OASIS 🕅

1

Abstract Code-Signing Profile of the OASIS Digital Signature Services

4 2nd Committee Draft, 11th September 2006 (WD 06)

5	Document identifier:
6	oasis-dss-1.0-profiles-codesigning-spec-cd-r2
7	Location:
8	http://docs.oasis-open.org/dss/v1.0/
9	Editor:
10	Andreas Kuehne, individual
11	Contributors:
12	Trevor Perrin, <i>individual</i>
13	Pieter Kasselman, Cybertrust
14 15	Abstract:
16	This draft profiles the OASIS DSS core protocols and the Asynchronous Processing
17	Abstract Profile of the OASIS Digital Signature Services for the purpose of creating code-
18	signing signatures.
19	Status:
20	This is a Public review Draft produced by the OASIS Digital Signature Service Technical
21	Committee. Comments may be submitted to the TC by any person by clicking on "Send
22	A Comment" on the TC home page at:
23	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dss
24 25 26 27	For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Digital Signature Service TC web page at http://www.oasis-open.org/committees/dss/ipr.php.

11th September 2006 Page 1 of 15

Table of Contents

29	1	Introduction	3			
30		1.1 Notation	3			
31		1.2 Namespaces	3			
32		1.3 Overview (Non-normative)				
33	2	Profile Features	6			
34		2.1 Identifier	6			
35		2.2 Scope	6			
36		2.3 Relationship To Other Profiles	6			
37		2.4 Signature Object	6			
38		2.5 Transport Binding	6			
39		2.6 Security Binding	6			
40	3	Profile of Signing Protocol	7			
41		3.1 Element <dss:signrequest></dss:signrequest>	7			
42		3.1.1 Element <dss:optionalinputs></dss:optionalinputs>	7			
43		3.1.2 Element <dss:inputdocuments></dss:inputdocuments>	7			
44		3.2 Element <dss:signresponse></dss:signresponse>	7			
45		3.2.1 Element <dss:result></dss:result>	7			
46		3.2.2 Element <dss:optionaloutputs></dss:optionaloutputs>	7			
47		3.2.3 Element <dss:signatureobject></dss:signatureobject>	7			
48	4	Profile of Verifying Protocol	8			
49	5	Profile of Code-signing Signatures	9			
50	6	Profile of Server Processing Rules 1	0			
51	7	Profile of Client Processing Rules 1	1			
52	8	Editorial Issues1	2			
53	9	References1	3			
54	9.1 Normative					
55	Ap	ppendix A. Revision History 1	4			
56	6 Appendix B. Notices 15					

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 2 of 15

57 **1 Introduction**

- 58 The DSS signing and verifying protocols are defined in **[DSS Core]** and asynchronous
- 59 processing for DSS messages are defined in **[DSS Async]**. As defined in those documents,
- 60 these protocols have a fair degree of flexibility and extensibility. This is an abstract profile of 61 **IDSS Corel** and **IDSS Asyncl**. It also profiles the processing rules followed by clients and
- [DSS Core] and [DSS Async]. It also profiles the processing rules followed by clients and
 servers when using these protocols.
- The resulting profile is an *abstract profile*. Further profiles will build on this one to provide a basis for implementation and interoperability.
- 65 The following sections provide guidance to interpreting the rest of this document.

66 1.1 Notation

- 67 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
- 68 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
- 69 interpreted as described in IETF RFC 2119 [RFC 2119]. These keywords are capitalized when

visual 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

- 72 meant in their natural-language sense.
- 73 This specification uses the following typographical conventions in text: <ns:Element>,
- 74 Attribute, **Datatype**, OtherCode.

75 1.2 Namespaces

The structures described in this specification are contained in the schema file **[CS-XSD]**. All schema listings in the current document are excerpts from the schema file. In the case of a

- schema listings in the current document are excerpts from the schema file. In the case of a
 disagreement between the schema file and this document, the schema file takes precedence.
- 79 This schema is associated with the following XML namespace:

80 urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0

- 81 If a future version of this specification is needed, it will use a different namespace.
- 82 Conventional XML namespace prefixes are used in this document:
- The prefix dsscs: (or no prefix) stands for the DSS code-signing namespace [CS-XSD].
- The prefix dss: stands for the DSS core namespace [Core-XSD].
- The prefix async: stands for this profiles namespace [Async-XSD].
- The prefix ds: stands for the W3C XML Signature namespace [XMLSig].
- 87 Applications MAY use different namespace prefixes, and MAY use whatever namespace

defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces
 in XML specification [XML-ns].

11th September 2006 Page 3 of 15

90 1.3 Overview (Non-normative)

The DSS signing and verifying protocols are defined in [DSS Core]. Asynchronous processing of
 DSS signing and verification protocols are defined in [DSS Async]. As defined in that document,
 these protocols have a fair degree of flexibility and extensibility.

94 This specification provides an abstract profile of the DSS signing messages for the case where 95 the object or input document that is being signed is a software program that can be executed on a 96 computing platform. The software program may be in source form, or in compiled form. The 97 process for signing these software programs is referred to as code-signing. Code-signing allows 98 the recipient of a software program to receive assurances regarding the origin and integrity of the 99 program. The recipient may use this information to make a trust decision on whether to install or 100 execute a software program.

- 101 Traditionally the task of generating the signature on the software program is left to the software 102 developer. However it may not always be appropriate to combine the roles of the software 103 developer and the code-signer. By centralizing the generation of signatures in the code-signing 104 process, the role of the software developer and the code signer is easily separated. This has the 105 advantage that keys used for signing software programs can be better managed, access to the 106 keys can be better controlled, audit trails can be centrally kept, event records can be reliably 107 archived and signing policies can be rigorously enforced.
- In the centralized code-signing model, the software developer is responsible for the creation and 108 109 development of a program. The software developer may also perform some basic testing of the 110 software program. Before distributing the software program, the software developer may need to have the software program digitally signed to convey assurances regarding the origin and 111 authenticity of the software program to the receiving platform or user. In order to obtain a 112 signature the software developer contacts the code-signing service and requests a signature for 113 the software program. Part of this request may include authentication information to allow the 114 code-signing service to make a decision on whether the software developer is authorized to 115 request a signature for the software program. The request may also include the software program. 116 in source form, compiled form or both. The centralized code-signing server may then generate a 117 signature. Generation of the signature may be subject to numerous criteria, including whether the 118 119 software developer is authorized to request the signature and whether the software program 120 conforms to the norms and standards set by the code-signing service. The code-signing service 121 may decline to generate a signature if all of its criteria and conditions are not met. The exact 122 criteria for generating a signature are subject to the policies of the code-signing service and are 123 beyond the scope of this document and may be further specified as part of a concrete profile. 124 Once the signature is generated, the result is returned to the software developer and the signed 125 software program may be distributed.
- Depending on the policies and criteria of the code-signing service, there may be a substantial
 delay between the time of submission of the software program and the time of signature
 generation. This delay may make synchronous message exchange impractical and necessitate
- the use of asynchronous message exchange. The use of asynchronous message exchange
- allows the software developer to submit the request to the code-signing service, without receiving
- an immediate response containing the signature. The code signing service responds by
 acknowledging the receipt of the request. The software developer may then periodically poll the
- acknowledging the receipt of the request. The software developer may then periodically pointine
 code-signing service to retrieve the signed software program, or may retrieve the signed software
 program once it receives a notification from the code-signing server.
- program once it receives a notification from the code-signing server.
- 135 This asynchronous behavior can be achieved by combining the [DSS Core] with the [DSS
- 136 Async] profile. The object for which the signature is requested is included under

- 138 message. If the server can not fulfill the code-signing request synchronously, it responds with a 139 <dss:ResultMajor> code indicating that the request is pending and the
- 140
- <dss:SignatureObject> element is left undefined, as specified in [DSS Async]. If the 141
- signature request is processed asynchronously, the client may request the signature from the 142
- server using the <async:PendingRequest> message. The client may poll the server 143
- periodically by sending this message, or it may send the message in response to a notification received. The server may respond to the <async:PendingRequest> message by either 144
- 145 indicating that the request is still pending or it may return the signature or signed software
- program. Either of these will be part of the <dss:SignResponse> message. 146
- This document profiles and extends the [DSS Core] and [DSS Async] specifications to enable 147 148 the code-signing scenarios sketched.
- 149 This document does not provide a profile of the DSS verification messages and does not specify
- 150 a notification mechanism. Notification are currently not included within the [DSS Async] 151 specification.

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 5 of 15

152 2 Profile Features

153 2.1 Identifier

- 154 urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0
- A server implementing this profile MAY support asynchronous processing as defined in the asynchronous processing profile as defined in **[DSS Async]**.
- 157

The client MUST implement asynchronous processing as defined in [DSS Async] and MUST
 include the urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing identifier in the
 <dss:AdditionalProfile> element.

161 **2.2 Scope**

162 This document profiles the DSS signing protocol and Asynchronous Processing Protocol as 163 defined in **[DSS Core]** and **[DSS Async]**.

164 2.3 Relationship To Other Profiles

- 165 This profile is based directly on the **[DSS Core]** and **[DSS Async]**.
- This profile is an abstract profile which can not be implemented directly, and may be furtherprofiled.

168 2.4 Signature Object

169 This profile is intended to provide a general framework for code-signing signature services and 170 does not specify or constrain the type of signature object. It is up to future profiles of this abstract 171 profile to constrain the type of signature object.

172 2.5 Transport Binding

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

174 2.6 Security Binding

- 175 This profile does not specify or constrain the security binding.
- 176

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 6 of 15

3 Profile of Signing Protocol

178 3.1 Element <dss:SignRequest>

179 3.1.1 Element <dss:OptionalInputs>

- 180 None of the optional inputs specified in the **[DSS Core]** are precluded in this abstract profile.
- 181 The <AdditionalProfile> element MUST be present, and MUST include the
- 182 urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing identifier.

183 3.1.2 Element <dss:InputDocuments>

184 This is an abstract profile and no constraints are imposed on the type of input document for which 185 a signature may be requested.

186 3.2 Element <dss:SignResponse>

187 3.2.1 Element <dss:Result>

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

189 3.2.2 Element <dss:OptionalOutputs>

190 None of the optional outputs specified in the **[DSS Core]** are precluded in this abstract profile.

191 3.2.3 Element <dss:SignatureObject>

- 192 This is an abstract profile and no constraints are imposed on the signature type that may be 193 returned. If a signature or signed software program is returned, it MUST be included under this
- 194 element.
- 195
- 196

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 7 of 15

197 **4 Profile of Verifying Protocol**

198	This document doe	es not provide a	a profile of the DS	SS verification messages.
-----	-------------------	------------------	---------------------	---------------------------

199

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 8 of 15

200 5 Profile of Code-signing Signatures

201 202	This is an abstract profile and no constraints are imposed on the type of signatures that are allowed.
203	

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 9 of 15

6 Profile of Server Processing Rules

205 206 207	A DSS server that performs code-signing SHOULD perform the following steps upon receiving a <dss:signrequest>.</dss:signrequest>
208 209 210 211 212 213 214 215 216 217 218 219	 Determine if the requesting user is authorized to submit a <dss:signrequest> or <async:pendingrequest> message for the purpose of code-signing request. This decision may be based on the authentication provided by the transport and security bindings.</async:pendingrequest></dss:signrequest> If a <dss:signrequest> message is received, the server may respond synchronously by generating the signature according to the concrete code-signing profile and include the requested signature in the <dss:signresponse> message. Alternatively the server may respond with a <dss:signresponse> message containing a <dss:resultmajor> error code of Pending, indicating asynchronous processing of the request. The signature may then be generated at the convenience of the server. The requested signature may then be retrieved by the client using subsequent <async:pendingrequest> messages.</async:pendingrequest></dss:resultmajor></dss:signresponse></dss:signresponse></dss:signrequest>
220	• If the <async: pendingrequest=""> message is used to retrieve a signature on a</async:>
221 222	previously submitted software program, the server may use the OriginalRequestId attribute to determine if the requested signature has been generated.
223	 If the signature is available, the server SHOULD respond with a
224	<pre><dss:signresponse> message containing the requested signature.</dss:signresponse></pre>
225	 If the signature request has not been processed, the server SHOULD respond
226	with a <dss:signresponse> message containing a <resultmajor> error</resultmajor></dss:signresponse>
227 228	code of Pending. The client SHOULD submit a <async:pendingrequest> at a later time to retrieve the signature.</async:pendingrequest>
220 229	 If the signature could not be generated the server SHOULD respond with a
230	<pre></pre>
231	indicating the reason why the signature could not be generated.
232	
233	
234	
235	
236	

11th September 2006 Page 10 of 15

7 Profile of Client Processing Rules

	· · · · · · · · · · · · · · · · · · ·		
238	A DSS client that requests signatures from a DSS code-signing server SHOULD perform the		
239	following steps.		
240			
241	• The client MUST support asynchronous processing as defined in [DSS Async].		
242	The client SHOULD authenticate itself using one of the mechanisms available through		
243	the transport or security bindings.		
244	• If this is a new signature request the client MUST submit an original signature request for		
245	a software program using the <dss:signrequest> message.</dss:signrequest>		
246	 If the client receives a <dss:signresponse> message containing the</dss:signresponse> 		
247	requested signature, the code-signing process is complete		
248	o If the client receives a <dss:signresponse> message containing a</dss:signresponse>		
249	<resultmajor> element indicating that the request is pending, the client</resultmajor>		
250	retains the RequestID attribute and MAY submit a		
251	<async:pendingrequest> message which MUST include the</async:pendingrequest>		
252	OriginalRequestID attribute. The value of the OriginalRequestID		
253	attribute MUST be set to the value of the RequestID attribute used in the initial		
254	signature request.		
255	 If this is a signature retrieval request (i.e. a request to retrieve a signature that was 		
256	previously requested but not returned) the client MUST use the		
257	<async:pendingrequest> message which MUST include the OriginalRequestID</async:pendingrequest>		
258	attribute. The value of the OriginalRequestID attribute MUST be set to the value of		
259	the RequestID attribute used in the initial signature request. The client may be required		
260	to provide authentication information through a mechanism defined through the transport		
261	or security binding.		
262	 If the client receives a <dss:signresponse> message containing the</dss:signresponse> 		
263	requested signature, or an error code other than Pending, the code-signing		
264	process is complete.		
265	 If the client receives a <dss:signresponse> message containing a</dss:signresponse> 		
266	<pre><dss:requestmajor> element indicating that the request is still pending, the</dss:requestmajor></pre>		
267	client MAY re-submit a <async:pendingrequest> message at a later time.</async:pendingrequest>		
268	The <async:pendingrequest> message MUST include the</async:pendingrequest>		
269	OriginalRequestID attribute. The value of the OriginalRequestID		
270	attribute MUST be set to the value of the Request ID attribute used in the initial		
271	signature request.		

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 11 of 15

272 8 Editorial Issues

273 274		Updated from version one. Removed code-signing specific async mechanism in favour of the async profile specified in [DSS Async].
275	2)	Updated from version 2 to use version 4 of async processing profile.

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 12 of 15

276 9 References

277 9.1 Normative

278 279	[Async-XSD]	A, Kuehne. Asynchronous Processing Profile Schema. OASIS, (MONTH/YEAR TBD)
280 281	[CS-XSD]	P.Kasselman, DSS Abstract Code-Signing Schema. OASIS, (MONTH/YEAR TBD)
282	[Core-XSD]	T. Perrin et al. DSS Schema. OASIS, (MONTH/YEAR TBD)
283 284	[DSS Core]	T. Perrin et al. <i>Digital Signature Service Core Protocols and Elements</i> . OASIS, (MONTH/YEAR TBD)
285 286	[DSS Async]	Asynchronous Processing Abstract Profile of the OASIS Digital Signature Services, Working Draft 04, 21 August 2004
287 288 289	[RFC2119]	S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.
290 291 292	[XML-ns]	T. Bray, D. Hollander, A. Layman. <i>Namespaces in XML.</i> W3C Recommendation, January 1999. http://www.w3.org/TR/1999/REC-xml-names-19990114
293 294 295 296	[XMLSig]	D. Eastlake et al. <i>XML-Signature Syntax and Processing.</i> W3C Recommendation, February 2002. http://www.w3.org/TR/1999/REC-xml-names-19990114

297 298

299 300

11th September 2006 Page 13 of 15

301 Appendix A. Revision History

Rev	Date	By Whom	What
wd-01	2004-01-08	Pieter Kasselman	Initial version based oasis-dss-1.0- profiles-XYZ-spec-wd-03.doc by Trevor Perrin
wd-02	2004-06-25	Pieter Kasselman	New version based on oasis-dss-1.0- profiles-XYZ-spec-wd-04.doc. Remove code-signing specific async capabilities in favor of [DSS Async] mechanisms.
wd-03	2004-10-13	Pieter Kasselman	Editorial corrections and updates to take into account version four of [DSS Async].
wd-04	2004-11-24	Pieter Kasselman	Editorial corrections based on feedback from Trevor Perrin
wd-05	2004-11-26	Pieter Kasselman	Removed reference to <responsemechanism> to reflect changes in version wd-05 of async profile</responsemechanism>
cd-01	2004-12-24	Pieter Kasselman	Approved Committee Draft
wd-06	2006-08-31	Andreas Kuehne	Editor changed, Updated reference to RFC 2119

oasis-dss-1.0- profiles-codesigning-spec-cd-r2 Copyright © OASIS Open 2006. All Rights Reserved. 11th September 2006 Page 14 of 15

302 Appendix B. Notices

303 OASIS takes no position regarding the validity or scope of any intellectual property or other rights 304 that might be claimed to pertain to the implementation or use of the technology described in this 305 document or the extent to which any license under such rights might or might not be available; 306 neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS 307 308 website. Copies of claims of rights made available for publication and any assurances of licenses 309 to be made available, or the result of an attempt made to obtain a general license or permission 310 for the use of such proprietary rights by implementors or users of this specification, can be 311 obtained from the OASIS Executive Director.

312 OASIS invites any interested party to bring to its attention any copyrights, patents or patent 313 applications, or other proprietary rights which may cover technology that may be required to

314 implement this specification. Please address the information to the OASIS Executive Director.

315 Copyright © OASIS Open 2006. All Rights Reserved.

316 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,

318 published and distributed, in whole or in part, without restriction of any kind, provided that the

above copyright notice and this paragraph are included on all such copies and derivative works.
 However, this document itself does not be modified in any way, such as by removing the

321 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS

- specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
- Property Rights document must be followed, or as required to translate it into languages otherthan English.
- The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.
- 327 This document and the information contained herein is provided on an "AS IS" basis and OASIS

328 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO

- 329 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
- ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
 PARTICULAR PURPOSE.

11th September 2006 Page 15 of 15