XACML v3.0 Multiple Decision Profile Version 1.0

Committee Specification 01

10 August 2010

Specification URIs:
This Version:
http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cs-01-en.html
http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cs-01-en.doc (Authoritative)
http://docs.oasis-open.org/xacml/3.0/xacml-3.0-multiple-v1-spec-cs-01-en.pdf

Previous 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

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

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

{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.

1.3 Terminology

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

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

```
Example code listings appear like this.
```

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

1.4 Normative References

[Hierarchical] OASIS Committee Specification 01, XACML v3.0 Hierarchical Resource Profile Version 1.0, August 2010, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-hierarchical-v1-spec-cs-01-en.doc

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[XACML] OASIS Committee Specification 01, eXtensible access control markup language (XACML) Version 3.0. August 2010. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.doc

[XPath] *XML Path Language (XPath)*, Version 1.0, W3C Recommendation 16, November 1999. Available at http://www.w3.org/TR/xpath

1.5 Non-Normative References

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

the Individual Resource. If the "resource-id" attribute in the original request context contained an Issuer, the "resource-id" attributes in the Individual Resource Request SHALL contain the same Issuer. The "resource-id" attributes in the Individual Decision Request SHALL contain the same IncludeInResult.value as the "resource-id" attribute in the original request context

Neither XACML nor this Profile specifies how the Context Handler obtains the information required to determine which ***nodes*** are children or descendants of a given ***node***.

2.2 Nodes identified by XPath

{Optional}

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

2.2.1 Profile URI

The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this Profile. This identifier represents metadata about this specification and implementations implementing this specification. This 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:xpath-expression

2.2.2 Original request context

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

2.2.3 Semantics

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

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

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

2.3 Repeated <Attributes> categories

{Optional}

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

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

2.3.1 Profile URI

The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this Profile. This identifier represents metadata about this specification and implementations implementing this specification. This 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:repeated-attribute-categories

2.3.2 Original request context

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

2.3.3 Semantics

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

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

2.4 By reference to <Attributes> elements

{Optional}

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

2.4.1 Profile URI

The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this Profile. This identifier represents metadata about this specification and implementations implementing this specification. This 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:reference

2.4.2 Original request context

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

2.4.3 Semantics

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

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

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

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

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

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

3 Requests for a combined decision

{Optional}

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

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

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

3.1 Profile URI

The following URI SHALL be used as the URI identifier for the functionality specified in this Section of this Profile. This identifier represents metadata about this specification and implementations implementing this specification. This 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:combined-decision

4 Conceptual model for creating Individual Decision Requests

{Mandatory}

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

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

5 New attribute identifiers

5.1 “scope”

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

- urn:oasis:names:tc:xacml:2.0:resource:scope

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

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

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

6 New profile identifiers

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

Section 2.1: "scope attribute of "`children`" or "`descendants`" in `<Attributes>`: Non-XML resources

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

Section 2.2: XPath expression in "`multiple:content-selector`" attribute

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

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

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

Section 2.4: By reference to `<Attributes>` elements

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

Section 3: Requests for a combined decision

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

7 Conformance

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

7.1 Processor of requests for multiple decisions as nodes identified by "scope"

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

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

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

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

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

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

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

7.5 Processor of requests for a combined decision

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

A. Acknowledgements

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

Anil Saldhana
Anil Tappetla
Anne Anderson
Anthony Nadalin
Bill Parducci
Craig Forster
David Chadwick
David Staggs
Dilli Arumugam
Duane DeCouteau
Erik Rissanen
Gareth Richards
Hal Lockhart
Jan Herrmann
John Tolbert
Ludwig Seitz
Michiharu Kudo
Naomaru Itoi
Paul Tyson
Prateek Mishra
Rich Levinson
Ronald Jacobson
Seth Proctor
Sridhar Muppidi
Tim Moses
Vernon Murdoch

B. Revision History

[optional; should not be included in OASIS Standards]

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