



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

XAdES Profiles of the OASIS Digital Signature Service

Committee Draft, 24 December 2004 (Working Draft 06)

Document identifier:

oasis-dss-1.0-profiles-XAdES-spec-cd-01

Location:

<http://docs.oasis-open.org/dss/>

Editor:

Juan Carlos Cruellas, *individual* <cruellas@ac.upc.es>

Contributors:

Nick Pope, *individual* <pope@secstan.com>

Ed Shallow, *Universal Post Union* <ed.shallow@rogers.com>

Trevor Perrin, *individual*

Abstract:

This draft defines one abstract profile of the OASIS DSS protocols for the purpose of creating and verifying XML or CMS based Advanced Electronic Signatures. It also defines two concrete sub-profiles: one for creating and verifying XML Advanced Electronic Signatures and the other for creating and verifying CMS based Advanced Electronic Signatures.

Status:

This is a **Committee Draft** produced by the OASIS Digital Signature Service Technical Committee. Committee members should send comments on this draft to dss@lists.oasis-open.org.

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 Digital Signature Service TC web page at <http://www.oasis-open.org/committees/dss/ipr.php>.

Table of Contents

31	1	Introduction	5
32	1.1	Notation	5
33	1.2	Namespaces	5
34	2	Overview	6
35	3	Advanced Electronic Signature abstract profile	7
36	3.1	Overview	7
37	3.2	Profile Features	8
38	3.2.1	Scope	8
39	3.2.2	Relationship To Other Profiles	8
40	3.2.3	Signature Object	8
41	3.3	Profile of Signing Protocol	8
42	3.3.1	Element <SignRequest>	9
43	3.3.1.1	Element <OptionalInputs>	9
44	3.3.1.1.1	Element <SignatureType>	9
45	3.3.1.1.2	Element <SignatureForm>	9
46	3.3.1.1.3	Optional inputs < ClaimedIdentity> / <KeySelector>	9
47	3.3.1.1.4	Element <AddTimeStamp>	9
48	3.3.1.1.5	Element <SignedProperties>	10
49	3.3.1.1.5.1	Requesting SigningTime	10
50	3.3.1.1.5.2	Requesting CommitmentTypeIndication	10
51	3.3.1.1.5.3	Requesting SignatureProductionPlace	10
52	3.3.1.1.5.4	Requesting SignerRole	10
53	3.3.1.1.5.5	Requesting <xades:IndividualDataObjectTimeStamp>	11
54	3.3.1.1.5.6	Requesting data objects format	11
55	3.3.2	Element <SignResponse>	12
56	3.3.2.1	Element <Result>	12
57	3.3.2.2	Element <SignatureObject>	12
58	3.3.2.3	Element <DocumentWithSignature>	12
59	3.4	Profile of Verifying Protocol	12
60	3.4.1	Element <VerifyRequest>	12
61	3.4.1.1	Attribute Profile	13
62	3.4.1.2	Element <OptionalInputs>	13
63	3.4.1.2.1	Element <ReturnUpdatedSignature>	13
64	4	Element <VerifyResponse>	14
65	4.1.1.1	Element <OptionalOutputs>	14
66	4.1.1.1.1	Element <UpdatedSignature>	14
67	5	XML Advanced Electronic Signatures concrete Profile	15
68	5.1	Overview	15
69	5.2	Profile features	15
70	5.2.1	Identifier	15
71	5.2.2	Scope	15
72	5.2.3	Relationship To Other Profiles	16
73	5.2.4	Signature Object	16
74	5.2.5	Transport Binding	16

75	5.2.6	Security Binding	16
76	5.3	Profile of Signing Protocol	16
77	5.3.1	Element <SignRequest>	16
78	5.3.1.1	Attribute Profile	16
79	5.3.1.2	Element <OptionalInputs>	16
80	5.3.1.2.1	Element <SignatureType>	16
81	5.3.1.2.2	Element <SignatureForm>	17
82	5.3.1.2.3	Optional inputs < ClaimedIdentity> / <KeySelector>	17
83	5.3.1.2.4	Element <AddTimeStamp>	17
84	5.3.1.2.5	Element <SignedProperties>	17
85	5.3.1.2.5.1	Requesting SigningTime	17
86	5.3.1.2.5.2	Requesting CommitmentTypeIndication	17
87	5.3.1.2.5.3	Requesting SignatureProductionPlace	17
88	5.3.1.2.5.4	Requesting SignerRole	17
89	5.3.1.2.5.5	Requesting <xades:IndividualDataObjectTimeStamp>	17
90	5.3.1.2.5.6	Requesting data objects format	17
91	5.3.2	Element <SignResponse>	17
92	5.3.2.1	Element <Result>	17
93	5.3.2.2	Element <SignatureObject>	18
94	5.4	Profile of Verifying Protocol	18
95	5.4.1	Element <VerifyRequest>	18
96	5.4.1.1	Attribute Profile	18
97	5.4.1.2	Element <OptionalInputs>	18
98	5.4.1.2.1	Element <ReturnUpdatedSignature>	18
99	5.4.1.3	Element <SignatureObject>	18
100	5.4.2	Element <VerifyResponse>	18
101	5.4.2.1	Attribute Profile	Error! Bookmark not defined.
102	5.4.2.2	Element <OptionalOutputs>	18
103	5.4.2.2.1	Element <UpdatedSignature>	18
104	5.5	Profile Bindings	18
105	5.5.1	Transport Bindings	18
106	5.5.2	Security Bindings	19
107	5.5.2.1	Security Requirements	19
108	5.5.2.2	TLS X.509 Mutual Authentication	19
109	6	CMS-based Advanced Electronic Signature profile	20
110	6.1	Overview	20
111	6.2	Profile features	20
112	6.2.1	Identifier	20
113	6.2.2	Scope	21
114	6.2.3	Relationship To Other Profiles	21
115	6.2.4	Signature Object	21
116	6.2.5	Transport Binding	21
117	6.2.6	Security Binding	21
118	6.3	Profile of Signing Protocol	21
119	6.3.1	Element <SignRequest>	21
120	6.3.1.1	Attribute Profile	21
121	6.3.1.2	Element <OptionalInputs>	21
122	6.3.1.2.1	Element <SignatureType>	21

123	6.3.1.2.2	Element <SignatureForm>.....	22
124	6.3.1.2.3	Optional inputs < ClaimedIdentity> / <KeySelector>.....	22
125	6.3.1.2.4	Element <AddTimeStamp>.....	22
126	6.3.1.2.5	Element <SignedProperties>.....	22
127	6.3.1.2.5.1	Requesting SigningTime	22
128	6.3.1.2.5.2	Requesting CommitmentTypeIndication.....	22
129	6.3.1.2.5.3	. Requesting SignatureProductionPlace	22
130	6.3.1.2.5.4	Requesting SignerRole.....	22
131	6.3.1.2.5.5	Requesting data objects format.....	22
132	6.3.2	Element <SignResponse>.....	22
133	6.3.2.1	Attribute Profile.....	Error! Bookmark not defined.
134	6.3.2.2	Element <Result>.....	23
135	6.3.2.3	Element <SignatureObject>.....	23
136	6.4	Profile of Verifying Protocol.....	23
137	6.4.1	Element <VerifyRequest>.....	23
138	6.4.1.1	Attribute Profile.....	23
139	6.4.1.2	Element <OptionalInputs>.....	23
140	6.4.1.2.1	Element <ReturnUpdatedSignature>.....	23
141	6.4.1.3	Element <SignatureObject>.....	23
142	6.4.2	Element <VerifyResponse>.....	23
143	6.4.2.1	Attribute Profile.....	Error! Bookmark not defined.
144	6.4.2.2	Element <OptionalOutputs>.....	23
145	6.4.2.2.1	Element <UpdatedSignature>.....	23
146	6.5	Profile Bindings.....	23
147	6.5.1	Transport Bindings.....	23
148	6.5.2	Security Bindings.....	23
149	6.5.2.1	Security Requirements.....	23
150	6.5.2.2	TLS X.509 Mutual Authentication.....	24
151	7.....		25
152	8	Identifiers defined in this specification.....	26
153	8.1	Predefined advanced electronic signature forms identifiers.....	26
154	9	Editorial Issues.....	27
155	10	References.....	29
156	10.1	Normative.....	29
157	Appendix A.	Revision History.....	30
158	Appendix B.	Notices.....	31
159			

160 1 Introduction

161 The DSS signing and verifying protocols are defined in **[DSSCore]**. As defined in that
162 document, the DSS protocols have a fair degree of flexibility and extensibility. This document
163 defines an abstract profile for the use of the DSS protocols for creating and verifying XML and
164 binary Advanced Electronic Signatures as defined in [COMMENT: Bold these and other
165 references:] **[XAdES]** and **[TS 101 733]**. This document also defines two concrete profiles
166 derived from the abstract one: one for creating and verifying XAdES signatures and the other
167 for creating and verifying signatures as defined in TS 101733.

168 1.1 Notation

169 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
170 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
171 interpreted as described in IETF RFC 2119 **[RFC 2119]**. These keywords are capitalized
172 when used to unambiguously specify requirements over protocol features and behavior that
173 affect the interoperability and security of implementations. When these words are not
174 capitalized, they are meant in their natural-language sense.

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

177 1.2 Namespaces

178 The structures described in this specification are contained in the schema file **[XAdES-ABS-**
179 **XSD]**. All schema listings in the current document are excerpts from the schema file. In the
180 case of a disagreement between the schema file and this document, the schema file takes
181 precedence.

182 This schema is associated with the following XML namespace:

183 `http://www.docs.oasis-open.org/dss/oasis-dss-1.0-profiles-XAdES-cd-01#`

184 If a future version of this specification is needed, it will use a different namespace.

185 Conventional XML namespace prefixes are used in this document:

- 186 o The prefix `dss:` (or no prefix) stands for the DSS core namespace **[Core-XSD]**.
- 187 o The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.
- 188 o The prefix `xades:` stands for ETSI XML Advanced Electronic Signatures (XAdES)
189 document **[XAdES]**.

190 Applications MAY use different namespace prefixes, and MAY use whatever namespace
191 defaulting/scoping conventions they desire, as long as they are compliant with the
192 Namespaces in XML specification **[XML-ns]**.

193 **2 Overview**

194 This document defines three profiles of the protocols specified in: “Digital Signature Services
195 Core Protocol and Elements” **[DSSCore]**.

196 The first one is an abstract profile defining messages for supporting the lifecycle of advanced
197 electronic signatures. Both, XML and binary advanced electronic signatures are supported by
198 this profile.

199 One concrete profile, derived from the aforementioned abstract profile, gives support to the
200 lifecycle of XML advanced electronic signatures as specified in **[XAdES]**.

201 A second concrete profile, also derived from the abstract one, gives support to the lifecycle of
202 binary advanced electronic signatures as specified in **[TS 101733]**.

203 Implementations should implement one of the concrete profiles (or both) in order to request
204 generation or validation of advanced electronic signatures in one of the two formats (or both).

205 3 Advanced Electronic Signature abstract 206 profile

207 3.1 Overview

208 This abstract profile supports operations within each phase of the lifecycle of two types of
209 advanced electronic signature:

- 210 • XML encoded signatures based on **[XMLSig]** such as specified in **[XAdES]**.
- 211 • Binary encoded signatures based on **[RFC 3161]** such as specified in **[TS 101733]**.

212 Henceforward, the document will use the term **advanced signature** when dealing with issues
213 that affect to both types of signatures. The document will use XAdES or TS 101733
214 signatures when dealing with issues that affect one or the other but not both of them.

215 For the generation of advanced signatures, the following operations apply:

- 216 • SignRequest. This operation supports requests for:
 - 217 ○ Generating predefined advanced signature forms as defined in **[XAdES]** and
218 **[TS 101733]**.
 - 219 ○ Generating XML signatures incorporating specific signed/unsigned properties
220 whose combination does not fit any predefined XAdES signature form. In
221 such cases, the form **MUST** have been defined in a proprietary specification
222 and **MUST** be identified by one URI.
 - 223 ○ Generating CMS signatures incorporating specific signed/unsigned attributes
224 whose combination does not fit any predefined **[TS 101733]** signature forms.
225 In such cases, the form **MUST** have been defined in a proprietary
226 specification and **MUST** be identified by one URI.
- 227 • SignResponse. This operation supports delivery of:
 - 228 ○ Predefined advanced signature forms as defined in **[XAdES]** and **[TS**
229 **101733]**.
 - 230 ○ XML signatures with specific properties whose combination does not fit any
231 predefined XAdES signature form. In such cases, the form **MUST** have been
232 defined in some other specification and **MUST** be identified by one URI.
 - 233 ○ CMS signatures incorporating specific signed attributes whose combination
234 does not fit any predefined **[TS 101733]** signature form. In such cases, the
235 form **MUST** have been defined in some other specification and **MUST** be
236 identified by one URI.

237 For advanced signature verification (and updating) the following operations apply:

- 238 • VerifyRequest. This operation supports requests for:
 - 239 ○ Verifying a predefined advanced signature form.
 - 240 ○ Verifying XML signatures incorporating specific properties whose combination
241 does not fit any predefined XAdES signature form.
 - 242 ○ Verifying any of the signatures mentioned above **PLUS** updating them by
243 addition of additional properties (time-stamps, validation data, etc) leading to
244 a predefined XAdES form.
 - 245 ○ Verifying CMS signatures incorporating specific attributes whose combination
246 does not fit any predefined **[TS 101733]** signature form.

- 247 ○ Verifying any of the signatures mentioned above PLUS updating them by
248 addition of additional attributes (time-stamps, validation data, etc) leading to a
249 predefined **[TS 101733]** form.
- 250 ○ Verifying a long-term advanced signature in a certain point of time.
- 251 • VerifyResponse. This operation supports delivery of:
- 252 ○ Advanced signature verification result of signatures mentioned above.
- 253 ○ Advanced signature verification result PLUS the updated signatures as
254 requested.
- 255 ○ Updated signatures as requested.
- 256 The material for each operation will clearly indicate the lifecycle phase it pertains to.

257 **3.2 Profile Features**

258 **3.2.1 Scope**

259 This document profiles the DSS signing and verifying protocols defined in **[DSSCore]**.

260 **3.2.2 Relationship To Other Profiles**

261 The profile in this document is based on the **[DSSCore]**. The profile in this document is not
262 directly implementable, and may be further profiled.

263 **3.2.3 Signature Object**

264 This profile supports the creation and verification of advanced signatures as defined in
265 **[XAdES]** and **[TS 101733]**.

266 This profile also supports update of advanced signatures by addition of unsigned properties
267 (time-stamps and different types of validation data), as specified in **[XAdES]** and **[TS**
268 **101733]**.

269 **3.3 Profile of Signing Protocol**

270 The present profile allows requesting:

- 271 • Predefined forms of advanced electronic signatures as defined in **[XAdES]** and **[TS**
272 **101733]**.
- 273 • Other forms of signatures based in **[XMLSig]** or **[RFC 3369]** defined in other
274 specifications,

275 In both cases, the specific requested form will be identified by an URI.

276 According to this profile, the following predefined advanced signature forms defined in
277 **[XAdES]** and **[TS 101733]** MAY be requested (those forms whose name begin by XAdES-
278 are forms names for XAdES signatures; the other ones denote forms for TS 101733
279 signatures):

- 280 • BES and XAdES-BES. In this form, the signing certificate is secured by the signature
281 itself.
- 282 • EPES and XAdES-EPES. This form incorporates an explicit identifier of the signature
283 policy that will govern the signature generation and verification.
- 284 • ES-T and XAdES-T. This form incorporates a trusted time, by means of a time-stamp
285 token or a time-mark.
- 286 • ES-C and XAdES-C.
- 287 • ES-X and XAdES-X.

288 • ES-X-L and XAdES-X-L.

289 • ES-A and XAdES-A.

290 In addition, the present profile provides means for requesting incorporation in any of the
291 aforementioned forms any of the following properties: SigningTime,
292 CommitmentTypeIndication, SignatureProductionPlace, SignerRole,
293 IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and DataObjectFormat.

294 Other electronic signature forms based in [XMLSig] or [RFC 3369], defined elsewhere, MAY
295 also be requested using the mechanisms defined in this profile.

296 **3.3.1 Element <SignRequest>**

297 **3.3.1.1 Element <OptionalInputs>**

298 None of the optional inputs specified in the [DSS Core] are precluded in this abstract profile.
299 It only constrains some of them and specifies additional optional inputs.

300 **3.3.1.1.1 Element <SignatureType>**

301 This element is OPTIONAL. If present, <SignatureType> SHALL be either:

302 urn:ietf:rfc:3275

303 for requesting XML Signatures, or

304 urn:-ietf:rfc:3369

305 for requesting CMS Signatures, as defined in 7.1 of [DSS Core].

306 If not present the signature type SHALL be implied by the selected <SignaturePolicy> or
307 the signature policy applied by the server.

308 **3.3.1.1.2 Element <SignatureForm>**

309 The form of signature required MAY be indicated using the following optional input

310 `<xs:element name="SignatureForm" type="xs:anyURI" />`

311 If not present the signature form SHALL be implied by the selected <SignaturePolicy> or
312 the signature policy applied by the server.

313 Section 8.1 of this abstract profile defines a set of URIs identifying the predefined advanced
314 electronic signature forms specified in [TS101733] and [XAdES].

315 Should other standard or proprietary specification define new signature forms and their
316 corresponding URIs, concrete sub-profiles of this abstract profile could be defined for giving
317 support to their verification and update.

318 Should a form identified by an URI, admit different properties combinations, the server will
319 produce a specific combination depending on its policy or configuration settings.

320 **3.3.1.1.3 Optional inputs <ClaimedIdentity> / <KeySelector>**

321 As forms defined in [XAdES] and [TS 101733] require that the signing certificate is protected
322 by the signature, the server MUST gain access to that certificate.

323 <dss:ClaimedIdentity> or <dss:KeySelector> optional inputs MAY be present. If
324 they are not present, the server may use means not specified in this profile to identify the
325 signer's key and gain access to its certificate.

326 **3.3.1.1.4 Element <AddTimeStamp>**

327 This element MAY be used by the client to request the inclusion in the advanced signature of
328 a time-stamp on all the data that are to be signed.

329 This profile defines the following value for its `Type` attribute.
330 `urn:oasis:names:tc:dss:1.0:profiles:XAdES:AllDataObjectTimeStamp`
331 Note: `IndividualDataObjectsTimeStamp` is requested using `<SignedProperty>`
332 element as defined in section 3.3.1.1.5.5.

333 3.3.1.1.5 Element `<SignedProperties>`

334 The requester MAY request to the server the addition of optional signed properties using the
335 `<dss:SignedProperties>` element's `<dss:Property>` child profiled as indicated in
336 clauses below.

337 Signed properties that MAY be requested are: `SigningTime`,
338 `CommitmentTypeIndication`, `SignerRole`, `SignatureProductionPlace`,
339 `DataObjectFormat`, and `IndividualDataObjectsTimeStamp`.

340 3.3.1.1.5.1 Requesting `SigningTime`

341 Value for `<Identifier>` element:
342 `urn:oasis:names:tc:dss:1.0:profiles:XAdES:SigningTime`
343 No content is required for `Value` element, since it will be generated by the server.

344 3.3.1.1.5.2 Requesting `CommitmentTypeIndication`

345 Value for `<Identifier>` element:
346 `urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication`
347 `n`
348 When the value of the commitment is fixed by the requester, this property will have a value
349 that the server will incorporate to the advanced signature. In such cases the `<Value>`
350 element MUST have the following content:

```
351 <xs:element name="Commitment">  
352   <xs:complexType>  
353     <xs:choice>  
354       <xs:element ref="xades:CommitmentTypeIndication"/>  
355       <xs:element name="BinaryValue" type="xs:base64Binary"/>  
356     </xs:choice>  
357   </xs:complexType>  
358 </xs:element>
```

359 Element `<xades:CommitmentTypeIndication>` will be present when requesting a XML
360 signature.

361 Element `<BinaryValue>` will be present when requesting a ASN.1 signature. Its contents
362 MUST be the base64 encoding of **CommitmentTypeIndication** ASN.1 type defined in [TS
363 101733], DER-encoded.

364 3.3.1.1.5.3 Requesting `SignatureProductionPlace`

365 Value for `<Identifier>` element:
366 `urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignatureProductionPlace`
367 No content is required for `<Value>` element, as it will be generated by the server.

368 3.3.1.1.5.4 Requesting `SignerRole`

369 Value for `<Identifier>` element:
370 `urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole`

371 When the value of the role is fixed by the requester, this property will have a value that the
372 server will incorporate to the advanced signature. In such cases the <Value> element MUST
373 have the following content:

```
374 <xs:element name="SignerRole">  
375   <xs:complexType>  
376     <xs:choice>  
377       <xs:element ref="xades:SignerRole"/>  
378       <xs:element name="BinaryValue" type="xs:base64Binary"/>  
379     </xs:choice>  
380   </xs:complexType>  
381 </xs:element>
```

382 Element <xades:SignerRole> will be present when requesting a XML signature.

383 Element <BinaryValue> will be present when requesting a ASN.1 signature. Its contents
384 MUST be the base64 encoding of **SignerAttribute** ASN.1 type defined in [TS 101733], DER-
385 encoded.

386 3.3.1.1.5.5 Requesting <xades:IndividualDataObjectTimeStamp>

387 This property is only incorporated in XAdES signatures, not in TS101733 signatures, because
388 an XML signature is able to sign several documents.

389 Value for <Identifier> element:

```
390 urn:oasis:names:tc:dss:1.0:profiles:XAdES:IndividualDataObjectTimeSta  
391 mp
```

392 In this case, the content of <Value> element will be the element
393 <DocsToBeTimeStamped>, defined as shown below.

```
394 <xs:element name="DocsToBeTimeStamped" type="DocReferencesType"/>  
395  
396 <xs:complexType name="DocReferencesType">  
397   <xs:sequence>  
398     <xs:element name="DocReference" maxOccurs="unbounded"  
399       type="DocReferenceType"/>  
400   </xs:sequence>  
401 </xs:complexType>  
402  
403 <xs:complexType name="DocReferenceType">  
404   <xs:attribute name="WhichDocument" type="xs:IDREF"  
405     use="required"/>  
406   <xs:attribute name="RefId" type="xs:string" use="optional"/>  
407 </xs:complexType>
```

408 WhichDocument attribute contains the reference to the document whose time-stamp is
409 requested (see attribute ID in [CoreDSS] section 2.4.1).

410 [XAdES] mandates that <ds:Reference> elements corresponding to signed documents
411 that have been individually time-stamped before being signed, must include an Id attribute.
412 [XAdES] also mandates <xades:IndividualDataObjectsTimeStamp> element to use
413 this Id attribute to indicate what signed documents have actually been time-stamped before
414 signing. See [XAdES] <xades:TimeStampType> and
415 <xades:IndividualDataObjectsTimeStamp> definitions for more details.

416 The client MAY request a value for the <ds:Reference> element's Id attribute using the
417 RefId optional attribute if a <dss:SignedReference> forcing a value for such an attribute
418 is not present in the request. If the request does not specify a value for this attribute, then the
419 server will automatically generate it.

420 3.3.1.1.5.6 Requesting data objects format

421 Both [XAdES] and [TS101733] specify signed properties containing information on the format
422 of the signed documents.

423 Value for Identifier element:

424 urn:oasis:names:tc:dss:1.0:profiles:XAdES:DataObjectFormat

425 When the value of the data object formats are fixed by the requester, this property will have a
426 value that the server will incorporate to the advanced signature. The content of <Value>
427 element will be the element <DocsFormat>, defined as shown below.

```
428 <xs:element name="DocsFormat" type="DocsFormatType" />
429
430
431 <xs:complexType name="DocsFormatType">
432   <xs:sequence>
433     <xs:choice>
434       <xs:element name="DocFormat" type="DocFormatType"
435 maxOccurs="unbounded" />
436       <xs:element name="BinaryValue" type="xs:base64Binary" />
437     </xs:choice>
438   </xs:sequence>
439 </xs:complexType>
440
441 <xs:complexType name="DocFormatType">
442   <xs:complexContent>
443     <xs:extension base="DocReferenceType">
444       <xs:sequence>
445         <xs:element ref="xades:DataObjectFormat" />
446       </xs:sequence >
447     </xs:extension>
448   </xs:complexContent>
449 </xs:complexType>
```

450 Element <DocFormat> will be present when requesting a XML signature.

451 Element <BinaryValue> will be present when requesting a ASN.1 signature. Its contents
452 MUST be the base64 encoding of **ContentHints** ASN.1 type defined in [RFC 2634] DER-
453 encoded.

454 **3.3.2 Element <SignResponse>**

455 This clause profiles the `dss:SignResponse` element for the requests profiled in clause 3.3.1

456 **3.3.2.1 Element <Result>**

457 This profile does not apply any restriction to the `dss:Result` element.

458 **3.3.2.2 Element <SignatureObject>**

459 The following restrictions apply to the contents of `dss:SignatureObject`:

- 460 • When the generation of a CMS based signature is requested, the base-64 encoded
461 signature MUST be present in the `dss:Base64Signature` element.
- 462 • When the generation of an enveloping or detached XMLSig based signature is
463 requested, this element will contain a `ds:Signature` element.

464 **3.3.2.3 Element <DocumentWithSignature>**

465 This element will only appear if an enveloped XML signature is requested.

466 **3.4 Profile of Verifying Protocol**

467 **3.4.1 Element <VerifyRequest>**

468 This clause specifies the profile for the contents of the `dss:VerifyRequest` when used for:

- 469 • Requesting verification of advanced signatures.
470 • Requesting verification of advanced signatures AND update of signatures to other
471 predefined forms.

472 **3.4.1.1 Attribute Profile**

473 The value for the `Profile` attribute, indicating the concrete sub-profile of this abstract profile,
474 MUST be present.

475 **3.4.1.2 Element <OptionalInputs>**

476 This profile specifies restrictions for the following possible children of `dss:OptionalInputs`
477 element:

478 `<dss:ReturnUpdatedSignature>`. This element SHALL be present when the client
479 requests verification of a signature and update to other predefined form of advanced
480 signature.

481 **3.4.1.2.1 Element <ReturnUpdatedSignature>**

482 This element MUST be present when the client requests verification of a signature and
483 update to a predefined form of advanced signature.

484 The `Type` attribute identifies the advanced signature form requested.

485 Acceptable predefined values for this attribute are the URIs specified in table 1 corresponding
486 to the following forms predefined in [TS101733] and [**XAdES**]: XAdES-T/ES-T, XAdES-C/ES-
487 C, XAdES-X/ES-X, XAdES-X-L, ES-X-L, XAdES-A, ES-A.

488 Should other standard or proprietary specification define new signature forms and their
489 corresponding URIs, concrete sub-profiles of this abstract profile could be defined for giving
490 support to their verification and update.

491 When the requested form allows for different contents, the server MUST decide the specific
492 contents of the updated signature delivered, according to its configuration and settings.

493 **4 Element <VerifyResponse>**

494 **4.1.1.1 Element <OptionalOutputs>**

495 This profile specifies restrictions for the following optional outputs:

496 <dss:UpdatedSignature>. This element **MUST** be present in a successful response of a
497 request containing <dss:ReturnUpdatedSignature>.

498 No additional restrictions are applied by this profile to the contents of any additional outputs.

499 **4.1.1.1.1 Element <UpdatedSignature>**

500 The content of the dss:UpdatedSignature will be a dss:SignatureObject element
501 profiled according to the following rules:

- 502 • When the update of a CMS based signature is requested, the base-64 encoded
503 signature itself **MUST** be present in the dss:Base64Signature element.
- 504 • When the update of a XMLSig based signature is requested, one of the following
505 elements **MUST** appear:
 - 506 • The ds:Signature containing a XMLSig based signature.
 - 507 • The dss:SignaturePtr pointing to the XMLSig based signature embedded in one of the
508 input documents.

509 5 XML Advanced Electronic Signatures 510 concrete Profile

511 5.1 Overview

512 This concrete profile supports operations within each phase of the lifecycle of XML Advanced
513 Electronic Signature based on **[XMLSig]** such as specified in **[XAdES]**. It will then provide all
514 the features related to XAdES signatures that are specified in the abstract profile defined in
515 section 3.

516 For the generation of XAdES signatures, the following operations apply:

- 517 • SignRequest. This operation supports requests for:
 - 518 ○ Generating predefined advanced signature forms as defined in **[XAdES]**.
 - 519 ○ Generating XML signatures incorporating specific signed/unsigned properties
520 whose combination does not fit any predefined XAdES signature form. In
521 such cases, the form **MUST** have been defined in a proprietary specification
522 and **MUST** be identified by one URI.
 - 523 ○ SignResponse. This operation supports delivery of:
 - 524 ○ Predefined advanced signature forms as defined in **[XAdES]**.
 - 525 ○ XML signatures with specific properties whose combination does not fit any
526 predefined XAdES signature form. In such cases, the form **MUST** have been
527 defined in a proprietary specification and **MUST** be identified by one URI.

528 For verification [and updating] of XAdES signatures the following operations apply:

- 529 • VerifyRequest. This operation supports requests for:
 - 530 ○ Verifying a predefined XAdES signature form.
 - 531 ○ Verifying XML signatures incorporating specific properties whose combination
532 does not fit any predefined XAdES signature form.
 - 533 ○ Verifying any of the signatures mentioned above **PLUS** updating them by
534 addition of additional properties (time-stamps, validation data, etc) leading to
535 a predefined XAdES form.
 - 536 ○ Verifying a long-term advanced signature in a certain point of time.
- 537 • VerifyResponse. This operation supports delivery of:
 - 538 ○ Advanced signature verification result of signatures mentioned above.
 - 539 ○ Advanced signature verification result **PLUS** the updated signatures as
540 requested.
 - 541 ○ Updated signatures as requested.

542 5.2 Profile features

543 5.2.1 Identifier

544 urn:oasis:names:tc:dss:1.0:profiles:XAdES.

545 5.2.2 Scope

546 This document profiles the DSS abstract profile defined in section 3 of the present document.

547 **5.2.3 Relationship To Other Profiles**

548 The profile in this section is based on the abstract profile for Advanced Electronic Signatures
549 defined in section 3.

550 **5.2.4 Signature Object**

551 This profile supports the creation and verification of XML advanced signatures as defined in
552 **[XAdES]**.

553 This profile also supports verification and update of advanced signatures by addition of
554 unsigned properties (time-stamps and different types of validation data), as specified in
555 **[XAdES]**

556 **5.2.5 Transport Binding**

557 This profile does not specify or constrain the transport binding.

558 **5.2.6 Security Binding**

559 This profile does not specify or constrain the security binding.

560 **5.3 Profile of Signing Protocol**

561 The present profile allows requesting:

- 562 • Predefined forms of advanced electronic signatures as defined in **[XAdES]**.
- 563 • Other forms of signatures based in **[XMLSig]** defined in other specifications,

564 In both cases, the specific requested form will be identified by an URI.

565 According to this profile, the following predefined advanced signature forms defined in
566 **[XAdES]** MAY be requested: XAdES-BES, XAdES-EPES, XAdES-T, XAdES-C, XAdES-X,
567 XAdES-X-L., and XAdES-A.

568 In addition, the present profile provides means for requesting incorporation in any of the
569 aforementioned forms any of the following properties: SigningTime,
570 CommitmentTypeIndication, SignatureProductionPlace, SignerRole,
571 IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and DataObjectFormat.

572 Other electronic signature forms based in **[XMLSig]** defined elsewhere MAY also be
573 requested using the mechanisms defined in this profile.

574 **5.3.1 Element <SignRequest>**

575 **5.3.1.1 Attribute Profile**

576 urn:oasis:names:tc:dss:1.0:profiles:XAdES.

577 **5.3.1.2 Element <OptionalInputs>**

578 None of the optional inputs specified in the **[DSS Core]** are precluded in this abstract profile.
579 It only constrains some of them and specifies additional optional inputs.

580 **5.3.1.2.1 Element <SignatureType>**

581 This element is MANDATORY. Its vaule MUST be:

582 urn:ietf/rfc:3275

583 **5.3.1.2.2 Element <SignatureForm>**

584 Usage of these elements is according to what is stated in section 3.3.1.1.2 .

585 **5.3.1.2.3 Optional inputs < ClaimedIdentity> / <KeySelector>**

586 Usage of these elements is according to what is stated in section 3.3.1.1.3.

587 **5.3.1.2.4 Element <AddTimeStamp>**

588 Usage of these elements is according to what is stated in section 3.3.1.1.4.

589 **5.3.1.2.5 Element <SignedProperties>**

590 **5.3.1.2.5.1 Requesting SigningTime**

591 Usage of these elements is according to what is stated in section 3.3.1.1.5.1.

592 **5.3.1.2.5.2 Requesting CommitmentTypeIndication**

593 The value for <Identifier> element is the one defined in section 3.3.1.1.5.2:

594 urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication

595 When the value of the commitment is established by the requester, the <Value> element
596 MUST contain a <Commitment> element as defined in section 3.3.1.1.5.2 with the
597 <xades:CommitmentTypeIndication> child.

598 **5.3.1.2.5.3 . Requesting SignatureProductionPlace**

599 Usage of these elements is according to what is stated in section 3.3.1.1.5.3

600 **5.3.1.2.5.4 Requesting SignerRole**

601 Value for <Identifier> element:

602 urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole

603 When the value of the role is fixed by the requester, the <Value> element MUST contain a
604 <SignerRole> element as defined in section 3.3.1.1.5.4 with the <xades:SignerRole>
605 child.

606 **5.3.1.2.5.5 Requesting <xades:IndividualDataObjectTimeStamp>**

607 Usage of these elements is according to what is stated in section 3.3.1.1.5.5.

608 **5.3.1.2.5.6 Requesting data objects format**

609 Value for <Identifier> element:

610 urn:oasis:names:tc:dss:1.0:profiles:XAdES:DataObjectFormat

611 When the value of the data object formats are fixed by the requester, the <Value> element
612 MUST contain a <DocsFormat> element as defined in section 3.3.1.1.5.6 with the
613 <DocFormat> child.

614 **5.3.2 Element <SignResponse>**

615 This section profiles the `dss:SignResponse` element for the requests profiled in clause
616 5.3.1.

617 **5.3.2.1 Element <Result>**

618 This profile does not apply any restriction to the `dss:Result` element.

619 **5.3.2.2 Element <SignatureObject>**

620 The content of this element MUST be one of the following:

- 621 • A `ds:Signature` element containing a XMLSig based signature.
- 622 • A `dss:SignaturePtr` pointing to the XMLSig based signature embedded in one of
623 the output documents. In such a case, the optional output element, containing the
624 signature created, MUST be present within the `dss:DocumentWithSignature`
625 element

626 **5.4 Profile of Verifying Protocol**

627 **5.4.1 Element <VerifyRequest>**

628 **5.4.1.1 Attribute Profile**

629 `urn:oasis:names:tc:dss:1.0:profiles:XAdES.`

630 **5.4.1.2 Element <OptionalInputs>**

631 **5.4.1.2.1 Element <ReturnUpdatedSignature>**

632 Usage of these elements is according to what is stated in section 3.4.1.2.1.

633 **5.4.1.3 Element <SignatureObject>**

634 The `dss:SignatureObject` element will have one of the following contents:

- 635 • A `ds:Signature` containing a XMLSig based signature.
- 636 • A `dss:SignaturePtr` pointing to the XMLSig based signature embedded in one of the
637 inputdocuments.

638 **5.4.2 Element <VerifyResponse>**

639 **5.4.2.1 Element <OptionalOutputs>**

640 Usage of these elements is according to what is stated in section 4.1.1.1.

641 **5.4.2.1.1 Element <UpdatedSignature>**

642 The content of the `dss:UpdatedSignature` will be a `dss:SignatureObject` element
643 with one of the following contents:

- 644 • A `ds:Signature` containing a XMLSig based signature.
- 645 • A `dss:SignaturePtr` pointing to the XMLSig based signature embedded in one of the
646 inputdocuments.

647 **5.5 Profile Bindings**

648 **5.5.1 Transport Bindings**

649 Messages transported in this profile MAY be transported by the HTTP POST Transport
650 Binding and the SOAP 1.2 Transport Binding defined in **[DSSCore]**.

651 **5.5.2 Security Bindings**

652 **5.5.2.1 Security Requirements**

653 This profile MUST use security bindings that:

- 654
- Authenticates the requester to the DSS server
 - 655 • Authenticates the DSS server to the DSS client
 - 656 • Protects the integrity of a request, response and the association of response to the
657 request.
 - 658 • Optionally, protects the confidentiality of a request and response.

659 The following MAY be used to meet these requirements.

660 **5.5.2.2 TLS X.509 Mutual Authentication**

661 This profile is secured using the TLS X.509 Mutual Authentication Binding defined in
662 **[DSSCore]**.

663

664 6 CMS-based Advanced Electronic Signature 665 profile

666 6.1 Overview

667 This concrete profile supports operations within each phase of the lifecycle of CMS based
668 Advanced Electronic Signature based on [RFC 3369] such as specified in [TS 101733]. It will
669 then provide all the features related to TS 101733 signatures that are specified in the abstract
670 profile defined in section 3.

671 For the generation of TS101733 signatures, the following operations apply:

- 672 • SignRequest. This operation supports requests for:
 - 673 ○ Generating predefined advanced signature forms as defined in [TS101733].
 - 674 ○ Generating CMS signatures incorporating specific signed/unsigned attributes
675 whose combination does not fit any predefined [TS 101733] signature forms.
676 In such cases, the form MUST have been defined in a proprietary
677 specification and MUST be identified by one URI.
 - 678 ○ SignResponse. This operation supports delivery of:
 - 679 ○ Predefined advanced signature forms as defined in [TS101733].
 - 680 ○ CMS signatures incorporating specific signed attributes whose combination
681 does not fit any predefined [TS 101733] signature form. In such cases, the
682 form MUST have been defined in a proprietary specification and MUST be
683 identified by one URI.

684 For verification [and updating] of signatures as specified in [TS 101733] the following
685 operations apply:

- 686 • VerifyRequest. This operation supports requests for:
 - 687 ○ Verifying a predefined [TS 101733] signature form.
 - 688 ○ Verifying CMS signatures incorporating specific attributes whose combination
689 does not fit any predefined [TS 101733] signature form.
 - 690 ○ Verifying any of the signatures mentioned above PLUS updating them by
691 addition of additional attributes (time-stamps, validation data, etc) leading to a
692 predefined [TS 101733] form.
 - 693 ○ Verifying a long-term advanced signature in a certain point of time.
- 694 • VerifyResponse. This operation supports delivery of:
 - 695 ○ Advanced signature verification result of signatures mentioned above.
 - 696 ○ Advanced signature verification result PLUS the updated signatures as
697 requested.
 - 698 ○ Updated signatures as requested.

699 6.2 Profile features

700 6.2.1 Identifier

701 urn:oasis:names:tc:dss:1.0:profiles:CAAdES.

702 **6.2.2 Scope**

703 This document profiles the DSS abstract profile defined in section 3 of the present document.

704 **6.2.3 Relationship To Other Profiles**

705 The profile in this document is based on the abstract profile for Advanced Electronic
706 Signatures defined in section 3.

707 **6.2.4 Signature Object**

708 This profile supports the creation and verification of CMS based advanced signatures as
709 defined in **[TS101733]**.

710 This profile also supports verification and update of advanced signatures by addition of
711 unsigned properties (time-stamps and different types of validation data), as specified in
712 **[TS101733]**

713 **6.2.5 Transport Binding**

714 This profile does not specify or constrain the transport binding.

715 **6.2.6 Security Binding**

716 This profile does not specify or constrain the security binding.

717 **6.3 Profile of Signing Protocol**

718 The present profile allows requesting:

- 719 • Predefined forms of advanced electronic signatures as defined in **[TS 101733]**.
- 720 • Other forms of signatures based in **[RFC 3369]** defined in other specifications,

721 In both cases, the specific requested form will be identified by an URI.

722 According to this profile, the following predefined advanced signature forms defined in **[TS**
723 **101733]** MAY be requested: BES, EPES, ES-T, ES-C, ES-X, ES-X-L, and ES-A

724 In addition, the present profile provides means for requesting incorporation in any of the
725 aforementioned forms any of the following properties: SigningTime,
726 CommitmentTypeIndication, SignatureProductionPlace, SignerRole,
727 IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and DataObjectFormat.

728 Other electronic signature forms based in **[RFC 3369]**, defined elsewhere, MAY also be
729 requested using the mechanisms defined in this profile.

730 **6.3.1 Element <SignRequest>**

731 **6.3.1.1 Attribute Profile**

732 urn:oasis:names:tc:dss:1.0:profiles:CAAdES.

733 **6.3.1.2 Element <OptionalInputs>**

734 None of the optional inputs specified in the **[DSS Core]** are precluded in this abstract profile.
735 It only constrains some of them and specifies additional optional inputs.

736 **6.3.1.2.1 Element <SignatureType>**

737 This element is MANDATORY. Its value MUST be:

738 urn:ietf:rfc:3369

739 **6.3.1.2.2 Element <SignatureForm>**

740 Usage of these elements is according to what is stated in section 3.3.1.1.2 .

741 **6.3.1.2.3 Optional inputs < ClaimedIdentity> / <KeySelector>**

742 Usage of these elements is according to what is stated in section 3.3.1.1.3.

743 **6.3.1.2.4 Element <AddTimeStamp>**

744 Usage of these elements is according to what is stated in section 3.3.1.1.4.

745 **6.3.1.2.5 Element <SignedProperties>**

746 This section profiles section 3.3.1.1.5.

747 **6.3.1.2.5.1 Requesting SigningTime**

748 Usage of these elements is according to what is stated in section 3.3.1.1.5.1.

749 **6.3.1.2.5.2 Requesting CommitmentTypeIndication**

750 The value for <Identifier> element is the one defined in section 3.3.1.1.5.2:

751 urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication

752 When the value of the commitment is established by the requester, the <Value> element
753 MUST contain a <Commitment> element as defined in section 3.3.1.1.5.2 with the
754 <BinaryValue> child containing the base64encoding of **CommitmentTypeIndication**
755 ASN.1 type as specified in [TS101733], DER-encoded.

756 **6.3.1.2.5.3 . Requesting SignatureProductionPlace**

757 Usage of these elements is according to what is stated in section 3.3.1.1.5.3

758 **6.3.1.2.5.4 Requesting SignerRole**

759 Value for <Identifier> element:

760 urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole

761 When the value of the role is fixed by the requester, the <Value> element MUST contain a
762 <SignerRole> element as defined in section 3.3.1.1.5.4 with the <BinaryValue> child
763 containing the base64encoding of **SignerAttribute** ASN.1 type as specified in [TS101733],
764 DER-encoded.

765 **6.3.1.2.5.5 Requesting data objects format**

766 Value for <Identifier> element:

767 urn:oasis:names:tc:dss:1.0:profiles:XAdES:DataObjectFormat

768 When the value of the data object formats are fixed by the requester, the <Value> element
769 MUST contain a <DocsFormat> element as defined in section 3.3.1.1.5.6 with the
770 <BinaryValue> child containing the base64encoding of **ContentHints** ASN.1 type as
771 specified in [TS101733], DER-encoded.

772 **6.3.2 Element <SignResponse>**

773 This section profiles the `dss:SignResponse` element for the requests profiled in clause
774 5.3.1.

775 **6.3.2.1 Element <Result>**

776 This profile does not apply any restriction to the `dss:Result` element.

777 **6.3.2.2 Element <SignatureObject>**

778 The `dss:SignatureObject` MUST contain the `dss:Base64Signature` child with a CMS
779 based signature base-64 encoded.

780 **6.4 Profile of Verifying Protocol**

781 **6.4.1 Element <VerifyRequest>**

782 **6.4.1.1 Attribute Profile**

783 `urn:oasis:names:tc:dss:1.0:profiles:CAAdES`.

784 **6.4.1.2 Element <OptionalInputs>**

785 **6.4.1.2.1 Element <ReturnUpdatedSignature>**

786 Usage of these elements is according to what is stated in section 3.4.1.2.1.

787 **6.4.1.3 Element <SignatureObject>**

788 The `dss:SignatureObject` element MUST contain the `dss:Base64Signature` child
789 with a CMS based signature base64 encoded.

790 **6.4.2 Element <VerifyResponse>**

791 **6.4.2.1 Element <OptionalOutputs>**

792 Usage of these elements is according to what is stated in section 4.1.1.1.

793 **6.4.2.1.1 Element <UpdatedSignature>**

794 The content of the `dss:UpdatedSignature` will be a `dss:SignatureObject` element
795 with one of the following contents:

- 796 • A `dss:Base64Signature` element with the CMS based signature base64 encoded.

797 **6.5 Profile Bindings**

798 **6.5.1 Transport Bindings**

799 Messages transported in this profile MAY be transported by the HTTP POST Transport
800 Binding and the SOAP 1.2 Transport Binding defined in **[DSSCore]**.

801 **6.5.2 Security Bindings**

802 **6.5.2.1 Security Requirements**

803 This profile MUST use security bindings that:

- 804 • Authenticates the requester to the DSS server
- 805 • Authenticates the DSS server to the DSS client
- 806 • Protects the integrity of a request, response and the association of response to the
807 request.

808 • Optionally, protects the confidentiality of a request and response.

809 The following MAY be used to meet these requirements.

810 **6.5.2.2 TLS X.509 Mutual Authentication**

811 This profile is secured using the TLS X.509 Mutual Authentication Binding defined in
812 **[DSSCore]**.

813

815 **8 Identifiers defined in this specification**

816 **8.1 Predefined advanced electronic signature forms**
817 **identifiers**

818 The table below shows the URIs for standard forms of advanced electronic signature:

819

Advanced signature FORM	URI
XAdES-BES BES	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:BES
XAdES-EPES EPES	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:EPES
XAdES-T ES-T	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-T
XAdES-C ES-C	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-C
XAdES-X ES-X	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-X
XAdES-X-L ES-X-L	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-X-L
XAdES-A ES-X-A	urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-A

820

Table 1.

9 Editorial Issues

821

822 1. The current text only allows request of signature forms identified by an URI defined
823 somewhere. There are in [XAdES] and in [TS 101733] certain forms that allow
824 different combination of properties. If we leave the standard as it is now, we are
825 leaving the server to decide which combination of properties to select or we assume
826 that any combination of properties in each form will have its unique identifier.

827 **Report:** Comments received in favor of leave the text as it is. Forms requested by
828 URI. If possible different combinations of properties, the server decides

829 **Satus:** CLOSED if no objections.

830 2. Should not this abstract profile also allow requesting the updating of the signature by
831 enumeration of the properties desired?. In this way, this profile would allow both
832 mechanisms to update signatures: by identifying the form or by identifying the set of
833 unsigned properties.

834 **Report:** Comments received in favor of leave the text as it is

835 **Satus:** CLOSED if no objections.

836 3. Section 3.3.1.1.3. A proposal has been made to associate the presence in the
837 request of <ClaimedIdentity> to the production of a signature where the signer's
838 certificate is protected by <xades:SigningCertificate> property, and if instead,
839 <KeySelector> is present then the signature will contain a signed <ds:KeyInfo>
840 with a <ds:X509Certificate> element. I would prefer not linking the information
841 required by the server to gain access to the signer's certificate with the mechanism
842 selected in the signature for protecting this certificate.

843 **Report:** Comments received in favor of leave the text as it is

844 **Satus:** CLOSED if no objections.

845 4. Section 3.3.1.1.4. Proposal for suppress any mention to time-marking and capabilities
846 for the protocol to differentiate, when requesting ES-T or XAdES-T a time-stamp or a
847 time-mark.

848 **Satus:** CLOSED. Leave as it is now.

849 5. Sections 3.3.1.1.5.2, 3.3.1.1.5.3, 3.3.1.1.5.4, and 3.3.1.1.5.6. In the former version
850 the values of the signed properties passed to the server was left open. A proposal
851 was made to force them to be elements whose types would be those defined in
852 [XAdES]. The current version allows them to be elements of types defined in
853 [XAdES] when requesting XML Sig based signatures OR <BinaryValue>
854 containing the base 64 encoded value of the corresponding property defined in [TS
855 101733], DER-encoded, since both kind of signatures may be requested to the
856 server.

857 **Report:** Comments received accepting changes.

858 **Satus:** CLOSED.

859 6. Section 3.3.1.1.5.5. A proposal was made to use <dss:DocumentBaseType> for
860 supporting request of individualDataObjectsTimeStamp property.

861 <dss:DocumentBaseType> does contain much more information than the required
862 for pointing to the data object to be time-stamped before signing and give indication
863 of the Id attribute that its corresponding <ds:Reference> must have. In fact the
864 <DocsToBeTimeStamped> element has quite a lot of commonality with
865 <dss:SignedReference>, except the <ds:Transforms>. But as there is not a
866 type definition, a new schema definition must be generated.

867 **Report:** Comments received accepting changes.

- 868 **Status:** CLOSED.
- 869 7. Previous version explicitly mentioned that this profile would allow for requesting
870 verification of signatures in one specific time. As this is something covered by the
871 core, this mention has been suppressed. Comments have been raised relating this
872 time with the time appearing in `<xades:SigningTime>`. From my point of view one
873 thing is the time that the signer claims to have signed and a different issue is the time
874 when the verifier verifies the signature.
- 875 **Report:** Comments received accepting changes.
- 876 **Status:** CLOSED.
- 877 8. A new element has been defined as optional input to allow requesting the server
878 update of a signature without verification.
- 879 **Report:** Comments pointing out the weakness that this element could bring in future
880 as it would mean that some servers could update signatures without verifying them,
881 and this would bring other ways to ascertain that a updated signature had actually
882 been verified.
- 883 **Decision:** to suppress this feature. The corresponding section has been eliminated
884 from this version.
- 885 **Status:** CLOSED.
- 886 9. Requesting CounterSignature is not explicitly mentioned here. I think that it would
887 always be possible to use the core to get a copy of `<ds:SignatureValue>` element
888 from the `<ds:Signature>` and send it to the server using core protocol. Does
889 anyone see any reason for including a new `<GenerateAsCounterSignature>`
890 optional input?
- 891 **Report:** Comments received suggesting not go into this issue.
- 892 **Status:** CLOSED until further consideration.
- 893 10. The comment has been made that in order to identify the signer's key and gain
894 access to the signer's certificate, there may be mechanisms out of the dss-core, so
895 that it does not make sense to make mandatory to use `<dss:ClaimedIdentity>` or
896 `<dss:KeySelector>` in section 3.3.1.1.3. I agree with that comment. Text has been
897 accordingly modified.
- 898 **Status:** CLOSED.
- 899 11. Section 3.3.1.1.5.5. A comment has been made to suppress optional `RefId` attribute
900 and leave the server decide. So far I have kept it, as I still think that it is worth to give
901 the client the opportunity to request it. In addition I have added text clarifying the
902 relationship with `RefId` attribute in `<dss:SignedReferences>`.
- 903 **Report:** Comments received accepting text as it is.
- 904 **Status:** CLOSED.

905 **10References**

906 **10.1 Normative**

- 907 **[Core-XSD]** T. Perrin et al. *DSS Schema*. OASIS, **(MONTH/YEAR TBD)**
- 908 **[DSSCore]** T. Perrin et al. *Digital Signature Service Core Protocols and Elements*. OASIS,
909 **(MONTH/YEAR TBD)**
- 910 **[RFC 2119]** S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF
911 RFC 2396, August 1998.
- 912 <http://www.ietf.org/rfc/rfc2396.txt>.
- 913 **[TS 101733]** Advanced Electronic Signatures. ETSI TS 101 733.
- 914 **[XAdES]** XML Advanced Electronic Signatures. ETSI TS 101 903, February 2002 (shortly
915 to be reissued).
- 916 **[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML*. W3C
917 Recommendation, January 1999.
- 918 <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- 919 **[XMLSig]** D. Eastlake et al. *XML-Signature Syntax and Processing*. W3C
920 Recommendation, February 2002.
- 921 <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- 922 **[RFC 2634]** P. Hoffman (ed.). Enhanced Security Services for S/MIME, June 1999.
- 923 **[RFC 3369]** Message Syntax (CMS). R. Housley. August 2002.
- 924

Appendix A. Revision History

Rev	Date	By Whom	What
wd-01	2004-03-08	Juan Carlos Cruellas	Initial, incomplete version: SignRequest for predefined forms + optional properties.
wd-02	2004-03-08	Juan Carlos Cruellas	Second version of the initial version: it incorporates SignRequest-SignResponse and VerifyRequest-VerifyResponse. No capability for requesting individually any property. This is still an on-going discussion.
wd-03	2004-06-18	Juan Carlos Cruellas	Third version. Quite a lot of editorial work done. No capability for requesting individually any property. This is still an on-going discussion.
wd-04	2004-08-09	Juan Carlos Cruellas	Fourth version: Suppressed <UpdateSignatureOnly> element. So far: signature forms identified by URI. Not possibility of requesting properties by enumeration. Solved most of editorial issues. Small editorial changes.
wd-05	2004-10-08	Juan Carlos Cruellas	Fifth version: Addition of two concrete sub-profiles: one for XAdES and the other for TS 101733
wd-06	2004-11-09	Juan Carlos Cruellas	Sixth version: Addition of bindings for concrete profiles. Additional changes from comments raised before voting as a CD

Appendix B. Notices

927 OASIS takes no position regarding the validity or scope of any intellectual property or other
928 rights that might be claimed to pertain to the implementation or use of the technology
929 described in this document or the extent to which any license under such rights might or might
930 not be available; neither does it represent that it has made any effort to identify any such
931 rights. Information on OASIS's procedures with respect to rights in OASIS specifications can
932 be found at the OASIS website. Copies of claims of rights made available for publication and
933 any assurances of licenses to be made available, or the result of an attempt made to obtain a
934 general license or permission for the use of such proprietary rights by implementors or users
935 of this specification, can be obtained from the OASIS Executive Director.

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

939 Copyright © OASIS Open 2003. *All Rights Reserved.*

940 This document and translations of it may be copied and furnished to others, and derivative
941 works that comment on or otherwise explain it or assist in its implementation may be
942 prepared, copied, published and distributed, in whole or in part, without restriction of any kind,
943 provided that the above copyright notice and this paragraph are included on all such copies
944 and derivative works. However, this document itself does not be modified in any way, such as
945 by removing the copyright notice or references to OASIS, except as needed for the purpose
946 of developing OASIS specifications, in which case the procedures for copyrights defined in
947 the OASIS Intellectual Property Rights document must be followed, or as required to translate
948 it into languages other than English.

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

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