



XACML v3.0 Multiple Decision Profile Version 1.0

Committee Draft 03

11 March 2010

Specification URIs:

This Version:

<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cd-03-en.html>
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cd-03-en.doc> (Authoritative)
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cd-03-en.pdf>

Previous Version:

<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cd-1-en.html>
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cd-1-en.doc> (Authoritative)
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cd-1-en.pdf>

Latest Version:

<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-en.html>
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-en.doc> (Authoritative)
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-en.pdf>

Technical Committee:

OASIS eXtensible Access Control Markup Language (XACML) TC

Chair(s):

Bill Parducci, <bill@parducci.net>
Hal Lockhart, Oracle <hal.lockhart@oracle.com>

Editor(s):

Erik Rissanen, Axiomatics AB <erik@axiomatics.com>

Related work:

This specification replaces or supercedes:

- [Multiple resource profile of XACML v2.0](#)

This specification is related to:

- [eXtensible Access Control Markup Language \(XACML\) Version 3.0, CD 03](#)

Declared XML Namespace(s):

None

Abstract:

This document provides a profile for requesting more than one access control decision in a single XACML Request Context, or for requesting a single combined decision based on multiple individual decisions.

Status:

This document was last revised or approved by the eXtensible Access Control Markup Language (XACML) TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

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

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/xacml/>.

Notices

Copyright © OASIS® 2010. All Rights Reserved.

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

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

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

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

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

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

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

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

Table of Contents

1	Introduction.....	5
1.1	Glossary.....	5
1.2	Abbreviated identifiers	5
1.3	Terminology	6
1.4	Normative References	6
1.5	Non-Normative References	6
2	Requests for multiple decisions.....	7
2.1	Nodes identified by “scope”	7
2.1.1	Profile URI	7
2.1.2	Original request context	7
2.1.3	Semantics.....	7
2.2	Nodes identified by XPath	8
2.2.1	Profile URI	8
2.2.2	Original request context	8
2.2.3	Semantics.....	8
2.3	Repeated <Attributes> categories	8
2.3.1	Profile URI	9
2.3.2	Original request context	9
2.3.3	Semantics.....	9
2.4	By reference to <Attributes> elements	9
2.4.1	Profile URI	9
2.4.2	Original request context	9
2.4.3	Semantics.....	9
3	Requests for a combined decision	11
3.1	Profile URI.....	11
4	Conceptual model for creating Individual Decision Requests	12
5	New attribute identifiers	13
5.1	“scope”	13
6	New profile identifiers	14
7	Conformance	15
7.1	Processor of requests for multiple decisions as nodes identified by “scope”	15
7.2	Processor of requests for multiple decisions as nodes identified by XPath	15
7.3	Processor of requests for multiple decisions by multiple <Attributes> elements	15
7.4	Processor of requests for multiple decisions by reference to <Attributes> elements.....	15
7.5	Processor of requests for a combined decision.....	15
A.	Acknowledgements	16
B.	Revision History.....	17

1 Introduction

{Non-normative}

The policy evaluation performed by an XACML Policy Decision Point, or PDP, is defined in terms of a single decision request in the XACML Specification **[XACML]**, with the authorization decision contained in a single `<Result>` element of the response context. A Policy Enforcement Point, or PEP, however, may wish to submit a single request context for multiple access control decisions, and may wish to obtain a single response context that contains a separate authorization decision (`<Result>` element) for each requested decision. Such a request context might be used to avoid sending multiple decision request messages between a PEP and PDP, for example. Additionally, a PEP may wish to submit a single request context for multiple decisions, and may wish to obtain a single authorization decision (`<Result>` element) that indicates whether access is permitted to all of the requested decisions. Such a request context might be used when the requester wants access to an entire XML document, to an entire sub-tree of elements in such a document, or to an entire file system directory with all its subdirectories and files, for example.

This Profile describes several ways in which a PEP can request multiple authorization decisions in a single request context, and how the result of each such authorization decision is represented in the single response context that is returned to the PEP.

This Profile also describes a mechanism by which a PEP can request a single combined authorization decision in response to a request for multiple decisions.

Support for each of the mechanisms described in this Profile is optional for compliant XACML implementations.

1.1 Glossary

Hierarchical resource

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

Node

An individual resource that is part of a *hierarchical resource*.

1.2 Abbreviated identifiers

Commonly used resource attributes are abbreviated as follows:

“resource-id” attribute

A resource attribute with an `AttributeId` of “urn:oasis:names:tc:xacml:1.0:resource:resource-id”.

“scope” attribute

A resource attribute with an `AttributeId` of “urn:oasis:names:tc:xacml:2.0:resource:scope”. See Section 5.1 for more information about this attribute.

“content-selector”

An attribute with an `AttributeId` of “urn:oasis:names:tc:xacml:3.0:content-selector”. See **[Hierarchical]** for more information about this attribute.

“multiple:content-selector”

An attribute with an `AttributeId` of “urn:oasis:names:tc:xacml:3.0:profile:multiple:content-selector”. See section 2.2 for more information about this attribute.

42 1.3 Terminology

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

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

49 `Example code listings appear like this.`

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

53 1.4 Normative References

- 54 **[Hierarchical]** OASIS Committee Draft 03, *XACML v3.0 Hierarchical Resource Profile Version*
55 *1.0*, 11 March 2010, [http://docs.oasis-open.org/xacml/3.0/xacml-3.0-hierarchical-
57 v1-spec-cd-03-en.doc](http://docs.oasis-open.org/xacml/3.0/xacml-3.0-hierarchical-
56 v1-spec-cd-03-en.doc)
- 57 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
58 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 59 **[XACML]** OASIS Committee Draft 03, *eXtensible Access Control Markup Language*
60 *(XACML) Version 3.0*, 11 March 2010, [http://docs.oasis-
62 open.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.doc](http://docs.oasis-
61 open.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.doc)
- 62 **[XPath]** *XML Path Language (XPath)*, Version 1.0, W3C Recommendation 16, November
63 1999. Available at <http://www.w3.org/TR/xpath>

64 1.5 Non-Normative References

65 **None**

2 Requests for multiple decisions

{Optional}

A single XACML request context MAY represent a request for multiple access control decisions. The syntax and semantics of such requests and responses are specified in this Section.

The <Result> elements produced by evaluating a request for multiple access control decisions SHALL be identical to those that would be produced from a series of requests, each requesting exactly one of the decisions. Each such decision is called an Individual Decision. The conceptual request context that corresponds to each <Result> element is called an Individual Decision Request. This mapping of an original request context containing multiple authorization decision requests to Individual Decision Requests, and the corresponding mapping of multiple authorization decisions to multiple <Result> elements in a single response context MAY be performed by the Context Handler described in the non-normative Data-flow model of the core XACML specification [XACML].

Several ways of specifying requests for multiple access control decisions are described in the following Sections. Each way of specifying requests describes the Individual Decision Requests that correspond to the <Result> elements in the response context.

A single XACML request context submitted by a PEP MAY use more than one of these ways of requesting access to multiple decisions.

2.1 Nodes identified by “scope”

{Optional}

This Section describes the use of two values for the “scope” resource attribute to specify a request for access to multiple resources in a hierarchy. This syntax MAY be used with any *hierarchical resource* [Hierarchical] which is not an XML document. The “scope” resource attribute is defined in Section 5.

2.1.1 Profile URI

The following URI SHALL be used as a URI identifier for the functionality specified in this Section of this Profile. This identifier represents metadata about this specification and implementations implementing this specification. The identifier MAY be used to describe capabilities of an implementation or to make other references to this specification.

- urn:oasis:names:tc:xacml:3.0:profile:multiple:scope

2.1.2 Original request context syntax

The original XACML request context <Attributes> element in the resource category SHALL contain a “scope” attribute with a value of either “Children”, or “Descendants”.

2.1.3 Semantics

Such a request context SHALL be interpreted as a request for access to a set of *nodes* in a hierarchy relative to the single *node* specified in the “resource-id” attribute. If the value of the “scope” attribute is “Children”, each Individual Decision Request is for the one *node* indicated by the “resource-id” attribute (or attributes, where the single resource has multiple normative identifiers) and all of its immediate child *nodes*. If the value of the “scope” attribute is “Descendants”, the Individual Decision Request is for the one *node* indicated by the “resource-id” attribute and all of its descendant *nodes*.

Each Individual Decision Request SHALL be identical to the original request context with two exceptions: the “scope” attribute SHALL NOT be present and the <Attributes> element in the resource category SHALL represent a single Individual Resource. This <Attributes> element SHALL contain at least one “resource-id” attribute, and all values for such attributes SHALL be unique, normative identities of

108 the Individual Resource. If the “resource-id” attribute in the original request context contained an
109 Issuer, the “resource-id” attributes in the Individual Resource Request SHALL contain the same
110 Issuer. The “resource-id” attributes in the Individual Decision Request SHALL contain the same
111 IncludeInResult.value as the “resource-id” attribute in the original request context
112 Neither XACML nor this Profile specifies how the Context Handler obtains the information required to
113 determine which **nodes** are children or descendants of a given **node**.

114 2.2 Nodes identified by XPath

115 {Optional}

116 This Section describes use of an XPath [**XPath**] expression in the “multiple:content-selector”
117 attribute to specify a request for access described by multiple **nodes** in an XML document. This syntax
118 SHALL be used only with resources, subjects, actions or other categories which are or are described by
119 XML documents.

120 2.2.1 Profile URI

121 The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this
122 Profile. This identifier represents metadata about this specification and implementations implementing this
123 specification. This identifier MAY be used to describe capabilities of an implementation or to make other
124 references to this specification.

- 125 • urn:oasis:names:tc:xacml:3.0:profile:multiple:xpath-expression

126 2.2.2 Original request context

127 The original XACML request context <Attributes> element SHALL contain a <Content> element and
128 a “multiple:content-selector” attribute with a DataType of “urn:oasis:names:tc:xacml:3.0:data-
129 type:xpathExpression”, such that the <AttributeValue> of the “multiple:content-selector”
130 attribute is an XPath expression that evaluates to a nodeset that represents multiple nodes in the
131 <Content> element.

132 2.2.3 Semantics

133 Such a request context SHALL be interpreted as a request for individual decisions regarding each of the
134 nodes in the nodeset selected by the XPath expression given in the <AttributeValue> of the
135 “multiple:content-selector” attribute.

136 Each Individual Decision Request SHALL be identical to the original request context with two exceptions:
137 the “multiple:content-selector” attribute SHALL NOT be present and an added “content-
138 selector” attribute value SHALL be an XPath expression that evaluates to a single node in the
139 <Content> element. If the “multiple:content-selector” attribute in the original request context
140 contained an Issuer, the “content-selector” attribute in the Individual Decision Request SHALL
141 contain the same Issuer. The “content-selector” attribute in the Individual Decision Request
142 SHALL contain the same IncludeInResult as the “multiple:content-selector” attribute in the
143 original request context,

144 If multiple <Attributes> elements in different categories contain a “multiple:content-selector”
145 attribute, then the set of Individual Decision Requests will be formed from the the cross product of the
146 nodesets selected by the “multiple:content-selector” XPath expressions in the different different
147 categories. See Section 4 for detailed description of the processing model.

148 2.3 Repeated <Attributes> categories

149 {Optional}

150 This Section describes use of multiple <Attributes> elements with repeated category in a request
151 context to specify a request for access to multiple decisions. This syntax MAY be used with any resource

152 or resources, or any other category, regardless of whether they are XML documents or not and
153 regardless of whether they are *hierarchical resources* [**Hierarchical**] or not.

154 **2.3.1 Profile URI**

155 The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this
156 Profile. This identifier represents metadata about this specification and implementations implementing this
157 specification. This identifier MAY be used to describe capabilities of an implementation or to make other
158 references to this specification

- 159 • urn:oasis:names:tc:xacml:3.0:profile:multiple:repeated-attribute-categories

160 **2.3.2 Original request context**

161 The XACML request context SHALL contain multiple `<Attributes>` elements with equal category.

162 **2.3.3 Semantics**

163 Such a request context SHALL be interpreted as a request for access to all situations specified in the
164 individual `<Attributes>` elements. Each `<Attributes>` element SHALL represent one Individual
165 Resource, subject, or another category unless that element utilizes the other mechanisms described in
166 this Profile.

167 For each combination of repeated `<Attributes>` elements, one Individual Decision Request SHALL be
168 created. This Individual Request SHALL be identical to the original request context with one exception:
169 only one `<Attributes>` element of each repeated category SHALL be present. If such a
170 `<Attributes>` element contains a “scope” attribute having any value other than “Immediate”, then the
171 Individual Request SHALL be further processed according to the processing model specified in Section 4.
172 This processing may involve decomposing the one Individual Decision Request into other Individual
173 Decision Requests before evaluation by the PDP.

174 **2.4 By reference to `<Attributes>` elements**

175 **{Optional}**

176 This section describes use of a list of references to `<Attributes>` elements to construct multiple
177 individual `<Request>` elements.

178 **2.4.1 Profile URI**

179 The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this
180 Profile. This identifier represents metadata about this specification and implementations implementing this
181 specification. This identifier MAY be used to describe capabilities of an implementation or to make other
182 references to this specification.

- 183 • urn:oasis:names:tc:xacml:3.0:profile:multiple:reference

184 **2.4.2 Original request context**

185 The original XACML `<Request>` element SHALL contain a `<MultiRequests>` element.

186 **2.4.3 Semantics**

187 Such a request context SHALL be interpreted as multiple individual request contexts specified by
188 references to `<Attributes>` elements.

189 The context handler MUST construct a new `<Request>` element for each `<RequestReference>`
190 element contained in the `<MultiRequests>` element, and process the generated `<Request>` element.

191 Each `<RequestReference>` element contains one or more `<AttributesReference>` elements,
192 each of which refers to the `xml:id` XML attribute of one of the `<Attributes>` elements in the enclosing

193 original <Request> element. The generated <Request> element MUST be identical to a <Request>
194 element which contains the referenced <Attributes> elements.

195 The result(s) of each such generated <Request> element MUST be included as one or more <Result>
196 elements in the <Response> element corresponding to the original <Request> element. There may be
197 multiple results for a single generated <Request> element when the generated <Request> element
198 makes use of one or more of the other multiple decision request schemes in this profile. There MUST be
199 exactly one <Response> element for the original <Request> element.

200 If a <RequestReference> contains an invalid reference, then the corresponding <Result> MUST
201 contain an Indeterminate decision with status code urn:oasis:names:tc:xacml:1.0:status:syntax-error.

202 3 Requests for a combined decision

203 {Optional}

204 A request for multiple decisions as specified by any of the schemes in section 2 MAY in addition specify
205 that the Individual Decisions be combined into a single aggregated decision and that only this single
206 combined decision will be returned to the *PEP*.

207 If the `CombinedDecision` attribute of the initial `<Request>` is `True`, then the `<Response>` **MUST**
208 contain only a single combined decision in a single `<Result>` element, and the following apply to the
209 combined decision, in the given order.

- 210 1. There **MUST NOT** be any `<Attributes>` elements in the combined `<Result>`, regardless of
211 the values of any of the `IncludeInResult` attributes of the `<Attributes>` elements.
- 212 2. If any of the individual results to be combined contain any obligations or advice, then the
213 combined decision **MUST** be Indeterminate, with status code
214 `urn:oasis:names:tc:xacml:1.0:status:processing-error`.
- 215 3. If all the individual results to be combined have the same decision value (Permit, Deny,
216 NotApplicable or Indeterminate), then the combined decision **MUST** be equal to this common
217 decision value. If the common decision value is Indeterminate, then the status code **MUST** be
218 `urn:oasis:names:tc:xacml:1.0:status:processing-error`. If the common decision value is not
219 Indeterminate, then the status code **MUST** be `urn:oasis:names:tc:xacml:1.0:status:ok`.
- 220 4. Otherwise the combined Decision **MUST** be Indeterminate, with status code
221 `urn:oasis:names:tc:xacml:1.0:status:processing-error`.

222 3.1 Profile URI

223 The following URI **SHALL** be used as the URI identifier for the functionality specified in this Section of this
224 Profile. This identifier represents metadata about this specification and implementations implementing this
225 specification. This identifier **MAY** be used to describe capabilities of an implementation or to make other
226 references to this specification.

- 227 • `urn:oasis:names:tc:xacml:3.0:profile:multiple:combined-decision`

228

229 4 Conceptual model for creating Individual Decision 230 Requests

231 {Mandatory}

232 This profile specifies several independent schemes for Multiple Decision Requests in sections 2 and 3.
233 Any combination of features described by these schemes MAY be used in an initial request. This section
234 defines a normative processing model to create Individual Decision Requests from an initial request
235 context in which one or more features of the multiple profile are present. This Profile does NOT
236 REQUIRE that the implementation of the evaluation of a request for access to multiple decisions conform
237 to the model below or that actual Individual Decision Requests be constructed. The Profile REQUIRES
238 only that the <Result> elements SHALL be the same as if the model below were used. An
239 implementation MUST produce identical results to those that would be produced by performing the
240 following operations in the given order.

- 241 1. If there is no <MultiRequests> element, then use the initial request context as input to step 2
242 and skip the processing in this step. If the initial request contains a <MultiRequests> element,
243 then the initial request is processed as specified by section 2.4. If there are any Indeterminate
244 results during this processing, include the Indeterminates in the final result defined in step 5
245 below, while each valid request is processed in turn as defined by step 2.
- 246 2. For each request from the previous step which contains <Attributes> elements with repeated
247 values for the Category XML attribute, perform the processing defined in section 2.3. The
248 outputs of this processing and any requests without repeated categories form the inputs for the
249 step 3. If there are any Indeterminate results, include them in the final result defined in step 5
250 below, while each valid request is processed in turn as defined by step 3.
- 251 3. At this stage each request from the previous step can contain a request for multiple decisions as
252 either a scope attribute or as an XPath expression in a “multiple:content-selector” attribute. If
253 neither is present, proceed to step 4. If either is present, then process the request as defined of
254 one of the following sub steps:
 - 255 a. If the request specifies a scope attribute, process the request as specified by section 2.1.
256 If there are any Indeterminate results, include them in the final result defined in step 5
257 below, while each valid request is processed in turn as defined by step 4.
 - 258 b. If the request specifies a “multiple:content-selector” attribute with an XPath, process the
259 request as specified by section 2.2. If there are any Indeterminate results, include them in
260 the final result defined in step 5 below, while each valid request is processed in turn as
261 defined by step 4.
- 262 4. At this stage each request is a request for an individual authorization decision. Each request
263 MUST be processed by the **PDP** as an individual access control request according to the XACML
264 core specification and any implemented profiles and extensions.
- 265 5. At this stage all requests have been processed by the **PDP** and the inputs to this step are all
266 collected Indeterminate results from the previous steps and all the individual results from step 4. If
267 applicable, perform the processing defined in Section 3.

268

269 5 New attribute identifiers

270 5.1 “scope”

271 The following identifier is used as the `AttributeId` of a resource attribute that indicates the scope of a
272 request for access in a single `<Attributes>` element of a request context.

- 273 • `urn:oasis:names:tc:xacml:2.0:resource:scope`

274 The attribute SHALL have a `DataType` of “`http://www.w3.org/2001/XMLSchema#string`”.

275 The valid values for this attribute are listed below, along with a reference to the Section of this Profile or to
276 the core XACML specification that describes how the `<Attributes>` element with the resource category
277 is to be processed. An implementation MAY support any subset of these values, including the empty set.

- 278 • “Immediate” - The `<Attributes>` element refers to a single non-**hierarchical resource** or to a
279 single **node** in a **hierarchical resource**. This is the default value, if no “scope” attribute is present.
280 The `<Attributes>` element SHALL be processed according to the core XACML specification [XACML].
- 281 • “Children” - The `<Attributes>` element refers to multiple resources in a hierarchy. The set of
282 resources consists of a single **node** described by the “resource-id” resource attribute and of all
283 that **node**'s immediate children in the hierarchy. The `<Attributes>` element SHALL be processed
284 according to Section 2.1 of this Profile.
- 285 • “Descendants” - The `<Attributes>` element refers to multiple resources in a hierarchy. The set
286 of resources consists of a single **node** described by the “resource-id” resource attribute and of all
287 that **node**'s descendants in the hierarchy. The `<Attributes>` element SHALL be processed
288 according to Section 2.1 of this Profile.

289 6 New profile identifiers

290 The following URI values SHALL be used as URI identifiers for the functionality specified in various
291 Sections of this Profile. These identifiers represent metadata about this specification and implementations
292 implementing this specification. These identifiers MAY be used to describe capabilities of an
293 implementation or to make other references to this specification

294 Section 2.1: “scope attribute of “children” or “descendants” in <Attributes>: Non-XML resources

- 295 • urn:asis:names:tc:xacml:3.0:profile:multiple:scope

296 Section 2.2: XPath expression in “multiple:content-selector” attribute

- 297 • urn:asis:names:tc:xacml:3.0:profile:multiple:xpath-expression

298 Section 2.3: Multiple <Attributes> elements with repeated attribute categories

- 299 • urn:asis:names:tc:xacml:3.0:profile:multiple:repeated-attribute-categories

300 Section 2.4: By reference to <Attributes> elements

- 301 • urn:asis:names:tc:xacml:3.0:profile:multiple:reference

302 Section 3: Requests for a combined decision

- 303 • urn:asis:names:tc:xacml:3.0:profile:multiple:combined-decision

304 7 Conformance

305 An implementation may conform to this specification in one or more of the following ways.

306 7.1 Processor of requests for multiple decisions as nodes identified 307 by “scope”

308 An implementation conforms as a processor of requests for multiple resources as nodes identified by
309 “scope” if it is able to process XACML requests in the manner described in sections 2.1 and 4 of this
310 specification. Conformance to this MAY be indicated with the identifier
311 urn:oasis:names:tc:xacml:3.0:profile:multiple:scope.

312 7.2 Processor of requests for multiple decisions as nodes identified 313 by XPath

314 An implementation conforms as a processor of requests for multiple decisions as nodes identified by
315 XPath if it is able to process XACML requests in the manner described in sections 2.2 and 4 of this
316 specification. Conformance to this MAY be indicated with the identifier
317 urn:oasis:names:tc:xacml:3.0:profile:multiple:xpath-expression.

318 7.3 Processor of requests for multiple decisions by multiple 319 <Attributes> elements

320 An implementation conforms as a processor of requests for multiple decisions by multiple
321 <Attributes> elements if it is able to process XACML requests in the manner described in sections 2.3
322 and 4 of this specification. Conformance to this MAY be indicated with the identifier
323 urn:oasis:names:tc:xacml:3.0:profile:multiple:repeated-attribute-categories.

324 7.4 Processor of requests for multiple decisions by reference to 325 <Attributes> elements

326 An implementation conforms as a processor of requests for multiple decisions by references to
327 <Attributes> elements if it is able to process XACML requests in the manner described in sections 2.4
328 and 4 of this specification. Conformance to this MAY be indicated with the identifier
329 urn:oasis:names:tc:xacml:3.0:profile:multiple:reference.

330 7.5 Processor of requests for a combined decision

331 An implementation conforms as a processor of requests for a combined decision if it is able to process
332 XACML requests in the manner described in section 3 and 4 of this specification. Conformance to this
333 MAY be indicated with the identifier urn:oasis:names:tc:xacml:3.0:profile:multiple:combined-decision.

334 **A. Acknowledgements**

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

337
338 Anil Saldhana
339 Anil Tappetla
340 Anne Anderson
341 Anthony Nadalin
342 Bill Parducci
343 Craig Forster
344 David Chadwick
345 David Staggs
346 Dilli Arumugam
347 Duane DeCouteau
348 Erik Rissanen
349 Gareth Richards
350 Hal Lockhart
351 Jan Herrmann
352 John Tolbert
353 Ludwig Seitz
354 Michiharu Kudo
355 Naomaru Itoi
356 Paul Tyson
357 Prateek Mishra
358 Rich Levinson
359 Ronald Jacobson
360 Seth Proctor
361 Sridhar Muppidi
362 Tim Moses
363 Vernon Murdoch
364
365
366

367

B. Revision History

368

[optional; should not be included in OASIS Standards]

369

Revision	Date	Editor	Changes Made
WD 1	[Rev Date]	Erik Rissanen	Initial update to XACML 3.0.
WD 2	28 Dec 2007	Erik Rissanen	Update to current OASIS template.
WD 3	4 Nov 2008	Erik Rissanen	Define behavior for the IncludeInResult attribute.
WD 4	3 Mar 2009	Erik Rissanen	Added the new <MultiRequests> scheme.
WD 5		Erik Rissanen	Changed error behavior in <MultiRequests> Clarified some text Editorial cleanups Conformance section
WD 6	14 Dec 2009	Erik Rissanen	Renamed to “Multiple Decision Profile”. Clarified meaning of metadata identifiers. Remove “scope” for XML resources. Replaced scope EntireHierarchy with decision combining algorithm. Added more detailed text about nesting of schemes.
WD 07	17 Dec 2009	Erik Rissanen	Update acknowledgments Don’t allow obligations in combined decisions Fix formatting issues
WD 08		Erik Rissanen	Drop decision combining algorithms in favor of a more restricted (and safer) decision combining scheme.
WD 09	12 Jan 2010	Erik Rissanen	Updated cross references Fix typos and improve wording. Updated acknowledgments
WD 10	8 Mar 2010	Erik Rissanen	Updated cross references Fixed OASIS formatting issues

370

371