



Basic Metadata Aggregation Profile

Version 0.3

[Stage]

August 16th, 2010

Specification URIs:

This Version:

[http://docs.oasis-open.org/sstc/\[additional path/filename\].html](http://docs.oasis-open.org/sstc/[additional path/filename].html)
[http://docs.oasis-open.org/sstc/\[additional path/filename\].doc](http://docs.oasis-open.org/sstc/[additional path/filename].doc)
[http://docs.oasis-open.org/sstc/\[additional path/filename\].pdf](http://docs.oasis-open.org/sstc/[additional path/filename].pdf)

Previous Version:

[http://docs.oasis-open.org/sstc/\[additional path/filename\].html](http://docs.oasis-open.org/sstc/[additional path/filename].html)
[http://docs.oasis-open.org/sstc/\[additional path/filename\].doc](http://docs.oasis-open.org/sstc/[additional path/filename].doc)
[http://docs.oasis-open.org/sstc/\[additional path/filename\].pdf](http://docs.oasis-open.org/sstc/[additional path/filename].pdf)

Latest Version:

[http://docs.oasis-open.org/sstc/\[additional path/filename\].html](http://docs.oasis-open.org/sstc/[additional path/filename].html)
[http://docs.oasis-open.org/sstc/\[additional path/filename\].doc](http://docs.oasis-open.org/sstc/[additional path/filename].doc)
[http://docs.oasis-open.org/sstc/\[additional path/filename\].pdf](http://docs.oasis-open.org/sstc/[additional path/filename].pdf)

Technical Committee:

OASIS [official name of technical committee] TC

Chair(s):

[Chair name]
[Chair name]

Editor(s):

Andreas Åkre Solberg, UNINETT AS, andreas.solberg@uninett.no

Related work:

This specification is related to:

- Metadata for the OASIS Security Assertion Markup Language (SAML)

Abstract:

This profile defines a basic set of processing rules when aggregating SAML Metadata from multiple sources. The profile aims at being extensible.

Status:

This document was last revised or approved by the [TC name | membership of OASIS] 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/sstc/>.

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/sstc/ipr.php>).

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

Notices

Copyright © OASIS® 2008. All Rights Reserved.

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

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

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

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

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

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

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary

rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", [insert specific trademarked names and abbreviations here] 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	7
1.1	Terminology	7
1.2	Normative References	7
1.3	Non-Normative References	7
2	Aggregator Configuration Statement	8
3	Registration of Input Sources and Trust Management	10
4	Processing	11
4.1	Fetching metadata from input sources	11
4.2	Processing each entity	12
4.2.1	Entity Acceptance Processing	12
4.2.2	Removing information	12
4.2.3	Calculating entity expiration time	12
4.3	Processing Extensions	13
4.4	Processing the RoleDescriptors of an Entity	13
4.5	Cache Entity	13
4.6	Preparing the output document	14
5	Extending this profile	15
#	Conformance	16
A.	Acknowledgements	17
B.	Example Aggregator Configuration Statement	18
C.	Aggregator Extension for Documentation and Registration Information	19
C.1	Processing Metadata Document	19
C.2	Entity Acceptance Processing	19
C.3	Processing Extensions	19
C.4	Processing RoleDescriptors	19
D.	Aggregator Extension for Entity Attributes	20
D.1	Processing Metadata Document	20
D.2	Entity Acceptance Processing	20
D.3	Processing Extensions	20
D.4	Processing RoleDescriptors	20
E.	eduGAIN Aggregator Profile	21
E.1	Processing Metadata Document	21
E.2	Entity Acceptance Processing	21
E.3	Processing Extensions	21
E.4	Processing RoleDescriptors	21
F.	Revision History	22

1 Introduction

This profile describes the processing rules of a Metadata Aggregator; a component that fetches *SAML V.2.0 Metadata* [SAML2Meta] from multiple input sources over HTTP, merges the metadata and publishes the result set of metadata.

The processing rules of aggregating metadata, is split into these sections:

1. Fetching metadata documents
2. Processing metadata documents
3. Processing each entity
4. Preparing output document

1.1 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].

1.2 Normative References

- [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.
- [SAML2Meta] OASIS, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*
<http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>
- [MDAttrib] OASIS, *SAML V2.0 Metadata Extension for Entity Attributes Version 1.0*
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.html>
- [MetaReg] OASIS, *SAML Metadata Documentation and Registration Information Extensions, version 0.2*
https://spaces.internet2.edu/download/attachments/9731/saml_md_dri_02.odt
- [MDX] *IETF Internet-Draft, Metadata Query Protocol*
<http://tools.ietf.org/pdf/draft-lajoie-md-query-00.pdf>

1.3 Non-Normative References

- [Reference] [Full reference citation]

34

35 2 Aggregator Configuration Statement

36 A deployment of an aggregator, MUST in order to satisfy this profile make available an
37 *aggregator configuration statement* document to all consumers of the output metadata.

38 The configuration statement MUST include at least the following information about this
39 deployment:

40 Contact information

41 Contact information about the operators of the deployed aggregator.

42 Supported Aggregation Profiles

43 This profile, as well as any number of complementary profiles.

44 Metadata access protocols

45 List of supported metadata access protocols, and corresponding HTTP
46 endpoints.

47 Trust root

48 Public key or certificate of the aggregator or a corresponding trust root (CA),
49 used for signing the output document(s).

50 Aggregator Fetch Interval

51 The maximum interval at which the aggregator pulls metadata from each Input
52 Source.

53 Minimum Required Validity of Input Entities

54 The shortest validity period the input documents are required to have. The
55 validity period refers to a cached version of the entity, which means that the
56 `cacheDuration` MUST not exceed this value either. This is a soft limit, which
57 means entities will not be rejected because of this, but the aggregator may only
58 fulfill its validity requirements if this limit is respected by all entities on all input
59 sources. May be set to "NA", if not applicable.

60 Minimum Validity of Output Entities

61 The minimum validity of an entity outputted by the aggregator, given that all input
62 sources respect the "*Minimum Required Validity of Input Entities*". If the option
63 parameter above is set to "NA", this MUST as well.

64 Maximum Signing Interval

65 The aggregator MAY choose to sign less often than the "*Aggregator Fetch*
66 *Interval*", in cases where there is no changes in the metadata. If so, the
67 aggregator MAY use this parameter to state the maximum duration between
68 each time metadata is signed. May be set to "NA", if not applicable.

69 Default Extension Processing Policy

70 See section 4.3 for more details.

71 Default RoleDescriptor Processing Policy

72 See section 4.4 for more details.

73 The “*Minimum Validity of Output Entities*” may typically be calculated by taking the
74 “*Minimum Required Validity of Input Entities*” and subtract the “*Maximum Signing*
75 *Interval*”.

76 Complementary profiles MAY define additional information that is required to be stated
77 in the aggregator configuration statement.

78 **3 Registration of Input Sources and Trust**
79 **Management**

80 For each input source, from where the aggregator fetches metadata, the aggregator
81 MUST register at least the following information associated with each input source:

- 82 • PublisherID (adopting terminology from [MetaReg])
- 83 • A list of one or more equivalent HTTP endpoints, from where a full metadata
84 document can be retrieved. Multiple endpoints MAY be provided for improved
85 reliability. The list is not prioritized, and all endpoints are equivalent.
- 86 • For each endpoint as mentioned above, one or more public key / certificate
87 MUST be provided. This public key / certificate represents the signature on the
88 top-level element of the published metadata document. Multiple public keys /
89 certificates MAY be provided in order for a smooth rollover.

90 4 Processing

91 4.1 Fetching metadata from input sources

92 The aggregator fetches metadata from each input source at a configured interval. The
93 retrieved metadata document will be processed for some basic verification, described in
94 this section. If any of these verification steps fails, further processing **MUST** be stopped,
95 and aggregator will move on to processing the next input source. The aggregator
96 **SHOULD** implement re-tries depending on the nature of the failure; in example the
97 aggregator may wait and retry downloading the document if a network error occurred.

98 The aggregator **MUST** support HTTP-redirection (HTTP 1.1 302 with Location header).

99 The input source **MUST** present the metadata document including a `Content-Type`
100 HTTP header with the value of `application/samlmetadata+xml`. The aggregator **MAY**
101 reject the document due to invalid `Content-Type`.

102 In the case of a network or access problem, the aggregator **SHOULD** iterate through the
103 list of endpoints, if multiple are configured, as described in section 3.

104 The aggregator **MUST** perform the following verification steps:

- 105 1. The provided document is well formed XML.
- 106 2. The root element is an `md:EntitiesDescriptor` or `md:EntityDescriptor`.
- 107 3. The document **MUST** be signed at the root element. The signing key / certificate
108 **MUST** correspond to the configured key / certificate / CA for this particular input
109 source, as described in section 3.
- 110 4. The metadata document **MUST** be schema-valid, validating the following XML
111 namespaces:
 - 112 • `urn:oasis:names:tc:SAML:2.0:metadata`
 - 113 • `urn:oasis:names:tc:SAML:2.0:assertion`
 - 114 • `http://www.w3.org/2000/09/xmldsig#`
 - 115 • `http://www.w3.org/2001/04/xmlenc#`
- 116 5. Complementary profiles referred to in the *Aggregator Configuration Statement*
117 may add further verification steps under the header "Processing Metadata
118 Document", including in example additional XML namespaces to validate.

119 **NOTE:** `RoleDescriptors` with a namespace unknown to the aggregator,
120 not mentioned in this profile or any supported complementary profiles,
121 **SHOULD** be treated as if the element was not there in the validation step.
122 Without access to the schemas, the aggregator will not know if the
123 element is inheriting from the abstract `RoleDescriptor` element or not, and
124 **SHOULD** consider it as it was valid.

125 Failure up to or at this point **MUST NOT** invalidate previously cached entities from the
126 same input source.

127 If verification succeeded, the aggregator **MUST** remove previously cached entities,
128 originating from the same input source, but is not present in the new metadata
129 document.

130 The metadata document MAY include any level of nested `md:EntitiesDescriptor`
131 elements. No semantic meaning should be associated with the hierarchy in how
132 `md:EntityDescriptor` elements are located, beyond what is defined in [SAMLMeta].
133 Further processing will be performed for each `md:EntitiesDescriptor`.

134 4.2 Processing each entity

135 The aggregator will process each entity individually as specified in the following
136 sections.

137 Important: content that is not associated with any explicit process instruction in this
138 section, the aggregator MUST pass through unmodified.

139 4.2.1 Entity Acceptance Processing

140 An entity rejected at this point MUST NOT invalidate an already cached version of the
141 same entity.

142 An aggregator MUST provide a mechanism to protect Entities from being hijacked by
143 another input source. This section describes the default processing rules protecting
144 against Entity hijacking. A complementary profile MAY provide an alternative measures
145 of protection, such as a controlled prefixed entityID namespace.

146 The aggregator SHOULD store an association table mapping an EntityID with the
147 PublisherID of the input source where it was first discovered. This association SHOULD
148 be stored for at least the validity period of the contained metadata.

149 If the aggregator receives an `md:EntityDescriptor` with an EntityID registered at
150 another Input source, the aggregator MUST reject this `md:EntityDescriptor`.

151 A complementary profile MAY provide additional entity acceptance processing rules
152 under a section named "Entity Acceptance Processing".

153 4.2.2 Removing information

154 If the `md:EntityDescription` element is signed, this signature MUST be removed. The
155 reason is that it will not be valid, due to the modifications necessary in the following
156 processing.

157 The aggregator MUST ensure that there will be no collisions in the XML ID attributes of
158 all of its entities. The RECOMMENDED approach is to remove the XML ID attributes at
159 the `md:EntityDescription`.

160 4.2.3 Calculating entity expiration time

161 The expiration time of an entity is calculated to be the first of these times:

- 162 • The `validUntil` attribute of the `md:EntityDescriptor` element
- 163 • The `validUntil` attribute of each of the surrounding `md:EntitiesDescriptor`
164 elements
- 165 • The maximum validity period configured for the aggregator
- 166 • The `cacheDuration` attribute of the `md:EntityDescriptor` element + time the
167 metadata was fetched

168 • The `cacheDuration` attribute of each of the surrounding `md:EntitiesDescriptor`
169 elements + time the metadata was fetched

170 The `validUntil` attribute of the `md:EntityDescriptor` MUST be set to the new
171 calculated expiration time (replace the current attribute value if present).

172 The aggregator MAY remove the `cacheDuration` attribute of the `md:EntityDescriptor` or
173 leave it unmodified.

174 Note: expiration information on `RoleDescriptor` elements or other elements inside an
175 `md:EntityDescriptor` SHOULD not be considered by the aggregator but passed
176 through.

177 **4.3 Processing Extensions**

178 The Basic Aggregator Profile does not define any special processing of any particular
179 Extension namespace, but any complementary profile MAY define a set of Extension
180 namespaces associated with processing rules under the section header “Processing
181 Extensions”.

182 The aggregator MUST express a default extension namespace processing rule in the
183 aggregator configuration statement. It may select one of the following default policies:

184 **PASSTHRU**

185 Extension is passed through without further processing.

186 **REMOVE**

187 Extension is removed from metadata.

188 **4.4 Processing the RoleDescriptors of an Entity**

189 The Basic Aggregator Profile does not define any special processing of any particular
190 `RoleDescriptor`, but any complementary profile MAY define a set of namespaced
191 `RoleDescriptor` elements associated with processing rules under the section header
192 “Processing the `RoleDescriptor` of an Entity”.

193 The aggregator SHOULD enforce the PASSTHRU policy for all the `RoleDescriptor`,
194 unless explicitly overridden in a complementary profile.

195 The aggregator MUST express a default `RoleDescriptor` processing policy in its
196 *Aggregator Configuration Statement*. It may select one of the following default policies:

197 **PASSTHRU**

198 `RoleDescriptor` is passed through without further processing.

199 **REMOVE**

200 `RoleDescriptor` is removed from metadata.

201 **4.5 Cache Entity**

202 After Entity is processed and accepted, the processed Entity MUST be cached by the
203 aggregator for the validity period of the entity.

204 Each cached Entity instance MUST be independent of other Entities, even those
205 originating from the same input source.

206 **4.6 Preparing the output document**

207 The aggregator **MUST** examine the expiration time of each entity, and if an entity is
208 expired, the aggregator **MUST** not include the entity in the output.

209 It is **RECOMMENDED** that the aggregator adds non-destructive line-breaks and
210 indentation of easier human inspection – before signing.

211 The aggregator **SHOULD** perform non-destructive XML namespace declaration
212 optimization.

213 The aggregator **SHOULD** prepare an output metadata document including all entities in
214 a flat structure (only one level of `md:EntitiesDescriptor`, the root note). If the
215 aggregator provides the whole set of entities in a document, the document **MUST** be
216 provided over HTTP and the document **MUST** be signed at the root note with the public
217 key / certificate indicated in the configuration statement. The aggregator **MUST** provide
218 a HTTP header indicating that the `Content-Type` is `application/samlmetadata+xml`.

219 The aggregator **MUST** be able to sign the output document after each time any of the
220 input entities are changed (even if only the `validUntil` attribute is updated). If the input
221 data is not modified, the aggregator may choose to wait with re-signing the output
222 document. The maximum interval between signing **MAY** be upper limited to the
223 “*Maximum Signing Interval*”, if specified in the *Aggregator Configuration Statement*.

224 This profile does not dictate any particular access protocol for the output metadata, but
225 **RECOMMENDS** taking the [MDX] protocol into consideration.

226 **5 Extending this profile**

227 A profile that extends the basic aggregator profile **MUST** have a name, and an
228 introduction that explains the purpose of the profile. Next it **MAY** include any of these
229 sections:

- 230 6. Aggregator configuration statement
- 231 7. Processing Metadata Document
- 232 8. Entity Acceptance Processing
- 233 9. Processing Extensions
- 234 10. Processing the RoleDescriptor of an Entity

235 Description of processing rules in these sections are handled as described in the
236 appropriate location in this profile.

237 **# Conformance**

238 The last numbered section in the specification must be the Conformance section.

239 Conformance Statements/Clauses go here.

240 **A. Acknowledgements**

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

243 **Participants:**

244 Leif Johansson, NorduNet
245 Lukas Hämmerle, SWITCH
246 Thomas Legenhagger, SWITCH
247 Mikael Linden, SCS
248 Brook Schofield, Terena
249 Miroslav Milinovic, SRCE
250 Dubravko Penezic, SRCE
251 Dubravko Voncina, SRCE
252 Mehdi Hached, RENATER

253

254 B. Example Aggregator Configuration Statement

255 Contact information

256 Dante Inc., support@dante.org

257 Supported Profiles

- 258 • Basic Aggregation Profile
- 259 • eduGAIN Aggregator Profile
- 260 • Aggregator Extension for Documentation and Registration Information
- 261 • Aggregator Extension for Entity Attributes

262 Metadata Access Protocols

- 263 • Basic HTTP access:
264 <http://aggregator.edugain.edu/all.xml>
- 265 • MDX:
266 <http://aggregator.edugain.edu/mdx/>

267 Signing certificate

```
268 -----BEGIN CERTIFICATE-----  
269 MIICgTCCAeoCCQCbOlrWdX7FTANBgkqhkiG9w0BAQUFADCbDELMAkGA1UEBhMC  
270 Tk8xGDAWBgNVBAgTD0FuZHZJlYXMGU29sYmVyZzEMMAoGA1UEBxMDRm9vMRAwDgYD  
271 [snipp]  
272 BUoWQ9+VMQx1ASDohBjwOsg2WykUqRXF+dLfcUH9dWR63CtZIKFDbStNomPnQz7n  
273 bK+onygwBspVEbnHuUihZq3ZUdmumQqCw4Uvs/1Uvq3orOo/WJVhTyvLgFVK2Qar  
274 Q4/67OzfHd7R+POBXhophSMv1ZOo  
275 -----END CERTIFICATE-----
```

276 Aggregator fetch interval

277 1 hour

278 Maximum metadata validity period

279 14 days

280 Minimum Required Validity of Input Entities

281 4 days

282 Minimum Validity of Output Entities

283 72 hours

284 Maximum Signing Interval

285 24 hours

286 Default Extension Processing Policy

287 REMOVE

288 Default RoleDescriptor Processing Policy

289 REMOVE

290 C. Aggregator Extension for Documentation and 291 Registration Information

292 This profile extends the Basic Aggregator Profile. It assumes that both the input sources
293 and the aggregator support the SAML Metadata Documentation and Registration
294 Information Extensions [MDReg].

295 C.1 Processing Metadata Document

296 When validating the document, the aggregator MUST in addition consider the following
297 additional namespace:

- 298 • `urn:oasis:names:tc:SAML:2.0:metadata:dri`

299 All `md:EntitiesDescriptor` elements containing an `md:Extension` with a
300 `dri:DocumentInfo` or a `dri:RegistrationInfo` MUST apply the `dri:DocumentInfo` and
301 `dri:RegistrationInfo` elements to all containing `md:EntityDescriptor`, to be
302 processed in the “Entity Acceptance Processing” step.

303 C.2 Entity Acceptance Processing

304 If multiple `dri:DocumentInfo` or `dri:RegistrationInfo` elements are associated with the
305 entity, values from the inner elements MUST take precedence.

306 If a cached version of the Entity contains a `dri:DocumentInfo/SerialNumber` the
307 aggregator MUST not accept a new version of the Entity unless the new Entity contains
308 a `dri:DocumentInfo/SerialNumber` being the same or a higher integer value.

309 If a cached version of the Entity contains a `dri:DocumentInfo/CreationInstant` the
310 aggregator MUST not accept a new version of the Entity unless the new Entity contains
311 a `dri:DocumentInfo/CreationInstant` being the same or a more recent timestamp.

312 C.3 Processing Extensions

313 If multiple `dri:DocumentInfo` or `dri:RegistrationInfo` elements are associated with
314 the entity, values from the inner elements MUST take precedence. The aggregator
315 MUST merge multiple `dri:DocumentInfo` and `dri:RegistrationInfo` elements into one
316 of each. Each value (sub element with content) MUST be considered independent from
317 the others.

318 C.4 Processing RoleDescriptors

319 NA

320 D. Aggregator Extension for Entity Attributes

321 This profile extends the Basic Aggregator Profile adding support and processing rules
322 for SAML V2.0 Metadata Extension for Entity Attributes [MDAttrib].

323 D.1 Processing Metadata Document

324 When validation the document, the aggregator MUST in addition consider the following
325 additional namespace:

- 326 • `urn:oasis:names:tc:SAML:metadata:attribute`

327 Extension elements in this namespace, associated with any `md:EntitiesDescriptor`
328 MUST be ignored.

329 D.2 Entity Acceptance Processing

330 NA

331 D.3 Processing Extensions

332 If an Entity contains an `mdattr:EntityAttribute` this element SHOULD be sent through
333 the aggregator unmodified.

334 No further processing of the `mdattr:EntityAttribute` content is defined in this profile
335 extension.

336 D.4 Processing RoleDescriptors

337 NA

338 **E. eduGAIN Aggregator Profile**

339 This profile extends the Basic Aggregator Profile.

340 NOTE: This appendix is only temporarily part of this document. Will
341 eventually be made into a separate document. It is convenient to have this
342 inside, until the basic profile is somewhat more mature.

343 **E.1 Processing Metadata Document**

344 The metadata document SHOULD be schema valid against the XML namespaces in the
345 Basic Aggregator Profile, but in addition the following XML namespaces:

346 ...

347 **E.2 Entity Acceptance Processing**

348 The entity MUST include the required information as specified in eduGAIN Metadata
349 Profile.

350 **E.3 Processing Extensions**

351 ...

352 **E.4 Processing RoleDescriptors**

353 The aggregator MUST REJECT all RoleDescriptors except IdPSSODescriptor and
354 SPSSODescriptor.

355 **F. Revision History**

356

Revision	Date	Editor	Changes Made
0.2	August 9 th 2010	Andreas Åkre Solberg	Incorporating comments from Thomas Lenggenhager and Lukas Hämmerle
0.1	July 17 th 2010	Andreas Åkre Solberg	Initial draft

357

358