

---

# Client Attribute Requirements Markup Language (CARML) Specification

Working draft 03, 24 November 2006

**Document Identifier:**

IGF-CARML-spec-03

**Editor(s):**

Phil Hunt, Oracle

**Contributor(s):**

Prateek Mishra, Oracle  
Mark Wilcox, Oracle

**Subject/Keywords:**

Identity services, authority, client attribute requirements

**Related work:**

This specification is related to:

- IGF-AAPML-spec

**Abstract:**

Client Attribute Requirements Markup Language is a specification that allows applications to define their attribute requirements as it relates to identity. CARML can be used to automate configuration of identity attribute services and to expose the set of identity-related data consumed by a specific application or groups of applications.

**Status:**

This document is a working draft.

---

## Notices

Copyright © 2006, Oracle. All Rights Reserved.

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

---

# Table of Contents

1	Introduction .....	4
1.1	Terminology .....	4
1.2	Notation .....	4
1.3	Normative References.....	4
1.4	Non-Normative References .....	4
2	CARML Specification .....	5
2.1	Data Dictionaries, Schemas, and CARML.....	5
2.2	Subject Indexes .....	6
2.3	Properties .....	6
2.4	Attributes .....	7
2.5	CARML top-level Containers .....	7
2.5.1	Meta-data Elements.....	8
3	Schema .....	10
	Appendix A. Revision History .....	12

---

# 1 Introduction

CARML (pronounced “car – mull”) is a specification describing a declarative way for a client application to define the set of identity-related attribute data from an identity service provider. A CARML document is an XML document format that allows applications consuming identity-related data to declare identity data requirements and intended usage for personally identifiable information otherwise known as attributes, claims, and assertions. The information specified includes the desired operations and intended usage (indexes, read/write, sharing, privacy).

The identity service provider is a service that allows queries against user attributes based on a set of Subject Indexes. Subject Indexes refer to a set of information that, in a certain context, can be used to uniquely disambiguate and access some of the information associated with a user.

A CARML declaration may be used to determine whether an identity service can provide the attribute information required by an application and, further to appropriately configure the service to provide such information. This specification does not specify any particular identity service; the intent here is that CARML should be usable with a range of existing (e.g. JNDI, JDBC, JAAS) and new identity service providers.

## 1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

## 1.2 Notation

- The prefix `saml:` stands for the SAML 2.0 assertion namespace, `urn:oasis:names:tc:SAML:2.0:assertion`.
- The prefix `carml:` stands for the namespace `urn:igf:client:0.3:carml`

## 1.3 Normative References

- |                       |  |
|-----------------------|--|
| <b>[RFC2119]</b>      | S. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , <a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a> , IETF RFC 2119, March 1997.   |
| <b>[SAML2Core]</b>    | S. Cantor et al, <i>Assertions and Protocols for SAML 2.0</i> , Document identifier: <code>saml-core-2.0-os</code> . Available at: <a href="http://docs.oasis-open.org/security/saml/v2.0">http://docs.oasis-open.org/security/saml/v2.0</a>                                 |
| <b>[SAML2Profile]</b> | J. Hughes et al, <i>Profiles for OASIS Security Attribute Markup Language (SAML) V2.0</i> . Document identifier: <code>saml-profiles-2.0-os</code> . Available at: <a href="http://docs.oasis-open.org/security/saml/v2.0">http://docs.oasis-open.org/security/saml/v2.0</a> |

## 1.4 Non-Normative References

- |                       |  |
|-----------------------|--|
| <b>IGF Overview</b>   | P. Mishra and Phil Hunt, <i>High-level Overview of Identity Governance Framework</i> |
| <b>IGF-client-api</b> | Sample API and examples that use CARML   |

---

## 2 CARML Specification

41

42 A CARML-conformant identity service SHOULD generate appropriate responses and exceptions when  
43 interacting with an application based upon a CARML document. CARML is a specification describing a  
44 declarative way for a client application to define the set of identity-related attribute data required from a  
45 service provider. CARML does not guarantee that the client application will receive what is requested.  
46 CARML based applications are expected observe instructions and exceptions encoded in response  
47 messages when using an IGF Attribute Service.

48 In this specification, the assumption is that many applications only require a fixed set of *interactions*  
49 around identity data. Each interaction consists is usually a read operation but in some cases may involve  
50 update of some identity-related attributes.

51 Each interaction has the following properties:

- 52 i. One or more *indices* are used to lookup information about a user. Typically, this is some form  
53 of a “subject name” derived thru authentication but could also be an attribute such as a GUID  
54 or social security number.
- 55 ii. Tests for a set of boolean *properties* which test whether a subject has some property (with  
56 unknown value), a property with a value that will be known at run-time, or a property with a  
57 fixed value.
- 58 iii. Retrieval of values for a sequence of named properties, this corresponds to the traditional  
59 idea that an *attribute* is a name and value pair.
- 60 iv. Based on some form of “subject name”, one or more attribute name/value pairs can be  
61 modified that are associated with the subject.

62 A programmer with limited knowledge of LDAP, SAML and DB protocols and deployment scenarios,  
63 should be able to describe the application requirements for identity-related data and implement it using a  
64 simple application programming interface.

65 To define required attributes, CARML uses the syntax of <saml:Attribute> described in [SAML2Core] and  
66 attribute profiles provide [SAML2Profiles] for most of CARML. Any extensions required are also noted.

### 2.1 Data Dictionaries, Schemas, and CARML

67

68 CARML is intended to define an application’s identity information requirements from the perspective of  
69 the client application. CARML documents are intended to be defined without making a lot of assumptions  
70 of existing data dictionaries or schemas. One of the main features of CARML is to define a localized data  
71 dictionary for an application. In many situations, especially for commercial software developers, the  
72 deployment environment and its data dictionary or schema is unknown. Since the deployment  
73 environment is unknown, the developer can only follow industry “best practices” (which can be supported  
74 by IDE tools). However, even with the best of intentions, developers are forced to make difficult choices  
75 especially when there is disagreement on schema between vendors or implementations.

76 Another problem encountered with standard schemas is defining actual need, vs. an all-encompassing  
77 general purpose set of schema. For example, consider the LDAP “inetorgperson” object class. This  
78 definition is specified by RFC 2798 and includes many attributes that are either required or optional. The  
79 challenge at deployment time, is that administrators are interested in finding out what specific attributes  
80 are actually used by a client application and not what attributes are permissible or optional. Knowing  
81 exactly what is needed allows administrators to focus on how to gather specific attributes rather than a  
82 much wider set of schema that will not actually be used or supported. CARML’s intent is to focus-in on  
83 actual use rather than potential use from a consumer and deployment perspective.

84 In practice however, it is expected that developer tools using CARML would promote usage of standard  
85 schema or data dictionaries already specified by an enterprise. It is important that the developer be  
86 encouraged where possible to use standard data dictionaries rather than defining new schemas. Use of  
87 standard definitions and schema greatly increases the deploy-ability of an application.

## 88 2.2 Subject Indexes

89 **SubjectIndexes** contains a SAML Name Identifier and/or a set of KeyAttribute values that the application  
90 is able to supply to retrieve a Digital Subject. Each key term provided is “ANDed” with the other indexes. If  
91 multiple search indexes are needed to retrieve a subject (e.g. an OR condition), a second  
92 NamedInteraction should be defined. The Attribute Service may then use one or more of these attributes  
93 to uniquely locate a subject. As a minimum at least one IndexNameIdentifier or IndexAttribute is required.  
94 There is a maximum of one IndexNameIdentifier and an unlimited number of IndexAttributes allowed.

95 When a IndexAttribute term is used, a static value may also be declared. If a static value is not supplied,  
96 the value is assumed to be provided at run time.

97 Example: a pair of values – e-mail address and Country will be used. In this case, the Country code of  
98 “US” is statically defined. This example indicates that the client will supply an email address and that the  
99 identity service provider is to assume that an additional static qualifying attribute of Country=“US” is to be  
100 used.

```
101 <SubjectIndexes>  
102   <IndexNameIdentifier>  
103     urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress  
104   </IndexNameIdentifier>  
105   <IndexAttribute Name="Country"  
106     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"  
107     xsi:type="countrycode">  
108     <saml:AttributeValue xsi:type="xs:string">US</saml:AttributeValue>  
109   </IndexAttribute>  
110 </SubjectIndexes>
```

111 A **IndexAttribute** is based on the SAML Attribute [SAML2Core] as defined in section 2.7.3.1.  
112 IndexAttribute supports the additional attributes: Description, and Optional. The “Optional” attribute  
113 means the use of the value if supplied is optional. The element **saml:AttributeValue** is as defined in  
114 section 2.7.3.1.1.

115 A **IndexNameIdentifier** is based on the SAML Name Identifier Format Identifier [SAML2Core] as defined  
116 in section 8.3.

## 117 2.3 Properties

118 Properties are used to specify boolean questions that the application would like answered by the identity  
119 service provider. Properties are used to minimize information transfer to both improve performance and  
120 security. Examples of this might be “AboveEighteen” -- used to ask if the subject older than eighteen  
121 years. Properties may need to include a description that aids the identity service managers in defining the  
122 appropriate values. Example, “AboveEighteen” is answered by taking today’s date and subtracting the  
123 subject’s birthdate to see if it is greater than 18.

124 These take the form of a name only or a name with a value to be provided by the application, or a name  
125 and one or more acceptable values.

126 Example – a sequence of properties. The first asks the question whether the user has property  
127 “AboveEighteen”, the second says that an “EmploymentLevel” will be provided at runtime and the third  
128 that the application wishes to check whether the user has “Location” property and whether it is set to  
129 “HQ”.

```
130 <Properties>  
131   <CarmlProperty Name="AboveEighteen"  
132     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"  
133     Description="Is the subject over eighteen as of current date?"/>  
134   <CarmlProperty Name="EmploymentLevel">  
135     <saml:AttributeValue xsi:type="xs:string"></saml:AttributeValue>  
136   </CarmlProperty>  
137   <CarmlProperty Name="Department">  
138     <saml:AttributeValue xsi:type="xs:string">Information Technology  
139   </saml:AttributeValue>  
140   </CarmlProperty>  
141   <CarmlProperty Name="City">  
142     <saml:AttributeValue xsi:type="xs:string">Redwood Shores</saml:AttributeValue>  
143     <saml:AttributeValue xsi:type="xs:string">Chicago</saml:AttributeValue>  
144     <saml:AttributeValue xsi:type="xs:string">New York</saml:AttributeValue>
```

```
145     </CarmlProperty>
146 </Properties>
```

147 An optional **description** tag is used by the developer to clarify the meaning of a property being  
148 requested.

149 A **CarmlProperty** is based on the SAML Attribute [SAML2Core] as defined in section 2.7.3.1.  
150 CarmlProperty supports the additional attributes: Description, and Optional. “Optional” attribute which  
151 shall mean use of the value if supplied is optional. If optional is true and the CarmlProperty cannot be fully  
152 evaluated, than the result will be as though the property did not exist. The element **saml:AttributeValue**  
153 is as defined in section 2.7.3.1.1.

154

## 155 2.4 Attributes

156 Attributes are the list of attributes (claims or assertions) the client application is requesting to be returned  
157 with the DigitalSubject by the identity service. The list of attributes specified is based on the SAML  
158 attribute schema.

159 Example –

```
160 <Attributes>
161 <CarmlAttribute Name="FirstName"
162     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" />
163 <CarmlAttribute Name="LastName"
164     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" />
165 <CarmlAttribute Name="EmployeeId"
166     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
167     xsi:type="string" />
168 </Attributes>
```

169 For each attribute returned, there may be: no values, one or more values, or an exception indicating a  
170 problem such as an unauthorized request or filtering due to consent or policy condition.

171 To support “modify” requests, it is proposed that the attribute declaration be extended to include a modify  
172 permission flag (Modifiable).

```
173 <Attributes>
174 <CarmlAttribute Name="Language"
175     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" Modifiable="true"/>
176 <CarmlAttribute Name="Country"
177     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" />
178 </Attributes>
```

179 In the above example, Language is modifiable, while Country is treated as read only.

180 A **CarmlAttribute** is based on the SAML Attribute [SAML] as defined in section 2.7.3.1. CarmlAttribute  
181 supports the additional attributes: Description, Optional, and Modifiable. The “Optional” attribute shall  
182 mean the requirement to supply a value is optional. “Modifiable” indicates that the client application is  
183 requesting the right to modify the attribute’s values.

184

## 185 2.5 CARML top-level Containers

186

187 **<CARML:IRdata>** contains one or more **<CARML:NamedInteraction>**’s. Each  
188 **<CARML:NamedInteraction>** has a string name attribute, a mode attribute. It has the following sub-  
189 elements: Indexes, and a sequence of **<carml:Property>** and **<saml:Attribute>**’s as well as usage and  
190 policy request information.

191

192 Example:

193

```
194 <IRData xmlns="urn:oracle:identity:security:0.0:carml"
195     xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
196     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
```

```

197 <NamedInteraction Name="UserProfileUS" xmlns="">
198   <PropagateTo>
199     <Partner>*.oracle.com</Partner>
200   </PropagateTo>
201   <LegalUseRef>http://www.myorg.com/PartnerIdentityUseAgreement.htm</LegalUseRef>
202   <DataQualityRef>
203     http://www.myorg.com/PartnerIdentityQualityAgreement.htm
204   </DataQualityRef>
205   <CacheTTL>12:00:00.00</CacheTTL>
206   <SubjectIndexes>
207     <IndexNameIdentifier>
208       urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
209     </IndexNameIdentifier>
210     <IndexAttribute Name="Country"
211       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
212       xsi:type="countrycode">
213       <saml:AttributeValue xsi:type="xs:string">US</saml:AttributeValue>
214     </IndexAttribute>
215   </SubjectIndexes>
216   <Attributes>
217     <CarmlAttribute Name="Language"
218       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
219       Modifiable="true"/>
220     <CarmlAttribute Name="FirstName"
221       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" />
222     <CarmlAttribute Name="LastName"
223       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" />
224     <CarmlAttribute Name="EmployeeId"
225       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"
226       xsi:type="string" />
227   </Attributes>
228   <Properties>
229     <CarmlProperty Name="AboveEighteen"
230       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"
231       Description="Is the subject over the age of eighteen as of current date?"/>
232     <CarmlProperty Name="EmploymentLevel">
233       <saml:AttributeValue xsi:type="xs:string"></saml:AttributeValue>
234     </CarmlProperty>
235     <CarmlProperty Name="Department">
236       <saml:AttributeValue xsi:type="xs:string">
237         Information Technoloy
238       </saml:AttributeValue>
239     </CarmlProperty>
240     <CarmlProperty Name="City">
241       <saml:AttributeValue xsi:type="xs:string">Redwood Shores</saml:AttributeValue>
242       <saml:AttributeValue xsi:type="xs:string">Chicago</saml:AttributeValue>
243       <saml:AttributeValue xsi:type="xs:string">New York</saml:AttributeValue>
244     </CarmlProperty>
245   </Properties>
246 </NamedInteraction>
247 </IRData>

```

248  
249

## 250 2.5.1 Meta-data Elements

251 Certain elements that represent different aspects of attribute meta-data is an additional element that can  
 252 be used to indicate the usage policy that the client wishes to declare. It consists of the following optional  
 253 elements:

### 254 LegalUseRef

255 A URL representing legal documentation for how the information will be used. This is for  
 256 documentation purposes only.

### 257 QualityStatement

258 A URL representing a statement about the quality of verifiability of information requested. This is  
 259 for documentation purposes only.

260 **MayPropagate**

261 A “true” or “false” term. If true, it may also contain a sub-element indicating the parties with which  
262 propagation is intended. If missing, false shall be assumed.

263 **CacheTTL**

264 A time value expressed as a string in the form “DDD HH:MM:SS” expressing how long the value  
265 may be cached or stored by the client. If missing, caching is assumed to be not allowed (or zero  
266 days, hours, minutes, or seconds).

267 **Example:**

```
268 <IRData>  
269 . . .  
270 <NamedInteraction Name="UserProfileUS" xmlns="">  
271   <PropagateTo>  
272     <Partner>*.oracle.com</Partner>  
273   </PropagateTo>  
274   <LegalUseRef>http://www.myorg.com/PartnerIdentityUseAgreement.htm</LegalUseRef>  
275   <DataQualityRef>  
276     http://www.myorg.com/PartnerIdentityQualityAgreement.htm  
277   </DataQualityRef>  
278   <CacheTTL>12:00:00.00</CacheTTL>  
279   . . .  
280 </IRData>
```

281

## 3 Schema

```

283 <?xml version="1.0" encoding="UTF-8" ?>
284 <schema targetNamespace="urn:igf:client:0.3:carml"
285         xmlns="http://www.w3.org/2001/XMLSchema"
286         xmlns:carml="urn:igf:client:0.3:carml"
287         xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
288         elementFormDefault="unqualified" attributeFormDefault="unqualified"
289         blockDefault="substitution">
290   <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
291           schemaLocation="saml-schema-assertion-2.0.xsd"/>
292   <annotation>
293     <documentation>Document identifier: igf-carml-03
294                   Location: TBD Revision
295                   history: V0.1 (November, 2006): Initial Draft Schema.
296                   V0.2 (November, 2006) Changed term key to index
297                   V0.3 (November, 2006) Changed namespace to urn:igf:client:0.3:carml
298                   (C)2006, Oracle Corporation</documentation>
299   </annotation>
300   <complexType name="CarmlAttrNameType">
301     <attribute name="Name" type="string" use="required"/>
302     <attribute name="NameFormat" type="anyURI" use="optional"/>
303     <attribute name="FriendlyName" type="string" use="optional"/>
304     <attribute name="Description" type="string" use="optional"/>
305     <anyAttribute namespace="##other" processContents="lax"/>
306   </complexType>
307   <complexType name="CarmlPropertyType">
308     <complexContent>
309       <extension base="saml:AttributeType">
310         <attribute name="Description"/>
311         <attribute name="Optional" type="boolean" default="false"/>
312       </extension>
313     </complexContent>
314   </complexType>
315   <complexType name="CarmlAttrType">
316     <complexContent>
317       <extension base="carml:CarmlAttrNameType">
318         <attribute name="Modifiable" default="false" type="boolean"/>
319         <attribute name="Optional" type="boolean" default="false"/>
320       </extension>
321     </complexContent>
322   </complexType>
323   <element name="IRData">
324     <complexType>
325       <sequence>
326         <element name="NamedInteraction" maxOccurs="unbounded">
327           <complexType>
328             <sequence>
329               <element name="PropagateTo" minOccurs="0">
330                 <complexType>
331                   <sequence>
332                     <element name="Partner" type="anyURI" minOccurs="1"
333                           maxOccurs="unbounded"/>
334                   </sequence>
335                 </complexType>
336               </element>
337               <element name="LegalUseRef" minOccurs="0" maxOccurs="1"
338                       type="anyURI"/>
339               <element name="DataQualityRef" minOccurs="0" maxOccurs="1"
340                       type="anyURI"/>
341               <element name="CacheTTL" minOccurs="0" maxOccurs="1" type="time"/>
342               <element name="SubjectIndexes" minOccurs="0" maxOccurs="1">
343                 <complexType>
344                   <sequence maxOccurs="1" minOccurs="1">
345                     <element name="IndexNameIdentifier" type="saml:NameIDType"
346                             minOccurs="0"/>
347                     <element name="IndexAttribute" type="carml:CarmlPropertyType"
348                             maxOccurs="unbounded" minOccurs="0"/>
349                   </sequence>

```

```
350     </complexType>
351 </element>
352 <element name="Attributes" minOccurs="0">
353   <complexType>
354     <sequence maxOccurs="1" minOccurs="1">
355       <element name="CarmlAttribute" type="carml:CarmlAttrType"
356         maxOccurs="unbounded"/>
357     </sequence>
358   </complexType>
359 </element>
360 <element name="Properties" minOccurs="0">
361   <complexType>
362     <sequence maxOccurs="1" minOccurs="1">
363       <element name="CarmlProperty" type="carml:CarmlPropertyType"
364         maxOccurs="unbounded"/>
365     </sequence>
366   </complexType>
367 </element>
368 </sequence>
369 <attribute name="Name" use="required"/>
370 </complexType>
371 </element>
372 </sequence>
373 </complexType>
374 </element>
375 </schema>
```

376

377

---

## Appendix A. Revision History

378

Revision	Date	Editor	Changes Made
00	19 Sep 2006	Phil Hunt	Initial contribution (replaces EPFIS-CARML-02)
01	15 Nov 2006	Phil Hunt	Addition of "optional" attribute
02	17 Nov 2006	Phil Hunt	Replace Key with Index to avoid confusion with crypto terminology

379