# XACML Profile for Hierarchical Resources

## Working Draft 04, 3 June 2004

**Document identifier:**
xacml-profile-hierarchical-resources-1.0-draft-04

**Location:**
http://www.oasis-open.org/committees/documents.php?wg_abbrev=xacml

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

**Contributors:**
Frank Siebenlist, Argonne National Laboratory
Daniel Engovatov, BEA
Tim Moses, Entrust
Michiharu Kudo, IBM
Michael McIntosh, IBM
Anthony Nadalin, IBM
Steve Anderson, Open Network
Simon Godik, OverXeer
Bill Parducci, OverXeer
Seth Proctor, Sun Microsystems
Polar Humenn, Syracuse University

**Abstract:**
This document provides a profile for the use XACML with resources that are structured as hierarchies. The profile includes both XML document resources and resources that are not XML documents. The profile covers requesting access to hierarchical resources and specifying policies that apply to hierarchical.

**Status:**
This version of the specification is a working draft of the committee. As such, it is expected to change prior to adoption as an OASIS Standard.

Committee members should send comments on this specification to the xacml@lists.oasis-open.org list. Others should subscribe to and send comments to the xacml-comment@lists.oasis-open.org list. To subscribe, send an email message to xacml-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

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 XACML TC web page (http://www.oasis-open.org/committees/xacml/).

For any errata page for this specification, please refer to the XACML Profile for Hierarchical Resources section of the XACML TC web page (http://www.oasis-open.org/committees/xacml/).

# Table of Contents

# 1 Introduction

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

In XACML, a ***resource*** organized as a hierarchy may be a “tree” (a hierarchy with a single root) or a “forest” (a hierarchy with multiple roots), but the hierarchy may not have cycles. Another term for this type of hierarchy is a Directed Acyclic Graph or DAG. Such ***resources*** are called ***hierarchical resources*** in this Profile.

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

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

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

Support for each of these facilities is optional.

In dealing with a ***hierarchical resource***, it may be useful to request authorization decisions for multiple ***nodes*** in the ***resource*** in a single authorization decision request. Ways to make such requests are specified in a separate profile – the *XACML Profile for Requests for Multiple Resources* [MULTIPLE].

This profile for ***hierarchical resources*** assumes that all requests for ***access*** to multiple ***nodes*** in a ***hierarchical resource*** have been resolved to individual requests for ***access*** to a single ***node***.

## 1.1 Terminology

***Access - Performing an action***

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

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

***Attribute*** - Characteristic of a ***subject***, ***resource, action*** or ***environment*** that may be referenced in a ***predicate*** or ***target*** (see also – ***named attribute***)

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

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

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

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

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

***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.***

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

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

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

***Policy decision point (PDP) -*** The system entity that evaluates ***applicable policy*** and renders an ***authorization decision***. This term is defined in a joint effort by the IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in [RFC3198]. This term corresponds to "Access Decision Function" (ADF) in [ISO10181-3].

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

***Resource*** - Data, service or system component

## 1.1. Notation

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

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

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

The phrase ***{Normative, but 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 component values, square brackets (“[“, “]”) enclose optional elements, elements in quotes are literal components, and “*” 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 identity of a node in a ***resource*** that is an XML document instance SHALL be an XPath expression that evaluates to exactly that one node.

## 2.2 Nodes in resources that are not XML documents

***{Normative, but optional}***

The identity of a ***node*** in a ***hierarchical resource*** that is not 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 Distributed Name Service (DNS) name or a numeric IPv4 or IPv6 address, the DNS name SHALL be used.
- The components of the `<pathname>` portion of the URI SHALL be specified using the canonical form for such path components at the `<authority>`.
- In accordance with [RFC2396], the separator character between hierarchical components of the `<pathname>` portion of the URI SHALL be the character "`/`". Sequences of the "`/`" character SHALL be resolved to a single "`/`". ***Node*** identities SHALL NOT terminate with the "`/`" character.

- All links in the <pathname> SHALL be resolved.
- All <pathname> values SHALL be absolute.
- The "`..`" and "`.`" <pathname> components used to specify "level above this hierarchy level" and "this hierarchy level", respectively, SHALL be resolved to their actual component values.
- If there is more than one fully resolved, absolute path from a <root> at the <authority> to the represented ***node***, then a separate ***resource attribute*** with `AttributeId` "`urn:oasis:names:tc:xacml:2.0:resource:resource-id`" SHALL be included in the XACML Request for each such path.

# 3 Requesting access to a node

***{Normative}***

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

The following sections describe RECOMMENDED request ***context*** descriptions of ***access*** to ***nodes*** in ***hierarchical resources.*** Alternative representations of such requests 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 consistently for all requests for ***access*** to ***nodes*** in the given ***resource***.

## 3.1 Nodes in an XML document

***{Normative, but optional}***

In order to request ***access*** to a ***node*** in an XML document, the Request Context `<Resource>` element SHALL contain the following elements and ***attributes***.

- A `<ResourceContent>` element that contains the entire XML document instance of which the requested ***node*** is a part.
- An `<Attribute>` element with an `AttributeId` of "`urn:oasis::names:tc:xacml:2.0:resource:resource-id`" and a `DataType` of "`urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression`". The `<AttributeValue>` of this `<Attribute>` SHALL be an XPath expression that evaluates to a single ***node*** in the `<ResourceContent>` element. That single ***node*** SHALL be the ***node*** to which ***access*** is requested. This `<Attribute>` MAY specify an `Issuer`.
- An `<Attribute>` element with an `AttributeId` of "`urn:oasis::names:tc:xacml:2.0:resource:resource-parent`" and a `DataType` of "`urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression`". The `<AttributeValue>` of this ***attribute*** SHALL be an XPath expression that evaluates to a nodeset containing only a single ***node*** in the `<ResourceContent>` element, and that ***node*** SHALL be the immediate parent of the ***node*** represented in the "`resource-id`" ***attribute.*** This `<Attribute>` MAY specify an `Issuer`
- For each node in the XML document instance that is an ancestor of the ***node*** represented by the "`resource-id`" ***attribute***, an `<Attribute>` element with an AttributeId of "`urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor`" and a DataType of "`urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression`". The `<AttributeValue>` of each such ***attribute*** SHALL be an XPath expression that evaluates to a nodeset containing only a single ***node***, in the `<ResourceContent>` element, and that ***node*** SHALL be the respective ancestor ***node***. This `<Attribute>` MAY specify an `Issuer`.

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

- An `<Attribute>` element with an `AttributeId` of "`urn:oasis::names:tc:xacml:2.0:resource:document-id`" and a `DataType` of "`urn:oasis:names:tc:xacml:2.0:data-type:anyURI`". The `<AttributeValue>` of this `<Attribute>` SHALL be a URI that identifies the XML document of which the requested ***resource*** is a part. This `<Attribute>` MAY specify an `Issuer`.

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

***{Normative, but optional}***

In order to request ***access*** to a ***node*** in a ***hierarchical resource*** that is not an XML document, the Request Context <Resource> element SHALL contain the following elements and ***attributes***.

- For each normative representation of the requested ***node,*** an <Attribute> element with AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-id". The <AttributeValue> of this <Attribute> SHALL be a unique, normative identity of the ***node*** to which ***access*** is requested. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer.
- For each immediate parent of the ***node*** specified in the "resource-id" ***attribute*** or ***attributes,*** and for each normative representation of that parent ***node,*** an <Attribute> element with AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-parent". The <AttributeValue> of this <Attribute> SHALL be the normative identity of the parent ***node***. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer. Note that there may be multiple instances of this ***attribute*** if the requested ***node*** is part of a forest rather than part of a single tree, or if the parent ***node*** has more than one normative representation.
- For each ancestor of the ***node*** specified in the "resource-id" ***attribute*** or ***attributes,*** and for each normative representation of that ancestor ***node,*** an <Attribute> element with AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor". The <AttributeValue> of this <Attribute> SHALL be the normative identity of the ancestor ***node***. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer. For each "resource-parent" ***attribute***, there SHALL be a corresponding "resource-ancestor" ***attribute***. Note that there may be multiple instances of this ***attribute*** if the requested ***node*** is part of a forest rather than part of a single tree, or if the ancestor ***node*** has more than one normative representation. The values for this ***attribute*** do not necessarily reflect the position of each ancestor ***node*** in the hierarchy.

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 XML documents

***{Non-normative}***

For ***hierarchical resources*** that are 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 an XML document contained in the `<ResourceContent>` element of a request ***context.***

## 4.2 Policies applying to nodes in non XML resources

***{Non-normative}***

For ***hierarchical resources*** that are not 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*** in that ***resource.***

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

`urn:oasis:names:tc:xacml:2.0:function:anyURI-match`

## 4.3 Policies applying to nodes in any hierarchical resource

***{Non-normative}***

***Resource attributes*** with the following `AttributeId` values, described in Section 6 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-ancestor`

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

Note that a `<ResourceAttributeDesignator>` that refers to the “`resource-ancestor`” or “`resource-parent`” ***attribute*** will return a bag of values representing all ancestors or parents, respectively, of the ***resource*** to which ***access*** is being requested.

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 an `<AttributeSelector>` that selects a portion of the `<ResourceContent>` element of the `<Resource>` or may be specified using a `<ResourceAttributeDesignator>` with the “`resource-parent`” or “`resource-ancestor`” `AttributeId` value.

# 5 New data-types for hierarchical resources

***{Normative, but optional}***

The following `DataType` values MAY be supported for use with ***hierarchical resources*** or with other uses of XML schema instances within XACML request ***contexts*** or ***policies.***

## 5.1 xpath-expression

***Attribute*** values having the following `DataType` SHALL be strings that SHALL be evaluated as XPath expressions. The result of evaluating such an ***attribute*** value SHALL be the nodeset resulting from an evaluation of the XPath expression.

```
urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression
```

# 6 New attribute identifiers for hierarchical resources

***{Normative, but optional}***

## 6.1 resource-ancestor

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

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

## 6.2 resource-parent

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

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

# 7 References

## 7.1 Normative References


**[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.

**[MULTIPLE]** A. Anderson, ed., *XACML Profile for Requests for Multiple Resources*,

**[XACML]** T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML) Version 2.0, http://www.oasis-open.org/committees/xacml*

# A. Acknowledgments

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

- Frank Siebenlist, Argonne National Laboratory
- Daniel Engovatov, BEA Systems, Inc.
- Hal Lockhart, BEA Systems, Inc.
- Tim Moses, Entrust
- Maryann Hondo, IBM
- Michiharu Kudo, IBM
- Michael McIntosh, IBM
- Anthony Nadalin, IBM
- Rebekah Lepro, NASA
- Steve Anderson, OpenNetwork
- Simon Godik, Overxeer
- Bill Parducci, Overxeer
- Anne Anderson, Sun Microsystems
- Seth Proctor, Sun Microsystems
- Polar Humenn, Syracuse University

# B. Revision History

| Rev | Date | By Whom | What |
|---|---|---|---|
| 01 | 14 Apr 2004 | Anne Anderson | Initial rewrite of Section 7.13. |
| 02 | 13 May 2004 | Anne Anderson | “xpath-expression” DataType. Remove resource attributes no longer needed. New section for requesting multiple resources. Require <ResourceContent> for XML resources. Added “resource-ancestor” and “resource-parent”. |
| 03 | 25 May 2004 | Anne Anderson | Standard URI representation of non-XML nodes. Multiple resource-id Atttributes if multiple normative representations. “resource-ancestor” and “resource-parent” for any hierarchical resource. Referenced “anyURI-equal” and “anyURI-match”. |
| 04 | 2 Jun 2004 | Anne Anderson | Formatted as a separate profile, making each feature optional. |

# C. Notices

*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's procedures with respect to rights in OASIS specifications can be found at 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 implementors or users of this specification, can be obtained from the OASIS Executive Director.*

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