



# OASIS Digital Signature Services Version 1.0 Profile for Comprehensive Multi-Signature Verification Reports Version 1.0

**Committee Draft 02**

**27 September 2010**

**Specification URIs:**

**This Version:**

<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd02.html>  
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd02.doc>  
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd02.pdf>  
(Authoritative)

**Previous Version:**

<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd01.html>  
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd01.doc>  
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd01.pdf>  
(Authoritative)

**Latest Version:**

<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr.html>  
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr.doc>  
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr.pdf>  
(Authoritative)

**Technical Committee:**

OASIS Digital Signature Services eXtended (DSS-X) TC

**Chair(s):**

Juan Carlos Cruellas, *UPC-DAC* <[cruellas@ac.upc.edu](mailto:cruellas@ac.upc.edu)>  
Stefan Drees, *Individual Member*, <[stefan@drees.name](mailto:stefan@drees.name)>.

**Editor(s):**

Detlef Hühnlein, *Federal Office for Information Security, Germany* <[detlef.huehnlein@ecsec.de](mailto:detlef.huehnlein@ecsec.de)>

**Related work:**

This specification is based on

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

and may be combined with other existing profiles, such as

- [oasis-dss-profiles-AdES-v1.0-os](#)
- [oasis-dss-profiles-german\\_signature\\_law-spec-v1.0-os](#)

for example.

**Declared XML Namespace(s):**

urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#

**Abstract:**

This document defines a protocol and processing profile of the DSS Verifying Protocol specified in Section 4 of **[DSSCore]**, which allows to return individual signature verification reports for each signature in a verification request and include detailed information of the different steps taken during verification.

**Status:**

This document was last revised or approved by the Digital Signature Services Extended (DSS-X) TC 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/dss-x/>.

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/dss-x/ipr.php>)

---

## Notices

Copyright © OASIS® 2010. 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" and "DSS" 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

OASIS Digital Signature Services eXtended (DSS-X) TC .....	1
1 Introduction .....	5
1.1 Terminology.....	5
1.2 Normative References.....	5
1.3 Namespaces.....	6
2 Profile Features.....	8
2.1 Overview.....	8
2.2 Scope .....	8
2.3 Relationship To Other Profiles.....	8
2.4 Profile Identifier.....	8
3 Verification Reports within DSS Verifying Protocol .....	9
3.1 Element <ReturnVerificationReport> .....	9
3.2 Element <VerificationReport>.....	10
3.3 Element <IndividualReport> .....	11
3.4 VerificationResultType.....	13
3.5 Element <DetailedSignatureReport> .....	14
3.5.1 SignatureValidityType .....	14
3.5.2 AlgorithmValidityType.....	15
3.5.3 CertificatePathValidityType .....	15
3.5.4 PropertiesType .....	26
3.5.5 Element <IndividualTimeStampReport>.....	37
3.5.6 Element <IndividualCertificateReport>.....	37
3.5.7 Element <IndividualAttributeCertificateReport>.....	37
3.5.8 Element <IndividualCRLReport> .....	37
3.5.9 Element <IndividualOCSPReport> .....	37
3.5.10 Element <EvidenceRecordReport>.....	37
4 Conformance.....	41
4.1 Level 1 – “Basic” .....	41
4.2 Level 2 – “Comprehensive” .....	41
4.3 Level 3 – “Convenient” .....	42
A. Acknowledgements .....	43
B. Revision History .....	44

---

# 1 Introduction

This document defines a protocol and processing profile of the DSS Verifying Protocol specified in Section 4 of [DSSCore], which allows to support the verification of multiple signatures within some <VerifyRequest> and include detailed information of the different steps taken during verification. The following sections describe how to understand the rest of this 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].

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

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

## 1.2 Normative References

- [CAAdES] ETSI: “*Electronic Signature Formats*”, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V1.7.4, 2008-07
- [Core-XSD] S. Drees, T. Perrin, J. C. Cruellas, N. Pope, K. Lanz, et al.: “*DSS Schema*”, February 2007 <http://docs.oasis-open.org/dss/v1.0/DSS-XML-SCHEMAS-v1.0-os/oasis-dss-core-schema-v1.0-os.xsd>
- [DSSCore] OASIS Standard, *Digital Signature Service Core Protocols and Elements*, April 2007 <http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf>
- [DSSAdES] OASIS Standard, *Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0*, April 2007 <http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf>
- [DSSSigG] OASIS Standard, *German Signature Law Profile of the OASIS Digital Signature Service Version 1.0*, April 2007 [http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles\\_german\\_signature\\_law-spec-v1.0-os.pdf](http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles_german_signature_law-spec-v1.0-os.pdf)
- [DSSVR-XSD] D. Hühnlein, I. Henkel, J. C. Cruellas, S. Drees, A. Kuehne, et. al.: “*DSS Verification Report Schema*”, July 2009 <http://www.oasis-open.org/committees/download.php/33059/VerificationReport-CD1.xsd>
- [DSSVisSig] OASIS Committee Draft 01, *Visual Signature Profile of the OASIS Digital Signature Services*, April 2009 <http://docs.oasis-open.org/dss-x/profiles/visualseg/v1.0/cd01/oasis-dssx-1.0-profiles-visualseg-cd1.pdf>
- [EC/1999/93] *Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures* (<http://europa.eu.int/eurlex/pri/en/oj/dat/2000/l/013/l/01320000119en00120020.pdf>)
- [ETSI102231-2.1.1] ETSI: “*Provision of harmonized Trust-service status information*”, Electronic Signatures and Infrastructure (ESI) – Technical Specification, ETSI TS 102231 Version 2.1.1 of March 2006
- [ETSI102231-3.1.2] ETSI: “*Provision of harmonized Trust-service status information*”, Electronic Signatures and Infrastructure (ESI) – Technical Specification, ETSI TS 102231, Version 3.1.2 of December 2009 (<http://uri.etsi.org/02231/v3.1.2/>)

45	<b>[RFC2119]</b>	S. Bradner: "Key words for use in RFCs to Indicate Requirement Levels", IETF RFC 2119 ( <a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a> )
46		
47	<b>[RFC2560]</b>	M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: "X.509 Internet Public Key Infrastructure - Online Certificate Status Protocol – OCSP", IETF RFC 2560
48		
49		( <a href="http://www.ietf.org/rfc/rfc3161.txt">http://www.ietf.org/rfc/rfc3161.txt</a> )
50	<b>[RFC3161]</b>	C. Adams, P. Cain, D. Pinkas, R. Zuccherato: "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", IETF RFC 3161
51		
52		( <a href="http://www.ietf.org/rfc/rfc3161.txt">http://www.ietf.org/rfc/rfc3161.txt</a> )
53	<b>[RFC3275]</b>	D. Eastlage, J. Reagle, D. Solo: "(Extensible Markup Language) XML Signature Syntax and Processing", IETF RFC 3275 ( <a href="http://www.ietf.org/rfc/rfc3275.txt">http://www.ietf.org/rfc/rfc3275.txt</a> )
54		
55	<b>[RFC3281]</b>	S. Farrell, R. Housley: "An Internet Attribute Certificate Profile for Authorization", IETF RFC 3281 ( <a href="http://www.ietf.org/rfc/rfc3281.txt">http://www.ietf.org/rfc/rfc3281.txt</a> )
56		
57	<b>[RFC3852]</b>	R. Housley: "Cryptographic Message Syntax (CMS)". IETF RFC 3852,
58		( <a href="http://www.ietf.org/rfc/rfc3852.txt">http://www.ietf.org/rfc/rfc3852.txt</a> )
59	<b>[RFC4514]</b>	K. Zeilenga, Ed.: "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", IETF RFC 4514
60		
61		( <a href="http://www.ietf.org/rfc/rfc4514.txt">http://www.ietf.org/rfc/rfc4514.txt</a> )
62	<b>[RFC4998]</b>	T. Gondrom, R. Brandner, U. Pordesch: "Evidence Record Syntax (ERS)", IETF RFC 4998 ( <a href="http://www.ietf.org/rfc/rfc4998.txt">http://www.ietf.org/rfc/rfc4998.txt</a> )
63		
64	<b>[RFC5280]</b>	D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: "Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile", IETF RFC 5280 ( <a href="http://www.ietf.org/rfc/rfc5280.txt">http://www.ietf.org/rfc/rfc5280.txt</a> )
65		
66		
67	<b>[SAMLCore1.1]</b>	OASIS Standard, <i>Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V 1.1</i> , September 2003 <a href="http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf">http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf</a>
68		
69		
70	<b>[SAMLCore2.0]</b>	OASIS Standard, <i>Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0</i> , March 2005 <a href="http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf">http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf</a>
71		
72		
73	<b>[XAdES]</b>	ETSI: "XML Advanced Electronic Signatures (XAdES)", ETSI TS 101 903, Version 1.3.2, March 2006
74		
75	<b>[XML-ns]</b>	T. Bray, D. Hollander, A. Layman: "Namespaces in XML", W3C Recommendation, January 1999 ( <a href="http://www.w3.org/TR/1999/REC-xml-names-19990114">http://www.w3.org/TR/1999/REC-xml-names-19990114</a> )
76		
77		
78	<b>[XMLSig]</b>	D. Eastlake et al. "XML-Signature Syntax and Processing", W3C Recommendation, June 2008 ( <a href="http://www.w3.org/TR/xmlsig-core/">http://www.w3.org/TR/xmlsig-core/</a> )
79		

## 80 1.3 Namespaces

81 The structures described in this specification are contained in the schema file **[DSSVR-XSD]**. All schema  
 82 listings in the current document are excerpts from the schema file. In the case of a disagreement between  
 83 the schema file and this document, the schema file takes precedence.

84 This schema is associated with the following XML namespace:

85 `urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#`

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

87

88 Conventional XML namespace prefixes are used in this document:

- 89 • The prefix `vr:` (or no prefix) stands for this profiles namespace **[DSSVR-XSD]**.
- 90 • The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.
- 91 • The prefix `dss:` stands for the DSS core namespace **[Core-XSD]**.

- 92 • The prefix `saml`: stands for the OASIS SAML Schema namespace [**SAMLCore1.1**].
- 93 • The prefix `xades`: stands for ETSI XML Advanced Electronic Signatures (XAdES) document
- 94 [**XAdES**].

95

96 Applications MAY use different namespace prefixes, and MAY use whatever namespace

97 defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces in XML

98 specification [**XML-ns**].

99

---

## 100 2 Profile Features

### 101 2.1 Overview

102 While the DSS Verifying Protocol specified in Section 4 of **[DSSCore]** allows to verify digital signatures  
103 and time stamps, this protocol is fairly limited with respect to the verification of multiple signatures in a  
104 single request (cf. Section 4.3.1 of **[DSSCore]**).

105 In a similar manner it is possible to request and provide processing details (cf. Section 4.5.5 of  
106 **[DSSCore]**), but this simple mechanism does not support the verification of multiple signatures in a single  
107 request and there are no defined structures yet, which reflect the necessary steps in the verification of a  
108 complex signature, like an advanced electronic signature according to the European Directive  
109 **[EC/1999/93]** for example.

110 Therefore the present profile defines how

- 111 • individual verification results may be returned, if multiple signatures are part of a  
112 `<dss:VerifyRequest>` and
- 113 • detailed information gathered in the various steps taken during verification may be included in the  
114 response to form a comprehensive verification report.

115 The requester MAY request the activation of this profile by sending a `<ReturnVerificationReport>`  
116 element (cf. Section 3.1) in `<dss:OptionalInputs>`. A responder, which conforms to the present  
117 profile SHALL return a `<VerificationReport>` element (cf. Section 3.2) in  
118 `<dss:OptionalOutputs>`.

### 119 2.2 Scope

120 This document profiles the DSS Verifying Protocol (cf. **[DSSCore]**, Section 4).

121 It does *not* profile the DSS Signing Protocol (cf. **[DSSCore]**, Section 3) and does *neither specify nor*  
122 constrain

- 123 • the type of signature object,
- 124 • the transport binding or
- 125 • the security binding.

### 126 2.3 Relationship To Other Profiles

127 This profile is based directly on the **[DSSCore]**. This profile is intended to be combined with other profiles  
128 freely.

### 129 2.4 Profile Identifier

130 The DSS-client MAY use the following identifier in the `Protocol` attribute of a `VerifyRequest`:

131 `urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport`

132 The DSS-server MAY use this identifier in the `VerifyResponse`.



---

## 3 Verification Reports within DSS Verifying Protocol

133  
134

### 3.1 Element <ReturnVerificationReport>

135  
136  
137  
138

The <ReturnVerificationReport>-element is an optional input for the DSS Verifying Protocol to request an individual report for each signature. It is defined as follows:

139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156

```
<element name="ReturnVerificationReport">
  <complexType>
    <sequence>
      <element name="IncludeVerifier" type="boolean"
        maxOccurs="1" minOccurs="0" default="true" />
      <element name="IncludeCertificateValues" type="boolean"
        maxOccurs="1" minOccurs="0" default="false" />
      <element name="IncludeRevocationValues" type="boolean"
        maxOccurs="1" minOccurs="0" default="false" />
      <element name="ExpandBinaryValues" type="boolean"
        maxOccurs="1" minOccurs="0" default="false"/>
      <element name="ReportDetailLevel" type="anyURI"
        maxOccurs="1" minOccurs="0"
        default="urn:oasis:names:tc:dss:1.0:profiles:
        verificationreport:reportdetail:allDetails" />
    </sequence>
  </complexType>
</element>
```

157

It contains the following elements:

158  
159

<IncludeVerifier> [Default]

160  
161

This option specifies, whether the identity of the verifier should be included into the report or not. This is especially useful when (possibly time stamped) reports are archived. It defaults to 'true'.

162

<IncludeCertificateValues> [Default]

163  
164

With this option it is possible to include the certificate values, which are used to verify the signature (in binary form or as equivalent XML structure) into the report. This option defaults to 'false'.

165

<IncludeRevocationValues> [Default]

166  
167

This option specifies, whether the used revocation values (OCSP responses, CRLs and TSLs) should be included (in binary form or as equivalent XML structure) into the report or not. It defaults to 'false'.

168

<ExpandBinaryValues> [Default]

169  
170  
171

If this element is set to true a server which fulfills the conformance level "Convenient" MUST include the content of certificates and revocation information not only as ASN.1-coded binary values into the verification report, but also as equivalent XML structures. This option defaults to 'false'.

172

<ReportDetailLevel> [Optional]

173

This option specifies the detail level of the verification report. The following options are defined:

174  
175

- [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:noDetails](#)  
For every signature only the final result of the verification is reported.

176

- [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:noPathDetails](#)

177  
178  
179

Additionally to the final result also the details of the signature verification including the result of the certificate path validation are reported. The details concerning the validation of individual certificates in the path are omitted however.

- 180 – [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:allDetails](#)  
181 For every signature, the certificate path details and details on the validation of individual  
182 certificates in the path are requested. For every signature, the certificate path and each individual  
183 certificate the details are reported. If the <ReportDetailLevel>-element is missing, this  
184 option is assumed as default.

### 185 3.2 Element <VerificationReport>

186 If the element <ReturnVerificationReport> is provided as optional input in the request, the server  
187 MUST include in the response the element <VerificationReport> as optional output:

188

189

```
<element name="VerificationReport" type="vr:VerificationReportType" />
```

190

191 The **VerificationReportType** is the base structure for verification reports defined by this profile. It is  
192 defined as follows:

193

194

195

196

197

198

199

200

201

202

203

```
<complexType name="VerificationReportType">  
  <sequence>  
    <element ref="dss:VerificationTimeInfo" maxOccurs="1"  
      minOccurs="0" />  
    <element name="VerifierIdentity" type="vr:IdentifierType"  
      maxOccurs="1" minOccurs="0" />  
    <element name="IndividualReport" maxOccurs="unbounded"  
      type="vr:IndividualReportType" minOccurs="0" />  
  </sequence>  
</complexType>
```

204

205 It contains the following elements:

206 <VerificationTimeInfo> [Optional]

207 This element MAY contain the verification time, which was used by the server and other relevant time  
208 instants.

209 <VerifierIdentity> [Optional]

210 This element contains the identity of the verifier, if the report option <IncludeVerifier> was set to  
211 'true'. It is of type **vr:IdentifierType**, which is defined below.

212 <IndividualReport> [Optional, Unbounded]

213 For each independent<sup>1</sup> signed object (signature, time stamp, certificate, CRL, OCSP-response,  
214 evidence record etc.) that has been used in the signature verification process there will be one

---

<sup>1</sup> A signed object *x* is called independent of another signed object *y*, if *x* was produced and can be verified without *y* and *y* was produced and can be verified without *x*.

If a time stamp, certificate, CRL, OCSP-response etc. is included as unsigned attribute or property in an advanced electronic signature it is not independent of the signature for example.

215 <IndividualReport>-element in the verification report. The details of this element are specified in  
216 the following section.

217 The **IdentifierType** MAY contain different types of identifiers. It is defined as follows:

218

```
219 <complexType name="IdentifierType">  
220 <sequence>  
221 <element ref="ds:X509Data" maxOccurs="1" minOccurs="0" />  
222 <element name="SAMLv1Identifier" type="saml:NameIdentifierType"  
223 maxOccurs="1" minOccurs="0" />  
224 <element name="SAMLv2Identifier" type="saml2:NameIDType"  
225 maxOccurs="1" minOccurs="0" />  
226 <element name="Other" type="dss:AnyType" maxOccurs="1"  
227 minOccurs="0" />  
228 </sequence>  
229 </complexType>
```

230

231 It MAY contain the following elements or other identifying information:

232 <ds:X509Data> [Optional]

233 This element contains, if present, an X.509-certificate or certificate related information. Please refer to  
234 **[RFC3275]** for further details with respect to the ds:X509Data-element.

235 <SAMLv1Identifier> [Optional]

236 This element contains, if present, an identifier of type **saml:NameIdentifierType** as defined in  
237 **[SAMLCore1.1]**.

238 <SAMLv2Identifier> [Optional]

239 This element contains, if present, an identifier of type **saml2:NameIDType** as defined in  
240 **[SAMLCore2.0]**.

241 <Other> [Optional]

242 This element MAY contain, if present, other identifying information.

243

### 244 **3.3 Element <IndividualReport>**

245

246 The element <IndividualReport> is part of the <VerificationReport>-element (see Section 3.2)  
247 and is of type **IndividualReportType**, which is defined as follows:

248

```
249 <complexType name="IndividualReportType">  
250 <sequence>  
251 <element name="SignedObjectIdentifier"  
252 type="vr:SignedObjectIdentifierType"/>  
253 <element ref="dss:Result"/>  
254 <element name="Details" type="dss:AnyType" maxOccurs="1"  
255 minOccurs="0" />  
256 </sequence>  
257 </complexType>
```

258

259 It contains the following elements:

260 <SignedObjectIdentifier> [Required]

261 This element identifies the signature or validation data under consideration. The details of the  
262 SignedObjectIdentifierType are specified below.

263 <Result> [Required]

264 The result of the signature verification as defined in section 2.6 of [DSSCore].

265 <Details> [Optional]

266 The <Details> element MAY contain a detailed report for the signature or validation data under  
267 consideration or any other signature-specific optional output defined in Section 4.5 of [DSSCore].  
268 The corresponding elements, which are specified in this document for this purpose are listed in  
269 Section 4.2.

270

271 The **SignedObjectIdentifierType** is defined as follows:

272

```
273 <complexType name="SignedObjectIdentifierType">  
274   <sequence>  
275     <element name="DigestAlgAndValue"  
276       type="XAdES:DigestAlgAndValueType" maxOccurs="1" minOccurs="0"/>  
277     <element ref="ds:CanonicalizationMethod" maxOccurs="1" minOccurs="0" />  
278     <element name="SignedProperties"  
279       type="vr:SignedPropertiesType" maxOccurs="1" minOccurs="0" />  
280     <element ref="ds:SignatureValue" maxOccurs="1" minOccurs="0" />  
281     <element name="Other" type="dss:AnyType" maxOccurs="1" minOccurs="0" />  
282   </sequence>  
283   <attribute name="WhichDocument" type="IDREF" use="optional"/>  
284   <attribute name="XPath" type="string" use="optional"/>  
285   <attribute name="Offset" type="integer" use="optional"/>  
286   <attribute name="FieldName" type="string" use="optional"/>  
287 </complexType>
```

288

289 The set of child elements of the **SignedObjectIdentifierType** SHOULD be chosen to identify the  
290 signature or validation data in a given context in an unambiguous manner.

291 It contains the following attributes and elements:

292 <DigestAlgAndValue> [Optional]

293 This element contains, if present, the hash value of the signature or validation data under  
294 consideration, where the signed object itself (e.g. the <ds:Signature>-element in case of an XML-  
295 signature according to [RFC3275], the SignedData-structure in case of a CMS-signature according  
296 to [RFC3852] or a time stamp according to [RFC3161], the Certificate- or CertificateList-  
297 structure in case of an X.509-certificate or CRL according to [RFC5280] or the OCSPResponse-  
298 structure in case of an OCSP-response according to [RFC2560] for example) serves as input for the  
299 hash-calculation. The structure of the DigestAlgAndValueType is defined in [XAdES]. This  
300 element SHOULD NOT be used if the unique identification can be guaranteed by other elements.

301 <ds:CanonicalizationMethod> [Optional]

302 This element indicates, if present, the canonicalization method to be used before hashing XML-  
303 formatted data. Please refer to [RFC3275] for details of this element. This element is only necessary if  
304 XML-based structures are subject to hashing.

305 <SignedProperties> [Optional]

306 This element contains, if present, any number of signed properties, which may be useful to identify the  
307 signature under consideration. This MAY comprise information about the signatory and the signing  
308 time for example. The structure of the SignedPropertiesType is defined in Section 3.5.4.2. In case  
309 of signatures according to [RFC3275] or [RFC3852] this element SHOULD be present.

310 <ds:SignatureValue> [Optional]

311 This element specifies, if present, the binary signature value of the signature under consideration. This  
312 element SHOULD be present – particularly if the used signature algorithm is randomized and hence  
313 this element may serve as unique identifier.

314 <Other> [Optional]  
 315 This element MAY contain other elements, which (help to) identify a signature or related validation  
 316 data in a unique manner.

317 WhichDocument [Optional]  
 318 This attribute MAY specify the document which contains the signature under consideration. Note that  
 319 this identifier is only unique with respect to a specific request message (see [DSSCore], Section  
 320 2.4.1).

321 XPath [Optional]  
 322 This attribute MAY be used to point to a specific signature within an XML document.

323 Offset [Optional]  
 324 This attribute specifies the first byte of some signature and MAY be used to point to a specific  
 325 signature within some binary document.

326 FieldName [Optional]  
 327 This attribute specifies the name of a signature field and MAY be used to point to a specific signature  
 328 within some document format, in which there are field names such as PDF for example.

### 329 3.4 VerificationResultType

330 The **VerificationResultType** defined below is extensively used in the present profile to indicate the  
 331 success or failure of individual verification steps.

332 This type draws from the `dss:Result`-element and the **dss:DetailType** defined in [DSSCore] and is  
 333 defined as follows:

```

334 <complexType name="VerificationResultType">
335   <sequence>
336     <element name="ResultMajor" type="anyURI"/>
337     <element name="ResultMinor" type="anyURI" minOccurs="0"/>
338     <element name="ResultMessage" type="dss:InternationalStringType"
339       minOccurs="0"/>
340     <any namespace="##other" processContents="lax" minOccurs="0"
341       maxOccurs="unbounded"/>
342   </sequence>
343 </complexType>
  
```

344

345 <ResultMajor> [Required]  
 346 This element MUST indicate whether the verification result is valid, invalid or indetermined using the  
 347 URIs defined in [DSSCore]:

- 348 • urn:oasis:names:tc:dss:1.0:detail:valid
- 349 • urn:oasis:names:tc:dss:1.0:detail:invalid
- 350 • urn:oasis:names:tc:dss:1.0:detail:indetermined

351 <ResultMinor> [Optional]  
 352 In case of an invalid or indetermined verification step, further details MAY be provided using a specific  
 353 URI defined in this document or other profiles.

354 <ResultMessage> [Optional]  
 355 Especially in case of an invalid or indetermined verification step, further details MAY be provided in  
 356 textual form.

357 Furthermore an element of type **VerificationResultType** MAY contain other elements.

## 358 3.5 Element <DetailedSignatureReport>

359 The <DetailedSignatureReport>-element MAY appear in the <Details>-element within the  
360 <IndividualReport>-element, which is specified in Section 3.3 above. This element is defined as  
361 follows:

```
362 <element name="DetailedSignatureReport"  
363         type="vr:DetailedSignatureReportType" />
```

364  
365 The **DetailedSignatureReportType** in turn is specified as follows:  
366

```
367 <complexType name="DetailedSignatureReportType">  
368   <sequence>  
369     <element name="FormatOK" type="vr:VerificationResultType" />  
370     <element name="Properties" type="vr:PropertiesType"  
371             maxOccurs="1" minOccurs="0" />  
372     <element ref="dss:VerifyManifestResults" maxOccurs="1"  
373             minOccurs="0" />  
374     <element name="SignatureHasVisibleContent" type="boolean"  
375             maxOccurs="1" minOccurs="0"/>  
376     <element name="SignatureOK"  
377             type="vr:SignatureValidityType" />  
378     <element name="CertificatePathValidity"  
379             type="vr:CertificatePathValidityType" />  
380   </sequence>  
381 </complexType>
```

382  
383 It contains the following elements:

384 <FormatOK> [Required]

385 This element indicates, whether the format of the signature is ok or not. More information on the use of  
386 the **VerificationResultType** may be found in Section 3.4.

387 <Properties> [Optional]

388 This element contains information gathered during the verification of signed or unsigned properties.  
389 The structure of the **PropertiesType** is defined in Section 3.5.4.

390 <VerifyManifestResults> [Optional]

391 This element is present, if a manifest verification has been performed. The structure and the  
392 semantics of this element is described in Section 4.5.1 of **[DSSCore]**.

393 <SignatureHasVisibleContent> [Optional]

394 This element is only present if the `FieldName`-attribute (cf. Section 3.3) is present and indicates  
395 whether the signature under consideration has visual signature content as explained in **[DSSVisSig]**.

396 <SignatureOK> [Required]

397 This element contains information about the mathematical validity of the digital signature under  
398 consideration. It is of type **SignatureValidityType**, which is specified in Section 3.5.1.

399 <CertificatePathValidity> [Required]

400 This element contains the results of the certificate path validation. The **CertificatePathValidityType** is  
401 defined in section 3.5.3.

### 402 3.5.1 SignatureValidityType

403 The **SignatureValidityType** is used in the definition of the <DetailedSignatureReport>-element  
404 above for example and it is specified as follows:

405

```
406 <complexType name="SignatureValidityType">
407   <sequence>
408     <element name="SigMathOK" type="vr:VerificationResultType" />
409     <element name="SignatureAlgorithm"
410       type="vr:AlgorithmValidityType"
411       maxOccurs="1" minOccurs="0"/>
412   </sequence>
413 </complexType>
```

414

415 It comprises the following elements:

416 <SigMathOK> [Required]

417 Contains information about the mathematical validity of the digital signature under consideration, More  
418 information on the use of the **VerificationResultType** may be found in Section 3.4.

419 <SignatureAlgorithm> [Optional]

420 This element MAY contain information about the applied signature algorithm. It is of type  
421 **AlgorithmValidityType**, which is defined below.

422

### 423 3.5.2 AlgorithmValidityType

424 The **AlgorithmValidityType** is used in the definition of the **SignatureValidityType** above for example  
425 and is specified as follows:

426

```
427 <complexType name="AlgorithmValidityType">
428   <sequence>
429     <element name="Algorithm" type="anyURI" />
430     <element name="Parameters" type="dss:AnyType" maxOccurs="1"
431       minOccurs="0" />
432     <element name="Suitability" type="vr:VerificationResultType"
433       maxOccurs="1" minOccurs="0"/>
434   </sequence>
435 </complexType>
```

436

437 <Algorithm> [Required]

438 This element contains the URI for the algorithm.

439 <Parameters> [Optional]

440 This element MAY contain further parameters for the cryptographic algorithm.

441 <Suitability> [Optional]

442 This element MAY contain the information about the suitability of the algorithm under consideration.  
443 Note that it MAY depend on the policy of the specific signature and/or the policy under which the DSS  
444 server is operated, whether the suitability of the algorithms is verified and what kind of algorithms are  
445 considered appropriate under given circumstances and which are not. More information on the use of  
446 the **VerificationResultType** may be found in Section 3.4.

### 447 3.5.3 CertificatePathValidityType

448 The <CertificatePathValidity>-element is of type **CertificatePathValidityType** and is used in the  
449 definition of

- 450 • **DetailedSignatureReportType** (see above),
- 451 • **AttributeCertificateValidityType** (see Section 3.5.4.3),

- 452 • **CRLValidityType** (see Section 3.5.3.4),
- 453 • **OCSPValidityType** (see Section 3.5.3.5) and
- 454 • **TimeStampValidityType** (see Section 3.5.4.4).

455

456 It is specified as follows:

457

```

458 <complexType name="CertificatePathValidityType">
459   <sequence>
460     <element name="PathValiditySummary"
461       type="vr:VerificationResultType" />
462     <element name="CertificateIdentifier"
463       type="ds:X509IssuerSerialType" />
464     <element name="PathValidityDetail"
465       type="vr:CertificatePathValidityDetailType"
466       minOccurs="0" maxOccurs="1"/>
467   </sequence>
468 </complexType>

```

469

470 It contains the following elements:

471 <PathValiditySummary> [Required]

472 This element is of type **VerificationResultType** (see Section 3.4) and contains a summary of the  
 473 result of the certificate path validation.

474 <CertificateIdentifier> [Required]

475 This element is of type **ds:X509IssuerSerialType** (see Section 4.4.4 of [RFC3275]) and contains a  
 476 unique reference to the certificate whose path has been checked.

477 <PathValidityDetail> [Optional]

478 Contains detailed results of the certificate path validation, if the element <ReportDetailLevel> in  
 479 the report options (see Section 3.1) was set to [urn:oasis:names:tc:dss:1.0:  
 480 profiles:verificationreport:reportdetail:allDetails](urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails) and the detailed validity information has not been  
 481 included elsewhere in the verification report.

482

483 The structure of **CertificatePathValidityDetailType** is specified as follows:

484

```

485 <complexType name="CertificatePathValidityDetailType">
486   <sequence>
487     <sequence maxOccurs="unbounded" minOccurs="0">
488       <element name="CertificateValidity"
489         type="vr:CertificateValidityType" />
490     </sequence>
491     <element name="TSLValidity"
492       type="dss:AnyType" maxOccurs="1" minOccurs="0" />
493     <element name="TrustAnchor" type="vr:VerificationResultType" />
494   </sequence>
495 </complexType>

```

496

497 It contains the following elements:

498 <CertificateValidity> [Optional, Unbounded]

499 For every certificate in the certificate path there will be a <CertificateValidity>-element, which  
 500 provides information about the validity of the specific certificate. The structure of the  
 501 **CertificateValidityType** is defined below.



502 <TSLValidity> [Optional]

503 This element contains information about the validity of a Trust-service Status List (TSL) according to  
504 [ETSI102231-2.1.1] or [ETSI102231-3.1.2] for example. This element SHOULD contain information  
505 about

- 506 • the TSL-scheme under consideration, as provided by a `SchemeInformation` element,
- 507 • the Trust-service providers and their services, as provided by a  
508 `TrustServiceProviderList` element,
- 509 • the measures for protecting the integrity and authenticity of the TSL-related information and  
510 the result of the corresponding verification step. If the integrity and authenticity is protected by  
511 means of an electronic signature, it is RECOMMENDED to include a  
512 `DetailedSignatureReport` element. If the integrity is protected by a time stamp it is  
513 RECOMMENDED to include an `IndividualTimeStamp` element etc. .

514 <TrustAnchor> [Required]

515 This element indicates how the trusted root certificate, which is used as trust anchor within the  
516 verification process, is stored. The following URIs are defined for this purpose:

- 517 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:SSCD](#) – indicates that the  
518 trusted root certificate is stored within a secure signature creation device according to  
519 [EC/1999/93].
- 520 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:otherCard](#) – indicates that  
521 the trusted root certificate is stored within some other hardware token.
- 522 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:certDataBase](#) – indicates  
523 that the trusted root certificate is stored within some certificate data base.
- 524 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:other](#) – indicates that the  
525 trusted root certificate is stored using other means.

526

### 527 3.5.3.1 CertificateValidityType

528

529 The **CertificateValidityType** contains information about the validity of a single certificate and is defined  
530 as follows:

531

```
532 <complexType name="CertificateValidityType">  
533   <sequence>  
534     <element name="CertificateIdentifier" type="ds:X509IssuerSerialType" />  
535     <element name="Subject" type="string" />  
536     <element name="ChainingOK" type="vr:VerificationResultType"  
537       maxOccurs="1" minOccurs="0"/>  
538     <element name="ValidityPeriodOK" type="vr:VerificationResultType" />  
539     <element name="ExtensionsOK" type="vr:VerificationResultType" />  
540     <element name="CertificateValue" type="base64Binary"  
541       maxOccurs="1" minOccurs="0" />  
542     <element name="CertificateContent"  
543       type="vr:CertificateContentType" maxOccurs="1" minOccurs="0" />  
544     <element name="SignatureOK"  
545       type="vr:SignatureValidityType" />  
546     <element name="CertificateStatus" type="vr:CertificateStatusType" />  
547   </sequence>  
548 </complexType>
```

549

550 It contains the following elements:

551 <CertificateIdentifier> [Required]

552 This element is of type **ds:X509IssuerSerialType** (see [RFC3275], Section 4.4.4) and identifies the  
553 certificate under consideration.

554 <Subject> [Required]

555 This element contains the subject of the certificate, where the string representation of distinguished  
556 names defined in [RFC4514] MUST be used and hence an example of a <Subject>-element may be  
557 CN=John Doe,O=Foo Inc.,OU=Sales etc.

558 <ChainingOK> [Optional]

559 If present, this element indicates whether the chaining to a previous certificate in the certificate path is  
560 ok or not. If the certificate under consideration is the first certificate in the certificate path, this element  
561 SHOULD be omitted. More information on the use of the **VerificationResultType** may be found in  
562 Section 3.4.

563 <ValidityPeriodOK> [Required]

564 This element indicates, whether the reference point in time is within the validity period of the  
565 certificate. More information on the use of the **VerificationResultType** may be found in Section 3.4.

566 <ExtensionsOK> [Required]

567 This element indicates, whether the certificate extensions are correct. More information on the use of  
568 the **VerificationResultType** may be found in Section 3.4.

569 <CertificateValue> [Optional]

570 If present, this element contains the certificate in binary form (coded in ASN.1), if the report option  
571 <IncludeCertificateValues> is set to 'true' and if the certificate is not already included in the  
572 verification report.

573 <CertificateContent> [Optional]

574 If present, this element contains detailed information about the content of the certificate, if the report  
575 option <ExpandBinaryValues> is set to 'true' and if the certificate content is not already included in  
576 the verification report.

577 <SignatureOK> [Required]

578 This element indicates, whether the digital signature of the certificate is mathematically correct or not.  
579 The **SignatureValidityType** is defined in section 3.5.1.

580 <CertificateStatus> [Required]

581 This element contains information about the result of the certificate revocation check. The  
582 **CertificateStatusType** is defined in Section 3.5.3.3.

583

### 584 3.5.3.2 CertificateContentType

585

586 The **CertificateContentType** is used in **CertificateValidityType** and derived from the  
587 TBSCertificate-structure defined in [RFC5280] specified as follows:

588

```
589 <complexType name="CertificateContentType">  
590   <sequence>  
591     <element name="Version" type="integer" maxOccurs="1"  
592       minOccurs="0" />  
593     <element name="SerialNumber" type="integer" />  
594     <element name="SignatureAlgorithm" type="anyURI" />  
595     <element name="Issuer" type="string" />  
596     <element name="ValidityPeriod" type="vr:ValidityPeriodType" />  
597     <element name="Subject" type="string" />  
598     <element name="Extensions" type="vr:ExtensionsType"  
599       minOccurs="0" />
```

```
600     </sequence>
601 </complexType>
```

602  
603 It contains the following elements:

604 <Version> [Optional]

605 This element contains, if present, the version of the certificate structure.

606 <SerialNumber> [Required]

607 This element MUST contain the serial number of the certificate.

608 <SignatureAlgorithm> [Required]

609 This element MUST contain an identifier of the used signature algorithm. The  
610 `vr:VerificationResultType` is defined in Section 3.4.

611 <Issuer> [Required]

612 This element MUST contain the issuer of the certificate, where different relative distinguished names  
613 in a sequence MAY be separated by “.”.

614 <ValidityPeriod> [Required]

615 This element MUST contain the validity period of the certificate. The **ValidityPeriodType** is defined  
616 below.

617 <Subject> [Required]

618 This element contains the subject of the certificate, where the string representation of distinguished  
619 names defined in **[RFC4514]** MUST be used and hence an example of a <Subject>-element may be  
620 `CN=John Doe,O=Foo Inc.,OU=Sales etc.`

621  
622 <Extensions> [Optional]

623 If present, this element contains information about the list of extensions present in the certificate under  
624 consideration. The **ExtensionsType** is defined below.

625  
626 The **ValidityPeriodType** is specified as follows:

```
627  
628 <complexType name="ValidityPeriodType">
629   <sequence>
630     <element name="NotBefore" type="dateTime" />
631     <element name="NotAfter" type="dateTime" />
632   </sequence>
633 </complexType>
```

634  
635 It contains the following elements:

636 <NotBefore> [Required]

637 The certificate is not valid before this point in time.

638 <NotAfter> [Required]

639 The certificate is not valid after this point in time.

640  
641 The **ExtensionsType** is specified as follows:

```
642  
643 <complexType name="ExtensionsType">
644   <sequence minOccurs="0" maxOccurs="unbounded">
```

```
645         <element name="Extension" type="vr:ExtensionType" />
646     </sequence>
647 </complexType>
```

648  
649 It contains an unbounded number <Extension>-elements of type **ExtensionType**. This type is defined  
650 as follows:

```
651  
652 <complexType name="ExtensionType">
653     <sequence>
654         <element name="ExtnId" type="XAdES:ObjectIdentifierType" />
655         <element name="Critical" type="boolean" />
656         <element name="ExtnValue" type="dss:AnyType" maxOccurs="1"
657             minOccurs="0" />
658         <element name="ExtensionOK" type="vr:VerificationResultType" />
659     </sequence>
660 </complexType>
```

661  
662 It contains the following elements:

663 <ExtnId> [Required]

664 This element MUST contain the identifier of the extension as urn:oid: ... in the <Identifier>-  
665 element and MAY contain further information in the <Description>- and  
666 <DocumentationReferences>-elements. Please refer to **[XAdES]** for more information on the  
667 **XAdES:ObjectIdentifierType**.

668 <Critical> [Required]

669 This element specifies, whether the extension is critical or not.

670

671 <ExtnValue> [Optional]

672 This element SHOULD contain the value of the extension as an XML-structure, which mirrors the  
673 original ASN.1-definition of the extension.

674 <ExtensionOK> [Required]

675 This element contains information about the validity of the specific extension within the given context  
676 of the certificate.

677

### 678 3.5.3.3 CertificateStatusType

679

680 The **CertificateStatusType** is defined as follows:

681

```
682 <complexType name="CertificateStatusType">
683     <sequence>
684         <element name="CertStatusOK" type="vr:VerificationResultType" />
685         <element name="RevocationInfo" maxOccurs="1"
686             minOccurs="0">
687             <complexType>
688                 <sequence>
689                     <element name="RevocationDate" type="dateTime" />
690                     <element name="RevocationReason"
691                         type="vr:VerificationResultType" />
692                 </sequence>
693             </complexType>
```

```

694     </element>
695     <element name="RevocationEvidence" maxOccurs="1" minOccurs="0">
696         <complexType>
697             <choice>
698                 <element name="CRLValidity"
699                     type="vr:CRLValidityType" />
700                 <element name="CRLReference"
701                     type="XAdES:CRLIdentifierType" />
702                 <element name="OCSPValidity"
703                     type="vr:OCSPValidityType" />
704                 <element name="OCSPReference"
705                     type="XAdES:OCSPIdentifierType" />
706                 <element name="Other" type="dss:AnyType"/>
707             </choice>
708         </complexType>
709     </element>
710 </sequence>
711 </complexType>

```

712

713 It contains the following elements:

714 <CertStatusOK> [Required]

715 This element MUST contain the status of the certificate.

716 <RevocationInfo> [Optional]

717 If the certificate is revoked this element will contain more information about the revocation. It is defined  
718 to be a sequence, which contains the following elements:

- 719 • <RevocationDate>  
720 contains the date and time of revocation.
- 721 • <RevocationReason>  
722 contains the reason for revocation. Following the definition of CRLReason in **[RFC5280]** there are  
723 the following URIs to specify the revocation reason:
  - 724 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:unspecified>
  - 725 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:keyCompromise>
  - 726 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:cACompromise>
  - 727 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:affiliationChanged>
  - 728 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:superseded>
  - 729 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:cessationOfOperation>
  - 731 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:certificateHold>
  - 732 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:removeFromCRL>
  - 733 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:privilegeWithdrawn>
  - 734 • <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:aACompromise>

735 <RevocationEvidence> [Optional, Choice]

736 This element contains, if present, the used source of revocation information. This can be one of the  
737 following elements:

- 738 • <CRLValidity>  
739 This element contains information about the used CRL and its validity. The **CRLValidityType** is  
740 defined in Section 3.5.3.4.

- 741 • <CRLReference>  
742 This element contains a reference to the CRL in case it is already included elsewhere in the  
743 present verification report. The **XAdES:CRLIdentifierType** is defined in [XAdES].
- 744 • <OCSPValidity>  
745 This element contains information about the used OCSP response and its validity. The  
746 **OCSPValidityType** is defined in Section 3.5.3.5.
- 747 • <OCSPReference>  
748 This element contains a reference to the used OCSP response, if it is already included elsewhere  
749 in the present verification report. The **XAdES:OCSPIdentifierType** is defined in [XAdES].
- 750 • <Other>  
751 This element MAY contain information about alternative sources of revocation information.

### 752 3.5.3.4 CRLValidityType

753 The **CRLValidityType** contains information about a CRL and its validity and is specified as follows:

754

```

755 <complexType name="CRLValidityType">
756   <sequence>
757     <element name="CRLIdentifier" type="XAdES:CRLIdentifierType"
758       maxOccurs="1" minOccurs="1" />
759     <element name="CRLValue" type="base64Binary"
760       maxOccurs="1" minOccurs="0" />
761     <element name="CRLContent" type="vr:CRLContentType"
762       maxOccurs="1" minOccurs="0" />
763     <element name="SignatureOK" type="vr:SignatureValidityType" />
764     <element name="CertificatePathValidity"
765       type="vr:CertificatePathValidityType" />
766   </sequence>
767   <attribute name="Id" type="ID" use="optional" />
768 </complexType>

```

769

770 It contains the following attributes and elements:

771 Id [Optional]

772 This attribute contains an optional identifier for the element.

773 <CRLIdentifier> [Required]

774 This element refers to an X.509v2 CRL according to [RFC5280].

775 <CRLValue> [Optional]

776 If present, this element contains the CRL (encoded in ASN.1) if the report option  
777 <IncludeRevocationValues> is set to 'true'.

778 <CRLContent> [Optional]

779 This element contains, if present, the CRL in form of an equivalent XML structure if the report option  
780 <ExpandBinaryValues> is set to 'true'. The **CRLContentType** is defined below.

781 <SignatureOK> [Required]

782 This element indicates, whether the digital signature of the CRL is mathematically correct or not. The  
783 **SignatureValidityType** is defined in section 3.5.1.

784 <CertificatePathValidity> [Required]

785 This element contains the result of the validation of the certificate path of the certificate which has  
786 been used to sign the CRL. The **CertificatePathValidityType** is defined at the beginning of Section  
787 3.5.3.

788

789 The **CRLContentType** is aligned to [RFC5280] specified as follows:

790

```
791 <complexType name="CRLContentType">
792   <sequence>
793     <element name="Version" minOccurs="0" type="integer" />
794     <element name="Signature" type="anyURI" />
795     <element name="Issuer" type="string" />
796     <element name="ThisUpdate" type="dateTime" />
797     <element name="NextUpdate" minOccurs="0" type="dateTime" />
798     <element name="RevokedCertificates" minOccurs="0">
799       <complexType>
800         <sequence minOccurs="0" maxOccurs="unbounded">
801           <element name="UserCertificate" type="integer" />
802           <element name="RevocationDate" type="dateTime" />
803           <element name="CrlEntryExtensions" minOccurs="0"
804             type="vr:ExtensionsType" />
805         </sequence>
806       </complexType>
807     </element>
808     <element name="CrlExtensions" type="vr:ExtensionsType"
809       minOccurs="0" />
810   </sequence>
811 </complexType>
```

812

813 It contains the following elements:

814 <Version> [Optional]

815 This element contains, if present, the version of the CRL-structure.

816 <Signature> [Required]

817 This element contains the algorithm identifier for the algorithm used to sign the CRL.

818 <Issuer> [Required]

819 This element contains the issuer of the CRL, where different relative distinguished names in a  
820 sequence MAY be separated by “.”.

821 <ThisUpdate> [Required]

822 This element contains the issue date of the CRL.

823 <NextUpdate> [Optional]

824 This element contains, if present, the date by which the next CRL will be issued.

825 <RevokedCertificates> [Optional]

826 The revoked certificates are contained in an unbounded sequence. They are listed by their serial  
827 numbers (element <UserCertificate>). Certificates revoked by the CA are uniquely identified by  
828 their certificate serial number. The date on which the revocation occurred is contained in the element  
829 <RevocationDate>. Additional information MAY be supplied in the element  
830 <CrlEntryExtensions>.

831 <CrlExtensions> [Optional]

832 If present, this element contains information about the list of extensions present in the CRL under  
833 consideration. The **ExtensionType** is defined in Section 3.5.3.2.

### 834 3.5.3.5 OCSPValidityType

835 The **OCSPValidityType** contains information about an OCSP-response and its validity and is specified as  
836 follows:

837

```

838     <complexType name="OCSPValidityType">
839         <sequence>
840             <element name="OCSPIdentifier" type="XAdES:OCSPIdentifierType" />
841             <element name="OCSPValue" type="base64Binary"
842                 maxOccurs="1" minOccurs="0" />
843             <element name="OCSPContent" type="vr:OCSPContentType"
844                 maxOccurs="1" minOccurs="0" />
845             <element name="SignatureOK" type="vr:SignatureValidityType" />
846             <element name="CertificatePathValidity"
847                 type="vr:CertificatePathValidityType" />
848         </sequence>
849         <attribute name="Id" type="ID" use="optional" />
850     </complexType>

```

851

852 It contains the following attributes and elements:

853 Id [Optional]

854 This attribute contains an optional identifier for the element.

855 <OCSPIdentifier> [Required]

856 This element refers to an OCSP response according to [RFC2560].

857 <OCSPValue> [Optional]

858 This element contains the OCSP response (encoded in ASN.1) if the report option  
859 <IncludeRevocationValues> has been set to 'true'.

860 <OCSPContent> [Optional]

861 This element contains the OCSP response in form of an equivalent XML structure if the report option  
862 <ExpandBinaryValues> has been set to 'true'. The **OCSPContentType** is defined below.

863 <SignatureOK> [Required]

864 This element indicates whether the digital signature of the OCSP-response is mathematically correct  
865 or not. The **SignatureValidityType** is defined in section 3.5.1.

866

867

868 <CertificatePathValidity> [Required]

869 This element contains the result of the validation of the certificate path of the certificate which has  
870 been used to sign the OCSP-response. The **CertificatePathValidityType** is defined at the beginning  
871 of Section 3.5.3.

872

873 The **OCSPContentType** is aligned to [RFC2560] specified as follows:

```

875 <complexType name="OCSPContentType">
876     <sequence>
877         <element name="Version" type="integer" />
878         <element name="ResponderID" type="string" />
879         <element name="producedAt" type="dateTime" />
880         <element name="Responses">
881             <complexType>
882                 <sequence maxOccurs="unbounded" minOccurs="0">
883                     <element name="SingleResponse"
884                         type="vr:SingleResponseType" />
885                 </sequence>
886             </complexType>
887         </element>
888         <element name="ResponseExtensions" type="vr:ExtensionsType"
889             maxOccurs="1" minOccurs="0"/>

```



```
890     </sequence>
891 </complexType>
```

892  
893 It contains the following elements:

894 <Version> [Required]

895 This element contains the version of the OCSP-response syntax.

896 <ResponderID> [Required]

897 This element contains the name of the OCSP-responder.

898 <producedAt> [Required]

899 This element contains the time at which the OCSP-responder produced the response.

900 <Responses> [Required]

901 This element contains an unbounded sequence of <SingleResponse> entries. The  
902 **SingleResponseType** is defined below.

903 <ResponseExtensions> [Optional]

904 If present, this element contains information about the list of extensions present in the OCSP-response  
905 under consideration. The **ExtensionsType** is defined in Section 3.5.3.2.

906  
907 The **SingleResponseType** is specified as follows:

```
908  
909 <complexType name="SingleResponseType">
910   <sequence>
911     <element name="CertID">
912       <complexType>
913         <sequence>
914           <element name="HashAlgorithm" type="anyURI" />
915           <element name="IssuerNameHash" type="hexBinary" />
916           <element name="IssuerKeyHash" type="hexBinary" />
917           <element name="SerialNumber" type="integer" />
918         </sequence>
919       </complexType>
920     </element>
921     <element name="CertStatus" type="vr:VerificationResultType" />
922     <element name="ThisUpdate" type="dateTime" />
923     <element name="NextUpdate" type="dateTime" maxOccurs="1"
924       minOccurs="0" />
925     <element name="SingleExtensions" type="vr:ExtensionsType"
926       maxOccurs="1" minOccurs="0" />
927   </sequence>
928 </complexType>
```

929  
930 It contains the following elements:

931 <CertID> [Required]

932 This element contains a sequence of elements, which uniquely identify the certificate (cf. [RFC2560],  
933 Section 4.1.1).

934 <CertStatus> [Required]

935 This element contains information about the status of the certificate according to [RFC2560] using the  
936 following URI in the ResultMajor-element:

- 937
- <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:good>
  - <urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:revoked>
- 938

- [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:unknown](#)
- If the certificate is revoked and the revocation reason is present, this information **MUST** be included in the `ResultMinor`-element as a URI defined in Section 3.5.3.4. In a similar fashion the revocation time **MUST** be indicated in the `ResultMessage`-element.

943 <ThisUpdate> [Required]

944 This element contains the time at which the status being indicated is known to be correct (cf.  
945 [RFC2560], Section 2.4).

946 <NextUpdate> [Optional]

947 This element contains, if present, the time until more recent information about the status of the  
948 certificate will be available (cf. [RFC2560], Section 2.4).

949 <SingleExtensions> [Optional]

950 If present, this element contains information about the list of extensions present in the  
951 `SingleResponse`-element. The **ExtensionType** is defined in Section 3.5.3.2.

## 952 3.5.4 PropertiesType

953 The **PropertiesType** is used in the definition of the `<DetailedReport>`-element (see Section 3.5) and  
954 is specified as follows:

955

```
956 <complexType name="PropertiesType">  
957   <sequence>  
958     <element name="SignedProperties"  
959       type="vr:SignedPropertiesType" minOccurs="0" />  
960     <element name="UnsignedProperties"  
961       type="vr:UnsignedPropertiesType" minOccurs="0" />  
962   </sequence>  
963   <attribute name="Id" type="ID" use="optional" />  
964 </complexType>
```

965

966 It contains the following attributes and elements:

967 Id [Optional]

968 This attribute contains, if present, an optional identifier for the element.

969 <SignedProperties> [Optional]

970 This element contains information gathered during the verification of signed properties. Details of the  
971 `SignedPropertiesType` are specified in Section 3.5.4.1.

972 <UnsignedProperties> [Optional]

973 This element contains information gathered during the verification of unsigned properties. Details of  
974 the `UnsignedPropertiesType` are specified in Section 3.5.4.2.

### 975 3.5.4.1 Signed Properties

976 The **SignedPropertiesType** is aligned to [XAdES] structured as follows:

977

```
978 <complexType name="SignedPropertiesType">  
979   <sequence>  
980     <element name="SignedSignatureProperties"  
981       type="vr:SignedSignaturePropertiesType" maxOccurs="1"  
982       minOccurs="0" />  
983     <element name="SignedDataObjectProperties"  
984       type="vr:SignedDataObjectPropertiesType"  
985       minOccurs="0" />
```

```

986         <element name="Other" type="dss:AnyType"
987             maxOccurs="1" minOccurs="0" />
988     </sequence>
989     <attribute name="Id" type="ID" use="optional" />
990 </complexType>

```

991  
992 It contains the following attributes and elements:

993 Id [Optional]

994 This attribute contains an optional identifier for the element.

995 <SignedSignatureProperties> [Optional]

996 This element contains information gathered during the verification of signed properties related to the  
997 signature itself. The **SignedSignaturePropertiesType** is defined in Section 3.5.4.1.1.

998 <SignedDataObjectProperties> [Optional]

999 This element contains information gathered during the verification of signed properties related to the  
1000 signed data object. The **SignedDataObjectPropertiesType** is defined in Section 3.5.4.1.2.

1001 <Other> [Optional]

1002 This element contains, if present, information about other signed properties.

### 1003 3.5.4.1.1 SignedSignaturePropertiesType

1004 The **SignedSignaturePropertiesType** is aligned to [RFC3275] defined as follows:

1005

```

1006 <complexType name="SignedSignaturePropertiesType">
1007 <sequence>
1008     <element ref="XAdES:SigningTime" maxOccurs="1" minOccurs="0" />
1009     <element ref="XAdES:SigningCertificate" maxOccurs="1"
1010         minOccurs="0" />
1011     <element ref="XAdES:SignaturePolicyIdentifier" maxOccurs="1"
1012         minOccurs="0" />
1013     <choice maxOccurs="1" minOccurs="0">
1014         <element ref="XAdES:SignatureProductionPlace" />
1015         <element name="Location" type="string" />
1016     </choice>
1017     <element name="SignerRole" type="vr:SignerRoleType" minOccurs="0" />
1018 </sequence>
1019 </complexType>

```

1020

1021 It MAY contain the following elements:

1022 <XAdES:SigningTime> [Optional]

1023 This element contains, if present, the signing time (see Section 5.2.1 of [XAdES]).

1024 <XAdES:SigningCertificate> [Optional]

1025 This element contains, if present, a reference to the certificate upon which the signature is based (see  
1026 Section 5.2.2 of [XAdES]).

1027 <XAdES:SignaturePolicyIdentifier> [Optional]

1028 This element references, if present, the policy under which the signature was produced (see Section  
1029 5.2.3 of [XAdES]).

1030 <XAdES:SignatureProductionPlace> [Optional, Choice]

1031 This element contains, if present, information about the place where the signature was generated (see  
1032 Section 5.2.7 of [XAdES]). This element SHOULD be used in case of a XAdES- or CAdES-based  
1033 signature.

1034 <Location> [Optional, Choice]

1035 This element contains, if present, information about the place where the signature was generated (see  
1036 Section 5.2.7 of [XAdES]). This element SHOULD be used in case of a PDF-based signature.

1037 <SignerRole> [Optional]

1038 This element contains, if present, information about the role of the signer (see Section 5.2.8 of  
1039 [XAdES]).

1040

1041 The **SignerRoleType** is specified as follows:

1042

```
1043 <complexType name="SignerRoleType">  
1044   <sequence>  
1045     <element name="ClaimedRoles"  
1046       type="XAdES:ClaimedRolesListType" minOccurs="0" />  
1047     <element name="CertifiedRoles"  
1048       type="vr:CertifiedRolesListType" minOccurs="0" />  
1049   </sequence>  
1050 </complexType>
```

1051

1052 It MAY contain the following elements:

1053 <ClaimedRoles> [Optional]

1054 This element contains information about the claimed roles of the signer. The information is directly  
1055 extracted from the signature.

1056 <CertifiedRoles> [Optional]

1057 This element contains information gathered during the verification of attribute certificates.

1058

1059 The **CertifiedRolesListType** is specified as follows:

1060

```
1061 <complexType name="CertifiedRolesListType">  
1062   <sequence>  
1063     <element name="AttributeCertificateValidity"  
1064       type="vr:AttributeCertificateValidityType"  
1065       maxOccurs="unbounded" />  
1066   </sequence>  
1067 </complexType>
```

1068

1069 It contains at least one <AttributeCertificateValidity>-element, which contains information  
1070 about the content and validity of an attribute certificate according to [RFC3281]. The  
1071 **AttributeCertificateValidityType** is defined in Section 3.5.4.3.

### 1072 3.5.4.1.2 SignedDataObjectPropertiesType

1073 The **SignedDataObjectPropertiesType** is defined as follows:

1074

```
1075 <complexType name="SignedDataObjectPropertiesType">  
1076   <sequence>  
1077     <element ref="XAdES:DataObjectFormat" maxOccurs="unbounded"  
1078       minOccurs="0" />  
1079     <choice maxOccurs="1" minOccurs="0">  
1080       <element ref="XAdES:CommitmentTypeIndication"  
1081         maxOccurs="unbounded" minOccurs="1"/>  
1082       <element name="Reason" type="string" />
```

```

1083     </choice>
1084     <element name="AllDataObjectsTimeStamp"
1085           type="vr:TimeStampValidityType" minOccurs="0"
1086           maxOccurs="unbounded" />
1087     <element name="IndividualDataObjectsTimeStamp"
1088           type="vr:TimeStampValidityType" minOccurs="0"
1089           maxOccurs="unbounded" />
1090   </sequence>
1091   <attribute name="Id" type="ID" use="optional" />
1092 </complexType>

```

1093  
1094 It contains the following attributes and elements:

1095 Id [Optional]

1096 This attribute contains an optional identifier for the element.

1097 <XAdES:DataObjectFormat> [Optional, Unbounded]

1098 This element contains information about the format of the signed data object (see Section 5.2.5 of [XAdES]). This information is simply extracted from the signature.

1100 <XAdES:CommitmentTypeIndication> [Choice, Unbounded]

1101 This element contains, if present, an indication of the type of commitment implied by the signature (see Section 5.2.6 of [XAdES]). This element SHOULD be used in case of a XAdES- or CAdES-based signature.

1104 <Reason> [Choice]

1105 This element contains, if present, a description of the reason of the signature generation. This element is only relevant in case of a PDF-based signature identified by a `FieldName`-attribute (cf. Section 3.3).

1108 <AllDataObjectsTimeStamp> [Optional, Unbounded]

1109 This element contains, if present, verification results for time stamps covering all data objects (see Section 5.2.6 of [XAdES]). The **TimeStampValidityType** is described in Section 3.5.4.4.

1111 <IndividualDataObjectsTimeStamp> [Optional, Unbounded]

1112 This element contains, if present, verification results for time stamps covering only certain data objects (see Section 5.2.10 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.

### 1114 3.5.4.2 Unsigned Properties

1115 The **UnsignedPropertiesType** is specified as follows:

```

1117 <complexType name="UnsignedPropertiesType">
1118   <sequence>
1119     <element name="UnsignedSignatureProperties"
1120           type="vr:UnsignedSignaturePropertiesType" minOccurs="0" />
1121     <element ref="XAdES:UnsignedDataObjectProperties"
1122           maxOccurs="1" minOccurs="0" />
1123     <element name="Other" type="dss:AnyType" maxOccurs="1"
1124           minOccurs="0">
1125     </element>
1126   </sequence>
1127   <attribute name="Id" type="ID" use="optional" />
1128 </complexType>

```

1129  
1130 It contains the following attributes and elements:

1131 Id [Optional]

1132 This attribute contains an optional identifier for the element.

1133 <UnsignedSignatureProperties> [Optional]

1134 This element contains information gathered during the verification of the unsigned properties related to  
1135 the signature itself. The **UnsignedSignaturePropertiesType** is defined below.

1136 <XAdES:UnsignedDataObjectProperties> [Optional]

1137 This element contains unsigned properties referring to the signed data objects. These properties are  
1138 directly extracted from the signature.

1139 <Other> [Optional]

1140 This element MAY contain information about other unsigned properties.

1141

1142 The **UnsignedSignaturePropertiesType** is defined as follows:

1143

```
1144 <complexType name="UnsignedSignaturePropertiesType">  
1145   <choice maxOccurs="unbounded">  
1146     <element name="CounterSignature" type="vr:SignatureValidityType" />  
1147     <element name="SignatureTimeStamp" type="vr:TimeStampValidityType" />  
1148     <element ref="XAdES:CompleteCertificateRefs" />  
1149     <element ref="XAdES:CompleteRevocationRefs" />  
1150     <element ref="XAdES:AttributeCertificateRefs" />  
1151     <element ref="XAdES:AttributeRevocationRefs" />  
1152     <element name="SigAndRefsTimeStamp"  
1153       type="vr:TimeStampValidityType" />  
1154     <element name="RefsOnlyTimeStamp" type="vr:TimeStampValidityType" />  
1155     <element name="CertificateValues" type="vr:CertificateValuesType" />  
1156     <element name="RevocationValues" type="vr:RevocationValuesType" />  
1157     <element name="AttrAuthoritiesCertValues"  
1158       type="vr:CertificateValuesType" />  
1159     <element name="AttributeRevocationValues"  
1160       type="vr:RevocationValuesType" />  
1161     <element name="ArchiveTimeStamp" type="vr:TimeStampValidityType" />  
1162   </choice>  
1163   <attribute name="Id" type="ID" use="optional" />  
1164 </complexType>
```

1165

1166 It contains the following attributes and elements:

1167 Id [Optional]

1168 This attribute contains an optional identifier for the element.

1169 <CounterSignature> [Choice]

1170 This element contains the results of the verification of a counter signature (see Section 7.2.4 of  
1171 **[XAdES]**). The **SignatureValidityType** is described in section 3.5.1.

1172 <SignatureTimeStamp> [Choice]

1173 This element contains verification results of a time stamp of the signature (see Section 7.3 of  
1174 **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

1175 <XAdES:CompleteCertificateRefs> [Choice]

1176 This element contains references to the certificates used during verification of the signature (see  
1177 Section 7.4.1 of **[XAdES]**). This information is simply extracted from the signature.

1178 <XAdES:CompleteRevocationRefs> [Choice]

1179 Contains references to the revocation data used for the verification of the signature (see Section 7.4.2  
1180 of **[XAdES]**). This information is simply extracted from the signature.

1181 <XAdES:AttributeCertificateRefs> [Choice]

1182 Contains the references to the full set of attribute authorities certificates that have been used to  
1183 validate the attribute certificate (see section 7.4.3 of [XAdES]). This information is simply extracted  
1184 from the signature.

1185 <XAdES:AttributeRevocationRefs> [Choice]

1186 Contains the references to the full set of revocation data that have been used in the validation of the  
1187 attribute certificate(s) present in the signature (see section 7.4.4 of [XAdES]).

1188 <SigAndRefsTimeStamp> [Choice]

1189 Contains verification results for a time stamp referring to the signature and references on certificates  
1190 and revocation data (see section 7.5.1 of [XAdES]). The **TimeStampValidityType** is described in  
1191 section 3.5.4.4.

1192 <RefsOnlyTimeStamp> [Choice]

1193 Contains verification results for a time stamp referring only to references on certificates and revocation  
1194 data (see section 7.5.2 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.

1195 <CertificateValues> [Choice]

1196 Contains verification results for the certificates, which were used in the verification of the signature  
1197 (see section 7.6.1 of [XAdES]). The **CertificateValuesType** is defined below.

1198 <RevocationValues> [Choice]

1199 Contains verification results of the revocation data used in the verification of the signature (see section  
1200 7.6.2 of [XAdES]). The **RevocationValuesType** is defined below.

1201 <AttrAuthoritiesCertValues> [Choice]

1202 Contains verification results of the certificates of Attribute Authorities that have been used to validate  
1203 the attribute certificates, which are contained in the signature (see section 7.6.3 of [XAdES]). The  
1204 **CertificateValuesType** is defined below.

1205 <AttributeRevocationValues> [Choice]

1206 Contains verification results of the revocation data that have been used to validate the attribute  
1207 certificate when present in the signature (see section 7.6.4 of [XAdES]). The **RevocationValuesType**  
1208 is defined below.

1209 <ArchiveTimeStamp> [Choice]

1210 Contains verification results for a time stamp covering the complete signature including all attributes  
1211 (see section 7.7 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.

1212

1213 The **CertificateValuesType** is defined as follows:

1214

```
1215 <complexType name="CertificateValuesType">  
1216   <choice minOccurs="0" maxOccurs="unbounded">  
1217     <element name="EncapsulatedX509Certificate"  
1218       type="vr:CertificateValidityType" />  
1219     <element name="OtherCertificate" />  
1220   </choice>  
1221   <attribute name="Id" type="ID" use="optional" />  
1222 </complexType>
```

1223

1224 It defines the following attributes and elements:

1225 Id [Optional]

1226 This attribute contains an optional identifier for the element.

1227 <EncapsulatedX509Certificate> [Optional, Unbounded, Choice]

1228 Contains verification results for an X.509 certificate included in the signature. The  
1229 **CertificateValidityType** is defined in Section 3.5.3.1.

1230 <OtherCertificate> [Optional, Unbounded, Choice]

1231 This element contains verification results for other certificates included in the signature. If a certificate  
1232 with unknown format is included in the signature, a warning (error code  
1233 [urn:oasis:names:tc:dss:1.0:resultminor:certificateFormatNotCorrectWarning](#)) SHOULD be returned.

1234

1235 The **RevocationValuesType** is defined as follows:

1236

```
1237 <complexType name="RevocationValuesType">  
1238   <sequence>  
1239     <element name="CRLValues" minOccurs="0">  
1240       <complexType>  
1241         <sequence maxOccurs="unbounded" minOccurs="1">  
1242           <element name="VerifiedCRL"  
1243             type="vr:CRLValidityType" />  
1244         </sequence>  
1245       </complexType>  
1246     </element>  
1247     <element name="OCSPValues" minOccurs="0">  
1248       <complexType>  
1249         <sequence maxOccurs="unbounded" minOccurs="1">  
1250           <element name="VerifiedOCSPResponse"  
1251             type="vr:OCSPValidityType" />  
1252         </sequence>  
1253       </complexType>  
1254     </element>  
1255     <element name="OtherValues" type="dss:AnyType" minOccurs="0" />  
1256   </sequence>  
1257   <attribute name="Id" type="ID" use="optional" />  
1258 </complexType>
```

1259

1260 It contains the following attributes and elements:

1261 Id [Optional]

1262 This attribute contains an optional identifier for the element.

1263 <CRLValues> [Optional]

1264 Contains the verification results for all CRLs included in a signature. The **CRLValidityType** is defined  
1265 in Section 3.5.3.4.

1266 <OCSPValues> [Optional]

1267 Contains the verification results for all OCSP responses included in a signature. The  
1268 **OCSPValidityType** is defined in Section 3.5.3.5.

1269 <OtherValues> [Optional]

1270 This element MAY contain verification results for other revocation data included in the signature. If  
1271 other revocation data with unknown format is included in the signature, a warning (error  
1272 [urn:oasis:names:tc:dss:1.0:resultminor:improperRevocationInformation](#)) SHOULD be returned.

1273

### 1274 3.5.4.3 AttributeCertificateValidityType

1275 The **AttributeCertificateValidityType** is defined as follows:

1276

```
1277 <complexType name="AttributeCertificateValidityType">  
1278   <sequence>  
1279     <element name="AttributeCertificateIdentifier"  
1280       type="vr:AttrCertIDType" maxOccurs="1" minOccurs="0" />
```



```

1281         <element name="AttributeCertificateValue" type="base64Binary"
1282             maxOccurs="1" minOccurs="0" />
1283         <element name="AttributeCertificateContent"
1284             type="vr:AttributeCertificateContentType" maxOccurs="1"
1285             minOccurs="0" />
1286         <element name="SignatureOK" type="vr:SignatureValidityType" />
1287         <element name="CertificatePathValidity"
1288             type="vr:CertificatePathValidityType" />
1289     </sequence>
1290 </complexType>

```

1291

1292 It contains the following elements:

1293 <AttributeCertificateIdentifier> [Optional]

1294 This element MAY refer to an X.509v3 attribute certificate according to [RFC3281]. The structure of  
1295 the **AttrCertIDType** is defined below.

1296 <AttributeCertificateValue> [Optional]

1297 This element MAY contain the certificate in binary form (coded in ASN.1), if the report option  
1298 <IncludeCertificateValues> is set to 'true'.

1299 <AttributeCertificateContent> [Optional]

1300 This element MAY contain an XML-based analogue of the content of the certificate, if the report option  
1301 <ExpandBinaryValues> is set to 'true'. The structure of the  
1302 AttributeCertificateContentType is defined below.

1303 <SignatureOK> [Required]

1304 This element indicates, whether the digital signature is mathematically valid or not. The  
1305 **SignatureValidityType** is defined in section 3.5.1.

1306 <CertificatePathValidity> [Required]

1307 This element contains the result of the validation of the certificate path of the certificate which has  
1308 been used to sign the attribute certificate. The **CertificatePathValidityType** is defined at the  
1309 beginning of Section 3.5.3.

1310

1311 The **AttrCertIDType** is structured as follows:

1312

```

1313 <complexType name="AttrCertIDType">
1314     <sequence>
1315         <element name="Holder" type="vr:EntityType" maxOccurs="1"
1316             minOccurs="0"/>
1317         <element name="Issuer" type="vr:EntityType" />
1318         <element name="SerialNumber" type="integer" />
1319     </sequence>
1320 </complexType>

```

1321

1322 It contains the following elements:

1323 <Holder> [Optional]

1324 This element contains, if present, information about the holder of the certificate. The structure of the  
1325 **EntityType** is defined below.

1326 <Issuer> [Required]

1327 This element contains information about the issuer of the attribute certificate. The structure of the  
1328 **EntityType** is defined below.

1329 <SerialNumber> [Required]

1330 This element contains the serial number of the attribute certificate, which (together with the information  
1331 provided in the <Issuer>-element) uniquely identifies the attribute certificate.

1332

1333 The **EntityType** is aligned to the structure of Holder and V2Form in [RFC3281] and is defined as  
1334 follows:

1335

```
1336 <complexType name="EntityType">
1337   <sequence>
1338     <element name="BaseCertificateID"
1339       type="ds:X509IssuerSerialType" maxOccurs="1"
1340       minOccurs="0"/>
1341     <element name="Name" type="string" maxOccurs="1"
1342       minOccurs="0"/>
1343     <element name="Other" type="dss:AnyType" maxOccurs="1"
1344       minOccurs="0"/>
1345   </sequence>
1346 </complexType>
```

1347

1348 It SHOULD contain sufficient information to identify the entity uniquely and MAY contain the following  
1349 optional elements:

1350 <BaseCertificateID> [Optional]

1351 This element identifies, if present, the public-key certificate of the entity. The structure of the  
1352 ds:X509IssuerSerialType is defined in [RFC3275].

1353 <Name> [Optional]

1354 This element contains, if present, the name of the entity.

1355 <Other> [Optional]

1356 This element MAY contain other information, which is used to identify the entity.

1357

1358 The **AttributeCertificateContentType** contains the content of an attribute certificate according to  
1359 [RFC3281] as XML structure and is structured as follows:

1360

```
1361 <complexType name="AttributeCertificateContentType">
1362   <sequence>
1363     <element name="Version" minOccurs="0" type="integer" />
1364     <element name="Holder" type="vr:EntityType" />
1365     <element name="Issuer" type="vr:EntityType" />
1366     <element name="SignatureAlgorithm" type="anyURI" />
1367     <element name="SerialNumber" type="integer" />
1368     <element name="AttCertValidityPeriod"
1369       type="vr:ValidityType" />
1370     <element name="Attributes">
1371       <complexType>
1372         <sequence minOccurs="0" maxOccurs="unbounded">
1373           <element name="Attribute"
1374             type="vr:AttributeType" />
1375         </sequence>
1376       </complexType>
1377     </element>
```

```

1378         <element name="IssuerUniqueID" type="hexBinary" maxOccurs="1"
1379             minOccurs="0"/>
1380         <element name="Extensions" minOccurs="0"
1381             type="vr:ExtensionsType" />
1382     </sequence>
1383 </complexType>

```

1384

1385 It contains the following elements:

1386 <Version> [Optional]

1387 This element contains, if present, the version of the attribute certificate.

1388 <Holder> [Required]

1389 This element contains information about the holder of the certificate. The structure of the **EntityType**  
1390 is defined above.

1391 <Issuer> [Required]

1392 This element contains the issuer of the attribute certificate. The structure of the **EntityType** is defined  
1393 above.

1394 <SignatureAlgorithm> [Required]

1395 This element contains an identifier of the used signature algorithm.

1396 <SerialNumber> [Required]

1397 This element contains the serial number of the attribute certificate.

1398 <AttCertValidityPeriod> [Required]

1399 This element contains the validity period of the attribute certificate. The **ValidityType** is defined in  
1400 section 3.5.3.2.

1401 <Attributes> [Optional, Unbounded]

1402 This element contains, if present, a list of attributes. The **AttributeType** is defined below.

1403 <IssuerUniqueID> [Optional]

1404 This element contains, if present, a unique identifier of the issuer of the attribute certificate.

1405 <Extensions> [Optional]

1406 If present, this element contains information about the list of extensions present in the attribute  
1407 certificate. The **ExtensionType** is defined in Section 3.5.3.2.

1408

1409 The **AttributeType** is defined as follows:

1410

```

1411 <complexType name="AttributeType">
1412     <sequence>
1413         <element name="Type" type="anyURI" />
1414         <element name="Value" type="dss:AnyType" maxOccurs="unbounded"
1415             minOccurs="0" />
1416     </sequence>
1417 </complexType>

```

1418

1419 It contains the following elements:

1420 <Type> [Required]

1421 This element MUST contain an identifier for the type of the attribute in the <Code>-element and MAY  
1422 contain further information.

1423 <Value> [Optional, Unbounded]

1424 This element MAY contain any number of attribute values.

1425

### 1426 3.5.4.4 TimeStampValidityType

1427 The **TimeStampValidityType** is structured as follows:

1428

```
1429 <complexType name="TimeStampValidityType">
1430   <sequence>
1431     <element name="FormatOK" type="vr:VerificationResultType" />
1432     <element name="TimeStampContent" type="vr:TstContentType"
1433       maxOccurs="1" minOccurs="0" />
1434     <element name="MessageHashAlgorithm"
1435       type="vr:AlgorithmValidityType"
1436       maxOccurs="1" minOccurs="0" />
1437     <element name="SignatureOK"
1438       type="vr:SignatureValidityType" />
1439     <element name="CertificatePathValidity"
1440       type="vr:CertificatePathValidityType" />
1441   </sequence>
1442   <attribute name="Id" type="ID" use="optional" />
1443 </complexType>
```

1444

1445 It contains the following elements and attributes:

1446 Id [Optional]

1447 This attribute contains an optional identifier for the element.

1448 <FormatOK> [Required]

1449 This element indicates, whether the format of the time stamp is ok or not. More information on the use  
1450 of the **VerificationResultType** may be found in Section 3.4.

1451 <TimeStampContent> [Optional]

1452 This element contains the content of time stamp in form of an XML structure, if the report option  
1453 <ExpandBinaryValues> is set to 'true'. The **TstContentType** is specified below.

1454 <MessageHashAlgorithm> [Optional]

1455 This element contains, if present, information about the message hash algorithm and its suitability.  
1456 The **AlgorithmValidityType** is defined in Section 3.5.2.

1457 <SignatureOK> [Required]

1458 This element indicates, whether the digital signature is mathematically valid or not. The  
1459 **SignatureValidityType** is defined in Section 3.5.1.

1460 <CertificatePathValidity> [Required]

1461 This element contains the result of the validity check of the certificate. The  
1462 **CertificatePathValidityType** is defined in Section 3.5.3.

1463

1464 The **TstContentType** complex type is defined as follows:

1465

```
1466 <complexType name="TstContentType">
1467   <sequence>
1468     <element ref="dss:TstInfo" maxOccurs="1" minOccurs="0"/>
1469     <element name="Other" type="dss:AnyType" maxOccurs="1"
1470       minOccurs="0"/>
1471   </sequence>
1472 </complexType>
```

1473 It contains the following elements:

1474 `<dss:TstInfo>` [Optional]

1475 This element MAY contain the standard content of a time stamp as defined in Section 5.1.2 of  
1476 **[DSSCore]**. Note that there is a straightforward mapping from the `TSTInfo`-Element according to  
1477 **[RFC3161]** to the present structure.

1478 `<Other>` [Optional]

1479 This element MAY contain other information included in the time stamp.

### 1480 **3.5.5 Element `<IndividualTimeStampReport>`**

1481 The `<IndividualTimeStampReport>`-element MAY appear in the `<Details>`-element within the  
1482 `<IndividualReport>`-element defined in Section 3.3. This element is defined as follows:

```
1483 <element name="IndividualTimeStampReport" type="vr:TimeStampValidityType"/>
```

1484 The **TimeStampValidityType** is defined in Section 3.5.4.4.

### 1485 **3.5.6 Element `<IndividualCertificateReport>`**

1486 The `<IndividualCertificateReport>`-element MAY appear in the `<Details>`-element within the  
1487 `<IndividualReport>`-element defined in Section 3.3. This element is defined as follows:

```
1488 <element name="IndividualCertificateReport"  
1489 type="vr:CertificateValidityType" />
```

1490 The **CertificateValidityType** is defined in Section 3.5.3.1.

### 1491 **3.5.7 Element `<IndividualAttributeCertificateReport>`**

1492 The `<IndividualAttributeCertificateReport>`-element MAY appear in the `<Details>`-  
1493 element within the `<IndividualReport>`-element defined in Section 3.3. This element is defined as  
1494 follows:

```
1495 <element name="IndividualAttributeCertificateReport"  
1496 type="vr:AttributeCertificateValidityType" />
```

1497 The **AttributeCertificateValidityType** is defined in Section 3.5.4.3.

### 1498 **3.5.8 Element `<IndividualCRLReport>`**

1499 The `<IndividualCRLReport>`-element MAY appear in the `<Details>`-element within the  
1500 `<IndividualReport>`-element defined in Section 3.3. This element is defined as follows:

```
1501 <element name="IndividualCRLReport" type="vr:CRLValidityType" />
```

1502 The **CRLValidityType** is defined in Section 3.5.3.4.

### 1503 **3.5.9 Element `<IndividualOCSPReport>`**

1504 The `<IndividualOCSPReport>`-element MAY appear in the `<Details>`-element within the  
1505 `<IndividualReport>`-element defined in Section 3.3. This element is defined as follows:

```
1506 <element name="IndividualOCSPReport" type="vr:OCSPValidityType" />
```

1507 The **OCSPValidityType** is defined in Section 3.5.3.5.

### 1508 **3.5.10 Element `<EvidenceRecordReport>`**

1509 The `<EvidenceRecordReport>`-element MAY appear in the `<Details>`-element within the  
1510 `<IndividualReport>`-element defined in Section 3.3. This element is defined as follows:

1511

```
<element name="EvidenceRecordReport" type="vr:EvidenceRecordValidityType" />
```

1512 The **EvidenceRecordValidityType** is based on the definition of the EvidenceRecord-element in  
1513 **[RFC4998]** defined as follows:

```
1514 <complexType name="EvidenceRecordValidityType">
1515   <sequence>
1516     <element name="FormatOK" type="vr:VerificationResultType" />
1517     <element name="Version" type="integer"
1518       maxOccurs="1" minOccurs="0" />
1519     <element name="DigestAlgorithm"
1520       type="vr:AlgorithmValidityType" maxOccurs="unbounded"
1521       minOccurs="0">
1522     </element>
1523     <element name="CryptoInfos" maxOccurs="1" minOccurs="0">
1524       <complexType>
1525         <sequence>
1526           <element name="Attribute" type="vr:AttributeType"
1527             maxOccurs="unbounded" minOccurs="1" />
1528         </sequence>
1529       </complexType>
1530     </element>
1531     <element name="EncryptionInfo" maxOccurs="1" minOccurs="0">
1532       <complexType>
1533         <sequence>
1534           <element name="EncryptionInfoType"
1535             type="vr:AlgorithmValidityType" />
1536           <element name="EncryptionInfoValue"
1537             type="dss:AnyType" />
1538         </sequence>
1539       </complexType>
1540     </element>
1541     <element name="ArchiveTimeStampSequence" maxOccurs="1"
1542       minOccurs="1">
1543       <complexType>
1544         <sequence maxOccurs="unbounded" minOccurs="0">
1545           <element name="ArchiveTimeStampChain">
1546             <complexType>
1547               <sequence maxOccurs="unbounded"
1548                 minOccurs="0">
1549                 <element name="ArchiveTimeStamp"
1550                   type="vr:ArchiveTimeStampValidityType"/>
1551               </sequence>
1552             </complexType>
1553           </element>
1554         </sequence>
1555       </complexType>
1556     </element>
1557   </sequence>
1558   <attribute name="Id" type="ID" use="optional" />
1559 </complexType>
```

1560

1561 It contains the following elements and attributes:

1562 Id [Optional]

1563 This attribute contains an optional identifier for the element.

1564 <FormatOK> [Required]

1565 This element indicates, whether the format of the evidence record according to **[RFC4998]** is ok or  
1566 not. More information on the use of the **VerificationResultType** may be found in Section 3.4.

1567 <Version> [Optional]

1568 This element contains, if present, the version of the Evidence Record Syntax.

1569 <DigestAlgorithm> [Optional, unbounded]

1570 This element appears for each hash algorithm used to produce the evidence record and contains  
 1571 information about the hash algorithm and possibly its suitability. The **AlgorithmValidityType** is  
 1572 defined in Section 3.5.2.

1573 <CryptoInfos> [Optional]

1574 This element MAY contain further data useful in the validation of the <ArchiveTimeStampSequence>-  
 1575 element. As explained in [RFC4998] this MAY include possible Trust Anchors, certificates, revocation  
 1576 information, or the information concerning the suitability of cryptographic algorithms.

1577 <EncryptionInfo> [Optional]

1578 This element MAY contain the necessary information to support encrypted content (cf. [RFC4998],  
 1579 Section 6.1).

1580 <ArchiveTimeStampSequence> [Required]

1581 This element is required and MAY contain a sequence of <ArchiveTimeStampChain>-elements (cf.  
 1582 [RFC4998], Section 5), which in turn MAY contain a sequence of <ArchiveTimeStamp>-elements,  
 1583 which are of type **ArchiveTimeStampValidityType** defined below.

1584

1585 The **ArchiveTimeStampValidityType** is based on the definition of the ArchiveTimeStamp-element in  
 1586 [RFC4998] defined as follows:

1587

```

1588 <complexType name="ArchiveTimeStampValidityType">
1589   <sequence>
1590     <element name="FormatOK" type="vr:VerificationResultType" />
1591     <element name="DigestAlgorithm" type="vr:AlgorithmValidityType"
1592       maxOccurs="1" minOccurs="0" />
1593     <element name="Attributes" maxOccurs="1" minOccurs="0">
1594       <complexType>
1595         <sequence>
1596           <element name="Attribute" type="vr:AttributeType"
1597             maxOccurs="unbounded" minOccurs="1"/>
1598         </sequence>
1599       </complexType>
1600     </element>
1601     <element name="ReducedHashTree" maxOccurs="1" minOccurs="0">
1602       <complexType>
1603         <sequence maxOccurs="unbounded" minOccurs="1">
1604           <element name="PartialHashTree">
1605             <complexType>
1606               <sequence maxOccurs="unbounded"
1607                 minOccurs="1">
1608                 <element name="HashValue"
1609                   type="vr:HashValueType"/>
1610               </sequence>
1611             </complexType>
1612           </element>
1613         </sequence>
1614       </complexType>
1615     </element>
1616     <element name="TimeStamp"
1617       type="vr:TimeStampValidityType" />
1618   </sequence>
1619   <attribute name="Id" type="ID" use="optional" />
1620 </complexType>

```

1621

1622 It contains the following elements and attributes:

- 1623 Id [Optional]
- 1624 This attribute contains an optional identifier for the element.
- 1625 <FormatOK> [Required]
- 1626 This element indicates, whether the format of the evidence record according to **[RFC4998]** is ok or  
1627 not. More information on the use of the **VerificationResultType** may be found in Section 3.4.
- 1628 <DigestAlgorithm> [Optional]
- 1629 This element contains, if present, information about the hash algorithm and possibly its suitability. The  
1630 **AlgorithmValidityType** is defined in Section 3.5.2.
- 1631 <Attributes> [Optional]
- 1632 This element contains, if present, information about further attributes related to the archive time  
1633 stamp.
- 1634 <ReducedHashTree> [Optional]
- 1635 This element MAY contain a sequence of <PartialHashTree>-elements, which in turn contain a  
1636 list of <HashValue>-elements of type **HashValueType** defined below.
- 1637 <TimeStamp> [Required]
- 1638 This element is of type **TimeStampValidityType** (cf. Section 3.5.4.4) and contains information about  
1639 the validity of the conventional time stamp, which is included in the present archive time stamp.

1640

1641 The **HashValueType** is used for the <HashValue>-element within the <PartialHashTree>-element  
1642 above and is defined as follows:

```
1643 <complexType name="HashValueType">
1644   <sequence>
1645     <element name="HashValue" type="hexBinary" />
1646   </sequence>
1647   <attribute name="HashedObject" type="IDREF" use="optional"/>
1648 </complexType>
```

1649 It contains the following elements and attributes:

- 1650 HashedObject [Optional]
- 1651 This attribute MAY be used to point to the object, which served as pre-image of the hash value.
- 1652 <HashValue> [Required]
- 1653 This element contains the hash value produced by applying the hash algorithm specified by the  
1654 <DigestAlgorithm>- or <TimeStamp>-element to the data specified by the HashedObject  
1655 attribute.
- 1656



---

## 1657 4 Conformance

1658 This profile defines three conformance levels:

- 1659 • Level 1 - “Basic”,
- 1660 • Level 2 - “Comprehensive” and
- 1661 • Level 3 - “Comfortable”.

### 1662 4.1 Level 1 – “Basic”

1663 The conformance level “Basic” allows to return individual verification results for each signature contained  
1664 in a `<dss:VerifyRequest>`. For this purpose the `<dss:VerifyResponse>` MUST contain in  
1665 `<dss:OptionalOutputs>` a `<VerificationReport>`-element, as specified in Section 3.2. The  
1666 `<VerificationReport>`-element MUST contain an `<IndividualSignatureReport>`-element (see  
1667 Section 3.3) for each signature or time stamp (i.e. `<dss:SignatureObject>`) contained in the  
1668 `<VerifyRequest>`-element.

1669 The `<Details>`-element within `<IndividualSignatureReport>` MAY contain other elements, such  
1670 as the Optional Outputs defined in Section 4.5 of [DSSCore].

### 1671 4.2 Level 2 – “Comprehensive”

1672 The conformance level “Advanced” comprises all requirements of conformance Level 1 (“Basic”), as  
1673 explained in Section 4.1. Furthermore the `<Details>`-element within each `<IndividualReport>`  
1674 MUST contain exactly one object-specific element, which documents the detailed verification results for  
1675 the signatures or validation data under consideration. While it is REQUIRED in this conformance level  
1676 that certificate values and revocation values are included into the verification report if requested by the  
1677 `IncludeCertificateValues-` and `IncludeRevocationValues-`element within the  
1678 `ReturnVerificationReport-`element (cf. Section 3.1), it is NOT REQUIRED in this conformance level  
1679 to expand those values and other relevant validation data to XML-structures if requested by the  
1680 `ExpandBinaryValues-`element.

1681 The object-specific detail elements defined in this specification are given as follows:

- 1682 • `<DetailedSignatureReport>` (cf. Section 3.5) - is used for the verification of (advanced)  
1683 electronic signatures.
- 1684 • `<IndividualTimeStampReport>` (cf. Section 3.5.5) – is used for the verification of individual time  
1685 stamps according to [RFC3161], which are not included in a signature.
- 1686 • `<IndividualCertificateReport>` (cf. Section 3.5.6) – is used for the verification of individual  
1687 certificates according to [RFC5280], which are not included in a signature.
- 1688 • `<IndividualAttributeCertificateReport>` (cf. Section 3.5.7) - is used for the verification  
1689 of individual attribute certificates according to [RFC3281], which are not included in a signature.
- 1690 • `<IndividualCRLReport>` (cf. Section 3.5.8) - is used for the verification of individual CRLs  
1691 according to [RFC5280], which are not included in a signature.
- 1692 • `<IndividualOCSPReport>` (cf. Section 3.5.9) - is used for the verification of individual OCSP-  
1693 responses according to [RFC2560], which are not included in a signature.
- 1694 • `<EvidenceRecordReport>` (cf. Section 3.5.10) – is used for the verification of evidence records  
1695 according to [RFC4998].

1696 Other object-specific detail elements MAY be defined in other profiles.

1697 **4.3 Level 3 – “Convenient”**

1698 The conformance Level 3 (“Convenient”) comprises all requirements of the conformance Level 2  
1699 (“Comprehensive”), as explained in Section 4.2. Furthermore the binary values of the validation data  
1700 MUST be expanded to the corresponding XML-structures, if this is requested by the  
1701 `ExpandBinaryValues`-element within the `ReturnVerificationReport`-element (cf. Section 3.1).

---

1702 **A. Acknowledgements**

1703 The following individuals have participated in the creation of this specification and are gratefully  
1704 acknowledged:

1705 **Participants:**

- 1706 • Juan-Carlos Cruellas
- 1707 • Andreas Kühne
- 1708 • Ingo Henkel
- 1709 • Ezer Farhi
- 1710 • Stefan Drees
- 1711 • Pim van der Eijk
- 1712 • Clemens Orthacker
- 1713 • Marta Cruellas
- 1714 • Konrad Lanz

1715

1716

## B. Revision History

1717

Revision	Date	Editor	Changes Made
R1	19.07.2009	Detlef Hühnlein	CD1 version on current OASIS template
R2	15.03.2010	Detlef Hühnlein	Draft of CS1 version, which includes a clarifying footnote and minor editing
R3	16.06.2010	Detlef Hühnlein	Potential CS1 version, which uses <code>TSLValidity</code> -element of <b>dss:AnyType</b> and drops the previously used <b>TrustStatusListValidityType</b> in order to support different TSL-versions.
R4	15.07.2010	Detlef Hühnlein	Potential CS1 version, which provides textual recommendations for filling the <code>TSLValidity</code> -element.
R5	27.09.2010	Detlef Hühnlein	Editorial correction of references section

1718

1719