

## Hierarchical Resource profile of XACML

### Committee Draft 01, 30 September 2004

**Document identifier:**

access\_control-xacml-2.0-hier\_profile-spec-cd-01

**Location:**

[http://docs.oasis-open.org/xacml/access\\_control-xacml-2.0-hier\\_profile-spec-cd-01.pdf](http://docs.oasis-open.org/xacml/access_control-xacml-2.0-hier_profile-spec-cd-01.pdf)

**Editor:**

Anne Anderson, Sun Microsystems ([anne.anderson@sun.com](mailto: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 Committee Draft within the OASIS Access Control TC.

Access Control TC members should send comments on this specification to the [xacml@lists.oasis-open.org](mailto:xacml@lists.oasis-open.org) list. Others may use the following link and complete the comment form: [http://oasis-open.org/committees/comments/form.php?wg\\_abbrev=xacml](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/tc\\_home.php?wg\\_abbrev=xacml](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)).

For any errata page for this specification, please refer to the Access Control TC web page ([http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=xacml](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)).

Copyright © OASIS Open 2004 All Rights Reserved.

---

29 **Table of Contents**

30 1 Introduction.....3

31 1.1 Terminology.....4

32 1.2 Notation.....5

33 2 Representing the identity of a node.....6

34 2.1 Nodes in XML documents.....6

35 2.2 Nodes in resources that are not XML documents.....6

36 3 Requesting access to a node.....8

37 3.1 Nodes in an XML document.....8

38 3.2 Nodes in a resource that is not an XML document.....9

39 4 Stating policies that apply to nodes.....11

40 4.1 Policies applying to nodes in any hierarchical resource.....11

41 4.2 Policies applying only to nodes in XML documents.....11

42 4.3 Policies applying only to nodes in non-XML resources.....11

43 5 New DataType.....13

44 5.1 xpath-expression.....13

45 6 New attribute identifiers.....14

46 6.1 document-id.....14

47 6.2 resource-parent.....14

48 6.3 resource-ancestor.....14

49 6.4 resource-ancestor-or-self.....14

50 7 New profile identifiers.....15

51 8 References.....16

52

---

# 1 Introduction

53

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

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

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

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

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

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

84 Support for each of these facilities is optional.

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

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

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

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

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

## 118 1.1 Terminology

119 **Access** - Performing an **action**.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 155 1.2 Notation

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

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

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

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

167 Example code listings appear like this.

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

---

## 2 Representing the identity of a node

*{Normative}*

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

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

### 2.1 Nodes in XML documents

*{Normative, but optional}*

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

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

### 2.2 Nodes in resources that are not XML documents

*{Normative, but optional}*

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

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

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

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

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

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

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

The following canonicalization SHALL be used.

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

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



---

## 3 Requesting access to a node

223

224 *{Normative}*

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

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

### 3.1 Nodes in an XML document

234

235 *{Normative, but optional}*

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

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

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



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

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

288 Additional **attributes** MAY be included in the <Resource> element. In particular, the following  
289 **attribute** MAY be included.

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

## 296 3.2 Nodes in a resource that is not an XML document

### 297 {Normative, but optional}

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

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

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

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

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

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

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

358 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*

400 in that **resource**.

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

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

---

## 403 5 New DataType

404 *{Normative, but optional}*

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

### 408 5.1 xpath-expression

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

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

---

## 415 6 New attribute identifiers

416 *{Normative, but optional}*

### 417 6.1 document-id

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

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

### 425 6.2 resource-parent

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

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

### 432 6.3 resource-ancestor

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

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

### 439 6.4 resource-ancestor-or-self

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

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

---

## 447 7 New profile identifiers

448 **{normative}**

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

451 Section 2.1: *Nodes in XML documents*

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

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

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

455 Section 3.1: *Nodes in an XML document*

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

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

459 urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-  
460 req:resource-parent

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

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

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

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

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

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

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

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



---

## 8 References

475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494

- [ISO10181-3]** ISO/IEC JTC 1, *Information technology -- Open Systems Interconnection -- Security frameworks for open systems: Access control framework*, ISO/IEC 10181-3:1996, 1996.
- [RFC1034]** P. Mockapetris, *DOMAIN NAMES – CONCEPTS AND FACILITIES*, IETF RFC 1034, November 1987, <ftp://ftp.isi.edu/in-notes/rfc1034.txt>
- [RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, IETF RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.
- [RFC2396]** T. Berners-Lee, et al., *Uniform Resource Identifiers (URI): Generic Syntax*, <http://www.ietf.org/rfc/rfc2396.txt>, IETF RFC 2396, August 1998.
- [RFC3198]** A. Westerinen, et al., *Terminology for Policy-Based Management*, <http://www.ietf.org/rfc/rfc3198.txt>, IETF RFC 3198, November 2001.
- [MULTIPLE]** A. Anderson, ed., *Multiple Resource profile of XACML*, Committee Draft 01, 30 September 2004, [http://docs.oasis-open.org/xacml/access\\_control-xacml-2.0-mult\\_profile-spec-cd-01.pdf](http://docs.oasis-open.org/xacml/access_control-xacml-2.0-mult_profile-spec-cd-01.pdf)
- [XACML]** S. Godik, T. Moses, eds., *OASIS eXtensible Access Control Markup Language (XACML) Version 2.0*, Committee Draft 01, 16 September 2004, [http://docs.oasis-open.org/xacml/access\\_control-xacml-2.0-core-spec-cd-01.pdf](http://docs.oasis-open.org/xacml/access_control-xacml-2.0-core-spec-cd-01.pdf).
- [XPath]** *XML Path Language (XPath)*, Version 1.0, W3C Recommendation 16, November 1999. Available at <http://www.w3.org/TR/xpath>

---

495 **A. Acknowledgments**

496 The editor would like to acknowledge the contributions of the OASIS XACML Technical Committee,  
497 whose voting members at the time of publication were:

- 498 • Frank Siebenlist, Argonne National Laboratory
- 499 • Daniel Engovatov, BEA Systems, Inc.
- 500 • Hal Lockhart, BEA Systems, Inc.
- 501 • Rebekah Metz, Booz Allen Hamilton
- 502 • Ronald Jacobson, Computer Associates
- 503 • Tim Moses, Entrust
- 504 • Simon Godik, GlueCode Software
- 505 • Bill Parducci, GlueCode Software
- 506 • Michiharu Kudo, IBM
- 507 • Michael McIntosh, IBM
- 508 • Anthony Nadalin, IBM
- 509 • Steve Anderson, OpenNetwork
- 510 • Anne Anderson, Sun Microsystems
- 511 • Seth Proctor, Sun Microsystems
- 512 • Polar Humenn, Syracuse University
- 513 • Edward Coyne, Veterans Health Administration

514

## B. Revision History

515

Rev	Date	By Whom	What
CD-01	30 Sept 2004	Anne Anderson	Committee Draft

---

## C. Notices

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

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

528 **Copyright © OASIS Open 2004. All Rights Reserved.**

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

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

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