

1

# Open Geospatial Consortium Inc.

2

Date: 31 Mar 2010

3

Reference number of this OGC® project document: **OGC 09-165r3**

4

Version: 0.1.3

5

Category: OpenGIS® Draft Implementation Standard

6

Editor(s): Farrukh Najmi, Panagiotis (Peter) A. Vretanos

7

## ebXML RegRep Spatio-temporal Application Profile

8

Copyright © 2010 Open Geospatial Consortium, Inc.

9

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

10

### Warning

11

This document is not an OGC Standard. It is distributed for review and comment. It is

12

subject to change without notice and may not be referred to as an OGC Standard.

13

Recipients of this document are invited to submit, with their comments, notification of any

14

relevant patent rights of which they are aware and to provide supporting documentation.

Document subtype: OGC® Application Profile  
Document subtype: OGC® Draft Implementation Standard  
Document stage: Draft  
Document language: English

1 **OGC 09-165r3**

15 **NOTICE**

16 Permission to use, copy, and distribute this document in any medium for any purpose and without fee or royalty is hereby granted,  
17 provided that you include the above list of copyright holders and the entire text of this NOTICE.

18 We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to  
19 the implementation of the contents of this document, or any portion thereof.

20 No right to create modifications or derivatives of OGC documents is granted pursuant to this license. However, if additional  
21 requirements (as documented in the Copyright FAQ at [http://www.opengis.org/legal/ipr\\_faq.htm](http://www.opengis.org/legal/ipr_faq.htm)) are satisfied, the right to create  
22 modifications or derivatives is sometimes granted by the OGC to individuals complying with those requirements.

23 THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR  
24 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY,  
25 FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE  
26 DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL  
27 NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

28 COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL  
29 DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE  
30 CONTENTS THEREOF.

31 The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents  
32 without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

33 RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision  
34 (c)(1)(ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013

35 OpenGIS® is a trademark or registered trademark of Open GIS Consortium, Inc. in the United States and in other countries.

## 36 Table of Contents

37	<a href="#">1 Scope.....</a>	<a href="#">8</a>
38	<a href="#">2 Conformance.....</a>	<a href="#">9</a>
39	<a href="#">3 Normative References.....</a>	<a href="#">9</a>
40	<a href="#">4 Terms and Definitions.....</a>	<a href="#">10</a>
41	<a href="#">5 Conventions.....</a>	<a href="#">10</a>
42	<a href="#">5.1 Symbols (and abbreviated terms).....</a>	<a href="#">10</a>
43	<a href="#">5.2 XML Namespaces Used.....</a>	<a href="#">10</a>
44	<a href="#">5.3 UML Notation.....</a>	<a href="#">13</a>
45	<a href="#">6 Spatial Data Encoding.....</a>	<a href="#">13</a>
46	<a href="#">6.1 Spatial Data GML Encoding.....</a>	<a href="#">13</a>
47	<a href="#">6.2 Spatial Data WKT Encoding.....</a>	<a href="#">14</a>
48	<a href="#">7 Temporal Data Encoding.....</a>	<a href="#">14</a>
49	<a href="#">7.1 Temporal Data GML Encoding.....</a>	<a href="#">14</a>
50	<a href="#">7.2 Temporal Data ISO8601 Encoding.....</a>	<a href="#">15</a>
51	<a href="#">8 Extension Profile for ebXML RegRep 4 .....</a>	<a href="#">15</a>
52	<a href="#">8.1 Encoding Spatio-temporal Data in Slots.....</a>	<a href="#">15</a>
53	<a href="#">8.2 Publishing RegistryObjects With Spatio-Temporal Data.....</a>	<a href="#">16</a>
54	<a href="#">8.3 Querying RegistryObjects Using Spatio-temporal Data.....</a>	<a href="#">16</a>
55	<a href="#">8.3.1 Canonical Function: spatialFilter.....</a>	<a href="#">17</a>
56	<a href="#">8.3.1.1 Parameter Summary.....</a>	<a href="#">17</a>
57	<a href="#">8.3.1.2 Function Semantics.....</a>	<a href="#">18</a>
58	<a href="#">8.3.1 Canonical Function: temporalFilter.....</a>	<a href="#">20</a>
59	<a href="#">8.3.1.1 Parameter Summary.....</a>	<a href="#">20</a>
60	<a href="#">8.3.1.2 Function Semantics.....</a>	<a href="#">21</a>
61	<a href="#">8.3.1.3 Temporal Filter Open Issues:.....</a>	<a href="#">22</a>
62	<a href="#">8.3.2 Supporting Canonical Functions in OGC Filter Queries.....</a>	<a href="#">22</a>
63	<a href="#">8.3.3 Invoking Spatio-Temporal Queries Via REST Interface.....</a>	<a href="#">23</a>
64	<a href="#">8.4 Discovering a Server's Spatio-temporal Capabilities.....</a>	<a href="#">24</a>
65	<a href="#">9 Extension Profile for ebXML RegRep 3.....</a>	<a href="#">26</a>
66	<a href="#">9.1 Encoding Spatio-temporal Data in Slots.....</a>	<a href="#">26</a>
67	<a href="#">9.2 Publishing RegistryObjects With Spatio-Temporal Data.....</a>	<a href="#">27</a>
68	<a href="#">9.3 Querying RegistryObjects Using Spatio-temporal Data.....</a>	<a href="#">27</a>
69	<a href="#">9.4 Discovering a Server's Spatio-temporal Capabilities.....</a>	<a href="#">27</a>
70	<a href="#">Annex A. Abstract Test Suite (normative).....</a>	<a href="#">29</a>
71	<a href="#">A.1. TBD.....</a>	<a href="#">29</a>
72	<a href="#">Annex B. Query Functions (normative).....</a>	<a href="#">29</a>
73	<a href="#">B.1. Query Functions.....</a>	<a href="#">29</a>

3	<b>OGC 09-165r3</b>	
74	<b>B.1.1. <a href="#">Using Functions in Query Expressions</a></b>	<b>29</b>
75	<b>B.1.2. <a href="#">Using Functions in Query Parameters</a></b>	<b>30</b>
76	<b>B.1.3. <a href="#">Function Processing Model</a></b>	<b>30</b>
77	<b>B.1.4. <a href="#">Function Processor BNF</a></b>	<b>31</b>
78	<b>B.2. <a href="#">Common Patterns In Query Functions</a></b>	<b>32</b>
79	<b>B.2.1. <a href="#">Specifying a null Value for string Param or Return Value</a></b>	<b>32</b>

## Index of Figures

Figure 1 — UML notation	13
-------------------------	----

## Index of Tables

Table 1: Namespaces Used	12
Table 2: Canonical Functions Defined By This Profile	17

## 80 **i. Preface**

81 The OASIS ebXML RegRep specifications define a general purpose registry and  
 82 repository capability with a rich feature set. This feature set currently lacks ability to tag  
 83 informatio with spatia-temporal metadata and to discover information based upon spatio-  
 84 temporal metadata.

85 OGC catalog specifications do not provide the range of features offered by OASIS  
 86 ebXML RegRep specifications, are restricted to the geo community and hence are not  
 87 interoperable with non-geospatial registries

88 The purpose of this specification is to define an extension to OASIS ebXML RegRep  
 89 supporting publish, discovery and governance of geospatial and related non-geospatial  
 90 information items in order to meet the broader needs of the OGC community.

## 91 **i. Submitting organizations**

92 The following organizations submitted this Implementation Specification to the Open  
 93 Geospatial Consortium Inc.:

- 94 a) Wellfleet Software
- 95 b) CubeWerx
- 96 b) 4CT/kZen
- 97 c) ERDAS (was IONIC Software s.a.)

## 98 **iv. Submission contact points**

99 All questions regarding this submission should be directed to the editor or the submitters:

CONTACT	COMPANY	EMAIL
Farrukh Najmi (Editor)	Wellfleet Software	<a href="mailto:farrukh@wellfleetsoftware.com">farrukh@wellfleetsoftware.com</a>
Panagiotis (Peter) A. Vretanos (Editor)	CubeWerx	<a href="mailto:pvretano@cubewerx.com">pvretano@cubewerx.com</a>
Jef Vanbockryck	4CT/kZen	<a href="mailto:jef@kzenlabs.com">jef@kzenlabs.com</a>

5 **OGC 09-165r3**

Frédéric Houbie	ERDAS	<a href="mailto:Frederic.Houbie@erdas.com">Frederic.Houbie@erdas.com</a>

100 **v. Revision history**

Date	Release	Author	Paragraph modified	Description
2009-11-19	0.1.0	Farrukh Najmi	All	First version of Discussion Paper.
2009-11-25	0.1.1	Farrukh Najmi	2.1, 3.1, 3.2, 3.4, 3.5	Added support for spatial queries via REST interface, spatial encoding in WKT, temporal encoding in ISO 8601, server capability discovery, function support in OGC Filter
2010-03-31	0.1.2	Farrukh Najmi	All	<ul style="list-style-type: none"> <li>Reformatted document as an application profile specification</li> <li>Reorganized sections to support common clauses followed by clauses specific to RegRep 4 and RegRep 3</li> </ul>

101 **vi. Changes to the OGC® Abstract Specification**

102 The OGC® Abstract Specification does not require changes to accommodate this OGC®  
 103 standard.

## 104 **Foreword**

105 This specification defines an extension profile for the OASIS ebXML RegRep  
106 specifications for the purpose of adding spatio-temporal capabilities to the OASIS  
107 ebXML RegRep specifications.

108 This discussion paper has the following broad goals:

- 109 ● To add spatio-temporal extensions to ebXML RegRep specifications to allow  
110 them to meet the needs of the geospatial community
- 111 ● To provide extensions to both version 4.0 and 3.0 of ebXML RegRep  
112 specifications

113 This paper builds upon past work done in OGC cat-ebrim SWG. It also depends upon  
114 several OGC specifications foremost being [GML].

## 115 Introduction

116 This specification defines spatio-temporal extension profiles for ebXML RegRep 4 and  
117 ebXML RegRep 3.

118 The specification builds upon existing OGC and ISO standards. This specification  
119 leverages the work of the csw-ebRim specification where applicable.

120 This remainder of this document is structured as follows:

- 121 ● [Clause 6](#) specifies the various encodings for spatial data
- 122 ● [Clause 7](#) specifies the various encodings for temporal data
- 123 ● [Clause 8](#) specifies the extension profile for ebXML RegRep 4 specifications
- 124 ● [Clause 9](#) specifies the extension profile for ebXML RegRep 3 specifications
- 125 ● Appendices – specifies various details (e.g. XML listings of canonical ebRIM  
126 data) referenced within this document (TBD)

### 127 1 Scope

128 This document describes how ebXML RegRep 4.0 and 3.0 specifications can be extended  
129 to provide the following spatio-temporal capabilities:

- 130 ● Publishing of RegistryObjects with spatial attributes using SOAP bindings of  
131 [ebRS]
- 132 ● Publishing of RegistryObjects with temporal attributes using SOAP bindings of  
133 [ebRS]
- 134 ● Querying of RegistryObjects by spatial attributes using SOAP or REST bindings  
135 of [ebRS]
- 136 ● Querying of RegistryObjects by temporal attributes SOAP or REST bindings of  
137 [ebRS]
- 138 ● A minimal set of canonical functions that provide spatio-temporal search  
139 capability to parameterized queries

140 The document assumes that all other base features of ebXML RegRep shall be leveraged  
141 “as is”.



## 142 2 Conformance

143 An implementation compliant with to this specification shall meet the following  
144 conformance requirements:

- 145 ● Implemenatation shall conform to Clause 6 Spatial Data Endcoing
- 146 ● Implemenatation shall conform to Clause 7 Temporal Data Endcoing
- 147 ● An ebXML RegRep 4 implementation shall conform to Clause 8 Extension  
148 Profile for ebXML RegRep 4
- 149 ● An ebXML RegRep 3 implementation shall conform to Clause 9 Extension  
150 Profile for ebXML RegRep 3

## 151 3 Normative References

152 The following normative documents contain provisions which, through reference in this  
153 text, constitute provisions of this OGC 09-165r1r3. For dated references, subsequent  
154 amendments to, or revisions of, any of these publications do not apply. However, parties  
155 to agreements based on this OGC 09-165r1r3 are encouraged to investigate the possibility  
156 of applying the most recent editions of the normative documents indicated below. For  
157 undated references, the latest edition of the normative document referred to applies.

- 158 ● [ebXML Registry Information Model Version 4.0 \[ebRIM\]](#)
- 159 ● [ebXML Registry Services and Protocols Version 4.0 \[ebRS\]](#)
- 160 ● [ebXML Registry Information Model Version 3.0 \[ebRIM3\]](#)
- 161 ● [ebXML Registry Services and Protocols Version 3.0 \[ebRS3\]](#)
- 162 ● [OpenGIS Geography Markup Language \(GML\) Encoding Standard 3.2.1 \[GML\]](#)
- 163 ● [OpenGIS Implementation Specification for Geographic information - Simple  
164 feature access - Part 1: Common architecture 1.2.0 \[SFA1\]](#)
- 165 ● [OpenGIS Implementation Specification for Geographic information - Simple  
166 feature access - Part 2: SQL option 1.2.0 \[SFA2\]](#)
- 167 ● [GML 3.1.1 simple features profile \(1.0.0\) \[SFP\]](#)
- 168 ● [ISO 19108:2002 Geographic information -- Temporal schema \[ISO19108\]](#)
- 169 ● [ISO 8601:2004 Data elements and interchange formats -- Information interchange  
170 -- Representation of dates and times \[ISO8601\]](#)
- 171 ● [CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW \(1.0.1\) \[CSW1\]](#)
- 172 ● [CSW-ebRIM Registry Service – Part 2: Basic extension package \(1.0.1\) \[CSW2\]](#)

173 **4 Terms and Definitions**

174 or the purposes of this specification, the definitions specified in Clause 4 of the [[OGC 05-](#)  
175 [008](#)] OGC® OWS Common Implementation Specification shall apply. In addition, the  
176 following terms and definitions apply:

- 177 ● server - An ebXML RegRep server that conforms to this specification
- 178 ● client - A client of an ebXML RegRep server that conforms to this specification

179 **5 Conventions**

180 **5.1 Symbols (and abbreviated terms)**

181 Some frequently used abbreviated terms:

- 182 API Application Program Interface
- 183 ISO International Organization for Standardization
- 184 OGC Open Geospatial Consortium (OGC®)
- 185 UML Unified Modeling Language
- 186 XML eXtended Markup Language

187 **5.2 XML Namespaces Used**

188 This following table lists the namespace prefixes defined and / or referenced by this  
189 specification.

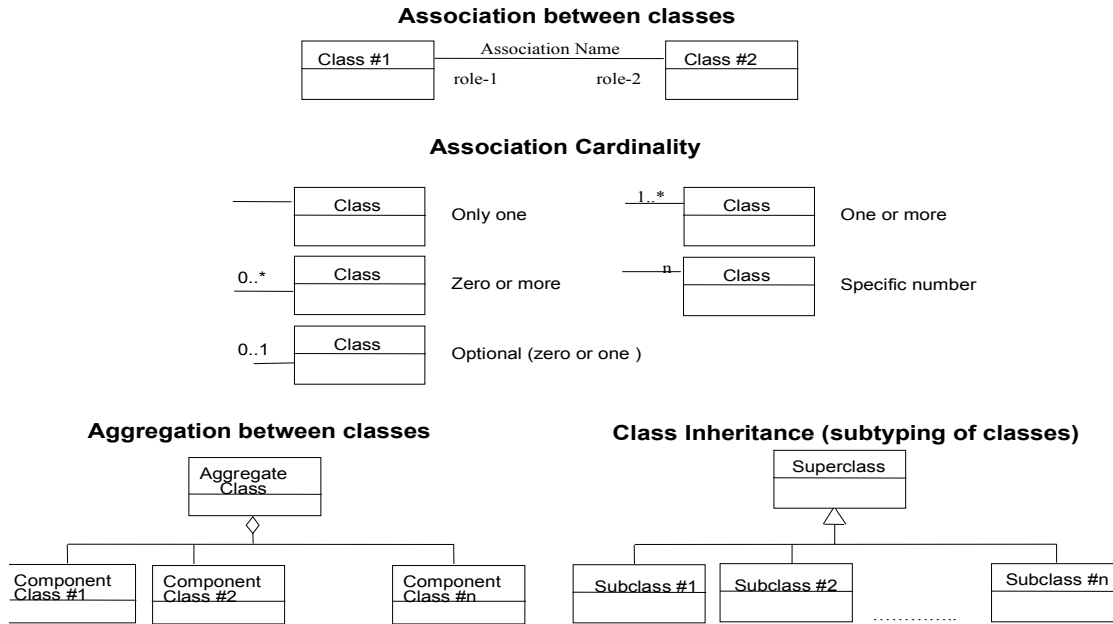
Namespace Prefix	Namespace URI	Defining Specification
enc	<a href="http://www.w3.org/2003/05/soap-encoding">http://www.w3.org/2003/05/soap-encoding</a>	A normative XML Schema [ <a href="#">XML Schema Part 1</a> ], [ <a href="#">XML Schema Part 2</a> ] document for the "http://www.w3.org/2003/05/soap-encoding" namespace can be found at <a href="http://www.w3.org/2003/05/soap-encoding">http://www.w3.org/2003/05/soap-encoding</a> .
env	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>	SOAP Version 1.2 Part 1.  A normative XML Schema [ <a href="#">XML Schema Part 1</a> ], [ <a href="#">XML Schema Part 2</a> ] document for the "http://www.w3.org/2003/05/soap-envelope" namespace can be found at <a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a> .
gml	<a href="http://www.opengis.net/gml/3.2">http://www.opengis.net/gml/3.2</a>	<a href="#">OpenGIS Geography Markup Language (GML) Encoding Standard 3.2.1 [GML]</a>
lcm	urn:oasis:names:tc:ebxml-regrep:xsd:lcm:4.0	ebXML RegRep Services and Protocols 4.0 (ebRS)
lcm3	urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0	ebXML RegRep Services and Protocols 3.0 (ebRS3)
mime	<a href="http://schemas.xmlsoap.org/wsdl/mime/">http://schemas.xmlsoap.org/wsdl/mime/</a>	WSDL namespace for WSDL MIME binding.
query	urn:oasis:names:tc:ebxml-regrep:xsd:query:4.0	ebXML RegRep Services and Protocols 4.0 (ebRS)
query3	urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0	ebXML RegRep Services and Protocols 3.0 (ebRS3)
rim	urn:oasis:names:tc:ebxml-	ebXML RegRep Registry

Namespace Prefix	Namespace URI	Defining Specification
	regrep:xsd:rim:4.0	Information Model 4.0 (ebRIM)
rim3	urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0	ebXML RegRep Registry Information Model 3.0 (ebRIM)
rs	urn:oasis:names:tc:ebxml-regrep:xsd:rs:4.0	ebXML RegRep Services and Protocols 4.0 (ebRS)
rs3	urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0	ebXML RegRep Services and Protocols 3.0 (ebRS)
wrs	http://www.opengis.net/cat/wrs/1.0	CSW-ebRIM profile (OGC 07-110r3)
wSDL	http://schemas.xmlsoap.org/wSDL/	WSDL 1.1 namespace defined by <a href="#">WSDL 1.1 specification</a> .
xacml	<a href="urn:oasis:names:tc:xacml:2.0:policy:os">urn:oasis:names:tc:xacml:2.0:policy:os</a>	<a href="#">XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0</a>
xacmlc	<a href="urn:oasis:names:tc:xacml:2.0:context:os">urn:oasis:names:tc:xacml:2.0:context:os</a>	<a href="#">XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0</a>
xlink	http://www.w3.org/1999/xlink	<a href="#">XML Linking Language (XLink) Version 1.1</a>
xs	http://www.w3.org/2001/XMLSchema	XML Schema [ <a href="#">XML Schema Part 1</a> ], [ <a href="#">XML Schema Part 2</a> ] specification
xsi	"http://www.w3.org/2001/XMLSchema-instance	W3C XML Schema specification [ <a href="#">XML Schema Part 1</a> ], [ <a href="#">XML Schema Part 2</a> ].

Table 1: Namespaces Used

### 5.3 UML Notation

192 The diagrams that appear in this standard are presented using the Unified Modeling  
193 Language (UML) static structure diagram.  
194 The UML notations used in this standard are described in the diagram below.



195 **Figure 1 — UML notation**

## 196 6 Spatial Data Encoding

197 This clause defines the various encoding schemes a server shall support for representing  
198 spatial data.

### 199 6.1 Spatial Data GML Encoding

200 The following requirements are defined for representing spatial data using GML  
201 encoding:

- 202 ● A client shall publish spatial data to the server using GML encoding such that the  
203 GML element is contained within rim:Slot elements. The details of encoding  
204 GML elements into rim:Slot elements is described later in this specification for  
205 each supported version of ebXML RegRep specification
- 206 ● Spatial data using GML encoding shall be represented as elements within the  
207 gml:AbstractGeometry substitution group as defined by [GML]

13 **OGC 09-165r3**

- 208 ● A server shall support the gml:Point, gml:LineString, gml:Curve, gml:Polygon,  
209 gml:Surface, gml:MultiPoint, gml:MultiCurve, gml:MultiSurface and  
210 gml:Geometry with the same restrictions as defined in table 6 of [SFP]
- 211 ● A server may support any other elements within the gml:AbstractGeometry  
212 substitution group for representing spatial data

213 **6.2 Spatial Data WKT Encoding**

214 The following requirements are defined for representing spatial data using WKT  
215 encoding:

216 The WKT format is specified in section 7 “Well-known Text Representation for  
217 Geometry” in [SFA1].

- 218 ● A client should specify spatial values (such as geometry parameter of the  
219 spatialFilter function) in WKT format when invoking queries to the server. The  
220 details of encoding WKT geometry in queries is described later in this  
221 specification for each supported version of ebXML RegRep specification

222 The following is an example of a polygon in WKT format:

223 `POLYGON((59 22, 78 22,78 38, 59 38, 59 22))`

224 **7 Temporal Data Encoding**

225 This clause defines the various encoding schemes a server shall support for representing  
226 temporal data.

227 **7.1 Temporal Data GML Encoding**

228 The following requirements are defined for representing temporal data using GML  
229 encoding:

- 230 ● A client shall publish temporal data to the server using GML encoding such that  
231 the GML element is contained within rim:Slot elements. The details of encoding  
232 GML elements into rim:Slot elements is described later in this specification for  
233 each supported version of ebXML RegRep specification
- 234 ● Temporal data within ebRIM encoding shall be represented as elements within the  
235 gml:AbstractTimeObject substitution group as defined by [GML]
- 236 ● A server shall support the gml:TimeInstant and gml:TimePeriod elements for  
237 representing temporal data
- 238 ● A server may support any other elements within the gml:AbstractTimeObject  
239 substitution group for representing temporal data

## 240 7.2 Temporal Data ISO8601 Encoding

241 The following requirements are defined for representing temporal data using [[ISO8601](#)]  
242 encoding:

- 243 ● A client should specify temporal values (such as timePeriod parameter of the  
244 temporalFilter function) in [[ISO8601](#)] format when invoking queries to the server.  
245 The details of encoding ISO8601 temporal values in queries is described later in  
246 this specification for each supported version of ebXML RegRep specification
- 247 ● When using [[ISO8601](#)] representation a time period shall be represented using the  
248 <starttime>/<endtime> format.

249 The following is an example of a timePeriod in WKT format:

```
250 2007-03-01T13:00:00Z/2008-05-11T15:30:00Z
```

## 251 8 Extension Profile for ebXML RegRep 4

252 This clause defines the spatio-temporal extension profile specific to ebXML RegRep 4.

### 253 8.1 Encoding Spatio-temporal Data in Slots

254 GML elements shall be encoded within eBRIM as specialized spatio-temporal Slot  
255 elements. A RegistryObjectType instance may have any number of spatio-temporal  
256 attributes, where each such attribute is represented by a spatio-temporal Slot element.

257 The following requirements are defined for encoding spatial and temporal GML values  
258 within a rim:RegistryObject/rim:Slot:

- 259 ● The Slot/@name attribute of the shall identify the spatio-temporal attribute within  
260 the RegistryObject
- 261 ● The Slot/@dataType attribute value shall be reference to a leaf GML DataType  
262 within the canonical DataType ClassificationScheme extended for GML types as  
263 defined by clause 8.3 DataTypes by [[CSW2](#)]
- 264 ● The GML element shall be an immediate sub-element of rim:ValueListItem sub-  
265 element of the Slot where the xsi:type attribute shall be rim:AnyValueType
- 266 ● The GML element shall be valid according to the XML Schema encoding defined  
267 by [[GML](#)]
- 268 ● If no srsName is defined anywhere in the GML element then the server shall  
269 assume a default value of “urn:ogc:def:crs:OGC:2:WGS84”

15 **OGC 09-165r3**

270 The following example shows how a gml:Envelope value is defined as a Slot with name  
271 geographicBoundingBox and dataType that identifies it as a gml:Envelope

```
272 <rim:RegistryObject xsi:type="ExtrinsicObjectType" ...>  
273   <rim:Slot  
274     name="geographicBoundingBox"  
275     dataType="urn:ogc:def:dataType:ISO-19107:GM_Envelope">  
276     <rim:ValueList>  
277       <rim:ValueListItem xsi:type="rim:AnyValueType">  
278         <gml:Envelope srsName="urn:ogc:def:crs:OGC:2:WGS84">  
279           <gml:lowerCorner>0 0</gml:lowerCorner>  
280           <gml:upperCorner>30 30</gml:upperCorner>  
281         </gml:Envelope>  
282       </rim:ValueListItem>  
283     </rim:ValueList>  
284   </rim:Slot>  
285   ...  
286 </rim:RegistryObject>
```

287

288 **8.2 Publishing RegistryObjects With Spatio-Temporal Data**

289 The following requirements are defined for publishing RegistryObjectType instances  
290 with spatio-temporal data to the server:

- 291 ● A server shall support the publishing of RegistryObjectType instances with  
292 spatio-temporal data encoded as GML elements within spatio-temporal Slots as  
293 described earlier using the SOAP binding for standard SubmitObjects protocol  
294 defined by [ebRS]
- 295 ● A server shall persist such GML data along with the parent RegistryObjectType  
296 instance and faithfully return it as part of the RegistryObjectType instance when a  
297 client retrieves the object using the standard Query protocol defined by [ebRS].

298 **8.3 Querying RegistryObjects Using Spatio-temporal Data**

299 This section describes how a client may query a server to discover RegistryObjectType  
300 instances based upon spatial and temporal filters.

301 The [ebRS] specification defines a set of canonical functions, their parameters, their  
302 semantics and how they may be used within queries and query parameters.

303 This profile defines several additional canonical functions that shall be supported by a  
304 server implementing this profile. Table 2 summarizes the functions defined by this  
305 profile. Subsequent sections specify them in detail.

306 The canonical functions defined in this section, along with encoding of spatio-temporal  
307 within spatio-temporal Slots in RegistryObjectType instances described earlier, together  
308 enable spatio-temporal search capability in standard [ebRS] queries.

309 A client may use these functions within the static part of a query expression or within the  
310 value of a parameter to a parameterized query. A server shall process the functions



311 according to their behavior as specified in this section. The function processing model is  
 312 specified by [ebRS].

313 A client shall use the “ogcp:” namespace prefix when using a canonical function defined  
 314 by this profile.

Function Name	Semantics
spatialFilter	Returns a list of ids of all RegistryObjectType instances that match the specified spatial filter criteria
temporalFilter	Returns a list of ids of all RegistryObjectType instances that match the specified temporal filter criteria

315 **Table 2: Canonical Functions Defined By This Profile**

316 **8.3.1 Canonical Function: spatialFilter**

317 This canonical function takes a spatial property name, spatial operator and spatial filter  
 318 geometry as parameter and returns all RegistryObjects that have a spatial property  
 319 matching the specified spatial filter.

320 **8.3.1.1 Parameter Summary**

Parameter	Description	Data Type
propertyName	The value of this parameter shall match the name attribute of a spatial Slot. A value of “*” indicates that the function should match any slot that has a geometry value that matches the specified filter	string
spatialOperator	The value of this parameter shall identify a spatial operator for the spatial filter. The value shall be one of the following: <i>contains, crosses, disjoint, equals, intersects, overlaps, relates, touches, within</i> . These values are based upon the spatial relationship routines defined in 7.2.8.1 of [SFA2] and have the same semantics	string
geometry	The value of this parameter specifies a spatial geometry. This value is used as a geometry filter for the spatial search. As server shall support the following formats for this parameter: <ul style="list-style-type: none"> <li>● GML encoding as specified in <a href="#">GML Encoding</a></li> <li>● WKT representation as specified in <a href="#">Spatial Data</a></li> </ul>	string

	<a href="#">WKT Encoding</a>	
crs	<p>The value of this parameter specifies the Coordinate Reference System by its URI.</p> <ul style="list-style-type: none"> <li>• This parameter is optional. A value of <i>null</i> indicates that the parameter value is unspecified</li> <li>• The default value of this parameter is “urn:ogc:def:crs:OGC:1.3:CRS84” representing EPSG:4326</li> <li>• The crs value (specified or default) shall be used by the server when geometry parameter is in WKT format</li> <li>• The crs value (specified or default) shall be used by the server when geometry parameter is in GML format and no srsName is specified within the GML element</li> </ul>	string
delimiter	The value of this parameter specifies the delimiter string to be used as separator between the tokens representing the ids matched by the function	string
template	The value of this parameter specifies a template to contain each id returned by the function. The template may contain one or more occurrences of template parameter string “\${id}” as placeholder for the id of a matched RegistryObjectType instance	string

321 **8.3.1.2 Function Semantics**

- 322       • The server shall return a string if the query is processed without any exceptions
- 323       • The string shall be “ebrs:null” if no RegistryObjectType instance is found to
- 324       match the specified spatial filter using the specified spatial operator
- 325       • The string shall consist of RegistryObjectType ids separated by the appropriate
- 326       delimiter character when RegistryObjectType instances are found to match the
- 327       specified spatial filter using the specified spatial operator

328 The following example shows the use of the spatialFilter function within the parameter of

329 the canonical parameterized query FindAssociatedObjects which is defined within

330 [ebRS]. The query uses the spatial filter (using WKT format) to get a list of ids of

331 Organizations that have a headquarter *within* a specified gml:Envelope and returns all

332 Persons who have an Employee association type with those Organizations.

```

333 <query:QueryRequest ...>
334   ...
335   <query:Query queryDefinition="urn:oasis:names:tc:ebxml-
336   regrep:query:FindAssociatedObjects">
337     <rim:Slot name="associationType">
338       <rim:ValueList>
339         <rim:ValueListItem xsi:type="StringValue"
340         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
341           <rim:Value>urn:oasis:names:tc:ebxml-
342   regrep:AssociationType:AffiliatedWith:EmployeeOf</rim:Value>
343         </rim:ValueListItem>
344       </rim:ValueList>
345     </rim:Slot>
346     ... sourceObjectType=Person omitted for brevity ...
347     ... targetObjectType=Organization omitted for brevity ...
348     <rim:Slot name="targetObjectId">
349       <rim:ValueList>
350         <rim:ValueListItem xsi:type="StringValue"
351         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
352           <rim:Value>
353   ogcp:spatialFilter (
354     "headquarters",
355     "within",
356     "POLYGON((59 22, 78 22,78 38, 59 38, 59 22))",
357     "urn:ogc:def:crs:OGC:1.3:CRS84",
358     ",",
359     "${id}"
360   )
361     #@@#
362     </rim:Value>
363   </rim:ValueListItem>
364   </rim:ValueList>
365   </rim:Slot>
366 </query:Query>
367 </query:QueryRequest>

```

368 The following example shows an ad hoc spatial query using EJBQL query language that  
369 finds all ExtrinsicObjectType instances (representing for example, land parcels for an  
370 Organization's headquarters) that touches a bounding box represented as a filter by the  
371 specified GML envelope:

```

372 <query:QueryRequest ...>
373   ...
374   <query:Query queryDefinition="urn:oasis:names:tc:ebxml-
375   regrep:query:AdhocQuery">
376     <rim:Slot name="queryLanguage">
377       <rim:ValueList>
378         <rim:ValueListItem xsi:type="StringValue"
379         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
380           <rim:Value>urn:oasis:names:tc:ebxml-
381   regrep:QueryLanguage:EJBQL</rim:Value>
382         </rim:ValueListItem>
383       </rim:ValueList>
384     </rim:Slot>
385     <rim:Slot name="queryExpression">
386       <rim:ValueList>

```

19  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412

```

    <rim:ValueListItem xsi:type="StringValueType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <rim:Value>
#@1
SELECT Object(eo) FROM ExtrinsicObjectType eo WHERE eo.id IN (
@#
ogcp:spatialFilter (
    "headquarters",
    "touches",
    "<gml:Envelope srsName=\"urn:ogc:def:crs:OGC:2:WGS84\">
      <gml:lowerCorner>0 0</gml:lowerCorner>
      <gml:upperCorner>30 30</gml:upperCorner>
    </gml:Envelope>",
    null,
    ",",
    "${id}"
)
#@
)
@#
    </rim:Value>
  </rim:ValueListItem>
</rim:ValueList>
</rim:Slot>

</query:Query>
</query:QueryRequest>

```

413 **8.3.1 Canonical Function: temporalFilter**

414 This canonical function takes a temporal property name, spatial operator and spatial filter  
415 geometry as parameter and returns all RegistryObjects that have a spatial property  
416 matching the specified spatial filter.

417 **8.3.1.1 Parameter Summary**

Parameter	Description	Data Type
propertyName	The value of this parameter shall match the name attribute of a temporal Slot. A value of "*" indicates that the function should match any slot that has a time instant or interval value that is within the specified filter	string
temporalOperator	The value of this parameter shall identify a temporal comparison operator for the temporal filter. The value shall be one of the following: <i>After, Before, Begins, BegunBy, Contains, During, EndedBy, Ends, Equals, Meets, MetBy, Overlaps, OverlappedBy</i> . These temporal operations and their semantics are defined by [ISO19108]	string
timePeriod	The value of this parameter specifies a time period. This	string

20 <sup>1</sup>The #@ and @# delimiters are defined by ebRS to delimit non-function parts of a query expression from function  
21 parts

	<p>value is used as a temporal filter for the temporal search. A server shall support the following formats for this parameter:</p> <ul style="list-style-type: none"> <li>● GML representation as specified Temporal Data Representation in GML in <a href="#">Temporal Data GML Encoding</a></li> <li>● ISO 8601 representation as specified in <a href="#">Temporal Data ISO 8601 Encoding</a></li> </ul>	
delimiter	The value of this parameter specifies the delimiter string to be used as separator between the tokens representing the ids matched by the function	string
template	The value of this parameter specifies a template to contain each id returned by the function. The template may contain one or more occurrences of template parameter string “\${id}” as placeholder for the id of a matched RegistryObjectType instance	string

#### 418 8.3.1.2 Function Semantics

- 419 ● The server shall return a string if the query is processed without any exceptions
- 420 ● The string shall be “ebrs:null” if no RegistryObject is found to match the
- 421 specified temporal filter using the specified parameters
- 422 ● The string shall consist of RegistryObject ids separated by the appropriate
- 423 delimiter character when RegistryObjects are found to match the specified
- 424 temporal filter for temporal slots with name

425 The following example shows the use of the temporalFilter function within the parameter

426 of the canonical parameterized query FindAssociatedObjects which is defined within

427 [ebRS]. The query uses the spatial filter to get a list of ids of Organizations that have

428 were incorporated within a time interval (the year 2009) specified using ISO 8601

429 representation, and returns all Persons who have an Employee association type with those

430 Organizations.

```

431 <query:QueryRequest ...>
432   ...
433   <query:Query queryDefinition="urn:oasis:names:tc:ebxml-
434   regrep:query:FindAssociatedObjects">
435     <rim:Slot name="associationType">
436       <rim:ValueList>
437         <rim:ValueListItem xsi:type="StringValue"
438         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
439           <rim:Value>urn:oasis:names:tc:ebxml-
440   regrep:AssociationType:AffiliatedWith:EmployeeOf</rim:Value>

```

```

441     </rim:ValueListItem>
442   </rim:ValueList>
443 </rim:Slot>

444   ... sourceObjectType=Person omitted for brevity ...
445   ... targetObjectType=Organization omitted for brevity ...

446   <rim:Slot name="targetObjectId">
447     <rim:ValueList>
448       <rim:ValueListItem xsi:type="StringValue"
449   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
450         <rim:Value>
451   ogcp:temporalFilter (
452     "dateOfIncorporation",
453     "During",
454     "2009-01-01T00:00-00:00/2009-12-31T23:59-00:00"
455     ","
456   )
457     #@@#
458     </rim:Value>
459   </rim:ValueListItem>
460   </rim:ValueList>
461 </rim:Slot>

462 </query:Query>
463 </query:QueryRequest>

```

#### 464 8.3.1.3 Temporal Filter Open Issues:

- 465 ● Should we allow a timeInstant as an alternate temporal filter. Currently only time  
466 interval is allowed (adds complexity)
- 467 ● Should we allow temporal property to be a time interval (with a start and end time  
468 instant). Current it is only allowed to be a single time instant (adds complexity)
- 469 ● Not all temporal operators make sense depending upon what is allowed for  
470 datatype of temporal property and datatype for temporal filter. Need to make the  
471 spec tighter in this regard
- 472 ● Should we allow GML as format when publishing temporal data. Restricting to  
473 ISO8601 format simplifies implementation quite a bit. What does GML buy in  
474 terms of functionality?

#### 475 8.3.2 Supporting Canonical Functions in OGC Filter Queries

476 A server may support OGC Filter query syntax as a supported query expression syntax.  
477 The spatio-temporal canonical functions may be embedded within OFC Filter query  
478 expressions using the template parameter of the canonical functions. The following is an  
479 example of what this using OGC Filter query expressions with embedded spatio-temporal  
480 functions:

```

481 <query:QueryRequest ...>
482   ...
483   <query:Query
484 queryDefinition="urn:oasis:names:tc:ebxmlregrep:query:AdhocQuery">
485     <rim:Slot name="queryLanguage">
486       <rim:ValueList>
487         <rim:ValueListItem xsi:type="StringValueType"
488           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
489 <rim:Value>urn:oasis:names:tc:ebxmlregrep:QueryLanguage:CSW-
490 Query</rim:Value>
491         </rim:ValueListItem>
492       </rim:ValueList>
493     </rim:Slot>
494     <rim:Slot name="queryExpression">
495       <rim:ValueList>
496         <rim:ValueListItem xsi:type="StringValueType"
497           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
498 <rim:Value>
499 #@
500 <csw:GetRecords ...>
501   <csw:Query typeName="csw:Record">
502     <csw:ElementSetName>brief</csw:ElementSetName>
503     <csw:Constraint version="1.1.0">
504       <ogc:Filter>
505 @#
506 ogcp:spatialFilter("headquarters", "contains", "<gml:Envelope
507 srsName=\"urn:ogc:def:crs:OGC:2:WGS84\"><gml:lowerCorner>0
508 0</gml:lowerCorner><gml:upperCorner>30
509 30</gml:upperCorner></gml:Envelope>", NULL, "<RecordId rid=\"\${id}\"/>")
510 #@
511       </ogc:Filter>
512     </csw:Constraint>
513   </csw:Query>
514 </csw:GetRecords>
515 @#
516       </rim:Value>
517     </rim:ValueListItem>
518   </rim:ValueList>
519 </rim:Slot>
520 </query:Query>
521 </query:QueryRequest>

```

### 522 8.3.3 Invoking Spatio-Temporal Queries Via REST Interface

523 A server shall support a REST interface as defined by [ebRS]. This section defines how a  
524 client may invoke spatio-temporal queries using the REST interface of [ebRS]. A server  
525 shall support function invocation within parameter values of a REST URL as specified by  
526 [ebRS]. In particular a server shall support the invocation of canonical functions defined  
527 by this specification via the REST interface.

528 The following RESTful GET operation<sup>2</sup> uses the canonical FindAssociatedObjects query  
529 to find all Persons that are employees of all organizations that have a headquarter within  
530 the geometry specified by the WKT Polygon:

25 <sup>2</sup>The URL is shown without required URL encoding for clarity of understanding

```

531 GET http://acme.com/myregistry/search?queryId=urn:oasis:names:tc:ebxml-
532 regrep:query:FindAssociatedObjects&associationType=
533 urn:oasis:names:tc:ebxml-
534 regrep:AssociationType:AffiliatedWith:EmployeeOf&targetObjectId=ogc:spa
535 tialFilter("headquarters", "within", "POLYGON((59 22, 78 22, 78 38, 59
536 38, 59 22))", "urn:ogc:def:crs:OGC:1.3:CRS84", "", "${id}")#@@#

```

537 The following RESTful GET operation uses the canonical FindAssociatedObjects query  
 538 to find all Persons that employees of all Organizations that were incorporated in specified  
 539 time period:

```

540 GET http://acme.com/myregistry/search?queryId=urn:oasis:names:tc:ebxml-
541 regrep:query:FindAssociatedObjects&associationType=
542 urn:oasis:names:tc:ebxml-
543 regrep:AssociationType:AffiliatedWith:EmployeeOf&targetObjectId=ogc:tem
544 poralFilter("dateOfIncorporation", "During", "2009-01-01T00:00-
545 00:00/2009-12-31T23:59-00:00)", "", "${id}")#@@#

```

#### 546 8.4 Discovering a Server's Spatio-temporal Capabilities

547 This section defines how a server shall advertise its spatio-temporal capabilities so a  
 548 client may determine whether it is compatible with the server or not.

549 The following requirements are defined for a server to advertise its spatio-temporal  
 550 capabilities:

- 551 ● A server shall advertise its capabilities using a RegistryType instances  
 552 representing the server as defined by [ebRIM]
- 553 ● The RegistryType instance shall have a canonical Slot declaring support for a  
 554 specific version of this profile specification as follows (assumes version is 1.0):

```

555 <rim:RegistryObject xsi:type="RegistryType" ...>
556   <rim:Slot
557     name="urn:ogc:def:ebxml:regrep:4.0:profile:spatio-temporal:version"
558     dataType="urn:oasis:names:tc:ebxml-regrep:DataType:String">
559     <rim:ValueList>
560       <rim:ValueListItem xsi:type="rim:StringValueType">
561         <rim:Value>1.0</rim:Value>
562       </rim:ValueListItem>
563     </rim:ValueList>
564   </rim:Slot>
565   ...
566 </rim:RegistryObject>

```

- 567 ● The RegistryType instance shall have a canonical Slot declaring support for  
 568 specific versions of spatial encodings as follows (assumes support for GML 3.1 ,  
 569 GML 3.2 and WKT):

```

570 <rim:RegistryObject xsi:type="RegistryType" ...>
571   <rim:Slot

```



27  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588

```

    name="urn:ogc:def:ebxml:regrep:4.0:profile:spatio-
temporal:encoding:spatial"
    dataType="urn:oasis:names:tc:ebxml-regrep:DataType:String">
    <rim:ValueList>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>http://www.opengis.net/gml/3.1</rim:Value>
      </rim:ValueListItem>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>http://www.opengis.net/gml/3.2</rim:Value>
      </rim:ValueListItem>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>WKT</rim:Value>
      </rim:ValueListItem>
    </rim:ValueList>
  </rim:Slot>
  ...
</rim:RegistryObject>

```

- 589           ○ GML encoding's version shall be identified by the namespace for its XML  
590           schema encoding
- 591           ○ WKT encoding shall be identified by “WKT”<sup>3</sup>
- 592           ○ Other spatial encoding shall be identified as defined by extensions of this  
593           specification or by deployments and implementations
- 594           ● The RegistryType instance shall have a canonical Slot declaring support for  
595           specific versions of temporal encodings as follows (assumes support for GML 3.1  
596           , GML 3.2 and ISO8601):

597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615

```

<rim:RegistryObject xsi:type="RegistryType" ...>
  <rim:Slot
    name="urn:ogc:def:ebxml:regrep:4.0:profile:spatio-
temporal:encoding:temporal"
    dataType="urn:oasis:names:tc:ebxml-regrep:DataType:String">
    <rim:ValueList>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>http://www.opengis.net/gml/3.1</rim:Value>
      </rim:ValueListItem>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>http://www.opengis.net/gml/3.2</rim:Value>
      </rim:ValueListItem>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>urn:ogc:def:trs:ISO-8601:::</rim:Value>
      </rim:ValueListItem>
    </rim:ValueList>
  </rim:Slot>
  ...
</rim:RegistryObject>

```

- 616           ○ GML encoding's version shall be identified by the namespace for its XML  
617           schema encoding
- 618           ○ ISO8601 encoding shall be identified by “urn:ogc:def:trs:ISO-8601:::”<sup>4</sup>

28 <sup>3</sup>Issue: Is there a URN we could use that has WKT version in it. Nothing found in the [OGC urn resolver?](#)

29 <sup>4</sup>This is based on [OGC naming authority entry](#)

30 **OGC 09-165r3**  
619       ○ Other temporal encoding shall be identified as defined by extensions of this  
620           specification or by deployments and implementations

## 621 **9 Extension Profile for ebXML RegRep 3**

### 622 **9.1 Encoding Spatio-temporal Data in Slots**

623 GML elements shall be encoded within ebRIM3 as specialized spatio-temporal Slot  
624 elements. A RegistryObjectType instance may have any number of spatio-temporal  
625 attributes, where each such attribute is represented by a spatio-temporal Slot element.

626 The following requirements are defined for encoding spatial and temporal GML values  
627 within a rim3:RegistryObject/rim3:Slot:

- 628       ● The Slot/@name attribute of the shall identify the spatio-temporal attribute within  
629           the RegistryObject
- 630       ● The Slot/@dataType attribute value shall be reference to a leaf GML DataType  
631           within the canonical DataType ClassificationScheme extended for GML types as  
632           defined by clause 8.3 DataTypes by [CSW2]
- 633       ● The GML element shall be an immediate sub-element of  
634           wrs:ValueList/wrs:AnyValue sub-element of the Slot
- 635       ● The GML element shall be valid according to the XML Schema encoding defined  
636           by [GML]
- 637       ● If no srsName is defined anywhere in the GML element then the server shall  
638           assume a default value of “urn:ogc:def:crs:OGC:2:WGS84”

639 The following example shows how a gml:Envelope value is defined as a Slot with name  
640 geographicBoundingBox and dataType that identifies it as a gml:Envelope

```
641 <rim3:ExtrinsicObject ...>  
642   <rim3:Slot  
643     name="geographicBoundingBox"  
644     dataType="urn:ogc:def:dataType:ISO-19107:GM_Envelope">  
645     <wrs:ValueList>  
646       <wrs:AnyValue>  
647         <gml:Envelope srsName="urn:ogc:def:crs:OGC:2:WGS84">  
648           <gml:lowerCorner>0 0</gml:lowerCorner>  
649           <gml:upperCorner>30 30</gml:upperCorner>  
650         </gml:Envelope>  
651       </wrs:AnyValue>  
652     </wrs:ValueList>  
653   </rim3:Slot>  
654   ...  
655 </rim3:ExtrinsicObject>
```

656

## 9.2 Publishing RegistryObjects With Spatio-Temporal Data

658 The following requirements are defined for publishing RegistryObject instances with  
659 spatio-temporal data to the server:

- 660 ● A server shall support the publishing of RegistryObjects with spatio-temporal data  
661 encoded as GML elements within spatio-temporal Slots as described earlier using  
662 the SOAP binding for standard SubmitObjects protocol defined by [ebRS]
- 663 ● A server shall persist such GML data along with the parent RegistryObjectType  
664 instance and faithfully return it as part of the RegistryObjectType instance when a  
665 client retrieves the object using the standard Query protocol defined by [ebRS].

## 9.3 Querying RegistryObjects Using Spatio-temporal Data

667 Unlike RegRep 4, the RegRep 3 specifications do not define a query functions feature.  
668 The query functions feature from RegRep 4 has been specified in [Annex B](#) of this  
669 specification. The following requirements are defined for querying RegistryObjects using  
670 spatio temporal data:

- 671 ● A RegRep 3 server shall support the query functions feature specified by in  
672 [Annex B](#) of this specification
- 673 ● A RegRep 3 server shall support the spatialFilter and temporalFilter canonical  
674 functions as specified for the RegRep 4 extension profile using the canonical  
675 functions feature specified in [Annex B](#) of this specification

## 9.4 Discovering a Server's Spatio-temporal Capabilities

677 This section defines how a server shall advertise its spatio-temporal capabilities so a  
678 client may determine whether it is compatible with the server or not.

679 The following requirements are defined for a server to advertise its spatio-temporal  
680 capabilities:

- 681 ● A server shall advertise its capabilities using a Registry instances representing the  
682 server as defined by [ebRIM3]
- 683 ● The Registry instance shall have a canonical Slot declaring support for a specific  
684 version of this profile specification as follows (assumes version is 1.0):

```
685 <rim3:Registry ...>  
686   <rim3:Slot  
687     name="urn:ogc:def:ebxml:regrep:4.0:profile:spatio-temporal:version"  
688     dataType="urn:oasis:names:tc:ebxml-regrep:DataType:String">  
689     <rim3:ValueList>  
690       <rim3:Value>1.0</rim3:Value>  
691     </rim3:ValueList>  
692   </rim3:Slot>  
693   ...  
694 </rim3:Registry>
```

### OGC 09-165r3

- 32
- 695 ● The Registry instance shall have a canonical Slot declaring support for specific
- 696 versions of spatial encodings as follows (assumes support for GML 3.1 , GML 3.2
- 697 and WKT):

```
698 <rim3:Registry ...>
699   <rim3:Slot
700     name="urn:ogc:def:ebxml:regrep:4.0:profile:spatio-
701     temporal:encoding:spatial"
702     dataType="urn:oasis:names:tc:ebxml-regrep:DataType:String">
703     <rim3:ValueList>
704       <rim3:Value>http://www.opengis.net/gml/3.1</rim3:Value>
705       <rim3:Value>http://www.opengis.net/gml/3.2</rim3:Value>
706       <rim3:Value>WKT</rim3:Value>
707     </rim3:ValueList>
708   </rim3:Slot>
709   ...
710 </rim3:Registry>
```

- 711 ○ GML encoding's version shall be identified by the namespace for its XML
- 712 schema encoding
- 713 ○ WKT encoding shall be identified by “WKT”<sup>5</sup>
- 714 ○ Other spatial encoding shall be identified as defined by extensions of this
- 715 specification or by deployments and implementations
- 716 ● The Registry instance shall have a canonical Slot declaring support for specific
- 717 versions of temporal encodings as follows (assumes support for GML 3.1 , GML
- 718 3.2 and ISO8601):

```
719 <rim3:Registry ...>
720   <rim3:Slot
721     name="urn:ogc:def:ebxml:regrep:4.0:profile:spatio-
722     temporal:encoding:temporal"
723     dataType="urn:oasis:names:tc:ebxml-regrep:DataType:String">
724     <rim3:ValueList>
725       <rim3:Value>http://www.opengis.net/gml/3.1</rim3:Value>
726       <rim3:Value>http://www.opengis.net/gml/3.2</rim3:Value>
727       <rim3:Value>urn:ogc:def:trs:ISO-8601:::</rim3:Value>
728     </rim3:ValueList>
729   </rim3:Slot>
730   ...
731 </rim3:RegistryObject>
```

- 732 ○ GML encoding's version shall be identified by the namespace for its XML
- 733 schema encoding
- 734 ○ ISO8601 encoding shall be identified by “urn:ogc:def:trs:ISO-8601:::”<sup>6</sup>
- 735 ○ Other temporal encoding shall be identified as defined by extensions of this
- 736 specification or by deployments and implementations

33 <sup>5</sup>Issue: Is there a URN we could use that has WKT version in it. Nothing found in the [OGC urn resolver?](#)

34 <sup>6</sup>This is based on [OGC naming authority entry](#)

## 737 **Annex A. Abstract Test Suite (normative)**

### 738 **A.1. TBD**

## 739 **Annex B. Query Functions (normative)**

740 This annex specifies the query functions feature from RegRep 4 by copying it nearly  
 741 verbatim with modification to align it with RegRep 3. The ebXML RegRep 3  
 742 specification lacks this feature so it is specified here in order to allow the “[Extension](#)  
 743 [Profile for RegRep 3](#)” in this specification to use this feature.

### 744 **B.1. Query Functions**

745 A server may support any number of pre-defined functions known as *Query Functions*,  
 746 that may be used within a query expression or query parameter. Query functions are  
 747 similar in concept to functions in SQL. Query functions may be used within the query  
 748 expression of a parameterized query as well as within its invocation parameter values.  
 749 Query functions enable parameterized queries to use reusable specialized search  
 750 algorithms to augment their capabilities.

751 This specification defines a number of [canonical functions](#) that are standard functions that  
 752 shall be supported by a server. Profiles, implementations and deployments may define  
 753 additional query functions beyond the canonical functions defined by this specification.

#### 754 **B.1.1. Using Functions in Query Expressions**

755 A parameterized query stored as a rim3:AdhocQuery instance may have a  
 756 rims:QueryExpression which defines defines a query expression within its sub-nodes. A  
 757 client may submit a rims:AdhocQuery such that its query expression may use any number  
 758 of query functions supported by the server any where within the query expression where  
 759 it is syntactically correct to use value returned by the function.

760 If a query expression contains one or more function invocations then the query  
 761 expression shall delimit the parts of the query expression that are not a function  
 762 invocation with the leading characters “#@” and trailing characters “@#”. This is similar  
 763 in syntax to a java multi-line comment syntax where a comment is delimited by leading  
 764 characters “/\*” and trailing characters “\*/”. The delimiters serve the following purposes:

- 765 ● Allows a parser to recognize the non-function parts of the query expression that  
 766 shall be preserved *as is*
- 767 ● Allows implementations to optimize by skipping function parsing and evaluation  
 768 if the special delimiter characters are not present in query expression

769 The following is an example of an SQL query expression which uses the  
 770 subClassificationNode function to match all RegistryObjects that are targets of  
 771 Association with specified sourceObject and type that a subnode of AffiliatedWith node  
 772 upto a depth of 2 levels in the descendant hierarchy. The delimiter characters are in bold  
 773 font while the function invocations is in bold and italic font below:

## OGC 09-165r3

```
36
774 --example of a query expression with query functions
775 #@SELECT targetObject.* FROM
776 RegistryObject targetObject, Association a WHERE
777
778 a.sourceObject = :sourceObject AND
779 a.type IN (@# subClassificationNode('urn:oasis:names:tc:ebxml-
780 regrep:AssociationType:AffiliatedWith', 2, ",") #@) AND
targetObject.id = a.targetObject@#
```

### 881 B.1.2. Using Functions in Query Parameters

882 A client may use query functions supported by a server within parameter values specified  
883 when invoking a parameterized query. A client may invoke a parameterized query using  
884 the AdhocQuery protocol such that its query parameter values may use any number of  
885 query functions supported by the server any where within the query parameter where it is  
886 syntactically correct to use value returned by the function.

887 If a query parameter value contains one or more function invocations then the query  
888 expression shall delimit the parts of the query parameter that are not a function invocation  
889 with the leading characters “#@” and trailing characters “@#”. If a query parameter  
890 value only has function invocations and contains no non-function parts then it must  
891 include at least one leading or trailing “#@@#” delimiter token pair to allow optimized  
892 parsing and evaluation of query functions only when needed.

893 The following is an example of a query expression that has no query functions. Its two  
894 parameters are shown in bold font:

```
895 --Following is the query expression within the server
896 --This time it has no query functions as they are in the query
897 parameters
898 SELECT targetObject.* FROM
899 RegistryObject targetObject, Association a WHERE
900
901 a.sourceObject = :sourceObject AND
902 a.type IN ( :types ) AND
903 targetObject.id = a.targetObject
```

### 904 B.1.3. Function Processing Model

905 A server shall meet the following function processing requirements during the processing  
906 of a QueryRequest:

- 907 ● When processing a query expression elements (rims:QueryExpression) the server  
908 SHOULD NOT perform function processing if the special delimiter sequences of  
909 “#@” and “@#” are not found in the query expression
- 910 ● When processing query invocation parameter elements the server SHOULD NOT  
911 perform function processing if the special delimiter sequences of “#@” and “@#”  
912 are not found in the query expression
- 913 ● When processing a query expression element if the special delimiter sequences of  
914 “#@” and “@#” are found then the server shall process query expression elements

37

815 to replace all function invocations with the value returned when the function is  
816 invoked with specified parameters

817 ● When processing query invocation parameter elements if the special delimiter  
818 sequences of “#@” and “@#” are found then the server shall process each query  
819 parameter element to replace all function invocations with the value returned  
820 when the function is invoked with specified parameters

821 ● When invoking a function that has another function invocation as its parameter  
822 the inner most functions shall be invoked first so that the outer function can be  
823 invoked with the value returned by the inner function invocation

824 ● When processing a query expression or query parameter the special delimiter  
825 characters “#@” and “@#” shall be removed and the value contained within them  
826 shall be preserved without any change

#### 827 B.1.4. Function Processor BNF

828 The following BNF grammar normatively describes the grammar for query expressions  
829 and query invocation parameters with embedded function invocations. The *start*  
830 production describes the grammar for query expressions and query invocation parameters  
831 with embedded function invocations.

```

832 <DEFAULT> SKIP : {
833 " "
834 | "\t"
835 | "\r"
836 | "\n"
837 }

838

839 <DEFAULT> TOKEN : {
840 <FLOAT: <INTEGER> "." <INTEGER> | "." <INTEGER> | <INTEGER> ".">
841 | <INTEGER: (<DIGIT>)+>
842 | <DIGIT: ["0"- "9"]>
843 | <BOOLEAN: "true" | "false">
844 }

845

846 <DEFAULT> TOKEN : {
847 <S_IDENTIFIER: (<LETTER>+ (<DIGIT> | <LETTER> | <SPECIAL_CHARS>)*>
848 | <#LETTER: ["a"- "z", "A"- "Z"]>
849 | <#SPECIAL_CHARS: " ">
850 | <S_CHAR_LITERAL: "\'" (~["\''])* \' ("\'" (~["\''])* \'')*>
851 | <S_QUOTED_IDENTIFIER: "\"" (~["\n", "\r", "\""])* "\"">
852 | <OPENPAREN: "(">
853 | <CLOSEPAREN: ")">
854 | <COMMA: ",">
855 | <COLON: ":">
856 | <DELIMITED_TEXT: "#@" (~["@"])* "@#">
857 }

858 start ::= ( textOrFunctionCall )+ <EOF>

```

38 **OGC 09-165r3**

```
859 text ::= ( ( <DELIMITED_TEXT> ) )
860 textOrFunctionCall ::= ( text | FunctionCall )
861 FunctionCall ::= FunctionReference <OPENPAREN>
862 ( FunctionArgumentList )* <CLOSEPAREN>
863 FunctionReference ::= <S_IDENTIFIER> <COLON> <S_IDENTIFIER>
864 FunctionArgumentList ::= FunctionArgument ( <COMMA>
865 FunctionArgument )*
866 FunctionArgument ::= ( FunctionCall | <S_CHAR_LITERAL> |
867 <S_QUOTED_IDENTIFIER> | <FLOAT> | <INTEGER> | <BOOLEAN> )
```

868 **B.2. Common Patterns In Query Functions**

869 This section defines some commonly occurring patterns in query functions and defines  
870 some common solutions to addressing these patterns. Profiles are should conform to the  
871 solutions defined in this section whenever possible.

872 **B.2.1. Specifying a null Value for string Param or Return Value**

873 A function that accepts a string parameter should treat a value of “ebrs:null” as a null  
874 string. A null string is a string whose value is unspecified.

875 When a function returns a “string” type it should return a null value string as the  
876 canonical value “ebrs:null”.