



# SAML V2.0 Subject Identifier Attributes Profile Version 1.0

## Committee Specification 01

16 January 2019

### Specification URIs

#### This version:

<https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.odt> (Authoritative)  
<https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.html>  
<https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.pdf>

#### Previous version:

<http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.odt> (Authoritative)  
<http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.html>  
<http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.pdf>

#### Latest version:

<https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.odt> (Authoritative)  
<https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.html>  
<https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.pdf>

#### Technical Committee:

OASIS Security Services (SAML) TC

#### Chair:

Thomas Hardjono ([hardjono@mit.edu](mailto:hardjono@mit.edu)), M.I.T.

#### Editor:

Scott Cantor ([cantor.2@osu.edu](mailto:cantor.2@osu.edu)), Internet2

#### Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- XML schema: <https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/schema/saml-subject-id-attr-v1.0.xsd>

#### Related work:

This specification is related to:

- eduPerson Object Class Specification (201602)  
<http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201602.html>.

#### Declared XML namespace:

- `urn:mace:shibboleth:metadata:1.0`

### Abstract:

This specification standardizes two new SAML Attributes to identify security subjects, as a replacement for long-standing inconsistent practice with the <saml:NameID> and <saml:Attribute> constructs, and to address recognized deficiencies with the SAML V2.0 `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` Name Identifier format.

### Status:

This document was last revised or approved by the OASIS Security Services (SAML) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=security#technical](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#technical).

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the Technical Committee's web page at <https://www.oasis-open.org/committees/security/>.

This specification is provided under the [RF on RAND Terms](#) Mode of the [OASIS IPR Policy](#), the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (<https://www.oasis-open.org/committees/security/ipr.php>).

Note that any machine-readable content (aka [Computer Language Definitions](#)) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

### Citation format:

When referencing this Work Product the following citation format should be used:

#### **[SAML-SubjectID-v1.0]**

*SAML V2.0 Subject Identifier Attributes Profile Version 1.0*. Edited by Scott Cantor. 16 January 2019. OASIS Committee Specification 01. <https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.html>. Latest version: <https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.html>.

# Notices

Copyright © OASIS Open 2019. 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 name "OASIS" is a trademark 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 <https://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

# Table of Contents

1	Introduction.....	5
1.1	IPR Policy.....	5
1.2	Terminology and Notation.....	5
1.3	Normative References.....	5
1.4	Non-Normative References.....	6
2	Motivation.....	7
2.1	Problem Statement.....	7
2.2	Relationship to Existing Work.....	8
3	SAML V2.0 Subject Identifier Attributes Profile Version 1.0.....	9
3.1	Required Information.....	9
3.2	Overview.....	9
3.3	General Purpose Subject Identifier.....	9
3.3.1	Syntax and Handling.....	9
3.3.2	Semantics and Practices.....	10
3.3.3	Example.....	10
3.4	Pairwise Subject Identifier.....	11
3.4.1	Syntax and Handling.....	11
3.4.2	Semantics and Practices.....	11
3.4.3	Implementation Strategies.....	12
3.4.4	Differences from "persistent" NameIDs.....	12
3.4.5	Example.....	12
3.5	Considerations for SAML Profiles.....	12
3.5.1	Requirements Signaling.....	12
3.5.2	Scope Filtering.....	13
3.5.2.1	Element <shibmd:Scope>.....	14
3.5.2.2	Usage Considerations.....	14
3.5.3	NameID Considerations.....	15
3.5.4	Security Considerations.....	15
4	Conformance.....	16
4.1	Conformance Clause 1: Asserting Party Implementations.....	16
4.2	Conformance Clause 2: Relying Party Implementations.....	16
	Appendix A Acknowledgments.....	17
	Appendix B Revision History.....	18

# 1 Introduction

## 1.1 IPR Policy

This specification is provided under the [RF on RAND Terms](#) Mode of the [OASIS IPR Policy](#), the mode chosen when the Technical Committee was established.

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 TC's web page (<https://www.oasis-open.org/committees/security/ipr.php>).

## 1.2 Terminology and Notation

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 [\[RFC2119\]](#).

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace <a href="#">[SAML2Core]</a> .
samlp:	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace <a href="#">[SAML2Core]</a> .
md:	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace <a href="#">[SAML2Meta]</a> .
mdattr:	urn:oasis:names:tc:SAML:metadata:attributes	This is the SAML V2.0 metadata extension for entity attributes namespace <a href="#">[MetaAttr]</a> .
shibmd:	urn:mace:shibboleth:metadata:1.0	This is a SAML V2.0 metadata extension namespace defined by this document and its accompanying schema.
xsd:	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification <a href="#">[XMLSCHEMA-2]</a> .

This specification uses the following typographical conventions in text: `<ns:Element>`, `Attribute`, **Datatype**, `OtherCode`.

This specification uses the following typographical conventions in XML listings:

```
Listings of XML schemas appear like this.
```

```
Listings of XML examples appear like this. These listings are non-normative.
```

## 1.3 Normative References

- [RFC2119]** Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- [RFC2234]** Crocker, D, Overell, P., “Augmented BNF for Syntax Specifications: ABNF”, RFC 2234, November 1997. <http://www.ietf.org/rfc/rfc2234.txt>.
- [SAML2Core]** *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. Edited by Scott Cantor, John Kemp, Rob Philpott, Eve Maler. 15 March 2005. OASIS Standard. <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

- [MetaAttr]** *SAML V2.0 Metadata Extension for Entity Attributes Version 1.0*. Edited by Scott Cantor. 4 August 2009. OASIS Committee Specification. <http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr-cs-01.pdf>. Latest version: <http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf>.
- [SAML2Errata]** *SAML V2.0 Errata*. Edited by Scott Cantor. 1 May 2012. OASIS Approved Errata. <http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.pdf>. Latest version: <http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf>
- [SAML2Meta]** *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. Edited by Scott Cantor, Jahan Moreh, Rob Philpot, Eve Maler. 15 March 2005. OASIS Standard. <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>
- [SAML2Prof]** *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. Edited by John Hughes, Scott Cantor, Jeff Hodges, Frederick Hirsch, Prateek Mishra, Rob Philpot, Eve Maler. 15 March 2005. OASIS Standard. <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>
- [XMLSCHEMA-2]** *XML Schema Part 2: Datatypes Second Edition*. Paul V. Biron, A. Malhotra, Editors. W3C Recommendation. October 28, 2004. <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>. Latest version: <http://www.w3.org/TR/xmlschema-2/>.

## 22 1.4 Non-Normative References

- [eduPerson]** Internet2, “eduPerson Object Class Specification (201602)”, February 2016. <http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201602.html>.
- [RFC4648]** Josefson, S., “The Base16, Base32, and Base64 Data Encodings”, RFC 4648, October 2006. <http://www.ietf.org/rfc/rfc4648.txt>.
- [ShibMetaExt]** Shibboleth Project, “Shibboleth Metadata Extensions V1.0”, July 2018. <https://wiki.shibboleth.net/confluence/x/QACt>.

---

## 23 2 Motivation

### 24 2.1 Problem Statement

25 Identification of subjects in security protocols and applications has a fraught history of inconsistent syntax,  
26 bugs, terrible but deeply cemented practices such as misuse of email addresses, vertical market-specific  
27 approaches, and failure to precisely communicate intended semantics and constraints. These problems  
28 lead to overly complex burdens on both asserting and relying parties to issue and consume a variety of  
29 different identifiers in different formats, many of which work poorly with off the shelf applications. Much of  
30 this is self-inflicted fragmentation due to the constant tension between fixing problems with new solutions  
31 and avoiding new solutions to ensure wider adoption.

32 SAML itself has its origins in a design philosophy that tried to avoid breaking new ground in this area, and  
33 instead attempted to design for generality, which is valuable, but did not ease adoption due to a lack of  
34 guidance. SAML also complicates itself by providing an optional, singly-appearing construct for  
35 identification (the `<saml:NameID>` element) *and* a more general multiply-appearing  
36 `<saml:Attribute>` construct that inherently overlap.

37 This, together with inconsistent technical precision by implementers and deployers, creates complexity.  
38 Deployment experience has shown that use of the NameID feature is confusing in many implementations.  
39 It also, through its presence in the SAML Single Logout protocol, potentially appears (indirectly but  
40 recoverably) in web access logs, leading to the added complexity of encryption when privacy is a  
41 consideration.

42 There is a general consensus by most federated identity practitioners around a few common  
43 requirements:

- 44 • Identifiers should be as stable as possible and should have little or no risk of reassignment to  
45 different subjects due to the lack of tight synchronization<sup>1</sup> inherent between loosely-coupled  
46 systems.
- 47 • Opaque (i.e., superficially random) identifiers are inherently more stable than name-based  
48 identifiers or email addresses in many organizations.
- 49 • Identifiers should be compact and simple to handle and manipulate.
- 50 • The ability to clearly express the scope of an identifier's uniqueness and enforce policy stipulating  
51 the asserting parties permitted to issue an identifier is crucial to federated systems and the lack of  
52 such policy has led to widely-publicized breaches.

53 Another requirement perhaps more common to education and research is the ability for different asserting  
54 parties to issue the same identifier. This is facilitated by ensuring the scope of an identifier is part of its  
55 value and not implicit in a protocol-specific construct specific to an asserting party.

56 SAML does not define an identifier that meets all of these requirements well. It does standardize a kind of  
57 NameID termed "persistent" that meets some of them in the particular case of so-called "pairwise"  
58 identification, where a given subject's identifier varies by relying party. It has seen minimal adoption  
59 outside of a few contexts, and fails at the "compact" and "simple to handle" criteria above, on top of the  
60 disadvantages inherent with all NameID usage.

61 Pairwise identification may help meet certain privacy and regulatory requirements (though this is far from  
62 clear to date), but does not address many common use cases that demand cross-system correlation  
63 without the friction of complex linking protocols and the involvement of the data subject.

64 In addition, it has come to light that many, if not most, applications have a predisposition to handle  
65 identifiers case-insensitively, partly due to a long-standing, though factually untrue, assumption that e-mail  
66 address mailbox names are case-insensitive data. SAML's "persistent" NameID definition explicitly  
67 requires case-sensitive handling, making them impossible to use safely with such applications without

---

1 It's worth noting that SAML actually defines a protocol for managing changes to NameID values, but it has seen very little adoption, further demonstrating the lack of value of NameID usage.

68 resorting to additional layers of profiling. Note that any other specification promulgating such identifiers is  
69 potentially unsafe in combination with such applications and should be used with caution.

70 For all of these reasons, this profile attacks these problems by taking a clean-slate approach that  
71 abandons existing practice instead of attempting to layer more profiling and out of band agreements on  
72 top of existing solutions, an approach that has seemingly reached its breaking point.

## 73 2.2 Relationship to Existing Work

74 A clean slate notwithstanding, this profile is based on a thorough review of practice within the higher  
75 education sector, which has seen extensive adoption of SAML and partially-successful efforts to  
76 standardize subject identification and avoid the “email address” trap that most of the technical world fell  
77 into many years ago.

78 Among the significant work in this space, the [\[eduPerson\]](#) schema includes a number of identifier  
79 attributes, some widely adopted and some less so. This profile is particularly influenced by:

- 80 • Experience with the SAML “persistent” NameID construct and the related eduPersonTargetedID  
81 attribute.
- 82 • The eduPersonPrincipalName and eduPersonUniqueid attributes, the former successful but  
83 deeply flawed, the latter less successful but more carefully defined.
- 84 • Success with DNS domain-based scoping of values and managing policy around their use in  
85 SAML.
- 86 • Challenges in the adoption of profiles required to accommodate the limitations of widely deployed  
87 identifiers.

88 Portions of this specification are borrowed liberally from the [\[eduPerson\]](#) specification in a deliberate  
89 desire to remain consistent with the formulation of the eduPersonUniqueid attribute.

90 This specification also incorporates the relevant subset of a SAML Metadata extension schema, originally  
91 defined by the Shibboleth Project [\[ShibMetaExt\]](#). This extension has seen extensive adoption, and is  
92 included here to support centralizing and automating policy for authorizing asserting parties to issue  
93 identifiers in particular scopes. The XML namespace of this extension (a URN issued by the Shibboleth  
94 Project) is maintained to remain compatible with existing implementations and deployments dating back  
95 many years.



---

## 3 SAML V2.0 Subject Identifier Attributes Profile Version 1.0

### 3.1 Required Information

**Identification:** urn:oasis:names:tc:SAML:profiles:subject-id

**Contact information:** security-services-comment@lists.oasis-open.org

**Description:** Given below.

**Updates:** None.

### 3.2 Overview

This profile defines a pair of SAML Attributes providing for unique identification of security subjects (which are generally but not exclusively people). One is designed for general use as a globally-unique identifier, and the other is a pairwise identifier suitable for more specialized uses.

Both SAML Attributes are limited to a single value when expressed in SAML assertions and other constructs. They may be mapped to and from other technical forms (e.g., LDAP attributes) but this profile does not include such mappings.

In the terminology used in this profile:

- "asserting party" refers to a uniquely-named SAML entity that issues assertions containing one or both of these Attributes
- "relying party" refers to one or more uniquely-named SAML entities that receive assertions containing one or both of these Attributes

In addition, this profile defines a signaling mechanism for a relying party to express its subject identification requirements via SAML metadata [[SAML2Meta](#)], by means of the `<mdattr:EntityAttributes>` extension [[MetaAttr](#)]. This allows asserting parties to unambiguously understand the requirements of a peer and facilitates deployment profiles that wish to mandate support for one or both of these Attributes, while maintaining appropriate privacy expectations.

Finally, this profile incorporates and re-publishes in a standards-based context an existing SAML metadata extension element that documents attribute "scopes" an asserting party is authorized to use for its SAML Attributes (according to the issuer of that metadata).

### 3.3 General Purpose Subject Identifier

For general purpose identification of subjects, the following SAML Attribute is defined:

**Name:** urn:oasis:names:tc:SAML:attribute:subject-id

**NameFormat:** urn:oasis:names:tc:SAML:2.0:attrname-format:uri

This is a long-lived, non-reassignable, omni-directional identifier suitable for use as a globally-unique external key. Its value for a given subject is independent of the relying party to whom it is given.

#### 3.3.1 Syntax and Handling

The `<saml:Attribute>` element MUST contain exactly one `<saml:AttributeValue>` element, whose `xsi:type` SHOULD be absent or if present MUST BE bound to the XML Schema `xsd:string` data type [[XMLSCHEMA-2](#)].

Any leading or trailing whitespace, as defined by XML (ASCII 32, ASCII 9, ASCII 10, ASCII 13), present in the `<saml:AttributeValue>` element's content is not significant and MUST be stripped by the relying party prior to evaluation or comparison.

136 The value consists of two substrings (termed a “unique ID” and a “scope” in the remainder of this  
137 definition) separated by an @ symbol (ASCII 64) as an inline delimiter.  
138 The unique ID consists of 1 to 127 ASCII characters, each of which is either an alphanumeric ASCII  
139 character, an equals sign (ASCII 61), or a hyphen (ASCII 45). The first character MUST be alphanumeric.  
140 The scope consists of 1 to 127 ASCII characters, each of which is either an alphanumeric ASCII  
141 character, a hyphen (ASCII 45), or a period (ASCII 46). The first character MUST be alphanumeric. The  
142 scope deliberately resembles, and often is, a DNS domain name, but is drawn from a more limited  
143 character set due to case folding considerations, and no attempt is made to limit the allowable grammar to  
144 legal domain names (e.g., it allows consecutive periods).

145 The ABNF [RFC2234] grammar is therefore:

```
146 <value> = <uniqueID> "@" <scope>  
147 <uniqueID> = (ALPHA / DIGIT) 0*126(ALPHA / DIGIT / "=" / "-")  
148 <scope> = (ALPHA / DIGIT) 0*126(ALPHA / DIGIT / "-" / ".")
```

149 Value comparison MUST be performed case-insensitively (that is, values that differ only by case are the  
150 same, and MUST refer to the same subject).

151 In the grammar above, the ALPHA production contains characters that can be expressed in both upper  
152 and lower case. It is RECOMMENDED that the unique ID be exclusively upper- or lower-case when  
153 expressed or stored to facilitate ease of comparison. Further, it is RECOMMENDED that scopes be  
154 expressed in lower case, since they are generally chosen independently of more “entrenched” decisions  
155 and are frequently, though not required to be, in the form of DNS domains. See also Section 3.5.2.2 for  
156 additional motivation.

### 157 3.3.2 Semantics and Practices

158 A value (the unique ID and scope together) MUST be bound to one and only one subject, but the same  
159 unique ID given a different scope may refer to the same or (far more likely) a different subject.

160 The relationship between an asserting party and a scope is an arbitrary one and does not reflect any  
161 assumed relationship between a scope in the form of a domain name and a domain found in a given  
162 SAML entity identifier. This indirect relationship is formally expressible in SAML metadata via the  
163 extension defined in Section 3.5.2.

164 A value MUST NOT be assigned to more than a single subject over its lifetime of use under any  
165 circumstances. The unique ID should therefore be constructed in a fashion that reduces the probability of  
166 non-technical or political considerations leading to a violation of this requirement, and any such violation  
167 should be treated as a potential security risk to the relying parties to which the value may have been given.

168 Relying parties should not treat this identifier as an email address for the subject as it is unlikely (though  
169 not precluded) for it to be valid for that purpose. Most organizations will find that existing email address  
170 values will not serve well as values for this Attribute.

171 The unique ID should not change as a result of a change to any other data associated with the subject  
172 (e.g., name, email address, age, organizational role).

173 A given value MUST identify the same subject regardless of the context of use or the relying parties to  
174 which the Attribute is given. It is therefore to be assumed by relying parties that receive a given value that  
175 the same subject has been identified.

176 Note that, policy permitting, a given value could be provided by any asserting party, and the requirement  
177 still holds: identical values correspond to the same subject. While it will be common in many deployments  
178 to limit values with a given scope to a single asserting party, this is ultimately left to the discretion of the  
179 relying party and the use case.

180 A single subject MAY be identified simultaneously by a single asserting party by multiple values, but this  
181 should be minimized to the extent possible.

### 182 3.3.3 Example

183 The following is an example of the SAML Attribute defined in this section:

```
184 <saml:Attribute Name="urn:oasis:names:tc:SAML:attribute:subject-id"
```

```
185     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
186     <saml:AttributeValue>idm123456789@example.com</saml:AttributeValue>
187 </saml:Attribute>
```

## 188 3.4 Pairwise Subject Identifier

189 For pairwise identification of subjects, the following SAML Attribute is defined:

190 **Name:** urn:oasis:names:tc:SAML:attribute:pairwise-id

191 **NameFormat:** urn:oasis:names:tc:SAML:2.0:attrname-format:uri

192 This is a long-lived, non-reassignable, uni-directional identifier suitable for use as a unique external key  
193 specific to a particular relying party. Its value for a given subject depends upon the relying party to whom it  
194 is given, thus preventing unrelated systems from using it as a basis for correlation.

### 195 3.4.1 Syntax and Handling

196 The requirements for this Attribute are identical to those described in Section 3.3.1. That is, values of this  
197 Attribute are indistinguishable, lacking the context, from the other.

### 198 3.4.2 Semantics and Practices

199 Given a particular relying party, a value (the unique ID and scope together) **MUST** be bound to only one  
200 subject, but the same unique ID given a different scope may refer to the same or (far more likely) a  
201 different subject. The same value provided to different relying parties **MAY** refer to different subjects, and  
202 indeed that is the primary distinguishing characteristic of this identifier Attribute.

203 The relationship between an asserting party and a scope is an arbitrary one and does not reflect any  
204 assumed relationship between a scope in the form of a domain name and a domain found in a given  
205 SAML entity identifier. This indirect relationship is formally expressible in SAML metadata via the  
206 extension defined in Section 3.5.2.

207 A value **MUST NOT** be assigned to more than a single subject over its lifetime of use under any  
208 circumstances. The unique ID should therefore be constructed in a fashion that reduces the probability of  
209 non-technical or political considerations leading to a violation of this requirement, and any such violation  
210 should be treated as a potential security risk to the relying parties to which the value may have been given.

211 The value **MUST NOT** be mappable by a relying party into a non-pairwise identifier for the subject through  
212 ordinary effort. This precludes the degenerate case of providing a non-pairwise value to all relying parties  
213 for a given subject.

214 Relying parties should not treat this identifier as an email address for the subject as it is unlikely (though  
215 not precluded) for it to be valid for that purpose. Most organizations will find that existing email address  
216 values will not serve well as values for this Attribute.

217 The unique ID should not change as a result of a change to any other data associated with the subject  
218 (e.g., name, email address, age, organizational role).

219 Assuming a particular scope, a given subject **MUST** be identified with a different, though consistent,  
220 unique ID for each relying party to which a value is provided; however, the relationship between relying  
221 parties and SAML entities is not defined by this profile and is interpreted from the perspective of the  
222 asserting party. For example, in the context of the SAML Web Browser SSO profile [[SAMLProf](#)] it would  
223 be typical for an Identity Provider to base its notion of a relying party boundary on a single Service  
224 Provider's entity identifier, but that is not specifically required by this profile. The boundary **MAY** be larger  
225 or even smaller, at the Identity Provider's discretion or as addressed by additional profiles.

226 While it will be common in many deployments to limit values with a given scope to a single asserting party,  
227 this is ultimately left to the discretion of the relying party and the use case. It is unspecified by this profile  
228 whether a given value provided by two or more asserting parties correspond to the same subject. This  
229 would depend on out of band arrangements made between the parties. If you want a relying party to  
230 understand that two or more asserting parties are referring to the same subject, using the general-  
231 purpose subject identifier defined in Section 3.3 is likely to be a much better choice.

### 232 3.4.3 Implementation Strategies

233 Supporting pairwise identifiers typically involves either the generation and storage of random values, or the  
234 computation of reproducible values that can be produced on demand but need not be stored. This profile  
235 does not require any specific approach, but implementers should be aware that some techniques for  
236 computing values may result in an unacceptable risk of case conflicts. For example, a salted hash over a  
237 seed identifier together with a relying party identifier produces a "safe" generated value, but becomes  
238 unsafe when encoded in Base64 [RFC4648] (and the allowable character set is defined in part to preclude  
239 this choice). However, encoding hashes in Base32 [RFC4648] is a safe choice, and the equals sign is  
240 included in the allowable character set to accommodate this.

### 241 3.4.4 Differences from "persistent" NameIDs

242 This Attribute is a direct replacement for the `urn:oasis:names:tc:SAML:2.0:nameid-`  
243 `format:persistent` NameID Format defined in SAML [SAML2Core]. There are obvious syntactic  
244 differences, in a deliberate attempt at simplification. The XML syntax and data "triple" are replaced with a  
245 simpler id/scope pair encoded into a string, and the awkward use of a pair of URIs to qualify the value is  
246 replaced with a simpler, shorter, and more flexible approach that more easily emulates the email address  
247 syntax required by many applications, and decouples identifier scoping from SAML entity naming.

248 One functional gap is the interoperable mechanism of SAML "affiliations" to group entities for the purpose  
249 of targeting pairwise identifiers to multiple Service Providers, which was designed into the SAML protocol.  
250 It has been left out of this profile due to the general lack of adoption by implementers or deployers in the  
251 intervening years since the publication of the standard. Were there demand, it could be incorporated into a  
252 future revision.

### 253 3.4.5 Example

254 The following is an example of the SAML Attribute defined in this section:

```
255 <saml:Attribute Name="urn:oasis:names:tc:SAML:attribute:pairwise-id"  
256     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">  
257   <saml:AttributeValue>  
258 HA2TKNZZGE2TOZDCGMZWKOLDHQBQWIMBSGM4TGZBYGUYGINRQHAYTINBZGYZDOZBZMZRGKNZTME3TMN  
259 BXGYTIOBYGMYWKNLFMYDAYY=@osu.edu  
260   </saml:AttributeValue>  
261 </saml:Attribute>
```

## 262 3.5 Considerations for SAML Profiles

263 The Attributes defined in this profile are designed to be used in conjunction with any SAML profiles that  
264 support the use of SAML Attributes, though its predominant expected use is with the various SAML single  
265 sign-on profiles [SAML2Prof] such as the Web Browser SSO Profile and Enhanced Client or Proxy (ECP)  
266 Profile.

### 267 3.5.1 Requirements Signaling

268 In the event that SAML metadata [SAML2Meta] is used, a relying party MUST express its identifier  
269 requirements by including an `<mdattr:EntityAttribute>` extension [MetaAttr] in its metadata  
270 containing the following Attribute:

271 **Name:** `urn:oasis:names:tc:SAML:profiles:subject-id:req`

272 **NameFormat:** `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

273 This Attribute, MUST contain exactly one `<saml:AttributeValue>` element, whose `xsi:type`  
274 SHOULD be absent or if present MUST BE bound to the XML Schema `xsd:string` data type  
275 [XMLSCHEMA-2].

276 The value MUST be one of the following, signaling the corresponding requirement:

- 277 • `subject-id`
- 278 ◦ The relying party requires the standard identifier Attribute defined in Section 3.3.

- 279 • pairwise-id
- 280 ◦ The relying party requires the pair-wise identifier Attribute defined in Section 3.4.
- 281 • none
- 282 ◦ The relying party does not require any subject identifier and is designed to operate without a
- 283 specific user identity (e.g., with authorization based on non-identifying data).
- 284 • any
- 285 ◦ The relying party will accept any of the identifier Attributes defined in this profile but requires at
- 286 least one.

287 This profile does not define specific normative behavior on the part of asserting parties in response to this  
288 metadata, but it is expected that other profiles will do so in the future.

289 This profile does not provide (nor preclude) any guidance around the use of the  
290 <md:RequestedAttribute> element for signaling requirements, but notably it is impossible without  
291 additional specification work to reflect the semantics of the any value defined above using that  
292 mechanism.

### 293 3.5.2 Scope Filtering

294 A critical obligation of any federated relying party is to limit the ability of asserting parties to supply  
295 identifiers they are not authorized to assert. While this is commonly done in SAML based on the asserting  
296 party's entityID, that approach generally requires artificially combining an identifier's value with the entityID  
297 for storage and comparison. The Attributes defined in this specification include a scope expression in their  
298 values that makes this step unnecessary but introduce the need for a binding between scopes and  
299 asserting parties.

300 In the event that SAML metadata [[SAML2Meta](#)] is used, an asserting party MUST express the scope(s)  
301 within which it will issue subject identifiers by including one or more <shibmd:Scope> elements (defined  
302 below) in its metadata.

303 The <shibmd:Scope> element MUST appear within the <md:Extensions> element of an  
304 <md:EntityDescriptor> element or the <md:Extensions> element of an assertion-issuing role  
305 descriptor element (such as <md:IDPSSODescriptor> or  
306 <md:AttributeAuthorityDescriptor>). The use of the <shibmd:Scope> element outside of  
307 these contexts is undefined.

308 When a <shibmd:Scope> element appears in the <md:Extensions> element of an  
309 <md:EntityDescriptor> element it applies to all descendant role descriptor elements. That is to say,  
310 this usage is equivalent to putting an identical <shibmd:Scope> on every descendant role descriptor.

311 In processing the identifiers defined in this specification, the scope component is intended to be compared  
312 against the collection of scopes designated as permissible for the asserting party in its metadata. Any  
313 values whose scope is not permissible SHOULD be discarded, thus ensuring that all scoped identifier  
314 values accepted by the relying party and passed to an application will have permissible scopes.

315 The final arbiter of any such policy is the relying party, and metadata-based policy via this extension MAY  
316 be supplemented or overridden by local policy.

317 This profile does not mandate a particular exchange or trust model by which the metadata and its content  
318 are expected to be verified, but it is common for metadata containing this extension to come from a  
319 trusted third party able to independently validate an asserting party's right to the claimed scope(s).

320 For compatibility reasons, the matching between values of this extension and the scope component of the  
321 identifiers defined in this specification is done in a case-sensitive manner. To avoid unintentional  
322 mismatches, it is RECOMMENDED that scopes be expressed in lower case (both in this extension and in  
323 the values themselves, per Section 3.3.1).

324 Finally, note that the concept of scope and scope filtering need not be limited to the Attributes defined in  
325 this specification, but such applicability is outside the purview of this specification.



### 326 3.5.2.1 Element <shibmd:Scope>

327 This element extends the **xsd:string** schema type with the following attribute:

328 `regexp` [Optional]  
329 Boolean regular expression indicator

330 Each <shibmd:Scope> element's text content identifies a permissible identifier scope for the issuing  
331 entity/role, per the definition of "scope" in Section 3.3.1.

332 If `regexp` is "false" or "0" or absent, the text content of the <shibmd:Scope> element is interpreted  
333 as the literal scope value (matched case-sensitively for compatibility reasons, see below).

334 If `regexp` is "true" or "1", the text content of the <shibmd:Scope> element is interpreted as  
335 specifying a regular expression (also see below).

336 The schema for the <shibmd:Scope> element is as follows:

```
337 <element name="Scope">  
338   <complexType>  
339     <simpleContent>  
340       <extension base="string">  
341         <attribute name="regexp" type="boolean" use="optional"  
342         default="false"/>  
343       </extension>  
344     </simpleContent>  
345   </complexType>  
346 </element>
```

### 347 3.5.2.2 Usage Considerations

348 Because this extension has an extensive history of use, its definition is not optimal and there are some  
349 important caveats.

350 Comparison of literal scope values expressed via this extension is defined to be case-sensitive, despite  
351 the overall rule for comparison of the Attributes defined in this specification as case-insensitive. This is for  
352 reasons of historical compatibility and generality, and is easily addressed by adhering to this  
353 specification's guidance to express scopes in lower-case.

354 The XML Schema definition of the <shibmd:Scope> element includes an explicit default value for the  
355 `regexp` attribute. One effect of this is that the meaning of an omitted `regexp` attribute will be different for  
356 a schema-validating processor than for one which does not schema-validate. If a document containing a  
357 <shibmd:Scope> element with an omitted `regexp` attribute is digitally signed, the signature value will  
358 therefore depend on whether the signer schema-validates, and validation of such a signature will only  
359 succeed if the validator has chosen to take the same approach.

360 To ensure interoperability between signers and validators no matter whether each schema validates or  
361 does not, it is therefore strongly RECOMMENDED that any <shibmd:Scope> element appearing in a  
362 metadata document that is to be digitally signed incorporate an explicit `regexp` attribute (i.e.,  
363 `regexp="false"` or `regexp="0"` SHOULD always be used instead of an omitted `regexp` attribute).

364 Furthermore, great care should be taken in using `regexp="true"` as it is extremely easy to write regular  
365 expressions which match the desired patterns but also permit additional, sometimes surprising, matches.  
366 This can lead to an asserting party being permitted a wider range of scopes than intended. Common  
367 mistakes are not appropriately quoting meta-characters such as ". ", and not appropriately anchoring the  
368 ends of the match.

369 Additionally, regular expressions are implemented with a degree of inconsistency in specifics and features  
370 and this extension does not include a formal reference to any single "standard" version of regular  
371 expressions because it would be impractical to force SAML implementations to follow only one.

372 As a result, deployments SHOULD avoid the use of regular expressions and implementations MAY omit  
373 support for this capability and reject its use. Its presence is again an issue of legacy compatibility more-so  
374 than current practice.

### 375 3.5.3 NameID Considerations

376 While the Attributes defined in this profile have as a goal the explicit replacement of the `<saml:NameID>`  
377 element as a means of subject identification, it is certainly possible to compose them with existing  
378 NameID usage provided the same subject is being identified. This can also serve as a migration strategy  
379 for existing applications.

380 Some profiles such as the Single Logout Profile [[SAML2Prof](#)] require the use of a `<saml:NameID>`  
381 element, which implies the earlier use of a NameID. In such cases, it is RECOMMENDED that the  
382 `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` NameID Format be used.

383 This specification does not define any syntax by which the SAML Attributes defined within would be used  
384 directly within the NameID construct. Such use is discouraged, but is not precluded by this specification. In  
385 practice, the most appropriate mechanism to express any string-valued SAML Attribute in a  
386 `<saml:NameID>` element is to express the Attribute's Name as a `Format` and omit any qualifiers, and  
387 such an approach is safe to use with the Attributes defined in this specification.

### 388 3.5.4 Security Considerations

389 All identifiers have inherent and generally well-understood concerns; most applications traditionally  
390 associate users directly with resources, privileges, and/or data by uniquely identifying those users and  
391 remembering them during subsequent interactions. Federated protocols don't alter these concerns, but  
392 can complicate them because of the particular issues introduced by multiple asserting parties that may  
393 (but usually do not) share a common identifier namespace.

394 Applications not originally designed to support federation often treat each asserting party as a kind of silo  
395 of identity, and the identifiers used are inherently segregated by these silos such that global uniqueness  
396 (or lack thereof) is irrelevant. In such cases, the asserting party's own identifier acts as an implicit "scope"  
397 for all of the identifiers it asserts. In some cases, a lack of this implicit enforcement of scope has led to  
398 security vulnerabilities involving impersonation of users across asserting parties, demonstrating that, no  
399 matter what kind of identifier is used, some form of scoping of user identifiers is an absolute necessity in  
400 federated systems. This requirement is more obvious when applications are truly federated and combine  
401 identifiers from multiple asserting parties within a data set.

402 The identifier attributes defined in this specification contain an explicit scope as part of their syntax,  
403 providing globally uniqueness, but, more subtly, creating indirection between the scopes and the asserting  
404 party or parties that provide them. That is, the scope is explicit, but the relationship between that scope  
405 and an asserting party is indirect, at least when looking solely at the identifier. This indirection adds power,  
406 in that use cases involving identity linking between asserting parties become simpler to support, and it  
407 adds simplicity from the point of view of safe handling of identifier values since the scope is harder to  
408 "lose" or ignore. But this also adds complexity because a policy decision is required to authorize an  
409 asserting party to supply identifiers in a given scope.

410 As an example, consider an identifier such as "abcdef123@osu.edu"; SAML doesn't define anything in its  
411 core machinery that associates "osu.edu" with the Identity Provider representing The Ohio State  
412 University. Domain ownership proofs are of course a common and sensible practice to use to establish  
413 this association, but nothing in SAML specifies that, so it's an additional step and is not represented "in-  
414 band".

415 This specification does not impose a single such policy layer, but does standardize (in Section 3.5.2) a  
416 long-standing SAML metadata extension that associates authorized scope values with asserting parties.  
417 By using SAML metadata, the problem of self-assertion is addressed; if an asserting party were able to  
418 self-authorize its ability to supply an identifier in a different asserting party's scope, impersonation  
419 becomes easy. Communities that rely on curated, third-party sources of metadata have a vehicle for  
420 automating policy around scopes, and for off-loading domain/scope verification. Thus, use of metadata in  
421 this fashion and use of scoped identifiers become mutually reinforcing.

---

## 422 4 Conformance

### 423 4.1 Conformance Clause 1: Asserting Party Implementations

424 An asserting party implementation conforms to this specification if it can be configured to produce both  
425 identifier Attributes conforming to the normative requirements in Sections 3.3 and 3.4.

426 If the asserting party implementation provides a mechanism for generation and/or publication of SAML  
427 metadata, then it MUST support the inclusion of the extension defined in Section 3.5.2.

### 428 4.2 Conformance Clause 2: Relying Party Implementations

429 A relying party implementation conforms to this specification if it can be configured to consume neither,  
430 either, and both of the two identifier Attributes conforming to the normative requirements in Sections 3.3  
431 and 3.4.

432 If the relying party implementation provides a mechanism for generation and/or publication of SAML  
433 metadata, then it MUST support the inclusion of the extension defined in Section 3.5.1.

434 If the relying party supports the consumption of SAML metadata, then it MUST support configuring its  
435 acceptance of values of the Attributes defined in this specification based on authorization of their scopes  
436 via the extension defined in Section 3.5.2.



---

## 437 **Appendix A Acknowledgments**

438 The following individuals have participated in the creation of this specification and are gratefully acknowl-  
439 edged:

Scott Cantor, Internet2  
Thomas Hardjono, MIT  
Mohammad Jafari, Veterans Health Administration  
Hal Lockhart, Oracle Corporation  
Madalina Sultan, Connectis

Contributors to the InCommon Deployment Profile Working Group  
Past contributors to the Shibboleth Project

---

440 **Appendix B Revision History**

<b>Revision</b>	<b>Date</b>	<b>Editor</b>	<b>Changes Made</b>
WD 01	30 Aug 2017	Scott Cantor	Initial draft
WD 02	13 Sep 2017	Scott Cantor	Added considerations for other profiles
WD 03	15 Sep 2017	Scott Cantor	Added hyphen as legal character in unique ID
WD 04	1 Feb 2018	Scott Cantor	Many nits, missing references, clarifying changes in response to public review
WD 05	3 Jul 2018	Scott Cantor	Second public review updates
WD 06	5 Sep 2018	Scott Cantor	Expansion of scope to include, umm, Scope
WD 07	16 Nov 2018	Scott Cantor	Editorial nits and corrections for final vote

441