Description:	Can get the list of Policies applied to the Object
2343		(getAppliedPolicies)
2344	Base Object:	cmis:document, cmis:folder
2345	Operand:	Object
2346	Key:	canGetAppliedPolicies.Object
2347	Permission:	Read
2348		
2349	<b>ACL Services:</b>	
2350	<b>canGetACL</b>	
2351	Description:	Can get ACL for object (getACL)
2352	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2353	Operand:	Object
2354	Key:	canGetACL.Object
2355	Permission:	Read
2356		
2357	<b>canApplyACL</b>	
2358	Description:	Can apply ACL to this object (applyACL)
2359	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2360	Operand:	Object
2361	Key:	canApplyACL.Object
2362	Permission:	Write

2363

## 23642.1.9 Versioning

2365CMIS supports versioning of Document objects. Folder objects, relationship objects, and policy objects  
2366cannot be versioned.

2367Whether or not a Document object is versionable (i.e. whether or not operations performed on the object  
2368via the Versioning Services MUST be allowed) is specified by the “*versionable*” attribute on its Object-  
2369type.

2370A **version** of a Document object is an explicit/“deep” copy of the object, preserving its state at a certain  
2371point in time. Each version of a Document object is itself a Document object, i.e. has its own *ObjectId*,  
2372property values, MAY be acted upon using all CMIS services that act upon Document objects, etc.

### 23732.1.9.1 Version Series

2374A **version series** for a Document object is a transitively closed collection of all Document objects that  
2375have been created from an original Document in the Repository. Each version series has a unique,  
2376system-assigned, and immutable **version series ID**.

2377The version series has transitive closure -- that is, if object B is a version of object A, and object C is a  
2378version of object B, then object C is also a version of object A. The objects in a version series can be  
2379conceptually sequenced by their respective *CreationDate* properties.

2380Additionally, the repository MAY expose a textual **VersionLabel** that describes to a user the position of  
2381an individual object with respect to the version series. (For example, version 1.0).

2382**Note:** A Document object that is NOT versionable will always have a single object in its Version Series. A  
2383versionable Document object MAY have one or more objects in its Version Series.

### 23842.1.9.2 Latest Version

2385The version that has the most recent *LastModificationDate* is called the **Latest Version** of the series, or  
2386equivalently, the latest version of any Document object in the series.

2387When the latest version of a version series is deleted, a previous version (if there is one) becomes the  
2388latest version.

#### 23892.1.9.2.1 Behavioral constraints on non-Latest Versions

2390Repositories NEED NOT allow the non-latest versions in a Version Series to be updated, queried, or  
2391searched.

### 23922.1.9.3 Major Versions

2393A Document object in a Version Series MAY be designated as a **Major Version**.

2394The CMIS specification does not define any semantic/behavioral differences between Major and non-  
2395Major versions in a Version Series. Repositories may enforce/apply additional constraints or semantics for  
2396Major versions, if the effect on CMIS services remains consistent with an allowable behavior of the CMIS  
2397model.

2398If the Version Series contains one or more Major versions, the one that has the most recent  
2399*LastModificationDate* is the **Latest Major Version** of the version series.

2400(Note that while a Version Series MUST always have a *Latest Version*, it NEED NOT have a *Latest Major*  
2401*Version*.)

2402When the latest major version is deleted, a previous major version (if there is one) becomes the latest  
2403major version.

## 24042.1.9.4 Services that modify Version Series

### 24052.1.9.4.1 Checkout

2406A new version of a versionable Document object is created when the *checkIn* service is invoked on the  
2407Private Working copy (PWC) of this object. A PWC is created by invoking *checkOut* on a versionable  
2408Document object. A repository MAY allow *any* Document object in a version series to be checked out, or  
2409MAY only allow the *Latest Version* to be checked out.

2410 The effects of invoking the *checkout* service MUST be as follows:

- 2411 • A new Document object, referred to herein as the **Private Working Copy (PWC)**, is created.
  - 2412 ○ The PWC NEED NOT be visible to users who have permissions to view other Document
  - 2413 objects in the Version Series.
  - 2414 ○ Until it is checked in (using the *checkIn* service), the PWC MUST NOT be considered the
  - 2415 *LatestMajorVersion* in the Version Series.
  - 2416 ○ The property values for the PWC SHOULD be identical to the properties of the Document
  - 2417 object on which the *checkout* service was invoked. Certain properties such as *cmis:objectId*
  - 2418 may be different. Properties such as *cmis:creationDate* most likely will be different. The
  - 2419 content-stream of the PWC MAY be identical to the content-stream of the Document object
  - 2420 on which the *checkout* service was invoked, or MAY be “not set”.

2421After a successful *checkout* operation is completed, and until such time when the PWC is deleted (via the  
2422*cancelCheckOut* service) or checked-in (via the *checkIn*) service, the effects on other Documents in the  
2423Version Series MUST be as follows:

- 2424 • The repository MUST throw an exception if the *checkout* service is invoked on any Document in
- 2425 the Version Series. (I.e. there can only be one PWC for a version series at a time.)
- 2426 • The value of the *cmis:isVersionSeriesCheckedOut* property MUST be TRUE.
- 2427 • The value of the *cmis:versionSeriesCheckedOutBy* property MAY be set to a value indicating
- 2428 which user created the PWC. (The Repository MAY still show the “not set” value for this
- 2429 property.)
- 2430 • The value of the *cmis:versionSeriesCheckedOutId* property MAY be set to the *ObjectId* of the
- 2431 PWC. (The Repository MAY still show the “not set” value for this property).
- 2432 • The repository MAY prevent operations that modify or delete the other Documents in the Version
- 2433 Series.

### 24342.1.9.4.2 Updates to the Private Working Copy

2435If the repository supports the optional “PWCUpdatable” capability, then the repository MUST allow  
2436authorized users to modify the PWC Object using the Object services (e.g. *UpdateProperties*).

2437If the repository does NOT support the “PWCUpdatable” capability, then the PWC object can only be  
2438modified as part of the *checkIn* service call.

### 24392.1.9.4.3 Discarding Check out

2440An authorized user MAY discard the check-out using the *cancelCheckOut* service on any Document in  
2441the Version Series or by using the *deleteObject* service on the PWC Object. The effects of discarding a  
2442check-out MUST be as follows:

- 2443 • The PWC Object MUST be deleted.
- 2444 • For all other Documents in the Version Series:
  - 2445 ○ The value of the *cmis:isVersionSeriesCheckedOut* property MUST be FALSE.
  - 2446 ○ The value of the *cmis:versionSeriesCheckedOutBy* property MUST be “not set”.
  - 2447 ○ The value of the *cmis:versionSeriesCheckedOutId* property MUST be “not set”.

2448           ○ The repository MUST allow authorized users to invoke the *checkout* service.

#### 2449**2.1.9.4.4 Checkin**

2450An authorized user/application MAY “check in” the Private Working Copy object via the *checkIn* service.

2451The *checkIn* service allows users/applications to provide update property values and a content-stream for  
2452the PWC object.

2453The effects of the *checkIn* service MUST be as follows for successful checkins:

- 2454     • The PWC object MUST be updated as specified by the inputs to the *checkIn* service. (Note that  
2455       for repositories that do NOT support the “PWCUpdatable” property, this is the only way to update  
2456       the PWC object.)
- 2457     • The Document object resulting from the *checkIn* operation MUST be considered the *Latest*  
2458       *Version* in the Version Series.
- 2459     • If the inputs to the *checkIn* service specified that the PWC MUST be a “major version”, then the  
2460       PWC MUST be considered the *Latest Major Version* in the Version Series.
- 2461     • If the checkin returns a new cmis:objectid, then the PWC object MUST disappear if the *checkIn*  
2462       call was successful and the new checked in version will use the new specified id.
- 2463     • For all Documents in the Version Series:
  - 2464       ○ The value of the cmis:isVersionSeriesCheckedOut property MUST be FALSE.
  - 2465       ○ The value of the cmis:versionSeriesCheckedOutBy property MUST be “not set”.
  - 2466       ○ The value of the cmis:versionSeriesCheckedOutId property MUST be “not set”.
  - 2467       ○ The repository MUST allow authorized users to invoke the *checkout* service.

2468*Note:* The Repository MAY change the ID of the PWC upon completion of the *checkin* service invocation.

2469*Note:* A repository MAY automatically create new versions of Document objects without an explicit  
2470invocation of the checkout/checkin services.

#### 2471**2.1.9.5 Versioning Properties on Document Objects**

2472All Document objects will have the following read-only property values pertaining to versioning:

2473

2474     **cmis:isLatestVersion**                             Boolean

2475       TRUE if the Document object is the *Latest Version* in its *Version Series*. FALSE otherwise.

2476

2477     **cmis:isMajorVersion**                             Boolean

2478       TRUE if the Document object is a *Major Version* in its *Version Series*. FALSE otherwise.

2479

2480     **cmis:isLatestMajorVersion**                     Boolean

2481       TRUE if the Document object is the *Latest Major Version* in its *Version Series*. FALSE otherwise.

2482

2483     **cmis:versionLabel**                             String (optional)

2484       Optional textual description the position of an individual object with respect to the version series.  
2485       (For example, version 1.0).

2486

2487     **cmis:versionSeriesId**                             ID

2488       ID of the Version Series for this Object.

2489

2490 **cmis:isVersionSeriesCheckedOut** Boolean  
 2491 TRUE if there currently exists a Private Working Copy for this Version Series. FALSE otherwise  
 2492  
 2493 **cmis:versionSeriesCheckedOutBy** String  
 2494 If IsVersionSeriesCheckedOut is TRUE: then an identifier for the user who created the Private  
 2495 Working Copy. "Not set" otherwise.  
 2496  
 2497 **cmis:versionSeriesCheckedOutId** ID  
 2498 If IsVersionSeriesCheckedOut is TRUE: The Identifier for the Private Working Copy. "Not set"  
 2499 otherwise.  
 2500  
 2501 **cmis:checkinComment** String  
 2502 Textual comment associated with the given version.

2503 *Note:* Changes made via the Versioning Services that affect the values of these properties MUST NOT  
 2504 constitute modifications to the Document objects in the Version Series (e.g. MUST NOT affect the  
 2505 cmis:lastModificationDate, etc.)

## 2506 2.1.9.6 Document Creation and Initial Versioning State

2507 A repository MAY create new Document objects in a "Private Working Copy" state when they are created  
 2508 via the *createDocument* or *createDocumentFromSource* services. This state is logically equivalent to  
 2509 having a Version Series that contains exactly one object (the PWC) and 0 other documents.

2510 The repository MAY also create new Document objects in a "Major Version" state. This state is logically  
 2511 equivalent to having a Version Series that contains exactly one Major Version and 0 other documents.

2512 The repository MAY also create new Document objects in a "Non-Major Version" state. This state is  
 2513 logically equivalent to having a Version Series that contains exactly one Non-Major Version and 0 other  
 2514 documents.

2515 If the repository does not support versioning the repository MAY create new Document objects in a "Major  
 2516 Version" state.

## 2517 2.1.9.7 Version Specific/Independent membership in Folders

2518 Repositories MAY treat membership of a Document object in a folder collection as "version-specific" or  
 2519 "version-independent".

2520 Repositories MUST indicate whether they support version-specific membership in a folder via the  
 2521 "VersionSpecificFiling" optional capability flag.

2522 If the repository is treating folder collection membership as "version-independent", then:

- 2523 • Moving or Filing a Document Object into a folder MUST result in ALL Documents in the Version  
 2524 Series being moved/filed into the folder.
- 2525 • The Repository MAY return only the latest-version OR latest major-version Document object in a  
 2526 version series in the response to Navigation service requests (*getChildren*, *getDescendants*), and  
 2527 NEED NOT return other Document Objects filed in the folder that are in the Version Series.

2528 If the repository is treating folder collection membership as "version-specific", then moving or Filing a  
 2529 Document Object into a folder MUST NOT result in other Documents in the Version Series being  
 2530 moved/filed.

## 25312.1.9.8 Version Specific/Independent membership in Relationships

2532A relationship object MAY have either a version-specific or version-independent binding to its source  
2533and/or target objects. This behavior MAY vary between repositories and between individual relationship  
2534types defined for a Repository.

2535If a relationship object has a version-independent binding to its source/target object, then:

- 2536 • The getObjectRelationships service invoked on a Document Object MUST return the relationship if  
2537 Relationship was source/target is set to ANY Document Object in the Version Series.

2538If a relationship object has a version-specific binding to its source/target object, then:

- 2539 • The getObjectRelationships service invoked on a Document Object MUST return the relationship if  
2540 Relationship was source/target is set to the ID of the Document Object on which the service was  
2541 invoked.

## 25422.1.9.9 Versioning visibility in Query Services

2543Repositories MAY include non-latest-versions of Document Objects in results to the Discovery Services  
2544(query).

2545Repositories MUST indicate whether they support querying for non-latest-versions via the  
2546“AllVersionsSearchable” optional capability flag.

2547If “AllVersionsSearchable” is TRUE then the Repository MUST include in the query results ANY  
2548Document Object in the Version Series that matches the query criteria. (subject to other query constraints  
2549such as security.)

2550Additionally, repositories MAY include Private Working Copy objects in results in results to the Discovery  
2551Services (query).

2552Repositories MUST indicate whether they support querying for Private Working Copy objects via the  
2553“PWCSearchable” optional capability flag.

2554If “PWCSearchable” is TRUE then the Repository MUST include in the query results ANY Private Working  
2555Copy Document Objects that matches the query criteria (subject to other query constraints such as  
2556security.)

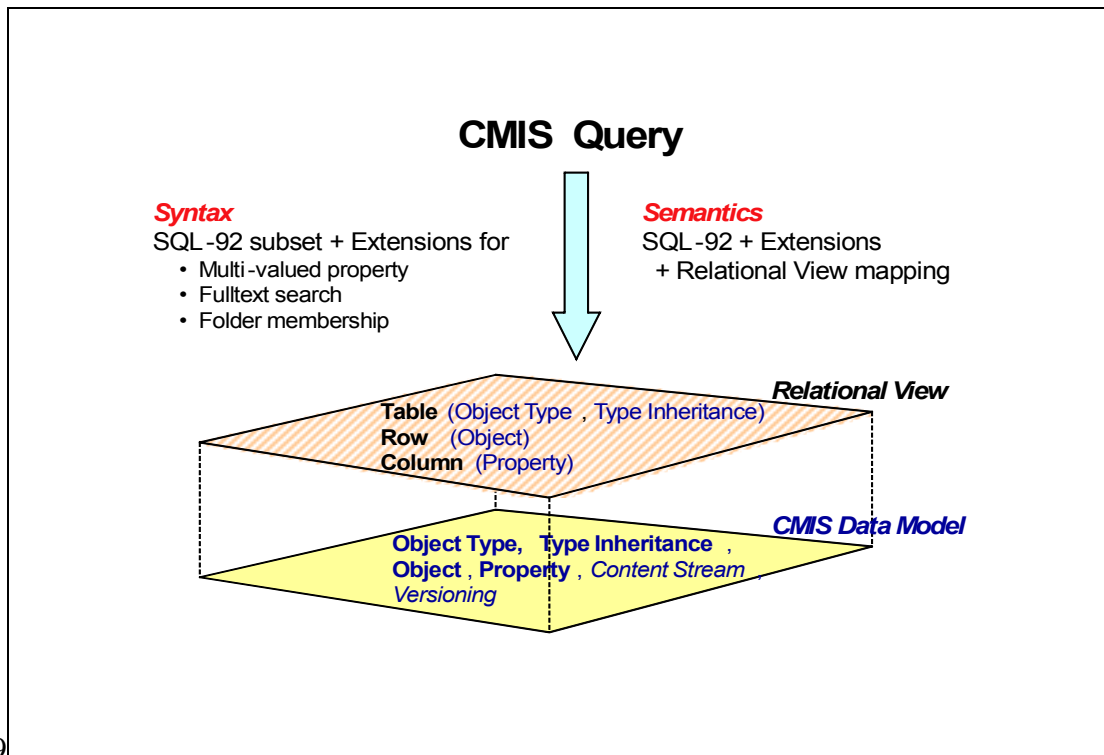
2557If “PWCSearchable” is FALSE then the Repository MUST NOT include in the query results ANY Private  
2558Working Copy Document Objects that match the query criteria (subject to other query constraints such as  
2559security.)

## 25602.1.10 Query

2561CMIS provides a type-based query service for discovering objects that match specified criteria, by  
2562defining a read-only projection of the CMIS data model into a *Relational View*.

2563Through this relational view, queries may be performed via a simplified SQL SELECT statement. This  
2564query language is based on a subset of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language  
2565SQL), with a few extensions to enhance its filtering capability for the CMIS data model, such as existential  
2566quantification for multi-valued property, full-text search, and folder membership. Other statements of the  
2567SQL language are not adopted by CMIS. The semantics of this query language is defined by the SQL-92  
2568standard, plus the extensions, in conjunction with the model mapping defined by CMIS’s relational view.





2569

## 25702.1.10.1 Relational View Projection of the CMIS Data Model

2571The relational view of a CMIS repository consists of a collection of virtual tables that are defined on top of  
2572the CMIS data model. This relational view is used for query purposes only.

2573In this relational view a **Virtual Table** is implicitly defined for each *queryable* Object-Type defined in the  
2574repository. (Non-queryable Object-Types are NOT exposed through this Relational View.)

2575In each **Virtual Table**, a **Virtual Column** is implicitly defined for each property defined in the Object-Type  
2576Definition AND for all properties defined on ANY ancestor-type of the Object-Type but NOT defined in the  
2577Object-Type definition. Virtual Columns for properties defined on ancestor-types of the Object-type but  
2578NOT defined in the Object-Type definition MUST contain the SQL NULL value. Virtual Columns for  
2579properties whose value is “not set” MUST contain the SQL NULL value.

2580An object-type’s *queryName* attribute is used as the table name for the corresponding virtual table, and a  
2581property’s *queryName* attribute is used as the column name for the corresponding table column. Please  
2582see the restrictions on *queryName* in the appropriate data model section.

2583The Virtual Column for a multi-valued property MUST contain a single list value that includes all values of  
2584the property.

### 25852.1.10.1.1 Object-Type Hierarchy in the Relational View Projection

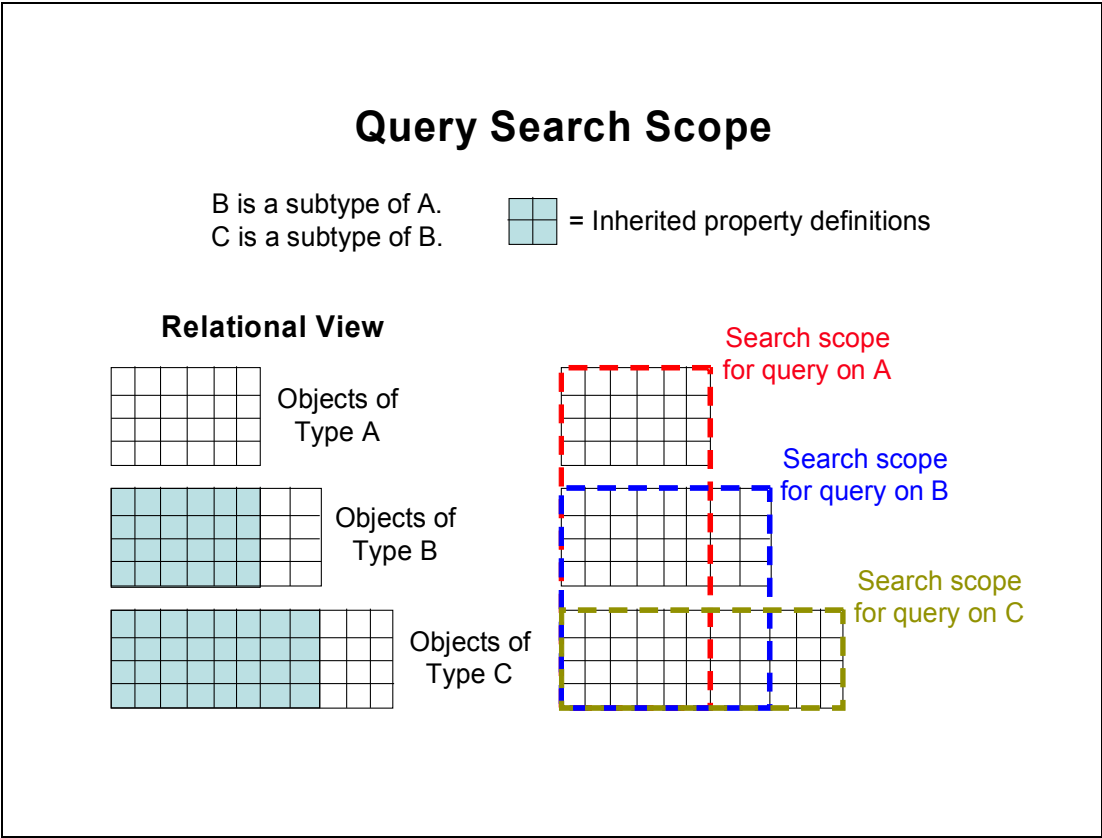
2586The Relational View projection of the CMIS Data Model ensures that the Virtual Table for a particular  
2587Object-type is a complete super-set of the Virtual Table for any and all of its ancestor types.

2588Additionally, an Object-Type definition’s “*includedInSupertypeQuery*” specifies whether objects of that  
2589Object-Type MUST be included in the Virtual Table for any of its ancestor types. If the  
2590“*includedInSupertypeQuery*” attribute of the Object-Type is FALSE, then objects of that Object-Type  
2591MUST NOT be included in the Virtual Table for any of its ancestor types.



2592Thus the Virtual Table for an Object-type includes a row not only for each Object of that type, but all  
2593Objects of any of that Object-types' Descendant Types for which the *"includedInSupertypeQuery"* attribute  
2594is TRUE.

2595But since the Virtual Table will include only columns for properties defined in the Object-Type underlying  
2596the Virtual Table, a row that is a query result representing an Object of a Descendant Type can only  
2597include those columns for properties defined on the Object-Type underlying the Virtual Table.



2598

2599**2.1.10.1.2 Content Streams**

2600Content-streams are NOT exposed through this relational view.

2601**2.1.10.1.3 Result Set**

2602When a query is submitted, a set of pseudo CMIS objects will be returned. These pseudo objects are  
2603comprised of the properties specified in the select clause of the query statement.

2604For each property in each object in the result set, the Repository MUST include the property definition ID  
2605as well as either the query name (if no alias is used) or the alias in place of the query name (if an alias is  
2606used).

2607If the select clause of the query statement contains properties from a single type reference then the  
2608repository MAY represent these pseudo-objects with additional object information.

2609**2.1.10.2 Query Language Definition**

2610This query languages is based on a subset of the SQL-92 grammar. CMIS-specific language extensions  
2611to SQL-92 are called out explicitly.

2612The basic structure of a CMIS query is a SQL statement that MUST include the following clauses:

- 2613 • **SELECT [virtual columns]:** This clause identifies the set of virtual columns that will be included  
2614 in the query results for each row.
- 2615 • **FROM [Virtual Table Names]:** This clause identifies which Virtual Table(s) the query will run  
2616 against.
- 2617 Additionally, a CMIS query MAY include the following clauses:
- 2618 • **WHERE [conditions]:** This clause identifies the constraints that rows MUST satisfy to be  
2619 considered a result for the query.
- 2620 • **ORDER BY [sort specification]:** This clause identifies the order in which the result rows MUST  
2621 be sorted in the result row set.

## 2622 2.1.10.2.1 BNF Grammar

2623 This BNF grammar is a “subset” of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language  
2624 SQL), except for some production alternatives. Specifically, except for these extensions, the following  
2625 production rules are derived from the SQL-92 grammar. The non-terminals used in this grammar are also  
2626 borrowed from the SQL-92 grammar without altering their semantics. Accordingly, the non-terminal  
2627 <column name> is used for single-valued properties only so that the semantics of SQL can be preserved  
2628 and borrowed. This approach not only facilitates comparison of the two query languages, and simplifies  
2629 the translation of a CMIS query to a SQL query for a RDBMS-based implementation, but also allows  
2630 future expansion of this query language to cover a larger subset of SQL with minimum conflict. The CMIS  
2631 extensions are introduced primarily to support multi-valued properties and full-text search, and to test  
2632 folder membership. Multi-valued properties are handled separately from single-valued properties, using  
2633 separate non-terminals and separate production rules to prevent the extensions from corrupting SQL-92  
2634 semantics.

2635  
2636<CMIS 1.0 query statement> ::= <simple table> [ <order by clause> ]  
2637<simple table> ::= SELECT <select list> <from clause> [ <where clause> ]  
2638<select list> ::= "\*" |  
2639 | <select sublist> [ { " , " <select sublist> }... ]  
2640<select sublist> ::= <value expression> [ [ AS ] <column name> ]  
2641 | <qualifier> "." |  
2642 | <multi-valued-column reference>  
2643<value expression> ::= <column reference> | <numeric value function>  
2644<column reference> ::= [ <qualifier> "." ] <column name>  
2645<multi-valued-column reference> ::= [ <qualifier> "." ] <multi-valued-column name>  
2646<numeric value function> ::= SCORE()  
2647<qualifier> ::= <table name> | <correlation name>  
2648<from clause> ::= FROM <table reference>  
2649<table reference> ::= <table name> [ [ AS ] <correlation name> ]  
2650 | <joined table>  
2651<joined table> ::= "(" <joined table> ")"  
2652 | <table reference> [ <join type> ] JOIN <table reference> <join specification>  
2653<join type> ::= INNER | LEFT [ OUTER ]  
2654<join specification> ::= ON <column reference> "=" <column reference>  
2655<where clause> ::= WHERE <search condition>  
2656<search condition> ::= <boolean term> | <search condition> OR <boolean term>  
2657<boolean term> ::= <boolean factor> | <boolean term> AND <boolean factor>  
2658<boolean factor> ::= [ NOT ] <boolean test>  
2659<boolean test> ::= <predicate> | "(" <search condition> ")"  
2660<predicate> ::= <comparison predicate> | <in predicate> | <like predicate> | <null predicate>  
2661 | <quantified comparison predicate> | <quantified in predicate>  
2662 | <text search predicate> | <folder predicate>  
2663<comparison predicate> ::= <value expression> <comp op> <literal>  
2664<comp op> ::= "=" | "<" | "<" | ">" | "<=" | ">=" |  
2665<literal> ::= <signed numeric literal> | <character string literal> | <datetime literal> | <boolean literal>  
2666<in predicate> ::= <column reference> [ NOT ] IN "(" <in value list> ")"  
2667<in value list> ::= <literal> [ { " , " <literal> }... ]  
2668<like predicate> ::= <column reference> [ NOT ] LIKE <character string literal>  
2669<null predicate> ::= { <column reference> | <multi-valued-column reference> } IS [ NOT ] NULL  
2670<quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>  
2671<quantified in predicate> ::= ANY <multi-valued-column reference> [ NOT ] IN "(" <in value list> ")"  
2672<text search predicate> ::= CONTAINS "("  
2673 [ <qualifier> " , " ] <quote> <text search expression> <quote> ")"  
2674<folder predicate> ::= { IN\_FOLDER | IN\_TREE } "(" [ <qualifier> " , " ] <folder id> ")"  
2675<order by clause> ::= ORDER BY <sort specification> [ { " , " <sort specification> }... ]



### 27182.1.10.2.3.1 Join Support

2719CMIS repositories MAY support the use of SQL JOIN queries, and MUST indicate their support level  
2720using the [Optional Capability](#) attribute “capabilityJoin”.

- 2721 • If the Repository’s value for the capabilityJoin attribute is **none**, then no JOIN clauses can be  
2722 used in queries.
- 2723 • If the Repository’s value for the capabilityJoin attribute is **inneronly**, then only inner JOIN clauses  
2724 can be used in queries.
- 2725 • If the Repository’s value for the capabilityJoin attribute is **innerandouter**, then inner and/or outer  
2726 JOIN clauses can be used in queries.

2727Only explicit joins using the “JOIN” keyword is supported. Queries MUST NOT include implicit joins as  
2728part of the WHERE clause of a CMIS query.

2729CMIS queries MUST only support join operations using the “equality” predicate on single-valued  
2730properties.

### 27312.1.10.2.4 WHERE Clause

2732This clause identifies the constraints that rows MUST satisfy to be considered a result for the query.

2733All column names MUST be valid “queryName” or their aliased values for properties that are defined as  
2734“queryable” in the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

2735Properties are defined to not support a “null” value, therefore the <null predicate> MUST be interpreted as  
2736testing the not set or set state of the specified property.

### 27372.1.10.2.4.1 Comparisons permitted in the WHERE clause.

2738SQL’s simple comparison predicate, IN predicate, and LIKE predicate are supported, for single-valued  
2739properties only (so that SQL’s semantics is preserved). Boolean conjunction (AND), disjunction (OR), and  
2740negation (NOT) of predicates are also supported.

2741Repositories SHOULD support the comparisons for the property types as described in the list below.

2742Repositories MAY support additional comparisons and operators. Any additional operators not specified  
2743are repository-specific:

2744

2745<Property Type>

2746Supported Operators: <List of Operators supported on Type>

2747Supported Literal: <Supported type of Literal in comparison>

2748

2749String (Single)

2750Supported Operators: =, <>, [NOT] LIKE

2751Supported Literal: String

2752

2753String (IN)

2754Supported Operators: [NOT] IN

2755Supported Literal: List of Strings

2756

2757Decimal

2758Supported Operators: =, <>, <, <=, >, >=

2759Supported Literal: Decimal

2760  
2761Decimal (IN)  
2762Supported Operators: [NOT] IN  
2763Supported Literal: List of Decimal  
2764  
2765Integer  
2766Supported Operators: =, <>, <, <=, >, >=  
2767Supported Literal: Integer  
2768  
2769Integer (IN)  
2770Supported Operators: [NOT] IN  
2771Supported Literal: List of Integer  
2772  
2773Boolean  
2774Supported Operators: =  
2775Supported Literal: <boolean literal>  
2776  
2777DateTime  
2778Supported Operators: =, <>, <\*, <=\*, >\*, >=\*  
2779Supported Literal: <datetime literal>  
2780\* - comparison is based on chronological before or after date.  
2781  
2782DateTime (IN)  
2783Supported Operators: [NOT] IN  
2784Supported Literal: List of <datetime literal>'s  
2785  
2786ID  
2787Supported Operators: =, <>  
2788Supported Literal: String  
2789  
2790ID (IN)  
2791Supported Operators: [NOT] IN  
2792Supported Literal: List of strings  
2793  
2794URI  
2795Supported Operators: =, <>  
2796Supported Literal: String  
2797  
2798URI (IN)  
2799Supported Operators: [NOT] IN  
2800Supported Literal: List of strings  
2801

2802URI

2803Supported Operators: [NOT] LIKE

2804Supported Literal: String

2805

2806Operations on the SCORE() output MUST be treated the same as decimal operations.

2807

2808When using properties in a join statement, comparison MUST be allowed on properties of the same types

2809as defined by the table above. Repositories MAY extend this behavior.

2810

2811The ANY operation argument MUST be one of the properties found in the table above which supports

2812equality operations

#### 2813**2.1.10.2.4.2 Multi-valued property support (SQL-92 Extension)**

2814The CMIS query language includes several new non-terminals to expose semantics for querying multi-

2815valued properties, in a way that does not alter the semantics of existing SQL-92 production rules.

##### 2816**2.1.10.2.4.2.1 Multi-valued column references**

2817**BNF grammar structure:** <Multi-valued-column reference>, <multi-valued-column name>

- 2818 • These are non-terminals defined for multi-valued properties whereas SQL-92's <column  
2819 reference> and <column name> are retained for single-valued properties only. This is to preserve  
2820 the single-value semantics of a regular "column" in the SQL-92 grammar.

##### 2821**2.1.10.2.4.2.2 <Quantified comparison predicate>**

2822The SQL-92 production rule for <quantified comparison predicate> is extended to accept a multi-valued  
2823property in place of a <table subquery>. This operation is restricted to equality tests only.

2824

2825<Table subquery> is not supported in CMIS-SQL.

2826

2827The SQL-92 <quantifier> is restricted to ANY only.

2828

2829The SQL-92 <row value constructor> is restricted to a literal only.

2830

#### **Example:**

```
2831 SELECT      Y.CLAIM_NUM, X.PROPERTY_ADDRESS, Y.DAMAGE_ESTIMATES
2832 FROM POLICY AS X JOIN CLAIMS AS Y ON ( X.POLICY_NUM = Y.POLICY_NUM )
2833 WHERE        ( 100000 = ANY Y.DAMAGE_ESTIMATES )
```

2834 *(Note: DAMAGE\_ESTIMATES is a multi-valued Integer property.)*

##### 2835**2.1.10.2.4.2.3 IN/ANY Predicate**

2836**BNF grammar structure:** <Quantified in predicate>

2837

2838CMIS-SQL exposes a new IN predicate defined for a multi-valued property. It is modeled after the SQL-92

2839IN predicate, but since the entire predicate is different semantically, it has its own production rule in the

2840BNF grammar below.

2841

2842The quantifier is restricted to ANY. The predicate MUST be evaluated to TRUE if at least one of the  
2843property's values is (or, is not, if NOT is specified) among the given list of literal values. Otherwise the  
2844predicate is evaluated to FALSE.

2845

2846The ANY operation argument MUST be one of the properties found in the comparison list above which  
2847supports IN operations.

2848       **Example:**

```
2849      SELECT      *  
2850      FROM  CAR_REVIEW  
2851      WHERE      (MAKE = "buick" ) OR  
2852                ( ANY FEATURES IN ("NAVIGATION SYSTEM", "SATELLITE RADIO", "MP3") )
```

2853       *(Note: FEATURES is a multi-valued String property.)*

#### 2854**2.1.10.2.4.3   CONTAINS() predicate function (CMIS-SQL Extension)**

2855**BNF grammar structure::** CONTAINS ( [ <qualifier> ,] <text search expression> )

2856**Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided  
2857by a Repository ([See previous section.](#))

2858**Inputs:**

2859       **<Qualifier>**

2860       The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the  
2861       FROM clause for the query.

- 2862
  - If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
  - 2863       Table, but a repository MAY ignore the value of the parameter.
  - 2864       • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
  - 2865       throw an exception if the qualifier is not specified.

2866       **<Text Search Expression>**

2867       The <text search expression> parameter MUST be a character string literal in quotes, specifying  
2868       the full-text search criteria.

2869       The Text Search Expression may be a set of terms or phrases with an optional '-' to signal  
2870       negation. A phrase is defined as a word or group of words. A group of words must be  
2871       surrounded by quotes to be considered a single phrase.  
2872

2873

2874       Terms separated by whitespace are AND'ed together.

2875       Terms separated by "OR" are OR'ed together

2876       Implicit "AND" has higher precedence than "OR"

2877       Within a word or phrase, each double quote must also be escaped by a preceding backslash "\"

2878**Return value:**

2879       The predicate returns a Boolean value.

2880       The predicate MUST return TRUE if the object is considered by the repository as "relevant" with  
2881       respect to the given <text search expression> parameter.

2882       The predicate MUST return FALSE if the object is considered by the repository as not "relevant"  
2883       with respect to the given <text search expression> parameter.

2884**Constraints:**



2885 At most one CONTAINS() function MUST be included in a single query statement. The repository  
2886 MUST throw an exception if more than one CONTAINS() function is found.

2887  
2888 The return value of the CONTAINS() function MAY only be included conjunctively (ANDed) with the  
2889 aggregate of all other predicates, if there is any, in the WHERE clause.

#### 2890 2.1.10.2.4.4 SCORE() predicate function

2891 **BNF grammar structure:** SCORE ( )

2892 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided  
2893 by a Repository ([See previous section.](#))

2894 **Inputs:** No inputs MUST be provided for this predicate function.

2895 **Return value:**

2896 The SCORE() predicate function returns a decimal value in the interval [0,1] .

2897 A repository MUST return the value 0 if the object is considered by the repository as having  
2898 absolutely no relevance with respect to the CONTAINS() function specified in the query.

2899 A repository MUST return the value 1 if the object is considered by the repository as having  
2900 absolutely complete relevance with respect to the CONTAINS() function specified in the query.

2901 **Constraints:**

2902 The SCORE() function MUST only be used in queries that also include a CONTAINS() predicate  
2903 function

2904 The SCORE() function MUST only be used in the SELECT clause of a query. It MUST NOT be  
2905 used in the WHERE clause or in the ORDER BY clauses.

2906 An alias column name defined for the SCORE() function call in the SELECT clause (i.e.,  
2907 "SELECT SCORE() AS column\_name ...") may be used in the ORDER BY clause.

2908 If SCORE() is included in the SELECT clause and an alias column name is not provided, then a  
2909 query name of SEARCH\_SCORE is used for the query output, and the property definition ID is  
2910 repository-specific.

#### 2911 2.1.10.2.4.5 IN\_FOLDER() predicate function

2912 **BNF grammar structure:** IN\_FOLDER( [ <qualifier>, ] <folder id> )

2913 **Usage:** This is a predicate function that tests whether or not a candidate object is a child-object of the  
2914 folder object identified by the given <folder id>.

2915 **Inputs:**

2916 **<qualifier>**

2917 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the  
2918 FROM clause for the query.

- 2919
- If specified, then the predicate SHOULD only be applied to objects in the specified Virtual  
2920 Table, but a repository MAY ignore the value of the parameter.
  - If not specified, applies to the single virtual table. If the query is a join, a server SHOULD  
2921 throw an exception if the qualifier is not specified.

2923 **<folder id>**

2924 The value of this parameter MUST be the ID of a folder object in the repository.

2925 **Return value:**

2926 The predicate function MUST return TRUE if the object is a child-object of the folder specified by  
2927 <folder id>.  
2928 The predicate function MUST return FALSE if the object is a NOT a child-object of the folder  
2929 specified by <folder id>.

#### 2930**2.1.10.2.4.6 IN\_TREE() predicate function**

2931**BNF grammar structure:** IN\_TREE( [ <qualifier>, ] <folder id> )

2932**Usage:** This is a predicate function that tests whether or not a candidate object is a descendant-object of  
2933the folder object identified by the given <folder id>.

2934**Inputs:**

2935 **<qualifier>**

2936 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the  
2937 FROM clause for the query.

2938 • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual  
2939 Table, but a repository MAY ignore the value of the parameter.

2940 • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD  
2941 throw an exception if the qualifier is not specified.

2942 **<folder id>**

2943 The value of this parameter MUST be the ID of a folder object in the repository.

2944**Return value:**

2945 The predicate function MUST return TRUE if the object is a descendant-object of the folder  
2946 specified by <folder id>.

2947 The predicate function MUST return FALSE if the object is a NOT a descendant -object of the  
2948 folder specified by <folder id>.

#### 2949**2.1.10.2.5 ORDER BY Clause**

2950This clause MUST contain a comma separated list of one or more column names.

2951All column names referenced in this clause MUST be valid “queryName” or their aliased values for  
2952properties defined as *orderable* in the Object-type(s) whose Virtual Tables are listed in the FROM clause.

2953Only columns in the SELECT clause MAY be in the ORDER BY clause.

2954Collation rules for the ORDER BY clause are repository specific.

#### 2955**2.1.10.3 Escaping**

2956Repositories MUST support the escaping of characters using a backslash (\) in the query statement. The  
2957backslash character (\) will be used to escape characters *within quoted strings* in the query as follows:

- 2958 1. \" will represent a double-quote (") character
- 2959 2. \' will represent a single-quote(') character
- 2960 3. \\ will represent a backslash (\) character
- 2961 4. Within a LIKE string, \% and \\_ will represent the literal characters % and \_, respectively.
- 2962 5. All other instances of a \ are errors.

#### 2963**2.1.11 Change Log**

2964CMIS provides a “change log” mechanism to allow applications to easily discover the set of changes that  
2965have occurred to objects stored in the repository since a previous point in time. This change log can then  
2966be used by applications such as search services that maintain an external index of the repository to

efficiently determine how to synchronize their index to the current state of the repository (rather than having to query for all objects currently in the repository).

Entries recorded in the change log are referred to below as “change events”.

Note that change events in the change log MUST be returned in ascending order from the time when the change event occurred.

### 2.1.11.1 Completeness of the Change Log

The Change Log mechanism exposed by a repository MAY be able to return an entry for every change ever made to content in the repository, or may only be able to return an entry for all changes made since a particular point in time. This “completeness” level of the change log is indicated via the [optional `changesIncomplete` value found on the `getRepositoryInfo` service](#) response

However, repositories MUST ensure that if an application requests the entire contents of the repository’s change log, that the contents of the change log includes ALL changes made to any object in the repository *after* the first change listed in the change log. (I.e. repositories MAY truncate events from the change log on a “first-in first-out” basis, but not in any other order.)

A Repository MAY record events such as filing/unfiling/moving of Documents as change events on the Documents, their parent Folder(s), or both the Documents and the parent Folders.

### 2.1.11.2 Change Log Token

The primary index into the change log of a repository is the “change log token”. The change log token is an opaque string that uniquely identifies a particular change in the change log.

#### 2.1.11.2.1 “Latest Change Token” repository information

Repositories that support the `changeLogToken` event MUST expose the latest change log token (i.e. the change log token corresponding to the most recent change to any object in the repository) as a property returned by the `getRepositoryInfo` service.

This will enable applications to begin “subscribing” to the change log for a repository by discovering what change log token they should use on a going-forward basis to discover change events to the repository.

### 2.1.11.3 Change Event

A change event represents a single action that occurred to an object in the repository that affected the persisted state of the object.

A Repository that supports the change log capability MUST expose at least the following information for each change object:

- **ID `ObjectId`:** The `ObjectId` of the object to which the change occurred
- **Enum `ChangeType`:** An enumeration that indicates the type of the change. Valid values are:
  - `created`: The object was created.
  - `updated`: The object was updated.
  - `deleted`: The object was deleted
  - `security`: The access control or security policy for the object were changed.
- **<Properties> `properties`:** Additionally, for events of `changeType` “updated”, the repository MAY optionally include the new values of properties on the object (if any).

Repositories MUST indicate whether they include properties for “updated” change events via the [optional `enumCapabilityChanges` capability](#).

## 30082.2 Services

3009Part I of the CMIS specification defines a set of services that are described in a protocol/binding-agnostic  
3010fashion.

3011Every protocol binding of the CMIS specification MUST implement all of the methods described in this  
3012section or explain why the service is not implemented.

3013However, the details of how each service & method is implemented will be described in those protocol  
3014binding specifications.

### 30152.2.1 Common Service Elements

3016The following elements are common across many of the CMIS services.

#### 30172.2.1.1 Paging

3018All of the methods that allow for the retrieval of a collection of CMIS objects support paging of their result  
3019sets except where explicitly stated otherwise. The following pattern is used:

3020**Input Parameters:**

- 3021 • **(optional) Integer maxItems:** This is the maximum number of items to return in a response. The  
3022 repository MUST NOT exceed this maximum. Default is repository-specific.
- 3023 • **(optional) Integer skipCount:** This is the number of potential results that the repository MUST  
3024 skip/page over before returning any results. Defaults to 0.

3025**Output Parameters:**

- 3026 • **Boolean hasMoreItems:** TRUE if the Repository contains additional items after those contained  
3027 in the response. FALSE otherwise. If TRUE, a request with a larger skipCount or larger  
3028 maxItems is expected to return additional results (unless the contents of the repository has  
3029 changed).
- 3030 • **Integer numItems:** If the repository knows the total number of items in a result set, the  
3031 repository SHOULD include the number here. If the repository does not know the number of  
3032 items in a result set, this parameter SHOULD not be set. The value in the parameter MAY NOT  
3033 be accurate the next time the client retrieves the result set or the next page in the result set.

3034If the caller of a method does not specify a value for maxItems, then the Repository MAY select an  
3035appropriate number of items to return, and MUST use the hasMoreItems output parameter to indicate if  
3036any additional results were not returned.

3037Repositories MAY return a smaller number of items than the specified value for maxItems.

3038Each binding will express the above in context and may have different mechanisms for communicating  
3039hasMoreItems and numItems.

#### 30402.2.1.2 Retrieving additional information on objects in CMIS service calls

3041Several CMIS services that return object information have the ability to return dependent object  
3042information as part of their response, such as the Allowable Actions for an object, rendition information,  
3043etc.

3044The CMIS service methods that support returning a result set of objects will include the ability to return  
3045the following object information:

- 3046 • Properties (retrieves a subset instead of additional information)
- 3047 • Relationships
- 3048 • Renditions
- 3049 • ACLs
- 3050 • AllowableActions

3051

3052 This section describes the input parameter & output pattern for those services. All input parameters are  
3053 optional.

#### 3054 2.2.1.2.1 Properties

3055 **Description:** All of the methods that allow for the retrieval of properties for CMIS Objects have a  
3056 "Property Filter" as an optional parameter, which allows the caller to specify a subset of properties for  
3057 Objects that MUST be returned by the repository in the output of the method.

3058 **Optional Input Parameter:**

- 3059 • **String filter:** Value indicating which properties for Objects MUST be returned. Values are:
- 3060     o **Not set:** The set of properties to be returned MUST be determined by the repository.
- 3061     o **A comma-delimited list of property definition Query Names: The properties listed**  
3062       **MUST be returned.**
- 3063     o **""** : All properties MUST be returned for all objects.

3064 Repositories SHOULD return only the properties specified in the property filter.

#### 3065 2.2.1.2.2 Relationships

3066 **Description:** Used to retrieve the relationships in which the object(s) are participating.

3067 **Optional Input Parameter:**

- 3068 • **Enum includeRelationships:** Value indicating what relationships in which the objects returned  
3069 participate MUST be returned, if any. Values are:
- 3070     **none** : No relationships MUST be returned. (Default).
- 3071     **source** : Only relationships in which the objects returned are the source MUST be  
3072       returned.
- 3073     **target** : Only relationships in which the objects returned are the target MUST be  
3074       returned.
- 3075     **both** : Relationships in which the objects returned are the source or the target MUST be  
3076       returned.

3077 **Output Parameter for each object:**

- 3078 • **<Array> Relationships:** A collection of the relationship objects.

#### 3079 2.2.1.2.3 Policies

3080 **Description:** Used to retrieve the policies currently applied to the object(s).

3081 **Optional Input Parameter:**

- 3082 • **Boolean includePolicyIds:** If TRUE, then the Repository MUST return the Ids of the policies  
3083 applied to the object. Defaults to FALSE.

3084 **Output Parameter or each object:**

- 3085 • **<Array> Policies:** A collection of the policy objects.

#### 3086 2.2.1.2.4 Renditions

3087 **Description:** Used to retrieve the renditions of the object(s).

3088 **Optional Input Parameter:**

- 3089 • **String renditionFilter:** The Repository MUST return the set of renditions whose kind matches  
3090 this filter. See section below for the filter grammar.

3091           ○ Defaults to “cmis:none”.

3092**Output Parameter for each object:**

3093     • **<Array> Renditions:** The set of renditions.

#### 3094**2.2.1.2.4.1 Rendition Filter Grammar**

3095The Rendition Filter grammar is defined as follows:

3096       <renditionInclusion> ::= <none> | <wildcard> | <termlist>

3097       <termlist> ::= <term> | <term> ',' <termlist>

3098       <term> ::= <kind> | <mimetype>

3099       <kind> ::= <text>

3100       <mimetype> ::= <type> '/' <subtype>

3101       <type> ::= <text>

3102       <subtype> ::= <text> | <wildcard>

3103       <text> ::= /\* any char except whitespace \*/

3104       <wildcard> ::= '\*'

3105       <none> ::= 'cmis:none'

3106An inclusion pattern allows:

- 3107     • **Wildcard** : include all associated Renditions
- 3108     • **Comma-separated list of Rendition kinds or mimetypes** : include only those Renditions
- 3109       that match one of the specified kinds or mimetypes
- 3110     • **cmis:none**: (Default) exclude all associated Renditions

3111Examples:

- 3112     • \* (include all Renditions)
- 3113     • cmis:thumbnail (include only Thumbnails)
- 3114     • Image/\* (include all image Renditions)
- 3115     • application/pdf, application/x-shockwave-flash (include web ready Renditions)
- 3116     • cmis:none (exclude all Renditions)

#### 3117**2.2.1.2.5 ACLs**

3118**Description:** Used to retrieve the ACLs for the object(s) described in the service response.

3119**Optional Input Parameter:**

- 3120     • **Boolean includeACL:** If TRUE, then the Repository MUST return the ACLs for each object in
- 3121       the result set. Defaults to FALSE.

3122**Output Parameter for each object:**

- 3123     • **<Array> ACLs:** The list of access control entries of the ACL for the object.

#### 3124**2.2.1.2.6 Allowable Actions**

3125**Description:** Used to retrieve the allowable actions for the object(s) described in the service response.

3126**Optional Input Parameter:**

- 3127     • **Boolean includeAllowableActions:** If TRUE, then the Repository MUST return the
- 3128       available actions for each object in the result set. Defaults to FALSE.

3129**Output Parameter for each object:**

3130 • **AllowableActions:** See `cmisAllowableActionsType` in the CMIS schema.

### 31312.2.1.3 Change Tokens

3132The CMIS base object-type definitions include an opaque string “ChangeToken” property that a  
3133Repository MAY use for optimistic locking and/or concurrency checking to ensure that user updates do  
3134not conflict.

3135If a Repository provides values for the `ChangeLogToken` property for an Object, then all invocations of the  
3136“update” methods on that object (`updateProperties`, `setContentStream`, `deleteContentStream`) MUST  
3137provide the value of the `changeLogToken` property as an input parameter, and the Repository MUST  
3138throw an `updateConflictException` if the value specified for the `changeLogToken` does NOT match the  
3139`changeLogToken` value for the object being updated.

### 31402.2.1.4 Exceptions

3141The following sections list the complete set of exceptions that MAY be returned by a repository in  
3142response to a CMIS service method call.

#### 31432.2.1.4.1 General Exceptions

3144The following exceptions MAY be returned by a repository in response to ANY CMIS service method call.

3145The “Cause” field indicates the circumstances under which a repository SHOULD return a particular  
3146exception.

3147 **invalidArgument**

3148 Cause: One or more of the input parameters to the service method is missing or invalid.

3149

3150 **objectNotFound**

3151 Cause: The service call has specified an object that does not exist in the Repository.

3152

3153 **notSupported**

3154 Cause: The service method invoked requires [an optional capability](#) not supported by the  
3155 repository.

3156

3157 **permissionDenied**

3158 Cause: The caller of the service method does not have sufficient permissions to perform the  
3159 operation.

3160

3161 **runtime**

3162 Cause: Any other cause not expressible by another CMIS exception.

#### 31632.2.1.4.2 Specific Exceptions

3164The following exceptions MAY be returned by a repository in response to one or more CMIS service  
3165methods calls.

3166For each exception, the general intent is listed as well as a list of the methods which MAY cause the  
3167exception to be thrown.

3168 **constraint**

3169 Intent: The operation violates a Repository- or Object-level constraint defined in the CMIS  
3170 domain model.

3171 Methods:

3172	•	<b>Navigation Services:</b>	
3173	○	getObjectParents	
3174	•	<b>Object Services:</b>	
3175	○	createDocument	
3176	○	createDocumentFromSource	
3177	○	createFolder	
3178	○	createRelationship	
3179	○	createPolicy	
3180	○	updateProperties	
3181	○	moveObject	
3182	○	deleteObject	
3183	○	setContentStream	
3184	○	deleteContentStream	
3185	•	<b>Multi-filing Services:</b>	
3186	○	addObjectToFolder	
3187	•	<b>Versioning Services:</b>	
3188	○	checkOut	
3189	○	cancelCheckOut	
3190	○	checkIn	
3191	•	<b>Policy Services:</b>	
3192	○	applyPolicy	
3193	○	removePolicy	
3194	•	<b>Change Log Services:</b>	
3195	○	getContentChanges	
3196			
3197		<b>contentAlreadyExists</b>	
3198	Intent:	The operation attempts to set the content stream for a Document that already has a	
3199		content stream without explicitly specifying the “overwriteFlag” parameter.	
3200	Methods:		
3201	•	<b>Object Services:</b>	
3202	○	setContentStream	
3203			
3204		<b>filterNotValid</b>	
3205	Intent:	The property filter or rendition filter input to the operation is not valid.	
3206	Methods:		
3207	•	<b>Navigation Services:</b>	
3208	○	getDescendants	
3209	○	getChildren	
3210	○	getFolderParent	
3211	○	getObjectParents	



3212	○	getCheckedOutDocs
3213	•	<b>Object Services:</b>
3214	○	getProperties
3215	○	getRenditions
3216	○	getObject
3217	○	getObjectByPath
3218	•	<b>Versioning Services:</b>
3219	○	getPropertiesOfLatestVersion
3220	○	getAllVersions
3221	•	<b>Policy Services:</b>
3222	○	getAppliedPolicies
3223		
3224		<b>nameConstraintViolation</b>
3225	Intent:	The repository is not able to store the object that the user is creating/updating due to
3226		a name constraint violation.
3227	Methods:	
3228	•	<b>Object Services:</b>
3229	○	createDocument
3230	○	createDocumentFromSource
3231	○	createFolder
3232	○	updateProperties
3233	○	moveObject
3234		
3235		<b>storage</b>
3236	Intent:	The repository is not able to store the object that the user is creating/updating due to
3237		an internal storage problem.
3238	Methods:	
3239	•	<b>Object Services:</b>
3240	○	createDocument
3241	○	createDocumentFromSource
3242	○	createFolder
3243	○	createRelationship
3244	○	createPolicy
3245	○	updateProperties
3246	○	moveObject
3247	○	setContentStream
3248	○	deleteContentStream
3249	•	<b>Versioning Services:</b>
3250	○	checkOut
3251	○	checkIn

3252

3253	<b>streamNotSupported</b>	
3254	Intent:	The operation is attempting to get or set a contentStream for a Document whose
3255		Object-type specifies that a content stream is not allowed for Document's of that type.
3256	Methods:	
3257	•	<b>Object Services:</b>
3258	○	createDocument
3259	○	createDocumentFromSource
3260	○	getContentStream
3261	○	setContentStream
3262	•	<b>Versioning Services:</b>
3263	○	checkIn
3264		
3265	<b>updateConflict</b>	
3266	Intent:	The operation is attempting to update an object that is no longer current (as
3267		determined by the repository).
3268	Methods:	
3269	•	<b>Object Services:</b>
3270	○	updateProperties
3271	○	moveObject
3272	○	deleteObject
3273	○	deleteTree
3274	○	setContentStream
3275	○	deleteContentStream
3276	•	<b>Versioning Services:</b>
3277	○	checkOut
3278	○	cancelCheckOut
3279	○	checkIn
3280		
3281	<b>versioning</b>	
3282	Intent:	The operation is attempting to perform an action on <a href="#">a non-current version</a> of a
3283		Document that cannot be performed on a non-current version.
3284	Methods:	
3285	•	<b>Object Services:</b>
3286	○	updateProperties
3287	○	moveObject
3288	○	setContentStream
3289	○	deleteContentStream
3290	•	<b>Versioning Services:</b>
3291	○	checkOut
3292	○	cancelCheckOut
3293	○	checkIn

## 3294 2.2.1.5 ACLs

3295 Those services which allow for the setting of ACLs may take the optional macro `cmis:user` which allows  
3296 the caller to indicate the operation applies to the current authenticated user.

## 3297 2.2.2 Repository Services

3298 The Repository Services (`getRepositories`, `getRepositoryInfo`, `getTypeChildren`, `getTypeDescendants`,  
3299 `getTypeDefinition`) are used to discover information about the repository, including information about the  
3300 repository and the object-types defined for the repository.

### 3301 2.2.2.1 getRepositories

3302 **Description:** Returns a list of CMIS repositories available from this CMIS service endpoint.

#### 3303 2.2.2.1.1 Inputs

3304 None.

#### 3305 2.2.2.1.2 Outputs

3306 A list of repository information, with (at least) the following information for each entry:

- 3307 • **ID repositoryId:** The identifier for the Repository.
- 3308 • **String repositoryName:** A display name for the Repository.

### 3309 2.2.2.1.3 Exceptions Thrown & Conditions

3310 See section [2.2.1.4.1 General Exceptions](#)

### 3311 2.2.2.2 getRepositoryInfo

3312 **Description:** Returns information about the CMIS repository, the [optional capabilities](#) it supports and its  
3313 Access Control information if applicable. .

#### 3314 2.2.2.2.1 Inputs

3315 **Required:**

- 3316 • **ID repositoryId:** The identifier for the Repository.

#### 3317 2.2.2.2.2 Outputs

- 3318 • **ID repositoryId:** The identifier for the Repository.
- 3319     o **Note:** This MUST be the same identifier as the input to the method.
- 3320 • **String repositoryName:** A display name for the Repository.
- 3321 • **String repositoryDescription:** A display description for the Repository.
- 3322 • **String vendorName:** A display name for the vendor of the Repository's underlying application.
- 3323 • **String productName:** A display name for the Repository's underlying application.
- 3324 • **String productVersion:** A display name for the version number of the Repository's underlying  
3325 application.
- 3326 • **ID rootFolderId:** The ID of the Root Folder Object for the Repository.
- 3327 • **<List of capabilities>:** The set of values for the repository-optional capabilities specified in  
3328 [section 2.1.1.1 Optional Capabilities](#)

- 3329 • **String** `latestChangeLogToken`: The change log token corresponding to the most recent  
3330 change event for any object in the repository.
- 3331 • **Decimal** `cmisVersionSupported`: A decimal that indicates what version of the CMIS  
3332 specification this repository supports as specified in 2.1.1.2 Implementation Information.
- 3333 • **URI** `thinClientURI`: A optional repository-specific URI pointing to the repository's web  
3334 interface.
- 3335 • **Boolean** `changesIncomplete`: Indicates whether or not the repository's change log can return  
3336 all changes ever made to any object in the repository or only changes made after a particular  
3337 point in time. Applicable when the repository's optional capability `capabilityChanges` is not  
3338 none.
  - 3339 ○ If FALSE, then the change log can return all changes ever made to every object.
  - 3340 ○ If TRUE, then the change log includes all changes made since a particular point in time,  
3341 but not all changes ever made.
- 3342 • **<List of enum values>** `changesOnType`: Indicates whether changes are available for base  
3343 types in the repository. Valid values are from `enumBaseObjectTypesIds`. See section 2.1.11  
3344 [Change Log](#).
  - 3345 ○ **cmis:document**
  - 3346 ○ **cmis:folder**
  - 3347 ○ **cmis:policy**
  - 3348 ○ **cmis:relationship**
- 3349 • **Enum** `supportedPermissions`: specifies which types of permissions are supported.
  - 3350 ○ **basic**: indicates that the CMIS Basic permissions are supported.
  - 3351 ○ **repository**: Indicates that repository specific permissions are supported.
  - 3352 ○ **both**: indicates that both CMIS basic permissions and repository specific permissions are  
3353 supported.
- 3354 • **Enum** `propagation`: The list of allowed values for `applyACL`, which control how non-direct ACEs  
3355 are handled by the repository:
  - 3356 ○ **objectonly**: indicates that the repository is able to apply ACEs without changing the  
3357 ACLs of other objects – i.e. ACEs are applied, potentially “breaking” the “sharing”  
3358 dependency for non-direct ACEs.
  - 3359 ○ **propagate**: indicates that the repository is able to apply ACEs to a given object and  
3360 propagate this change to all inheriting objects – i.e. ACEs are applied with the (intended)  
3361 side effect to inheriting objects.
  - 3362 ○ **repositorydetermined**: indicates that the repository uses its own mechanisms to  
3363 handle non-direct ACEs when applying ACLs.
- 3364 • **<Array> Permission** `permissions`: The list of repository-specific permissions the repository  
3365 supports for managing ACEs (see section 2.8 Access Control).
- 3366 • **<Array> PermissionMapping** `mapping`: The list of mappings for the CMIS Basic permissions to  
3367 allowable actions (see section 2.8 Access Control).
- 3368 • **String** `principalAnonymous`: If set, this field holds the principal who is used for anonymous  
3369 access. This principal can then be passed to the ACL services to specify what permissions  
3370 anonymous users should have.
- 3371 • **String** `principalAnyone`: If set, this field holds the principal who is used to indicate any  
3372 authenticated user. This principal can then be passed to the ACL services to specify what  
3373 permissions any authenticated user should have.

3374The `cmisRepositoryInfoType` schema describes the markup that will be included in all CMIS protocol  
3375bindings to implement this service.

### 33762.2.2.3 Exceptions Thrown & Conditions

3377See section [2.2.1.4.1 General Exceptions](#)

### 33782.2.2.3 getTypeChildren

3379**Description:** Returns the list of [Object-Types](#) defined for the Repository that are children of the specified  
3380Type.

#### 33812.2.2.3.1 Inputs

3382**Required:**

3383 • **String repositoryId:** The identifier for the Repository.

3384**Optional:**

3385 • **String typeId:** The typeId of an Object-Type specified in the Repository.

3386     ○ If specified, then the Repository MUST return all of child types of the specified type.

3387     ○ If not specified, then the Repository MUST return all Base Object-Types.

3388 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the  
3389 property definitions for each Object-Type returned.

3390     ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

3391 • **Integer maxItems:** See section 2.2.1.1 Paging.

3392 • **Integer skipCount:** See section 2.2.1.1 Paging.

#### 33932.2.2.3.2 Outputs

3394 **<Array> Object-Types:** The list of child [Object-Types](#) defined for the given typeId.

3395 **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3396**Optional:**

3397 **Integer numItems:** See section 2.2.1.1 Paging.

### 33982.2.2.3.3 Exceptions Thrown & Conditions

3399See section [2.2.1.4.1 General Exceptions](#)

### 34002.2.2.4 getTypeDescendants

3401**Description:** Returns the set of descendant [Object-Types](#) defined for the Repository under the specified  
3402Type.

3403**Notes:**

3404 • This method does NOT support paging as defined in the 2.2.1.1 *Paging* section.

3405 • The order in which results are returned is repository-specific.

#### 34062.2.2.4.1 Inputs

3407**Required:**

3408 • **String repositoryId:** The identifier for the Repository.

3409**Optional:**

3410 • **String typeId:** The typeId of an Object-Type specified in the Repository.

- 3411           ○ If specified, then the Repository MUST return all descendant types for the specified type.
- 3412           ○ If not specified, then the Repository MUST return all types and MUST ignore the value of
- 3413           the depth parameter
- 3414       • **Integer depth:** The number of levels of depth in the type hierarchy from which to return results.
- 3415       Valid values are:
- 3416           ○ **1:** Return only types that are children of the type.
- 3417           ○ **<Integer value greater than 1>:** Return only types that are children of the type and
- 3418           descendants up to <value> levels deep.
- 3419           ○ **-1:** Return ALL descendant types at all depth levels in the CMIS hierarchy.
- 3420           ○ The default value is repository specific and SHOULD be at least 2 or -1.
- 3421       • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
- 3422       definitions for each Object-Type returned.
- 3423           ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

#### 3424**2.2.4.2 Outputs**

3425    **<Array> Object-Types:** The hierarchy of [Object-Types](#) defined for the Repository.

#### 3426**2.2.4.3 Exceptions Thrown & Conditions**

3427See section [2.2.1.4.1 General Exceptions](#)

- 3428       • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
- 3429       an invalid depth.

#### 3430**2.2.5 getTypeDefinition**

3431**Description:** Gets the definition of the specified Object-Type.Inputs

##### 3432**2.2.5.1 Inputs**

3433**Required:**

- 3434       • **String repositoryId:** The identifier for the Repository.
- 3435       • **String typeId:** The typeId of an Object-Type specified in the Repository.

##### 3436**2.2.5.2 Outputs**

- 3437       • Object-type including all property definitions.

##### 3438**2.2.5.3 Exceptions Thrown & Conditions**

3439See section [2.2.1.4.1 General Exceptions](#)

#### 3440**2.2.3 Navigation Services**

3441The Navigation Services (getDescendants, getChildren, getFolderParent, getObjectParents,  
3442getCheckedoutDocs), are used to traverse the folder hierarchy in a CMIS **Repository**, and to locate  
3443Documents that are checked out.

##### 3444**2.2.3.1 getChildren**

3445**Description:** Gets the list of child objects contained in the specified folder.

3446**Notes:**

- 3447 • If the Repository supports the optional “VersionSpecificFiling” capability, then the repository
- 3448 MUST return the document versions filed in the specified folder.
- 3449 ○ Otherwise, the latest version of the documents MUST be returned.

### 3450 2.2.3.1.1 Inputs

#### 3451 Required:

- 3452 • **ID repositoryId:** The identifier for the Repository.
- 3453 • **ID folderId:** The identifier for the folder.

#### 3454 Optional:

- 3455 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3456 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3457 • **String orderBy:** See ORDER BY Clause in the query
- 3458 • **String filter:** See Error: Reference source not found section 2.2.1.2.1 Properties.
- 3459 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3460 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3461 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3462 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
- 3463 child object for use in constructing that object’s path.

### 3464 2.2.3.1.2 Outputs

- 3465 • **<Array> ObjectResults:** A list of the child objects for the specified folder. Each object
- 3466 result MUST include the following elements if they are requested:
  - 3467 ○ **<Array> Properties:** The list of properties for the object.
  - 3468 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3469 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3470 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
  - 3471 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.
- 3472 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

#### 3473 Optional:

- 3474 **Integer numItems:** See section 2.2.1.1 Paging.

### 3475 2.2.3.1.3 Exceptions Thrown & Conditions

- 3476 • See section [2.2.1.4.1 General Exceptions](#)
- 3477 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
- 3478 parameter is not valid.
- 3479 • **invalidArgument:** if the specified folder is not a folder

### 3480 2.2.3.2 getDescendants

3481 **Description:** Gets the set of [descendant objects](#) contained in the specified folder or any of its child-

3482 folders.

#### 3483 Notes:

- 3484 • This method does NOT support paging as defined in the 2.2.1.1 *Paging* section.

- 3485 • The order in which results are returned is repository-specific..
- 3486 • If the Repository supports the optional capability `capabilityVersionSpecificFiling`, then  
3487 the repository MUST return the document versions filed in the specified folder or its descendant  
3488 folders. Otherwise, the latest version of the documents MUST be returned.
- 3489 • If the Repository supports the optional capability `capabilityMutlifiling` and the same  
3490 document is encountered multiple times in the hierarchy, then the repository MUST return that  
3491 document each time is encountered.

### 34922.2.3.2.1 Inputs

#### 3493Required:

- 3494 • **ID repositoryId:** The identifier for the Repository.
- 3495 • **ID folderId:** The identifier for the folder.

#### 3496Optional:

- 3497 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return  
3498 results. Valid values are:
  - 3499 ○ **1:** Return only objects that are children of the folder.
  - 3500 ○ **<Integer value greater than 1>:** Return only objects that are children of the folder and  
3501 descendants up to <value> levels deep.
  - 3502 ○ **-1:** Return ALL descendant objects at all depth levels in the CMIS hierarchy.
  - 3503 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 3504 • **String filter:** See section 2.2.1.2.1 Properties.
- 3505 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3506 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3507 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3508 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each  
3509 child object for use in constructing that object's path.

### 35102.2.3.2.2 Outputs

- 3511 • **<Array> ObjectResults:** A list of the descendant objects for the specified folder. Each  
3512 object result MUST include the following elements if they are requested:
  - 3513 ○ **<Array> Properties:** The list of properties for the object.
  - 3514 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3515 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3516 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
  - 3517 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

### 35182.2.3.2.3 Exceptions Thrown & Conditions

3519 See section 2.2.1.4.1 General Exceptions

- 3520 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
3521 parameter is not valid.
- 3522 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with  
3523 "depth = 0".
- 3524 • **invalidArgument:** if the specified folder is not a folder



### 35252.2.3.3 getFolderTree

3526**Description:** Gets the set of descendant folder objects contained in the specified folder.

3527

3528**Notes:**

- 3529 • This method does NOT support paging as defined in the 2.2.1.1 *Paging* section.
- 3530 • The order in which results are returned is repository-specific..

### 35312.2.3.3.1 Inputs

3532**Required:**

- 3533 • **ID repositoryId:** The identifier for the Repository.
- 3534 • **ID folderId:** The identifier for the folder.

3535**Optional:**

- 3536 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return  
3537 results. Valid values are:
  - 3538 ○ **1:** Return only folders that are children of the folder.
  - 3539 ○ **<Integer value greater than 1>:** Return only folders that are children of the folder and  
3540 descendant folders up to <value> levels deep.
  - 3541 ○ **-1:** Return ALL descendant folders at all depth levels in the CMIS hierarchy.
  - 3542 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 3543 • **String filter:** See section 2.2.1.2.1 Properties.
- 3544 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3545 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3546 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3547 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each  
3548 child object for use in constructing that object's path.

### 35492.2.3.3.2 Outputs

- 3550 • **<Array> ObjectResults:** A list of the descendant folders for the specified folder. Each  
3551 object result MUST include the following elements if they are requested:
  - 3552 ○ **<Array> Properties:** The list of properties for the object.
  - 3553 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3554 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3555 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
  - 3556 ○ **String pathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

### 35572.2.3.3.3 Exceptions Thrown & Conditions

- 3558 • See section 2.2.1.4.1 General Exceptions
- 3559 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
3560 parameter is not valid.
- 3561 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with  
3562 an invalid depth
- 3563 • **invalidArgument:** if the specified folder is not a folder

3564

3565

#### 3566 **2.2.3.4 getFolderParent**

3567 **Description:** Gets the parent folder object for the specified folder object.

##### 3568 **2.2.3.4.1 Inputs**

3569 **Required:**

3570 • **ID repositoryId:** The identifier for the Repository.

3571 • **ID folderId:** The identifier for the folder.

3572 **Optional:**

3573 • **String filter:** See section 2.2.1.2.1 Properties.Error: Reference source not found

##### 3574 **2.2.3.4.2 Outputs**

3575 • **Object:** The parent folder object of the specified folder.

##### 3576 **2.2.3.4.3 Exceptions Thrown & Conditions**

3577 • See section [2.2.1.4.1 General Exceptions](#)

3578 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
3579 parameter is not valid.

3580 • **invalidArgument:** The Repository MUST throw this exception if the folderId input is  
3581 the root folder.

#### 3582 **2.2.3.5 getObjectParents**

3583 **Description:** Gets the parent folder(s) for the specified non-folder, fileable object.

##### 3584 **2.2.3.5.1 Inputs**

3585 **Required:**

3586 • **ID repositoryId:** The identifier for the Repository.

3587 • **ID objectId:** The identifier for the object.

3588 **Optional:**

3589 • **String filter:** See section 2.2.1.2.1 **Properties**

3590 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.

3591 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.

3592 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3593 • **Boolean includeRelativePathSegment:** See section 2.1.5.3 Paths.

##### 3594 **2.2.3.5.2 Outputs**

3595 • **<Array> ObjectResults:** A list of the parent folder(s) of the specified objects. Empty for  
3596 unfiled objects or for the root folder. Each object result MUST include the following elements if  
3597 they are requested:

3598 ○ **<Array> Properties:** The list of properties for the object.

3599 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.

- 3600      ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3601      ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3602      ○ **String relativePathSegment:** If includeRelativePathSegment was TRUE. See section
- 3603      2.1.5.3 Paths.

### 3604 **2.2.3.5.3 Exceptions Thrown & Conditions**

- 3605      • See section [2.2.1.4.1 General Exceptions](#)
- 3606      • **constraint:** The Repository MUST throw this exception if this method is invoked on an
- 3607      object who Object-Type Definition specifies that it is not fileable.
- 3608      • **filterNotValid:** The Repository MUST throw this exception if this property filter input
- 3609      parameter is not valid.
- 3610      • **invalidArgument:** if the specified folder is not a folder

### 3611 **2.2.3.6 getCheckedOutDocs**

3612 **Description:** Gets the list of documents that are checked out that the user has access to.

#### 3613 **2.2.3.6.1 Inputs**

3614 **Required:**

- 3615      • **ID repositoryId:** The identifier for the Repository.

3616 **Optional:**

- 3617      • **ID folderId:** The identifier for a folder in the repository from which documents should be
- 3618      returned.
- 3619      ○ If specified, the Repository MUST only return checked out documents that are child-
- 3620      objects of the specified folder.
- 3621      ○ If not specified, the Repository MUST return checked out documents from anywhere in
- 3622      the repository hierarchy.
- 3623      • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3624      • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3625      • **String orderBy:** See ORDER BY Clause in the query
- 3626      • **String filter:** See section 2.2.1.2.1 Properties.
- 3627      • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3628      • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3629      • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

#### 3630 **2.2.3.6.2 Outputs**

- 3631      • **<Array> ObjectResults:** A list of checked out documents. Each object result MUST
- 3632      include the following elements if they are requested:
- 3633      ○ **<Array> Properties:** The list of properties for the object.
- 3634      ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3635      ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3636      ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3637      • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3638 **Optional:**

3639 **Integer numItems:** See section 2.2.1.1 Paging.

3640

### 3641 2.2.3.6.3 Exceptions Thrown & Conditions

3642 • See section [2.2.1.4.1 General Exceptions](#)

3643 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
3644 parameter is not valid.

## 3645 2.2.4 Object Services

3646 CMIS provides ID-based CRUD (**Create, Retrieve, Update, Delete**), operations on objects in a Repository.

### 3647 2.2.4.1 createDocument

3648 **Description:** Creates a document object of the specified type (given by the `cmis:objectTypeId` property)  
3649 in the (optionally) specified location.

#### 3650 2.2.4.1.1 Inputs

3651 **Required:**

3652 • **ID repositoryId:** The identifier for the Repository.

3653 • **<Array> properties:** The property values that MUST be applied to the newly-created Document  
3654 Object.

3655 **Optional:**

3656 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the  
3657 newly-created Document Object.

3658 ○ This parameter MUST be specified if the Repository does NOT support the optional  
3659 “unfiling” capability.

3660 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the  
3661 newly-created Document Object. The method of passing the contentStream to the server and the  
3662 encoding mechanism will be specified by each specific binding. MUST be required if the type  
3663 requires it.

3664 • **Enum versioningState:** An enumeration specifying what the versioning state of the  
3665 newly-created object MUST be. Valid values are:

3666 ○ **none:** The document MUST be created as a non-versionable document.

3667 ○ **checkedout:** The document MUST be created in the checked-out state.

3668 ○ **major (default):** The document MUST be created as a major version

3669 ○ **minor:** The document MUST be created as a minor version.

3670 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document  
3671 object.

3672 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document  
3673 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.

3674 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3675 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is  
3676 specified.

#### 3677 2.2.4.1.2 Outputs

3678 **ID objectId:** The ID of the newly-created document.

### 36792.2.4.1.3 Exceptions Thrown & Conditions

- 3680 • See section [2.2.1.4.1 General Exceptions](#)
- 3681 • **constraint:** The Repository MUST throw this exception if ANY of the following  
3682 conditions are met:
  - 3683 ○ The cmis:objectTypeId property value is not an Object-Type whose baseType is  
3684 "Document".
  - 3685 ○ The cmis:objectTypeId property value is NOT in the list of AllowedChildObjectTypes of  
3686 the parent-folder specified by folderId.
  - 3687 ○ The value of any of the properties violates the min/max/required/length constraints  
3688 specified in the property definition in the Object-Type.
  - 3689 ○ The "contentStreamAllowed" attribute of the Object-Type definition specified by the  
3690 cmis:objectTypeId property value is set to "required" and no contentStream input  
3691 parameter is provided.
  - 3692 ○ The "versionable" attribute of the Object-Type definition specified by the  
3693 cmis:objectTypeId property value is set to FALSE and a value for the versioningState  
3694 input parameter is provided that is something other than "none".
  - 3695 ○ The "versionable" attribute of the Object-Type definition specified by the  
3696 cmis:objectTypeId property value is set to TRUE and the value for the versioningState  
3697 input parameter is provided that is "none".
  - 3698 ○ The "controllablePolicy" attribute of the Object-Type definition specified by the  
3699 cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
  - 3700 ○ The "controllableACL" attribute of the Object-Type definition specified by the  
3701 cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
  - 3702 ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3703 repository.
- 3704 • **nameConstraintViolation:** See section [2.2.1.4.2 Specific Exceptions](#). If the  
3705 repository detects a violation with the given cmis:name property value, the repository MAY throw  
3706 this exception or chose a name which does not conflict.
- 3707 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).
- 3708 • **streamNotSupported:** The Repository MUST throw this exception if the  
3709 "contentStreamAllowed" attribute of the Object-Type definition specified by the cmis:objectTypeId  
3710 property value is set to "not allowed" and a contentStream input parameter is provided.

### 37112.2.4.2 createDocumentFromSource

3712**Description:** Creates a document object as a copy of the given source document in the (optionally)  
3713specified location.

#### 37142.2.4.2.1 Inputs

3715**Required:**

- 3716 • **ID repositoryId:** The identifier for the Repository.
- 3717 • **ID sourceId:** The identifier for the source document.

3718**Optional:**

- 3719 • **<Array> properties:** The property values that MUST be applied to the Object. This list  
3720 of properties SHOULD only contain properties whose values differ from the source document.
- 3721 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the  
3722 newly-created Document Object.

- 3723           ○ This parameter MUST be specified if the Repository does NOT support the optional  
3724           “unfiling” capability.
- 3725       •     **Enum versioningState:** An enumeration specifying what the versioning state of the  
3726       newly-created object MUST be. Valid values are:
  - 3727           ○ none: The document MUST be created as a non-versionable document.
  - 3728           ○ checkedout: The document MUST be created in the checked-out state.
  - 3729           ○ major **(default)**: The document MUST be created as a major version
  - 3730           ○ minor: The document MUST be created as a minor version.
- 3731       •     **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document  
3732       object.
- 3733       •     **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document  
3734       object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3735       •     **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3736       Document object, either using the ACL from folderId if specified, or being ignored if no folderId is  
3737       specified.

## 3738 2.2.4.2.2 Outputs

3739     **ID objectId:** The ID of the newly-created document.

## 3740 2.2.4.2.3 Exceptions Thrown & Conditions

- 3741       •     See section [2.2.1.4.1 General Exceptions](#)
- 3742       •     **constraint:** The Repository MUST throw this exception if ANY of the following  
3743       conditions are met:
  - 3744           ○ The sourceId is not an Object whose baseType is “Document”.
  - 3745           ○ The source document’s cmis:objectTypeId property value is NOT in the list of  
3746           AllowedChildObjectTypelds of the parent-folder specified by folderId.
  - 3747           ○ The “versionable” attribute of the Object-Type definition specified by the  
3748           cmis:objectTypeId property value is set to FALSE and a value for the versioningState  
3749           input parameter is provided that is something other than “none”.
  - 3750           ○ The “versionable” attribute of the Object-Type definition specified by the  
3751           cmis:objectTypeId property value is set to TRUE and the value for the versioningState  
3752           input parameter is provided that is “none”.
  - 3753           ○ The “controllablePolicy” attribute of the Object-Type definition specified by the  
3754           cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
  - 3755           ○ The “controllableACL” attribute of the Object-Type definition specified by the  
3756           cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
  - 3757           ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3758           repository.
- 3759       •     **nameConstraintViolation:** See section [2.2.1.4.2 Specific Exceptions](#). If the  
3760       repository detects a violation with the given cmis:name property value, the repository MAY throw  
3761       this exception or chose a name which does not conflict.
- 3762       •     **storage:** See section [2.2.1.4.2 Specific Exceptions](#).
- 3763       •     **streamNotSupported:** The Repository MUST throw this exception if the  
3764       “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId  
3765       property value is set to “not allowed” and a contentStream input parameter is provided.

### 3766**2.2.4.3 createFolder**

3767**Description:** Creates a folder object of the specified type in the specified location.

#### 3768**2.2.4.3.1 Inputs**

3769**Required:**

- 3770 • **ID repositoryId:** The identifier for the Repository.
- 3771 • **<Array> properties:** The property values that MUST be applied to the newly-created Folder  
3772 Object.
- 3773 • **ID folderId:** The identifier for the folder that MUST be the parent folder for the newly-  
3774 created Folder Object.

3775**Optional:**

- 3776 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Folder object.
- 3777 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Folder object,  
3778 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3779 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3780 Folder object, either using the ACL from folderId if specified, or being ignored if no folderId is  
3781 specified.

#### 3782**2.2.4.3.2 Outputs**

- 3783 • **ID objectId:** The ID of the newly-created folder.

#### 3784**2.2.4.3.3 Exceptions Thrown & Conditions**

- 3785 • See section [2.2.1.4.1 General Exceptions](#)
- 3786 • **constraint:** The Repository MUST throw this exception if ANY of the following  
3787 conditions are met:
  - 3788 ○ The cmis:objectId property value is not an Object-Type whose baseType is “Folder”.
  - 3789 ○ The value of any of the properties violates the min/max/required/length constraints  
3790 specified in the property definition in the Object-Type.
  - 3791 ○ The cmis:objectId property value is NOT in the list of AllowedChildObjectIds of  
3792 the parent-folder specified by folderId.
  - 3793 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the  
3794 cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3795 ○ The “controllableACL” attribute of the Object-Type definition specified by the  
3796 cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3797 ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3798 repository.
- 3799 • **nameConstraintViolation:** See section [2.2.1.4.2 Specific Exceptions](#). If the  
3800 repository detects a violation with the given cmis:name property value, the repository MAY throw  
3801 this exception or chose a name which does not conflict.
- 3802 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).

#### 3803**2.2.4.4 createRelationship**

3804**Description:** Creates a relationship object of the specified type



#### 38052.2.4.4.1 Inputs

##### 3806Required:

- 3807 • **ID repositoryId:** The identifier for the Repository.
- 3808 • **<Array> properties:** The property values that MUST be applied to the newly-created
- 3809 Relationship Object.

##### 3810Optional:

- 3811 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Relationship
- 3812 object.
- 3813 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Relationship
- 3814 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3815 **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
- 3816 Relationship object, either using the ACL from folderId if specified, or being ignored if no folderId
- 3817 is specified.

#### 38182.2.4.4.2 Outputs

- 3819 • **ID objectId:** The ID of the newly-created relationship.

#### 38202.2.4.4.3 Exceptions Thrown & Conditions

- 3821 • See section 2.2.1.4.1 General Exceptions
- 3822 • **constraint:** The Repository MUST throw this exception if ANY of the following
- 3823 conditions are met:
  - 3824 ○ The cmis:objectId property value is not an Object-Type whose baseType is
  - 3825 "Relationship".
  - 3826 ○ The value of any of the properties violates the min/max/required/length constraints
  - 3827 specified in the property definition in the Object-Type.
  - 3828 ○ The sourceObjectId's ObjectType is not in the list of "allowedSourceTypes" specified by
  - 3829 the Object-Type definition specified by cmis:objectId property value.
  - 3830 ○ The targetObjectId's ObjectType is not in the list of "allowedTargetTypes" specified by the
  - 3831 Object-Type definition specified by cmis:objectId property value.
  - 3832 ○ The "controllablePolicy" attribute of the Object-Type definition specified by the
  - 3833 cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3834 ○ The "controllableACL" attribute of the Object-Type definition specified by the
  - 3835 cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3836 ○ At least one of the permissions is used in an ACE provided which is not supported by the
  - 3837 repository.
- 3838 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).

#### 38392.2.4.5 createPolicy

3840**Description:** Creates a policy object of the specified type

#### 38412.2.4.5.1 Inputs

##### 3842Required:

- 3843 • **ID repositoryId:** The identifier for the Repository.
- 3844 • **<Array> properties:** The property values that MUST be applied to the newly-created Policy
- 3845 Object.



3846**Optional:**

- 3847 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-  
3848 created Policy Object.
- 3849 ○ This parameter MUST be specified if the Repository does NOT support the optional “unfiling”  
3850 capability.
- 3851 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Policy object.
- 3852 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Policy object,  
3853 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3854 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created Policy  
3855 object, either using the ACL from folderId if specified, or being ignored if no folderId is specified.

## 3856**2.2.4.5.2 Outputs**

- 3857 • **ID objectId:** The ID of the newly-created Policy Object.

## 3858**2.2.4.5.3 Exceptions Thrown & Conditions**

- 3859 • See section [2.2.1.4.1 General Exceptions](#)
- 3860 • **constraint:** The Repository MUST throw this exception if ANY of the following  
3861 conditions are met:
  - 3862 ○ The cmis:objectId property value is not an Object-Type whose baseType is “Policy”.
  - 3863 ○ The value of any of the properties violates the min/max/required/length constraints  
3864 specified in the property definition in the Object-Type.
  - 3865 ○ The cmis:objectId property value is NOT in the list of AllowedChildObjectIds of  
3866 the parent-folder specified by folderId.
  - 3867 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the  
3868 cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3869 ○ The “controllableACL” attribute of the Object-Type definition specified by the  
3870 cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3871 ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3872 repository.
- 3873 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).

## 3874**2.2.4.6 getAllowableActions**

3875**Description:** Gets the list of allowable actions for an Object (see section.2.2.1.2.6 Allowable Actions).

### 3876**2.2.4.6.1 Inputs**

3877**Required:**

- 3878 • **ID repositoryId:** The identifier for the Repository.
- 3879 • **ID objectId:** The identifier for the object

### 3880**2.2.4.6.2 Outputs**

- 3881 • **<Array> AllowableActions:** see section 2.2.1.2.6 Allowable Actions.

## 3882**2.2.4.6.3 Exceptions Thrown & Conditions**

3883 See section [2.2.1.4.1 General Exceptions](#)

#### 3884 **2.2.4.7 getObject**

3885 **Description:** Gets the specified information for the Object.

##### 3886 **2.2.4.7.1 Inputs**

3887 **Required:**

- 3888 • **ID repositoryId:** The identifier for the Repository.
- 3889 • **ID objectId:** The identifier for the object

3890 **Optional:**

- 3891 • **String filter:** See Error: Reference source not found section 2.2.1.2.1 Properties.
- 3892 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3893 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 3894 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3895 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 3896 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

##### 3897 **2.2.4.7.2 Outputs**

- 3898 **<Array> Properties:** The list of properties for the object.
- 3899 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3900 **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 3901 **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3902 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 3903 **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

##### 3904 **2.2.4.7.3 Exceptions Thrown & Conditions**

- 3905 See section [2.2.1.4.1 General Exceptions](#)
- 3906 **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter
- 3907 is not valid.

#### 3908 **2.2.4.8 getProperties**

3909 **Description:** Gets the list of properties for an Object.

##### 3910 **2.2.4.8.1 Inputs**

3911 **Required:**

- 3912 • **ID repositoryId:** The identifier for the Repository.
- 3913 • **ID objectId:** The identifier for the object

3914 **Optional:**

- 3915 • **String filter:** See Error: Reference source not found section 2.2.1.2.1 Properties.

##### 3916 **2.2.4.8.2 Outputs**

- 3917 **<Array> Properties:** The list of properties for the object.

### 39182.2.4.8.3 Exceptions Thrown & Conditions

3919 See section [2.2.1.4.1 General Exceptions](#)

3920 `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter  
3921 is not valid.

### 39222.2.4.9 getObjectByPath

3923**Description**: Gets the specified object.

#### 39242.2.4.9.1 Inputs

3925**Required**:

- 3926 • **ID repositoryId**: The identifier for the Repository.
- 3927 • **String path**: The path to the object. See section 2.1.5.3 Paths.

3928**Optional**:

- 3929 • **String filter**: See section 2.2.1.2.1 Properties.
- 3930 • **Boolean includeAllowableActions**: See section 2.2.1.2.6 Allowable Actions.

#### 39312.2.4.9.2 Outputs

- 3932 **<Array> Properties**: The list of properties for the object.
- 3933 **AllowableActions**: See section 2.2.1.2.6 Allowable Actions.

### 39342.2.4.9.3 Exceptions Thrown & Conditions

3935 • See section [2.2.1.4.1 General Exceptions](#)

3936 `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter  
3937 is not valid.

### 39382.2.4.10 getContentStream

3939**Description**: Gets the content stream for the specified Document object, or gets a rendition stream for a  
3940specified rendition of a document or folder object.

3941**Notes**: Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,  
3942in a manner appropriate to that protocol.

#### 39432.2.4.10.1 Inputs

3944**Required**:

- 3945 • **ID repositoryId**: The identifier for the Repository.
- 3946 • **ID objectId**: The identifier for the object

3947**Optional**:

- 3948 • **ID streamId**: The identifier for the rendition stream, when used to get a rendition stream. For  
3949 Documents, if not provided then this method returns the content stream. For Folders, it MUST be  
3950 provided.

#### 39512.2.4.10.2 Outputs

- 3952 • **<Stream> ContentStream**: The specified content stream or rendition stream for the object.

### 3953 2.2.4.10.3 Exceptions Thrown & Conditions

- 3954 • See section [2.2.1.4.1 General Exceptions](#)

3955 **constraint:** The Repository MUST throw this exception if the object specified by objectId does  
3956 NOT have a content stream or rendition stream.

### 3957 2.2.4.11 getRenditions

3958 **Description:** Gets the list of associated Renditions for the specified object. Only rendition attributes are  
3959 returned, not rendition stream.

3960 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,  
3961 in a manner appropriate to that protocol.

### 3962 2.2.4.11.1 Inputs

3963 **Required:**

- 3964 • **ID repositoryId:** The identifier for the Repository.
- 3965 • **ID objectId:** The identifier for the object

3966 **Optional:**

- 3967 • **String renditionFilter:** See Section 2.2.1.2.4
- 3968 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3969 • **Integer skipCount:** See section 2.2.1.1 Paging.

### 3970 2.2.4.11.2 Outputs

- 3971 • **<Array> Renditions:** The set of renditions available on this object

### 3972 2.2.4.11.3 Exceptions Thrown & Conditions

- 3973 • See section [2.2.1.4.1 General Exceptions](#)
- 3974 • **notSupported:** The service method requires functionality that is not supported by the  
3975 repository
- 3976 • **filterNotValid:** The filter specified is not valid

### 3977 2.2.4.12 updateProperties

3978 **Description:** Updates properties of the specified object.

3979 **Notes:**

- 3980 • A Repository MAY automatically create new Document versions as part of an update properties  
3981 operation. Therefore, the objectId output NEED NOT be identical to the objectId input.
- 3982 • Each CMIS protocol bindings MUST specify whether the updateProperties service MUST always  
3983 include all updatable properties, or only those properties whose values are different than the  
3984 original value of the object.

### 3985 2.2.4.12.1 Inputs

3986 **Required:**

- 3987 • **ID repositoryId:** The identifier for the Repository.
- 3988 **ID objectId:** The identifier of the object to be updated.
- 3989 • **<Array> properties:** The updated property values that MUST be applied to the Object.

3990 **Optional:**

3991 • **String changeToken:** See section 2.2.1.3 Change Tokens.

## 3992 **2.2.4.12.2 Outputs**

3993 • **ID objectId:** The ID of the updated object.

3994 • **String changeToken:** See section 2.2.1.3 Change Tokens.

## 3995 **2.2.4.12.3 Exceptions Thrown & Conditions**

3996 • See section [2.2.1.4.1 General Exceptions](#)

3997 • **constraint:** The Repository MUST throw this exception if the value of any of the  
3998 properties violates the min/max/required/length constraints specified in the property definition in  
3999 the Object-Type.

4000 • **nameConstraintViolation:** See section [2.2.1.4.2 Specific Exceptions](#). The  
4001 repository MAY throw this exception or chose a name which does not conflict.

4002 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).

4003 • **updateConflict:** See section [2.2.1.4.2 Specific Exceptions](#).

4004 • **versioning:** The Repository MUST throw this exception if ANY of the following  
4005 conditions are met:

4006 ○ The object is not checked out and ANY of the properties being updated are defined in  
4007 their Object-Type definition have an attribute value of *Updatability* when checked-out.

4008 ○ Additionally, the repository MAY throw this exception if the object is a non-current  
4009 Document Version.

## 4010 **2.2.4.13 moveObject**

4011 **Description:** Moves the specified file-able object from one folder to another.

## 4012 **2.2.4.13.1 Inputs**

4013 **Required:**

4014 • **ID repositoryId:** The identifier for the Repository.

4015 • **ID objectId:** The identifier of the object to be moved.

4016 • **ID targetFolderId:** The folder into which the object is to be moved.

4017 • **ID sourceFolderId:** The folder from which the object is to be moved.

## 4018 **2.2.4.13.2 Outputs**

4019 • **ID objectId:** The identifier of the object to be moved.

## 4020 **2.2.4.13.3 Exceptions Thrown & Conditions**

4021 • See section [2.2.1.4.1 General Exceptions](#)

4022 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with a  
4023 missing sourceFolderId or the sourceFolderId doesn't match the specified object's parent folder  
4024 (or one of the parent folders if the repository supports multifiling.).

4025 • **constraint:** The Repository MUST throw this exception if the cmis:objectTypeId  
4026 property value of the given object is NOT in the list of AllowedChildObjectTypelds of the parent-  
4027 folder specified by targetFolderId.

- 4028       •       nameConstraintViolation: See section [2.2.1.4.2 Specific Exceptions](#). The  
4029 repository MAY throw this exception or chose a name which does not conflict.
- 4030       •       storage: See section [2.2.1.4.2 Specific Exceptions](#).
- 4031       •       updateConflict: See section [2.2.1.4.2 Specific Exceptions](#).
- 4032       •       versioning: The repository MAY throw this exception if the object is a non-current  
4033 Document Version.

#### 4034 **2.2.4.14 deleteObject**

4035 **Description:** Deletes the specified object.

##### 4036 **2.2.4.14.1 Inputs**

4037 **Required:**

- 4038       •       **ID repositoryId:** The identifier for the Repository.
- 4039       •       **ID objectId:** The identifier of the object to be deleted.

4040 **Optional:**

- 4041       •       **Boolean allVersions:** If TRUE (default), then delete all versions of the document. If FALSE,  
4042 delete only the document object specified. The Repository MUST ignore the value of this  
4043 parameter when this service is invoke on a non-document object or non-versionable document  
4044 object.

4045

##### 4046 **2.2.4.14.2 Exceptions Thrown & Conditions**

- 4047       •       See section [2.2.1.4.1 General Exceptions](#)
- 4048       •       constraint: The Repository MUST throw this exception if the method is invoked on a  
4049 Folder object that contains one or more objects.
- 4050       •       updateConflict: See section [2.2.1.4.2 Specific Exceptions](#).

#### 4051 **2.2.4.15 deleteTree**

4052 **Description:** Deletes the specified folder object and all of its child- and descendant-objects.

4053 **Notes:**

- 4054       •       A Repository MAY attempt to delete child- and descendant-objects of the specified folder in any  
4055 order.
- 4056       •       Any child- or descendant-object that the Repository cannot delete MUST persist in a valid state in  
4057 the CMIS domain model.
- 4058       •       This is not atomic.
- 4059       •       However, if deletesinglefiled is chosen and some objects fail to delete, then single-filed objects  
4060 are either deleted or kept, never just unfiled. This is so that a user can call this command again to  
4061 recover from the error by using the same tree.

##### 4062 **2.2.4.15.1 Inputs**

4063 **Required:**

- 4064       •       **ID repositoryId:** The identifier for the Repository.
- 4065       •       **ID folderId:** The identifier of the folder to be deleted.

4066 **Optional:**

- 4067 • **Boolean allVersions:** If TRUE (default), then delete all versions of the document. If FALSE,  
4068 delete only the document object specified. The Repository MUST ignore the value of this  
4069 parameter when this service is invoke on a non-document object or non-versionable document  
4070 object.
- 4071 • **Enum unfileObjects:** An enumeration specifying how the repository MUST process file-able  
4072 child- or descendant-objects. Valid values are:
  - 4073 ○ **unfile:** Unfile all fileable objects.
  - 4074 ○ **deletesinglefiled:** Delete all fileable non-folder objects whose only parent-folders  
4075 are in the current folder tree. Unfile all other fileable non-folder objects from the current folder  
4076 tree.
  - 4077 ○ **delete (default):** Delete all fileable objects.
- 4078 • **boolean continueOnFailure:** If TRUE, then the repository SHOULD continue attempting to  
4079 perform this operation even if deletion of a child- or descendant-object in the specified folder  
4080 cannot be deleted.
  - 4081 ○ If FALSE (**default**), then the repository SHOULD abort this method when it fails to delete  
4082 a single child- or descendant-object.

## 4083 2.2.4.15.2 Outputs

- 4084 • **<Array> ID failedToDelete:** A list of identifiers of objects in the folder tree that were not deleted.

## 4085 2.2.4.15.3 Exceptions Thrown & Conditions

- 4086 • See section [2.2.1.4.1 General Exceptions](#)
- 4087 • **updateConflict:** See section [2.2.1.4.2 Specific Exceptions](#).

## 4088 2.2.4.16 setContentStream

4089 **Description:** Sets the content stream for the specified Document object.

4090 **Notes:** A Repository MAY automatically create new Document versions as part of this service method.  
4091 Therefore, the objectId output NEED NOT be identical to the objectId input.

## 4092 2.2.4.16.1 Inputs

4093 **Required:**

- 4094 • **ID repositoryId:** The identifier for the Repository.
- 4095 • **ID objectId:** The identifier for the Document object.
- 4096 • **<contentStream> contentStream:** The Content Stream

4097 **Optional:**

- 4098 • **Boolean overwriteFlag:** If TRUE (**default**), then the Repository MUST replace the  
4099 existing content stream for the object (if any) with the input contentStream.
  - 4100 ○ If FALSE, then the Repository MUST only set the input contentStream for the object if the  
4101 object currently does not have a content-stream.
- 4102 • **String changeToken:** See section 2.2.1.3 Change Tokens.

## 4103 2.2.4.16.2 Outputs

- 4104 • **ID objectId:** The ID of the document.
- 4105 • **String changeToken:** See section 2.2.1.3 Change Tokens.



#### 4106**2.2.4.16.3 Exceptions Thrown & Conditions**

- 4107
  - See section [2.2.1.4.1 General Exceptions](#)
- 4108
  - `contentAlreadyExists`: The Repository MUST throw this exception if the input
  - 4109 parameter `overwriteFlag` is FALSE and the Object already has a content-stream.
- 4110
  - `storage`: See section [2.2.1.4.2 Specific Exceptions](#).
- 4111
  - `streamNotSupported`: The Repository MUST throw this exception if the
  - 4112 “`contentStreamAllowed`” attribute of the Object-Type definition specified by the `cmis:objectType` id
  - 4113 property value of the given document is set to “notallowed”.
- 4114
  - `updateConflict`: See section [2.2.1.4.2 Specific Exceptions](#).
- 4115
  - `versioning`: The repository MAY throw this exception if the object is a non-current
  - 4116 Document Version.

#### 4117**2.2.4.17 deleteContentStream**

4118**Description:** Deletes the content stream for the specified Document object.

4119**Notes:** A Repository MAY automatically create new Document versions as part of this service method.

4120Therefore, the `objectId` output NEED NOT be identical to the `objectId` input.

#### 4121**2.2.4.17.1 Inputs**

4122**Required:**

- 4123
  - **ID repositoryId**: The identifier for the Repository.
- 4124
  - **ID objectId**: The identifier for the Document object.

4125**Optional:**

- 4126
  - **String changeToken**: See section 2.2.1.3 Change Tokens.

#### 4127**2.2.4.17.2 Outputs**

- 4128
  - **ID objectId**: The ID of the Document object.
- 4129
  - **String changeToken**: See section 2.2.1.3 Change Tokens.

#### 4130**2.2.4.17.3 Exceptions Thrown & Conditions**

- 4131
  - See section [2.2.1.4.1 General Exceptions](#)
- 4132
  - `constraint`: The Repository MUST throw this exception if the Object’s Object-Type
  - 4133 definition “`contentStreamAllowed`” attribute is set to “*required*”.
- 4134
  - `storage`: See section [2.2.1.4.2 Specific Exceptions](#).
- 4135
  - `updateConflict`: See section [2.2.1.4.2 Specific Exceptions](#).
- 4136
  - `versioning`: The repository MAY throw this exception if the object is a non-current
  - 4137 Document Version.

#### 4138**2.2.5 Multi-filing Services**

4139The Multi-filing services (*addObjectToFolder*, *removeObjectFromFolder*) are supported only if the

4140repository supports the multifiling or unfiling [optional capabilities](#). The Multi-filing Services are used to

4141file/un-file objects into/from folders.

4142This service is NOT used to create or delete objects in the repository.



### 4143 **2.2.5.1 addObjectToFolder**

4144 **Description:** Adds an existing fileable non-folder object to a folder.

#### 4145 **2.2.5.1.1 Inputs**

4146 **Required:**

- 4147 • **ID repositoryId:** The identifier for the Repository.
- 4148 • **ID objectId:** The identifier of the object.
- 4149 • **ID folderId:** The folder into which the object is to be filed.

4150 **Optional:**

- 4151 • **Boolean allVersions:** Add all versions of the object to the folder if the repository supports  
4152 version-specific filing. Defaults to TRUE.

#### 4153 **2.2.5.1.2 Exceptions Thrown & Conditions**

- 4154 • See section [2.2.1.4.1 General Exceptions](#).
- 4155 • **constraint:** The Repository MUST throw this exception if the cmis:objectTypeId  
4156 property value of the given object is NOT in the list of AllowedChildObjectTypelds of the parent-  
4157 folder specified by folderId.

### 4158 **2.2.5.2 removeObjectFromFolder**

4159 **Description:** Removes an existing fileable non-folder object from a folder.

#### 4160 **2.2.5.2.1 Inputs**

4161 **Required:**

- 4162 • **ID repositoryId:** The identifier for the Repository.
- 4163 • **ID objectId:** The identifier of the object.

4164 **Optional:**

- 4165 • **ID folderId:** The folder from which the object is to be removed.
  - 4166 ○ If no value is specified, then the Repository MUST remove the object from all folders in which  
4167 it is currently filed.

#### 4168 **2.2.5.2.2 Exceptions Thrown & Conditions**

- 4169 • See section [2.2.1.4.1 General Exceptions](#)

### 4170 **2.2.6 Discovery Services**

4171 The Discovery Services (*query*) are used to search for query-able objects within the Repository.

#### 4172 **2.2.6.1 query**

4173 **Description:** Executes a CMIS query statement against the contents of the Repository.

#### 4174 **2.2.6.1.1 Inputs**

4175 **Required:**

- 4176 • **ID repositoryId:** The identifier for the Repository.
- 4177 • **String statement:** CMIS query to be executed. (See section 2.1.10 Query.)

4178**Optional:**

4179 • **Boolean searchAllVersions:**

- 4180 ○ If TRUE, then the Repository MUST include latest and non-latest versions of document  
4181 objects in the query search scope.
- 4182 ○ If FALSE (**default**), then the Repository MUST only include latest versions of documents  
4183 in the query search scope.
- 4184 ○ If the Repository does not support the optional `capabilityAllVersionsSearchable`  
4185 capability, then this parameter value MUST be set to FALSE.

4186 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.

- 4187 ○ Note: For query statements where the SELECT clause contains properties from only one  
4188 virtual table reference (i.e. referenced object-type), any value for this enum may be used.  
4189 If the SELECT clause contains properties from more than one table, then the value of this  
4190 parameter MUST be "none".

4191 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.

4192 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

- 4193 ○ Note: For query statements where the SELECT clause contains properties from only one  
4194 virtual table reference (i.e. referenced object-type), any value for this parameter may be  
4195 used. If the SELECT clause contains properties from more than one table, then the value  
4196 of this parameter MUST be "FALSE".

4197 • **Integer maxItems:** See section 2.2.1.1 Paging.

4198 • **Integer skipCount:** See section 2.2.1.1 Paging.

4199**2.2.6.1.2 Outputs**

4200 • **<Array> Object QueryResults:** The set of results for the query. (See section 2.1.10  
4201 Query.). Each object result MUST include the following elements if they are requested:

- 4202 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 4203 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 4204 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4205 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4206**Optional:**

4207 • **Integer numItems:** See section 2.2.1.1 Paging.

4208

4209**2.2.6.1.3 Exceptions Thrown & Conditions**

4210 • See section [2.2.1.4.1 General Exceptions](#)

- 4211 • If the select clause includes properties from more than a single type reference, then the  
4212 repository SHOULD throw an exception if includeRelationships is something other than "none" or  
4213 includeAllowableActions is specified as TRUE.

4214**2.2.6.2 getContentChanges**

4215**Description:** Gets a list of content changes. This service is intended to be used by search crawlers or  
4216other applications that need to efficiently understand what has changed in the repository.

4217**Notes:**

- 4218 • The content stream is NOT returned for any change event.

- 4219 • The definition of the authority needed to call this service is repository specific.
- 4220 • The latest change log token for a repository can be acquired via the getRepositoryInfo
- 4221 service.

## 42222.2.6.2.1 Inputs

4223**Required:**

- 4224 • **ID repositoryId:** The identifier for the Repository.

4225**Optional:**

- 4226 • **String changeLogToken:**
  - 4227 ○ If specified, then the Repository MUST return the change event corresponding to the value of
  - 4228 the specified change log token as the first result in the output.
  - 4229 ○ If not specified, then the Repository MUST return the first change event recorded in the
  - 4230 change log.
- 4231 • **Boolean includeProperties:**
  - 4232 ○ If TRUE, then the Repository MUST include the updated property values for “updated”
  - 4233 change events.
  - 4234 ○ If FALSE (default), then the Repository MUST NOT include the updated property values for
  - 4235 “updated” change events. The single exception to this is that the objectId MUST always be
  - 4236 included.
- 4237 **Boolean includePolicyIDs:**
  - 4238 If TRUE, then the Repository MUST include the IDs of Policies applied to the object referenced in
  - 4239 each change event, if the change event modified the set of policies applied to the object.
  - 4240 If FALSE (default), then the Repository will not include policy information.
- 4241 **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4242 **Integer maxItems:** See section 2.2.1.1 Paging.

## 42432.2.6.2.2 Outputs

- 4244 • **<Array> changeEvents:** A collection of CMIS objects that MUST include the information
- 4245 as specified in 2.1.11.3. Each result MUST include the following elements if they are requested:
- 4246 ○ **<Array> policyIDs:** The IDs of Policies applied to the object referenced in the change event.
- 4247 ○ **<Array> ACLs:** The ACLs applied to the object reference in the change event.
- 4248 • **String lastChangeLogToken:** The change log token corresponding to the last change
- 4249 event in changeEvents.
- 4250 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.
- 4251**Optional:**
- 4252 • **Integer numItems:** See section 2.2.1.1 Paging.

## 42532.2.6.2.3 Exceptions Thrown & Conditions

- 4254 • See section 2.2.1.4.1 General Exceptions
- 4255 • **constraint:** The Repository MUST throw this exception if the event corresponding to the
- 4256 change log token provided as an input parameter is no longer available in the change log. (E.g.
- 4257 because the change log was truncated).

## 4258 2.2.7 Versioning Services

4259 The Versioning services (checkOut, cancelCheckOut, getPropertiesOfLatestVersion, getAllVersions,  
4260 deleteAllVersions) are used to navigate or update a Document Version Series.

### 4261 2.2.7.1 checkOut

4262 **Description:** Create a private working copy of the document.

#### 4263 2.2.7.1.1 Inputs

4264 **Required:**

- 4265 • **ID repositoryId:** The identifier for the Repository.
- 4266 • **ID objectId:** The identifier of the object.

#### 4267 2.2.7.1.2 Outputs

- 4268 • **ID objectId:** The identifier for the “Private Working Copy” document.
- 4269 • **Boolean contentCopied:** TRUE if the content-stream of the Private Working Copy is a  
4270 copy of the contentStream of the Document that was checked out.
- 4271 • FALSE if the content-stream of the Private Working Copy is “not set”.

#### 4272 2.2.7.1.3 Exceptions Thrown & Conditions

- 4273 • See section [2.2.1.4.1 General Exceptions](#)
- 4274 • **constraint:** The Repository MUST throw this exception if the Document’s Object-Type  
4275 definition’s *versionable* attribute is FALSE.
- 4276 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).
- 4277 • **updateConflict:** See section [2.2.1.4.2 Specific Exceptions](#).
- 4278 • **versioning:** The repository MAY throw this exception if the object is a non-current  
4279 Document Version.

### 4280 2.2.7.2 cancelCheckOut

4281 **Description:** Reverses the effect of a check-out. Removes the private working copy of the checked-out  
4282 document, allowing other documents in the version series to be checked out again.

#### 4283 2.2.7.2.1 Inputs

4284 **Required:**

- 4285 • **ID repositoryId:** The identifier for the Repository.
- 4286 • **ID objectId:** The identifier of the document.

#### 4287 2.2.7.2.2 Exceptions Thrown & Conditions

- 4288 • See section [2.2.1.4.1 General Exceptions](#)
- 4289 • **constraint:** The Repository MUST throw this exception if the Document’s Object-Type  
4290 definition’s *versionable* attribute is FALSE.
- 4291 • **updateConflict:** See section [2.2.1.4.2 Specific Exceptions](#).
- 4292 • **versioning:** The repository MAY throw this exception if the object is a non-current  
4293 Document Version.

### 42942.2.7.3 checkIn

4295**Description:** Checks-in the Private Working Copy document.

4296**Notes:**

- 4297 • For repositories that do NOT support the optional “[capabilityPWCUpdatable](#)” **capability**, the  
4298 *properties* and *contentStream* input parameters MUST be provided on the checkIn method for  
4299 updates to happen as part of checkIn.
- 4300 • Each CMIS protocol bindings MUST specify whether the checkin service MUST always include all  
4301 updatable properties, or only those properties whose values are different than the original value  
4302 of the object.

### 43032.2.7.3.1 Inputs

4304**Required:**

- 4305 • **ID repositoryId:** The identifier for the Repository.
- 4306 • **ID objectId:** The identifier of the document.

4307**Optional:**

- 4308 • **Boolean major:** TRUE (**default**) if the checked-in Document Object MUST be a major version.  
4309     ◦ FALSE if the checked-in Document Object MUST NOT be a major version.
- 4310 • **<Array> properties:** The property values that MUST be applied to the checked-in Document  
4311 Object.
- 4312 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the checked-in  
4313 Document Object. The method of passing the contentStream to the server and the encoding  
4314 mechanism will be specified by each specific binding.
- 4315 • **String checkinComment:** See section 2.1.9.5 Versioning Properties on Document Objects.
- 4316 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document  
4317 object.
- 4318 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document  
4319 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 4320 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
4321 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is  
4322 specified.

### 43232.2.7.3.2 Outputs

- 4324 • **ID objectId:** The ID of the checked-in document.

### 43252.2.7.3.3 Exceptions Thrown & Conditions

- 4326 • See section [2.2.1.4.1 General Exceptions](#)
- 4327 • **constraint:** The Repository MUST throw this exception if the Document’s Object-Type  
4328 definition’s *versionable* attribute is FALSE.
- 4329 • **storage:** See section [2.2.1.4.2 Specific Exceptions](#).
- 4330 • **streamNotSupported:** The Repository MUST throw this exception if the  
4331 “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId  
4332 property value is set to “not allowed” and a contentStream input parameter is provided.
- 4333 • **updateConflict:** See section [2.2.1.4.2 Specific Exceptions](#).

#### 4334 **2.2.7.4 getObjectOfLatestVersion**

4335 **Description:** Get a the latest Document object in the Version Series.

##### 4336 **2.2.7.4.1 Inputs**

4337 **Required:**

- 4338 • **ID repositoryId:** The identifier for the Repository.
- 4339 • **ID objectId:** The identifier for the Version Series.

4340 **Optional:**

- 4341 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major  
4342 version object in the Version Series.
  - 4343 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or  
4344 non-major) version object in the Version Series.
- 4345 • **String filter:** See Error: Reference source not found section 2.2.1.2.1 Properties.
- 4346 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4347 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 4348 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4349 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4350 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

##### 4351 **2.2.7.4.2 Outputs**

- 4352 • **<Array> Properties:** The list of properties for the object.
- 4353 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 4354 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 4355 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 4356 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 4357 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

##### 4358 **2.2.7.4.3 Exceptions Thrown & Conditions**

- 4359 • See section [2.2.1.4.1 General Exceptions](#)
- 4360 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
4361 parameter is not valid.
- 4362 • **objectNotFound:** The Repository MUST throw this exception if the input parameter  
4363 major is TRUE and the Version Series contains no major versions.

#### 4364 **2.2.7.5 getPropertiesOfLatestVersion**

4365 **Description:** Get a subset of the properties for the latest Document Object in the Version Series.

##### 4366 **2.2.7.5.1 Inputs**

4367 **Required:**

- 4368 • **ID repositoryId:** The identifier for the Repository.
- 4369 • **ID objectId:** The identifier for the Version Series.

4370 **Optional:**

- 4371 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major  
4372 version object in the Version Series.
- 4373 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or  
4374 non-major) version object in the Version Series.
- 4375 • **String filter:** See Error: Reference source not foundsection 2.2.1.2.1 Properties.

## 43762.2.7.5.2 Outputs

4377 **<Array> Properties:** The list of properties for the object.

## 43782.2.7.5.3 Exceptions Thrown & Conditions

- 4379 • See section [2.2.1.4.1 General Exceptions](#)
- 4380 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
4381 parameter is not valid.
- 4382 • **objectNotFound:** The Repository MUST throw this exception if the input parameter major is  
4383 TRUE and the Version Series contains no major versions.

## 43842.2.7.6 getAllVersions

4385**Description:** Returns the list of all Document Objects in the specified Version Series, sorted by  
4386cmis:creationDate descending.

4387**Notes:**

- 4388 • The result set for this operation MUST include the Private Working Copy, subject to caller's  
4389 access privileges.

## 43902.2.7.6.1 Inputs

4391**Required:**

- 4392 • **ID repositoryId:** The identifier for the Repository.
- 4393 • **ID objectId:** The identifier for the Version Series.

4394**Optional:**

- 4395 • **String filter:** See Error: Reference source not foundsection 2.2.1.2.1 Properties.
- 4396 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

## 43972.2.7.6.2 Outputs

- 4398 • **<Array> ObjectResults:** A list of Document Objects in the specified Version Series.  
4399 Each object result MUST include the following elements if they are requested:
- 4400 ○ **<Array> Properties:** The list of properties for the object.
- 4401 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4402

## 44032.2.7.6.3 Exceptions Thrown & Conditions

- 4404 • See section [2.2.1.4.1 General Exceptions](#)
- 4405 • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
4406 parameter is not valid.



## 44072.2.8 Relationship Services

4408The Relationship Services (*getObjectRelationships*) are used to retrieve the dependent Relationship  
4409objects associated with an independent object.

### 44102.2.8.1 getObjectRelationships

4411**Description:** Gets all or a subset of relationships associated with an independent object.

#### 44122.2.8.1.1 Inputs

4413**Required:**

4414 **ID repositoryId:** The identifier for the Repository.

4415 • **ID objectId:** The identifier of the object.

4416

4417**Optional:**

4418 • **Boolean includeSubRelationshipTypes:** If TRUE, then the Repository MUST return all  
4419 relationships whose Object-Types are descendant-types of the given object's cmis:objectTypeId  
4420 property value.

4421 ○ Default is FALSE

4422 ○ If FALSE, then the Repository MUST only return relationships whose Object-Type is  
4423 equivalent to the given object's cmis:objectTypeId property value.

4424 • **Enum relationshipDirection:** An enumeration specifying whether the Repository  
4425 MUST return relationships where the specified Object is the source of the relationship, the target  
4426 of the relationship, or both. Valid values are:

4427 ○ **source: (default)** The Repository MUST return only relationship objects where the specified  
4428 object is the source object.

4429 ○ **target:** The Repository MUST return only relationship objects where the specified object is  
4430 the target object.

4431 ○ **either:** The Repository MUST return relationship objects where the specified object is  
4432 either the source or the target object.

4433 • **ID typeId:** If specified, then the Repository MUST return only relationships whose  
4434 Object-Type is of the type specified (and possibly its descendant-types – see next parameter.)

4435 ○ If not specified, then the repository MUST return Relationship objects of all types.

4436 • **Integer maxItems:** See section 2.2.1.1 Paging.

4437 • **Integer skipCount:** See section 2.2.1.1 Paging.

4438 • **String filter:** See section 2.2.1.2.1 Properties.

4439 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

#### 44402.2.8.1.2 Outputs

4441 • **<Array> Objects:** A list of the relationship objects. Each object result MUST include the  
4442 following elements if they are requested:

4443 ○ **<Array> Properties:** The list of properties for the object.

4444 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4445 **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4446**Optional:**

4447 • **Integer numItems:** See section 2.2.1.1 Paging.



4448

### 4449 **2.2.8.1.3 Exceptions Thrown & Conditions**

- 4450 • See section [2.2.1.4.1 General Exceptions](#)
- 4451 • `filterNotValid`: The Repository MUST throw this exception if this property filter input  
4452 parameter is not valid.

## 4453 **2.2.9 Policy Services**

4454 The Policy Services (*applyPolicy*, *removePolicy*, *getAppliedPolicies*) are used to apply or remove a policy  
4455 object to a controllablePolicy object.

### 4456 **2.2.9.1 applyPolicy**

4457 **Description:** Applies a specified policy to an object.

#### 4458 **2.2.9.1.1 Inputs**

4459 **Required:**

- 4460 • **ID repositoryId:** The identifier for the Repository.
- 4461 • **ID policyId:** The identifier for the Policy to be applied.
- 4462 • **ID objectId:** The identifier of the object.

### 4463 **2.2.9.1.2 Exceptions Thrown & Conditions**

- 4464 See section [2.2.1.4.1 General Exceptions](#)
- 4465 `constraint`: The Repository MUST throw this exception if the specified object's Object-Type  
4466 definition's attribute for *controllablePolicy* is FALSE.

### 4467 **2.2.9.2 removePolicy**

4468 **Description:** Removes a specified policy from an object.

#### 4469 **2.2.9.2.1 Inputs**

4470 **Required:**

- 4471 • **ID repositoryId:** The identifier for the Repository.
- 4472 • **ID policyId:** The identifier for the Policy to be removed.
- 4473 • **ID objectId:** The identifier of the object.

### 4474 **2.2.9.2.2 Exceptions Thrown & Conditions**

- 4475 • See section [2.2.1.4.1 General Exceptions](#)
- 4476 • `constraint`: The Repository MUST throw this exception if the specified object's Object-  
4477 Type definition's attribute for *controllablePolicy* is FALSE.

### 4478 **2.2.9.3 getAppliedPolicies**

4479 **Description:** Gets the list of policies currently applied to the specified object.

#### 4480 **2.2.9.3.1 Inputs**

4481 **Required:**

4482       •       **ID repositoryId:** The identifier for the Repository.

4483       •       **ID objectId:** The identifier of the object.

4484**Optional:**

4485       **String filter:** See Error: Reference source not foundsection 2.2.1.2.1 Properties.

## 4486**2.2.9.3.2 Outputs**

4487       **<Array> Objects:** A list of Policy Objects.

## 4488**2.2.9.3.3 Exceptions Thrown & Conditions**

4489       •       See section [2.2.1.4.1 General Exceptions](#)

4490       •       **filterNotValid:** The Repository MUST throw this exception if this property filter input  
4491       parameter is not valid.

## 4492**2.2.10 ACL Services**

### 4493**2.2.10.1 getACL**

4494**Description:** Get the ACL currently applied to the specified document or folder object.

### 4495**2.2.10.1.1 Inputs**

4496**Required:**

4497       •       **ID repositoryId:** The identifier for the repository.

4498       •       **ID objectId:** The identifier for the object

4499**Optional:**

4500       •       **Boolean onlyBasicPermissions:** See section 2.8 Access Control. The repository SHOULD  
4501       make a best effort to fully express the native security applied to the object

4502           ○ **TRUE:** (default value if not provided) indicates that the client requests that the returned  
4503           ACL be expressed using only the CMIS Basic permissions.

4504           ○ **FALSE:** indicates that the server may respond using either solely CMIS Basic  
4505           permissions, or repository specific permissions or some combination of both.

### 4506**2.2.10.1.2 Outputs**

4507       •       **<Array> AccessControlEntryType:** The list of access control entries of the ACL for the object.

4508 **Optional:**

4509       •       **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –  
4510       i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.

### 4511**2.2.10.1.3 Exceptions Thrown & Conditions**

4512       •       See section [2.2.1.4.1 General Exceptions](#)

### 4513**2.2.10.1.4 Notes**

4514This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=discover* or  
4515*=manage*.

4516How an ACL for the object is computed is up to the repository. A client MUST NOT assume that the ACEs  
4517from the ACL as returned by this service can be applied via *applyACL*.

## 45182.2.10.2 applyACL

4519**Description:** Adds or removes the given ACEs to or from the ACL of document or folder object.

## 45202.2.10.2.1 Inputs

4521**Required:**

4522 • **ID repositoryId:** The identifier for the repository.

4523 • **ID objectId:** The identifier for the object

4524**Optional:**

4525 • **<Array> AccessControlEntryType addACEs:** The ACEs to be added.

4526 • **<Array> AccessControlEntryType removeACEs:** The ACEs to be removed.

4527 • **Enum ACLPropagation:** Specifies how ACEs should be handled:

4528     o **objectonly:** ACEs must be applied without changing the ACLs of other objects.

4529     o **propagate:** ACEs must be applied by propagate the changes to all “inheriting” objects.

4530     o **repositorydetermined: Default value.** Indicates that the client leaves the behavior  
4531 to the repository.

## 45322.2.10.2.2 Outputs

4533 • **<Array> AccessControlEntryType:** The list of access control entries of the resulting ACL for the  
4534 object

4535**Optional:**

4536 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –  
4537 i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.

4538 • **String changeToken:** See section 2.2.1.3 Change Tokens.

## 45392.2.10.2.3 Exceptions Thrown & Conditions

4540 • See section [2.2.1.4.1 General Exceptions](#)

4541 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are  
4542 met:

4543     o The specified object's Object-Type definition's attribute for *controllableACL* is FALSE.

4544     o The value for *ACLPropagation* does not match the values as returned via  
4545 *getACLCapabilities*.

4546     o At least one of the *specified values for permission* in ANY of the *ACEs does not match*  
4547 *ANY of the permissionNames as returned by getACLCapability and is not a CMIS Basic*  
4548 *permission*

## 45492.2.10.2.4 Notes

4550This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=manage*.

4551How ACEs are added or removed to or from the object is up to the repository – with respect to the  
4552*ACLPropagation* provided by the client. For “shared” ACEs (e.g. via inheritance), the repository MAY  
4553merge the ACEs provided with the ACEs of the ACL already applied to the object (i.e. the ACEs provided  
4554MAY not be completely added or removed from the effective ACL for the object).

4555

4556

---

## 4557 **3 Restful AtomPub Binding**

### 4558 **3.1 Overview**

4559 This binding is based upon the Atom (RFC4287) and Atom Publishing Protocol (RFC5023).

4560 Implementations of CMIS MUST be compliant with RFC4287 and RFC5023.

4561

4562 In this binding, the client interacts with the repository by acquiring the service document. The client will  
4563 request the service document by the URI provided by the vendor. The client will then choose a CMIS  
4564 collection, and then start accessing the repository by following the references in the returned documents.

4565

4566 This binding consists of a service document specifying at least CMIS service collections, atom collections,  
4567 feeds and entry documents. CMIS extends the Atom and AtomPub documents utilizing the Atom and  
4568 AtomPub extension mechanism. CMIS also leverages link tags to specify additional resources related to  
4569 the requested resource.

4570

4571 When requesting a resource, optional parameters may be specified to change default behavior via query  
4572 parameters.

### 4573 **3.1.1 Namespaces**

4574 This specification uses the following namespaces and prefixes when referring to xml or xml schema  
4575 elements in the text or examples:

- 4576 • CMIS-Core: <http://docs.oasis-open.org/ns/cmis/core/200908/>
  - 4577 ○ Prefix: cmis
- 4578 • CMIS-RestAtom: <http://docs.oasis-open.org/ns/cmis/restatom/200908/>
  - 4579 ○ Prefix: cmisra
- 4580 • Atom : <http://www.w3.org/2005/Atom>
  - 4581 ○ Prefix: atom
- 4582 • AtomPub: <http://www.w3.org/2007/app>
  - 4583 ○ Prefix: app

4584

### 4585 **3.1.2 Authentication**

4586 Authentication SHOULD be handled by the transport protocol. Please see AtomPub (RFC5023) section  
4587 14.

4588

### 4589 **3.1.3 Response Formats**

4590 The client can specify, in HTTP the Accept header, which formats are acceptable to the client. With this  
4591 mechanism the client can chose which response format the CMIS implementation should respond with.  
4592 The CMIS compliant implementation MUST support the appropriate Media Types specified in this  
4593 document.

### 4594**3.1.4 Optional Arguments**

4595The binding supports adding optional parameters to CMIS resources to modify the default behavior.

4596CMIS implementations MUST support arguments being specified as HTTP query string parameters.

4597Names and valid values for HTTP query string parameters are as described in the appropriate CMIS

4598Service descriptions [see CMIS Domain Model]. Valid values of enumeration types are also represented

4599in the CMIS Core XML Schema

### 4600**3.1.5 Errors and Exceptions**

4601Exceptions MUST be mapped to the appropriate HTTP status code.

4602Repositories SHOULD provide sufficient information in the body of the HTTP response for a user to

4603determine corrective action.

4604See Section 3.2.4 HTTP Status Codes for more information.

### 4605**3.1.6 Renditions**

4606Each Rendition included in a CMIS AtomPub response is represented as an Atom link with relationship

4607alternate.

4608

4609The following attributes SHOULD be included on the link element:

4610     • href: URI to the rendition content stream

4611     • type: The Media Type of the Rendition

4612     • cmisra:renditionKind: The Rendition Kind for the Rendition

4613

4614The following attributes MAY be included

4615     • title: The Filename (or name property if object) of Rendition

4616     • length: The length of the rendition

### 4617**3.1.7 Content Streams**

4618The content stream for a document SHOULD be referenced by the content src attribute as well as the

4619edit-media link relation.

4620A CMIS Repository MAY use different URIs for both content src attribute and the edit-media link relation

4621for the same content stream.

4622The following attributes SHOULD be included on the link element:

4623     • href: URI to the content stream

4624     • type: The Media Type of the content stream

### 4625**3.1.8 Paging of Feeds**

4626For paging, please see the AtomPub RFC. CMIS leverages first, next, previous, and last link relations to

4627express paging.

4628If the repository can include the number of items (numItems in CMIS Domain Model) in a feed, then the

4629repository SHOULD include the cmisra:numItems extension element in the feed.

### 4630**3.1.9 Services not Exposed**

4631The following services are not exposed in this binding:

4632     • getRenditions: This is exposed as part of getObject

- 4633 • getProperties: This is exposed as part of getObject
- 4634 • createDocumentFromSource: This is not exposed in this binding except as the client saving the
- 4635 resource and resubmitting it without the cmis:objectId.
- 4636
- 4637

## 46383.2 HTTP

### 46393.2.1 Entity Tag

4640CMIS changeTokens are represented as Entity Tags and follow HTTP's use of Entity Tags. CMIS server  
4641implementations SHOULD support Entity Tags. ChangeTokens are also provided as properties and  
4642SHOULD be provided when the object is included inside an atom entry or feed.

### 46433.2.2 HTTP Range

4644Repositories MAY support HTTP Range requests on Content Streams.

### 46453.2.3 HTTP OPTIONS Method

4646The repository MAY support the HTTP OPTIONS method on all the resources defined in this  
4647specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP  
4648methods specified for that resource in the Allow header.

### 46493.2.4 HTTP Status Codes

#### 46503.2.4.1 Common CMIS Exceptions

4651The following listing defines the HTTP status codes that repositories MUST return for the various common  
4652exceptions defined in CMIS Domain Model.

4653	CMIS Services Exception	HTTP Status Code
4654	invalidArgument	400
4655	objectNotFound	404
4656	permissionDenied	403
4657	operationNotSupported	405
4658	updateConflict	409
4659	runtime	500

#### 46603.2.4.2 Other Exceptions

4661	CMIS Services Exception	HTTP Status Code
4662	constraint	409
4663	filterNotValid	400
4664	streamNotSupported	403
4665	storage	500
4666	contentAlreadyExists	409
4667	versioning	409

#### 46683.2.4.3 Notable HTTP Status Codes

- 4669 • 415 Unsupported Media Type

4670           ○ When a document is POST'ed to a collection that does not support the media type of the  
4671           document, this status code MUST be returned  
4672       • 422 Unprocessable Entity (Defined in RFC4918 Section 11.2)  
4673           ○ When a request has been POST'ed but cannot be processed, this status code MUST be  
4674           returned  
4675  
4676Please see RFC2616 Section 10 for more information.  
4677

### 46783.3 Media Types

4679CMIS introduces new media types for:  
4680       • a CMIS Query document (application/cmismquery+xml)  
4681       • a CMIS AllowableActions document (application/cmismallowableactions+xml)  
4682       • an Atom Document (Entry or Feed) with any CMIS Markup (application/cmismatom+xml)  
4683       • an Atom Feed Document with CMIS Hierarchy extensions (application/cmismtree+xml)  
4684  
4685In addition to those media types specified by CMIS, CMIS also leverages these media types:  
4686       • AtomPub Service (application/atomsvc+xml)  
4687       • Atom Entry (application/atom+xml;type=entry)  
4688       • Atom Feed (application/atom+xml;type=feed)

#### 46893.3.1 CMIS Atom

4690Media Type: application/cmismatom+xml  
4691Starting tag: atom:feed or atom:entry  
4692Type Parameters:  
4693       • type – the semantics of the type parameter MUST be the same as the media type parameter for  
4694       atom documents.  
4695  
4696This allows clients to differentiate between repositories that require atom media type with CMIS  
4697extensions (application/cmismatom+xml) for creation and repositories that allow generic atom media type  
4698without CMIS extensions (application/atom+xml).  
4699  
4700This is only used for CMIS repositories to advertise what media types are accepted for adding to a  
4701collection (e.g., creating resources in a collection).  
4702  
4703All feeds and entries from a CMIS repository MUST utilize the atom media type for exposing Atom  
4704resources. Please see the individual resources for more information on the media type. This provides  
4705the interoperability with Atom clients.  
4706  
4707Example:  
4708  
4709       <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

4710 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4711 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4712 xmlns:atom="http://www.w3.org/2005/Atom"
4713 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4714 open.org/ns/cmis/restatom/200908/">
4715   <atom:author>
4716     <atom:name>Al Brown</atom:name>
4717   </atom:author>
4718   <atom:id>urn:uuid:5f716462-7b6b-4a0d-bd1d-34677d5514d7</atom:id>
4719   <atom:title type="text">CMIS Example Document</atom:title>
4720   <atom:updated>2009-10-19T10:09:59.296-07:00</atom:updated>
4721   <atom:content type="text">some text</atom:content>
4722   <cmisra:object>
4723     <cmis:properties>
4724       <cmis:propertyId localName="rep-cmis:objectTypeId"
4725 propertyDefinitionId="cmis:objectTypeId">
4726         <cmis:value>invoice</cmis:value>
4727       </cmis:propertyId>
4728       <cmis:propertyString localName="rep-cmis:name"
4729 propertyDefinitionId="cmis:name">
4730         <cmis:value>CMIS Example Document</cmis:value>
4731       </cmis:propertyString>
4732     </cmis:properties>
4733   </cmisra:object>
4734 </atom:entry>
4735

```

### 47363.3.2 CMIS Query

4737Media Type: application/cmisquery+xml

4738Starting tag: cmis:query

4739

4740This document contains the representation of a query to be executed in a CMIS repository.

4741

4742Example:

```

4743 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4744 <cmis:query xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4745 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4746 xmlns:atom="http://www.w3.org/2005/Atom"
4747 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4748 open.org/ns/cmis/restatom/200908/">
4749   <cmis:statement>SELECT * FROM cmis:document</cmis:statement>
4750   <cmis:searchAllVersions>true</cmis:searchAllVersions>
4751   <cmis:includeAllowableActions>>false</cmis:includeAllowableActions>
4752   <cmis:includeRelationships>none</cmis:includeRelationships>
4753   <cmis:renditionFilter>*</cmis:renditionFilter>
4754   <cmis:maxItems>50</cmis:maxItems>
4755   <cmis:skipCount>0</cmis:skipCount>
4756 </cmis:query>

```

4757

4758Please also see the example documents included with the schema.

4759

### 47603.3.3 CMIS Allowable Actions

4761Media Type: application/cmisallowableactions+xml

4762Starting tag: cmis:allowableactions



4763

4764 This document contains the representation of the allowable actions the user may perform on the  
4765 referenced object.

4766

4767 Example:

```
4768 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4769 <cmis:allowableActions xmlns:cmis="http://docs.oasis-
4770 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
4771 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
4772 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4773 open.org/ns/cmis/restatom/200908/">
4774   <cmis:canDeleteObject>true</cmis:canDeleteObject>
4775   <cmis:canUpdateProperties>true</cmis:canUpdateProperties>
4776   <cmis:canGetProperties>true</cmis:canGetProperties>
4777   <cmis:canGetObjectRelationships>true</cmis:canGetObjectRelationships>
4778   <cmis:canGetObjectParents>true</cmis:canGetObjectParents>
4779   <cmis:canMoveObject>true</cmis:canMoveObject>
4780   <cmis:canDeleteContentStream>true</cmis:canDeleteContentStream>
4781   <cmis:canCheckOut>true</cmis:canCheckOut>
4782   <cmis:canCancelCheckOut>true</cmis:canCancelCheckOut>
4783   <cmis:canCheckIn>true</cmis:canCheckIn>
4784   <cmis:canSetContentStream>true</cmis:canSetContentStream>
4785   <cmis:canGetAllVersions>true</cmis:canGetAllVersions>
4786   <cmis:canAddObjectToFolder>true</cmis:canAddObjectToFolder>
4787   <cmis:canRemoveObjectFromFolder>true</cmis:canRemoveObjectFromFolder>
4788   <cmis:canGetContentStream>true</cmis:canGetContentStream>
4789   <cmis:canApplyPolicy>true</cmis:canApplyPolicy>
4790   <cmis:canGetAppliedPolicies>true</cmis:canGetAppliedPolicies>
4791   <cmis:canRemovePolicy>true</cmis:canRemovePolicy>
4792   <cmis:canCreateDocument>true</cmis:canCreateDocument>
4793 </cmis:allowableActions>
```

4794

4795 Please also see the example documents included with the schema.

4796

### 4797 3.3.4 CMIS Tree

4798 Media Type: application/cmistree+xml

4799 Starting tag: atom:feed

4800

4801 This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.

4802

4803 Please see Section 3.5.2.1 for more information.

4804

4805 Example:

```
4806 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4807 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4808 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4809 xmlns:atom="http://www.w3.org/2005/Atom"
4810 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4811 open.org/ns/cmis/restatom/200908/">
4812   <atom:title type="text">Feed for folder1</atom:title>
4813   <atom:author>
4814     <atom:name>Al Brown</atom:name>
4815     <atom:uri>http://www.ibm.com/</atom:uri>
```

```

4816         <atom:email>albertcbrown@us.ibm.com</atom:email>
4817     </atom:author>
4818     <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4819     <atom:id>urn:uuid:08d13325-cba1-4290-95e3-f485068b08e8</atom:id>
4820     <atom:link type="application/atom+xml;type=feed" rel="self"
4821 href="http://cmisexample.oasis-open.org/rep1/4f942e30-895c-4c82-9b31-
4822 58f639b6c2d7/3"/>
4823     <atom:link type="application/atomsvc+xml" rel="service"
4824 href="http://cmisexample.oasis-open.org/rep1//service"/>
4825     <atom:link type="application/atom+xml;type=entry" rel="via"
4826 href="http://cmisexample.oasis-open.org/rep1/4f942e30-895c-4c82-9b31-
4827 58f639b6c2d7"/>
4828     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
4829 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
4830 open.org/rep1/4f942e30-895c-4c82-9b31-58f639b6c2d7/foldertree"/>
4831     <atom:link type="application/atom+xml;type=feed" rel="down"
4832 href="http://cmisexample.oasis-open.org/rep1/4f942e30-895c-4c82-9b31-
4833 58f639b6c2d7/children"/>
4834     <atom:link type="application/atom+xml;type=entry" rel="up"
4835 href="http://cmisexample.oasis-open.org/rep1/b8803070-0253-4160-9ad6-
4836 26ad7adbf561"/>
4837     <atom:entry>
4838         <atom:author>
4839             <atom:name>Al Brown</atom:name>
4840             <atom:uri>http://www.ibm.com/</atom:uri>
4841             <atom:email>albertcbrown@us.ibm.com</atom:email>
4842         </atom:author>
4843         <atom:content src="http://cmisexample.oasis-open.org/rep1/962b4fb8-
4844 aeafa-4d0c-8562-a84e297e8107"/>
4845         <atom:id>urn:uuid:962b4fb8-aeafa-4d0c-8562-a84e297e8107</atom:id>
4846         <atom:title type="text">CMIS Example Folder as Customer
4847 type</atom:title>
4848         <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4849         <atom:link rel="self" href="http://cmisexample.oasis-
4850 open.org/rep1/962b4fb8-aeafa-4d0c-8562-a84e297e8107"/>
4851         <atom:link rel="edit" href="http://cmisexample.oasis-
4852 open.org/rep1/962b4fb8-aeafa-4d0c-8562-a84e297e8107"/>
4853         <atom:link type="application/cmis+xml;type=allowableActions"
4854 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
4855 href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aeafa-4d0c-8562-
4856 a84e297e8107/allowableactions"/>
4857         <atom:link type="application/atom+xml;type=entry" rel="describedby"
4858 href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aeafa-4d0c-8562-
4859 a84e297e8107/type"/>
4860         <atom:link type="application/atomsvc+xml" rel="service"
4861 href="http://cmisexample.oasis-open.org/rep1//service"/>
4862         <atom:published>2009-10-19T10:09:59.515-07:00</atom:published>
4863         <atom:summary type="html">HTML summary of Entry 962b4fb8-aeafa-4d0c-
4864 8562-a84e297e8107</atom:summary>
4865         <atom:link type="application/atom+xml;type=entry" rel="up"
4866 href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aeafa-4d0c-8562-
4867 a84e297e8107/up"/>
4868         <atom:link type="application/atom+xml;type=feed" rel="down"
4869 href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aeafa-4d0c-8562-
4870 a84e297e8107/children"/>
4871         <atom:link type="application/cmistree+xml" rel="down"
4872 href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aeafa-4d0c-8562-
4873 a84e297e8107/tree"/>
4874         <atom:link type="application/atom+xml;type=feed"
4875 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
4876 href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aeafa-4d0c-8562-
4877 a84e297e8107/foldertree"/>

```

```

4878     <atom:link type="application/atom+xml;type=feed"
4879 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
4880 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aeefa-4d0c-8562-
4881 a84e297e8107/relationships"/>
4882     <atom:link type="application/atom+xml;type=feed"
4883 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
4884 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aeefa-4d0c-8562-
4885 a84e297e8107/policies"/>
4886     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
4887 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
4888 open.org/repl/962b4fb8-aeefa-4d0c-8562-a84e297e8107/acl"/>
4889     <cmisra:object>
4890       <cmis:properties>
4891         <cmis:propertyId localName="rep-cmisis:objectId"
4892 propertyDefinitionId="cmis:objectId">
4893           <cmis:value>962b4fb8-aeefa-4d0c-8562-
4894 a84e297e8107</cmis:value>
4895         </cmis:propertyId>
4896       </cmis:properties>
4897     </cmisra:object>
4898     <cmisra:pathSegment>customer</cmisra:pathSegment>
4899     <cmisra:children>
4900       <atom:title type="text">CMIS Example Folder as Customer
4901 type</atom:title>
4902       <atom:author>
4903         <atom:name>Al Brown</atom:name>
4904         <atom:uri>http://www.ibm.com/</atom:uri>
4905         <atom:email>albertcbrown@us.ibm.com</atom:email>
4906       </atom:author>
4907       <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4908       <atom:id>urn:uuid:084edab9-418a-42b9-813f-cf450418181d</atom:id>
4909       <atom:link type="application/atom+xml;type=feed" rel="self"
4910 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aeefa-4d0c-8562-
4911 a84e297e8107/3"/>
4912       <atom:link type="application/atomsvc+xml" rel="service"
4913 href="http://cmisexample.oasis-open.org/repl//service"/>
4914       <atom:link type="application/atom+xml;type=entry" rel="via"
4915 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aeefa-4d0c-8562-
4916 a84e297e8107"/>
4917       <atom:link type="application/atom+xml;type=feed"
4918 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
4919 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aeefa-4d0c-8562-
4920 a84e297e8107/foldertree"/>
4921       <atom:link type="application/atom+xml;type=feed" rel="down"
4922 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aeefa-4d0c-8562-
4923 a84e297e8107/children"/>
4924       <atom:link type="application/atom+xml;type=entry" rel="up"
4925 href="http://cmisexample.oasis-open.org/repl/4f942e30-895c-4c82-9b31-
4926 58f639b6c2d7"/>
4927     <atom:entry>
4928       <atom:author>
4929         <atom:name>Al Brown</atom:name>
4930         <atom:uri>http://www.ibm.com/</atom:uri>
4931         <atom:email>albertcbrown@us.ibm.com</atom:email>
4932       </atom:author>
4933       <atom:content src="http://cmisexample.oasis-
4934 open.org/repl/fba295fd-c8ed-40d1-8ddc-93671a9c89a5"/>
4935       <atom:id>urn:uuid:fba295fd-c8ed-40d1-8ddc-
4936 93671a9c89a5</atom:id>
4937       <atom:title type="text">CMIS Example Doc as Invoice
4938 type</atom:title>
4939       <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>

```

```

4940         <atom:link rel="self" href="http://cmisexample.oasis-
4941 open.org/rep1/fba295fd-c8ed-40d1-8ddc-93671a9c89a5"/>
4942         <atom:link rel="edit" href="http://cmisexample.oasis-
4943 open.org/rep1/fba295fd-c8ed-40d1-8ddc-93671a9c89a5"/>
4944         <atom:link type="application/cmismedia+xml;type=allowableActions"
4945 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
4946 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4947 93671a9c89a5/allowableactions"/>
4948         <atom:link type="application/atom+xml;type=entry"
4949 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-
4950 40d1-8ddc-93671a9c89a5/type"/>
4951         <atom:link type="application/atomsvc+xml" rel="service"
4952 href="http://cmisexample.oasis-open.org/rep1//service"/>
4953         <atom:published>2009-10-19T10:09:59.515-07:00</atom:published>
4954         <atom:summary type="html">HTML summary of Entry fba295fd-
4955 c8ed-40d1-8ddc-93671a9c89a5</atom:summary>
4956         <atom:link rel="edit-media" href="http://cmisexample.oasis-
4957 open.org/rep1/fba295fd-c8ed-40d1-8ddc-93671a9c89a5/edit-media"/>
4958         <atom:link rel="alternate" href="http://cmisexample.oasis-
4959 open.org/rep1/fba295fd-c8ed-40d1-8ddc-93671a9c89a5/alternate"/>
4960         <atom:link type="application/atom+xml;type=feed" rel="up"
4961 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4962 93671a9c89a5/parents"/>
4963         <atom:link type="application/atom+xml;type=feed" rel="version-
4964 history" href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4965 93671a9c89a5/allversions"/>
4966         <atom:link type="application/atom+xml;type=entry"
4967 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/fba295fd-
4968 c8ed-40d1-8ddc-93671a9c89a5/latest"/>
4969         <atom:link type="application/atom+xml;type=feed"
4970 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
4971 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4972 93671a9c89a5/relationships"/>
4973         <atom:link type="application/atom+xml;type=feed"
4974 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
4975 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4976 93671a9c89a5/policies"/>
4977         <atom:link type="application/cmisacl+xml"
4978 rel="http://docs.oasis-open.org/ns/cmis/link/200908/acl"
4979 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4980 93671a9c89a5/acl"/>
4981         <cmisra:object>
4982             <cmis:properties>
4983                 <cmis:propertyId localName="rep-cmis:objectId"
4984 propertyDefinitionId="cmis:objectId">
4985                     <cmis:value>fba295fd-c8ed-40d1-8ddc-
4986 93671a9c89a5</cmis:value>
4987                 </cmis:propertyId>
4988             </cmis:properties>
4989         </cmisra:object>
4990         <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
4991     </atom:entry>
4992 </cmisra:children>
4993 </atom:entry>
4994 <atom:entry>
4995     <atom:author>
4996         <atom:name>Al Brown</atom:name>
4997         <atom:uri>http://www.ibm.com/</atom:uri>
4998         <atom:email>albertbrown@us.ibm.com</atom:email>
4999     </atom:author>
5000     <atom:content src="http://cmisexample.oasis-open.org/rep1/affc2158-
5001 6c8f-4245-a045-688225f8c2ad"/>
5002     <atom:id>urn:uuid:affc2158-6c8f-4245-a045-688225f8c2ad</atom:id>

```

```

5003     <atom:title type="text">CMIS Example Doc as Invoice type</atom:title>
5004     <atom:updated>2009-10-19T10:09:59.531-07:00</atom:updated>
5005     <atom:link rel="self" href="http://cmisexample.oasis-
5006 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad"/>
5007     <atom:link rel="edit" href="http://cmisexample.oasis-
5008 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad"/>
5009     <atom:link type="application/cmismedia+xml;type=allowableActions"
5010 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/allowableactions"
5011 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5012 688225f8c2ad/allowableactions"/>
5013     <atom:link type="application/atom+xml;type=entry" rel="describedby"
5014 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5015 688225f8c2ad/type"/>
5016     <atom:link type="application/atomsvc+xml" rel="service"
5017 href="http://cmisexample.oasis-open.org/rep1//service"/>
5018     <atom:published>2009-10-19T10:09:59.531-07:00</atom:published>
5019     <atom:summary type="html">HTML summary of Entry affc2158-6c8f-4245-
5020 a045-688225f8c2ad</atom:summary>
5021     <atom:link rel="edit-media" href="http://cmisexample.oasis-
5022 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad/edit-media"/>
5023     <atom:link rel="alternate" href="http://cmisexample.oasis-
5024 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad/alternate"/>
5025     <atom:link type="application/atom+xml;type=feed" rel="up"
5026 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5027 688225f8c2ad/parents"/>
5028     <atom:link type="application/atom+xml;type=feed" rel="version-history"
5029 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5030 688225f8c2ad/allversions"/>
5031     <atom:link type="application/atom+xml;type=entry" rel="current-
5032 version" href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5033 688225f8c2ad/latest"/>
5034     <atom:link type="application/atom+xml;type=feed"
5035 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/relationships"
5036 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5037 688225f8c2ad/relationships"/>
5038     <atom:link type="application/atom+xml;type=feed"
5039 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/policies"
5040 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
5041 688225f8c2ad/policies"/>
5042     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
5043 open.org/ns/cmismedia/link/200908/acl" href="http://cmisexample.oasis-
5044 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad/acl"/>
5045     <cmisra:object>
5046       <cmis:properties>
5047         <cmis:propertyId localName="rep-cmis:objectId"
5048 propertyDefinitionId="cmis:objectId">
5049           <cmis:value>affc2158-6c8f-4245-a045-
5050 688225f8c2ad</cmis:value>
5051         </cmis:propertyId>
5052       </cmis:properties>
5053     </cmisra:object>
5054     <cmisra:pathSegment>invoice3.pdf</cmisra:pathSegment>
5055   </atom:entry>
5056 </atom:feed>

```

5057

5058Note: This media type is used on links with relation down (see section 3.4.3.2 Hierarchy Navigation  
5059Internet Draft Link Relations). When the individual resources are returned by the CMIS repository they  
5060will use the atom media type (application/atom+xml)

5061

5062Please also see the example documents included with the schema.

5063

### 5064**3.3.5 CMIS ACL**

5065Media Type: application/cmisacl+xml

5066Starting tag: cmis:acl

5067

5068This document specifies an Access Control List based on the schema in CMIS Domain Model.

5069

5070Example:

```
5071 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5072 <cmis:acl xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
5073 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
5074 xmlns:atom="http://www.w3.org/2005/Atom"
5075 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5076 open.org/ns/cmis/restatom/200908/">
5077   <cmis:permission>
5078     <cmis:principal>
5079       <cmis:principalId>Al Brown</cmis:principalId>
5080     </cmis:principal>
5081     <cmis:permission>cmis:read</cmis:permission>
5082     <cmis:permission>cmis:write</cmis:permission>
5083     <cmis:permission>cmis:all</cmis:permission>
5084     <cmis:permission>publish</cmis:permission>
5085     <cmis:direct>true</cmis:direct>
5086   </cmis:permission>
5087 </cmis:acl>
```

5088

5089Please also see the example documents included with the schema.

5090

## 5091**3.4 Atom Extensions for CMIS**

### 5092**3.4.1 Atom Element Extensions**

#### 5093**3.4.1.1 AtomPub Workspace**

##### 5094**3.4.1.1.1 cmisra:collectionType**

5095This element is included inside the app:collection element. This specifies the cmis collection type.

##### 5096**3.4.1.1.2 cmisra:repositoryInfo**

5097This element is included inside the app:workspace element. This specifies information about the CMIS  
5098repository.

##### 5099**3.4.1.1.3 cmis:uritemplate**

5100This element is included inside the app:workspace element. This specifies information about URI  
5101templates

## 5102**3.4.1.2 Atom Feed**

### 5103**3.4.1.2.1 cmisra:numItems**

5104This element is included inside the atom:feed element. This specifies the number of items in the feed.

## 5105**3.4.1.3 Atom Entry**

### 5106**3.4.1.3.1 cmisra:children**

5107This element is included inside the atom:entry element. This includes the children of the atom entry.

5108

### 5109**3.4.1.3.2 cmisra:object**

5110This element is included inside the atom:entry element for CMIS Document, Folder, Relationship and  
5111Policy objects. This specifies the CMIS object information for the atom entry.

5112

### 5113**3.4.1.3.3 cmisra:pathSegment**

5114This element is included inside the atom:entry element for CMIS Type Definitions that are filable. This  
5115specifies the pathSegment for this object in the folder representing the feed.

5116

### 5117**3.4.1.3.4 cmisra:relativePathSegment**

5118This element is included inside the atom:entry element. This specifies the relative pathSegment for the  
5119object in that particular folder. This MUST be used only inside an object parents feed.

### 5120**3.4.1.3.5 cmisra:type**

5121This element is included inside the atom:entry element for CMIS Type Definitions. This specifies the type  
5122definition the atom entry represents.

### 5123**3.4.1.3.6 cmisra:content**

5124This element specifies the content of the atom:entry element. The content is base64 encoded.

5125This element MUST take precedence over atom:content on submission of an atom entry to a repository.

5126A repository MUST use the atom:content element to return back to the client the content of the document.

5127This is required when the client has an XML document stored that is not well formed and thus would not  
5128be able to be included inside atom:content element.

## 5129**3.4.2 Attributes**

5130These attributes are in the CMIS RestAtom namespace (cmisra).

### 5131**3.4.2.1 cmisra:id**

5132This attribute is used on the atom:link element to specify the cmis id of the resource. This attribute  
5133SHOULD be on all link relations that point to a CMIS object.

5134

5135This attribute MAY also be on cmisra:type. The value of the attribute on cmis:type MUST be the same as  
5136the type definition id.

5137

5138Example:

```
5139 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5140 <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
5141 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
5142 xmlns:atom="http://www.w3.org/2005/Atom"
5143 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5144 open.org/ns/cmis/restatom/200908/" type="application/atom+xml;type=feed"
5145 rel="down" href="http://cmisexample.oasis-open.org/repl1/children/ba2cb793-
5146 6d6d-48cd-b6c2-62dcddedc4b5/1" cmisra:id="ba2cb793-6d6d-48cd-b6c2-
5147 62dcddedc4b5"/>
```

5148

5149Please also see the example documents included with the schema.

5150

### 51513.4.2.2 cmisra:renditionKind

5152This attribute is used on the atom:link element with relation alternate to specify the renditionKind of the  
5153resource. This attribute SHOULD be on all link elements with relation alternate that are a CMIS rendition.

5154

5155Example:

```
5156 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5157 <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
5158 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
5159 xmlns:atom="http://www.w3.org/2005/Atom"
5160 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5161 open.org/ns/cmis/restatom/200908/" type="text/html" rel="alternate"
5162 href="http://cmisexample.oasis-open.org/repl1/rendition/ba2cb793-6d6d-48cd-
5163 b6c2-62dcddedc4b5/1" cmisra:renditionKind="cmis:thumbnail"/>
```

5164

5165Please also see the example documents included with the schema.

### 51663.4.3 CMIS Link Relations

5167The listing below outlines the different link relation types in CMIS. This is in addition to the link relations  
5168specified by Atom and Atom Publishing Protocol. The registry for link relations is located at  
5169<http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

5170

5171The link element with a specified relation MUST be included if client can perform the operation. The  
5172repository SHOULD omit the link relation if the operation is not available. The operation may not be  
5173available due to a variety of reasons such as access control, administrative policies, or other  
5174mechanisms.

5175

5176Links may have the following attribute in addition to the ones specified by Atom and Atom Publishing  
5177Protocol:

- 5178 • (CMIS) id: Specifies the CMIS ID of the resource referenced by the link. Repositories SHOULD  
5179 include this attribute for elements such as atom:link that point to CMIS resources that have an id.

5180

5181These are the link relation types specified by CMIS:



### 51823.4.3.1 Existing Link Relations

5183 Existing link relations should be used where appropriate by the implementation. In addition, the following  
5184 link relations are leveraged for the CMIS specification:

- 5185     • self
  - 5186         ○ This link relation provides the URI to retrieve this resource again.
  - 5187         ○ Service: The appropriate service that generated the atom entry or feed.
  - 5188         ○ Resources: All except AllowableActions, ACL and Content Streams
- 5189     • service
  - 5190         ○ The service link relation when provided on a CMIS resource MUST point to an AtomPub  
5191             service document with only one workspace element. This workspace element MUST  
5192             represent the repository containing that resource.
  - 5193         ○ Media Type: application/atomsvc+xml
  - 5194         ○ Resources: All except AllowableActions, ACL and Content Streams
- 5195     • describedby
  - 5196         ○ When used on a CMIS resource, this link relation MUST point to an atom entry that  
5197             describes the type of that resource.
  - 5198         ○ Service: getTypeDefinition on specified object
  - 5199         ○ Media Type: application/atom+xml;type=entry
  - 5200         ○ Resources: CMIS Document, CMIS Folder, CMIS Relationship, CMIS Policy objects and  
5201             CMIS Types
- 5202     • via
  - 5203         ○ When used on an Atom Feed document, this link relation MUST point to the atom entry  
5204             representing the CMIS resource from whom this feed is derived.
  - 5205         ○ Media Type: application/atom+xml;type=entry
  - 5206         ○ Resources: All CMIS Feeds and Collections
- 5207     • edit-media
  - 5208         ○ When used on a CMIS document resource, this link relation MUST point to the URI for  
5209             content stream of the CMIS document. This URI MUST be used to set or delete the  
5210             content stream. This URI MAY be used to retrieve the content stream for the document.
  - 5211         ○ Service: setContentStream (PUT) , deleteContentStream (DELETE)
  - 5212         ○ Media Type: Specific to resource
  - 5213         ○ Resources: CMIS Document
- 5214     • edit
  - 5215         ○ When used on a CMIS resource, this link relation MUST provide an URI that can be used  
5216             with the HTTP PUT method to modify the atom:entry for the CMIS resource
  - 5217         ○ Service: getObject (GET), updateProperties (PUT)
  - 5218         ○ Media Type: application/atom+xml;type=entry
  - 5219         ○ Resources: CMIS Documents, CMIS Folders, CMIS Relationships and CMIS Policies
- 5220     • alternate
  - 5221         ○ This is used to express Renditions on a CMIS resource. See section 3.1.6 Renditions.
  - 5222         ○ Service: getContentStream for specified rendition
  - 5223         ○ Resources: CMIS Document, CMIS Folder and CMIS Policies

- 5224 • first
  - 5225 ○ This is used for Paging. Please see the AtomPub specification.
  - 5226 ○ Media Type: application/atom+xml;type=feed
  - 5227 ○ Resources: All Feeds
- 5228 • previous
  - 5229 ○ This is used for Paging. Please see the AtomPub specification.
  - 5230 ○ Media Type: application/atom+xml;type=feed
  - 5231 ○ Resources: All Feeds
- 5232 • next
  - 5233 ○ This is used for Paging. Please see the AtomPub specification.
  - 5234 ○ Media Type: application/atom+xml;type=feed
  - 5235 ○ Resources: All Feeds
- 5236 • last
  - 5237 ○ This is used for Paging. Please see the AtomPub specification.
  - 5238 ○ Media Type: application/atom+xml;type=feed
  - 5239 ○ Resources: All Feeds

5240

5241

5242 Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on  
5243 these link relations.

### 5244 3.4.3.2 Hierarchy Navigation Internet Draft Link Relations

5245 Please refer to the Internet Draft here: <http://www.ietf.org/internet-drafts/draft-divilly-atom-hierarchy-03.txt>.

5247

5248 CMIS leverages the following link relations from the Internet Draft:

- 5249 • up
  - 5250 ○ Service: getFolderParent, getObjectParents, getTypeDefinition, getObject
  - 5251 ○ Media Type: application/atom+xml;type=feed, application/atom+xml;type=entry
  - 5252 ○ Resources: CMIS Document, CMIS Folder, CMIS Type Definitions, CMIS Folder Children, CMIS Folder Descendants, CMIS FolderTree, CMIS Type Children, CMIS Type Descendants
    - 5253
    - 5254
    - 5255 ■ This link relation is not included on CMIS Base Type Definitions or the CMIS Root Folder
    - 5256
- 5257 • down
  - 5258 ○ Service: getChildren, getDescendants, getTypeChildren, getTypeDescendants
  - 5259 ○ Media Type:
    - 5260 ■ For children: application/atom+xml;type=feed
    - 5261 ■ For descendants: application/cmistree+xml
    - 5262 • The descendants feed resource when retrieved from the CMIS repository will use the Atom Feed Media Type (application/atom+xml;type=feed)
    - 5263
  - 5264 ○ Resources: CMIS Folder, Type

### 52653.4.3.3 Versioning Internet Draft Link Relations

5266Please refer to the Internet Draft here: <http://www.ietf.org/internet-drafts/draft-brown-versioning-link-5267relations-01.txt>.

5268

5269CMIS leverages the following link relations from the Internet Draft:

- 5270 • version-history
  - 5271 ○ Service: getAllVersions
  - 5272 ○ Media Type: application/atom+xml;type=feed
  - 5273 ○ Resources: CMIS Document
- 5274 • current-version
  - 5275 ○ Service: getObjectForLatestVersion
  - 5276 ○ Media Type: application/atom+xml;type=entry
  - 5277 ○ Resources: CMIS Document
- 5278 • working-copy
  - 5279 ○ Service: getObject for private-working-copy specified by  
5280 cmis:versionSeriesCheckedOutId property
  - 5281 ○ Media Type: application/atom+xml;type=entry
  - 5282 ○ Resources: CMIS Document

### 52833.4.3.4 CMIS Specific Link Relations

5284 CMIS defines the following link relations:

- 5285 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>
  - 5286 ■ This link relation MUST point to a resource containing a CMIS AllowableActions
  - 5287 document for the CMIS resource containing this link relation.
  - 5288 ■ Service: getAllowableActions
  - 5289 ■ Media Type: application/cmisallowableactions+xml
  - 5290 ■ Resources: CMIS Documents, CMIS Folders, CMIS Policies, and CMIS
  - 5291 Relationships
- 5292 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>
  - 5293 ■ This link relation MUST point to a resource containing an Atom Feed of CMIS
  - 5294 relationship resources for the CMIS resource containing this link relation.
  - 5295 ■ Service: getObjectRelationships
  - 5296 ■ Media Type: application/atom+xml;type=feed
  - 5297 ■ Resources: CMIS Documents, CMIS Folders, and CMIS Policies
- 5298 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/source>
  - 5299 ■ When used on a CMIS Relationship resource, this link relation MUST point to an
  - 5300 atom entry document for the CMIS Resource specified by the cmis:sourceld
  - 5301 property on the relationship.
  - 5302 ■ Source Link on Relationship
  - 5303 ■ Media Type: application/atom+xml;type=entry
  - 5304 ■ Resources: CMIS Relationships
- 5305 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/target>

- 5306                   ▪ When used on a CMIS Relationship resource, this link relation MUST point to an  
5307                   atom entry document for the CMIS Resource specified by the cmis:targetId  
5308                   property on the relationship.
- 5309                   ▪ Target Link on Relationship
- 5310                   ▪ Media Type: application/atom+xml;type=entry
- 5311                   ▪ Resources: CMIS Relationships
- 5312                   ○ <http://docs.oasis-open.org/ns/cmis/link/200908/policies>
- 5313                   ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS  
5314                   Policy resources for the CMIS resource containing this link relation.
- 5315                   ▪ Service: getAppliedPolicies
- 5316                   ▪ Media Type: application/atom+xml;type=feed
- 5317                   ▪ Resources: CMIS Documents and CMIS Folders
- 5318                   ○ <http://docs.oasis-open.org/ns/cmis/link/200908/acl>
- 5319                   ▪ This link relation MUST point to a resource containing a CMIS ACL document for  
5320                   the CMIS resource containing this link relation.
- 5321                   ▪ Service: getACL
- 5322                   ▪ Media Type: application/cmisacl+xml
- 5323                   ▪ Resources: CMIS Documents, CMIS Folders, CMIS Relationships, and CMIS  
5324                   Policies that are securable
- 5325                   ○ <http://docs.oasis-open.org/ns/cmis/link/200908/changes>
- 5326                   ▪ This link relation MUST point to an Atom Feed containing the set of changes
- 5327                   ▪ Service: getContentChanges
- 5328                   ▪ Media Type: application/atom+xml;type=feed
- 5329                   ▪ Resources: AtomPub Workspace Element in Service Document
- 5330                   ○ <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>
- 5331                   ▪ Used in AtomPub Service Document to identify the folder tree for a specified  
5332                   folder
- 5333                   ▪ Service: getFolderTree
- 5334                   ▪ Media Type: application/atom+xml;type=feed
- 5335                   ▪ Resources: CMIS Folder, also used in AtomPub Service Document for root folder
- 5336                   ○ <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>
- 5337                   ▪ Used in AtomPub Service Document to identify the base types descendants
- 5338                   ▪ Service: getTypeDescendants
- 5339                   ▪ Media Type: application/atom+xml;type=feed
- 5340                   ▪ Resources: AtomPub Workspace Element in Service Document
- 5341                   ○ <http://docs.oasis-open.org/ns/cmis/link/200908/rootdescendants>
- 5342                   ▪ Used in AtomPub Service Document to identify the root folder descendants
- 5343                   ▪ Service: getDescendants for root folder
- 5344                   ▪ Media Type: application/atom+xml;type=feed
- 5345                   ▪ Resources: AtomPub Workspace Element in Service Document
- 5346

## 5347**3.5 Atom Resources**

5348For all Atom Resources used in this specification, the following MUST be followed:

### 5349**3.5.1 Feeds**

5350Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis objects:

- 5351 • atom:updated SHOULD be the latest time the folder or its contents was updated. If unknown by  
5352 the underlying repository, it MUST be the current time.
- 5353 • atom:author/atom:name MUST be the CMIS property cmis:createdBy
- 5354 • atom:title MUST be the CMIS property cmis:name
- 5355 • The atom:link with relation self MUST be generated to return the URI of the feed. If paging or any  
5356 other mechanism is used to filter, sort, or change the representation of the feed, the URI MUST  
5357 point back a resource with the same representation.
- 5358 • A feed SHOULD contain the element app:collection, describing the appropriate media types  
5359 supported for creation of new entries in the feed
- 5360 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's  
5361 specification and be a valid URI.
- 5362 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than  
5363 contained in the feed, then a link with the relation next MUST be included in the feed.

5364

5365Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis types:

- 5366 • atom:updated SHOULD be the latest time type definition was updated. If unknown by the  
5367 underlying repository, it MUST be the current time.
- 5368 • atom:author/atom:name is repository specific
- 5369 • atom:title MUST be the displayName attribute of the CMIS Type Definition.
- 5370 • The atom:link with relation self MUST be generated to return the URI of the feed
- 5371 • atom:id SHOULD be derived from the id attribute of the CMIS Type Definition. This id MUST be  
5372 compliant with atom's specification and be a valid URI.
- 5373 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than  
5374 contained in the feed, then a link with the relation next MUST be included in the feed.

5375

5376If on the root type, all fields are repository specific.

5377

5378Ordering of entries in a feed is repository-specific if orderBy argument is not specified. If orderBy  
5379argument is specified, the order of the entries in the feed SHOULD conform to the ordering specified by  
5380the orderBy argument.

5381

5382Note: Please see [feedvalidator.org](http://feedvalidator.org) to validate Atom compliance.

### 5383**3.5.2 Entries**

5384At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry  
5385document and conform to the guidelines below for a cmis object:

- 5386 • atom:title MUST be the cmis:name property
- 5387 • app:edited MUST be cmis:lastModifiedDate
- 5388 • atom:updated MUST be cmis:lastModifiedDate

- 5389 • atom:published MUST be cmis:createdDate
- 5390 • atom:author/atom:name MUST be cmis:createdBy
- 5391 • All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are  
5392 duplicated in an atom element
- 5393 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's  
5394 specification and be a valid URI.
- 5395 • The repository SHOULD populate the atom:summary tag with text that best represents a  
5396 summary of the object. For example, an HTML table containing the properties and their values or  
5397 the description of the document if available.

5398

5399 For Documents that support Content Streams:

- 5400 The repository SHOULD use the atom:content/src attribute to point to the content stream.
- 5401 The client SHOULD use cmisra:content if the content is not well-formed or would have
- 5402 trouble fitting inside an atom:content element. The repository MUST use the
- 5403 cmisra:content element if provided by the client over the atom:content element.

5404

5405 Other Objects (Folders, Relationships, and other Document Types that do not support Content  
5406 Streams, etc):

- 5407 The repository MUST comply with the atom specification and have an atom:content
- 5408 element. This is repository specific. Any value in the content field MUST be ignored if the
- 5409 atom entry represents a non-document object by the CMIS repository when the atom
- 5410 entry is POST'ed to a collection or sent to the repository via a PUT.

5411

5412 When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding  
5413 writable CMIS property. For example, atom:title will overwrite cmis:name.

5414

5415 At any point where an Atom document of CMIS Type is sent or returned, it must be a valid Atom Entry  
5416 document and conform to the guidelines below for a cmis type definition:

- 5417 • atom:title MUST be the cmis:displayName
- 5418 • The repository SHOULD populate the atom:summary tag with text that best represents a  
5419 summary of the object. For example, the type description if available.
- 5420 • The repository MUST comply with the atom specification and have an atom:content element. This  
5421 is repository specific. Any value in the content field MUST be ignored if the atom entry represents  
5422 a non-document object by the CMIS repository when the atom entry is POST'ed to a collection or  
5423 sent to the repository via a PUT.

5424

5425

5426 Any atom element that is not specified is repository-specific.

### 5427 3.5.2.1 Hierarchical Atom Entries

5428 The repository SHOULD NOT provide any links to hierarchical objects if those capabilities are not  
5429 supported with the exception of getTypeDescendants which is required

5430

5431 For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate  
5432a cmisra:children element in the atom:entry with the enclosing feed of its direct children. This pattern  
5433 continues until the depth is satisfied.

5434

5435The cmisra:children element that MUST be included in an atom entry:

5436<xs:element name="children" type="atom:feedType" />

5437

5438If an entry does not contain cmisra:children element, then the entry MAY have children even though it is

5439not represented in the atom entry.

5440

5441For Example, here is a minimal Atom Entry with CMIS Children Extension Element:

```
5442 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5443 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
5444 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
5445 xmlns:atom="http://www.w3.org/2005/Atom"
5446 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5447 open.org/ns/cmis/restatom/200908/">
5448   <atom:author>
5449     <atom:name>Al Brown</atom:name>
5450   </atom:author>
5451   <atom:content src="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-
5452 4233-85d3-39cad56fa091"/>
5453   <atom:id>urn:uuid:78a66015-ffd3-4233-85d3-39cad56fa091</atom:id>
5454   <atom:title type="text">CMIS Example Folder as Customer type</atom:title>
5455   <atom:updated>2009-10-19T10:09:58.906-07:00</atom:updated>
5456   <cmisra:object>
5457     <cmis:properties>
5458       <cmis:propertyId localName="rep-cmis:objectId"
5459 propertyDefinitionId="cmis:objectId">
5460         <cmis:value>78a66015-ffd3-4233-85d3-39cad56fa091</cmis:value>
5461       </cmis:propertyId>
5462     </cmis:properties>
5463   </cmisra:object>
5464   <cmisra:pathSegment>customer</cmisra:pathSegment>
5465   <cmisra:children>
5466     <atom:title type="text">CMIS Example Folder as Customer
5467 type</atom:title>
5468     <atom:author>
5469       <atom:name>Al Brown</atom:name>
5470       <atom:uri>http://www.ibm.com/</atom:uri>
5471       <atom:email>albertcbrown@us.ibm.com</atom:email>
5472     </atom:author>
5473     <atom:updated>2009-10-19T10:09:58.906-07:00</atom:updated>
5474     <atom:id>urn:uuid:d05a1c90-bb02-42e2-baba-f51366e701c1</atom:id>
5475     <atom:link type="application/atom+xml;type=feed" rel="self"
5476 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5477 39cad56fa091/3"/>
5478     <atom:link type="application/atomsvc+xml" rel="service"
5479 href="http://cmisexample.oasis-open.org/rep1//service"/>
5480     <atom:link type="application/atom+xml;type=entry" rel="via"
5481 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5482 39cad56fa091"/>
5483     <atom:link type="application/atom+xml;type=feed"
5484 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
5485 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5486 39cad56fa091/foldertree"/>
5487     <atom:link type="application/atom+xml;type=feed" rel="down"
5488 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5489 39cad56fa091/children"/>
5490     <atom:link type="application/atom+xml;type=entry" rel="up"
5491 href="http://cmisexample.oasis-open.org/rep1/c84ae842-15a2-4571-93fe-
5492 65cca95faa24"/>
5493   </cmisra:children>
5494   </atom:entry>
5495   <atom:author>
```



```

5495         <atom:name>Al Brown</atom:name>
5496         </atom:author>
5497         <atom:content src="http://cmisexample.oasis-
5498 open.org/rep1/78a66015-ffd3-4233-85d3-39cad56fa091"/>
5499         <atom:id>urn:uuid:78a66015-ffd3-4233-85d3-39cad56fa091</atom:id>
5500         <atom:title type="text">CMIS Example Child of Folder</atom:title>
5501         <atom:updated>2009-10-19T10:09:58.906-07:00</atom:updated>
5502         <cmisra:object>
5503             <cmis:properties>
5504                 <cmis:propertyId localName="rep-cmis:objectId"
5505 propertyDefinitionId="cmis:objectId">
5506                     <cmis:value>78a66015-ffd3-4233-85d3-
5507 39cad56fa091</cmis:value>
5508                 </cmis:propertyId>
5509             </cmis:properties>
5510         </cmisra:object>
5511         <cmisra:pathSegment>document</cmisra:pathSegment>
5512     </atom:entry>
5513 </cmisra:children>
5514 </atom:entry>
5515

```

5516 Please also see the example documents included with the schema.

### 5517 3.6 AtomPub Service Document (Repository)

5518 The AtomPub Service Document contains the set of repositories that are available. Each repository is  
5519 mapped to a `app:workspace` element in the AtomPub Service document.

5520

5521 CMIS Services exposed:

5522 GET: `getRepositories`, `getRepositoryInfo`

5523

5524 Media Type: `application/atomsvc+xml`

5525

5526 How the client will get the initial AtomPub (APP) service document or the URI for the service document is  
5527 repository specific. Examples are via URI, or loading the service document from disk.

5528

5529 The service document will be available from Atom Entry and Atom Feed documents via a link relationship,  
5530 service. That AtomPub service document MUST contain only one workspace element which MUST be  
5531 the workspace representing the repository containing the Atom Entry or Atom Feed document.

5532

5533 A workspace element for a CMIS repository MUST have a collection element for each of following  
5534 collections: Each collection MUST also contain a `cmisra:collectionType` attribute with the given value:

- 5535 • Root Folder Collection: Root folder of the Repository
  - 5536 ○ 'root' for the children collection of the root folder
  - 5537 ○ `cmisra:collectiontype='root'`
- 5538 • Types Collection: Collection containing all the types in the repository
  - 5539 ○ 'types' for the children collection
  - 5540 ○ `cmisra:collectiontype='types'`

5541

5542 The workspace element SHOULD contain these collections if the repository supports this functionality:

- 5543 • CheckedOut collection: collection containing all checked out documents user can see



- 5544           ○ 'checkedout'
- 5545           ○ cmisra:collectiontype='checkedout'
- 5546       • Query collection: Collection for posting queries to be executed
- 5547           ○ 'query'
- 5548           ○ cmisra:collectiontype='query'
- 5549       • Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
- 5550           ○ 'unfiled'
- 5551           ○ cmisra:collectiontype='unfiled'

5552

5553The repository MUST include the URI templates in the workspace elements.

5554

5555The workspace element MUST also contain the following link element with the relation:

- 5556       • <http://docs.oasis-open.org/ns/cmis/link/200908/typesdescendants>: This link relation points to the
- 5557           types descendants for the base types in the repository.

5558

5559The workspace element MUST contain the following link relations for those services which are supported  
5560by the repository:

- 5561       • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: This link relation points to the folder
- 5562           tree of the root folder. See Folder Tree resource for more information.
- 5563       • <http://docs.oasis-open.org/ns/cmis/link/200908/rootdescendants>: This link relation points to the
- 5564           descendants feed for the root folder.
- 5565       • <http://docs.oasis-open.org/ns/cmis/link/200908/changes>: This link relation points to the changes
- 5566           feed for the repository.

5568The workspace element may include app:collection element for the collections that represent folders in  
5569the repository. However, an alternative approach, especially for a repository with many folders, is to not  
5570enumerate those collections here, but include the app:collection element per RFC5023 in the Atom Feed  
5571document.

## 55723.6.1 URI Templates

5573

5574CMIS defines the following URI Templates:

- 5575       • objectbyid
- 5576       • objectbypath
- 5577       • query
- 5578       • typebyid

5579

5580Repositories MUST provide the following URI Templates:

- 5581       • objectbyid
- 5582       • objectbypath
- 5583       • typebyid

5584

5585Repositories MUST provide the URI Template query if the repository supports query.

5586

5587Repositories MAY extend that set of templates. Those URI Template Types will be repository specific.  
5588Repositories MAY have more than one entry per URI Template type if the entries have different media  
5589types.

5590

5591URI Templates are simple replacement of the template parameter with the specified value. If a client  
5592does not want to specify a value for some of these variables, then the client MUST substitute an empty  
5593string for the variable.

5594

5595For example, if the URI template that supports the variable {id} is

```
5596 http://example.org/repl/getbyid/{id}
```

5597

5598If the client wants to find the entry for an object with an id of 'obj\_1' then the URI would be:

```
5599 http://example.org/repl/getbyid/obj_1
```

5600

5601Arguments that are substituted for URI template parameters MUST be percent escaped according to  
5602RFC3986. Please see that RFC for more information.

5603

5604All variables MUST be in the template.

5605

5606Structure of URI Template:

```
5607 <xs:complexType name="cmisUriTemplateType">
5608   <xs:sequence>
5609     <xs:element name="template" type="xs:string" />
5610     <xs:element name="type" type="xs:string" />
5611     <xs:element name="mediatype" type="xs:string" />
5612     <xs:any processContents="lax" namespace="##other"
5613 minOccurs="0"
5614           maxOccurs="unbounded" />
5615   </xs:sequence>
5616 </xs:complexType>
```

5617

5618

5619Example of URI Template element in an AtomPub Workspace Element:

```
5620 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5621 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5622 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5623 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5624 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5625 open.org/ns/cmis/restatom/200908/">
5626   <cmisra:template>http://cmisexample.oasis-open.org/repl/objectbyid/{id}?
5627 filter={filter}&includeAllowableActions={includeAllowableActions}&incl
5628 udePolicyIds={includePolicyIds}&includeRelationships={includeRelationships
5629 }&includeACL={includeACL}</cmisra:template>
5630   <cmisra:type>objectbyid</cmisra:type>
5631   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5632 </cmisra:uritemplate>
```

5633

5634Please also see the example documents included with the schema.

5635

### 56363.6.1.1 Object By Id

5637This URI template provides a method for creating an URI that directly accesses an atom entry  
5638representing documents, folders, policies or relationship objects. See section 3.10 for more information.

5639

5640Type: objectbyid

5641Media Type: application/atom+xml;type=entry

5642

5643Service: getObjectById

5644

5645Variables that are supported by the template:

- 5646 • {id}: Id of object
- 5647 • {filter}: Property Filter
- 5648 • {includeAllowableActions}
  - 5649 ○ Valid values: true, false
- 5650 • {includePolicyIds}: Include Policy Ids:
  - 5651 ○ Valid values: true, false
- 5652 • {includeRelationships}: Include relationships
  - 5653 ○ Valid values: See enumIncludeRelationships
- 5654 • {includeACL}: Include ACLs
  - 5655 ○ Valid values: true, false
- 5656 • {renditionFilter}
  - 5657 ○ Valid values: Please see renditionFilter in CMIS Domain Model

5658

5659Example:

```
5660 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5661 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5662 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5663 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5664 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5665 open.org/ns/cmis/restatom/200908/">
5666   <cmisra:template>http://cmisexample.oasis-open.org/repl/objectbyid/{id}?
5667   filter={filter}&includeAllowableActions={includeAllowableActions}&incl
5668   udePolicyIds={includePolicyIds}&includeRelationships={includeRelationships
5669   }&includeACL={includeACL}</cmisra:template>
5670   <cmisra:type>objectbyid</cmisra:type>
5671   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5672 </cmisra:uritemplate>
```

5673

5674

5675Please also see the example documents included with the schema.

### 56763.6.1.2 Object By Path

5677This URI template provides a method for creating an URI that directly accesses an atom entry  
5678representing documents, folders or policy objects. See section 3.10 for more information.

5679

5680Type: objectbypath

5681Media Type: application/atom+xml;type=entry

5682

5683Service: getObjectByPath

5684

5685Variables that are supported by the template:

- 5686 • {path}: Path of Object
- 5687 • {filter}: Property Filter
- 5688 • {includeAllowableActions}: Boolean for include Allowable Actions
  - 5689 ○ Valid values: true, false
- 5690 • {includePolicyIds}: Include Policy Ids:
  - 5691 ○ Valid values: true, false
- 5692 • {includeRelationships}: Include relationships
  - 5693 ○ Valid values: See enumIncludeRelationships
- 5694 • {includeACL}: Include ACLs
  - 5695 ○ Valid values: true, false
- 5696 • {renditionFilter}
  - 5697 ○ Valid values: Please see renditionFilter in CMIS Domain Model

5698

5699

5700Example:

```
5701 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5702 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5703 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5704 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5705 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5706 open.org/ns/cmis/restatom/200908/">
5707   <cmisra:template>http://cmisexample.oasis-open.org/repl/objectbypath?
5708 p={path}&amp;filter={filter}&amp;includeAllowableActions={includeAllowableActi
5709 ons}&amp;includePolicyIds={includePolicyIds}&amp;includeRelationships={include
5710 Relationships}&amp;includeACL={includeACL}</cmisra:template>
5711   <cmisra:type>objectbypath</cmisra:type>
5712   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5713 </cmisra:uritemplate>
```

5714

5715

5716Please also see the example documents included with the schema.

5717

### 57183.6.1.3 Query

5719Type: query

5720Media Type: application/atom+xml;type=feed

5721

5722Service: query

5723

5724Variables that are supported by the template:

- 5725 • {q}: CMIS Query Statement

- 5726 • {searchAllVersions}: Boolean, true if to search all versions
- 5727 • {maxItems}: Integer, Max items to return
- 5728 • {skipCount}: Integer, Items to skip
- 5729 • {includeAllowableActions}: Boolean
- 5730 • {includeRelationships}: Boolean

5731

5732

5733Example:

```
5734 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5735 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5736 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5737 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5738 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5739 open.org/ns/cmis/restatom/200908/">
5740   <cmisra:template>http://cmisexample.oasis-open.org/repl/query?
5741 q={q}&searchAllVersions={searchAllVersions}&maxItems={maxItems}&sk
5742 ipCount={skipCount}&includeAllowableActions={includeAllowableActions}=&
5743 ;includeRelationships={includeRelationships}</cmisra:template>
5744   <cmisra:type>query</cmisra:type>
5745   <cmisra:mediatype>application/atom+xml;type=feed</cmisra:mediatype>
5746 </cmisra:uritemplate>
```

5747

5748

5749Please also see the example documents included with the schema.

5750

### 57513.6.1.4 Type By Id

5752Type: typebyid

5753Media Type: application/atom+xml;type=entry

5754

5755Service: getTypeDefinition

5756

5757Variables that are supported by the template:

- 5758 • {id}: CMIS Type Id

5759

5760

5761Example:

```
5762 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5763 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5764 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5765 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5766 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5767 open.org/ns/cmis/restatom/200908/">
5768   <cmisra:template>http://cmisexample.oasis-open.org/repl/type?
5769 id={id}</cmisra:template>
5770   <cmisra:type>query</cmisra:type>
5771   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5772 </cmisra:uritemplate>
```

5773

5774Please also see the example documents included with the schema.

5775

5776

## 5777**3.6.2 HTTP Methods**

### 5778**3.6.2.1 GET**

5779This retrieves the AtomPub Service document for a specified repository. This exposes the capabilities  
5780defined in `getRepositories` and `getRepositoryInfo` in the Domain Model.

5781

5782The optional argument MAY be specified:

5783     • `repositoryId`:

5784         ○ This query parameter allows a client to specify a different repository than the one that is  
5785             referenced by the URI.

5786         ○ If specified, the repository MUST return the AtomPub services document for the specified  
5787             repository if that repository exists.

5788         ○ If not specified, the repository MUST return the service document for the repository that is  
5789             referenced by URI.

5790

## 5791**3.7 Service Collections**

5792These are the collections that are included on an AtomPub Service document in the workspace element.

### 5793**3.7.1 Root Folder Collection**

5794This is a collection described in the service document. Please see [Folder Children](#).

### 5795**3.7.2 Query Collection**

5796This is a collection for processing queries. If the implementation supports GET on this collection, then the  
5797implementation SHOULD at least return a feed consisting of zero or more atom entries. These atom  
5798entries should represent persisted objects related to query such as persisted queries, long running  
5799queries or search templates.

5800

5801CMIS Services exposed via HTTP verbs:

5802     POST: Query

5803

5804Media Type: `application/atom+xml;type=feed`

5805Accept:

5806     • MUST support CMIS Query document,

5807     • MAY support other media type

5808

5809Link Relations on resulting feed from Query Collection:

5810     • `service`: Points to service document containing the CMIS repository. The service document  
5811         MUST contain only one workspace element.

5812         ○ Media Type: `application/atomsvc+xml`

5813     • paging link relations as appropriate: first, next, previous, last

5814

5815The following CMIS Atom extension element MAY be included inside the atom feed:

- 5816
  - cmisra:numItems

5817

5818The following CMIS Atom extension element MUST be included inside the atom entries:

- 5819
  - cmisra:object inside atom:entry

5820

### 5821**3.7.2.1 POST**

5822This collection MUST accept CMIS Query documents (application/cmisquery+xml).

5823

5824Upon submission (creation) of a query document, a response must be returned with a Location header  
5825representing the feed for that query. If the query cannot be performed and an atom feed returned, the  
5826repository MUST return the appropriate HTTP status code. In addition, the server SHOULD return the  
5827feed directly. If the server does so, the server should also return the Content-Location header.

5828

5829The feed returned MUST contain a set of atom entries representing the result set from the query.

5830

5831The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The  
5832atom entries MUST contain the CMIS extension element (cmis:object) containing the properties specified  
5833by the query in the select clause of the query statement.

5834

5835If all the selected properties can be mapped to the same type reference, then the repository MAY include  
5836additional information in the atom entry.

5837

5838Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

5839

5840Status Codes:

- 5841
  - 201 Success

5842

5843Headers returned:

- 5844
  - Location Header

5845     
  - Content-Location Header

5846

5847Link Relations on resulting feed from POST to Query Collection:

- 5848
  - service: Points to service document containing the CMIS repository. The service document  
5849         MUST contain only one workspace element.
  - Media Type: application/atomsvc+xml

5850     
  - paging link relations as appropriate: first, next, previous, last

5851

5852Example client request:

```
5854 POST /Query HTTP/1.1
5855 Host: example.org
5856 Content-Length: 756
```

```

5857 Content-Type: application/cmisquery+xml
5858
5859
5860 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5861 <cmis:query xmlns:app="http://www.w3.org/2007/app"
5862 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5863 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5864 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5865 open.org/ns/cmis/restatom/200908/">
5866   <cmis:statement>SELECT cmis:objectId FROM cmis:document</cmis:statement>
5867   <cmis:searchAllVersions>true</cmis:searchAllVersions>
5868   <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
5869   <cmis:includeRelationships>none</cmis:includeRelationships>
5870   <cmis:renditionFilter>*</cmis:renditionFilter>
5871   <cmis:maxItems>50</cmis:maxItems>
5872   <cmis:skipCount>0</cmis:skipCount>
5873 </cmis:query>
5874
5875
5876

```

#### 5877Example server response:

```

5878 HTTP/1.1 201 Created
5879 Date: Mon, 19 Oct 2009 10:10:01 -0700
5880 Content-Length: 1830
5881 Content-Type: application/atom+xml;type=feed
5882 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/6e23b7c6-
5883 e377-4a6b-b21c-7450f28b4b5d
5884 Location: http://cmisexample.oasis-open.org/rep1/queryresult/6e23b7c6-e377-
5885 4a6b-b21c-7450f28b4b5d
5886
5887
5888 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5889 <atom:feed xmlns:app="http://www.w3.org/2007/app"
5890 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5891 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5892 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5893 open.org/ns/cmis/restatom/200908/">
5894   <atom:title type="text">CMIS Query Result for SELECT cmis:objectId FROM
5895 cmis:document</atom:title>
5896   <atom:author>
5897     <atom:name>Al Brown</atom:name>
5898     <atom:uri>http://www.ibm.com/</atom:uri>
5899     <atom:email>albertcbrown@us.ibm.com</atom:email>
5900   </atom:author>
5901   <atom:updated>2009-10-19T10:10:01.078-07:00</atom:updated>
5902   <atom:id>urn:uuid:7fd19974-9597-4eb9-88ca-3a2297cd893c</atom:id>
5903   <atom:link type="application/atom+xml;type=feed" rel="self"
5904 href="http://cmisexample.oasis-open.org/rep1/c7f915fb-c222-45c9-8fa2-
5905 5062102c62b9/3"/>
5906   <atom:link type="application/atomsvc+xml" rel="service"
5907 href="http://cmisexample.oasis-open.org/rep1//service"/>
5908   <atom:entry>
5909     <atom:author>
5910       <atom:name>Al Brown</atom:name>
5911     </atom:author>
5912     <atom:content src="http://cmisexample.oasis-open.org/rep1/b4519062-
5913 6b13-4160-8abe-1c43bfbfe32e"/>
5914     <atom:id>urn:uuid:b4519062-6b13-4160-8abe-1c43bfbfe32e</atom:id>
5915     <atom:title type="text">Resulting Document</atom:title>
5916     <atom:updated>2009-10-19T10:10:01.078-07:00</atom:updated>
5917     <cmisra:object>
5918       <cmis:properties>

```



```

5919         <cmis:propertyId queryName="cmis:objectId" localName="rep-
5920 cmis:objectId" propertyDefinitionId="cmis:objectId">
5921             <cmis:value>b4519062-6b13-4160-8abe-
5922 1c43bfbfe32e</cmis:value>
5923         </cmis:propertyId>
5924     </cmis:properties>
5925 </cmisra:object>
5926 </atom:entry>
5927 </atom:feed>

```

5928

5929

5930 Please also see the example documents included with the schema.

5931

### 5932 **3.7.3 Checked Out Collection**

5933 This is a collection described in the service document that contains all the checkedout documents

5934 CMIS Services:

5935 GET: getCheckedOutDocs

5936 POST: checkOut

5937 Media Type: application/atom+xml;type=feed

5938 Accept:

5939 • MUST support Atom Entry Documents with CMIS extensions

5940     ○ application/atom+xml;type=entry or

5941     ○ application/cmisaatom+xml

5942 • MAY support other media type

5943

5944 Link Relations:

5945 • service: Points to service document containing the CMIS repository. The service document  
5946 MUST contain only one workspace element.

5947     ○ Media Type: application/atomsvc+xml

5948 • paging link relations as appropriate: first, next, previous, last

5949

5950 The following CMIS Atom extension element MAY be included inside the atom feed:

5951 • cmisra:numItems

5952

5953 The following CMIS Atom extension element MUST be included inside the atom entries:

5954 • cmisra:object inside atom:entry

5955

#### 5956 **3.7.3.1 GET**

5957 The following arguments may be supplied. Please see the domain model for more information:

5958 • filter

5959 • folderId

5960 • maxItems

5961 • skipCount

- 5962 • includeAllowableActions
- 5963 • includeRelationships

### 59643.7.3.2 POST

5965When an atom entry is POST'ed to this collection, the atom entry will be checked out. A Content-Location  
5966header MUST be returned containing the location of the private working copy.

5967

5968Example client request:

```
5969 POST /CheckedOut HTTP/1.1
5970 Host: example.org
5971 Content-Length: 1044
5972 Content-Type: application/atom+xml;type=entry
5973
5974 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5975 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5976 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5977 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5978 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5979 open.org/ns/cmis/restatom/200908/">
5980   <atom:author>
5981     <atom:name>Al Brown</atom:name>
5982   </atom:author>
5983   <atom:content src="http://cmisexample.oasis-open.org/rep1/46559af4-db97-
5984 471d-b229-d9b27322bf43"/>
5985   <atom:id>urn:uuid:46559af4-db97-471d-b229-d9b27322bf43</atom:id>
5986   <atom:title type="text">CMIS Example Document to checkout</atom:title>
5987   <atom:updated>2009-10-19T10:10:01.031-07:00</atom:updated>
5988   <cmisra:object>
5989     <cmis:properties>
5990       <cmis:propertyId localName="rep-cmis:objectId"
5991 propertyDefinitionId="cmis:objectId">
5992         <cmis:value>46559af4-db97-471d-b229-d9b27322bf43</cmis:value>
5993       </cmis:propertyId>
5994     </cmis:properties>
5995   </cmisra:object>
5996 </atom:entry>
5997
5998
5999
```

6000Example server response:

```
6001 HTTP/1.1 201 Created
6002 Date: Mon, 19 Oct 2009 10:10:01 -0700
6003 Content-Length: 7846
6004 Content-Type: application/atom+xml;type=entry
6005 Content-Location: http://cmisexample.oasis-open.org/rep1/64f55634-f2de-443c-
6006 bf6b-e9e30341581f
6007 Location: http://cmisexample.oasis-open.org/rep1/64f55634-f2de-443c-bf6b-
6008 e9e30341581f
6009
6010 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6011 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6012 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6013 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6014 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6015 open.org/ns/cmis/restatom/200908/">
6016   <atom:author>
6017     <atom:name>Al Brown</atom:name>
6018
```

```

6019         <atom:uri>http://www.ibm.com/</atom:uri>
6020         <atom:email>albertcbrown@us.ibm.com</atom:email>
6021     </atom:author>
6022     <atom:content src="http://cmisexample.oasis-open.org/repl/64f55634-f2de-
6023 443c-bf6b-e9e30341581f"/>
6024     <atom:id>urn:uuid:64f55634-f2de-443c-bf6b-e9e30341581f</atom:id>
6025     <atom:title type="text">CMIS Example Child of Folder</atom:title>
6026     <atom:updated>2009-10-19T10:10:01.046-07:00</atom:updated>
6027     <atom:link rel="self" href="http://cmisexample.oasis-
6028 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f"/>
6029     <atom:link rel="edit" href="http://cmisexample.oasis-
6030 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f"/>
6031     <atom:link type="application/cmisis+xml;type=allowableActions"
6032 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
6033 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
6034 e9e30341581f/allowableactions"/>
6035     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6036 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
6037 e9e30341581f/type"/>
6038     <atom:link type="application/atomsvc+xml" rel="service"
6039 href="http://cmisexample.oasis-open.org/repl//service"/>
6040     <atom:published>2009-10-19T10:10:01.046-07:00</atom:published>
6041     <atom:summary type="html">HTML summary of Entry 64f55634-f2de-443c-bf6b-
6042 e9e30341581f</atom:summary>
6043     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6044 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/edit-media"/>
6045     <atom:link rel="alternate" href="http://cmisexample.oasis-
6046 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/alternate"/>
6047     <atom:link type="application/atom+xml;type=feed" rel="up"
6048 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
6049 e9e30341581f/parents"/>
6050     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6051 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
6052 e9e30341581f/allversions"/>
6053     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6054 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
6055 e9e30341581f/latest"/>
6056     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6057 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
6058 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/relationships"/>
6059     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6060 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
6061 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/policies"/>
6062     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
6063 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
6064 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/acl"/>
6065     <atom:link type="application/atom+xml;type=feed" rel="working-copy"
6066 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
6067 e9e30341581f/pwc"/>
6068     <cmisra:object>
6069         <cmis:properties>
6070             <cmis:propertyId localName="rep-cmis:objectId"
6071 propertyDefinitionId="cmis:objectId">
6072                 <cmis:value>64f55634-f2de-443c-bf6b-e9e30341581f</cmis:value>
6073             </cmis:propertyId>
6074             <cmis:propertyId localName="rep-cmis:objectTypeId"
6075 propertyDefinitionId="cmis:objectTypeId">
6076                 <cmis:value>customer</cmis:value>
6077             </cmis:propertyId>
6078             <cmis:propertyString localName="rep-cmis:name"
6079 propertyDefinitionId="cmis:name">
6080                 <cmis:value>CMIS Example Child of Folder</cmis:value>
6081             </cmis:propertyString>

```

```

6082         <cmis:propertyDateTime localName="rep-cmis:creationDate"
6083 propertyDefinitionId="cmis:creationDate">
6084         <cmis:value>2009-10-19T10:10:01.046-07:00</cmis:value>
6085         </cmis:propertyDateTime>
6086         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6087 propertyDefinitionId="cmis:lastModificationDate">
6088         <cmis:value>2009-10-19T10:10:01.046-07:00</cmis:value>
6089         </cmis:propertyDateTime>
6090         <cmis:propertyId localName="rep-cmis:baseTypeId"
6091 propertyDefinitionId="cmis:baseTypeId">
6092         <cmis:value>cmis:document</cmis:value>
6093         </cmis:propertyId>
6094         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6095 propertyDefinitionId="cmis:lastModifiedBy">
6096         <cmis:value>Al Brown</cmis:value>
6097         </cmis:propertyString>
6098         <cmis:propertyString localName="rep-cmis:createdBy"
6099 propertyDefinitionId="cmis:createdBy">
6100         <cmis:value>Al Brown</cmis:value>
6101         </cmis:propertyString>
6102         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6103 propertyDefinitionId="cmis:isLatestVersion">
6104         <cmis:value>true</cmis:value>
6105         </cmis:propertyBoolean>
6106         <cmis:propertyBoolean localName="rep-
6107 cmis:isVersionSeriesCheckedOut"
6108 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6109         <cmis:value>true</cmis:value>
6110         </cmis:propertyBoolean>
6111         <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6112 propertyDefinitionId="cmis:isMajorVersion">
6113         <cmis:value>false</cmis:value>
6114         </cmis:propertyBoolean>
6115         <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6116 propertyDefinitionId="cmis:isLatestMajorVersion">
6117         <cmis:value>false</cmis:value>
6118         </cmis:propertyBoolean>
6119         <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6120 propertyDefinitionId="cmis:isImmutable">
6121         <cmis:value>false</cmis:value>
6122         </cmis:propertyBoolean>
6123         <cmis:propertyString localName="rep-cmis:checkinComment"
6124 propertyDefinitionId="cmis:checkinComment">
6125         <cmis:value>Checkin comment</cmis:value>
6126         </cmis:propertyString>
6127         <cmis:propertyString localName="rep-cmis:versionLabel"
6128 propertyDefinitionId="cmis:versionLabel">
6129         <cmis:value>0.1</cmis:value>
6130         </cmis:propertyString>
6131         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6132 propertyDefinitionId="cmis:contentStreamMimeType">
6133         <cmis:value>text/plain</cmis:value>
6134         </cmis:propertyString>
6135         <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6136 propertyDefinitionId="cmis:contentStreamFileName">
6137         <cmis:value>text.txt</cmis:value>
6138         </cmis:propertyString>
6139         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6140 propertyDefinitionId="cmis:contentStreamLength">
6141         <cmis:value>4234</cmis:value>
6142         </cmis:propertyInteger>
6143         <cmis:propertyString displayName="Keywords for Document"
6144 localName="keywords" propertyDefinitionId="keywords">

```

```

6145         <cmis:value>document</cmis:value>
6146         <cmis:value>example</cmis:value>
6147         <cmis:value>sample</cmis:value>
6148         <cmis:value>cmis</cmis:value>
6149     </cmis:propertyString>
6150     <cmis:propertyId localName="rep-cmis:versionSeriesCheckedOutId"
6151 propertyDefinitionId="cmis:versionSeriesCheckedOutId">
6152         <cmis:value>64f55634-f2de-443c-bf6b-e9e30341581f</cmis:value>
6153     </cmis:propertyId>
6154     <cmis:propertyString localName="rep-
6155 cmis:versionSeriesCheckedOutBy"
6156 propertyDefinitionId="cmis:versionSeriesCheckedOutBy">
6157         <cmis:value>Al Brown</cmis:value>
6158     </cmis:propertyString>
6159 </cmis:properties>
6160 </cmisra:object>
6161 </atom:entry>

```

6162

6163

6164 Please also see the example documents included with the schema.

6165

### 6166 3.7.4 Unfiled Collection

6167 This is a collection described in the service document that contains all the unfiled documents in the  
6168 repository. If unfiled is supported by the repository

6169 CMIS Services:

6170 GET: getUnfiled

6171 POST: removeObjectFromFolder

6172 Media Type: application/atom+xml;type=feed

6173 Accept:

6174 • MUST support Atom Entry Documents with CMIS extensions

6175     ○ application/atom+xml;type=entry or

6176     ○ application/cmisaatom+xml

6177 • MAY support other media type

6178

6179 Link Relations:

6180 • service: Points to service document containing the CMIS repository. The service document  
6181 MUST contain only one workspace element.

6182     ○ Media Type: application/atomsvc+xml

6183 • paging link relations as appropriate: first, next, previous, last

6184

6185 The following CMIS Atom extension element MAY be included inside the atom feed:

6186 • cmisra:numItems

6187

6188 The following CMIS Atom extension element MUST be included inside the atom entries:

6189 • cmisra:object inside atom:entry

6190

#### 61913.7.4.1 GET

6192The following arguments may be supplied. Please see the domain model for more information:

- 6193 • filter
- 6194 • folderId
- 6195 • maxItems
- 6196 • skipCount
- 6197 • includeAllowableActions
- 6198 • includeRelationships

#### 61993.7.4.2 POST

6200This removes the object from all folders in the repository by default. If the optional argument removeFrom  
6201is specified, the object will only be removed from that folder only.

6202

6203If the Atom Entry POST'ed, does not have the CMIS extensions with a valid cmis:objectId, the document  
6204does not exist, or the document is not in that folder, the appropriate HTTP status code MUST be returned.

6205

6206This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 6207 • HTTP Success: 201
- 6208 • Location Header

6209

6210The following arguments may be supplied. Please see the domain model for more information:

- 6211 • removeFrom: For repositories which support multi-filing, this parameter identifies which folder to  
6212 remove this object from. If specified, it indicates the folder from which the object shall be moved.  
6213 If not specified, the object will be removed from all folders.

6214

6215Example client request:

```
6216 POST /Unfiled HTTP/1.1
6217 Host: example.org
6218 Content-Length: 1043
6219 Content-Type: application/atom+xml;type=entry
6220
6221 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6222 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6223 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6224 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6225 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6226 open.org/ns/cmis/restatom/200908/">
6227   <atom:author>
6228     <atom:name>Al Brown</atom:name>
6229   </atom:author>
6230   <atom:content src="http://cmisexample.oasis-open.org/repl/1f8ceb22-cc15-
6231 4d75-9221-00588cd22bdc"/>
6232   <atom:id>urn:uuid:1f8ceb22-cc15-4d75-9221-00588cd22bdc</atom:id>
6233   <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6234   <atom:updated>2009-10-19T10:10:01.078-07:00</atom:updated>
6235   <cmisra:object>
6236     <cmis:properties>
6237       <cmis:propertyId localName="rep-cmis:objectId"
6238 propertyDefinitionId="cmis:objectId">
6239         <cmis:value>1f8ceb22-cc15-4d75-9221-00588cd22bdc</cmis:value>
6240
```

```

6241         </cmis:propertyId>
6242     </cmis:properties>
6243     </cmisra:object>
6244 </atom:entry>

```

6245

# 6246 Example server response:

```

6248 HTTP/1.1 201 Created
6249 Date: Mon, 19 Oct 2009 10:10:01 -0700
6250 Content-Length: 7234
6251 Content-Type: application/atom+xml;type=entry
6252 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/e2d2c9b3-
6253 dab4-4c65-866a-d8006d1bb16f
6254 Location: http://cmisexample.oasis-open.org/rep1/queryresult/e2d2c9b3-dab4-
6255 4c65-866a-d8006d1bb16f
6256
6257 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6258 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6259 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6260 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6261 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6262 open.org/ns/cmisis/restatom/200908/">
6263     <atom:author>
6264         <atom:name>Al Brown</atom:name>
6265         <atom:uri>http://www.ibm.com/</atom:uri>
6266         <atom:email>albertcbrown@us.ibm.com</atom:email>
6267     </atom:author>
6268     <atom:content src="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-
6269 4c65-866a-d8006d1bb16f"/>
6270     <atom:id>urn:uuid:e2d2c9b3-dab4-4c65-866a-d8006d1bb16f</atom:id>
6271     <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6272     <atom:updated>2009-10-19T10:10:01.093-07:00</atom:updated>
6273     <atom:link rel="self" href="http://cmisexample.oasis-
6274 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f"/>
6275     <atom:link rel="edit" href="http://cmisexample.oasis-
6276 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f"/>
6277     <atom:link type="application/cmisis+xml;type=allowableActions"
6278 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
6279 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
6280 d8006d1bb16f/allowableactions"/>
6281     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6282 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
6283 d8006d1bb16f/type"/>
6284     <atom:link type="application/atomsvc+xml" rel="service"
6285 href="http://cmisexample.oasis-open.org/rep1//service"/>
6286     <atom:published>2009-10-19T10:10:01.093-07:00</atom:published>
6287     <atom:summary type="html">HTML summary of Entry e2d2c9b3-dab4-4c65-866a-
6288 d8006d1bb16f</atom:summary>
6289     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6290 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/edit-media"/>
6291     <atom:link rel="alternate" href="http://cmisexample.oasis-
6292 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/alternate"/>
6293     <atom:link type="application/atom+xml;type=feed" rel="up"
6294 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
6295 d8006d1bb16f/parents"/>
6296     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6297 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
6298 d8006d1bb16f/allversions"/>
6299     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6300 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
6301 d8006d1bb16f/latest"/>
6302

```

```

6303     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6304 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
6305 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/relationships"/>
6306     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6307 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
6308 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/policies"/>
6309     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
6310 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
6311 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/acl"/>
6312     <cmisra:object>
6313         <cmis:properties>
6314             <cmis:propertyId localName="rep-cmis:objectId"
6315 propertyDefinitionId="cmis:objectId">
6316                 <cmis:value>e2d2c9b3-dab4-4c65-866a-d8006d1bb16f</cmis:value>
6317             </cmis:propertyId>
6318             <cmis:propertyId localName="rep-cmis:objectTypeId"
6319 propertyDefinitionId="cmis:objectTypeId">
6320                 <cmis:value>customer</cmis:value>
6321             </cmis:propertyId>
6322             <cmis:propertyString localName="rep-cmis:name"
6323 propertyDefinitionId="cmis:name">
6324                 <cmis:value>CMIS Example Document to unfiled</cmis:value>
6325             </cmis:propertyString>
6326             <cmis:propertyDateTime localName="rep-cmis:creationDate"
6327 propertyDefinitionId="cmis:creationDate">
6328                 <cmis:value>2009-10-19T10:10:01.093-07:00</cmis:value>
6329             </cmis:propertyDateTime>
6330             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6331 propertyDefinitionId="cmis:lastModificationDate">
6332                 <cmis:value>2009-10-19T10:10:01.093-07:00</cmis:value>
6333             </cmis:propertyDateTime>
6334             <cmis:propertyId localName="rep-cmis:baseTypeId"
6335 propertyDefinitionId="cmis:baseTypeId">
6336                 <cmis:value>cmis:document</cmis:value>
6337             </cmis:propertyId>
6338             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6339 propertyDefinitionId="cmis:lastModifiedBy">
6340                 <cmis:value>Al Brown</cmis:value>
6341             </cmis:propertyString>
6342             <cmis:propertyString localName="rep-cmis:createdBy"
6343 propertyDefinitionId="cmis:createdBy">
6344                 <cmis:value>Al Brown</cmis:value>
6345             </cmis:propertyString>
6346             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6347 propertyDefinitionId="cmis:isLatestVersion">
6348                 <cmis:value>true</cmis:value>
6349             </cmis:propertyBoolean>
6350             <cmis:propertyBoolean localName="rep-
6351 cmis:isVersionSeriesCheckedOut"
6352 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6353                 <cmis:value>false</cmis:value>
6354             </cmis:propertyBoolean>
6355             <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6356 propertyDefinitionId="cmis:isMajorVersion">
6357                 <cmis:value>false</cmis:value>
6358             </cmis:propertyBoolean>
6359             <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6360 propertyDefinitionId="cmis:isLatestMajorVersion">
6361                 <cmis:value>false</cmis:value>
6362             </cmis:propertyBoolean>
6363             <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6364 propertyDefinitionId="cmis:isImmutable">
6365                 <cmis:value>false</cmis:value>

```



```

6366         </cmis:propertyBoolean>
6367         <cmis:propertyString localName="rep-cmis:checkinComment"
6368 propertyDefinitionId="cmis:checkinComment">
6369             <cmis:value>Checkin comment</cmis:value>
6370         </cmis:propertyString>
6371         <cmis:propertyString localName="rep-cmis:versionLabel"
6372 propertyDefinitionId="cmis:versionLabel">
6373             <cmis:value>0.1</cmis:value>
6374         </cmis:propertyString>
6375         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6376 propertyDefinitionId="cmis:contentStreamMimeType">
6377             <cmis:value>text/plain</cmis:value>
6378         </cmis:propertyString>
6379         <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6380 propertyDefinitionId="cmis:contentStreamFileName">
6381             <cmis:value>text.txt</cmis:value>
6382         </cmis:propertyString>
6383         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6384 propertyDefinitionId="cmis:contentStreamLength">
6385             <cmis:value>4234</cmis:value>
6386         </cmis:propertyInteger>
6387         <cmis:propertyString displayName="Keywords for Document"
6388 localName="keywords" propertyDefinitionId="keywords">
6389             <cmis:value>document</cmis:value>
6390             <cmis:value>example</cmis:value>
6391             <cmis:value>sample</cmis:value>
6392             <cmis:value>cmis</cmis:value>
6393         </cmis:propertyString>
6394     </cmis:properties>
6395 </cmisra:object>
6396 </atom:entry>

```

6398

6399 Please also see the example documents included with the schema.

6400

### 6401 3.7.5 Types Children Collection

6402 This is a collection described in the service document that contains the types in the repository under the  
6403 specified parent type. If no parent type is specified, then the base types are returned in the feed. This  
6404 feed does not include any nesting and is a flat feed.

6405 CMIS Services:

6406 GET: getTypeChildren

6407 Media Type: application/atom+xml;type=feed

6408

6409 Link Relations:

- 6410 • service: Points to service document containing the CMIS repository. The service document  
6411 MUST contain only one workspace element.
- 6412 • Media Type: application/atomsvc+xml
- 6413 • via: points to the type definition entry whose children represent this feed
- 6414 • down: points to the atom feed document representing the descendants collection for this same  
6415 type with media type of application/cmistree+xml
- 6416 • paging link relations as appropriate: first, next, previous, last
- 6417 • up: points to the parent type definition

6418           ○ If this is a children feed for a base object type, this link is not present.

6419

6420 This feed contains a set of atom entries for each child type definition.

6421

6422 The following CMIS Atom extension element MAY be included inside the atom feed:

6423     • cmisra:numItems

6424

6425 The following CMIS Atom extension element MUST be included inside the atom entries:

6426     • cmisra:type inside atom:entry

6427

6428

### 6429 **3.7.5.1 GET**

6430 The following arguments may be supplied. Please see the domain model for more information:

6431     • includePropertyDefinitions

6432     • maxItems

6433     • skipCount

6434     • typeId

## 6435 **3.8 Collections**

### 6436 **3.8.1 Relationships Collection**

6437 This is the set of relationships available (either source or target or both) from a specific item such as a  
6438 document, folder or policy.

6439 CMIS Services:

6440     GET: getObjectRelationships

6441     POST: createRelationship

6442 Media Type: application/atom+xml;type=feed

6443 Accept:

6444     • MUST support Atom Entry Documents with CMIS extensions

6445           ○ application/atom+xml;type=entry or

6446           ○ application/cmisatom+xml

6447     • MAY support other media type

6448

6449 Link Relations:

6450     • service: Points to service document containing the CMIS repository. The service document  
6451       MUST contain only one workspace element.

6452           ○ Media Type: application/atomsvc+xml

6453     • paging link relations as appropriate: first, next, previous, last

6454

6455 The following CMIS Atom extension element MAY be included inside the atom feed:

6456     • cmisra:numItems

6457

6458The following CMIS Atom extension element MUST be included inside the atom entries:

- 6459     • cmisra:object inside atom:entry

6460

### 64613.8.1.1 GET

6462The following arguments may be supplied. Please see the domain model for more information:

- 6463     • typeId  
6464     • includeSubRelationshipTypes  
6465     • relationshipDirection  
6466     • maxItems  
6467     • skipCount  
6468     • filter  
6469     • includeAllowableActions

### 64703.8.1.2 POST

6471When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new  
6472CMIS relationship, then that relationship will be created.

6473The server MUST return the appropriate HTTP status code if the source is different than the sourceId or  
6474target different than the targetId for the source and targets specified in this collection.

6475The server MUST return the appropriate status code if the cmis:objectTypeId is not specified.

6476

6477Example client request:

```
6478 POST /relationships/source/76a5d5bc-cdfe-42ac-9fde-dd09ad384a71 HTTP/1.1
6479 Host: example.org
6480 Content-Length: 1432
6481 Content-Type: application/atom+xml;type=entry
6482
6483 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6484 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6485 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6486 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6487 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6488 open.org/ns/cmis/restatom/200908/">
6489   <atom:author>
6490     <atom:name>Al Brown</atom:name>
6491   </atom:author>
6492   <atom:content src="http://cmisexample.oasis-open.org/repl/235d5fdc-6193-
6493 4ec0-8a78-c0da832a06d9"/>
6494   <atom:id>urn:uuid:235d5fdc-6193-4ec0-8a78-c0da832a06d9</atom:id>
6495   <atom:title type="text">New Relationship</atom:title>
6496   <atom:updated>2009-10-19T10:09:59.796-07:00</atom:updated>
6497   <cmisra:object>
6498     <cmis:properties>
6499       <cmis:propertyId localName="rep-cmis:objectTypeId"
6500 propertyDefinitionId="cmis:objectTypeId">
6501         <cmis:value>customerRelationships</cmis:value>
6502       </cmis:propertyId>
6503       <cmis:propertyId localName="rep-cmis:sourceId"
6504 propertyDefinitionId="cmis:sourceId">
6505         <cmis:value>76a5d5bc-cdfe-42ac-9fde-dd09ad384a71</cmis:value>
6506
```

```

6507         </cmis:propertyId>
6508         <cmis:propertyId localName="rep-cmis:targetId"
6509 propertyDefinitionId="cmis:targetId">
6510             <cmis:value>5f3afe79-1e78-4ff9-a3bf-782c5a492e11</cmis:value>
6511         </cmis:propertyId>
6512     </cmis:properties>
6513 </cmisra:object>
6514 </atom:entry>

```

6515

6516

6517Example server response:

```

6518 HTTP/1.1 201 Created
6519 Date: Mon, 19 Oct 2009 10:09:59 -0700
6520 Content-Length: 4684
6521 Content-Type: application/atom+xml;type=entry
6522 Content-Location: http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-
6523 8cd2-30900d7a73a6
6524 Location: http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-8cd2-
6525 30900d7a73a6
6526
6527 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6528 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6529 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6530 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6531 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6532 open.org/ns/cmis/restatom/200908/">
6533     <atom:author>
6534         <atom:name>Al Brown</atom:name>
6535         <atom:uri>http://www.ibm.com/</atom:uri>
6536         <atom:email>albertcbrown@us.ibm.com</atom:email>
6537     </atom:author>
6538     <atom:content src="http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-
6539 4e5c-8cd2-30900d7a73a6"/>
6540     <atom:id>urn:uuid:5f3287a5-7442-4e5c-8cd2-30900d7a73a6</atom:id>
6541     <atom:title type="text">New Relationship</atom:title>
6542     <atom:updated>2009-10-19T10:09:59.812-07:00</atom:updated>
6543     <atom:link rel="self" href="http://cmisexample.oasis-
6544 open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6"/>
6545     <atom:link rel="edit" href="http://cmisexample.oasis-
6546 open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6"/>
6547     <atom:link type="application/cmis+xml;type=allowableActions"
6548 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6549 href="http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-8cd2-
6550 30900d7a73a6/allowableactions"/>
6551     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6552 href="http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-8cd2-
6553 30900d7a73a6/type"/>
6554     <atom:link type="application/atomsvc+xml" rel="service"
6555 href="http://cmisexample.oasis-open.org/rep1//service"/>
6556     <atom:published>2009-10-19T10:09:59.812-07:00</atom:published>
6557     <atom:summary type="html">HTML summary of Entry 5f3287a5-7442-4e5c-8cd2-
6558 30900d7a73a6</atom:summary>
6559     <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
6560 open.org/ns/cmis/link/200908/source" href="http://cmisexample.oasis-
6561 open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/source"/>
6562     <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
6563 open.org/ns/cmis/link/200908/target" href="http://cmisexample.oasis-
6564 open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/target"/>
6565     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6566 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
6567 open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/policies"/>
6568

```

```

6569     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6570 open.org/ns/cmisacl/link/200908/acl" href="http://cmisexample.oasis-
6571 open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/acl"/>
6572     <cmisra:object>
6573       <cmis:properties>
6574         <cmis:propertyId localName="rep-cmis:objectId"
6575 propertyDefinitionId="cmis:objectId">
6576           <cmis:value>5f3287a5-7442-4e5c-8cd2-30900d7a73a6</cmis:value>
6577         </cmis:propertyId>
6578         <cmis:propertyId localName="rep-cmis:objectTypeId"
6579 propertyDefinitionId="cmis:objectTypeId">
6580           <cmis:value>customerRelationships</cmis:value>
6581         </cmis:propertyId>
6582         <cmis:propertyString localName="rep-cmis:name"
6583 propertyDefinitionId="cmis:name">
6584           <cmis:value>New Relationship</cmis:value>
6585         </cmis:propertyString>
6586         <cmis:propertyDateTime localName="rep-cmis:creationDate"
6587 propertyDefinitionId="cmis:creationDate">
6588           <cmis:value>2009-10-19T10:09:59.812-07:00</cmis:value>
6589         </cmis:propertyDateTime>
6590         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6591 propertyDefinitionId="cmis:lastModificationDate">
6592           <cmis:value>2009-10-19T10:09:59.812-07:00</cmis:value>
6593         </cmis:propertyDateTime>
6594         <cmis:propertyId localName="rep-cmis:baseTypeId"
6595 propertyDefinitionId="cmis:baseTypeId">
6596           <cmis:value>cmis:relationship</cmis:value>
6597         </cmis:propertyId>
6598         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6599 propertyDefinitionId="cmis:lastModifiedBy">
6600           <cmis:value>Al Brown</cmis:value>
6601         </cmis:propertyString>
6602         <cmis:propertyString localName="rep-cmis:createdBy"
6603 propertyDefinitionId="cmis:createdBy">
6604           <cmis:value>Al Brown</cmis:value>
6605         </cmis:propertyString>
6606         <cmis:propertyId localName="rep-cmis:sourceId"
6607 propertyDefinitionId="cmis:sourceId">
6608           <cmis:value>36f25b8d-c920-4d1f-86ab-3fb8d7ea5f97</cmis:value>
6609         </cmis:propertyId>
6610         <cmis:propertyId localName="rep-cmis:targetId"
6611 propertyDefinitionId="cmis:targetId">
6612           <cmis:value>3f4b0e37-a49d-4bf0-b71d-f8ed0c865029</cmis:value>
6613         </cmis:propertyId>
6614       </cmis:properties>
6615     </cmisra:object>
6616   </atom:entry>

```

6617

6618

6619 Please also see the example documents included with the schema.

6620

## 6621 3.8.2 Folder Children Collection

6622 This is a collection comprised of all the direct children of a particular folder represented as a feed.

6623 CMIS Services:

6624 GET: getChildren

6625 POST:

6626                   createDocument  
6627                   or createFolder  
6628                   or createPolicy  
6629                   or moveObject  
6630                   or addObjectToFolder  
6631  
6632Media Type: application/atom+xml;type=feed  
6633  
6634Accept:  
6635     • MUST support Atom Entry Documents with CMIS extensions  
6636     • MAY support other media type  
6637  
6638Link Relations:  
6639     • service: Points to service document containing the CMIS repository. The service document  
6640       MUST contain only one workspace element.  
6641       ○ Media Type: application/atomsvc+xml  
6642     • via: points to the atom entry of the folder generating this collection  
6643     • up: points to the atom entry document for this folder's parent  
6644       ○ If the root folder, this link relation MUST NOT be included.  
6645       ○ Media Type: application/atom+xml;type=entry  
6646     • down: points to the atom feed document representing the descendants feed with a media type of  
6647       application/cmistree+xml  
6648       ○ If a repository does not support capabilityGetDescendants, then this link SHOULD NOT  
6649       be included.  
6650     • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder.  
6651       This is represented as a feed with CMIS hierarchy extensions.  
6652       ○ Media Type: application/atom+xml;type=feed  
6653     • paging link relations as appropriate: first, next, previous, last  
6654  
6655The following CMIS Atom extension element MAY be included inside the atom feed:  
6656     • cmisra:numItems  
6657  
6658The following CMIS Atom extension element MUST be included inside the atom entries:  
6659     • cmisra:object inside atom:entry  
6660     • cmisra:pathSegment inside atom:entry  
6661

### 66623.8.2.1 GET

6663HTTP Code:  
6664     • 200 OK (Success)  
6665

6666The following arguments may be supplied. Please see the domain model for more information:

- 6667           • maxItems
- 6668           • skipCount
- 6669           • filter
- 6670           • includeAllowableActions
- 6671           • includeRelationships
- 6672           • renditionFilter
  - 6673               ○ If specified, renditions will be returned as links with relation alternate.
- 6674           • orderBy
- 6675           • includePathSegment

### 66763.8.2.2 POST

6677CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please  
 6678see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 6679           • HTTP Success: 201
- 6680           • Location Header

6681

6682The following arguments MAY be supplied.

- 6683           • sourceFolderId: This parameter indicates the folder from which the object shall be moved from to  
 6684               the current specified folder. This parameter is not allowed for create operations.
    - 6685               ○ If specified moveObject will be performed.
    - 6686               ○ If not specified, addObjectToFolder will be performed.
  - 6687           • versioningState: The optional argument versioningState MAY specify additional versioning  
 6688               behavior such as checkIn as major or minor. Please see CMIS Domain Model for more  
 6689               information on this parameter.
- 6690

6691POSTing an Atom Entry document with CMIS markup:

6692       Adding a document to a folder:

6693       If the atom entry has a cmis property cmis:objectId that is valid for the repository, the object will  
 6694       be added to the folder.

6695

6696       When an object is added to the folder, in repositories that do not support multi-filing it will be  
 6697       removed from the previous folder and the operation treated as move. If the repository supports  
 6698       multiple folders, it will be added to the new folder.

6699       If the optional argument sourceFolderId is specified, then the object will be removed from the  
 6700       folder specified.

6701

6702       Example client request:

```
6703 POST /obj/8a7761eb-2a0c-4400-b515-964308d6cb5e?sourceFolderId=0571b0e4-8043-
6704 4e48-8a89-448206c2b365 HTTP/1.1
6705 Host: example.org
6706 Content-Length: 1227
6707 Content-Type: application/atom+xml;type=entry
6708
6709 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6710
```

```

6711 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6712 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6713 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6714 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6715 open.org/ns/cmis/restatom/200908/">
6716   <atom:author>
6717     <atom:name>Al Brown</atom:name>
6718   </atom:author>
6719   <atom:id>urn:uuid:8a7761eb-2a0c-4400-b515-964308d6cb5e</atom:id>
6720   <atom:title type="text">Document - To Be Moved</atom:title>
6721   <atom:updated>2009-10-19T10:09:59.640-07:00</atom:updated>
6722   <atom:content src="http://cmisexample.oasis-
6723 open.org/rep1/content/8a7761eb-2a0c-4400-b515-964308d6cb5e"/>
6724   <cmisra:object>
6725     <cmis:properties>
6726       <cmis:propertyId localName="rep-cmis:objectId"
6727 propertyDefinitionId="cmis:objectId">
6728         <cmis:value>8a7761eb-2a0c-4400-b515-964308d6cb5e</cmis:value>
6729       </cmis:propertyId>
6730       <cmis:propertyId localName="rep-cmis:objectTypeId"
6731 propertyDefinitionId="cmis:objectTypeId">
6732         <cmis:value>invoice</cmis:value>
6733       </cmis:propertyId>
6734     </cmis:properties>
6735   </cmisra:object>
6736 </atom:entry>

```

#### Example server response:

```

6740 HTTP/1.1 201 Created
6741 Date: Mon, 19 Oct 2009 10:09:59 -0700
6742 Content-Length: 7213
6743 Content-Type: application/atom+xml;type=entry
6744 Content-Location: http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-
6745 826b-3aa0a8eb8b15
6746 Location: http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6747 3aa0a8eb8b15
6748
6749 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6750 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6751 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6752 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6753 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6754 open.org/ns/cmis/restatom/200908/">
6755   <atom:author>
6756     <atom:name>Al Brown</atom:name>
6757     <atom:uri>http://www.ibm.com/</atom:uri>
6758     <atom:email>albertcbrown@us.ibm.com</atom:email>
6759   </atom:author>
6760   <atom:content src="http://cmisexample.oasis-open.org/rep1/69553581-2311-
6761 4aa4-826b-3aa0a8eb8b15"/>
6762   <atom:id>urn:uuid:69553581-2311-4aa4-826b-3aa0a8eb8b15</atom:id>
6763   <atom:title type="text">Document - To Be Moved</atom:title>
6764   <atom:updated>2009-10-19T10:09:59.671-07:00</atom:updated>
6765   <atom:link rel="self" href="http://cmisexample.oasis-
6766 open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15"/>
6767   <atom:link rel="edit" href="http://cmisexample.oasis-
6768 open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15"/>
6769

```



```

6770     <atom:link type="application/cmism+xml;type=allowableActions"
6771     rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
6772     href="http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6773     3aa0a8eb8b15/allowableactions"/>
6774     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6775     href="http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6776     3aa0a8eb8b15/type"/>
6777     <atom:link type="application/atomsvc+xml" rel="service"
6778     href="http://cmisexample.oasis-open.org/rep1//service"/>
6779     <atom:published>2009-10-19T10:09:59.687-07:00</atom:published>
6780     <atom:summary type="html">HTML summary of Entry 69553581-2311-4aa4-826b-
6781     3aa0a8eb8b15</atom:summary>
6782     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6783     open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15/edit-media"/>
6784     <atom:link rel="alternate" href="http://cmisexample.oasis-
6785     open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15/alternate"/>
6786     <atom:link type="application/atom+xml;type=feed" rel="up"
6787     href="http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6788     3aa0a8eb8b15/parents"/>
6789     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6790     href="http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6791     3aa0a8eb8b15/allversions"/>
6792     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6793     href="http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6794     3aa0a8eb8b15/latest"/>
6795     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6796     open.org/ns/cmism/link/200908/relationships" href="http://cmisexample.oasis-
6797     open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15/relationships"/>
6798     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6799     open.org/ns/cmism/link/200908/policies" href="http://cmisexample.oasis-
6800     open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15/policies"/>
6801     <atom:link type="application/cmism+xml" rel="http://docs.oasis-
6802     open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
6803     open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15/acl"/>
6804     <cmism:object>
6805         <cmism:properties>
6806             <cmism:propertyId localName="rep-cmism:objectId"
6807             propertyDefinitionId="cmism:objectId">
6808                 <cmism:value>69553581-2311-4aa4-826b-3aa0a8eb8b15</cmism:value>
6809             </cmism:propertyId>
6810             <cmism:propertyId localName="rep-cmism:objectTypeId"
6811             propertyDefinitionId="cmism:objectTypeId">
6812                 <cmism:value>invoice</cmism:value>
6813             </cmism:propertyId>
6814             <cmism:propertyString localName="rep-cmism:name"
6815             propertyDefinitionId="cmism:name">
6816                 <cmism:value>Document - To Be Moved</cmism:value>
6817             </cmism:propertyString>
6818             <cmism:propertyDateTime localName="rep-cmism:creationDate"
6819             propertyDefinitionId="cmism:creationDate">
6820                 <cmism:value>2009-10-19T10:09:59.687-07:00</cmism:value>
6821             </cmism:propertyDateTime>
6822             <cmism:propertyDateTime localName="rep-cmism:lastModificationDate"
6823             propertyDefinitionId="cmism:lastModificationDate">
6824                 <cmism:value>2009-10-19T10:09:59.687-07:00</cmism:value>
6825             </cmism:propertyDateTime>
6826             <cmism:propertyId localName="rep-cmism:baseTypeId"
6827             propertyDefinitionId="cmism:baseTypeId">
6828                 <cmism:value>cmism:document</cmism:value>
6829             </cmism:propertyId>
6830             <cmism:propertyString localName="rep-cmism:lastModifiedBy"
6831             propertyDefinitionId="cmism:lastModifiedBy">
6832                 <cmism:value>Al Brown</cmism:value>

```

```

6833         </cmis:propertyString>
6834         <cmis:propertyString localName="rep-cmis:createdBy"
6835 propertyDefinitionId="cmis:createdBy">
6836             <cmis:value>Al Brown</cmis:value>
6837         </cmis:propertyString>
6838         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6839 propertyDefinitionId="cmis:isLatestVersion">
6840             <cmis:value>true</cmis:value>
6841         </cmis:propertyBoolean>
6842         <cmis:propertyBoolean localName="rep-
6843 cmis:isVersionSeriesCheckedOut"
6844 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6845             <cmis:value>false</cmis:value>
6846         </cmis:propertyBoolean>
6847         <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6848 propertyDefinitionId="cmis:isMajorVersion">
6849             <cmis:value>false</cmis:value>
6850         </cmis:propertyBoolean>
6851         <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6852 propertyDefinitionId="cmis:isLatestMajorVersion">
6853             <cmis:value>false</cmis:value>
6854         </cmis:propertyBoolean>
6855         <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6856 propertyDefinitionId="cmis:isImmutable">
6857             <cmis:value>false</cmis:value>
6858         </cmis:propertyBoolean>
6859         <cmis:propertyString localName="rep-cmis:checkinComment"
6860 propertyDefinitionId="cmis:checkinComment">
6861             <cmis:value>Checkin comment</cmis:value>
6862         </cmis:propertyString>
6863         <cmis:propertyString localName="rep-cmis:versionLabel"
6864 propertyDefinitionId="cmis:versionLabel">
6865             <cmis:value>0.1</cmis:value>
6866         </cmis:propertyString>
6867         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6868 propertyDefinitionId="cmis:contentStreamMimeType">
6869             <cmis:value>text/plain</cmis:value>
6870         </cmis:propertyString>
6871         <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6872 propertyDefinitionId="cmis:contentStreamFileName">
6873             <cmis:value>text.txt</cmis:value>
6874         </cmis:propertyString>
6875         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6876 propertyDefinitionId="cmis:contentStreamLength">
6877             <cmis:value>4234</cmis:value>
6878         </cmis:propertyInteger>
6879         <cmis:propertyString displayName="Keywords for Document"
6880 localName="keywords" propertyDefinitionId="keywords">
6881             <cmis:value>document</cmis:value>
6882             <cmis:value>example</cmis:value>
6883             <cmis:value>sample</cmis:value>
6884             <cmis:value>cmis</cmis:value>
6885         </cmis:propertyString>
6886     </cmis:properties>
6887 </cmisra:object>
6888 </atom:entry>

```

Please also see the example documents included with the schema.

Creating a CMIS Object (in that folder):

If the cmis:objectId property is missing, the object will be created and then added to the folder. If the cmis:objectId property is present but not a valid object Id, the repository MUST return the appropriate HTTP status code.

For Documents:

If Content Stream is not provided and it is required by the type definition, the repository MUST return the appropriate HTTP status code.

Content Streams MAY be provided by any of the following mechanisms:

- As part of the atom entry via the src attribute on the content element (AtomPub)
  - src attribute: Implementers MAY support external references to content
  - If the URI in the src attribute is not reachable, then an appropriate http status code should be returned.
- As part of the atom entry inlining via the content element (AtomPub)
  - Please see the AtomPub specification RFC5023 for the processing model of the content element.
- If the cmisra:content is provided by the client inside the atom:entry, the cmisra:content element MUST take precedence over the atom:content element. (CMIS)
  - This element cmisra:content is base64 encoded
- At a later time (AtomPub)
  - 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 client request:

```
POST /obj/4f833ce6-f53d-4fb5-a268-7c53705a0807 HTTP/1.1
Host: example.org
Content-Length: 1190
Content-Type: application/atom+xml;type=entry

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:entry xmlns:app="http://www.w3.org/2007/app"
xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
open.org/ns/cmis/restatom/200908/">
  <atom:author>
    <atom:name>Al Brown</atom:name>
  </atom:author>
  <atom:id>urn:uuid:4f833ce6-f53d-4fb5-a268-7c53705a0807</atom:id>
  <atom:title type="text">New Invoice</atom:title>
  <atom:updated>2009-10-19T10:09:59.718-07:00</atom:updated>
  <atom:content type="text">this is the content of the new
document</atom:content>
  <cmisra:object>
    <cmis:properties>
```

```

6945         <cmis:propertyId localName="rep-cmis:objectId"
6946 propertyDefinitionId="cmis:objectId">
6947         <cmis:value>4f833ce6-f53d-4fb5-a268-7c53705a0807</cmis:value>
6948         </cmis:propertyId>
6949         <cmis:propertyId localName="rep-cmis:objectTypeId"
6950 propertyDefinitionId="cmis:objectTypeId">
6951         <cmis:value>invoice</cmis:value>
6952         </cmis:propertyId>
6953     </cmis:properties>
6954 </cmisra:object>
6955 </atom:entry>

```

#### Example server response:

```

6959 HTTP/1.1 201 Created
6960 Date: Mon, 19 Oct 2009 10:09:59 -0700
6961 Content-Length: 7191
6962 Content-Type: application/atom+xml;type=entry
6963 Content-Location: http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-
6964 8d72-ea72987af212
6965 Location: http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
6966 ea72987af212
6967
6968 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6969 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6970 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6971 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6972 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6973 open.org/ns/cmis/restatom/200908/">
6974   <atom:author>
6975     <atom:name>Al Brown</atom:name>
6976     <atom:uri>http://www.ibm.com/</atom:uri>
6977     <atom:email>albertcbrown@us.ibm.com</atom:email>
6978   </atom:author>
6979   <atom:content src="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-
6980 40ae-8d72-ea72987af212"/>
6981   <atom:id>urn:uuid:8f714453-f9cf-40ae-8d72-ea72987af212</atom:id>
6982   <atom:title type="text">New Invoice</atom:title>
6983   <atom:updated>2009-10-19T10:09:59.734-07:00</atom:updated>
6984   <atom:link rel="self" href="http://cmisexample.oasis-
6985 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212"/>
6986   <atom:link rel="edit" href="http://cmisexample.oasis-
6987 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212"/>
6988   <atom:link type="application/cmis+xml;type=allowableActions"
6989 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6990 href="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
6991 ea72987af212/allowableactions"/>
6992   <atom:link type="application/atom+xml;type=entry" rel="describedby"
6993 href="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
6994 ea72987af212/type"/>
6995   <atom:link type="application/atomsvc+xml" rel="service"
6996 href="http://cmisexample.oasis-open.org/rep1//service"/>
6997   <atom:published>2009-10-19T10:09:59.734-07:00</atom:published>
6998   <atom:summary type="html">HTML summary of Entry 8f714453-f9cf-40ae-8d72-
7000 ea72987af212</atom:summary>
7001   <atom:link rel="edit-media" href="http://cmisexample.oasis-
7002 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/edit-media"/>
7003   <atom:link rel="alternate" href="http://cmisexample.oasis-
7004 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/alternate"/>

```

```

7005     <atom:link type="application/atom+xml;type=feed" rel="up"
7006 href="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
7007 ea72987af212/parents"/>
7008     <atom:link type="application/atom+xml;type=feed" rel="version-history"
7009 href="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
7010 ea72987af212/allversions"/>
7011     <atom:link type="application/atom+xml;type=entry" rel="current-version"
7012 href="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
7013 ea72987af212/latest"/>
7014     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7015 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
7016 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/relationships"/>
7017     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7018 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
7019 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/policies"/>
7020     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
7021 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
7022 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/acl"/>
7023     <cmisra:object>
7024         <cmis:properties>
7025             <cmis:propertyId localName="rep-cmis:objectId"
7026 propertyDefinitionId="cmis:objectId">
7027                 <cmis:value>8f714453-f9cf-40ae-8d72-ea72987af212</cmis:value>
7028             </cmis:propertyId>
7029             <cmis:propertyId localName="rep-cmis:objectTypeId"
7030 propertyDefinitionId="cmis:objectTypeId">
7031                 <cmis:value>invoice</cmis:value>
7032             </cmis:propertyId>
7033             <cmis:propertyString localName="rep-cmis:name"
7034 propertyDefinitionId="cmis:name">
7035                 <cmis:value>New Invoice</cmis:value>
7036             </cmis:propertyString>
7037             <cmis:propertyDateTime localName="rep-cmis:creationDate"
7038 propertyDefinitionId="cmis:creationDate">
7039                 <cmis:value>2009-10-19T10:09:59.734-07:00</cmis:value>
7040             </cmis:propertyDateTime>
7041             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
7042 propertyDefinitionId="cmis:lastModificationDate">
7043                 <cmis:value>2009-10-19T10:09:59.734-07:00</cmis:value>
7044             </cmis:propertyDateTime>
7045             <cmis:propertyId localName="rep-cmis:baseTypeId"
7046 propertyDefinitionId="cmis:baseTypeId">
7047                 <cmis:value>cmis:document</cmis:value>
7048             </cmis:propertyId>
7049             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7050 propertyDefinitionId="cmis:lastModifiedBy">
7051                 <cmis:value>Al Brown</cmis:value>
7052             </cmis:propertyString>
7053             <cmis:propertyString localName="rep-cmis:createdBy"
7054 propertyDefinitionId="cmis:createdBy">
7055                 <cmis:value>Al Brown</cmis:value>
7056             </cmis:propertyString>
7057             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
7058 propertyDefinitionId="cmis:isLatestVersion">
7059                 <cmis:value>true</cmis:value>
7060             </cmis:propertyBoolean>
7061             <cmis:propertyBoolean localName="rep-
7062 cmis:isVersionSeriesCheckedOut"
7063 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7064                 <cmis:value>false</cmis:value>
7065             </cmis:propertyBoolean>
7066             <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
7067 propertyDefinitionId="cmis:isMajorVersion">

```

```

7068         <cmis:value>false</cmis:value>
7069     </cmis:propertyBoolean>
7070     <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
7071 propertyDefinitionId="cmis:isLatestMajorVersion">
7072         <cmis:value>false</cmis:value>
7073     </cmis:propertyBoolean>
7074     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7075 propertyDefinitionId="cmis:isImmutable">
7076         <cmis:value>false</cmis:value>
7077     </cmis:propertyBoolean>
7078     <cmis:propertyString localName="rep-cmis:checkinComment"
7079 propertyDefinitionId="cmis:checkinComment">
7080         <cmis:value>Checkin comment</cmis:value>
7081     </cmis:propertyString>
7082     <cmis:propertyString localName="rep-cmis:versionLabel"
7083 propertyDefinitionId="cmis:versionLabel">
7084         <cmis:value>0.1</cmis:value>
7085     </cmis:propertyString>
7086     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
7087 propertyDefinitionId="cmis:contentStreamMimeType">
7088         <cmis:value>text/plain</cmis:value>
7089     </cmis:propertyString>
7090     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
7091 propertyDefinitionId="cmis:contentStreamFileName">
7092         <cmis:value>text.txt</cmis:value>
7093     </cmis:propertyString>
7094     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7095 propertyDefinitionId="cmis:contentStreamLength">
7096         <cmis:value>4234</cmis:value>
7097     </cmis:propertyInteger>
7098     <cmis:propertyString displayName="Keywords for Document"
7099 localName="keywords" propertyDefinitionId="keywords">
7100         <cmis:value>document</cmis:value>
7101         <cmis:value>example</cmis:value>
7102         <cmis:value>sample</cmis:value>
7103         <cmis:value>cmis</cmis:value>
7104     </cmis:propertyString>
7105 </cmis:properties>
7106 </cmisra:object>
7107 </atom:entry>

```

7108

7109

7110 Please also see the example documents included with the schema.

7111

7112 POSTing other document formats: (AtomPub)

7113 The behavior is repository specific when a non Atom entry or an atom document without the  
7114 CMIS elements is posted to a folder collection.

7115 For example, the repository MAY auto-create a document with a specific type (document) the  
7116 client could edit.

7117 If the repository does not support this scenario or another exception occurs, then the repository  
7118 MUST return the appropriate HTTP status code.

7119

7120 Optional arguments:

7121 • versioningState (for createDocument)

7122 • sourceFolderId (for moveObject)

7123

### 71243.8.3 Policies Collection

7125 This is an atom feed of all the policy objects currently applied to a specific object

7126 CMIS Services:

7127 GET: getAppliedPolicies

7128 POST: applyPolicy (to object representing this collection of policies)

7129 Media Type: application/atom+xml;type=feed

7130 Accept:

7131 • MUST support Atom Entry Documents with CMIS extensions

7132     ○ application/atom+xml;type=entry or

7133     ○ application/cmisatom+xml

7134 • MAY support other media type

7135

7136 Link Relations:

7137 • service: Points to service document containing the CMIS repository. The service document  
7138 MUST contain only one workspace element.

7139     ○ Media Type: application/atomsvc+xml

7140 • via: points to the atom entry of the resource generating this collection

7141 • paging link relations as appropriate: first, next, previous, last

7142

7143 The policy entries displayed here are specific to the object generating this collection. A DELETE method  
7144 on those URIs will invoke removePolicy().

7145

7146 The following CMIS Atom extension element MAY be included inside the atom feed:

7147 • cmisra:numItems

7148

7149 The following CMIS Atom extension element MUST be included inside the atom entries:

7150 • cmisra:object inside atom:entry

7151

#### 71523.8.3.1 GET

7153 The following arguments may be supplied. Please see the domain model for more information:

7154 • filter

#### 71553.8.3.2 POST

7156 When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the  
7157 object.

7158

7159 Example client request:

7160 POST /policies/f6fc3972-ed27-4b49-9412-b4ef2b0f715e HTTP/1.1

7161 Host: example.org

7162 Content-Length: 1039

7163 Content-Type: application/atom+xml;type=entry

7164

7165

```

7166 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7167 <atom:entry xmlns:app="http://www.w3.org/2007/app"
7168 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
7169 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
7170 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
7171 open.org/ns/cmis/restatom/200908/">
7172   <atom:author>
7173     <atom:name>Al Brown</atom:name>
7174   </atom:author>
7175   <atom:content src="http://cmisexample.oasis-open.org/rep1/f6fc3972-ed27-
7176 4b49-9412-b4ef2b0f715e"/>
7177   <atom:id>urn:uuid:f6fc3972-ed27-4b49-9412-b4ef2b0f715e</atom:id>
7178   <atom:title type="text">Security Policy for Invoices</atom:title>
7179   <atom:updated>2009-10-19T10:09:59.765-07:00</atom:updated>
7180   <cmisra:object>
7181     <cmis:properties>
7182       <cmis:propertyId localName="rep-cmis:objectId"
7183 propertyDefinitionId="cmis:objectId">
7184         <cmis:value>f6fc3972-ed27-4b49-9412-b4ef2b0f715e</cmis:value>
7185       </cmis:propertyId>
7186     </cmis:properties>
7187   </cmisra:object>
7188 </atom:entry>
7189
7190 F

```

7191

#### 7192Example server response:

```

7193 HTTP/1.1 201 Created
7194 Date: Mon, 19 Oct 2009 10:09:59 -0700
7195 Content-Length: 4043
7196 Content-Type: application/atom+xml;type=entry
7197 Content-Location: http://cmisexample.oasis-open.org/rep1/1ee01165-98cf-45bf-
7198 9bce-b6238caa5616
7199 Location: http://cmisexample.oasis-open.org/rep1/1ee01165-98cf-45bf-9bce-
7200 b6238caa5616
7201
7202 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7203 <atom:entry xmlns:app="http://www.w3.org/2007/app"
7204 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
7205 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
7206 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
7207 open.org/ns/cmis/restatom/200908/">
7208   <atom:author>
7209     <atom:name>Al Brown</atom:name>
7210     <atom:uri>http://www.ibm.com</atom:uri>
7211     <atom:email>albertcbrown@us.ibm.com</atom:email>
7212   </atom:author>
7213   <atom:content src="http://cmisexample.oasis-open.org/rep1/1ee01165-98cf-
7214 45bf-9bce-b6238caa5616"/>
7215   <atom:id>urn:uuid:1ee01165-98cf-45bf-9bce-b6238caa5616</atom:id>
7216   <atom:title type="text">Security Policy for Invoices</atom:title>
7217   <atom:updated>2009-10-19T10:09:59.765-07:00</atom:updated>
7218   <atom:link rel="self" href="http://cmisexample.oasis-
7219 open.org/rep1/1ee01165-98cf-45bf-9bce-b6238caa5616"/>
7220   <atom:link rel="edit" href="http://cmisexample.oasis-
7221 open.org/rep1/1ee01165-98cf-45bf-9bce-b6238caa5616"/>
7222   <atom:link type="application/cmis+xml;type=allowableActions"
7223 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7224 href="http://cmisexample.oasis-open.org/rep1/1ee01165-98cf-45bf-9bce-
7225 b6238caa5616/allowableactions"/>
7226

```



```

7227     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7228 href="http://cmisexample.oasis-open.org/rep1/1ee01165-98cf-45bf-9bce-
7229 b6238caa5616/type"/>
7230     <atom:link type="application/atomsvc+xml" rel="service"
7231 href="http://cmisexample.oasis-open.org/rep1//service"/>
7232     <atom:published>2009-10-19T10:09:59.781-07:00</atom:published>
7233     <atom:summary type="html">HTML summary of Entry 1ee01165-98cf-45bf-9bce-
7234 b6238caa5616</atom:summary>
7235     <atom:link type="application/atom+xml;type=feed" rel="up"
7236 href="http://cmisexample.oasis-open.org/rep1/1ee01165-98cf-45bf-9bce-
7237 b6238caa5616/parents"/>
7238     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7239 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
7240 open.org/rep1/1ee01165-98cf-45bf-9bce-b6238caa5616/relationships"/>
7241     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7242 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7243 open.org/rep1/1ee01165-98cf-45bf-9bce-b6238caa5616/acl"/>
7244     <cmisra:object>
7245         <cmis:properties>
7246             <cmis:propertyId localName="rep-cmis:objectId"
7247 propertyDefinitionId="cmis:objectId">
7248                 <cmis:value>1ee01165-98cf-45bf-9bce-b6238caa5616</cmis:value>
7249             </cmis:propertyId>
7250             <cmis:propertyId localName="rep-cmis:objectTypeId"
7251 propertyDefinitionId="cmis:objectTypeId">
7252                 <cmis:value>generalSecurityPolicy</cmis:value>
7253             </cmis:propertyId>
7254             <cmis:propertyString localName="rep-cmis:name"
7255 propertyDefinitionId="cmis:name">
7256                 <cmis:value>Security Policy for Invoices</cmis:value>
7257             </cmis:propertyString>
7258             <cmis:propertyDateTime localName="rep-cmis:creationDate"
7259 propertyDefinitionId="cmis:creationDate">
7260                 <cmis:value>2009-10-19T10:09:59.781-07:00</cmis:value>
7261             </cmis:propertyDateTime>
7262             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
7263 propertyDefinitionId="cmis:lastModificationDate">
7264                 <cmis:value>2009-10-19T10:09:59.781-07:00</cmis:value>
7265             </cmis:propertyDateTime>
7266             <cmis:propertyId localName="rep-cmis:baseTypeId"
7267 propertyDefinitionId="cmis:baseTypeId">
7268                 <cmis:value>cmis:policy</cmis:value>
7269             </cmis:propertyId>
7270             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7271 propertyDefinitionId="cmis:lastModifiedBy">
7272                 <cmis:value>Al Brown</cmis:value>
7273             </cmis:propertyString>
7274             <cmis:propertyString localName="rep-cmis:createdBy"
7275 propertyDefinitionId="cmis:createdBy">
7276                 <cmis:value>Al Brown</cmis:value>
7277             </cmis:propertyString>
7278         </cmis:properties>
7279     </cmisra:object>
7280 </atom:entry>

```

7282

7283 Please also see the example documents included with the schema.

## 72843.9 Feeds

### 72853.9.1 Object Parents Feed

7286 This is the set of parents for a specific object.

7287 CMIS Services:

7288 GET: getObjectParents

7289 Media Type: application/atom+xml;type=feed

7290

7291 Link Relations:

- 7292 • service: Points to service document containing the CMIS repository. The service document  
7293 MUST contain only one workspace element.
- 7294 ○ Media Type: application/atomsvc+xml
- 7295 • via: points to the atom entry of object who's parents are represented by this collection

7296

7297 This feed contains a set of atom entries for each parent of the object that MUST contain:

- 7298 • cmisra:object inside atom:entry
- 7299 • cmisra:relativePathSegment inside atom:entry for the name of the object inside the folder

7300

7301 Example:

```
7302 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7303 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7304 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7305 xmlns:atom="http://www.w3.org/2005/Atom"
7306 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7307 open.org/ns/cmisis/restatom/200908/">
7308   <atom:title type="text">Parent Feed for 1b43e592-a780-4feb-a0dc-
7309 d25de6bf7ab4</atom:title>
7310   <atom:author>
7311     <atom:name>Al Brown</atom:name>
7312     <atom:uri>http://www.ibm.com/</atom:uri>
7313     <atom:email>albertcbrown@us.ibm.com</atom:email>
7314   </atom:author>
7315   <atom:updated>2009-10-19T10:10:00.562-07:00</atom:updated>
7316   <atom:id>urn:uuid:a8fc16cb-d385-4172-b994-da119636d954</atom:id>
7317   <atom:link type="application/atom+xml;type=feed" rel="self"
7318 href="http://cmisexample.oasis-open.org/repl/1b43e592-a780-4feb-a0dc-
7319 d25de6bf7ab4/3"/>
7320   <atom:link type="application/atomsvc+xml" rel="service"
7321 href="http://cmisexample.oasis-open.org/repl//service"/>
7322   <atom:link type="application/atom+xml;type=entry" rel="via"
7323 href="http://cmisexample.oasis-open.org/repl/1b43e592-a780-4feb-a0dc-
7324 d25de6bf7ab4"/>
7325   <atom:link type="application/atom+xml;type=feed" rel="first"
7326 href="http://cmisexample.oasis-open.org/repl/1b43e592-a780-4feb-a0dc-
7327 d25de6bf7ab4/first"/>
7328   <atom:link type="application/atom+xml;type=feed" rel="next"
7329 href="http://cmisexample.oasis-open.org/repl/1b43e592-a780-4feb-a0dc-
7330 d25de6bf7ab4/4"/>
7331   <atom:link type="application/atom+xml;type=feed" rel="previous"
7332 href="http://cmisexample.oasis-open.org/repl/1b43e592-a780-4feb-a0dc-
7333 d25de6bf7ab4/2"/>
```

```

7334     <atom:link type="application/atom+xml;type=feed" rel="last"
7335 href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
7336 d25de6bf7ab4/last"/>
7337     <cmisra:numItems>1</cmisra:numItems>
7338     <atom:entry>
7339         <atom:author>
7340             <atom:name>Al Brown</atom:name>
7341             <atom:uri>http://www.ibm.com/</atom:uri>
7342             <atom:email>albertcbrown@us.ibm.com</atom:email>
7343         </atom:author>
7344         <atom:content src="http://cmisexample.oasis-open.org/rep1/adb70b59-
7345 db45-4da2-81c0-11062fb063bd"/>
7346         <atom:id>urn:uuid:adb70b59-db45-4da2-81c0-11062fb063bd</atom:id>
7347         <atom:title type="text">Customer Folder</atom:title>
7348         <atom:updated>2009-10-19T10:10:00.562-07:00</atom:updated>
7349         <atom:link rel="self" href="http://cmisexample.oasis-
7350 open.org/rep1/adb70b59-db45-4da2-81c0-11062fb063bd"/>
7351         <atom:link rel="edit" href="http://cmisexample.oasis-
7352 open.org/rep1/adb70b59-db45-4da2-81c0-11062fb063bd"/>
7353         <atom:link type="application/cmisacl+xml;type=allowableActions"
7354 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7355 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7356 11062fb063bd/allowableactions"/>
7357         <atom:link type="application/atom+xml;type=entry" rel="describedby"
7358 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7359 11062fb063bd/type"/>
7360         <atom:link type="application/atomsvc+xml" rel="service"
7361 href="http://cmisexample.oasis-open.org/rep1//service"/>
7362         <atom:published>2009-10-19T10:10:00.562-07:00</atom:published>
7363         <atom:summary type="html">HTML summary of Entry adb70b59-db45-4da2-
7364 81c0-11062fb063bd</atom:summary>
7365         <atom:link type="application/atom+xml;type=entry" rel="up"
7366 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7367 11062fb063bd/up"/>
7368         <atom:link type="application/atom+xml;type=feed" rel="down"
7369 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7370 11062fb063bd/children"/>
7371         <atom:link type="application/cmistree+xml" rel="down"
7372 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7373 11062fb063bd/tree"/>
7374         <atom:link type="application/atom+xml;type=feed"
7375 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7376 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7377 11062fb063bd/foldertree"/>
7378         <atom:link type="application/atom+xml;type=feed"
7379 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7380 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7381 11062fb063bd/relationships"/>
7382         <atom:link type="application/atom+xml;type=feed"
7383 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7384 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7385 11062fb063bd/policies"/>
7386         <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7387 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7388 open.org/rep1/adb70b59-db45-4da2-81c0-11062fb063bd/acl"/>
7389         <cmisra:object>
7390             <cmis:properties>
7391                 <cmis:propertyId localName="rep-cmis:objectId"
7392 propertyDefinitionId="cmis:objectId">
7393                     <cmis:value>adb70b59-db45-4da2-81c0-
7394 11062fb063bd</cmis:value>
7395                 </cmis:propertyId>

```

```

7396         <cmis:propertyId localName="rep-cmis:objectId"
7397 propertyDefinitionId="cmis:objectId">
7398         <cmis:value>customer</cmis:value>
7399         </cmis:propertyId>
7400         <cmis:propertyString localName="rep-cmis:name"
7401 propertyDefinitionId="cmis:name">
7402         <cmis:value>Customer Folder</cmis:value>
7403         </cmis:propertyString>
7404         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7405 propertyDefinitionId="cmis:creationDate">
7406         <cmis:value>2009-10-19T10:10:00.562-07:00</cmis:value>
7407         </cmis:propertyDateTime>
7408         <cmis:propertyDateTime localName="rep-
7409 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7410         <cmis:value>2009-10-19T10:10:00.562-07:00</cmis:value>
7411         </cmis:propertyDateTime>
7412         <cmis:propertyId localName="rep-cmis:baseTypeId"
7413 propertyDefinitionId="cmis:baseTypeId">
7414         <cmis:value>cmis:folder</cmis:value>
7415         </cmis:propertyId>
7416         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7417 propertyDefinitionId="cmis:lastModifiedBy">
7418         <cmis:value>Al Brown</cmis:value>
7419         </cmis:propertyString>
7420         <cmis:propertyString localName="rep-cmis:createdBy"
7421 propertyDefinitionId="cmis:createdBy">
7422         <cmis:value>Al Brown</cmis:value>
7423         </cmis:propertyString>
7424         <cmis:propertyId localName="rep-cmis:parentId"
7425 propertyDefinitionId="cmis:parentId">
7426         <cmis:value>adb70b59-db45-4da2-81c0-
7427 11062fb063bdup</cmis:value>
7428         </cmis:propertyId>
7429         </cmis:properties>
7430     </cmisra:object>
7431     <cmisra:relativePathSegment>customer1</cmisra:relativePathSegment>
7432 </atom:entry>
7433 </atom:feed>

```

7434

7435 Please also see the example documents included with the schema.

### 7436 3.9.1.1 GET

7437 The following arguments may be supplied. Please see the domain model for more information:

- 7438 • filter
- 7439 • includeAllowableActions
- 7440 • includeRelationships
- 7441 • renditionFilter
- 7442 • includeRelativePathSegment
- 7443 • If true, then the cmisra:relativePathSegment element will be included in the response.

### 7444 3.9.2 Changes

7445 This is a link relationship described in the service document that contains the changes in the repository in  
7446 the workspace element. The link relation pointing to this feed is [http://docs.oasis-](http://docs.oasis-open.org/ns/cmis/link/200908/changes)  
7447 [open.org/ns/cmis/link/200908/changes](http://docs.oasis-open.org/ns/cmis/link/200908/changes).

7448

7449CMIS Services:

7450 GET: getContentChanges()

7451Media Type: application/atom+xml;type=feed

7452Link Relations:

- 7453 • service: Points to service document containing the CMIS repository. The service document  
7454 MUST contain only one workspace element.
  - 7455 ○ Media Type: application/atomsvc+xml
- 7456 • paging link relations as appropriate: first, next, previous, last
  - 7457 ○ ChangeLogToken is incorporated into the URI specified by the next link relation

7458

7459This feed MUST be ordered from oldest first to newest.

7460

7461If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is  
7462not available, the client should revisit the feed in the future and look for new items and the next link  
7463relation.

7464

7465The following CMIS Atom extension element MAY be included inside the atom feed:

- 7466 • cmisra:numItems

7467

7468The following CMIS Atom extension element MUST be included inside the atom entries:

- 7469 • cmisra:object inside atom:entry

7470

7471Example:

```
7472 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7473 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7474 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7475 xmlns:atom="http://www.w3.org/2005/Atom"
7476 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7477 open.org/ns/cmis/restatom/200908/">
7478   <atom:title type="text">changelog feed</atom:title>
7479   <atom:author>
7480     <atom:name>Al Brown</atom:name>
7481     <atom:uri>http://www.ibm.com/</atom:uri>
7482     <atom:email>albertcbrown@us.ibm.com</atom:email>
7483   </atom:author>
7484   <atom:updated>2009-10-19T10:09:59.953-07:00</atom:updated>
7485   <atom:id>urn:uuid:04393e94-6888-404e-b9de-765cf769b5de</atom:id>
7486   <atom:link type="application/atom+xml;type=feed" rel="self"
7487 href="http://cmisexample.oasis-open.org/repl/oId/3"/>
7488   <atom:link type="application/atomsvc+xml" rel="service"
7489 href="http://cmisexample.oasis-open.org/repl//service"/>
7490   <atom:link type="application/atom+xml;type=feed" rel="first"
7491 href="http://cmisexample.oasis-open.org/repl/oId/first"/>
7492   <atom:link type="application/atom+xml;type=feed" rel="next"
7493 href="http://cmisexample.oasis-open.org/repl/oId/4"/>
7494   <atom:link type="application/atom+xml;type=feed" rel="previous"
7495 href="http://cmisexample.oasis-open.org/repl/oId/2"/>
7496   <atom:link type="application/atom+xml;type=feed" rel="last"
7497 href="http://cmisexample.oasis-open.org/repl/oId/last"/>
7498   <cmisra:numItems>2</cmisra:numItems>
7499   <atom:entry>
7500     <atom:author>
```

```

7501         <atom:name>Al Brown</atom:name>
7502         <atom:uri>http://www.ibm.com/</atom:uri>
7503         <atom:email>albertcbrown@us.ibm.com</atom:email>
7504     </atom:author>
7505     <atom:content src="http://cmisexample.oasis-open.org/rep1/2ff37143-
7506 0d0d-4c1a-948d-bd8311261clf"/>
7507     <atom:id>urn:uuid:2ff37143-0d0d-4c1a-948d-bd8311261clf</atom:id>
7508     <atom:title type="text">CMIS Example Folder as Customer Policy
7509 type</atom:title>
7510     <atom:updated>2009-10-19T10:09:59.953-07:00</atom:updated>
7511     <atom:link rel="self" href="http://cmisexample.oasis-
7512 open.org/rep1/2ff37143-0d0d-4c1a-948d-bd8311261clf"/>
7513     <atom:link rel="edit" href="http://cmisexample.oasis-
7514 open.org/rep1/2ff37143-0d0d-4c1a-948d-bd8311261clf"/>
7515     <atom:link type="application/cmismaxml;type=allowableActions"
7516 rel="http://docs.oasis-open.org/ns/cmismaxml/link/200908/allowableactions"
7517 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7518 bd8311261clf/allowableactions"/>
7519     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7520 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7521 bd8311261clf/type"/>
7522     <atom:link type="application/atomsvc+xml" rel="service"
7523 href="http://cmisexample.oasis-open.org/rep1//service"/>
7524     <atom:published>2009-10-19T10:09:59.953-07:00</atom:published>
7525     <atom:summary type="html">HTML summary of Entry 2ff37143-0d0d-4c1a-
7526 948d-bd8311261clf</atom:summary>
7527     <atom:link type="application/atom+xml;type=entry" rel="up"
7528 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7529 bd8311261clf/up"/>
7530     <atom:link type="application/atom+xml;type=feed" rel="down"
7531 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7532 bd8311261clf/children"/>
7533     <atom:link type="application/cmistree+xml" rel="down"
7534 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7535 bd8311261clf/tree"/>
7536     <atom:link type="application/atom+xml;type=feed"
7537 rel="http://docs.oasis-open.org/ns/cmismaxml/link/200908/foldertree"
7538 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7539 bd8311261clf/foldertree"/>
7540     <atom:link type="application/atom+xml;type=feed"
7541 rel="http://docs.oasis-open.org/ns/cmismaxml/link/200908/relationships"
7542 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7543 bd8311261clf/relationships"/>
7544     <atom:link type="application/atom+xml;type=feed"
7545 rel="http://docs.oasis-open.org/ns/cmismaxml/link/200908/policies"
7546 href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7547 bd8311261clf/policies"/>
7548     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7549 open.org/ns/cmismaxml/link/200908/acl" href="http://cmisexample.oasis-
7550 open.org/rep1/2ff37143-0d0d-4c1a-948d-bd8311261clf/acl"/>
7551     <cmisra:object>
7552         <cmis:properties>
7553             <cmis:propertyId localName="rep-cmis:objectId"
7554 propertyDefinitionId="cmis:objectId">
7555                 <cmis:value>2ff37143-0d0d-4c1a-948d-
7556 bd8311261clf</cmis:value>
7557             </cmis:propertyId>
7558             <cmis:propertyId localName="rep-cmis:objectTypeId"
7559 propertyDefinitionId="cmis:objectTypeId">
7560                 <cmis:value>customerpolicy</cmis:value>
7561             </cmis:propertyId>
7562             <cmis:propertyString localName="rep-cmis:name"
7563 propertyDefinitionId="cmis:name">

```

```

7564         <cmis:value>CMIS Example Folder as Customer Policy
7565     type</cmis:value>
7566     </cmis:propertyString>
7567     <cmis:propertyDateTime localName="rep-cmis:creationDate"
7568     propertyDefinitionId="cmis:creationDate">
7569         <cmis:value>2009-10-19T10:09:59.953-07:00</cmis:value>
7570     </cmis:propertyDateTime>
7571     <cmis:propertyDateTime localName="rep-
7572     cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7573         <cmis:value>2009-10-19T10:09:59.953-07:00</cmis:value>
7574     </cmis:propertyDateTime>
7575     <cmis:propertyId localName="rep-cmis:baseTypeId"
7576     propertyDefinitionId="cmis:baseTypeId">
7577         <cmis:value>cmis:folder</cmis:value>
7578     </cmis:propertyId>
7579     <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7580     propertyDefinitionId="cmis:lastModifiedBy">
7581         <cmis:value>Al Brown</cmis:value>
7582     </cmis:propertyString>
7583     <cmis:propertyString localName="rep-cmis:createdBy"
7584     propertyDefinitionId="cmis:createdBy">
7585         <cmis:value>Al Brown</cmis:value>
7586     </cmis:propertyString>
7587     <cmis:propertyId localName="rep-cmis:parentId"
7588     propertyDefinitionId="cmis:parentId">
7589         <cmis:value>2ff37143-0d0d-4c1a-948d-
7590     bd8311261c1fup</cmis:value>
7591     </cmis:propertyId>
7592 </cmis:properties>
7593 <cmis:changeEventInfo>
7594     <cmis:changeType>updated</cmis:changeType>
7595     <cmis:changeTime>2009-10-19T10:09:59.953-
7596     07:00</cmis:changeTime>
7597 </cmis:changeEventInfo>
7598 </cmisra:object>
7599     <cmisra:pathSegment>policy</cmisra:pathSegment>
7600 </atom:entry>
7601 <atom:entry>
7602     <atom:author>
7603         <atom:name>Al Brown</atom:name>
7604         <atom:uri>http://www.ibm.com/</atom:uri>
7605         <atom:email>albertcbrown@us.ibm.com</atom:email>
7606     </atom:author>
7607     <atom:content src="http://cmisexample.oasis-open.org/rep1/95715cc8-
7608     5f9c-45ce-8c54-4c2d8d7c44fc"/>
7609     <atom:id>urn:uuid:95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc</atom:id>
7610     <atom:title type="text">CMIS Example Document</atom:title>
7611     <atom:updated>2009-10-19T10:09:59.968-07:00</atom:updated>
7612     <atom:link rel="self" href="http://cmisexample.oasis-
7613     open.org/rep1/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc"/>
7614     <atom:link rel="edit" href="http://cmisexample.oasis-
7615     open.org/rep1/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc"/>
7616     <atom:link type="application/cmiv+xml;type=allowableActions"
7617     rel="http://docs.oasis-open.org/ns/cmiv/link/200908/allowableactions"
7618     href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7619     4c2d8d7c44fc/allowableactions"/>
7620     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7621     href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7622     4c2d8d7c44fc/type"/>
7623     <atom:link type="application/atomsvc+xml" rel="service"
7624     href="http://cmisexample.oasis-open.org/rep1//service"/>
7625     <atom:published>2009-10-19T10:09:59.968-07:00</atom:published>

```

```

7626     <atom:summary type="html">HTML summary of Entry 95715cc8-5f9c-45ce-
7627 8c54-4c2d8d7c44fc</atom:summary>
7628     <atom:link rel="edit-media" href="http://cmisexample.oasis-
7629 open.org/rep1/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc/edit-media"/>
7630     <atom:link rel="alternate" href="http://cmisexample.oasis-
7631 open.org/rep1/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc/alternate"/>
7632     <atom:link type="application/atom+xml;type=feed" rel="up"
7633 href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7634 4c2d8d7c44fc/parents"/>
7635     <atom:link type="application/atom+xml;type=feed" rel="version-history"
7636 href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7637 4c2d8d7c44fc/allversions"/>
7638     <atom:link type="application/atom+xml;type=entry" rel="current-
7639 version" href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7640 4c2d8d7c44fc/latest"/>
7641     <atom:link type="application/atom+xml;type=feed"
7642 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7643 href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7644 4c2d8d7c44fc/relationships"/>
7645     <atom:link type="application/atom+xml;type=feed"
7646 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7647 href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7648 4c2d8d7c44fc/policies"/>
7649     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7650 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7651 open.org/rep1/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc/acl"/>
7652     <cmisra:object>
7653       <cmis:properties>
7654         <cmis:propertyId localName="rep-cmis:objectId"
7655 propertyDefinitionId="cmis:objectId">
7656           <cmis:value>95715cc8-5f9c-45ce-8c54-
7657 4c2d8d7c44fc</cmis:value>
7658         </cmis:propertyId>
7659         <cmis:propertyId localName="rep-cmis:objectTypeId"
7660 propertyDefinitionId="cmis:objectTypeId">
7661           <cmis:value>document</cmis:value>
7662         </cmis:propertyId>
7663         <cmis:propertyString localName="rep-cmis:name"
7664 propertyDefinitionId="cmis:name">
7665           <cmis:value>CMIS Example Document</cmis:value>
7666         </cmis:propertyString>
7667         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7668 propertyDefinitionId="cmis:creationDate">
7669           <cmis:value>2009-10-19T10:09:59.968-07:00</cmis:value>
7670         </cmis:propertyDateTime>
7671         <cmis:propertyDateTime localName="rep-
7672 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7673           <cmis:value>2009-10-19T10:09:59.968-07:00</cmis:value>
7674         </cmis:propertyDateTime>
7675         <cmis:propertyId localName="rep-cmis:baseTypeId"
7676 propertyDefinitionId="cmis:baseTypeId">
7677           <cmis:value>cmis:document</cmis:value>
7678         </cmis:propertyId>
7679         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7680 propertyDefinitionId="cmis:lastModifiedBy">
7681           <cmis:value>Al Brown</cmis:value>
7682         </cmis:propertyString>
7683         <cmis:propertyString localName="rep-cmis:createdBy"
7684 propertyDefinitionId="cmis:createdBy">
7685           <cmis:value>Al Brown</cmis:value>
7686         </cmis:propertyString>
7687         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
7688 propertyDefinitionId="cmis:isLatestVersion">

```



```

7689         <cmis:value>true</cmis:value>
7690     </cmis:propertyBoolean>
7691     <cmis:propertyBoolean localName="rep-
7692 cmis:isVersionSeriesCheckedOut"
7693 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7694         <cmis:value>>false</cmis:value>
7695     </cmis:propertyBoolean>
7696     <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
7697 propertyDefinitionId="cmis:isMajorVersion">
7698         <cmis:value>>false</cmis:value>
7699     </cmis:propertyBoolean>
7700     <cmis:propertyBoolean localName="rep-
7701 cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
7702         <cmis:value>>false</cmis:value>
7703     </cmis:propertyBoolean>
7704     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7705 propertyDefinitionId="cmis:isImmutable">
7706         <cmis:value>>false</cmis:value>
7707     </cmis:propertyBoolean>
7708     <cmis:propertyString localName="rep-cmis:checkinComment"
7709 propertyDefinitionId="cmis:checkinComment">
7710         <cmis:value>Checkin comment</cmis:value>
7711     </cmis:propertyString>
7712     <cmis:propertyString localName="rep-cmis:versionLabel"
7713 propertyDefinitionId="cmis:versionLabel">
7714         <cmis:value>0.1</cmis:value>
7715     </cmis:propertyString>
7716     <cmis:propertyString localName="rep-
7717 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
7718         <cmis:value>text/plain</cmis:value>
7719     </cmis:propertyString>
7720     <cmis:propertyString localName="rep-
7721 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
7722         <cmis:value>text.txt</cmis:value>
7723     </cmis:propertyString>
7724     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7725 propertyDefinitionId="cmis:contentStreamLength">
7726         <cmis:value>4234</cmis:value>
7727     </cmis:propertyInteger>
7728     <cmis:propertyString displayName="Keywords for Document"
7729 localName="keywords" propertyDefinitionId="keywords">
7730         <cmis:value>document</cmis:value>
7731         <cmis:value>example</cmis:value>
7732         <cmis:value>sample</cmis:value>
7733         <cmis:value>cmis</cmis:value>
7734     </cmis:propertyString>
7735 </cmis:properties>
7736 <cmis:changeEventInfo>
7737     <cmis:changeType>updated</cmis:changeType>
7738     <cmis:changeTime>2009-10-19T10:09:59.968-
7739 07:00</cmis:changeTime>
7740 </cmis:changeEventInfo>
7741 </cmisra:object>
7742 <cmisra:pathSegment>invoice.pdf</cmisra:pathSegment>
7743 </atom:entry>
7744 </atom:feed>

```

7745

7746 Please also see the example documents included with the schema.

### 7747 3.9.2.1 GET

7748 The following optional parameters may be supplied:

- 7749 • filter
- 7750 • maxItems
- 7751 • includeACL
- 7752 • includePolicyIds
- 7753 • includeProperties
- 7754 • filter

### 7755 3.9.3 Folder Descendants

7756 This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available  
 7757 via the link relation down with the application/cmistree+xml media type. Please see the Hierarchical Atom  
 7758 Entries for more information on format.

7759

7760 If a repository does not support capabilityGetDescendants, then these resources SHOULD NOT be  
 7761 exposed.

7762

7763 CMIS Services:

7764 GET: getDescendants

7765 DELETE: deleteTree

7766 Media Type: application/atom+xml;type=feed

7767

7768 Link Relations:

- 7769 • service: Points to service document containing the CMIS repository. The service document  
 7770 MUST contain only one workspace element.
  - 7771 ○ Media Type: application/atomsvc+xml
- 7772 • via: points to the atom entry of the folder generating this collection
- 7773 • up: points to the atom entry document for this folder's parent
  - 7774 ○ Media Type: application/atom+xml;type=entry
  - 7775 ○ If the root folder, this link relation MUST not be included.
- 7776 • down:
  - 7777 ○ points to the atom feed document representing the children feed for this same folder with  
 7778 media type of application/atom+xml;type=entry
  - 7779 ○ Since this is the descendants, the descendants link SHOULD NOT be included
- 7780 • paging link relations MAY be included as appropriate: first, next, previous, last
  - 7781 ○ Repositories may support these paging link relations on a particular cmisra:children  
 7782 element.
- 7783 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

7784

7785 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7786 • cmisra:numItems

7787

7788 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7789 • cmisra:object inside atom:entry

7790 • cmisra:pathSegment inside atom:entry

7791 • cmisra:children inside atom:entry

7792

7793Example:

```
7794 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7795 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7796 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7797 xmlns:atom="http://www.w3.org/2005/Atom"
7798 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7799 open.org/ns/cmis/restatom/200908/">
7800   <atom:title type="text">Feed for folder1</atom:title>
7801   <atom:author>
7802     <atom:name>Al Brown</atom:name>
7803     <atom:uri>http://www.ibm.com/</atom:uri>
7804     <atom:email>albertcbrown@us.ibm.com</atom:email>
7805   </atom:author>
7806   <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7807   <atom:id>urn:uuid:83638fc5-90e7-4d03-bd04-1a4cc5c5c98e</atom:id>
7808   <atom:link type="application/atom+xml;type=feed" rel="self"
7809 href="http://cmisexample.oasis-open.org/repl/80dfb434-883b-4850-871d-
7810 6524c1ec2a64/3"/>
7811   <atom:link type="application/atomsvc+xml" rel="service"
7812 href="http://cmisexample.oasis-open.org/repl//service"/>
7813   <atom:link type="application/atom+xml;type=entry" rel="via"
7814 href="http://cmisexample.oasis-open.org/repl/80dfb434-883b-4850-871d-
7815 6524c1ec2a64"/>
7816   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7817 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
7818 open.org/repl/80dfb434-883b-4850-871d-6524c1ec2a64/foldertree"/>
7819   <atom:link type="application/atom+xml;type=feed" rel="down"
7820 href="http://cmisexample.oasis-open.org/repl/80dfb434-883b-4850-871d-
7821 6524c1ec2a64/children"/>
7822   <atom:link type="application/atom+xml;type=entry" rel="up"
7823 href="http://cmisexample.oasis-open.org/repl/ba8910f0-0e99-4cda-alf6-
7824 befce0fa20e7"/>
7825   <cmisra:numItems>1</cmisra:numItems>
7826   <atom:entry>
7827     <atom:author>
7828       <atom:name>Al Brown</atom:name>
7829       <atom:uri>http://www.ibm.com/</atom:uri>
7830       <atom:email>albertcbrown@us.ibm.com</atom:email>
7831     </atom:author>
7832     <atom:content src="http://cmisexample.oasis-open.org/repl/c2e574a0-
7833 2d30-4834-b8a3-6333b33181c3"/>
7834     <atom:id>urn:uuid:c2e574a0-2d30-4834-b8a3-6333b33181c3</atom:id>
7835     <atom:title type="text">CMIS Example Folder as Customer
7836 type</atom:title>
7837     <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7838     <atom:link rel="self" href="http://cmisexample.oasis-
7839 open.org/repl/c2e574a0-2d30-4834-b8a3-6333b33181c3"/>
7840     <atom:link rel="edit" href="http://cmisexample.oasis-
7841 open.org/repl/c2e574a0-2d30-4834-b8a3-6333b33181c3"/>
7842     <atom:link type="application/cmis+xml;type=allowableActions"
7843 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7844 href="http://cmisexample.oasis-open.org/repl/c2e574a0-2d30-4834-b8a3-
7845 6333b33181c3/allowableactions"/>
7846     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7847 href="http://cmisexample.oasis-open.org/repl/c2e574a0-2d30-4834-b8a3-
7848 6333b33181c3/type"/>
7849     <atom:link type="application/atomsvc+xml" rel="service"
7850 href="http://cmisexample.oasis-open.org/repl//service"/>
```

```

7851         <atom:published>2009-10-19T10:10:00.203-07:00</atom:published>
7852         <atom:summary type="html">HTML summary of Entry c2e574a0-2d30-4834-
7853 b8a3-6333b33181c3</atom:summary>
7854         <atom:link type="application/atom+xml;type=entry" rel="up"
7855 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7856 6333b33181c3/up"/>
7857         <atom:link type="application/atom+xml;type=feed" rel="down"
7858 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7859 6333b33181c3/children"/>
7860         <atom:link type="application/cmistree+xml" rel="down"
7861 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7862 6333b33181c3/tree"/>
7863         <atom:link type="application/atom+xml;type=feed"
7864 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7865 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7866 6333b33181c3/foldertree"/>
7867         <atom:link type="application/atom+xml;type=feed"
7868 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7869 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7870 6333b33181c3/relationships"/>
7871         <atom:link type="application/atom+xml;type=feed"
7872 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7873 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7874 6333b33181c3/policies"/>
7875         <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7876 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7877 open.org/rep1/c2e574a0-2d30-4834-b8a3-6333b33181c3/acl"/>
7878         <cmisra:object>
7879             <cmis:properties>
7880                 <cmis:propertyId localName="rep-cmis:objectId">
7881 propertyDefinitionId="cmis:objectId">
7882                     <cmis:value>c2e574a0-2d30-4834-b8a3-
7883 6333b33181c3</cmis:value>
7884                 </cmis:propertyId>
7885                 <cmis:propertyId localName="rep-cmis:objectTypeId">
7886 propertyDefinitionId="cmis:objectTypeId">
7887                     <cmis:value>customer</cmis:value>
7888                 </cmis:propertyId>
7889                 <cmis:propertyString localName="rep-cmis:name">
7890 propertyDefinitionId="cmis:name">
7891                     <cmis:value>CMIS Example Folder as Customer
7892 type</cmis:value>
7893                 </cmis:propertyString>
7894                 <cmis:propertyDateTime localName="rep-cmis:creationDate">
7895 propertyDefinitionId="cmis:creationDate">
7896                     <cmis:value>2009-10-19T10:10:00.203-07:00</cmis:value>
7897                 </cmis:propertyDateTime>
7898                 <cmis:propertyDateTime localName="rep-
7899 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7900                     <cmis:value>2009-10-19T10:10:00.203-07:00</cmis:value>
7901                 </cmis:propertyDateTime>
7902                 <cmis:propertyId localName="rep-cmis:baseTypeId">
7903 propertyDefinitionId="cmis:baseTypeId">
7904                     <cmis:value>cmis:folder</cmis:value>
7905                 </cmis:propertyId>
7906                 <cmis:propertyString localName="rep-cmis:lastModifiedBy">
7907 propertyDefinitionId="cmis:lastModifiedBy">
7908                     <cmis:value>Al Brown</cmis:value>
7909                 </cmis:propertyString>
7910                 <cmis:propertyString localName="rep-cmis:createdBy">
7911 propertyDefinitionId="cmis:createdBy">
7912                     <cmis:value>Al Brown</cmis:value>
7913                 </cmis:propertyString>

```

```

7914         <cmis:propertyId localName="rep-cmis:parentId"
7915 propertyDefinitionId="cmis:parentId">
7916         <cmis:value>c2e574a0-2d30-4834-b8a3-
7917 6333b33181c3up</cmis:value>
7918         </cmis:propertyId>
7919     </cmis:properties>
7920 </cmisra:object>
7921 <cmisra:pathSegment>customer</cmisra:pathSegment>
7922 <cmisra:children>
7923     <atom:title type="text">CMIS Example Folder as Customer
7924 type</atom:title>
7925     <atom:author>
7926         <atom:name>Al Brown</atom:name>
7927         <atom:uri>http://www.ibm.com/</atom:uri>
7928         <atom:email>albertcbrown@us.ibm.com</atom:email>
7929     </atom:author>
7930     <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7931     <atom:id>urn:uuid:9550c57d-64cd-4f00-9ab4-5b44903717e8</atom:id>
7932     <atom:link type="application/atom+xml;type=feed" rel="self"
7933 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7934 6333b33181c3/3"/>
7935     <atom:link type="application/atomsvc+xml" rel="service"
7936 href="http://cmisexample.oasis-open.org/rep1//service"/>
7937     <atom:link type="application/atom+xml;type=entry" rel="via"
7938 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7939 6333b33181c3"/>
7940     <atom:link type="application/atom+xml;type=feed"
7941 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7942 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7943 6333b33181c3/foldertree"/>
7944     <atom:link type="application/atom+xml;type=feed" rel="down"
7945 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7946 6333b33181c3/children"/>
7947     <atom:link type="application/atom+xml;type=entry" rel="up"
7948 href="http://cmisexample.oasis-open.org/rep1/80dfb434-883b-4850-871d-
7949 6524c1ec2a64"/>
7950     <cmisra:numItems>1</cmisra:numItems>
7951     <atom:entry>
7952         <atom:author>
7953             <atom:name>Al Brown</atom:name>
7954             <atom:uri>http://www.ibm.com/</atom:uri>
7955             <atom:email>albertcbrown@us.ibm.com</atom:email>
7956         </atom:author>
7957         <atom:content src="http://cmisexample.oasis-
7958 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50"/>
7959         <atom:id>urn:uuid:d4bc9b02-f732-49ce-979a-
7960 da31175a4d50</atom:id>
7961         <atom:title type="text">CMIS Example Doc as Invoice
7962 type</atom:title>
7963         <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7964         <atom:link rel="self" href="http://cmisexample.oasis-
7965 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50"/>
7966         <atom:link rel="edit" href="http://cmisexample.oasis-
7967 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50"/>
7968         <atom:link type="application/cmis+xml;type=allowableActions"
7969 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7970 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7971 da31175a4d50/allowableactions"/>
7972         <atom:link type="application/atom+xml;type=entry"
7973 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-
7974 49ce-979a-da31175a4d50/type"/>
7975         <atom:link type="application/atomsvc+xml" rel="service"
7976 href="http://cmisexample.oasis-open.org/rep1//service"/>

```

```

7977         <atom:published>2009-10-19T10:10:00.203-07:00</atom:published>
7978         <atom:summary type="html">HTML summary of Entry d4bc9b02-
7979 f732-49ce-979a-da31175a4d50</atom:summary>
7980         <atom:link rel="edit-media" href="http://cmisexample.oasis-
7981 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50/edit-media"/>
7982         <atom:link rel="alternate" href="http://cmisexample.oasis-
7983 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50/alternate"/>
7984         <atom:link type="application/atom+xml;type=feed" rel="up"
7985 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7986 da31175a4d50/parents"/>
7987         <atom:link type="application/atom+xml;type=feed" rel="version-
7988 history" href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7989 da31175a4d50/allversions"/>
7990         <atom:link type="application/atom+xml;type=entry"
7991 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-
7992 f732-49ce-979a-da31175a4d50/latest"/>
7993         <atom:link type="application/atom+xml;type=feed"
7994 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7995 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7996 da31175a4d50/relationships"/>
7997         <atom:link type="application/atom+xml;type=feed"
7998 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7999 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
8000 da31175a4d50/policies"/>
8001         <atom:link type="application/cmisacl+xml"
8002 rel="http://docs.oasis-open.org/ns/cmis/link/200908/acl"
8003 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
8004 da31175a4d50/acl"/>
8005         <cmisra:object>
8006             <cmis:properties>
8007                 <cmis:propertyId localName="rep-cmis:objectId"
8008 propertyDefinitionId="cmis:objectId">
8009                     <cmis:value>d4bc9b02-f732-49ce-979a-
8010 da31175a4d50</cmis:value>
8011                 </cmis:propertyId>
8012                 <cmis:propertyId localName="rep-cmis:objectTypeId"
8013 propertyDefinitionId="cmis:objectTypeId">
8014                     <cmis:value>invoice</cmis:value>
8015                 </cmis:propertyId>
8016                 <cmis:propertyString localName="rep-cmis:name"
8017 propertyDefinitionId="cmis:name">
8018                     <cmis:value>CMIS Example Doc as Invoice
8019 type</cmis:value>
8020                 </cmis:propertyString>
8021                 <cmis:propertyDateTime localName="rep-
8022 cmis:creationDate" propertyDefinitionId="cmis:creationDate">
8023                     <cmis:value>2009-10-19T10:10:00.218-
8024 07:00</cmis:value>
8025                 </cmis:propertyDateTime>
8026                 <cmis:propertyDateTime localName="rep-
8027 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
8028                     <cmis:value>2009-10-19T10:10:00.218-
8029 07:00</cmis:value>
8030                 </cmis:propertyDateTime>
8031                 <cmis:propertyId localName="rep-cmis:baseTypeId"
8032 propertyDefinitionId="cmis:baseTypeId">
8033                     <cmis:value>cmis:document</cmis:value>
8034                 </cmis:propertyId>
8035                 <cmis:propertyString localName="rep-
8036 cmis:lastModifiedBy" propertyDefinitionId="cmis:lastModifiedBy">
8037                     <cmis:value>Al Brown</cmis:value>
8038                 </cmis:propertyString>

```

```

8039         <cmis:propertyString localName="rep-cmis:createdBy"
8040 propertyDefinitionId="cmis:createdBy">
8041         <cmis:value>Al Brown</cmis:value>
8042     </cmis:propertyString>
8043     <cmis:propertyBoolean localName="rep-
8044 cmis:isLatestVersion" propertyDefinitionId="cmis:isLatestVersion">
8045         <cmis:value>true</cmis:value>
8046     </cmis:propertyBoolean>
8047     <cmis:propertyBoolean localName="rep-
8048 cmis:isVersionSeriesCheckedOut"
8049 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
8050         <cmis:value>false</cmis:value>
8051     </cmis:propertyBoolean>
8052     <cmis:propertyBoolean localName="rep-
8053 cmis:isMajorVersion" propertyDefinitionId="cmis:isMajorVersion">
8054         <cmis:value>false</cmis:value>
8055     </cmis:propertyBoolean>
8056     <cmis:propertyBoolean localName="rep-
8057 cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
8058         <cmis:value>false</cmis:value>
8059     </cmis:propertyBoolean>
8060     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
8061 propertyDefinitionId="cmis:isImmutable">
8062         <cmis:value>false</cmis:value>
8063     </cmis:propertyBoolean>
8064     <cmis:propertyString localName="rep-
8065 cmis:checkinComment" propertyDefinitionId="cmis:checkinComment">
8066         <cmis:value>Checkin comment</cmis:value>
8067     </cmis:propertyString>
8068     <cmis:propertyString localName="rep-cmis:versionLabel"
8069 propertyDefinitionId="cmis:versionLabel">
8070         <cmis:value>0.1</cmis:value>
8071     </cmis:propertyString>
8072     <cmis:propertyString localName="rep-
8073 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
8074         <cmis:value>text/plain</cmis:value>
8075     </cmis:propertyString>
8076     <cmis:propertyString localName="rep-
8077 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
8078         <cmis:value>text.txt</cmis:value>
8079     </cmis:propertyString>
8080     <cmis:propertyInteger localName="rep-
8081 cmis:contentStreamLength" propertyDefinitionId="cmis:contentStreamLength">
8082         <cmis:value>4234</cmis:value>
8083     </cmis:propertyInteger>
8084     <cmis:propertyString displayName="Keywords for
8085 Document" localName="keywords" propertyDefinitionId="keywords">
8086         <cmis:value>document</cmis:value>
8087         <cmis:value>example</cmis:value>
8088         <cmis:value>sample</cmis:value>
8089         <cmis:value>cmis</cmis:value>
8090     </cmis:propertyString>
8091 </cmis:properties>
8092 </cmisra:object>
8093 <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
8094 </atom:entry>
8095 </cmisra:children>
8096 </atom:entry>
8097 </atom:feed>

```

8098

8099 Please also see the example documents included with the schema.

### 81003.9.3.1 GET

8101The following arguments may be supplied. Please see the domain model for more information:

- 8102     • filter
- 8103     • depth
- 8104     • includeAllowableActions
- 8105     • includeRelationships
- 8106     • renditionFilter
- 8107     • includePathSegment

### 81083.9.3.2 DELETE

8109This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the  
8110domain model for more information:

- 8111     • continueOnFailure
- 8112     • unfileObjects

8113

8114Status Code:

- 8115     • 200 OK if successful. Body contains entity describing the status
- 8116     • 202 Accepted, if accepted but deletion not yet taking place
- 8117     • 204 No Content, if successful with no content
- 8118     • 403 Forbidden, if permission is denied
- 8119     • 401 Unauthorized, if not authenticated
- 8120     • 500 Internal Server Error. The body SHOULD contain an entity describing the status

8121

8122If the delete method does not delete all items, invoking GET with infinite depth on this URI will return the  
8123items not deleted. Subsequent DELETE methods can be invoked on this URI.

8124Note: If the repository does not implement get on this resource, or the canGetDescendants is false, there  
8125is no mechanism to identify the resources that were not removed.

### 81263.9.4 Folder Tree

8127This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link  
8128relation foldertree with media type application/atom+xml;type=feed. Please see the Hierarchical Atom  
8129Entries for more information on format.

8130

8131CMIS Services:

- 8132     GET: getFolderTree
- 8133     DELETE: deleteTree

8134Media Type: application/atom+xml;type=feed

8135

8136Link Relations:

- 8137     • service: Points to service document containing the CMIS repository. The service document  
8138       MUST contain only one workspace element.
- 8139     ○ Media Type: application/atomsvc+xml



- 8140 • via: points to the atom entry of the folder generating this collection
- 8141 • up: points to the atom entry document of this folder's parent
  - 8142 ○ If the root folder, this link relation MUST not be included.
  - 8143 ○ Media Type: application/atom+xml;type=entry
- 8144 • down:
  - 8145 ○ application/atom+xml : Points to the atom feed document representing the children feed
    - 8146 for this same folder
  - 8147 ○ application/cmistree+xml: Points to the descendants feed of the same folder. If a
    - 8148 repository does not support capabilityGetDescendants, then this link SHOULD NOT be
    - 8149 included.
- 8150 • paging link relations MAY be included as appropriate: first, next, previous, last
  - 8151 ○ Repositories may support these paging link relations on a particular cmisra:children
    - 8152 element.

8153

8154 This feed contains a set of atom entries for each sub-folder in the folder.

8155

8156 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8157 • cmisra:numItems

8158

8159 The following CMIS Atom extension element MUST be included inside the atom entries:

- 8160 • cmisra:object inside atom:entry
- 8161 • cmisra:pathSegment inside atom:entry
- 8162 • cmisra:children inside atom:entry

8163

8164 Example:

```

8165 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8166 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
8167 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
8168 xmlns:atom="http://www.w3.org/2005/Atom"
8169 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8170 open.org/ns/cmis/restatom/200908/">
8171   <atom:title type="text">FolderTree Feed of Folder1</atom:title>
8172   <atom:author>
8173     <atom:name>Al Brown</atom:name>
8174     <atom:uri>http://www.ibm.com/</atom:uri>
8175     <atom:email>albertcbrown@us.ibm.com</atom:email>
8176   </atom:author>
8177   <atom:updated>2009-10-19T10:10:00.312-07:00</atom:updated>
8178   <atom:id>urn:uuid:e8627e01-9b28-4e5b-b7b4-d88c05bae4e9</atom:id>
8179   <atom:link type="application/atom+xml;type=feed" rel="self"
8180 href="http://cmisexample.oasis-open.org/repl/3bcbead0-ad9f-4e0c-a2e8-
8181 49d025e062c5/3"/>
8182   <atom:link type="application/atomsvc+xml" rel="service"
8183 href="http://cmisexample.oasis-open.org/repl//service"/>
8184   <atom:link type="application/atom+xml;type=entry" rel="via"
8185 href="http://cmisexample.oasis-open.org/repl/3bcbead0-ad9f-4e0c-a2e8-
8186 49d025e062c5"/>
8187   <atom:link type="application/cmistree+xml" rel="down"
8188 href="http://cmisexample.oasis-open.org/repl/3bcbead0-ad9f-4e0c-a2e8-
8189 49d025e062c5/tree"/>

```

```

8190     <atom:link type="application/atom+xml;type=feed" rel="down"
8191 href="http://cmisexample.oasis-open.org/rep1/3bcbead0-ad9f-4e0c-a2e8-
8192 49d025e062c5/children"/>
8193     <atom:link type="application/atom+xml;type=entry" rel="up"
8194 href="http://cmisexample.oasis-open.org/rep1/cdaf0998-582a-445f-9179-
8195 113a12470b04"/>
8196     <cmisra:numItems>1</cmisra:numItems>
8197     <atom:entry>
8198       <atom:author>
8199         <atom:name>Al Brown</atom:name>
8200         <atom:uri>http://www.ibm.com/</atom:uri>
8201         <atom:email>albertcbrown@us.ibm.com</atom:email>
8202       </atom:author>
8203       <atom:content src="http://cmisexample.oasis-open.org/rep1/7d9c26a2-
8204 adle-43e5-b176-247b64e4f4bc"/>
8205       <atom:id>urn:uuid:7d9c26a2-adle-43e5-b176-247b64e4f4bc</atom:id>
8206       <atom:title type="text">Customer Folder</atom:title>
8207       <atom:updated>2009-10-19T10:10:00.312-07:00</atom:updated>
8208       <atom:link rel="self" href="http://cmisexample.oasis-
8209 open.org/rep1/7d9c26a2-adle-43e5-b176-247b64e4f4bc"/>
8210       <atom:link rel="edit" href="http://cmisexample.oasis-
8211 open.org/rep1/7d9c26a2-adle-43e5-b176-247b64e4f4bc"/>
8212       <atom:link type="application/cmismvc+xml;type=allowableActions"
8213 rel="http://docs.oasis-open.org/ns/cmismvc/link/200908/allowableactions"
8214 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8215 247b64e4f4bc/allowableactions"/>
8216       <atom:link type="application/atom+xml;type=entry" rel="describedby"
8217 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8218 247b64e4f4bc/type"/>
8219       <atom:link type="application/atomsvc+xml" rel="service"
8220 href="http://cmisexample.oasis-open.org/rep1/service"/>
8221       <atom:published>2009-10-19T10:10:00.312-07:00</atom:published>
8222       <atom:summary type="html">HTML summary of Entry 7d9c26a2-adle-43e5-
8223 b176-247b64e4f4bc</atom:summary>
8224       <atom:link type="application/atom+xml;type=entry" rel="up"
8225 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8226 247b64e4f4bc/up"/>
8227       <atom:link type="application/atom+xml;type=feed" rel="down"
8228 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8229 247b64e4f4bc/children"/>
8230       <atom:link type="application/cmistree+xml" rel="down"
8231 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8232 247b64e4f4bc/tree"/>
8233       <atom:link type="application/atom+xml;type=feed"
8234 rel="http://docs.oasis-open.org/ns/cmismvc/link/200908/foldertree"
8235 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8236 247b64e4f4bc/foldertree"/>
8237       <atom:link type="application/atom+xml;type=feed"
8238 rel="http://docs.oasis-open.org/ns/cmismvc/link/200908/relationships"
8239 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8240 247b64e4f4bc/relationships"/>
8241       <atom:link type="application/atom+xml;type=feed"
8242 rel="http://docs.oasis-open.org/ns/cmismvc/link/200908/policies"
8243 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
8244 247b64e4f4bc/policies"/>
8245       <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8246 open.org/ns/cmismvc/link/200908/acl" href="http://cmisexample.oasis-
8247 open.org/rep1/7d9c26a2-adle-43e5-b176-247b64e4f4bc/acl"/>
8248       <cmisra:object>
8249         <cmis:properties>
8250           <cmis:propertyId localName="rep-cmis:objectId"
8251 propertyDefinitionId="cmis:objectId">

```

```

8252         <cmis:value>7d9c26a2-ad1e-43e5-b176-
8253 247b64e4f4bc</cmis:value>
8254     </cmis:propertyId>
8255     <cmis:propertyId localName="rep-cmis:objectTypeId"
8256 propertyDefinitionId="cmis:objectTypeId">
8257         <cmis:value>customer</cmis:value>
8258     </cmis:propertyId>
8259     <cmis:propertyString localName="rep-cmis:name"
8260 propertyDefinitionId="cmis:name">
8261         <cmis:value>Customer Folder</cmis:value>
8262     </cmis:propertyString>
8263     <cmis:propertyDateTime localName="rep-cmis:creationDate"
8264 propertyDefinitionId="cmis:creationDate">
8265         <cmis:value>2009-10-19T10:10:00.328-07:00</cmis:value>
8266     </cmis:propertyDateTime>
8267     <cmis:propertyDateTime localName="rep-
8268 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
8269         <cmis:value>2009-10-19T10:10:00.328-07:00</cmis:value>
8270     </cmis:propertyDateTime>
8271     <cmis:propertyId localName="rep-cmis:baseTypeId"
8272 propertyDefinitionId="cmis:baseTypeId">
8273         <cmis:value>cmis:folder</cmis:value>
8274     </cmis:propertyId>
8275     <cmis:propertyString localName="rep-cmis:lastModifiedBy"
8276 propertyDefinitionId="cmis:lastModifiedBy">
8277         <cmis:value>Al Brown</cmis:value>
8278     </cmis:propertyString>
8279     <cmis:propertyString localName="rep-cmis:createdBy"
8280 propertyDefinitionId="cmis:createdBy">
8281         <cmis:value>Al Brown</cmis:value>
8282     </cmis:propertyString>
8283     <cmis:propertyId localName="rep-cmis:parentId"
8284 propertyDefinitionId="cmis:parentId">
8285         <cmis:value>7d9c26a2-ad1e-43e5-b176-
8286 247b64e4f4bc</cmis:value>
8287     </cmis:propertyId>
8288 </cmis:properties>
8289 </cmisra:object>
8290 <cmisra:pathSegment>customer</cmisra:pathSegment>
8291 </atom:entry>
8292 </atom:feed>

```

8293

8294 Please also see the example documents included with the schema.

### 8295 3.9.4.1 GET

8296 The following arguments may be supplied. Please see the domain model for more information:

- 8297 • filter
- 8298 • depth
- 8299 • includeAllowableActions
- 8300 • includeRelationships
- 8301 • renditionFilter

### 8302 3.9.4.2 DELETE

8303 This is the same as DELETE on Folder Descendants. Please see that section.

### 83043.9.5 AllVersions Feed

8305 This is a feed comprised of all the versions of the given document.

8306 CMIS Services:

8307 GET: getAllVersions

8308 DELETE: deleteAllVersions

8309 Media Type: application/atom+xml;type=feed

8310

8311 The feed SHOULD contain the newest versions at the beginning of the feed.

8312

8313 Link Relations:

8314 • service: Points to service document containing the CMIS repository. The service document  
8315 MUST contain only one workspace element.

8316 ○ Media Type: application/atomsvc+xml

8317 • via: points to the atom entry of the resource generating this collection

8318 • paging link relations as appropriate: first, next, previous, last

8319

8320 This feed contains a set of atom entries for each version in the version series

8321 • cmisra:object inside atom:entry

8322 • cmisra:children inside atom:entry if atom:entry represents a CMIS Folder

8323

8324 Example:

```
8325 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8326 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
8327 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
8328 xmlns:atom="http://www.w3.org/2005/Atom"
8329 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8330 open.org/ns/cmis/restatom/200908/">
8331   <atom:title type="text">AllVersions for Document c3141e18-fb48-4742-a6f8-
8332 adef86dea79a</atom:title>
8333   <atom:author>
8334     <atom:name>Al Brown</atom:name>
8335     <atom:uri>http://www.ibm.com/</atom:uri>
8336     <atom:email>albertcbrown@us.ibm.com</atom:email>
8337   </atom:author>
8338   <atom:updated>2009-10-19T10:09:59.828-07:00</atom:updated>
8339   <atom:id>urn:uuid:309c322f-1e10-4306-85ba-98f1870c5882</atom:id>
8340   <atom:link type="application/atom+xml;type=feed" rel="self"
8341 href="http://cmisexample.oasis-open.org/repl/c3141e18-fb48-4742-a6f8-
8342 adef86dea79a/3"/>
8343   <atom:link type="application/atomsvc+xml" rel="service"
8344 href="http://cmisexample.oasis-open.org/repl//service"/>
8345   <atom:link type="application/atom+xml;type=entry" rel="via"
8346 href="http://cmisexample.oasis-open.org/repl/c3141e18-fb48-4742-a6f8-
8347 adef86dea79a"/>
8348   <cmisra:numItems>1</cmisra:numItems>
8349   <atom:entry>
8350     <atom:author>
8351       <atom:name>Al Brown</atom:name>
8352       <atom:uri>http://www.ibm.com/</atom:uri>
8353       <atom:email>albertcbrown@us.ibm.com</atom:email>
8354     </atom:author>
```

```

8355     <atom:content src="http://cmisexample.oasis-open.org/rep1/9c7860a7-
8356 be25-439e-b473-8da032402a6d"/>
8357     <atom:id>urn:uuid:9c7860a7-be25-439e-b473-8da032402a6d</atom:id>
8358     <atom:title type="text">Invoice (Version1)</atom:title>
8359     <atom:updated>2009-10-19T10:09:59.828-07:00</atom:updated>
8360     <atom:link rel="self" href="http://cmisexample.oasis-
8361 open.org/rep1/9c7860a7-be25-439e-b473-8da032402a6d"/>
8362     <atom:link rel="edit" href="http://cmisexample.oasis-
8363 open.org/rep1/9c7860a7-be25-439e-b473-8da032402a6d"/>
8364     <atom:link type="application/cmism+xml;type=allowableActions"
8365 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
8366 href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8367 8da032402a6d/allowableactions"/>
8368     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8369 href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8370 8da032402a6d/type"/>
8371     <atom:link type="application/atomsvc+xml" rel="service"
8372 href="http://cmisexample.oasis-open.org/rep1//service"/>
8373     <atom:published>2009-10-19T10:09:59.843-07:00</atom:published>
8374     <atom:summary type="html">HTML summary of Entry 9c7860a7-be25-439e-
8375 b473-8da032402a6d</atom:summary>
8376     <atom:link rel="edit-media" href="http://cmisexample.oasis-
8377 open.org/rep1/9c7860a7-be25-439e-b473-8da032402a6d/edit-media"/>
8378     <atom:link rel="alternate" href="http://cmisexample.oasis-
8379 open.org/rep1/9c7860a7-be25-439e-b473-8da032402a6d/alternate"/>
8380     <atom:link type="application/atom+xml;type=feed" rel="up"
8381 href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8382 8da032402a6d/parents"/>
8383     <atom:link type="application/atom+xml;type=feed" rel="version-history"
8384 href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8385 8da032402a6d/allversions"/>
8386     <atom:link type="application/atom+xml;type=entry" rel="current-
8387 version" href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8388 8da032402a6d/latest"/>
8389     <atom:link type="application/atom+xml;type=feed"
8390 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
8391 href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8392 8da032402a6d/relationships"/>
8393     <atom:link type="application/atom+xml;type=feed"
8394 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
8395 href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8396 8da032402a6d/policies"/>
8397     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8398 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
8399 open.org/rep1/9c7860a7-be25-439e-b473-8da032402a6d/acl"/>
8400     <cmisra:object>
8401       <cmis:properties>
8402         <cmis:propertyId localName="rep-cmis:objectId"
8403 propertyDefinitionId="cmis:objectId">
8404           <cmis:value>9c7860a7-be25-439e-b473-
8405 8da032402a6d</cmis:value>
8406         </cmis:propertyId>
8407       </cmis:properties>
8408     </cmisra:object>
8409   </atom:entry>
8410 </atom:feed>

```

8411

8412Please also see the example documents included with the schema.

### 84133.9.5.1 GET

8414The following arguments may be supplied. Please see the domain model for more information:

- 8415 • filter
- 8416 • includeAllowableActions

### 84173.9.5.2 DELETE

8418 This removes the entire version history of the document.

8419

8420 Success HTTP code: 204

### 84213.9.6 Type Descendants Feed

8422 This is a feed described in the service document that contains all the types under a specific type in the  
8423 repository to a specific depth. If no parent type is specified, then the base types and their descendants  
8424 are returned in the feed which is equivalent to all types in the repository if depth is infinite. The link  
8425 relation is <http://docs.oasis-open.org/ns/cmis/link/200908/typesdescendants>.

8426

8427 Types are nested using the CMIS hierarchy extension. Please see section 3.4.3.2 Hierarchy Navigation  
8428 Internet Draft Link Relations.

8429

8430 CMIS Services:

8431 GET: getTypeDescendants

8432

8433 Media Type: application/atom+xml;type=feed

8434

8435 Link Relations:

- 8436 • service: Points to service document containing the CMIS repository. The service document  
8437 MUST contain only one workspace element.
- 8438 ○ Media Type: application/atomsvc+xml
- 8439 • via: points to the type definition whose descendants represent this feed. This link is not present if  
8440 no parent type is specified.
- 8441 • down: points to the children feed for the same type
- 8442 • up: points to the parent type definition
- 8443 ○ If this is a descendants feed for a base object type, this link is not present.

8444

8445 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8446 • cmisra:numItems

8447

8448 Example:

```
8449 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8450 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
8451 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
8452 xmlns:atom="http://www.w3.org/2005/Atom"
8453 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8454 open.org/ns/cmis/restatom/200908/">
8455   <atom:title type="text">Base Types</atom:title>
8456   <atom:author>
8457     <atom:name>Al Brown</atom:name>
8458     <atom:uri>http://www.ibm.com/</atom:uri>
8459     <atom:email>albertcbrown@us.ibm.com</atom:email>
```

```

8460     </atom:author>
8461     <atom:updated>2009-10-19T10:10:00.656-07:00</atom:updated>
8462     <atom:id>urn:uuid:e1981abc-a130-4cf0-89f2-0fa26f2801fe</atom:id>
8463     <atom:link type="application/atom+xml;type=feed" rel="self"
8464 href="http://cmisexample.oasis-open.org/repl//3"/>
8465     <atom:link type="application/atomsvc+xml" rel="service"
8466 href="http://cmisexample.oasis-open.org/repl//service"/>
8467     <atom:link type="application/atom+xml;type=entry" rel="via"
8468 href="http://cmisexample.oasis-open.org/repl//"/>
8469     <atom:link type="application/atom+xml;type=feed" rel="down"
8470 href="http://cmisexample.oasis-open.org/repl//children"/>
8471     <cmisra:numItems>1</cmisra:numItems>
8472     <atom:entry>
8473         <atom:author>
8474             <atom:name>Al Brown</atom:name>
8475             <atom:uri>http://www.ibm.com/</atom:uri>
8476             <atom:email>albertcbrown@us.ibm.com</atom:email>
8477         </atom:author>
8478         <atom:content>Type Definition for cmis:document</atom:content>
8479         <atom:id>http://cmisexample.oasis-
8480 open.org/repl/type/cmis:document</atom:id>
8481         <atom:link type="application/atom+xml;type=entry" rel="self"
8482 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8483         <atom:link type="application/atomsvc+xml" rel="service"
8484 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8485         <atom:link type="application/atom+xml;type=entry" rel="describedby"
8486 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8487         <atom:link type="application/atom+xml;type=entry" rel="up"
8488 href="http://cmisexample.oasis-open.org/repl/type/cmis:document/parent"/>
8489         <atom:link type="application/atom+xml;type=feed" rel="down"
8490 href="http://cmisexample.oasis-
8491 open.org/repl/type/cmis:document/children/flat"/>
8492         <atom:link type="application/cmistree+xml" rel="down"
8493 href="http://cmisexample.oasis-
8494 open.org/repl/type/cmis:document/children/tree"/>
8495         <atom:published>2009-10-19T10:10:00.656-07:00</atom:published>
8496         <atom:summary type="html">HTML summary of Type Definition
8497 cmis:document</atom:summary>
8498         <atom:title type="text">Type Definition - cmis:document</atom:title>
8499         <atom:updated>2009-10-19T10:10:00.656-07:00</atom:updated>
8500         <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8501         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8502 xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="cmis:document">
8503             <cmis:id>dtcmis:document</cmis:id>
8504             <cmis:localName>myrepname-cmis:document</cmis:localName>
8505             <cmis:localNamespaces xsi:nil="true"/>
8506             <cmis:displayName>cmis:document</cmis:displayName>
8507             <cmis:queryName>cmis:document</cmis:queryName>
8508             <cmis:description>Description for type definition
8509 cmis:document</cmis:description>
8510             <cmis:baseId>cmis:document</cmis:baseId>
8511             <cmis:parentId>parent</cmis:parentId>
8512             <cmis:creatable>true</cmis:creatable>
8513             <cmis:fileable>true</cmis:fileable>
8514             <cmis:queryable>false</cmis:queryable>
8515             <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8516             <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8517         </cmisra:type>
8518         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8519         <cmis:controllableACL>true</cmis:controllableACL>
8520         <cmis:versionable>true</cmis:versionable>
8521         <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8522     </atom:entry>

```

```

8523     <cmisra:children>
8524         <atom:title type="text">Children for Document</atom:title>
8525         <atom:author>
8526             <atom:name>Al Brown</atom:name>
8527             <atom:uri>http://www.ibm.com/</atom:uri>
8528             <atom:email>albertcbrown@us.ibm.com</atom:email>
8529         </atom:author>
8530         <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8531         <atom:id>urn:uuid:5657c90b-b07f-4225-832d-28a5302f8424</atom:id>
8532         <atom:link type="application/atom+xml;type=feed" rel="self"
8533 href="http://cmisexample.oasis-open.org/repl/cmis:document/3"/>
8534         <atom:link type="application/atomsvc+xml" rel="service"
8535 href="http://cmisexample.oasis-open.org/repl//service"/>
8536         <atom:link type="application/atom+xml;type=entry" rel="via"
8537 href="http://cmisexample.oasis-open.org/repl/cmis:document"/>
8538         <atom:link type="application/atom+xml;type=feed" rel="down"
8539 href="http://cmisexample.oasis-open.org/repl/cmis:document/children"/>
8540         <atom:link type="application/atom+xml;type=entry" rel="up"
8541 href="http://cmisexample.oasis-open.org/repl/document"/>
8542         <cmisra:numItems>1</cmisra:numItems>
8543         <atom:entry>
8544             <atom:author>
8545                 <atom:name>Al Brown</atom:name>
8546                 <atom:uri>http://www.ibm.com/</atom:uri>
8547                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8548             </atom:author>
8549             <atom:content>Type Definition for invoice-
8550 document</atom:content>
8551             <atom:id>http://cmisexample.oasis-open.org/repl/type/invoice-
8552 document</atom:id>
8553             <atom:link type="application/atom+xml;type=entry" rel="self"
8554 href="http://cmisexample.oasis-open.org/repl/type/invoice-document"/>
8555             <atom:link type="application/atomsvc+xml" rel="service"
8556 href="http://cmisexample.oasis-open.org/repl/type/invoice-document"/>
8557             <atom:link type="application/atom+xml;type=entry"
8558 rel="describedby" href="http://cmisexample.oasis-
8559 open.org/repl/type/cmis:document"/>
8560             <atom:link type="application/atom+xml;type=entry" rel="up"
8561 href="http://cmisexample.oasis-open.org/repl/type/invoice-document/parent"/>
8562             <atom:link type="application/atom+xml;type=feed" rel="down"
8563 href="http://cmisexample.oasis-open.org/repl/type/invoice-
8564 document/children/flat"/>
8565             <atom:link type="application/cmistree+xml" rel="down"
8566 href="http://cmisexample.oasis-open.org/repl/type/invoice-
8567 document/children/tree"/>
8568             <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8569             <atom:summary type="html">HTML summary of Type Definition
8570 invoice-document</atom:summary>
8571             <atom:title type="text">Type Definition - invoice-
8572 document</atom:title>
8573             <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8574             <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8575             <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8576 instance" xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="invoice-
8577 document">
8578                 <cmis:id>dtinvoice-document</cmis:id>
8579                 <cmis:localName>myrepname-invoice-
8580 document</cmis:localName>
8581                 <cmis:localNamespace xsi:nil="true"/>
8582                 <cmis:displayName>invoice-document</cmis:displayName>
8583                 <cmis:queryName>invoice-document</cmis:queryName>
8584                 <cmis:description>Description for type definition invoice-
8585 document</cmis:description>

```



```

8586         <cmis:baseId>cmis:document</cmis:baseId>
8587         <cmis:parentId>parent</cmis:parentId>
8588         <cmis:creatable>true</cmis:creatable>
8589         <cmis:fileable>true</cmis:fileable>
8590         <cmis:queryable>false</cmis:queryable>
8591         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8592         <cmis:includedInSupertypeQuery>true</cmis:includedInSupert
8593 typeQuery>
8594         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8595         <cmis:controllableACL>true</cmis:controllableACL>
8596         <cmis:versionable>true</cmis:versionable>
8597         <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllo
8598 wed>
8599         </cmisra:type>
8600     </atom:entry>
8601 </cmisra:children>
8602 </atom:entry>
8603 <atom:entry>
8604     <atom:author>
8605         <atom:name>Al Brown</atom:name>
8606         <atom:uri>http://www.ibm.com/</atom:uri>
8607         <atom:email>albertcbrown@us.ibm.com</atom:email>
8608     </atom:author>
8609     <atom:content>Type Definition for cmis:folder</atom:content>
8610     <atom:id>http://cmisexample.oasis-
8611 open.org/repl/type/cmis:folder</atom:id>
8612     <atom:link type="application/atom+xml;type=entry" rel="self"
8613 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8614     <atom:link type="application/atomsvc+xml" rel="service"
8615 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8616     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8617 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8618     <atom:link type="application/atom+xml;type=entry" rel="up"
8619 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/parent"/>
8620     <atom:link type="application/atom+xml;type=feed" rel="down"
8621 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/children/flat"/>
8622     <atom:link type="application/cmistree+xml" rel="down"
8623 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/children/tree"/>
8624     <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8625     <atom:summary type="html">HTML summary of Type Definition
8626 cmis:folder</atom:summary>
8627     <atom:title type="text">Type Definition - cmis:folder</atom:title>
8628     <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8629     <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8630     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8631 xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="cmis:folder">
8632         <cmis:id>dtcmis:folder</cmis:id>
8633         <cmis:localName>myrepname-cmis:folder</cmis:localName>
8634         <cmis:localNamespace xsi:nil="true"/>
8635         <cmis:displayName>cmis:folder</cmis:displayName>
8636         <cmis:queryName>cmis:folder</cmis:queryName>
8637         <cmis:description>Description for type definition
8638 cmis:folder</cmis:description>
8639         <cmis:baseId>cmis:folder</cmis:baseId>
8640         <cmis:parentId>parent</cmis:parentId>
8641         <cmis:creatable>true</cmis:creatable>
8642         <cmis:fileable>true</cmis:fileable>
8643         <cmis:queryable>false</cmis:queryable>
8644         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8645         <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8646     >
8647         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8648         <cmis:controllableACL>true</cmis:controllableACL>

```

```

8649     </cmisra:type>
8650     <cmisra:children>
8651         <atom:title type="text">Children for Folder</atom:title>
8652         <atom:author>
8653             <atom:name>Al Brown</atom:name>
8654             <atom:uri>http://www.ibm.com/</atom:uri>
8655             <atom:email>albertcbrown@us.ibm.com</atom:email>
8656         </atom:author>
8657         <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8658         <atom:id>urn:uuid:7b556496-2a09-40ca-b2a5-3e4e5596fb89</atom:id>
8659         <atom:link type="application/atom+xml;type=feed" rel="self"
8660 href="http://cmisexample.oasis-open.org/repl/cmisis:folder/3"/>
8661         <atom:link type="application/atomsvc+xml" rel="service"
8662 href="http://cmisexample.oasis-open.org/repl//service"/>
8663         <atom:link type="application/atom+xml;type=entry" rel="via"
8664 href="http://cmisexample.oasis-open.org/repl/cmisis:folder"/>
8665         <atom:link type="application/atom+xml;type=feed" rel="down"
8666 href="http://cmisexample.oasis-open.org/repl/cmisis:folder/children"/>
8667         <atom:link type="application/atom+xml;type=entry" rel="up"
8668 href="http://cmisexample.oasis-open.org/repl/cmisis:folder"/>
8669         <cmisra:numItems>1</cmisra:numItems>
8670         <atom:entry>
8671             <atom:author>
8672                 <atom:name>Al Brown</atom:name>
8673                 <atom:uri>http://www.ibm.com/</atom:uri>
8674                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8675             </atom:author>
8676             <atom:content>Type Definition for customer-
8677 folder</atom:content>
8678             <atom:id>http://cmisexample.oasis-open.org/repl/type/customer-
8679 folder</atom:id>
8680             <atom:link type="application/atom+xml;type=entry" rel="self"
8681 href="http://cmisexample.oasis-open.org/repl/type/customer-folder"/>
8682             <atom:link type="application/atomsvc+xml" rel="service"
8683 href="http://cmisexample.oasis-open.org/repl/type/customer-folder"/>
8684             <atom:link type="application/atom+xml;type=entry"
8685 rel="describedby" href="http://cmisexample.oasis-
8686 open.org/repl/type/cmisis:folder"/>
8687             <atom:link type="application/atom+xml;type=entry" rel="up"
8688 href="http://cmisexample.oasis-open.org/repl/type/customer-folder/parent"/>
8689             <atom:link type="application/atom+xml;type=feed" rel="down"
8690 href="http://cmisexample.oasis-open.org/repl/type/customer-
8691 folder/children/flat"/>
8692             <atom:link type="application/cmistree+xml" rel="down"
8693 href="http://cmisexample.oasis-open.org/repl/type/customer-
8694 folder/children/tree"/>
8695             <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8696             <atom:summary type="html">HTML summary of Type Definition
8697 customer-folder</atom:summary>
8698             <atom:title type="text">Type Definition - customer-
8699 folder</atom:title>
8700             <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8701             <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8702             <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8703 instance" xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="customer-
8704 folder">
8705                 <cmis:id>dtcustomer-folder</cmis:id>
8706                 <cmis:localName>myrepname-customer-folder</cmis:localName>
8707                 <cmis:localNamespace xsi:nil="true"/>
8708                 <cmis:displayName>customer-folder</cmis:displayName>
8709                 <cmis:queryName>customer-folder</cmis:queryName>
8710                 <cmis:description>Description for type definition
8711 customer-folder</cmis:description>

```

```

8712         <cmis:baseId>cmis:folder</cmis:baseId>
8713         <cmis:parentId>parent</cmis:parentId>
8714         <cmis:creatable>true</cmis:creatable>
8715         <cmis:fileable>true</cmis:fileable>
8716         <cmis:queryable>false</cmis:queryable>
8717         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8718         <cmis:includedInSupertypeQuery>true</cmis:includedInSupert
8719 typeQuery>
8720         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8721         <cmis:controllableACL>true</cmis:controllableACL>
8722     </cmisra:type>
8723 </atom:entry>
8724 </cmisra:children>
8725 </atom:entry>
8726 <atom:entry>
8727     <atom:author>
8728         <atom:name>Al Brown</atom:name>
8729         <atom:uri>http://www.ibm.com/</atom:uri>
8730         <atom:email>albertcbrown@us.ibm.com</atom:email>
8731     </atom:author>
8732     <atom:content>Type Definition for cmis:relationship</atom:content>
8733     <atom:id>http://cmisexample.oasis-
8734 open.org/repl/type/cmis:relationship</atom:id>
8735     <atom:link type="application/atom+xml;type=entry" rel="self"
8736 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8737     <atom:link type="application/atomsvc+xml" rel="service"
8738 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8739     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8740 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8741     <atom:link type="application/atom+xml;type=entry" rel="up"
8742 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship/parent"/>
8743     <atom:link type="application/atom+xml;type=feed" rel="down"
8744 href="http://cmisexample.oasis-
8745 open.org/repl/type/cmis:relationship/children/flat"/>
8746     <atom:link type="application/cmistree+xml" rel="down"
8747 href="http://cmisexample.oasis-
8748 open.org/repl/type/cmis:relationship/children/tree"/>
8749     <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8750     <atom:summary type="html">HTML summary of Type Definition
8751 cmis:relationship</atom:summary>
8752     <atom:title type="text">Type Definition -
8753 cmis:relationship</atom:title>
8754     <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8755     <app:edited>2009-10-19T10:10:00.687-07:00</app:edited>
8756     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8757 xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8758 cmisra:id="cmis:relationship">
8759         <cmis:id>dtcmis:relationship</cmis:id>
8760         <cmis:localName>myrepname-cmis:relationship</cmis:localName>
8761         <cmis:localNamespace xsi:nil="true"/>
8762         <cmis:displayName>cmis:relationship</cmis:displayName>
8763         <cmis:queryName>cmis:relationship</cmis:queryName>
8764         <cmis:description>Description for type definition
8765 cmis:relationship</cmis:description>
8766         <cmis:baseId>cmis:relationship</cmis:baseId>
8767         <cmis:parentId>parent</cmis:parentId>
8768         <cmis:creatable>true</cmis:creatable>
8769         <cmis:fileable>false</cmis:fileable>
8770         <cmis:queryable>false</cmis:queryable>
8771         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8772         <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery
8773 >
8774         <cmis:controllablePolicy>true</cmis:controllablePolicy>

```

```

8775         <cmis:controllableACL>true</cmis:controllableACL>
8776         <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8777         <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8778         <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8779     </cmisra:type>
8780     <cmisra:children>
8781         <atom:title type="text">Children for Relationship</atom:title>
8782         <atom:author>
8783             <atom:name>Al Brown</atom:name>
8784             <atom:uri>http://www.ibm.com/</atom:uri>
8785             <atom:email>albertcbrown@us.ibm.com</atom:email>
8786         </atom:author>
8787         <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8788         <atom:id>urn:uuid:1bc33a0d-27ac-4976-9cab-58ea28e3ff6c</atom:id>
8789         <atom:link type="application/atom+xml;type=feed" rel="self"
8790 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/3"/>
8791         <atom:link type="application/atomsvc+xml" rel="service"
8792 href="http://cmisexample.oasis-open.org/repl//service"/>
8793         <atom:link type="application/atom+xml;type=entry" rel="via"
8794 href="http://cmisexample.oasis-open.org/repl/cmis:relationship"/>
8795         <atom:link type="application/atom+xml;type=feed" rel="down"
8796 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/children"/>
8797         <atom:link type="application/atom+xml;type=entry" rel="up"
8798 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8799         <cmisra:numItems>1</cmisra:numItems>
8800     <atom:entry>
8801         <atom:author>
8802             <atom:name>Al Brown</atom:name>
8803             <atom:uri>http://www.ibm.com/</atom:uri>
8804             <atom:email>albertcbrown@us.ibm.com</atom:email>
8805         </atom:author>
8806         <atom:content>Type Definition for customer-
8807 relationship</atom:content>
8808         <atom:id>http://cmisexample.oasis-open.org/repl/type/customer-
8809 relationship</atom:id>
8810         <atom:link type="application/atom+xml;type=entry" rel="self"
8811 href="http://cmisexample.oasis-open.org/repl/type/customer-relationship"/>
8812         <atom:link type="application/atomsvc+xml" rel="service"
8813 href="http://cmisexample.oasis-open.org/repl/type/customer-relationship"/>
8814         <atom:link type="application/atom+xml;type=entry"
8815 rel="describedby" href="http://cmisexample.oasis-
8816 open.org/repl/type/cmis:relationship"/>
8817         <atom:link type="application/atom+xml;type=entry" rel="up"
8818 href="http://cmisexample.oasis-open.org/repl/type/customer-
8819 relationship/parent"/>
8820         <atom:link type="application/atom+xml;type=feed" rel="down"
8821 href="http://cmisexample.oasis-open.org/repl/type/customer-
8822 relationship/children/flat"/>
8823         <atom:link type="application/cmistree+xml" rel="down"
8824 href="http://cmisexample.oasis-open.org/repl/type/customer-
8825 relationship/children/tree"/>
8826         <atom:published>2009-10-19T10:10:00.687-07:00</atom:published>
8827         <atom:summary type="html">HTML summary of Type Definition
8828 customer-relationship</atom:summary>
8829         <atom:title type="text">Type Definition - customer-
8830 relationship</atom:title>
8831         <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8832         <app:edited>2009-10-19T10:10:00.687-07:00</app:edited>
8833         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8834 instance" xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8835 cmisra:id="customer-relationship">
8836         <cmis:id>dtcustomer-relationship</cmis:id>

```

```

8837         <cmis:localName>myrepname-customer-
8838 relationship</cmis:localName>
8839         <cmis:localNamespace xsi:nil="true"/>
8840         <cmis:displayName>customer-relationship</cmis:displayName>
8841         <cmis:queryName>customer-relationship</cmis:queryName>
8842         <cmis:description>Description for type definition
8843 customer-relationship</cmis:description>
8844         <cmis:baseId>cmis:relationship</cmis:baseId>
8845         <cmis:parentId>parent</cmis:parentId>
8846         <cmis:creatable>true</cmis:creatable>
8847         <cmis:fileable>false</cmis:fileable>
8848         <cmis:queryable>false</cmis:queryable>
8849         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8850         <cmis:includedInSupertypeQuery>true</cmis:includedInSupert
8851 ypeQuery>
8852         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8853         <cmis:controllableACL>true</cmis:controllableACL>
8854         <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8855         <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourc
8856 eTypes>
8857         <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes
8858 >
8859     </cmisra:type>
8860   </atom:entry>
8861 </cmisra:children>
8862 </atom:entry>
8863 <atom:entry>
8864   <atom:author>
8865     <atom:name>Al Brown</atom:name>
8866     <atom:uri>http://www.ibm.com/</atom:uri>
8867     <atom:email>albertcbrown@us.ibm.com</atom:email>
8868   </atom:author>
8869   <atom:content>Type Definition for cmis:policy</atom:content>
8870   <atom:id>http://cmisexample.oasis-
8871 open.org/repl/type/cmis:policy</atom:id>
8872   <atom:link type="application/atom+xml;type=entry" rel="self"
8873 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8874   <atom:link type="application/atomsvc+xml" rel="service"
8875 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8876   <atom:link type="application/atom+xml;type=entry" rel="describedby"
8877 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8878   <atom:link type="application/atom+xml;type=entry" rel="up"
8879 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/parent"/>
8880   <atom:link type="application/atom+xml;type=feed" rel="down"
8881 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/flat"/>
8882   <atom:link type="application/cmistree+xml" rel="down"
8883 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/tree"/>
8884   <atom:published>2009-10-19T10:10:00.687-07:00</atom:published>
8885   <atom:summary type="html">HTML summary of Type Definition
8886 cmis:policy</atom:summary>
8887   <atom:title type="text">Type Definition - cmis:policy</atom:title>
8888   <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8889   <app:edited>2009-10-19T10:10:00.687-07:00</app:edited>
8890   <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8891 xsi:type="cmis:cmisTypePolicyDefinitionType" cmisra:id="cmis:policy">
8892     <cmis:id>dtcmis:policy</cmis:id>
8893     <cmis:localName>myrepname-cmis:policy</cmis:localName>
8894     <cmis:localNamespace xsi:nil="true"/>
8895     <cmis:displayName>cmis:policy</cmis:displayName>
8896     <cmis:queryName>cmis:policy</cmis:queryName>
8897     <cmis:description>Description for type definition
8898 cmis:policy</cmis:description>
8899     <cmis:baseId>cmis:policy</cmis:baseId>

```

```

8900         <cmis:parentId>parent</cmis:parentId>
8901         <cmis:creatable>true</cmis:creatable>
8902         <cmis:fileable>false</cmis:fileable>
8903         <cmis:queryable>false</cmis:queryable>
8904         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8905         <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8906     >
8907         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8908         <cmis:controllableACL>true</cmis:controllableACL>
8909     </cmisra:type>
8910 </atom:entry>
8911 </atom:feed>

```

8912

8913 Please also see the example documents included with the schema.

### 8914 3.9.6.1 GET

8915 The following arguments may be supplied. Please see the domain model for more information:

- 8916 • includePropertyDefinitions
- 8917 • depth

## 8918 3.10 Resources

### 8919 3.10.1 Type Entry

8920 This represents a type definition in the repository. This is enclosed as an atom entry

8921

8922 CMIS Services:

8923 GET: getTypeDefinition

8924 Media Type: application/atom+xml;type=entry

8925

8926 Link Relations:

- 8927 • service: Points to service document containing the CMIS repository. The service document
  - 8928 MUST contain only one workspace element.
- 8929 • Media Type: application/atomsvc+xml
- 8930 • up: Points to the parent type as atom entry if applicable
- 8931 • down: Points to the children of this type as atom feed if applicable
  - 8932 • (Children) Media Type: application/atom+xml;type=feed points to the atom feed
    - 8933 document representing the children feed for this same type
  - 8934 • (Descendants) Media Type: application/cmistree+xml points to the atom feed document
    - 8935 representing the descendants feed for this same type
- 8936 • describedby: Points to the type definition atom entry of the base type of this type definition.

8937

8938 The following CMIS Atom extension element MUST be included inside the atom entry:

- 8939 • cmisra:type

### 8940 3.10.1.1 GET

8941 There are no optional arguments for this resource.

8942

#### 8943Request:

```
8944 GET /obj/0010d88d-d006-4d50-aad0-c8f61f80f273 HTTP/1.1
8945 Host: example.org
8946
```

#### 8947

#### 8948Response:

```
8949 HTTP/1.1 200 Ok
8950 Date: Mon, 19 Oct 2009 10:10:01 -0700
8951 Content-Length: 2995
8952 Content-Type: application/atom+xml;type=entry
8953 Location: http://cmisexample.oasis-open.org/repl/cmis:document
8954
8955 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8956 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8957 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8958 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
8959 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8960 open.org/ns/cmis/restatom/200908/">
8961   <atom:author>
8962     <atom:name>Al Brown</atom:name>
8963     <atom:uri>http://www.ibm.com/</atom:uri>
8964     <atom:email>albertcbrown@us.ibm.com</atom:email>
8965   </atom:author>
8966   <atom:content>Type Definition for cmis:document</atom:content>
8967   <atom:id>http://cmisexample.oasis-
8968 open.org/repl/type/cmis:document</atom:id>
8969   <atom:link type="application/atom+xml;type=entry" rel="self"
8970 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8971   <atom:link type="application/atomsvc+xml" rel="service"
8972 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8973   <atom:link type="application/atom+xml;type=entry" rel="describedby"
8974 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8975   <atom:link type="application/atom+xml;type=entry" rel="up"
8976 href="http://cmisexample.oasis-open.org/repl/type/cmis:document/parent"/>
8977   <atom:link type="application/atom+xml;type=feed" rel="down"
8978 href="http://cmisexample.oasis-
8979 open.org/repl/type/cmis:document/children/flat"/>
8980   <atom:link type="application/cmistree+xml" rel="down"
8981 href="http://cmisexample.oasis-
8982 open.org/repl/type/cmis:document/children/tree"/>
8983   <atom:published>2009-10-19T10:10:01.015-07:00</atom:published>
8984   <atom:summary type="html">HTML summary of Type Definition
8985 cmis:document</atom:summary>
8986   <atom:title type="text">Type Definition - cmis:document</atom:title>
8987   <atom:updated>2009-10-19T10:10:01.015-07:00</atom:updated>
8988   <app:edited>2009-10-19T10:10:01.015-07:00</app:edited>
8989   <cmisra:type xsi:type="cmis:cmisTypeDocumentDefinitionType"
8990 cmisra:id="cmis:document" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8991 instance">
8992     <cmis:id>dtcmis:document</cmis:id>
8993     <cmis:localName>myrepname-cmis:document</cmis:localName>
8994     <cmis:localNamespace xsi:nil="true"/>
8995     <cmis:displayName>cmis:document</cmis:displayName>
8996     <cmis:queryName>cmis:document</cmis:queryName>
8997     <cmis:description>Description for type definition
8998 cmis:document</cmis:description>
8999     <cmis:baseId>cmis:document</cmis:baseId>
9000     <cmis:parentId>parent</cmis:parentId>
9001     <cmis:creatable>true</cmis:creatable>
9002     <cmis:fileable>true</cmis:fileable>
9003
```

```

9004     <cmis:queryable>false</cmis:queryable>
9005     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
9006     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
9007     <cmis:controllablePolicy>true</cmis:controllablePolicy>
9008     <cmis:controllableACL>true</cmis:controllableACL>
9009     <cmis:versionable>true</cmis:versionable>
9010     <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
9011   </cmisra:type>
9012 </atom:entry>

```

9013

9014

9015 Please also see the example documents included with the schema.

9016

## 9017 **3.10.2 Document Entry**

9018 This is a CMIS Document instance.

9019

9020 CMIS Services:

9021 GET: getObject, getObjectOfLatestVersion (getObject)

9022 PUT: updateProperties

9023 DELETE: deleteObject

9024 Media Type: application/atom+xml;type=entry

9025

9026 Link Relations:

- 9027 • self: Points to an URI that returns the atom entry for this document. Please see Atom for more  
9028 information.
- 9029 • edit: Points to an URI that accepts PUT of atom entry. Please see AtomPub for more information.
- 9030 • service: Points to service document containing the CMIS repository. The service document  
9031 MUST contain only one workspace element.
- 9032 • Media Type: application/atomsvc+xml
- 9033 • up: Points to the atom feed containing the set of parents. If there is only one parent, the  
9034 repository MAY point this link relation directly to the atom entry of the parent.
- 9035 • version-history: Points to atom feed containing the versions of this document
- 9036 • If the document is not versionable, this link relation may not be on the resource
- 9037 • current-version: Points to the latest version of the document
- 9038 • Uses query parameter 'returnVersion' and enumReturnVersion
- 9039 • If this version is the current-version, this link relation may not be on the resource
- 9040 • edit-media:
- 9041 • Same as setContentStream. Allows updating the content stream on this document
- 9042 • Please see AtomPub for more information
- 9043 • working-copy: Points to the private working copy if it exists.
- 9044 • describedby: Points to the type definition as an atom entry for the type of this document entry.
- 9045 • alternate: this is used to identify the renditions available for the specified object. Please see the  
9046 Renditions section.



- 9047 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions  
9048 document for this object.
- 9049 • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for  
9050 this object
- 9051 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 9052 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object  
9053
- 9054 The following CMIS Atom extension element MUST be included inside the atom entry:
- 9055 • cmisra:object  
9056

### 9057 3.10.2.1 GET

9058 The following arguments may be supplied. Please see the domain model for more information:

- 9059 • returnVersion
  - 9060 ○ Used to differentiate between getObject() and getObjectOfLatestVersion().
  - 9061 ○ valid values are described by the schema element cmisra:enumReturnVersion
  - 9062 ○ If not specified, return the version specified by the URI
- 9063 • includeAllowableActions
- 9064 • includeRelationships
- 9065 • includePolicyIds
- 9066 • includeACL
- 9067 • filter
- 9068 • renditionFilter
  - 9069 ○ If not specified, renditions will not be included.

9070

9071

9072 Request:

```
9073 GET /obj/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08?filter=cmis:objectId HTTP/1.1
9074 Host: example.org
9075
```

9076

9077 Response:

```
9078 HTTP/1.1 200 Ok
9079 Date: Mon, 19 Oct 2009 10:10:00 -0700
9080 Content-Length: 3403
9081 Content-Type: application/atom+xml;type=entry
9082 Location: /obj/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08?filter=cmis:objectId
9083
9084 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9085 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9086 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9087 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9088 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9089 open.org/ns/cmisis/restatom/200908/">
9090   <atom:author>
9091     <atom:name>Al Brown</atom:name>
9092     <atom:uri>http://www.ibm.com/</atom:uri>
```

```

9094         <atom:email>albertcbrown@us.ibm.com</atom:email>
9095     </atom:author>
9096     <atom:content src="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-
9097 49b9-acf7-94851ff7ad08"/>
9098     <atom:id>urn:uuid:8a7b8f30-3e9f-49b9-acf7-94851ff7ad08</atom:id>
9099     <atom:title type="text">Invoice</atom:title>
9100     <atom:updated>2009-10-19T10:10:00.843-07:00</atom:updated>
9101     <atom:link rel="self" href="http://cmisexample.oasis-
9102 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08"/>
9103     <atom:link rel="edit" href="http://cmisexample.oasis-
9104 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08"/>
9105     <atom:link type="application/cmism+xml;type=allowableActions"
9106 rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
9107 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
9108 94851ff7ad08/allowableactions"/>
9109     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9110 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
9111 94851ff7ad08/type"/>
9112     <atom:link type="application/atomsvc+xml" rel="service"
9113 href="http://cmisexample.oasis-open.org/rep1//service"/>
9114     <atom:published>2009-10-19T10:10:00.843-07:00</atom:published>
9115     <atom:summary type="html">HTML summary of Entry 8a7b8f30-3e9f-49b9-acf7-
9116 94851ff7ad08</atom:summary>
9117     <atom:link rel="edit-media" href="http://cmisexample.oasis-
9118 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/edit-media"/>
9119     <atom:link rel="alternate" href="http://cmisexample.oasis-
9120 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/alternate"/>
9121     <atom:link type="application/atom+xml;type=feed" rel="up"
9122 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
9123 94851ff7ad08/parents"/>
9124     <atom:link type="application/atom+xml;type=feed" rel="version-history"
9125 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
9126 94851ff7ad08/allversions"/>
9127     <atom:link type="application/atom+xml;type=entry" rel="current-version"
9128 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
9129 94851ff7ad08/latest"/>
9130     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9131 open.org/ns/cmism/link/200908/relationships" href="http://cmisexample.oasis-
9132 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/relationships"/>
9133     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9134 open.org/ns/cmism/link/200908/policies" href="http://cmisexample.oasis-
9135 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/policies"/>
9136     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9137 open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
9138 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/acl"/>
9139     <cmisra:object>
9140         <cmis:properties>
9141             <cmis:propertyId localName="rep-cmis:objectId"
9142 propertyDefinitionId="cmis:objectId">
9143                 <cmis:value>8a7b8f30-3e9f-49b9-acf7-94851ff7ad08</cmis:value>
9144             </cmis:propertyId>
9145         </cmis:properties>
9146     </cmisra:object>
9147 </atom:entry>

```

9148

9149

9150Please also see the example documents included with the schema.

9151

### 91523.10.2.2 PUT

9153 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9154 properties are not included, the repository SHOULD NOT modify them.

9155

9156 The server SHOULD respond with:

- 9157 • HTTP Status Code 200
- 9158 • Response Body containing the updated atom entry

9159

### 91603.10.2.3 DELETE

9161 This removes the document.

9162 Success HTTP code: 204

### 91633.10.3 Document Private Working Copy (PWC) Entry

9164 This is the private working copy of the document (checkedout version of document)

9165 CMIS Services:

9166 GET: getObject

9167 PUT: updateProperties or checkIn

9168 DELETE: cancelCheckOut

9169 Media Type: application/atom+xml;type=entry

9170

9171 Link relations:

- 9172 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9173 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more  
9174 information.
- 9175 • service: Points to service document containing the CMIS repository. The service document  
9176 MUST contain only one workspace element.
  - 9177 ○ Media Type: application/atomsvc+xml
- 9178 • up: Points to the atom feed containing the set of parents. If there is only one parent, the  
9179 repository MAY point this link relation directly to the atom entry of the parent.
- 9180 • version-history
  - 9181 ○ Points to an URI that returns the feed associated with the version history
- 9182 • edit-media
  - 9183 ○ Same as setContentStream. Allows updating the content stream on this document
  - 9184 ○ Please see AtomPub for more information
- 9185 • via: atom entry that created this private working copy
- 9186 • describedby: Points to the type definition as an atom entry for the type of this PWC entry.
- 9187 • alternate: this is used to identify the renditions available for the specified object. Please see the  
9188 Renditions section.
- 9189 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions  
9190 document for this object.
- 9191 • <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for  
9192 this object

- 9193 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 9194 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

9195

9196The following element MUST be included inside the atom entry:

- 9197 • cmisra:object

9198

### 91993.10.3.1 GET

9200The following arguments may be supplied. Please see the domain model for more information:

- 9201 • filter
- 9202 • includeAllowableActions
- 9203 • includeRelationships
- 9204 • renditionFilter
- 9205     o If not specified, renditions will not be included.

9206

9207Request:

```
9208 GET /obj/3ca7d7d0-1c98-4c38-9141-604d287fb881?filter=cmis:objectId HTTP/1.1
9209 Host: example.org
```

9210

9211

9212Response:

```
9213 HTTP/1.1 200 Ok
9214 Date: Mon, 19 Oct 2009 10:10:00 -0700
9215 Content-Length: 3564
9216 Content-Type: application/atom+xml;type=entry
9217 Location: /obj/3ca7d7d0-1c98-4c38-9141-604d287fb881?filter=cmis:objectId
9218
9219 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9220 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9221 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9222 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9223 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9224 open.org/ns/cmis/restatom/200908/">
9225   <atom:author>
9226     <atom:name>Al Brown</atom:name>
9227     <atom:uri>http://www.ibm.com/</atom:uri>
9228     <atom:email>albertcbrown@us.ibm.com</atom:email>
9229   </atom:author>
9230   <atom:content src="http://cmisexample.oasis-open.org/repl/3ca7d7d0-1c98-
9231 4c38-9141-604d287fb881"/>
9232   <atom:id>urn:uuid:3ca7d7d0-1c98-4c38-9141-604d287fb881</atom:id>
9233   <atom:title type="text">Invoice</atom:title>
9234   <atom:updated>2009-10-19T10:10:00.968-07:00</atom:updated>
9235   <atom:link rel="self" href="http://cmisexample.oasis-
9236 open.org/repl/3ca7d7d0-1c98-4c38-9141-604d287fb881"/>
9237   <atom:link rel="edit" href="http://cmisexample.oasis-
9238 open.org/repl/3ca7d7d0-1c98-4c38-9141-604d287fb881"/>
9239   <atom:link type="application/cmis+xml;type=allowableActions"
9240 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9241 href="http://cmisexample.oasis-open.org/repl/3ca7d7d0-1c98-4c38-9141-
9242 604d287fb881/allowableactions"/>
9243
```

```

9244     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9245 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
9246 604d287fb881/type"/>
9247     <atom:link type="application/atomsvc+xml" rel="service"
9248 href="http://cmisexample.oasis-open.org/rep1//service"/>
9249     <atom:published>2009-10-19T10:10:00.968-07:00</atom:published>
9250     <atom:summary type="html">HTML summary of Entry 3ca7d7d0-1c98-4c38-9141-
9251 604d287fb881</atom:summary>
9252     <atom:link rel="edit-media" href="http://cmisexample.oasis-
9253 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/edit-media"/>
9254     <atom:link rel="alternate" href="http://cmisexample.oasis-
9255 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/alternate"/>
9256     <atom:link type="application/atom+xml;type=feed" rel="up"
9257 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
9258 604d287fb881/parents"/>
9259     <atom:link type="application/atom+xml;type=feed" rel="version-history"
9260 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
9261 604d287fb881/allversions"/>
9262     <atom:link type="application/atom+xml;type=entry" rel="current-version"
9263 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
9264 604d287fb881/latest"/>
9265     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9266 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9267 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/relationships"/>
9268     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9269 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9270 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/policies"/>
9271     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9272 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9273 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/acl"/>
9274     <atom:link type="application/atom+xml;type=feed" rel="working-copy"
9275 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
9276 604d287fb881/pwc"/>
9277     <cmisra:object>
9278       <cmis:properties>
9279         <cmis:propertyId localName="rep-cmis:objectId"
9280 propertyDefinitionId="cmis:objectId">
9281           <cmis:value>3ca7d7d0-1c98-4c38-9141-604d287fb881</cmis:value>
9282         </cmis:propertyId>
9283       </cmis:properties>
9284     </cmisra:object>
9285   </atom:entry>

```

9286

9287

9288 Please also see the example documents included with the schema.

9289

### 9290 3.10.3.2 PUT

9291 This does a replacement of the atom entry with the atom entry document specified. If modifiable  
 9292 properties (whencheckedout or readwrite) are not included, the repository SHOULD NOT modify them.

9293

9294 The following arguments may be supplied. Please see the domain model for more information:

9295 • checkinComment

9296 • major

9297 • checkin

9298 ○ Used to differentiate between updateProperties() or checkin() services. If TRUE, execute  
 9299 checkin service.

9300

9301The server SHOULD respond with:

- 9302     • HTTP Status Code 200
- 9303     • Location header of the resource (if changed via checkin)
- 9304     • Response Body containing the updated atom entry

### 9305**3.10.3.3 DELETE**

9306This removes the document entry, in this case, cancels the check out. The PWC will be removed.

9307

9308Success HTTP code: 204

### 9309**3.10.4 Folder Entry**

9310This is a CMIS Folder instance. The properties of a folder map onto the feed tag.

9311CMIS Services:

9312         GET: getObject

9313         PUT: updateProperties

9314         DELETE: deleteObject (this is deletion of the folder only and not any contained objects)

9315Media Type: application/atom+xml;type=entry

9316

9317Link Relations:

- 9318     • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9319edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
- 9320     • service: Points to service document containing the CMIS repository. The service document
- 9321         MUST contain only one workspace element.
- 9322
  - Media Type: application/atomsvc+xml
- 9323     • describedby: Points to the type definition as an atom entry for the type of this folder entry.
- 9324     • down: Points to the children of this folder if they exist
- 9325
  - application/atom+xml : Points to the atom feed document representing the children feed
  - 9326             for this same folder
  - 9327             ○ application/cmistree+xml: Points to the descendants feed of the same folder
- 9328     • up: Points to the atom entry for the parent
- 9329
  - If the root folder, this link will not be present
- 9330     • alternate: this is used to identify the renditions available for the specified object. Please see the
- 9331         Renditions section.
- 9332     • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions
- 9333         document for this object.
- 9334     • <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for
- 9335         this object
- 9336     • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 9337     • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object
- 9338     • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

9339

9340The following CMIS Atom extension element MUST be included inside the atom entry:

- 9341 • cmisra:object

9342

### 9343**3.10.4.1 GET**

9344The following arguments may be supplied. Please see the domain model for more information:

- 9345 • filter
- 9346 • includeAllowableActions
- 9347 • includeRelationships
- 9348 • renditionFilter
- 9349 ○ If not specified, renditions will not be included.

9350

9351Request:

```
9352 GET /obj/24cbe125-1ffc-4459-8089-0e6f0a500150?filter=cmis:objectId HTTP/1.1
9353 Host: example.org
```

9354

9355

9356Response:

```
9357 HTTP/1.1 200 Ok
9358 Date: Mon, 19 Oct 2009 10:10:00 -0700
9359 Content-Length: 3332
9360 Content-Type: application/atom+xml;type=entry
9361 Location: /obj/24cbe125-1ffc-4459-8089-0e6f0a500150?filter=cmis:objectId
9362
9363 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9364 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9365 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9366 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9367 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9368 open.org/ns/cmisis/restatom/200908/">
9369   <atom:author>
9370     <atom:name>Al Brown</atom:name>
9371     <atom:uri>http://www.ibm.com/</atom:uri>
9372     <atom:email>albertcbrown@us.ibm.com</atom:email>
9373   </atom:author>
9374   <atom:content src="http://cmisexample.oasis-open.org/repl/24cbe125-1ffc-
9375 4459-8089-0e6f0a500150"/>
9376   <atom:id>urn:uuid:24cbe125-1ffc-4459-8089-0e6f0a500150</atom:id>
9377   <atom:title type="text">Customer Folder</atom:title>
9378   <atom:updated>2009-10-19T10:10:00.875-07:00</atom:updated>
9379   <atom:link rel="self" href="http://cmisexample.oasis-
9380 open.org/repl/24cbe125-1ffc-4459-8089-0e6f0a500150"/>
9381   <atom:link rel="edit" href="http://cmisexample.oasis-
9382 open.org/repl/24cbe125-1ffc-4459-8089-0e6f0a500150"/>
9383   <atom:link type="application/cmisis+xml;type=allowableActions"
9384 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9385 href="http://cmisexample.oasis-open.org/repl/24cbe125-1ffc-4459-8089-
9386 0e6f0a500150/allowableactions"/>
9387   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9388 href="http://cmisexample.oasis-open.org/repl/24cbe125-1ffc-4459-8089-
9389 0e6f0a500150/type"/>
9390   <atom:link type="application/atomsvc+xml" rel="service"
9391 href="http://cmisexample.oasis-open.org/repl//service"/>
9392   <atom:published>2009-10-19T10:10:00.875-07:00</atom:published>
```

```

9394     <atom:summary type="html">HTML summary of Entry 24cbe125-1ffc-4459-8089-
9395 0e6f0a500150</atom:summary>
9396     <atom:link type="application/atom+xml;type=entry" rel="up"
9397 href="http://cmisexample.oasis-open.org/repl/24cbe125-1ffc-4459-8089-
9398 0e6f0a500150/up"/>
9399     <atom:link type="application/atom+xml;type=feed" rel="down"
9400 href="http://cmisexample.oasis-open.org/repl/24cbe125-1ffc-4459-8089-
9401 0e6f0a500150/children"/>
9402     <atom:link type="application/cmistree+xml" rel="down"
9403 href="http://cmisexample.oasis-open.org/repl/24cbe125-1ffc-4459-8089-
9404 0e6f0a500150/tree"/>
9405     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9406 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
9407 open.org/repl/24cbe125-1ffc-4459-8089-0e6f0a500150/foldertree"/>
9408     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9409 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9410 open.org/repl/24cbe125-1ffc-4459-8089-0e6f0a500150/relationships"/>
9411     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9412 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9413 open.org/repl/24cbe125-1ffc-4459-8089-0e6f0a500150/policies"/>
9414     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9415 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9416 open.org/repl/24cbe125-1ffc-4459-8089-0e6f0a500150/acl"/>
9417     <cmisra:object>
9418       <cmis:properties>
9419         <cmis:propertyId localName="rep-cmis:objectId"
9420 propertyDefinitionId="cmis:objectId">
9421           <cmis:value>24cbe125-1ffc-4459-8089-0e6f0a500150</cmis:value>
9422         </cmis:propertyId>
9423       </cmis:properties>
9424     </cmisra:object>
9425 </atom:entry>

```

9426

9427

9428 Please also see the example documents included with the schema.

9429

### 9430 3.10.4.2 PUT

9431 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9432 properties are not included, the repository SHOULD NOT modify them.

9433

9434 The server SHOULD respond with:

- 9435
- HTTP Status Code 200
- 9436
- Response Body containing the updated atom entry

9437

### 9438 3.10.4.3 DELETE

9439 This removes the object (folder) from the repository.

9440 Success HTTP code: 204

## 9441 3.10.5 Relationship Entry

9442 This is a CMIS relationship instance. These objects are exposed via 'relationships' link type.

9443 CMIS Services:



9444 GET: getObject  
 9445 PUT: updateProperties  
 9446 DELETE: deleteObject  
 9447Media Type: application/atom+xml;type=entry  
 9448  
 9449Link Relations:

- 9450 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9451 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more  
 9452 information.
- 9453 • service: Points to service document containing the CMIS repository. The service document  
 9454 MUST contain only one workspace element.
  - 9455 ○ Media Type: application/atomsvc+xml
- 9456 • describedby: Points to the type definition as an atom entry for the type of this relationship entry.
- 9457 • <http://docs.oasis-open.org/ns/cmis/link/200908/target>
- 9458 • <http://docs.oasis-open.org/ns/cmis/link/200908/source>
- 9459 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions  
 9460 document for this object.
- 9461 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 9462 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

9463  
 9464The following element MUST be included inside the atom entry:

- 9465 • cmisra:object

9466

### 9467**3.10.5.1 GET**

9468The following arguments may be supplied. Please see the domain model for more information:

- 9469 • filter
- 9470 • includeAllowableActions

9471

9472Request:

```
9473 GET /obj/38af30f5-020d-4274-b25c-0821b6db899b?filter=cmis:objectId HTTP/1.1
9474 Host: example.org
9475
```

9476

9477Response:

```
9478 HTTP/1.1 200 Ok
9479 Date: Mon, 19 Oct 2009 10:10:01 -0700
9480 Content-Length: 2861
9481 Content-Type: application/atom+xml;type=entry
9482 Location: /obj/38af30f5-020d-4274-b25c-0821b6db899b?filter=cmis:objectId
9483
9484 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9485
```

```

9486 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9487 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9488 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9489 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9490 open.org/ns/cmis/restatom/200908/">
9491   <atom:author>
9492     <atom:name>Al Brown</atom:name>
9493     <atom:uri>http://www.ibm.com</atom:uri>
9494     <atom:email>albertcbrown@us.ibm.com</atom:email>
9495   </atom:author>
9496   <atom:content src="http://cmisexample.oasis-open.org/repl/38af30f5-020d-
9497 4274-b25c-0821b6db899b"/>
9498   <atom:id>urn:uuid:38af30f5-020d-4274-b25c-0821b6db899b</atom:id>
9499   <atom:title type="text">Customer Relationship</atom:title>
9500   <atom:updated>2009-10-19T10:10:01.000-07:00</atom:updated>
9501   <atom:link rel="self" href="http://cmisexample.oasis-
9502 open.org/repl/38af30f5-020d-4274-b25c-0821b6db899b"/>
9503   <atom:link rel="edit" href="http://cmisexample.oasis-
9504 open.org/repl/38af30f5-020d-4274-b25c-0821b6db899b"/>
9505   <atom:link type="application/cmis+xml;type=allowableActions"
9506 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9507 href="http://cmisexample.oasis-open.org/repl/38af30f5-020d-4274-b25c-
9508 0821b6db899b/allowableactions"/>
9509   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9510 href="http://cmisexample.oasis-open.org/repl/38af30f5-020d-4274-b25c-
9511 0821b6db899b/type"/>
9512   <atom:link type="application/atomsvc+xml" rel="service"
9513 href="http://cmisexample.oasis-open.org/repl//service"/>
9514   <atom:published>2009-10-19T10:10:01.000-07:00</atom:published>
9515   <atom:summary type="html">HTML summary of Entry 38af30f5-020d-4274-b25c-
9516 0821b6db899b</atom:summary>
9517   <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9518 open.org/ns/cmis/link/200908/source" href="http://cmisexample.oasis-
9519 open.org/repl/38af30f5-020d-4274-b25c-0821b6db899b/source"/>
9520   <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9521 open.org/ns/cmis/link/200908/target" href="http://cmisexample.oasis-
9522 open.org/repl/38af30f5-020d-4274-b25c-0821b6db899b/target"/>
9523   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9524 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9525 open.org/repl/38af30f5-020d-4274-b25c-0821b6db899b/policies"/>
9526   <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9527 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9528 open.org/repl/38af30f5-020d-4274-b25c-0821b6db899b/acl"/>
9529   <cmisra:object>
9530     <cmis:properties>
9531       <cmis:propertyId localName="rep-cmis:objectId"
9532 propertyDefinitionId="cmis:objectId">
9533         <cmis:value>38af30f5-020d-4274-b25c-0821b6db899b</cmis:value>
9534       </cmis:propertyId>
9535     </cmis:properties>
9536   </cmisra:object>
9537 </atom:entry>

```

9538

9539

9540 Please also see the example documents included with the schema.

9541

## 9542 3.10.5.2 PUT

9543 This does a replacement of the atom entry with the atom entry document specified. If readwrite

9544 properties are not included, the repository SHOULD NOT modify them.

9545

9546The server SHOULD respond with:

- 9547     • HTTP Status Code 200
- 9548     • Response Body containing the updated atom entry

9549

### 95503.10.5.3 DELETE

9551This removes the relationship entry.

9552Successful HTTP code: 204

### 95533.10.6 Policy Entry

9554This is a CMIS policy instance.

9555CMIS Services:

- 9556       GET: getObject
- 9557       PUT: updateProperties
- 9558       DELETE: deleteObject or removePolicy

9559Media Type: application/atom+xml;type=entry

9560

9561Link Relations:

- 9562     • self
- 9563     • edit
- 9564     • service: Points to service document containing the CMIS repository. The service document  
9565       MUST contain only one workspace element.
- 9566
  - Media Type: application/atomsvc+xml
- 9567     • describedby: Points to the type definition as an atom entry for the type of this policy entry.
- 9568     • alternate: this is used to identify the renditions available for the specified object. Please see the  
9569       Renditions section.
- 9570     • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions  
9571       document for this object.
- 9572     • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 9573     • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

9574

9575The following element MUST be included inside the atom entry:

- 9576     • `cmisra:object`

9577

### 95783.10.6.1 GET

9579The following arguments may be supplied. Please see the domain model for more information:

- 9580     • filter
- 9581     • includeAllowableActions
- 9582     • includeRelationships
- 9583     • renditionFilter

9584           ○ If not specified, renditions will not be included.

9585

9586Request:

```
9587 GET /obj/408fb453-4a18-4103-b495-e84a2a2761a0?filter=cmis:objectId HTTP/1.1
9588 Host: example.org
```

9589

9590

9591Response:

```
9592 HTTP/1.1 200 Ok
9593 Date: Mon, 19 Oct 2009 10:10:00 -0700
9594 Content-Length: 2608
9595 Content-Type: application/atom+xml;type=entry
9596 Location: /obj/408fb453-4a18-4103-b495-e84a2a2761a0?filter=cmis:objectId
9597
9598
9599 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9600 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9601 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9602 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9603 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9604 open.org/ns/cmis/restatom/200908/">
9605   <atom:author>
9606     <atom:name>Al Brown</atom:name>
9607     <atom:uri>http://www.ibm.com/</atom:uri>
9608     <atom:email>albertcbrown@us.ibm.com</atom:email>
9609   </atom:author>
9610   <atom:content src="http://cmisexample.oasis-open.org/rep1/408fb453-4a18-
9611 4103-b495-e84a2a2761a0"/>
9612   <atom:id>urn:uuid:408fb453-4a18-4103-b495-e84a2a2761a0</atom:id>
9613   <atom:title type="text">Security Policy</atom:title>
9614   <atom:updated>2009-10-19T10:10:00.937-07:00</atom:updated>
9615   <atom:link rel="self" href="http://cmisexample.oasis-
9616 open.org/rep1/408fb453-4a18-4103-b495-e84a2a2761a0"/>
9617   <atom:link rel="edit" href="http://cmisexample.oasis-
9618 open.org/rep1/408fb453-4a18-4103-b495-e84a2a2761a0"/>
9619   <atom:link type="application/cmism+xml;type=allowableActions"
9620 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9621 href="http://cmisexample.oasis-open.org/rep1/408fb453-4a18-4103-b495-
9622 e84a2a2761a0/allowableactions"/>
9623   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9624 href="http://cmisexample.oasis-open.org/rep1/408fb453-4a18-4103-b495-
9625 e84a2a2761a0/type"/>
9626   <atom:link type="application/atomsvc+xml" rel="service"
9627 href="http://cmisexample.oasis-open.org/rep1//service"/>
9628   <atom:published>2009-10-19T10:10:00.937-07:00</atom:published>
9629   <atom:summary type="html">HTML summary of Entry 408fb453-4a18-4103-b495-
9630 e84a2a2761a0</atom:summary>
9631   <atom:link type="application/atom+xml;type=feed" rel="up"
9632 href="http://cmisexample.oasis-open.org/rep1/408fb453-4a18-4103-b495-
9633 e84a2a2761a0/parents"/>
9634   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9635 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9636 open.org/rep1/408fb453-4a18-4103-b495-e84a2a2761a0/relationships"/>
9637   <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9638 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9639 open.org/rep1/408fb453-4a18-4103-b495-e84a2a2761a0/acl"/>
9640   <cmisra:object>
9641     <cmis:properties>
9642       <cmis:propertyId localName="rep-cmis:objectId"
9643 propertyDefinitionId="cmis:objectId">
```

```
<cmis:value>408fb453-4a18-4103-b495-e84a2a2761a0</cmis:value>
</cmis:propertyId>
</cmis:properties>
</cmisra:object>
</atom:entry>
```

9650

9651 Please also see the example documents included with the schema.

9652

### 9653 **3.10.6.2 PUT**

9654 This does a replacement of the atom entry with the atom entry document specified. If read/write  
9655 properties are not included, the repository SHOULD NOT modify them.

9656

9657 The server SHOULD respond with:

- 9658     • HTTP Status Code 200  
9659     • Response Body containing the updated atom entry

9660

### 9661 **3.10.6.3 DELETE**

9662 This removes the policy entry. If this policy entry was discovered through a policy collection on an object,  
9663 then removePolicy() is performed rather than deleteObject() on the policy itself.

9664

9665 Success HTTP code: 204

## 9666 **3.10.7 Content Stream**

9667 This is the content stream portion of the document object.

9668 CMIS Services:

- 9669     GET: getContentStream  
9670     PUT: setContentStream  
9671     DELETE: deleteContentStream

9672 Media Type: Mime/Type of resource (mime type of content stream on document)

### 9673 **3.10.7.1 GET**

9674 This returns the content stream.

9675

9676 It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED  
9677 that HTTP compression is also supported.

9678

9679 Please see RFC2616 for more information on HTTP Range requests.

### 9680 **3.10.7.2 PUT**

9681 This does a replacement of the content stream.

9682

9683 The following optional arguments may be supplied. Please see the domain model for more information:

- 9684     •    overwriteFlag.
- 9685             ○    If not specified, this defaults to 'true' in this binding and behaves consistent with
- 9686                     AtomPub.
- 9687
- 9688Success HTTP code: 201
- 9689
- 9690Returns headers:
- 9691     •    Content-Location: URI for content stream
- 9692     •    Location: URI for content stream

### 9693**3.10.7.3 DELETE**

9694This removes the content stream.

### 9695**3.10.8 ACL Resource**

9696CMIS Services:

9697         GET: getACL

9698         PUT: applyACL

9699

9700Media Type: application/cmisacl+xml

9701

#### 9702**3.10.8.1 GET**

9703This returns the CMIS ACL for a specified object. The client will follow the link on the atom entry to get

9704the CMIS ACL for that object.

9705

9706Request:

```
9707 GET /objacl/59fe373b-0348-493e-baaf-21089b596f0a HTTP/1.1
9708 Host: example.org
9709
```

9710

9711Response:

```
9712 HTTP/1.1 200 Ok
9713 Date: Mon, 19 Oct 2009 10:10:00 -0700
9714 Content-Length: 758
9715 Content-Type: application/cmisacl+xml
9716 Location: /objacl/59fe373b-0348-493e-baaf-21089b596f0a
9717
9718
9719 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9720 <cmis:acl xmlns:app="http://www.w3.org/2007/app"
9721 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9722 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9723 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9724 open.org/ns/cmis/restatom/200908/">
9725   <cmis:permission>
9726     <cmis:principal>
9727       <cmis:principalId>Al Brown</cmis:principalId>
9728     </cmis:principal>
9729     <cmis:permission>cmis:read</cmis:permission>
9730     <cmis:permission>cmis:write</cmis:permission>
```

```
9731         <cmis:permission>cmis:all</cmis:permission>
9732         <cmis:permission>publish</cmis:permission>
9733         <cmis:direct>true</cmis:direct>
9734     </cmis:permission>
9735 </cmis:acl>
```

9736  
9737

9738 Please also see the example documents included with the schema.

---

## 9739**4 Web Services Binding**

### 9740**4.1 Overview**

9741All services and operations defined in part I of the CMIS specification are presented in this Web Services  
9742binding.

9743The WSDL for these services reference two XSD documents. One defines elements for the primary data  
9744types of documents, folders, relationships and policies as well as collections of these types of objects.

9745The second XSD defines the message formats for each of the CMIS services; the messages often refer to  
9746the data types defined in the first XSD schema. The WSDL presents exactly the abstract services defined  
9747in the services section of Part I of the CMIS specification.

9748The normative CMIS Web Services binding is defined by the WSDL and XSD as well as the details given  
9749here in this part of the CMIS specification except the examples.

### 9750**4.1.1 WS-I**

9751A CMIS Web Services binding MUST comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

### 9752**4.1.2 Authentication**

9753A CMIS Web Services binding SHOULD support WS-Security 1.1 for Username Token Profile 1.1 and  
9754MAY also support other authentication mechanisms. A CMIS repository MAY grant access to all or a  
9755subset of the CMIS services to unauthenticated clients.

### 9756**4.1.3 Content Transfer**

9757A CMIS Web Services binding SHOULD support MTOM content transfers. It MUST accept content that is  
9758base64 encoded.

### 9759**4.1.4 Reporting Errors**

9760Services MUST report errors via SOAP faults. The CMIS-Messaging.xsd defines a basic fault structure  
9761that includes an error code and an error message and the WSDL for each service defines specific  
9762messages that have the basic fault format.

## 9763**4.2 Web Services Binding Mapping**

9764The Domain Model in Part I of the CMIS specification defines all services, operations, parameters and  
9765objects of CMIS. The Web Services binding is an exact one-to-one mapping of this definition with small  
9766exceptions that are explained in the next section. Operations and parameters are named exactly after  
9767their counterparts in Part I. All rules and exceptions defined in Part I apply to the Web Services binding.  
9768Optional parameters and optional return values are not set if they are missing or their value is NULL.

## 9769**4.3 Additions to Part I**

### 9770**4.3.1 updateProperties and checkIn Semantics**

9771This binding supports partial properties updates. All properties passed to updateProperties or checkIn will  
9772be updated to their new values. Properties that are passed without a value will be set to their default value  
9773or un-set if no default value is defined. All others property values remain untouched.



## 9774**4.3.2 Content Ranges**

9775This binding supports the retrieval of content ranges. The operation `getContentStream` accepts two  
9776optional parameters:

- 9777
  - **Integer offset:** The first byte of the content to retrieve. Default value is 0.
- 9778
  - **Integer length:** The length of the range in bytes. Default value is the size of the content
- 9779
  - minus the offset.

9780

9781If the **offset** value is greater than the size of the content the repository SHOULD throw a `constraint`  
9782exception.

9783If **offset** + **length** is greater than the size of the content the repository should deliver the content from the  
9784offset to the end of the content.

9785

## 9786**4.3.3 Extensions**

9787On all input messages and some output messages exists an element called extension. This element is  
9788used to provide vendor or repository-specific information between client and server.

9789All of the types referenced by the schema also support `xs:any` for vendor or repository-specific  
9790information.

## 9791**4.3.4 Web Services Specific Structures**

9792This binding requires specific structures that are not part of the general CMIS schema.

9793Please also see the example request and response documents included with the schema.

### 9794**4.3.4.1 cmisFaultType and cmisFault**

9795`cmisFaultType` and `cmisFault` SHOULD be used to generate SOAP faults. See 4.1.4 Reporting  
9796Errors.

### 9797**4.3.4.2 cmisRepositoryEntryType**

9798`cmisRepositoryEntryType` is the return structure of `getRepositories`. It contains the id and the name  
9799of a repository.

### 9800**4.3.4.3 cmisTypeContainer**

9801`cmisTypeContainer` is the return structure of `getTypeDescendants`. It holds a type hierarchy.

### 9802**4.3.4.4 cmisTypeDefinitionListType**

9803`cmisTypeDefinitionListType` is the return structure of `getTypeChildren`. It contains a list of types,  
9804the `hasMoreItems` flag and the `numItem` element.

### 9805**4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and** 9806**cmisObjectInFolderContainerType**

9807`cmisObjectInFolderType` holds, in addition to a `cmisObjectType` object, a path segment string. It  
9808is used in all operations that support the `includePathSegments` parameter.

9809`cmisObjectParentsType` is similar but has a relative path segment string instead of a path segment.

9810For details about path segments and relative path segments see Part I section 2.5.3 Paths.

9811`cmisObjectInFolderContainerType` contains a folder hierarchy.

#### 98124.3.4.6 cmisObjectListType and cmisObjectInFolderListType

9813cmisObjectListType and cmisObjectInFolderListType hold lists of cmisObjectType and  
9814cmisObjectInFolderType structures. They also contain the hasMoreItems flag and the numItems  
9815element that are returned by operations that return these lists.

#### 98164.3.4.7 cmisContentStreamType

9817cmisContentStreamType wraps a content stream and additional information about the stream.

		Client to Repository	Repository to Client
length	Length of the content stream in bytes. If set it MUST be a positive number. If the length is unknown it MUST NOT be set.	SHOULD be set	SHOULD be set
contentType	MIME Media Type of the content stream. For the primary content of a document it SHOULD match the value of the property cmis:contentType.	SHOULD be set	MUST be set
filename	Filename of the content stream. For the primary content of a document it SHOULD match the value of the property cmis:contentStreamFileName.	SHOULD be set	SHOULD be set
stream	The content stream. MUST be present even if the content stream has 0 bytes.	MUST be set	MUST be set

9818

#### 98194.3.4.8 cmisACLType

9820cmisACLType is the return structure of getACL and applyACL. It contains the current Access Control List  
9821(ACL) of the object and the exact flag that indicates if the ACL fully describes the permission of this  
9822object.

#### 98234.3.4.9 cmisExtensionType

9824cmisExtensionType is a placeholder for extensions. See 4.3.3 Extensions.

9825

---

## 9826 **5 IANA Considerations**

### 9827 **5.1 Content-Type Registration**

#### 9828 **5.1.1 CMIS Query**

9829 A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

9830

9831 MIME media type name: application

9832 MIME subtype name: cmisquery +xml

9833 Mandatory parameters: None

9834 Optional parameters:

9835 "charset": This parameter has semantics identical to the charset parameter of the

9836 "application/xml" media type as specified in [RFC3023].

9837 Encoding considerations:

9838 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

9839 Security considerations: As defined in this specification.

9840 In addition, as this media type uses the "+xml" convention, it shares the same security

9841 considerations as described in [RFC3023], Section 10.

9842 Interoperability considerations:

9843 There are no known interoperability issues.

9844 Published specification: This specification.

9845 Applications that use this media type:

9846 No known applications currently use this media type.

9847 Additional information:

9848 Magic number(s):

9849 As specified for "application/xml" in [RFC3023], Section 3.2.

9850 File extension: .cmisquery

9851 Fragment identifiers:

9852 As specified for "application/xml" in [RFC3023], Section 5.

9853 Base URI:

9854 As specified in [RFC3023], Section 6.

9855 Macintosh File Type code: TEXT

9856 Person and email address to contact for further information:

9857 Al Brown <albertcbrown@us.ibm.com>

9858 Intended usage: COMMON

9859 Author/Change controller: IESG

#### 9860 **5.1.2 CMIS AllowableActions**

9861 A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following

9862 media type:

9863  
9864 MIME media type name: application  
9865 MIME subtype name: cmisallowableactions +xml  
9866 Mandatory parameters: None.  
9867 Optional parameters:  
9868 "charset": This parameter has semantics identical to the charset parameter of the  
9869 "application/xml" media type as specified in [RFC3023].  
9870 Encoding considerations:  
9871 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9872 Security considerations: As defined in this specification.  
9873 In addition, as this media type uses the "+xml" convention, it shares the same security  
9874 considerations as described in [RFC3023], Section 10.  
9875 Interoperability considerations:  
9876 There are no known interoperability issues.  
9877 Published specification: This specification.  
9878 Applications that use this media type:  
9879 No known applications currently use this media type.  
9880 Additional information:  
9881 Magic number(s):  
9882 As specified for "application/xml" in [RFC3023], Section 3.2.  
9883 File extension: .cmisallowableactions  
9884 Fragment identifiers:  
9885 As specified for "application/xml" in [RFC3023], Section 5.  
9886 Base URI:  
9887 As specified in [RFC3023], Section 6.  
9888 Macintosh File Type code: TEXT  
9889 Person and email address to contact for further information:  
9890 Al Brown <albertcbrown@us.ibm.com>  
9891 Intended usage: COMMON  
9892 Author/Change controller: IESG  
9893

### 9894**5.1.3 CMIS Tree**

9895A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:

9896  
9897 MIME media type name: application  
9898 MIME subtype name: cmistree +xml  
9899 Mandatory parameters: None.  
9900 Optional parameters:  
9901 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9902 type as specified in [RFC3023].  
9903 Encoding considerations:  
9904 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

9905 Security considerations: As defined in this specification.  
9906In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9907described in [RFC3023], Section 10.  
9908 Interoperability considerations:  
9909There are no known interoperability issues.  
9910 Published specification: This specification.  
9911 Applications that use this media type:  
9912No known applications currently use this media type.  
9913 Additional information:  
9914 Magic number(s):  
9915As specified for "application/xml" in [RFC3023], Section 3.2.  
9916 File extension: .cmistree  
9917 Fragment identifiers:  
9918As specified for "application/xml" in [RFC3023], Section 5.  
9919 Base URI:  
9920As specified in [RFC3023], Section 6.  
9921 Macintosh File Type code: TEXT  
9922 Person and email address to contact for further information:  
9923Al Brown <albertcbrown@us.ibm.com>  
9924 Intended usage: COMMON  
9925 Author/Change controller: IESG  
9926

## 9927**5.1.4 CMIS Atom**

9928A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:  
9929  
9930 MIME media type name: application  
9931 MIME subtype name: cmisatom+xml  
9932 Mandatory parameters: None.  
9933 Optional parameters:  
9934"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9935type as specified in [RFC3023].  
9936"type": This parameter has semantics identical to the type parameter of the "application/atom+xml" as  
9937specified in [RFC4287]  
9938 Encoding considerations:  
9939Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9940 Security considerations: As defined in this specification.  
9941In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9942described in [RFC3023], Section 10.  
9943 Interoperability considerations:  
9944There are no known interoperability issues.  
9945 Published specification: This specification.  
9946 Applications that use this media type:

9947No known applications currently use this media type.  
9948 Additional information:  
9949 Magic number(s):  
9950As specified for "application/xml" in [RFC3023], Section 3.2.  
9951 File extension: .cmisatom  
9952 Fragment identifiers:  
9953As specified for "application/xml" in [RFC3023], Section 5.  
9954 Base URI:  
9955As specified in [RFC3023], Section 6.  
9956 Macintosh File Type code: TEXT  
9957 Person and email address to contact for further information:  
9958Al Brown <albertcbrown@us.ibm.com>  
9959 Intended usage: COMMON  
9960 Author/Change controller: IESG  
9961  
9962Please see section 3.1.1 on why this media type is needed above the Atom Media Type.

## 9963**5.1.5 CMIS ACL**

9964A CMIS ACL Document, when serialized as XML 1.0, can be identified with the following media type:  
9965  
9966 MIME media type name: application  
9967 MIME subtype name: cmisacl+xml  
9968 Mandatory parameters: None.  
9969 Optional parameters:  
9970"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9971type as specified in [RFC3023].  
9972 Encoding considerations:  
9973Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9974 Security considerations: As defined in this specification.  
9975In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9976described in [RFC3023], Section 10.  
9977 Interoperability considerations:  
9978There are no known interoperability issues.  
9979 Published specification: This specification.  
9980 Applications that use this media type:  
9981No known applications currently use this media type.  
9982 Additional information:  
9983 Magic number(s):  
9984As specified for "application/xml" in [RFC3023], Section 3.2.  
9985 File extension: .cmisacl  
9986 Fragment identifiers:  
9987As specified for "application/xml" in [RFC3023], Section 5.

9988 Base URI:  
9989As specified in [RFC3023], Section 6.  
9990 Macintosh File Type code: TEXT  
9991 Person and email address to contact for further information:  
9992Al Brown <albertcbrown@us.ibm.com>  
9993 Intended usage: COMMON  
9994 Author/Change controller: IESG  
9995

9996

## 6 Conformance

9997An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level  
9998requirements defined within this specification.

9999Specification:

10000 This specification references a number of other specifications (see the table above). In order to  
10001 comply with this specification, an implementation MUST implement the portions of referenced  
10002 specifications necessary to comply with the required provisions of this specification. Additionally,  
10003 the implementation of the portions of the referenced specifications that are specifically cited in  
10004 this specification MUST comply with the rules for those portions as established in the referenced  
10005 specification.

10006

10007 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level  
10008 requirements defined within this specification.

10009

10010

10011Domain Model:

10012 Normative text within this specification takes precedence over the CMIS Core XML Schema.

10013 That is, the normative text in this specification further constrains the schemas and/or WSDL that  
10014 are part of this specification; and this specification contains further constraints on the elements  
10015 defined in referenced schemas.

10016

10017 Clients:

10018 Client implementations MAY implement either Restful AtomPub Binding or the Web  
10019 Services Binding.

10020

10021 Repositories:

10022 Repositories MUST implement the following CMIS protocol bindings:

- 10023 ▪ Restful AtomPub Binding
- 10024 ▪ Web Services Binding

10025

10026Rest Binding:

10027 This specification references a number of other specifications. In order to comply with this  
10028 specification, an implementation MUST implement the portions of referenced specifications  
10029 necessary to comply with the required provisions of this specification. Additionally, the  
10030 implementation of the portions of the referenced specifications that are specifically cited in this  
10031 specification MUST comply with the rules for those portions as established in the referenced  
10032 specification.

10033 Additionally normative text within this specification takes precedence over the CMIS RestAtom  
10034 XML Schema. That is, the normative text in this specification further constrains the schemas  
10035 and/or WSDL that are part of this specification; and this specification contains further constraints  
10036 on the elements defined in referenced schemas.



10037 The CMIS RestAtom XML takes precedence over any examples or non-normative outlines  
10038 included either in this document or as standalone examples.

10039

10040Web Services Binding:

10041 Normative text within this specification takes precedence over the CMIS Messaging XML and  
10042 CMIS WSDL. That is, the normative text in this specification further constrains the schemas and  
10043 WSDL that are part of this specification; and this specification contains further constraints on the  
10044 elements defined in referenced schemas.

10045 The CMIS Messaging XML and CMIS WSDL takes precedence over any examples or non-  
10046 normative outlines included either in this document or as standalone examples.

---

## 10047 **A. Acknowledgements**

10048 The following individuals have participated in the creation of this specification and are gratefully  
10049 acknowledged:

10050

### 10051 **Participants:**

10052 Philippe Allart, Adullact  
10053 Florian Bartels, fme AG  
10054 Fred Boiscuvier, Exalead, Inc.  
10055 Al Brown, IBM  
10056 Jay Brown, IBM  
10057 Mark Carlson, Sun Microsystems  
10058 Derek Carr, IBM  
10059 David Caruana, Alfresco Software  
10060 Eric Chan, Oracle Corporation  
10061 Sameer Charles, Magnolia International AG  
10062 David Choy, EMC Corporation  
10063 Scott Conroy, Individual  
10064 Cornelia Davis, EMC Corporation  
10065 Kevin Dorr, Flatirons Solutions Corporation  
10066 Jason Dubreuil, Fidelity Investments  
10067 Betsy Fanning, AIIM  
10068 Steffen Frederiksen, Content Technologies ApS  
10069 Stephan Friedl, Quark  
10070 Dustin Friesenhahn, Microsoft Corporation  
10071 Gary Gershon, Individual  
10072 Paul Goetz, SAP AG  
10073 Gregory Grefenstette, Exalead, Inc.  
10074 Florent Guillaume, Nuxeo  
10075 Ethan Gur-esh, Microsoft Corporation  
10076 Martin Hermes, SAP AG  
10077 Jens Huebel, Open Text Corporation  
10078 David Izatt, Structured Software Systems Limited (3SL)  
10079 Gershon Janssen, Individual  
10080 Volker John, Saperion AG  
10081 Shane Johnson, Citytech, Inc.  
10082 Christophe Kijewska, Adullact  
10083 Ijonas Kisselbach, Vamosa  
10084 Mark Klamerus, Individual  
10085 Stephan Klevenz, SAP AG  
10086 Boris Kraft, Magnolia International AG

10087 Alison Macmillan, Oracle Corporation  
 10088 Michael Marth, Day Software  
 10089 Mary McRae, OASIS  
 10090 Ryan McVeigh, Oracle Corporation  
 10091 Juerg Meier, fme AG  
 10092 Gregory Melahn, IBM  
 10093 Pat Miller, Microsoft Corporation  
 10094 Florian Mueller, Open Text Corporation  
 10095 Thomas Mueller, Day Software  
 10096 John Newton, Alfresco Software  
 10097 David Nuescheler, Day Software  
 10098 Conleth O'Connell, Vignette Corporation  
 10099 Dominique Pfister, Day Software  
 10100 Peeter Piegaze, Day Software  
 10101 David Pitfield, Oracle Corporation  
 10102 Norrie Quinn, EMC Corporation  
 10103 Craig Randall, Adobe Corporation  
 10104 Celso Rodriguez, ASG Software Solutions  
 10105 Steve Roth, Oracle Corporation  
 10106 Patrick Ryan, IBM  
 10107 Angela Schreiber, Day Software  
 10108 Spencer Shearer, Exalead, Inc.  
 10109 Madi Solomon, Pearson PLC  
 10110 Wojciech Specht, fme AG  
 10111 Maik Uhlenberg, fme AG  
 10112 Oliver Walthard, Day Software  
 10113 Patrick Ward, Booz Allen Hamilton  
 10114  
 10115 Original Authors of the initial contribution:  
 10116 Al Brown, IBM  
 10117 David Choy, EMC  
 10118 Cornelia Davis, EMC  
 10119 Ethan Gur-Esh, Microsoft  
 10120  
 10121 Original Acknowledgements of the initial contribution:  
 10122 Al Brown, IBM  
 10123 David Caruana, Alfresco  
 10124 Derek Carr, IBM  
 10125 David Choy, EMC  
 10126 Cornelia Davis, EMC  
 10127 Paul Goetz, SAP  
 10128 Ethan Gur-Esh, Microsoft

10129	Martin Hermes, SAP
10130	Jens Hubel, OpenText
10131	Jay Brown, IBM
10132	Ryan McVeigh, Oracle
10133	Gregory Melahn, IBM
10134	Florian Mueller, OpenText
10135	John Newton, Alfresco
10136	Norrie Quinn, EMC
10137	Steve Roth, Oracle
10138	Craig Randall, EMC

---

10139 **B. Non-Normative Text**

10140

---

## 10141 C. Revision History

10142

Revision	Date	Editor	Changes Made
1.0		Al Brown	First specification

10143