



Hierarchical resource profile of XACML v2.0

OASIS Standard, 1 February 2005

Document identifier:

access_control-xacml-2.0-hier-profile-spec-os

Location:

http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-hier-profile-spec-os.pdf

Editor:

Anne Anderson, Sun Microsystems (anne.anderson@sun.com)

Abstract:

This document provides a profile for the use XACML with resources that are structured as hierarchies. The profile addresses resources represented as nodes in XML documents or represented in some non-XML way. The profile covers identifying nodes in a hierarchy, requesting access to nodes in a hierarchy, and specifying policies that apply to nodes in a hierarchy.

Status:

This version of the specification is an approved OASIS Standard.

Access Control TC members should send comments on this specification to the xacml@lists.oasis-open.org list. Others should use the comment form at http://oasis-open.org/committees/comments/form.php?wg_abbrev=xacml.

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 Access Control TC web page (<http://www.oasis-open.org/committees/xacml/ipr.php>).

For any errata page for this specification, please refer to the Access Control TC web page (<http://www.oasis-open.org/committees/xacml>).

Copyright © OASIS Open 2004-2005 All Rights Reserved.

29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

Table of Contents

1 Introduction.....	3
1.1 Terminology.....	4
1.2 Notation.....	5
2 Representing the identity of a node.....	6
2.1 Nodes in XML documents.....	6
2.2 Nodes in resources that are not XML documents.....	6
3 Requesting access to a node.....	8
3.1 Nodes in an XML document.....	8
3.2 Nodes in a resource that is not an XML document.....	9
4 Stating policies that apply to nodes.....	11
4.1 Policies applying to nodes in any hierarchical resource.....	11
4.2 Policies applying only to nodes in XML documents.....	11
4.3 Policies applying only to nodes in non-XML resources.....	11
5 New DataType.....	13
5.1 xpath-expression.....	13
6 New attribute identifiers.....	14
6.1 document-id.....	14
6.2 resource-parent.....	14
6.3 resource-ancestor.....	14
6.4 resource-ancestor-or-self.....	14
7 New profile identifiers.....	15
8 References.....	16
A. Acknowledgments.....	17
B. Notices.....	18

1 Introduction

55

56 It is often the case that a **resource** is organized as a hierarchy. Examples include file systems, XML
57 documents, and organizations. This Profile specifies how XACML can provide **access control** for a
58 **resource** that is organized as a hierarchy.

59 Why are **resources** organized as hierarchies special? First of all, policies over hierarchies frequently
60 apply the same **access controls** to entire sub-trees of the hierarchy. Being able to express a single
61 policy constraint that will apply to an entire sub-tree of **nodes** in the hierarchy, rather than having to
62 specify a separate constraint for each **node**, increases both ease of use and the likelihood that the policy
63 will correctly reflect the desired **access controls**. Another special characteristic of **hierarchical**
64 **resources** is that access to one **node** may depend on the value of another **node**. For example, a
65 medical patient might be granted access to the “diagnosis” **node** in a XML document medical record only
66 if the patient’s name matches the value in the “patient name” **node**. Where this is the case, the
67 requested **node** can not be processed in isolation from the rest of the **nodes** in the hierarchy, and the
68 PDP must have access to the values of other **nodes**. Finally, the identity of **nodes** in a hierarchy often
69 depends on the position of the **node** in the hierarchy; there also may be multiple ways to describe the
70 identity of a single **node**. In order for policies to apply to **nodes** as intended, attention must be paid to
71 consistent representations for the identity of the **nodes**. Otherwise, a requester may bypass **access**
72 **controls** by requesting a **node** using an identity that differs from the one used by the policy.

73 In this Profile, a **resource** organized as a hierarchy may be a “tree” (a hierarchy with a single root) or a
74 “forest” (a hierarchy with multiple roots), but the hierarchy may not have cycles. Another term for these
75 two types of hierarchy is “Directed Acyclic Graph” or “DAG”. All such **resources** are called **hierarchical**
76 **resources** in this Profile. An XML document is always structured as a “tree”. Other types of
77 **hierarchical resources**, such as files in a file system that supports links, may be structured as “forests”.

78 In this Profile, the **nodes** in a **hierarchical resource** are treated as individual **resources**. An
79 **authorization decision** that permits **access** to an interior **node** does not imply that **access** to its
80 descendant **nodes** is permitted. An **authorization decision** that denies **access** to an interior **node**
81 does not imply that **access** to its descendant **nodes** is denied.

82 There are three types of facilities specified in this Profile for dealing with **hierarchical resources**:

- 83 • Representing the identity of a **node**.
- 84 • Requesting access to a **node**.
- 85 • Stating policies that apply to one or more **nodes**.

86 Support for each of these facilities is optional.

87 This Profile addresses two ways of representing a hierarchical resource. In the first way, the hierarchy of
88 which the node is a part is represented as an XML document that is included in the the Request, and the
89 requested resource is represented as a node in that document. In the second way, the requested
90 resource is not represented as a node in an XML document, and there is no representation of the
91 hierarchy of which it is a part included in the Request. Note that the actual target resource in the first
92 case need not be part of an XML document - it is merely represented that way in the Request. Likewise,
93 the target resource in the second case might actually be part of an XML document, but is being
94 represented in some other way in the Request. Thus there is no assumed correlation between the
95 structure of the resource as represented in the Request and the actual structure of the physical resource
96 being accessed.

97 Facilities for dealing with **resources** represented as **nodes** in XML documents can make use of the fact
98 that the XML document itself is included in the **decision request**. [XPath] expressions can be used to
99 reference **nodes** in this document in a standard way, and can provide unique representations for a given
100 **node** in the document. These facilities are not available for **hierarchical resources** that are not
101 represented as XML documents. Other means must be provided in the case of such non-XML

102 **resources** for determining the location of the requested **node** in the hierarchy. In some cases this can
103 be done by including the **node's** position in the hierarchy as part of the **node's** identity. In other cases, a
104 **node** may have more than one normative identity, such as when the pathname of a file in a file system
105 can include hard links. In such cases, the XACML **PDP's** Context Handler may need to supply the
106 identities of all the **node's** ancestors. For all these reasons, the facilities for dealing with **nodes** in XML
107 documents differ from the facilities for dealing with **nodes** in other **hierarchical resources**.

108 In dealing with a **hierarchical resource**, it may be useful to request **authorization decisions** for
109 multiple **nodes** in the **resource** in a single **decision request**. Ways to make such requests are
110 specified in another Profile – the *Multiple resource profile of XACML v2.0* [MULTIPLE]. That Profile also
111 provides a way to return a single **authorization decision** when access to multiple **nodes** in a hierarchy
112 is requested. Readers of this Profile are encouraged to become familiar with the *Multiple resource*
113 *profile of XACML*. This Profile may be considered to be layered on top of the multiple resource profile,
114 which in turn is layered on top of the behavior specified in the core XACML specification [XACML]. The
115 functionality in this Profile MAY, however, be layered directly on the functionality in the core XACML
116 specification.

117 This Profile for **hierarchical resources** assumes that all requests for **access** to multiple **nodes** in a
118 **hierarchical resource** [MULTIPLE] have been resolved to individual requests for **access** to a single
119 **node**.

120 1.1 Terminology

121 **Access** - Performing an **action**.

122 **Access control** - Controlling **access** in accordance with a **policy**.

123 **Action** – An operation on a **resource**.

124 **Applicable policy** - The set of **policies** and **policy sets** that governs **access** for a specific **decision**
125 **request**.

126 **Attribute** - Characteristic of a **subject**, **resource**, **action** or **environment** that may be referenced in a
127 **predicate** or **target** (see also – **named attribute**) or provided in a **context**. May also refer to an XML
128 syntactic attribute, in which case the term will be qualified as “XML attribute.”

129 **Authorization decision** - The result of evaluating **applicable policy**, returned by the **PDP** to the **PEP**.
130 A function that evaluates to "Permit", "Deny", "Indeterminate" or "NotApplicable", and
131 (optionally) a set of **obligations**.

132 **Bag** – An unordered collection of values, in which there may be duplicate values.

133 **Context** - The canonical representation of a **decision request** and an **authorization decision**.

134 **Decision** – The result of evaluating a **rule**, **policy** or **policy set**.

135 **Decision request** - The request by a **PEP** to a **PDP** to render an **authorization decision**.

136 **Hierarchical resource** – A **resource** that is organized as a tree or forest (Directed Acyclic Graph) of
137 individual **resources** called **nodes**.

138 **Node** – An individual **resource** that is part of a **hierarchical resource**.

139 **Obligation** - An operation specified in a **policy** or **policy set** that should be performed by the **PEP** in
140 conjunction with the enforcement of an **authorization decision**.

141 **Policy** - A set of **rules**, an identifier for the **rule-combining algorithm** and (optionally) a set of
142 **obligations**. May be a component of a **policy set**.

143 **Policy administration point (PAP)** - The system entity that creates a **policy** or **policy set**.

144 **Policy decision point (PDP)** - The system entity that evaluates **applicable policy** and renders an
145 **authorization decision**. This term is defined in a joint effort by the IETF Policy Framework Working

146 Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in
147 [RFC3198]. This term corresponds to "Access Decision Function" (ADF) in [ISO10181-3].

148 **Policy enforcement point (PEP)** - The system entity that performs **access control**, by making
149 **decision requests** and enforcing **authorization decisions**. This term is defined in a joint effort by the
150 IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common
151 Information Model (CIM) in [RFC3198]. This term corresponds to "Access Enforcement Function" (AEF)
152 in [ISO10181-3].

153 **Policy set** – A set of **policies**, other **policy sets**, a policy-combining algorithm and {optionally} a set of
154 **obligations**. May be a component of another **policy set**.

155 **Resource** - Data, service or system component. The object for which **access** is requested in a
156 **decision request**.

157 1.2 Notation

158 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
159 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as
160 described in IETF RFC 2119 [RFC2119]:

161 "they MUST only be used where it is actually required for interoperation or to limit behavior which
162 has potential for causing harm (e.g., limiting retransmissions)"

163 These keywords are thus capitalized when used to unambiguously specify requirements over protocol
164 and application features and behavior that affect the interoperability and security of implementations.
165 When these words are not capitalized, they are meant in their natural-language sense.

166 The phrase **{Normative, but optional}** means that the described functionality is optional for compliant
167 XACML implementations, but, if the functionality is claimed as being supported according to this Profile,
168 then it SHALL be supported in the way described.

169 Example code listings appear like this.

170 In descriptions of syntax, elements in angle brackets (" $<$ ", " $>$ ") are to be replaced by appropriate values,
171 square brackets (" $[$ ", " $]$ ") enclose optional elements, elements in quotes are literal components, and " $*$ "
172 indicates that the preceding element may occur zero or more times.

2 Representing the identity of a node

173

174 *{Normative}*

174

175 In order for XACML *policies* to apply consistently to *nodes* in a *hierarchical resource*, it is necessary
176 for the *nodes* in that *resource* to be represented in a consistent way. If a *policy* refers to a *node* using
177 one representation, but a *request* refers to the *node* using a different representation, then the *policy* will
178 not apply, and security may be compromised.

179 The following sections describe RECOMMENDED representations for *nodes* in *hierarchical*
180 *resources*. Alternative representations of *nodes* in a given *resource* are permitted so long as all
181 *Policy Administration Points* and all *Policy Enforcement Points* that deal with that *resource* have
182 contracted to use the alternative representation.

2.1 Nodes in XML documents

183

184 *{Normative, but optional}*

184

185 The following URI SHALL be used as the identifier for the functionality specified in this Section of this
186 Profile: urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id.

187 The identity of a *node* in a *resource* that is represented as an XML document instance SHALL be an
188 XPath expression that evaluates to exactly that one *node* in the copy of the *resource* that is contained in
189 the <ResourceContent> element of the <Resource> element of the <Request>.

2.2 Nodes in resources that are not XML documents

190

191 *{Normative, but optional}*

191

192 The following URI SHALL be used as the identifier for the functionality specified in this Section of this
193 Profile: urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id.

194 The identity of a *node* in a *hierarchical resource* that is not represented as an XML document instance
195 SHALL be represented as a URI that conforms to [RFC2396]. Such URIs are of the following form.

196 <scheme> ":" <authority> "/" <pathname>

197 File system *resources* SHALL use the "file:" scheme. If no standard <scheme> for the *resource*
198 type is specified in [RFC2396] or in a related standard for a registered URI scheme, then the URI SHALL
199 use the "file:" scheme.

200 The <pathname> portion of the URI SHALL be of the form

201 <root name> ["/" <node name>]*

202 The sequence of <root name> and <node name> values SHALL correspond to the individual
203 hierarchical component names of ancestors of the represented *node* along the path from a <root>
204 *node* to the represented *node*.

205 The following canonicalization SHALL be used.

- 206 • The encoding of the URI SHALL be UTF8.
- 207 • Case-insensitive portions of the URI SHALL be lower case.
- 208 • Escaping of characters SHALL conform to [RFC2396].
- 209 • The <authority> portion of the URI SHALL be specified and SHALL be the standard authority
210 representation for the given *resource* type. Where the <authority> could be specified using either
211 a Domain Name Service (DNS) [RFC1034] name or a numeric IPv4 or IPv6 address, the DNS name
212 SHALL be used.

- 213 • The components of the <pathname> portion of the URI SHALL be specified using the canonical form
214 for such path components at the <authority>.
- 215 • In accordance with [RFC2396], the separator character between hierarchical components of the
216 <pathname> portion of the URI SHALL be the character “/”. Sequences of the “/” character SHALL
217 be resolved to a single “/”. **Node** identities SHALL NOT terminate with the “/” character.
- 218 • The <pathname> SHALL contain no soft links.
- 219 • All <pathname> values SHALL be absolute.
- 220 • If there is more than one fully resolved, absolute path from a <root> at the <authority> to the
221 represented **node**, then a separate **resource attribute** with AttributeId
222 “urn:oasis:names:tc:xacml:1.0:resource:resource-id” and DataType
223 http://urn:oasis:names:tc:xacml:1.0:data-type:anyURI SHALL be present in the
224 Request Context for each such path.

3 Requesting access to a node

225

226 *{Normative}*

227 In order for XACML *policies* to apply consistently to *nodes* in a *hierarchical resource*, it is necessary
228 for each request *context* that represents a request for *access* to a *node* in that *resource* to use a
229 consistent description of that *node access*. If a *policy* refers to certain expected *attributes* of a *node*,
230 but the request *context* does not contain those *attributes*, or if the *attributes* are not expressed in the
231 expected way, then the *policy* may not apply, and security may be compromised.

232 The following sections describe RECOMMENDED request *context* descriptions of *access* to *nodes* in
233 *hierarchical resources*. Alternative representations of such requests are permitted so long as all
234 *Policy Administration Points* and all *Policy Enforcement Points* that deal with that *resource* have
235 contracted to use the alternative representation.

3.1 Nodes in an XML document

236

237 *{Normative, but optional}*

238 The following URI SHALL be used as the identifier for the functionality specified in this Section of this
239 Profile: urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req. The
240 *attributes* with AttributeIds of "urn:oasis::names:tc:xacml:2.0:resource:resource-
241 parent", "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor" and
242 "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self" are optional to
243 implement. If supported for use in resources represented as XML documents, the following URIs SHALL
244 be used as identifiers for the functionality they represent:
245 "urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-
246 parent", "urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-
247 req:resource-ancestor", and
248 "urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-
249 ancestor-or-self".

250 In order to request *access* to a *resource* represented as a *node* in an XML document, the request
251 *context* <Resource> element SHALL contain the following elements and XML attributes.

- 252 • A <ResourceContent> element that contains the entire XML document instance of which the
253 requested *node* is a part.
- 254 • An <Attribute> element with an AttributeId of
255 "urn:oasis::names:tc:xacml:1.0:resource:resource-id" and a DataType of
256 "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". The
257 <AttributeValue> of this <Attribute> SHALL be an XPath expression whose context node
258 SHALL be the one and only child of the <ResourceContent> element. This XPath expression
259 SHALL evaluate to a nodeset containing the single *node* in the <ResourceContent> element that
260 is the *node* to which *access* is requested. This <Attribute> MAY specify an Issuer.
- 261 • An <Attribute> element with an AttributeId of
262 "urn:oasis::names:tc:xacml:2.0:resource:resource-parent" and a DataType of
263 "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". The
264 <AttributeValue> of this <Attribute> SHALL be an XPath expression; the context node for
265 this XPath expression SHALL be the one and only child of the <ResourceContent> element. This
266 XPath expression SHALL evaluate to a nodeset containing the single *node* in the
267 <ResourceContent> element that is the immediate parent of the *node* represented in the
268 "resource-id" *attribute*. This <Attribute> MAY specify an Issuer.
- 269 • For each *node* in the XML document instance that is an ancestor of the *node* represented by the
270 "resource-id" *attribute*, an <Attribute> element with an AttributeId of
271 "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor" and a DataType of

272 `“urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression”`. The
273 `<AttributeValue>` of this `<Attribute>` SHALL be an XPath expression; the context node for
274 this XPath expression SHALL be the one and only child of the `<ResourceContent>` element. This
275 XPath expression SHALL evaluate to a nodeset containing the single **node** in the
276 `<ResourceContent>` element that is the respective ancestor of the **node** represented in the
277 `“resource-id” attribute`. For each `“resource-parent” attribute`, there SHALL be a
278 corresponding `“resource-ancestor” attribute`. This `<Attribute>` MAY specify an Issuer.

279 • For each **node** in the XML document instance that is an ancestor of the **node** represented by the
280 `“resource-id” attribute`, and for the `“resource-id” node` itself, an `<Attribute>` element with
281 an `AttributeId` of `“urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-`
282 `or-self”` and a `DataType` of `“urn:oasis:names:tc:xacml:2.0:data-type:xpath-`
283 `expression”`. The `<AttributeValue>` of this `<Attribute>` SHALL be an XPath expression; the
284 context node for this XPath expression SHALL be the one and only child of the
285 `<ResourceContent>` element. This XPath expression SHALL evaluate to a nodeset containing the
286 single **node** in the `<ResourceContent>` element that is the respective ancestor of the **node**
287 represented in the `“resource-id” attribute`, or that is the `“resource-id” node` itself. For each
288 `“resource-parent”` and `“resource-id” attribute`, there SHALL be a corresponding `“resource-`
289 `ancestor-or-self” attribute`. This `<Attribute>` MAY specify an Issuer.

290 Additional **attributes** MAY be included in the `<Resource>` element. In particular, the following
291 **attribute** MAY be included.

292 • An `<Attribute>` element with an `AttributeId` of
293 `“urn:oasis::names:tc:xacml:2.0:resource:document-id”` and a `DataType` of
294 `“urn:oasis:names:tc:xacml:1.0:data-type:anyURI”`. The `<AttributeValue>` of this
295 `<Attribute>` SHALL be a URI that identifies the XML document of which the requested **resource** is
296 a part, and of which a copy is present in the `<ResourceContent>` element. This `<Attribute>`
297 MAY specify an Issuer.

298 3.2 Nodes in a resource that is not an XML document

299 *{Normative, but optional}*

300 The following URI SHALL be used as the identifier for the functionality specified in this Section of this
301 Profile: `urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req`. The
302 **attributes** with `AttributeIds` of `“urn:oasis::names:tc:xacml:2.0:resource:resource-`
303 `parent”`, `“urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor”`, and
304 `“urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self”` are optional to
305 implement. If supported for use in resources that are not represented as XML documents, the following
306 URIs SHALL be used as identifiers for the functionality they represent:
307 `“urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-`
308 `parent”`, `“urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-`
309 `req:resource-ancestor”`, and
310 `“urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-`
311 `ancestor-or-self”`.

312 In order to request **access** to a **node** in a **hierarchical resource** that is not represented as an XML
313 document, the request **context** `<Resource>` element SHALL NOT contain a `<ResourceContent>`
314 element. The request **context** `<Resource>` element SHALL contain the following elements and XML
315 attributes. Note that a **node** in a **hierarchical resource** that is not represented as an XML document
316 MAY have multiple parents. For example, in a file system that supports hard links, there may be multiple
317 normative paths to a single file. Each such path MAY contain different sets of parents and ancestors.

318 • For each normative representation of the requested **node**, an `<Attribute>` element with
319 `AttributeId` of `“urn:oasis::names:tc:xacml:1.0:resource:resource-id”`. The
320 `<AttributeValue>` of this `<Attribute>` SHALL be a unique, normative identity of the **node** to
321 which **access** is requested. The `DataType` of this `<Attribute>` SHALL depend on the

322 representation chosen for the identity of **nodes** in this particular **resource**. This <Attribute> MAY
323 specify an Issuer.

324 • For each immediate parent of the **node** specified in the “resource-id” **attribute** or **attributes**, and
325 for each normative representation of that parent **node**, an <Attribute> element with
326 AttributeId “urn:oasis::names:tc:xacml:2.0:resource:resource-parent”. The
327 <AttributeValue> of this <Attribute> SHALL be the normative identity of the parent **node**.
328 The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of
329 **nodes** in this particular **resource**. This <Attribute> MAY specify an Issuer. If the requested
330 **node** is part of a forest rather than part of a single tree, or if the parent **node** has more than one
331 normative representation, there SHALL be at least one instance of this **attribute** for each parent
332 along each path to the multiple roots of which the requested **node** is a descendant, and for each
333 normative representation of each such parent.

334 • For each ancestor of the **node** specified in the “resource-id” **attribute** or **attributes**, and for each
335 normative representation of that ancestor **node**, an <Attribute> element with AttributeId
336 “urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor”. The
337 <AttributeValue> of this <Attribute> SHALL be the normative identity of the ancestor **node**.
338 The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of
339 **nodes** in this particular **resource**. This <Attribute> MAY specify an Issuer. For each
340 “resource-parent” **attribute**, there SHALL be a corresponding “resource-ancestor” **attribute**.
341 If the requested **node** is part of a forest rather than part of a single tree, or if the ancestor **node** has
342 more than one normative representation, there SHALL be at least one instance of this **attribute** for
343 each ancestor along each path to the multiple roots of which the requested **node** is a descendant,
344 and for each normative representation of each such ancestor. The order of the values for this
345 **attribute** do not necessarily reflect the position of each ancestor **node** in the hierarchy.

346 • For each ancestor of the **node** specified in the “resource-id” **attribute** or **attributes**, and for each
347 normative representation of that ancestor **node**, and for each normative representation of the
348 “resource-id” **node** itself, an <Attribute> element with AttributeId
349 “urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self”. The
350 <AttributeValue> of this <Attribute> SHALL be the respective normative identity of the
351 ancestor **node** or of the “resource-id” **node** itself. The DataType of this <Attribute> SHALL
352 depend on the representation chosen for the identity of **nodes** in this particular **resource**. This
353 <Attribute> MAY specify an Issuer. For each “resource-ancestor” and “resource-id”
354 **attribute**, there SHALL be a corresponding “resource-ancestor-or-self” **attribute**. If the
355 requested **node** is part of a forest rather than part of a single tree, or if the ancestor **node** has more
356 than one normative representation, there SHALL be at least one instance of this **attribute** for each
357 ancestor along each path to the multiple roots of which the requested **node** is a descendant, and for
358 each normative representation of each such ancestor. The order of the values for this **attribute** do not
359 necessarily reflect the position of each ancestor **node** in the hierarchy.

360 Additional **attributes** MAY be included in the <Resource> element.

4 Stating policies that apply to nodes

{Non-normative}

This Section describes various ways to specify a *policy* predicate that can apply to multiple *nodes* in a *hierarchical resource*. This is not intended to be an exhaustive list.

4.1 Policies applying to nodes in any hierarchical resource

{Non-normative}

Resource attributes with the following `AttributeId` values, described in Section 6: *New attribute identifiers for hierarchical resources* of this Profile, MAY be used to state *policies* that apply to one or more *nodes* in any *hierarchical resource*.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-parent
```

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
```

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

Note that a `<ResourceAttributeDesignator>` that refers to the “resource-parent”, “resource-ancestor”, or “resource-ancestor-or-self” *attribute* will return a bag of values representing all normative identities of all parents, ancestors, or ancestors plus the *resource* itself, respectively, of the *resource* to which *access* is being requested. The representations of the identities of these parents, ancestors, or self will not necessarily indicate the path from the root of the hierarchy to the respective parent, ancestor, or self unless the representation recommended in Section 3.2: *Nodes in a resource that is not an XML document* is used.

The standard XACML [XACML] bag and higher-order bag functions MAY be used to state *policies* that apply to one or more *nodes* in any *hierarchical resource*. The *nodes* used as arguments to these functions MAY be specified using a `<ResourceAttributeDesignator>` with the “resource-parent”, “resource-ancestor”, or “resource-ancestor-or-self” `AttributeId` value.

4.2 Policies applying only to nodes in XML documents

{Non-normative}

For *hierarchical resources* that are represented as XML document instances, the following function, described in the XACML 2.0 Specification [XACML] MAY be used to state *policy* predicates that apply to one or more *nodes* in that *resource*.

```
urn:oasis:names:tc:xacml:2.0:function:xpath-node-match
```

The standard XACML `<AttributeSelector>` element MAY be used in *policies* to refer to all or portions of a *resource* represented as an XML document and contained in the `<ResourceContent>` element of a request *context*.

The standard XACML [XACML] bag and higher-order bag functions MAY be used to state *policies* that apply to one or more *nodes* in a resource represented as an XML document. The *nodes* used as arguments to these functions MAY be specified using an `<AttributeSelector>` that selects a portion of the `<ResourceContent>` element of the `<Resource>` element.

4.3 Policies applying only to nodes in non-XML resources

{Non-normative}

For *hierarchical resources* that are not represented as XML document instances, and where the URI representation of *nodes* specified in Section 2 of this Profile is used, the following functions described in the XACML 2.0 Specification [XACML] MAY be used to state *policies* that apply to one or more *nodes*

402 in that **resource**.

403 urn:oasis:names:tc:xacml:1.0:function:anyURI-equal

404 urn:oasis:names:tc:xacml:1.0:function:regex-uri-match

405 5 New DataType

406 *{Normative, but optional}*

407 The following value for the XML `DataType` attribute value MAY be supported for use with *hierarchical*
408 *resources* represented as XML documents. Support for this `DataType` is required in order to support
409 Section 3.1 in this Profile.

410 5.1 xpath-expression

411 The `DataType` represented by the following URI represents an XPath expression. *Attribute* values
412 having this `DataType` SHALL be strings that are to be interpreted as XPath expressions. The result of
413 evaluating such an *attribute* SHALL be the nodeset that results from evaluating the XPath expression. If
414 the string is not a valid XPath expression, the result of evaluating the *attribute* SHALL be
415 Indeterminate.

416 `Urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression.`

417 6 New attribute identifiers

418 *{Normative, but optional}*

419 6.1 document-id

420 The following identifier indicates the identity of the XML document that represents the hierarchy of which
421 the requested **resource** is a part, and of which a copy is present in the <ResourceContent> element.
422 Whenever **access** to a **node** in a **resource** represented as an XML document is requested, one or more
423 instances of an **attribute** with this `AttributeId` MAY be provided in the <Resource> element of the
424 request **context**. The `DataType` of these **attributes** SHALL be
425 “urn:oasis:names:tc:xacml:1.0:data-type:anyURI”.

426 urn:oasis:names:tc:xacml:2.0:resource:document-id

427 6.2 resource-parent

428 The following identifier indicates one normative identity of one parent **node** in the tree or forest of which
429 the requested **node** is a part. Whenever **access** to a **node** in a **hierarchical resource** is requested,
430 one instance of an **attribute** with this `AttributeId` SHALL be provided in the <Resource> element of
431 the request **context** for each normative representation of each **node** that is a parent of the requested
432 **node**.

433 urn:oasis:names:tc:xacml:2.0:resource:resource-parent

434 6.3 resource-ancestor

435 The following identifier indicates one normative identity of one ancestor **node** in the tree or forest of
436 which the requested **node** is a part. Whenever **access** to a **node** in a **hierarchical resource** is
437 requested, one instance of an **attribute** with this `AttributeId` SHALL be provided in the <Resource>
438 element of the request **context** for each normative representation of each **node** that is an ancestor of
439 the requested **node**.

440 urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor

441 6.4 resource-ancestor-or-self

442 The following identifier indicates one normative identity of one ancestor **node** in the tree or forest of
443 which the requested **node** is a part, or one normative identity of the requested **node** itself. Whenever
444 **access** to a **node** in a **hierarchical resource** is requested, one instance of an **attribute** with this
445 `AttributeId` SHALL be provided in the <Resource> element of the request **context** for each
446 normative representation of each **node** that is an ancestor of the requested **node**, and for each
447 normative representation of the requested **node** itself.

448 urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self

7 New profile identifiers

449

450 **{normative}**

451 The following URI values SHALL be used as identifiers for the functionality specified in various Sections
452 of this Profile:

453 Section 2.1: *Nodes in XML documents*

454 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id

455 Section 2.2: *Nodes in resources that are not XML documents*

456 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id

457 Section 3.1: *Nodes in an XML document*

458 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req

459 Support for the “resource-parent”, “resource-ancestor”, and “resource-ancestor-
460 or-self” **attributes** is optional within this Section, so these have separate identifiers:

461 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-
462 req:resource-parent

463 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-
464 req:resource-ancestor

465 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-
466 req:resource-ancestor-or-self

467 Section 3.2: *Nodes in a resource that is not an XML document*

468 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req

469 Support for the “resource-parent”, “resource-ancestor”, and “resource-ancestor-
470 or-self” **attributes** is optional within this Section, so these have separate identifiers:

471 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-
472 req:resource-parent

473 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-
474 req:resource-ancestor

475 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-
476 req:resource-ancestor-or-self

8 References

477

- 478 **[ISO10181-3]** ISO/IEC JTC 1, *Information technology -- Open Systems Interconnection --*
479 *Security frameworks for open systems: Access control framework*, ISO/IEC
480 10181-3:1996, 1996.
- 481 **[RFC1034]** P. Mockapetris, *DOMAIN NAMES – CONCEPTS AND FACILITIES*, IETF RFC
482 1034, November 1987, <ftp://ftp.isi.edu/in-notes/rfc1034.txt>
- 483 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, IETF
484 RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.
- 485 **[RFC2396]** T. Berners-Lee, et al., *Uniform Resource Identifiers (URI): Generic Syntax*,
486 <http://www.ietf.org/rfc/rfc2396.txt>, IETF RFC 2396, August 1998.
- 487 **[RFC3198]** A. Westerinen, et al., *Terminology for Policy-Based Management*,
488 <http://www.ietf.org/rfc/rfc3198.txt>, IETF RFC 3198, November 2001.
- 489 **[MULTIPLE]** A. Anderson, ed., *Multiple resource profile of XACML v2.0, OASIS Standard*, 1
490 February 2005, [http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-mult-profile-spec-os.pdf)
491 [mult-profile-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-mult-profile-spec-os.pdf)
- 492 **[XACML]** T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML)*
493 *Version 2.0, OASIS Standard*, 1 February 2005, [http://docs.oasis-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf)
494 [open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf).
- 495 **[XPath]** *XML Path Language (XPath)*, Version 1.0, W3C Recommendation 16,
496 November 1999. Available at <http://www.w3.org/TR/xpath>

497

A. Acknowledgments

498 The following individuals contributed to the development of the specification:

499 Anne Anderson
500 Anthony Nadalin
501 Bill Parducci
502 Daniel Engovatov
503 Don Flinn
504 Ed Coyne
505 Ernesto Damiani
506 Frank Siebenlist
507 Gerald Brose
508 Hal Lockhart
509 Haruyuki Kawabe
510 James MacLean
511 John Merrells
512 Ken Yagen
513 Konstantin Beznosov
514 Michiharu Kudo
515 Michael McIntosh
516 Pierangela Samarati
517 Pirasenna Velandai Thiyagarajan
518 Polar Humenn
519 Rebekah Metz
520 Ron Jacobson
521 Satoshi Hada
522 Sekhar Vajjhala
523 Seth Proctor
524 Simon Godik
525 Steve Anderson
526 Steve Crocker
527 Suresh Damodaran
528 Tim Moses
529 Von Welch
530

531

B. Notices

532 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
533 might be claimed to pertain to the implementation or use of the technology described in this document or
534 the extent to which any license under such rights might or might not be available; neither does it
535 represent that it has made any effort to identify any such rights. Information on OASIS's procedures with
536 respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights
537 made available for publication and any assurances of licenses to be made available, or the result of an
538 attempt made to obtain a general license or permission for the use of such proprietary rights by
539 implementors or users of this specification, can be obtained from the OASIS Executive Director.

540 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications,
541 or other proprietary rights which may cover technology that may be required to implement this
542 specification. Please address the information to the OASIS Executive Director.

543 **Copyright © OASIS Open 2004-2005. All Rights Reserved.**

544 This document and translations of it may be copied and furnished to others, and derivative works that
545 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
546 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
547 notice and this paragraph are included on all such copies and derivative works. However, this document
548 itself does not be modified in any way, such as by removing the copyright notice or references to OASIS,
549 except as needed for the purpose of developing OASIS specifications, in which case the procedures for
550 copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required
551 to translate it into languages other than English.

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

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