



# XML Timestamping Profile of the OASIS Digital Signature Services Version 1.0

## Committee Specification

13 February 2007

### Specification URIs:

#### This Version:

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-r1.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-r1.pdf>

#### Latest Version:

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-r1.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-r1.pdf>

### Technical Committee:

OASIS Digital Signature Services TC

### Chair(s):

Nick Pope, Thales eSecurity

Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)

### Editor(s):

Trevor Perrin, individual

Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)

### Related work:

This specification is related to:

- [oasis-dss-core-spec-cs-v1.0-r1](#)

### Abstract:

This document profiles the OASIS DSS core protocols for the purpose of creating and verifying XML-encoded time-stamps.

### Status:

This document was last revised or approved by the OASIS Digital Signature Services TC on the above date. The level of approval is also listed above. Check the current location

31 noted above for possible later revisions of this document. This document is updated  
32 periodically on no particular schedule.

33 Technical Committee members should send comments on this specification to the  
34 Technical Committee's email list. Others should send comments to the Technical  
35 Committee by using the "Send A Comment" button on the Technical Committee's web  
36 page at <http://www.oasis-open.org/committees/dss>.

37 For information on whether any patents have been disclosed that may be essential to  
38 implementing this specification, and any offers of patent licensing terms, please refer to  
39 the Intellectual Property Rights section of the Technical Committee web page  
40 (<http://www.oasis-open.org/committees/dss/ipr.php>).

41 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/dss)  
42 [open.org/committees/dss](http://www.oasis-open.org/committees/dss).

---

## Notices

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

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

56 Copyright © OASIS® 1993–2007. All Rights Reserved. OASIS trademark, IPR and other policies  
57 apply.

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

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

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

74 The names "OASIS" are trademarks of OASIS, the owner and developer of this specification, and  
75 should be used only to refer to the organization and its official outputs. OASIS welcomes  
76 reference to, and implementation and use of, specifications, while reserving the right to enforce  
77 its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for  
78 above guidance.

---

## Table of Contents

80	1	Introduction.....	5
81	1.1	Terminology .....	5
82	1.2	Normative References .....	5
83	1.3	Non-Normative References .....	5
84	1.4	Namespaces .....	5
85	2	Profile Features .....	7
86	2.1	Identifier .....	7
87	2.2	Scope .....	7
88	2.3	Relationship To Other Profiles .....	7
89	2.4	Signature Object .....	7
90	2.5	Transport Binding.....	7
91	2.6	Security Binding .....	7
92	3	Profile of Signing Protocol .....	8
93	3.1	Element <SignRequest> .....	8
94	3.1.1	Element <OptionalInputs>.....	8
95	3.1.2	Element <InputDocuments>.....	9
96	3.2	Element <SignResponse> .....	9
97	3.2.1	Element <Result> .....	9
98	3.2.2	Element <OptionalOutputs> .....	9
99	3.2.3	Element <SignatureObject> .....	9
100	4	Profile of Verifying Protocol .....	11
101	4.1	Element <VerifyRequest> .....	11
102	4.1.1	Element <OptionalInputs>.....	11
103	4.1.2	Element <SignatureObject> .....	11
104	4.1.3	Element <InputDocuments>.....	11
105	4.2	Element <VerifyResponse> .....	11
106	4.2.1	Element <Result> .....	11
107	4.2.2	Element <OptionalOutputs> .....	12
108	A.	Acknowledgements .....	13
109			

---

## 110 1 Introduction

111 The DSS signing and verifying protocols are defined in **[DSSCore]**. As defined in that document,  
112 these protocols have a fair degree of flexibility and extensibility. This document profiles these  
113 protocols to limit their flexibility and extend them in concrete ways. The resulting profile is  
114 suitable for implementation and interoperability.

115 The following sections describe how to understand the rest of this document.

### 116 1.1 Terminology

117 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”,  
118 “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be  
119 interpreted as described in IETF RFC 2119 **[RFC 2119]**. These keywords are capitalized when  
120 used to unambiguously specify requirements over protocol features and behavior that affect the  
121 interoperability and security of implementations. When these words are not capitalized, they are  
122 meant in their natural-language sense.

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

### 125 1.2 Normative References

126	<b>[Core-XSD]</b>	S. Drees et al. <i>DSS Schema</i> . OASIS, February 2007
127	<b>[DSSCore]</b>	S. Drees et al. <i>Digital Signature Service Core Protocols and Elements</i> . 128 OASIS, February 2007
129	<b>[TST-XSD]</b>	T. Perrin et al. <i>Timestamping Profile Schema</i> , OASIS, , February 2007
130	<b>[RFC 2119]</b>	S. Bradner. <i>Key words for use in RFCs to Indicate Requirement Levels</i> . 131 <a href="http://www.ietf.org/rfc/rfc2396.txt">http://www.ietf.org/rfc/rfc2396.txt</a> , IETF RFC 2396, August 1998.
132	<b>[XML-ns]</b>	T. Bray, D. Hollander, A. Layman. <i>Namespaces in XML</i> . 133 <a href="http://www.w3.org/TR/1999/REC-xml-names-19990114">http://www.w3.org/TR/1999/REC-xml-names-19990114</a> , W3C 134 Recommendation, January 1999.
135	<b>[XMLSig]</b>	D. Eastlake et al. <i>XML-Signature Syntax and Processing</i> . 136 <a href="http://www.w3.org/TR/1999/REC-xml-names-19990114">http://www.w3.org/TR/1999/REC-xml-names-19990114</a> , W3C 137 Recommendation, February 2002.

### 138 1.3 Non-Normative References

#### 139 1.4 Namespaces

140 The structures described in this specification are contained in the schema file [TST-XSD]. All  
141 schema listings in the current document are excerpts from the schema file. In the case of a  
142 disagreement between the schema file and this document, the schema file takes precedence.

143 This schema is associated with the following XML namespace:

144 `urn:oasis:names:tc:dss:1.0:profiles:TimeStamp:schema#`

145 Conventional XML namespace prefixes are used in this document:  
146 • The prefix `dss:` stands for the DSS core namespace [**Core-XSD**].  
147 Applications MAY use different namespace prefixes, and MAY use whatever namespace  
148 defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces  
149 in XML specification [**XML-ns**].  
150

---

151 **2 Profile Features**

152 **2.1 Identifier**

153 **urn:oasis:names:tc:dss:1.0:profiles:timestamping**

154 **2.2 Scope**

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

156 **2.3 Relationship To Other Profiles**

157 This profile is based directly on the **[DSSCore]**.

158 **2.4 Signature Object**

159 This profile supports the creation and verification of isolated `<dss:Timestamp>` elements as  
160 defined in **[DSSCore]**. These elements can wrap different types of time-stamp tokens; this profile  
161 does not specify or constrain the internal structure of the `<dss:Timestamp>`, unless the  
162 `<dss:SignatureType>` optional input is used (see section 3.1.1).

163 **2.5 Transport Binding**

164 This profile is transported using the HTTP POST Transport Binding defined in **[DSSCore]**.

165 **2.6 Security Binding**

166 This profile is secured using the TLS X.509 Server Authentication Binding defined in **[DSSCore]**.

167

168

---

## 169 3 Profile of Signing Protocol

### 170 3.1 Element <SignRequest>

#### 171 3.1.1 Element <OptionalInputs>

172 The <dss:SignatureType> optional input from **[DSSCore]** is supported and may be sent by  
173 the client. The timestamping specific optional input <RenewTimestamp> may also be supported  
174 and may be sent by the client. No other optional inputs are supported.

##### 175 3.1.1.1 Element <SignatureType>

176 The <dss:SignatureType> optional input may be one of these values, from section 7. of  
177 **[DSSCore]**:

178 urn:oasis:names:tc:dss:1.0:core:schema:XMLTimeStampToken

179 urn:ietf:rfc:3161

180 Servers may support other values. However, servers are under no obligation to support *any*  
181 particular values. Thus, clients using the <dss:SignatureType> optional input may not  
182 interoperate with certain servers.

##### 183 3.1.1.2 Element <RenewTimestamp>

184 The <RenewTimestamp> optional input element indicates to the server that the current sign  
185 request is a request for the renewal of an existing timestamp on data that were timestamped in  
186 the past, so that the validity period of the existing timestamp is effectively extended.

187

```
188 <xs:element name="RenewTimestamp">  
189   <xs:complexType>  
190     <xs:sequence>  
191       <xs:element ref="PreviousTimestamp">  
192         <xs:sequence>  
193       </xs:complexType>  
194     </xs:element>  
195   <xs:element name="PreviousTimestamp">  
196     <xs:complexType>  
197       <xs:sequence>  
198         <xs:element ref="dss:Timestamp">  
199       <xs:sequence>  
200     </xs:complexType>  
201   </xs:element>
```

202

203 If the <RenewTimestamp> optional input is present in the sign request submitted by the client to  
204 the server, and it is supported by the server, the <PreviousTimestamp> element contained in  
205 this optional input must also be present as an element of the resulting timestamp generated by

206 the server and returned to the client. For XML timestamps of type `<ds:signature>`, processing  
207 rules are described in Section 3.2.3.

208 Before submitting the sign request, the client must verify that the `<PreviousTimestamp>`  
209 element corresponds to the document(s) being re-timestamped, and the client should verify the  
210 `<PreviousTimestamp>` element.

211 Note: Legitimate reasons to renew a timestamp include (a) the public key certificate used to verify  
212 the digital signature in the timestamp is nearing its expiration date, or (b) the client needs to  
213 replace the hash value used for the timestamped data in the existing timestamp with a hash value  
214 using a stronger hash algorithm.

### 215 **3.1.2 Element `<InputDocuments>`**

216 The client MAY send any component of `<dss:InputDocument>` element as input document. The  
217 extraction and processing of these elements MUST be carried out as indicated in the core  
218 document, with the changes mentioned in the present document.

219 If the client is not sending the `<dss:SignatureType>` optional input, then the client SHOULD only  
220 send a single input document, since some types of time-stamps (e.g. RFC 3161) can only cover  
221 one document per time-stamp.

222 If the client is sending the `<dss:SignatureType>` optional input, then the client MAY send multiple  
223 input documents, if the client knows that the specified time-stamp type can handle them.

## 224 **3.2 Element `<SignResponse>`**

### 225 **3.2.1 Element `<Result>`**

226 This profile defines no additional `<ResultMinor>` codes.

### 227 **3.2.2 Element `<OptionalOutputs>`**

228 The server MUST NOT return any optional outputs.

### 229 **3.2.3 Element `<SignatureObject>`**

230 The server MUST return a `<dss:Timestamp>` signature object.

231 If the `<RenewTimestamp>` optional input is present in the sign request submitted by the client to  
232 the server, and it is supported by the server, the `<PreviousTimestamp>` element contained in  
233 this optional input must also be present as an element of the resulting timestamp generated by  
234 the server and returned to the client. Specifically, for XML processing rules for XML timestamps  
235 of type `<ds:signature>`, the server must include the `<PreviousTimestamp>` element  
236 contained in the optional input as a child of an additional `<ds:Signature>/<ds:Object>` in  
237 the newly generated timestamp (i.e. in addition to the `<ds:object>` containing the  
238 `<TstInfo>`). An additional `<ds:SignedInfo>/<ds:Reference>` referencing the  
239 `<ds:Object>/<dss:PreviousTimestamp>` must be included in the signature of the new  
240 timestamp signature.

241 The server generating the new timestamp in response to a request carrying the  
242 <RenewTimestamp> optional input need make no assertions about the validity of the  
243 <PreviousTimestamp> element submitted to it within this optional input.  
244 A server that does not support the <RenewTimestamp> optional input must reject the sign  
245 request with a <ResultMajor> code of RequesterError and a <ResultMinor> code  
246 urn:oasis:names:tc:dss:1.0:resultminor:NotSupported.

247

---

## 4 Profile of Verifying Protocol

248

### 4.1 Element <VerifyRequest>

249

#### 4.1.1 Element <OptionalInputs>

250

The client may submit the <UseVerificationTime> optional input to instruct the server to determine the timestamp's validity at the specified time, instead of the current time. No other optional inputs are supported.

251

252

253

#### 4.1.2 Element <SignatureObject>

254

The client MUST send a <dss:Timestamp> signature object.

255

256

257

258

259

260

261

Note: A timestamp  $T_2$  that was generated by a server in response to a renewal request for timestamp  $T_1$ , that is, in response to a sign request on the same data as for timestamp  $T_1$  and carrying timestamp  $T_1$  within the <PreviousTimestamp> element of the <RenewTimestamp> optional input, may be used to assert current time validity for timestamp  $T_1$ . This situation applies when timestamp  $T_1$ 's current time validity can no longer be asserted independently, for example, because the cryptographic primitives in timestamp  $T_1$  are considered compromised. Specifically, the client may:

262

263

264

265

- submit a verify request for timestamp  $T_2$ ,
- submit a verify request for timestamp  $T_1$  and include the optional input <UseVerificationTime> with a value set to the issue time of timestamp  $T_2$  (i.e. using element <SpecificTime>).

266

267

268

269

270

271

If the result codes in the server verify responses indicate that both timestamps are valid as requested, the client may assert that timestamp  $T_1$  is currently valid, as supported by the fact that timestamp  $T_1$  is considered valid at the issue time of timestamp  $T_2$ , and timestamp  $T_2$  is considered valid currently. This process may be generalized to timestamps that were generated after multiple renewal requests on the same data, that is, timestamp  $T_1$ , renewed by timestamp  $T_2$ , renewed by timestamp  $T_3$ , and so on.

272

#### 4.1.3 Element <InputDocuments>

273

274

275

The client MAY send any component of <dss:InputDocuments> element as input documents. The extraction and processing of these elements MUST be carried out as indicated in the core document, with the changes mentioned in the present document.

276

### 4.2 Element <VerifyResponse>

277

#### 4.2.1 Element <Result>

278

This profile defines no additional <dss:ResultMinor> codes.

279 **4.2.2 Element <OptionalOutputs>**

280 The server MUST return the `<dss:SigningTimeInfo>` optional output.

281 **4.2.2.1 Element <SigningTimeInfo>**

282 The server MUST return this optional output profiled as detailed below:

- 283 1. Its `<dss:SigningTime>` child will contain the time indicated in the timestamp token (the  
284 value in `<dss:CreationTime>` element of DSS XML timestamps or the `genTime` field  
285 in RFC 3161 timestamp tokens).
- 286 2. If the timestamp token verified includes an indication of the deviation around the time  
287 present in the timestamp token (like the `accuracy` field in RFC 3161 timestamps or the  
288 `<dss:ErrorBound>` element in DSS XML timestamps), its  
289 `<dss:SigningTimeBoundaries>` child MUST be present and it MUST contain the  
290 lower and the upper boundaries suitably computed within its children.

291 The server MUST NOT return any other optional outputs.

---

292 **A. Acknowledgements**

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

295 **Participants:**

296 Dimitri Andivahis, Surety  
297 Frederick Hirsch, Nokia  
298 Pieter Kasselmann, Betrusted  
299 Andreas Kuehne, individual  
300 Paul Madsen, Entrust  
301 John Messing, American Bar Association  
302 Tim Moses, Entrust  
303 Nick Pope, Thales eSecurity  
304 Rich Salz, DataPower  
305 Ed Shallow, Universal Postal Union  
306