Content Management Interoperability Services (CMIS) Version 1.0 Errata 01

OASIS Approved Errata 01

04 November 2011

Specification URIs

This version:

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/os/cmis-spec-v1.0-errata-01-os.doc (Authoritative)

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/os/cmis-spec-v1.0-errata-01-os.html

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/os/cmis-spec-v1.0-errata-01-os.pdf

Previous version:

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/csprd01/cmis-spec-v1.0-errata-01-csprd01.doc (Authoritative)

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/csprd01/cmis-spec-v1.0-errata-01-csprd01.html

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/csprd01/cmis-spec-v1.0-errata-01-csprd01.pdf

Latest version:

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/cmis-spec-v1.0-errata-01.doc (Authoritative)

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/cmis-spec-v1.0-errata-01.html

http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/cmis-spec-v1.0-errata-01.pdf

Technical Committee:

OASIS Content Management Interoperability Services (CMIS) TC

Chair:

David Choy (mdavid.choy@emc.com), EMC

Editors:

Ryan McVeigh (rmcveigh@ziaconsulting.com), Zia Consulting

Florian Müller (florian.mueller@alfresco.com), Alfresco

Related work:

This specification is related to:

- *Content Management Interoperability Services (CMIS) Version 1.0.* OASIS Standard.
  http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.html

Declared XML namespaces:

- http://docs.oasis-open.org/ns/cmis/core/200908/
- http://docs.oasis-open.org/ns/cmis/restatom/200908/
- http://docs.oasis-open.org/ns/cmis/messaging/200908/
- http://docs.oasis-open.org/ns/cmis/ws/200908/
- http://docs.oasis-open.org/ns/cmis/link/200908/

Abstract:

This document lists approved errata to the CMIS Version 1.0 OASIS Standard.

Status:

This document was last revised or approved by the OASIS Content Management Interoperability Services (CMIS) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

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

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

Citation format:

When referencing this specification the following citation format should be used:

[CMIS-v1.0-Errata-1]

Content Management Interoperability Services (CMIS) Version 1.0 Errata 01. 04 November 2011. OASIS Approved Errata 01.
http://docs.oasis-open.org/cmis/CMIS/v1.0/errata-01/os/cmis-spec-v1.0-errata-01-os.html.

Notices

Copyright © OASIS Open 2011. All Rights Reserved.

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

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

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

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

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

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

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

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

1 Introduction

This document lists the approved errata to the CMIS v1.0 OASIS Standard. Each one has a number that refers to the issue that triggered the erratum.

As required by the OASIS Technical Committee Process, the approved errata represent changes that are not "substantive". The changes focus on clarifications to ambiguous or conflicting specification text, where different compliant implementations might have reasonably chosen different interpretations. The intent of the CMIS TC has been to resolve such issues in service of improved interoperability based on implementation and deployment experience.

In this document, errata change instructions are presented with surrounding context as necessary to make the intent clear. Original specification text is often presented as follows, with problem text highlighted in bold:

> This is an original specification sentence. **The second sentence needs to be changed, removed, or replaced.**

New specification text is typically presented as follows, with new or changed text highlighted in bold:

> This is a **highly** original specification sentence. **This is the wholly new content to replace the old second sentence. It runs on and on and on.**

In a few cases, text needs only to be struck, in which case the change is shown as follows, with text to be removed both highlighted in bold and struck through:

> This is yet another original specification sentence which contains a~~n inappropriately~~ long description.

In addition to this normative document, non-normative "errata composite" documents may be provided that combine the prescribed corrections with the original specification text, illustrating the changes with margin change bars, struck-through original text, and highlighted new text. These documents, if available, will be found at the same location as this approved form.

All cited line numbers refer to the Word document of the original OASIS Standard specifications in question, not to line numbers in this document or in the errata composite documents.

1.1 Normative References

[CMIS1.0]   *CMIS v1.0 OASIS Standard*, http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.doc, May 2010

33 **2 Approved Errata**

34 Following are the approved errata to the CMIS v1.0 OASIS Standard.

35 **2.1 CMIS-534: getAllowableActions Service output seems to restrict to**
36 **XML based bindings**

37 Change Section 2.2.1.2.6 at line 2853.

38 Original:

> - AllowableActions: **See `cmisAllowableActionsType` in the CMIS schema.**

39 New:

> - **<Array>** AllowableActions: **The list of allowable actions for the object.**

40 **2.2 CMIS-537: getRenditionsExceptions – filterNotValid is overloaded**

41 Change Section 2.2.3.1.3 at line 3206-3207.

42 Original:

> - `filterNotValid`**:** The Repository MUST throw this exception if **this** property filter input parameter is not valid.

43 New:

> - `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.

44 Change Section 2.2.3.2.3 at line 3249-3250.

45 Original:

> - `filterNotValid`**:** The Repository MUST throw this exception if this property filter input parameter is not valid.

46 New:

> - `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.

47 Change Section 2.2.3.3.3 at line 3288-3289.

48 Original:

> - `filterNotValid`**:** The Repository MUST throw this exception if this property filter input parameter is not valid.

49 New:

> - `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.

50 Change Section 2.2.3.5.3 at line 3337-3338.

51 Original:

> - `filterNotValid`**:** The Repository MUST throw this exception if this property filter input

	parameter is not valid.
52	New:
	• `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.
53	Change Section 2.2.3.6.3 at line 3438.
54	Original:
	• `filterNotValid`**:** The Repository MUST throw this exception if this property filter input parameter is not valid.
55	New:
	• `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.
56	Change Section 2.2.4.7.3 at line 3643-3644.
57	Original:
	• `filterNotValid`**:** The Repository MUST throw this exception if this property filter input parameter is not valid.
58	New:
	• `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.
59	Change Section 2.2.4.9.3 at line 3677-3678.
60	Original:
	• `filterNotValid`**:** The Repository MUST throw this exception if this property filter input parameter is not valid.
61	New:
	• `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.
62	Change Section 2.2.4.11.3 at line 3717.
63	Original:
	• `filterNotValid` : The filter specified is not valid
64	New:
	• `filterNotValid` : The **rendition** filter specified is not valid
65	Change Section 2.2.7.4.3 at line 4103-4104.
66	Original:
	• `filterNotValid`**:** The Repository MUST throw this exception if this property filter input parameter is not valid.
67	New:
	• `filterNotValid`**:** The Repository MUST throw this exception if **the** property **or rendition** filter input parameter is not valid.

2.3 CMIS-564: createDocumentFromSource versioningState param needs clarification

Change section 2.2.4.2.1 at lines 3455-3456.

Original:

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

New:

- Enum versioningState: An enumeration specifying what the **versioning** state of the newly-created object MUST be. **If the repository does not support versioning, the repository MUST ignore the versioningState parameter.** Valid values are:

2.4 CMIS-641: applyACLResponse does not match domain model - no changeToken

Change section 2.2.10.2.2 at line 4279.

New:

- ~~**String changeToken:** See section 2.2.1.3.~~

2.5 CMIS-646: cmis:objectTypeId should not be required for documents

Change section 2.1.5.4.2 at line 1190.

Original:

Required:	**False**

New:

Required:	**True**

Change section 2.1.6.1.3 at line 1427.

Original:

Required:	**False**

New:

Required:	**True**

Change section 2.1.7.1.2 at line 1649.

Original:

Required:	**False**

New:

Required:	**True**

2.6 CMIS-647: ACL propagation in ACL capabilities is supposed to be an array

90   Change section 2.1.8.3 at lines 1744-1745.

91   New:

> - ~~<Array>~~ Enum propagation: specifies, how non-direct ACEs can be handled by the repository using the following values (see section 2.2.10.2 applyACL):

92   Change section 2.1.8.3 at lines 1748-1749.

93   Original:

> o   ~~propagate: indicates that the ACEs is to be applied to the given object and all inheriting objects.~~

94   New:

> o   propagate: indicates that the ACEs is to be applied to the given object and all inheriting objects. **Propagate incorporates the support for objectonly.**

2.7 CMIS-648: AllowableActions keys inconsistencies

96   Change section 2.1.8.3.1 at lines 1813-1818.

97   New:

> ~~canGetFolderTree~~
>
> ~~Description:~~   ~~Can get the sub-folder tree of the folder (getFolderTree)~~
>
> ~~Base Object:~~   ~~cmis:folder~~
>
> ~~Operand:~~   ~~cmis:folder~~
>
> ~~Key:~~   ~~canGetFolderTree.Folder~~
>
> ~~Permission:~~   ~~Read~~

98   Change section 2.1.8.3.1 at line 1831.

99   Original:

> ~~Key:~~   ~~canGetFolderParent.Folder~~

100   New:

> Key: canGetFolderParent.**Object**

101   Change section 2.1.8.3.1 at line 1838.

102   Original:

> ~~Key:~~   ~~canGet**Object**Parents.Object~~

103   New:

> Key: canGetParents.Object

104   Change section 2.1.8.3.1 at lines 1880-1885.

105   New:

> ~~canGetRenditions~~
>
> ~~Description:~~   ~~Can retrieve the renditions of this object (getRenditions)~~

Base Object:	cmis:document, or cmis:folder	
Operand:	Object	
Key:	canGetRenditions.Object	
Permission:	Read	

106 Change section 2.1.8.3.1 at lines 1887-1893.

107 New:

canGetContentStream

Description:	Can get the content stream for the Document object (getContentStream)
Base Object:	cmis:document
Operand:	Object
Key:	canGetContentStream.Object
Permission:	Read

108 Change section 2.1.8.3.1 at line 1913.

109 Original:

Key:	canMove**Object**.Target

110 New:

Key:	canMove.Target

111 Change section 2.1.8.3.1 at line 1920.

112 Original:

Key:	canMove**Object**.Source

113 New:

Key:	canMove.Source

114 Change section 2.1.8.3.1 at lines 1930-1935.

115 New:

canDeleteObject

Description:	Can delete an object that is a child of this folder (deleteObject)
Base Object:	cmis:folder
Operand:	Folder
Key:	canDelete.Folder
Permission:	Read

116 Change section 2.1.8.3.1, add after line 1935.

117 New:

canGetContentStream

Base Object:	cmis:document
Action:	Can get the content stream for the Document object (getContentStream)
Operand:	Object

Key:	canViewContent.Object
Permission:	Read

118 Change section 2.1.8.3.1 at line 1942.

119 Original:

Key:	canSetContent**Stream**.Document

120 New:

Key:	canSetContent.Document

121 Change section 2.1.8.3.1 at line 1950.

122 Original:

Key:	canDeleteContent**Stream**.Document

123 New:

Key:	canDeleteContent.Document

124 Change section 2.1.8.3.1 at line 1980.

125 Original:

Key:	canRemove**Object**FromFolder.Object

126 New:

Key:	canRemoveFromFolder.Object

127 Change section 2.1.8.3.1 at line 1996.

128 Original:

Key:	canCheckOut.Document

129 New:

Key:	**canCheckout**.Document

130 Change section 2.1.8.3.1 at line 2017.

131 Original:

Key:	canGetAllVersions.**Document**

132 New:

Key:	canGetAllVersions.**VersionSeries**

133 2.8 CMIS-650: change atom:id reference from URI to IRI

134 Change section 3.5.1 at lines 5130-5131.

135 Original:

- atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's specification and be a valid **URI**.

136 New:

- atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's specification and be a valid **IRI**.

137 2.9 CMIS-651: Clarify that URI templates are to be used with GET

138 Change section 3.6.1, add after line 5361.

139 New:

> URI Templates MUST only be used with HTTP GET.

140 2.10 CMIS-656: Clarify how ETags and cmis:changeToken interact

141 Change section 3.2.1, add after line 4402.

142 New:

> On updates, perform the following checks (HTTP & CMIS levels):
>
> 1. If If-Match header is sent by client, ETag value is pulled from HTTP header If-Match per RFC2616. The supplied ETag is compared against the ETag on the server. If the match fails, then status code 412 is used.
>
> 2. If cmis:changeToken property is supplied by the client, compare the supplied and the cmis:changeToken on the server. If the comparison fails, then return status code 409 per CMIS.
>
> 3. If ETag and cmis:changeToken are both specified, the HTTP If-Match check should be performed first.

143 2.11 CMIS- 657: Clarify cancelCheckout's proper behavior when doc
144 was created in a checked out state.

145 Change section 2.2.4.1.1 at line 3395.

146 Original:

> - ~~checkedout: The document MUST be created in the checked-out state.~~

147 New:

> - checkedout: The document MUST be created in the checked-out state. **The checked-out document MAY be visible to other users.**

148 Change section 2.2.7.2 at lines 4023-4024.

149 Original:

> ~~Description: Reverses the effect of a check-out. Removes the private working copy of the checked-out document, allowing other documents in the version series to be checked out again.~~

150 New:

> Description: Reverses the effect of a check-out. Removes the private working copy of the checked-out document, allowing other documents in the version series to be checked out again. **If the private working copy has been created by createDocument, cancelCheckOut MUST delete the created document.**

151 2.12 CMIS-659: Clarification for properties in state "not set" required

152 Change section 2.1.2.1 at lines 218-223.

153 Original:

> If a value is not provided for a property, the property is in a "value not set" state. There is no "null" value for a property. Through protocol binding, a property is either not set, or is set to a particular value or a list of values.
>
> A multi-valued property is either set or not set in its entirety. An individual value of a multi-valued property MUST NOT be in an individual "value not set" state and hold a position in the list of values. An empty list of values MUST NOT be allowed.

154 New:

> **A property, either single-valued or multi-valued, MAY be in a "not set" state. CMIS does not support "null" property value.**
>
> **If a multi-valued property is not in a "not set" state, its property value MUST be a non-empty list of individual values. Each individual value in the list MUST NOT be in a "not set" state and MUST conform to the property's property-type.**

155 Change section 2.2.1.2.1 at lines 2784-2785.

156 Original:

> If a property filter specifies a property that is 'not set', it MUST be represented as a property element without a value element.

157 New:

> **If a property is requested by a filter, a property element MUST be returned for that property. A repository MAY return additional properties. If a returned property is in a "not set" state, a value element MUST NOT be returned for that property.**

2.13 CMIS-663: localName for type definition is optional according to domain model but is required according to core.xsd schema

160 Change section 2.1.3.2.1 at line 327 to match the CMIS schema.

161 New:

localName	String **(optional)**

2.14 CMIS-674: Change BNF comment to !! style comment

163 Change section 2.2.1.2.4.1 at line 2826.

164 Original:

```
<text> ::= /* any char except whitespace */
```

165 New:

```
<text> ::= !! any char except whitespace
```

2.15 CMIS-687: cmis:changeToken should not be mandatory if the repository doesn't support change tokens

168 Change section 2.1.4.3.3 at lines 845-846.

169 Original:

> Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them

170 New:

> Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them. **If the repository does not support change tokens, this property SHOULD not be set.**

171 Change section 2.1.5.4.2 at lines 1260-1261.

172 Original:

> ~~Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them~~

173 New:

> Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them. **If the repository does not support change tokens, this property SHOULD not be set.**

174 Change section 2.1.6.1.3, add after line 1511.

175 New:

> **Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them. If the repository does not support change tokens, this property SHOULD not be set.**

176 Change section 2.1.7.1.2, add after line 1701.

177 New:

> **Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them. If the repository does not support change tokens, this property SHOULD not be set.**

178 2.16 CMIS-688: Query sample incorrect

179 Change section 2.1.10.2.4.2.2 at line 2549-2551.

180 Original:

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

181 New:

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

182 2.17 CMIS-689: invalid property name cmis:createdDate referenced in
183   3.5.2

184 Change section 3.5.2 at lines 5157-5159.

185 Original:

> - ~~app:edited MUST be cmis:lastModifiedDate~~
> - ~~atom:updated MUST be cmis:lastModifiedDate~~
> - ~~atom:published MUST be cmis:createdDate~~

186   New:

> - app:edited MUST be **cmis:lastModificationDate**
> - atom:updated MUST be **cmis:lastModificationDate**
> - atom:published MUST be **cmis:creationDate**

187   2.18 CMIS-690: cmis:versionLabel cardinality not specified

188   Change section 2.1.4.3.3, add after line 903.

189   New:

> Cardinality: Single

190   2.19 CMIS-695: Escaping section should specify which BNF symbols
191   it applies to

192   Replace entire section 2.1.10.3 at lines: 2676-2681.

193   Original:

> Repositories MUST support the escaping of characters using a backslash (\) in the query statement. The
> backslash character (\) will be used to escape characters *within quoted strings* in the query as follows:
>
> 1. \' will represent a single-quote(') character
> 2. \ \ will represent a backslash (\) character
> 3. Within a LIKE string, \% and _ will represent the literal characters % and _, respectively.
> 4. All other instances of a \ are errors.

194   New:

> Character escaping for character strings differs from SQL-92's escaping. A repository MUST support the
> escaping of certain literal characters in a character string, or in a text expression, using a backslash
> character (\) in the following manner. For a <character string literal>, which MUST BE a string enclosed
> in single-quotes according to the SQL-92 grammar, any occurrence of the single-quote character (') and
> the escape character (\) in the string MUST BE escaped. This applies to <folder id>, which is a <character
> string literal>. Furthermore, when a <character string literal> is used in a LIKE predicate, any occurrence
> of the percent character (%) and the underscore character (_) in the string as a literal MUST BE escaped
> also. Therefore, within a quoted string in a query:
>
> - The double character \' represents a literal single-quote (') character.
> - The double character \ \ represents a literal backslash (\) character.
> - Within a LIKE string, the double characters \% and _ represent a literal percent (%) character
>   and a literal underscore (_) character respectively.
> - All other instances of a backslash (\) character are errors.
>
> Using double single-quotes ('') as a SQL-92 way to escape a literal single-quote (') character SHOULD
> BE supported as an allowable alternative to the double character \'.
>
> For a <text search expression>, a second-level character escaping is required so that the <text search
> expression> sub-grammar is *isolatable* from the query statement-level grammar. When a text search
> expression is composed for a query according to the <text search expression> sub-grammar, any
> occurrence of the following three characters in the expression as a literal character MUST BE escaped:
> hyphen (-), single-quote ('), and the escape character (\). Then, before this expression is enclosed in
> single-quotes and inserted into a CONTAINS() predicate, the query statement-level escaping rules
> described in the above MUST BE applied. This two-level character escaping allows a query statement
> parser, using statement-level escaping rules, to correctly extract a <text search expression> as a

character string literal independent of the <text search expression> sub-grammar. This extracted <text search expression> can then be correctly interpreted by a full-text search parser independent of the query-statement grammar, using second-level escaping rules. Since the <text search expression> sub-grammar is isolated from the SQL-92 grammar, double single-quotes is not a valid way to escape a literal single-quote character for second-level character escaping.

An <identifier> in a query statement MUST conform to the SQL-92 identifier syntax, and MUST NOT require character escaping.

Example 1:
A query statement that contains a full-text search for the literal string "John'sPresentation-Version2" may be composed as:
    SELECT … FROM … WHERE … CONTAINS('John\\\'sPresentation\\-Version2') …
A query parser extracts from this statement the text search expression "John\'sPresentation\-Version2" as a character string literal, and passes it to a text-search parser, which interprets it as a single-word full-text search criteria: *John'sPresentation-Version2*.

Example 2:
A query statement that contains a full-text search for the phrase "Content Management" may be composed as:
    SELECT … FROM … WHERE … CONTAINS('\'Content Management\'') …

A query parser extracts from this statement the text search expression "'Content Management'" as a character string literal, and passes it to a text-search parser, which interprets it as a full-text search criteria consisting of a single phrase: *Content Management*. There is no second-level escaping.

2.20 CMIS-697: Word plus phrase example for CONTAINS query

Replace section 2.1.10.3. The same changes for CMIS-695.

2.21 CMIS-706: Domain model should state objects are not constrained

Add to section 2.1.2 following line 212.

New:

CMIS objects MAY expose additional information, such as vendor-specific workflow data, beyond the attributes described above. In this respect, the data model can be extended as desired. This specification does not standardize such extensions.

2.22 CMIS-725: Remove hyperlinks to namespace URIs in the specification document

Remove namespace URI hyperlinks from the following lines:

4319, 4321, 4323, 4325, 5057, 5064, 5070, 5077,5084, 5090, 5097, 5102, 5108, 5113, 5332, 5337, 5339, 5341, 6429, 7237, 7578, 8221, 8866, 8868, 8870, 8871, 9008, 9010, 9012, 9013, 9153, 9155, 9157, 9158, 9159, 9280, 9282, 9283, 9391, 9393, 9394

Remove all namespace URI hyperlinks from the title page (section ” Declared XML Namespace(s)”).

210 Additionally, remove other hyperlinks from the following lines that are either incorrect or rendered
211 incorrectly in some renditions of the specification document:

212 183, 285, 293, 294, 296, 297, 303, 371, 451, 584, 1075, 1076, 2124, 2292, 2440, 2577, 2613, 2672,
213 2695, 2724, 2877, 3007, 3037, 3053, 3069, 3104, 3119, 3126, 3150, 3210, 3880, 3989, 4041, 5574,
214 9710, 9743, 9778, 9812

215

216 Additionally, change section 1.2 at line 28-30. (Note the hyperlink.)

217 Original:

> J. Reschke Editor, A. Brown, G. Clemm, *Link Relation Types for Simple Version Navigation between Web Resources*, http://www.ietf.org/id/draft-brown-versioning-link-relations-07.txt, 2010

218 New:

> J. Reschke Editor, A. Brown, G. Clemm, *Link Relation Types for Simple Version Navigation between Web Resources*, http://www.ietf.org/id/draft-brown-versioning-link-relations-08.txt, 2010

219

220 **Appendix A. Acknowledgements**

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

223

224 **Participants:**
225   Michael Duerig, Adobe Systems
226   David Nuescheler, Adobe Systems
227   Peeter Piegaze, Adobe Systems
228   Craig Randall, Adobe Systems
229   Angela Schreiber, Adobe Systems
230   Philippe Allart, Adullact
231   Betsy Fanning, AIIM
232   David Caruana, Alfresco Software
233   Florian Mueller, Alfresco Software
234   John Newton, Alfresco Software
235   Celso Rodriguez, ASG Software Solutions
236   Patrick Ward, Booz Allen Hamilton*
237   Steffen Frederiksen, Content Technologies ApS
238   Alpesh Patel, Ektron*
239   David Choy, EMC Corporation
240   Cornelia Davis, EMC Corporation
241   Norrie Quinn, EMC Corporation
242   Raphael Jean, Entropysoft
243   Hareesh Kadlabalu, FatWire
244   Derek Chow, Genus Technologies, LLC
245   Randy Dufault, Genus Technologies, LLC
246   Thomas Pole, Harris Corp
247   David Churchland, Hewlett-Packard*
248   Andrew Ewing, Hewlett-Packard*
249   Jay Brown, IBM
250   Derek Carr, IBM
251   Jane Doong, IBM
252   Scott Malabarba, IBM
253   Gregory Melahn, IBM
254   Patrick Ryan, IBM
255   Scott Conroy, Individual
256   Gershon Janssen, Individual
257   Mark Klamerus, Individual
258   Paolo Garroni, ISIS Papyrus America Inc.
259   Ethan Gur-esh, Microsoft Corporation
260   Adam Harmetz, Microsoft Corporation
261   Neil Joyer, Microsoft Corporation
262   Pat Miller, Microsoft Corporation
263   Florent Guillaume, Nuxeo
264   Jens Huebel, Open Text Corporation
265   Mark Carlson, Oracle Corporation
266   Eric Chan, Oracle Corporation
267   Alison Macmillan, Oracle Corporation
268   David Pitfield, Oracle Corporation
269   Steve Roth, Oracle Corporation
270   Paul Goetz, SAP AG*
271   Martin Hermes, SAP AG*
272   Stephan Klevenz, SAP AG*

273	Paul Tazbaz, Wells Fargo*
274	Alexander Haag, WeWebU Software AG
275	James Michel, WeWebU Software AG
276	Rainer Pausch, WeWebU Software AG
277	Ryan McVeigh, Zia Consulting, Inc.

278 # Appendix B. Revision History

279

Revision	Date	Editor	Changes Made
1.0	10/15/2010	Florian Müller	First draft

280