

PKCS #11 Profiles Version 3.2

Committee Specification 01

14 November 2025

This stage:

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/cs01/pkcs11-profiles-v3.2-cs01.docx>

(Authoritative)

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/cs01/pkcs11-profiles-v3.2-cs01.html>

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/cs01/pkcs11-profiles-v3.2-cs01.pdf>

Previous stage:

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/csd01/pkcs11-profiles-v3.2-csd01.docx>

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/csd01/pkcs11-profiles-v3.2-csd01.html>

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/csd01/pkcs11-profiles-v3.2-csd01.pdf>

Latest stage:

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/pkcs11-profiles-v3.2.docx> (Authoritative)

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/pkcs11-profiles-v3.2.html>

<https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/pkcs11-profiles-v3.2.pdf>

Technical Committee:

OASIS PKCS 11 TC

Chairs:

Robert Relyea (rrelyea@redhat.com), Red Hat

Greg Scott (greg.scott@cryptsoft.com), Cryptsoft Pty Ltd

Editor:

Tim Hudson (tjh@cryptsoft.com), Cryptsoft Pty Ltd

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- PKCS #11 test cases: <https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/cs01/test-cases/>

Related work:

This specification replaces or supersedes:

- *PKCS #11 Profiles Version 3.1*. Edited by Tim Hudson. OASIS Standard. Latest stage: <https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.1/pkcs11-profiles-v3.1.html>.

This specification is related to:

- *PKCS #11 Specification Version 3.2*. Edited by Dieter Bong and Greg Scott. Latest stage: <https://docs.oasis-open.org/pkcs11/pkcs11-spec/v3.2/pkcs11-spec-v3.2.html>.

Abstract:

This document defines data types, functions and other basic components of the PKCS #11 Cryptoki interface.

Status:

This document was last revised or approved by the OASIS PKCS 11 TC on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pkcs11#technical.

44 TC members should send comments on this document to the TC's email list. Others should send
45 comments to the TC's public comment list, after subscribing to it by following the instructions at the "[Send](#)
46 [A Comment](#)" button on the TC's web page at <https://www.oasis-open.org/committees/pkcs11/>.

47 This specification is provided under the [RF on RAND Terms](#) Mode of the [OASIS IPR Policy](#), the mode
48 chosen when the Technical Committee was established. For information on whether any patents have
49 been disclosed that may be essential to implementing this specification, and any offers of patent licensing
50 terms, please refer to the Intellectual Property Rights section of the TC's web page ([https://www.oasis-
open.org/committees/pkcs11/ipr.php](https://www.oasis-
51 open.org/committees/pkcs11/ipr.php)).

52 Note that any machine-readable content ([Computer Language Definitions](#)) declared Normative for this
53 Work Product is provided in separate plain text files. In the event of a discrepancy between any such
54 plain text file and display content in the Work Product's prose narrative document(s), the content in the
55 separate plain text file prevails.

56 **Key words:**

57 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
58 NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to
59 be interpreted as described in BCP 14 [[RFC2119](#)] and [[RFC8174](#)] when, and only when, they appear in
60 all capitals, as shown here.

61 **Citation format:**

62 When referencing this document, the following citation format should be used:

63 **[PKCS11-Profiles-v3.2]**

64 *PKCS #11 Profiles Version 3.2*. Edited by Tim Hudson. 15 April 2025. OASIS Committee Specification
65 01. <https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/cs01/pkcs11-profiles-v3.2-cs01.html>. Latest
66 stage: <https://docs.oasis-open.org/pkcs11/pkcs11-profiles/v3.2/pkcs11-profiles-v3.2.html>.

67 **Notices:**

68 Copyright © OASIS Open 2025. All Rights Reserved.

69 Distributed under the terms of the OASIS IPR Policy, [<https://www.oasis-open.org/policies-guidelines/ipr/>].
70 For complete copyright information please see the full Notices section in an Appendix below.

71 Table of Contents

72	1	Introduction.....	5
73	2	Profiles.....	6
74	2.1	Profile Requirements	6
75	2.2	Guidelines for other Profiles	6
76	2.3	Defined Profile Identifiers.....	6
77	3	Conformance Test Cases.....	8
78	3.1	Permitted Test Case Variations	8
79	3.1.1	Variable Items.....	8
80	3.1.2	Variable behavior	9
81	4	PKCS#11 XML Representation.....	10
82	4.1	Normalizing Names	10
83	4.2	Omitted Items	10
84	4.3	Value Representation	10
85	4.3.1	Enumerated Type Representation	10
86	4.3.2	Boolean Representation.....	10
87	4.3.3	Flag Type Representation	10
88	4.3.4	Special Value Representation	10
89	4.3.5	Function Call and Return Representation	10
90	4.3.6	Array and List Representation.....	11
91	4.3.7	Determining Array or List Length.....	11
92	4.3.8	Hexadecimal String Encoding	11
93	4.4	XML Root Element.....	11
94	4.5	XML Namespaces	11
95	4.6	XML Element Encoding	11
96	4.6.1	Boolean	11
97	4.6.2	Text String	12
98	4.6.3	Byte String.....	12
99	4.6.4	Enumerated Type.....	12
100	4.6.5	Function Call and Return.....	12
101	4.6.6	Attribute	12
102	4.6.7	XML Element Attributes.....	13
103	5	Base Profiles	14
104	5.1	Baseline Provider.....	14
105	5.1.1	Baseline Provider Mandatory Test Cases	15
106	5.1.1.1	BL-M-1-32.....	15
107	5.2	Complete Provider	15
108	5.3	Extended Provider	15
109	5.3.1	Extended Provider Mandatory Test Cases	16
110	5.3.1.1	EXT-M-1-32	16
111	5.4	Authentication Token	16
112	5.4.1	Authentication Token Provider Mandatory Test Cases.....	16
113	5.4.1.1	AUTH-M-1-32.....	16
114	5.5	Public Certificates Token	17
115	5.5.1	Public Certificates Token Provider Mandatory Test Cases.....	17

116	5.5.1.1 CERT-M-1-32.....	17
117	5.6 HKDF TLS Token	17
118	5.7 Baseline Consumer	18
119	5.8 Extended Consumer	19
120	6 Conformance	20
121	6.1 Baseline Provider Profile Conformance.....	20
122	6.2 Complete Provider Profile Conformance	20
123	6.3 Extended Provider Profile Conformance	20
124	6.4 Authentication Token Provider Profile Conformance	20
125	6.5 Public Certificates Token Provider Profile Conformance	20
126	6.6 HKDF TLS Token Provider Profile Conformance	20
127	6.7 Baseline Consumer Profile Conformance	21
128	6.8 Authentication Token Consumer Profile Conformance	21
129	6.9 Public Certificates Token Consumer Profile Conformance	21
130	7 PKCS #11 Implementation Conformance	22
131	7.1 PKCS#11 Consumer Implementation Conformance	22
132	7.2 PKCS#11 Provider Implementation Conformance	22
133	Appendix A. References	23
134	A.1 Normative References.....	23
135	A.2 Informative References	23
136	Appendix B. Acknowledgments	24
137	B.1 Special Thanks	24
138	B.2 Participants.....	24
139	Appendix C. Revision History.....	25
140	Appendix D. Notices.....	26
141		

142 1 Introduction

143 This document intends to meet this OASIS requirement on conformance clauses for providers and
144 consumers of cryptographic services via PKCS#11 ([\[PKCS11_Spec\]](#) Section 7 - PKCS#11
145 Implementation Conformance) through profiles that define the use of PKCS#11 data types, objects,
146 functions and mechanisms within specific contexts of provider and consumer interaction. These profiles
147 define a set of normative constraints for employing PKCS#11 within a particular environment or context of
148 use. They may, optionally, require the use of specific PKCS#11 functionality or in other respects define
149 the processing rules to be followed by profile actors.

150 For normative definition of the elements of PKCS#11 specified in these profiles, see the PKCS#11
151 Specification [\[PKCS11_Spec\]](#).

152

2 Profiles

153

154 This document defines a selected set of conformance clauses which form PKCS #11 Profiles. A profile
155 may be standalone or may be specified in terms of changes relative to another profile.

156 The PKCS 11 TC also welcomes proposals for new profiles. PKCS 11 TC members are encouraged to
157 submit these proposals to the PKCS 11 TC for consideration for inclusion in a future version of this TC-
158 approved document.

2.1 Profile Requirements

159

160 The following items SHALL be addressed by each profile:

- 161 1. Specify the versions of the PKCS#11 specification that SHALL be supported if versions other
162 than [PKCS11_Spec] are supported
- 163 2. Specify the list of additional data types that SHALL be supported
- 164 3. Specify the list of additional attributes that SHALL be supported
- 165 4. Specify the list of additional objects that SHALL be supported
- 166 5. Specify the list of additional functions that SHALL be supported
- 167 6. Specify the list of additional mechanisms that SHALL be supported
- 168 7. Specify any other requirements that SHALL be supported
- 169 8. Specify any mandatory test cases that SHALL be supported by conforming implementations
- 170 9. Specify optional test cases that MAY be supported by conforming implementations

171 Note: items may be specified either directly in a profile or by reference to other profiles. Where another
172 profile is referenced as required, the combination of the requirements of all referenced required profiles
173 (directly or indirectly) SHALL apply.

2.2 Guidelines for other Profiles

174

175 Any vendor or organization, such as other standards bodies, MAY create a PKCS#11 Profile and publish
176 it.

- 177 1. The profile SHALL be publicly available.
- 178 2. The PKCS11 Technical Committee SHALL be formally advised of the availability of the profile
179 and the location of the published profile.
- 180 3. The profile SHALL meet all the requirements of section 2.1
- 181 4. The PKCS11 Technical Committee SHOULD review the profile prior to final publication.

2.3 Defined Profile Identifiers

182

183 Profile objects (object class *CKO_PROFILE*) describe which PKCS #11 profiles a provider implements.
184

184

185 The *CKA_PROFILE_ID* attribute identifies a profile that the provider implements.

Attributes	Data Types	Meaning
CKA_PROFILE_ID	CK_PROFILE_ID	ID of the supported profile

186

187 The following table defines the *CK_PROFILE_ID* values:

Constant	Meaning
CKP_INVALID_ID	Invalid Profile
CKP_BASELINE_PROVIDER	Baseline Provider
CKP_EXTENDED_PROVIDER	Extended Provider

CKP_AUTHENTICATION_TOKEN	Authentication Token Provider or Consumer
CKP_PUBLIC_CERTIFICATES_TOKEN	Public Certificates Token Provider or Consumer
CKP_COMPLETE_PROVIDER	Complete Provider
CKP_HKDF_TLS_TOKEN	HKDF TLS Token
CKP_VENDOR_DEFINED	Vendor defined

188 3 Conformance Test Cases

189 The test cases define a sequence of PKCS#11 function calls with specified input and output parameters.
190 Each test case is provided in the XML format specified in PKCS#11 XML Representation (4) intended to
191 be both human-readable and usable by automated tools.

192 Each test case has a unique label (the section name) which includes indication of mandatory (-M-) or
193 optional (-O-) status and the specification version major and minor numbers as part of the identifier.

194 The test cases may depend on a specific configuration of a PKCS#11 provider and consumer and being
195 configured in a manner consistent with the test case assumptions.

196 Where possible the flow of identifiers between tests, date values, and other dynamic items are indicated
197 using symbolic identifiers – in actual request and response messages these dynamic values will be filled
198 in with valid values.

199 Symbolic identifiers SHALL be of the form $\${ParameterName}$. Wherever a symbolic identifier occurs in a
200 test case the implementation must replace it with a reasonable appearing datum of the expected type.

201 The symbolic identifier may reference return parameters or array or list items by index number. Array
202 index numbers SHALL be of the form $\${ParameterName[ArrayIndex]}$ and the first element SHALL be
203 indicated by index zero.

204 The symbolic identifier may reference elements nested within other elements. Nested references SHALL
205 be of the form $\${ParameterName.SubElement}$ and MAY also include an array index.

206 Note: the values for the returned items are illustrative. Actual values from a real consumer or provider
207 MAY vary as specified in section 3.1.

208 3.1 Permitted Test Case Variations

209 Whilst the test cases provided in a Profile define the allowed call and return content, some inherent
210 variations MAY occur and are permitted within a successfully completed test case.

211 Each test case MAY include allowed variations in the description of the test case in addition to the
212 variations noted in this section.

213 Other variations not explicitly noted in this section SHALL be deemed non-conformant.

214 3.1.1 Variable Items

215 An implementation conformant to a Profile MAY vary the following values (expressed using the XML
216 name for the items):

217 Provider specific information within the Info, SlotInfo and TokenInfo elements:

- 218 1. LibraryDescription
- 219 2. LibraryVersion
- 220 3. ManufacturerID
- 221 4. SlotDescription
- 222 5. HardwareVersion
- 223 6. FirmwareVersion
- 224 7. serialNumber
- 225 8. label
- 226 9. model
- 227 10. utcTime

228 Session specific information:

- 229 1. SlotID
- 230 2. Object

- 231 3. Session
- 232 Object specific information:
- 233 1. Object
- 234 Operation specific information:
- 235 1. Data
- 236 2. EncryptedData
- 237 3. RandomData
- 238 Attribute specific information:
- 239 1. VALUE
- 240 2. PUBLIC_EXPONENT
- 241 3. PRIVATE_EXPONENT
- 242 4. PRIME_1
- 243 5. PRIME_2
- 244 6. EXPONENT_1
- 245 7. EXPONENT_2
- 246 8. COEFFICIENT
- 247 9. PRIME
- 248 10. SUBPRIME
- 249 11. BASE
- 250 12. EC_POINT
- 251 13. UNIQUE_ID

252 3.1.2 Variable behavior

- 253 An implementation conformant to a Profile SHALL allow variation of the following behavior:
- 254 1. A test may omit the clean-up functions at the end of the test provided there is a separate
255 mechanism to remove the created objects during testing.
 - 256 2. A test may omit the test identifiers in various attributes if the consumer is unable to include them
257 in calls.
 - 258 3. The number of entries and order of entries in the list returned in the *C_GetSlotList*,
259 *C_GetMechanismList*, and *C_GetInterfaceList* functions may vary, provided that at least one
260 entry within the list matches the logical context of the test case.
- 261

262 4 PKCS#11 XML Representation

263 4.1 Normalizing Names

264 PKCS#11 parameter and structure field names SHALL be normalized to create a 'CamelCase' format that
265 would be suitable to be used as a variable name in C/Java or an XML element name.

266 Hungarian notation type indicators are entirely omitted from names (i.e. *h*, *ph*, *ul*, *pul*, and *p* are omitted).

267 PKCS#11 function names are represented as-is (unchanged) as XML elements of the same name.

268 4.2 Omitted Items

269 PKCS#11 pointers for callback functions and reserved items are entirely omitted (i.e. *pApplication*,
270 *pReserved*, *Notify* are not present).

271 Hungarian notation type indicators are entirely omitted from names (i.e. *h*, *ph*, *ul*, *pul*, and *p* are omitted).

272 4.3 Value Representation

273 The value for PKCS#11 binary (*CK_BYTE*) information SHALL be encoded as hexadecimal strings.

274 The value for PKCS#11 textual information (*CK_CHAR*, *CK_UTF8CHAR*) SHALL be encoded as hex
275 strings.

276 The value for PKCS#11 numeric information SHALL be encoded as integers or as hexadecimal strings.

277 4.3.1 Enumerated Type Representation

278 Each PKCS#11 type value SHALL be represented in string/text form using the uppercase C macro name
279 with the type prefix omitted. E.g. *CKR_OK* has a representation of "OK".

280 4.3.2 Boolean Representation

281 Each PKCS#11 boolean value (*CK_BBOOL*) SHALL be represented in string/text form either as "true"
282 (non-zero) or "false" (zero). No other representation SHALL be used.

283 4.3.3 Flag Type Representation

284 Each PKCS#11 flag value SHALL be represented using the uppercase C macro names with the type
285 prefix omitted for each bit. If multiple bit flags are set then each SHALL be present separated by either a
286 space (' ') or a pipe ('|') character.

287 4.3.4 Special Value Representation

288 For PKCS#11 *CK_ULONG* values which have special interpretation as
289 *CK_UNAVAILABLE_INFORMATION* or *CK_EFFECTIVELY_INFINITE* the string values
290 "UnavailableInformation" and "EffectivelyInfinite" SHOULD be used instead of the numeric values to
291 improve readability. This approach is used in the *CK_TOKEN_INFO* structure for various count and
292 length and size values.

293 4.3.5 Function Call and Return Representation

294 PKCS#11 function calls are represented as an XML element of the same name containing the input
295 parameters each represented as XML elements and an XML element of the same name as the PKCS#11
296 function name with an XML element attribute named *rv* containing the return value. The XML element for
297 the input parameters is always immediately followed by the XML element for the output results.

298 PKCS#11 parameters and structure members that are not arrays or lists are represented as XML
299 elements with the value of the parameter or structure member contained within the XML element attribute
300 *value*.

301 4.3.6 Array and List Representation

302 PKCS#11 parameters and structure members that are arrays or lists are represented as XML elements
303 with the length of the array or list contained in XML element attribute *length* and the members of the array
304 or list represented as nested XML elements unless an XML element attribute-based representation has
305 been separately defined (e.g for *CK_ATTRIBUTE*).

306 PKCS#11 parameters and structure member elements that represent the count of arrays are omitted as
307 input parameters as the lengths can be determined by a count of the number of XML elements within the
308 call or return XML element within the element representing the PKCS#11 function call.

309 4.3.7 Determining Array or List Length

310 The PKCS#11 approach of passing in a NULL pointer value and using an input/output parameter to
311 determine the required pointer buffer length for a subsequent call SHALL be encoded as request where
312 the XML element for pointer has no specified value or length for the function call and the returned length
313 is contained in the XML element attribute *length*.

314 4.3.8 Hexadecimal String Encoding

315 Hexadecimal strings SHALL NOT include any white space.

316 Hexadecimal strings SHALL use either uppercase 'A'-'F' or lowercase 'a'-'f' along with '0' to '9'.

317 Numeric values represented as hexadecimal strings SHALL begin with '0x'.

318 Binary values represented as hexadecimal strings SHOULD omit the '0x'.

319 4.4 XML Root Element

320 XML documents representing a sequence of PKCS#11 function calls and returns SHALL have an XML
321 root element of *PKCS11*.

322 4.5 XML Namespaces

323 If namespaces are necessary within a specific context, then each XML element SHALL use the following
324 namespace:

325 urn:oasis:tc:pkcs11:xmlns

326 4.6 XML Element Encoding

327 For XML, each function call is represented as a sequence of two XML element with optional attributes.

328 The parameters to each call are represented as nested XML elements, and any structures used within
329 those parameters are represented as nested XML elements within the nested XML elements.

330 The types of each parameter or structure element are fixed within the PKCS#11 specification and are not
331 separately represented within the XML encoding. i.e. the types are inherently known by implementations
332 and are fixed, matching the underlying C static type declaration.

333 4.6.1 Boolean

334 XML value uses [XML-SCHEMA] type xsd:Boolean. The value SHALL be FALSE, false, TRUE or true.

335 `<TokenPresent value="false"/>`

336 4.6.2 Text String

337 XML value uses [XML-SCHEMA] type xsd:string

```
338 <Pin value="12345678"/>
```

339 4.6.3 Byte String

340 XML value uses [XML-SCHEMA] type xsd:hexBinary

```
341 <EncryptedData value="8dce78ad"/>
```

342 4.6.4 Enumerated Type

343 XML value uses [XML-SCHEMA] type xsd:string and is either a hexadecimal string or the *Enumerated*
344 *Type Representation* name. If an XSD with xsd:enumeration restriction is used to define valid values
345 parsers should also accept any hexadecimal string in addition to the defined enumeration values to allow
346 for user extensions and non-textual encoding parsers.

```
347 <Type value="AES_CBC"/>
```

```
348 <Type value="0x00001082"/>
```

```
349 <Type value="4426"/>
```

350 4.6.5 Function Call and Return

351 PKCS#11 function call and return SHALL be encoded as an XML element for the function call with any
352 required parameters as nested XML elements, followed by an XML element for the function return with an
353 XML element attribute of *rv* containing the return code from the function call encoded as an Enumerated
354 Type and any output parameters as nested XML elements.

```
355 <C_Initialize/>
```

```
356 <C_Initialize rv="OK"/>
```

```
357 <C_GetSlotList>
```

```
358   <TokenPresent value="false"/>
```

```
359   <SlotList/>
```

```
360 </C_GetSlotList>
```

```
361 <C_GetSlotList rv="OK">
```

```
362   <SlotList length="1"/>
```

```
363 </C_GetSlotList>
```

364 4.6.6 Attribute

365 PKCS#11 attributes (*CK_ATTRIBUTE*) SHALL be encoded as an XML element with an XML element
366 attribute *type* containing the name of the PKCS#11 attribute and an XML element attribute *value*
367 containing the value of the attribute. Where the PKCS#11 attribute has a specified type, the *value* SHALL
368 be encoding using the encoding rules for that type of PKCS#11 value.

```
369 <Attribute type="CLASS" value="SECRET_KEY"/>
```

```
370 <Attribute type="KEY_TYPE" value="AES"/>
```

```
371 <Attribute type="LABEL" value="timing-key"/>
```

```
372 <Attribute type="TOKEN" value="TRUE"/>
```

```
373 <Attribute type="PRIVATE" value="TRUE"/>
```

```
374 <Attribute type="EXTRACTABLE" value="TRUE"/>
```

```
375 <Attribute type="SENSITIVE" value="TRUE"/>
```

```
376 <Attribute type="ENCRYPT" value="TRUE"/>
```

```
377 <Attribute type="DECRYPT" value="TRUE"/>
```

```
378 <Attribute type="VALUE_LEN" value="16"/>
```

```
380
```

381 **4.6.7 XML Element Attributes**

382 XML element attributes other than “type”, “value”, “length” and “rv” as defined in this specification SHALL
383 not be used. All other PKCS#11 concepts are represented as XML elements and not XML element
384 attributes.
385

386

5 Base Profiles

387 The following subsections describe currently-defined profiles related to the use of PKCS #11. The profiles
388 define classes of PKCS #11 functionality to which an implementation can declare conformance.

5.1 Baseline Provider

390 A PKCS #11 provider makes cryptographic functionality available to a consuming application in terms of
391 the PKCS #11 API.

392 This profile specifies the most basic functionality that would be expected of a conformant PKCS #11
393 provider – the ability to provide information about the capabilities of the cryptographic services provided.

394 An implementation conforms to this specification as a Baseline Provider if it meets the following
395 conditions:

- 396 1. Supports the conditions required by the *PKCS#11 Provider Implementation Conformance*
397 clauses [\[PKCS11_Spec\]](#)
- 398 2. Supports the following data types [\[PKCS11_Spec\]](#):
 - 399 a. `CK_VERSION`
 - 400 b. `CK_INFO`
 - 401 c. `CK_SLOT_ID`
 - 402 d. `CK_SLOT_INFO`
 - 403 e. `CK_TOKEN_INFO`
 - 404 f. `CK_SESSION_HANDLE`
 - 405 g. `CK_USER_TYPE`
 - 406 h. `CK_SESSION_INFO`
 - 407 i. `CK_OBJECT_HANDLE`
 - 408 j. `CK_OBJECT_CLASS`
 - 409 k. `CK_ATTRIBUTE_TYPE`
 - 410 l. `CK_ATTRIBUTE`
 - 411 m. `CK_PROFILE_ID`
 - 412 n. `CK_RV`
 - 413 o. `CK_FUNCTION_LIST`
 - 414 p. `CK_INTERFACE`
 - 415 q. `CK_C_INITIALIZE_ARGS`
- 416 3. Supports the following attributes [\[PKCS11_Spec\]](#):
 - 417 a. `CKA_CLASS`
 - 418 b. `CKA_TOKEN`
 - 419 c. `CKA_VALUE`
 - 420 d. `CKA_ID`
 - 421 e. `CKA_PRIVATE`
 - 422 f. `CKA_MODIFIABLE`
 - 423 g. `CKA_LABEL`
 - 424 h. `CKA_UNIQUE_IDENTIFIER`
 - 425 i. `CKA_PROFILE_ID`
- 426 4. Supports the following objects [\[PKCS11_Spec\]](#):
 - 427 a. `CKO_PROFILE` with value `CKP_BASELINE_PROVIDER`
- 428 5. Supports the following functions [\[PKCS11_Spec\]](#):
 - 429 a. `C_GetFunctionList`
 - 430 b. `C_GetInterfaceList`
 - 431 c. `C_GetInterface`
 - 432 d. `C_Initialize`
 - 433 e. `C_Finalize`
 - 434 f. `C_GetInfo`
 - 435 g. `C_GetSlotList`
 - 436 h. `C_GetSlotInfo`
 - 437 i. `C_GetTokenInfo`

- 438 j. C_OpenSession
- 439 k. C_CloseSession
- 440 l. C_GetSessionInfo
- 441 m. C_FindObjectsInit
- 442 n. C_FindObjects
- 443 o. C_FindObjectsFinal
- 444 p. C_GetAttributeValue
- 445 6. Supports the following mechanisms:
- 446 a. None specified
- 447 7. Supports *Error Handling* [PKCS11_Spec] for any supported object, function or mechanism
- 448 8. Optionally supports any clause within [PKCS11_Spec] that is not listed above
- 449 9. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 450 extensions, conformance clauses) that do not contradict any PKCS #11 requirements

451 5.1.1 Baseline Provider Mandatory Test Cases

452 5.1.1.1 BL-M-1-32

453 See <test-cases/pkcs11-v3.2/mandatory/BL-M-1-32.xml>

454 5.2 Complete Provider

455 A PKCS #11 provider makes cryptographic functionality available to a consuming application in terms of
456 the PKCS #11 API.

457 This profile specifies the functionality that would be expected of a conformant PKCS #11 provider that
458 implements the entire specification.

459 An implementation conforms to this specification as a Complete Provider if it meets the following
460 conditions:

- 461 1. Supports the conditions required by the *PKCS#11 Provider Implementation Conformance*
- 462 clauses [PKCS11_Spec]
- 463 2. Supports all data types [PKCS11_Spec]
- 464 3. Supports all attributes [PKCS11_Spec]
- 465 4. Supports all objects [PKCS11_Spec]
- 466 5. Supports all functions [PKCS11_Spec]
- 467 6. Supports all mechanisms [PKCS11_Spec] Section 6
- 468 7. Supports *Error Handling* [PKCS11_Spec]
- 469 8. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 470 extensions, conformance clauses) that do not contradict any PKCS #11 requirements

471 5.3 Extended Provider

472 This profile builds on the PKCS#11 Baseline Provider to add support for mechanism-based usage.

473 An implementation conforms to this specification as an Extended Provider if it meets the following
474 conditions:

- 475 1. Supports the conditions required by the PKCS #11 conformance clauses ([PKCS11_Spec]
476 Section 7 (PKCS#11 Implementation Conformance))
- 477 2. Supports the conditions required by the PKCS #11 Baseline Provider clauses section5.1.
- 478 3. Supports the following data types [PKCS11_Spec]:
- 479 a. CK_MECHANISM_TYPE
- 480 b. CK_MECHANISM
- 481 4. Supports the following attributes [PKCS11_Spec]:
- 482 a. None specified
- 483 5. Supports the following objects [PKCS11_Spec]:
- 484 a. CKO_PROFILE with value CKP_EXTENDED_PROVIDER
- 485 6. Supports the following functions [PKCS11_Spec]:

- 486 a. *C_GetMechanismList*
- 487 b. *C_GetMechanismInfo*
- 488 c. *C_Login*
- 489 d. *C_LoginUser*
- 490 e. *C_Logout*
- 491 7. Supports the following mechanisms:
- 492 a. None specified
- 493 8. Supports *Error Handling* [PKCS11_Spec] for any supported object, function or mechanism
- 494 9. Optionally supports any clause within [PKCS11_Spec] that is not listed above
- 495 10. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 496 extensions, conformance clauses) that do not contradict any PKCS #11 requirements

497 5.3.1 Extended Provider Mandatory Test Cases

498 5.3.1.1 EXT-M-1-32

499 See <test-cases/pkcs11-v3.2/mandatory/EXT-M-1-32.xml>

500 5.4 Authentication Token

501 This profile builds on the PKCS #11 Baseline Provider and/or Baseline Consumer profiles to provide for
502 use in the context of an authentication token.

503 An implementation conforms to this specification as an Authentication Token if it meets the following
504 conditions:

- 505 1. If the implementation is a consumer then it SHALL support the conditions required by the
- 506 PKCS #11 Baseline Consumer Clause (Section 5.7)
- 507 2. If the implementation is a provider then it SHALL support the conditions required by the
- 508 PKCS #11 Baseline Provider Clause (Section 5.1)
- 509 3. Supports the following data types [PKCS11_Spec]:
- 510 a. None specified
- 511 4. Supports the following attributes [PKCS11_Spec]:
- 512 a. None specified
- 513 5. Supports the following objects [PKCS11_Spec]:
- 514 a. *CKO_PRIVATE_KEY*
- 515 b. *CKO_PUBLIC_KEY*
- 516 c. *CKO_PROFILE* with value *CKP_AUTHENTICATION_TOKEN*
- 517 6. Supports the following functions [PKCS11_Spec]:
- 518 a. *C_Login*
- 519 b. *C_LoginUser*
- 520 c. *C_Logout*
- 521 d. *C_SignInit*
- 522 e. *C_Sign* and/or *C_SignUpdate* and *C_SignFinal*
- 523 7. Supports the following mechanisms:
- 524 a. None specified
- 525 8. Supports *Error Handling* [PKCS11_Spec] for any supported object, function or mechanism
- 526 9. Optionally supports any clause within [PKCS11_Spec] that is not listed above
- 527 10. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 528 extensions, conformance clauses) that do not contradict any PKCS #11 requirements.

529 5.4.1 Authentication Token Provider Mandatory Test Cases

530 5.4.1.1 AUTH-M-1-32

531 See <test-cases/pkcs11-v3.2/mandatory/AUTH-M-1-32.xml>

532

533 5.5 Public Certificates Token

534 This profile builds on the PKCS #11 Baseline Provider and/or Baseline Consumer profiles to provide for
535 use in the context of a public certificates token.

536 An implementation conforms to this specification as a Public Certificates Token if it meets the following
537 conditions:

- 538 1. If the implementation is a consumer then it SHALL support the conditions required by the
539 PKCS #11 Baseline Consumer Clause (Section 5.7)
- 540 2. If the implementation is a provider then it SHALL support the conditions required by the
541 PKCS #11 Baseline Provider Clause (Section 5.1)
- 542 3. Supports the following data types [PKCS11_Spec]:
543 a. None specified
- 544 4. Supports the following attributes [PKCS11_Spec]:
545 a. None specified
- 546 5. Supports the following objects [PKCS11_Spec]:
547 a. *CKO_CERTIFICATE*
548 b. *CKO_PROFILE* with value *CKP_PUBLIC_CERTIFICATES_TOKEN*
- 549 6. Supports the following functions [PKCS11_Spec]:
550 a. None specified
- 551 7. Supports the following mechanisms [PKCS11_Spec]:
552 a. None specified
- 553 8. Supports the following object location requirements:
554 a. All certificates are publicly readable, able to be found on the token without a login
555 having been performed
556 b. All certificates for which a matching private key also exists on the token must have a
557 matching *CKA_ID* attribute for the certificate and private key
558 c. One or more of the following conditions must be met:
559 i. The matching private key for a certificate can be found via *C_FindObjects* using
560 the matching *CKA_ID* value without a login having been performed;
561 ii. The matching public key for a certificate can be found via *C_FindObjects* using
562 the matching *CKA_ID* value without a login having been performed
- 563 9. Supports Error Handling [PKCS11_Spec] for any supported object, function or mechanism
- 564 10. Optionally supports any clause within [PKCS11_Spec] that is not listed above
- 565 11. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
566 extensions, conformance clauses) that do not contradict any PKCS #11 requirements.

567 5.5.1 Public Certificates Token Provider Mandatory Test Cases

568 5.5.1.1 CERT-M-1-32

569 See <test-cases/pkcs11-v3.2/mandatory/CERT-M-1-32.xml>

570 5.6 HKDF TLS Token

571 This profile builds on the PKCS #11 Baseline Provider and/or Baseline Consumer profiles to provide for
572 use in the context of TLS 1.3 connections using the CKM_HKDF_DERIVE_DATA mechanism.

573 An implementation conforms to this specification as an HKDF TLS Token if it meets the following
574 conditions:

- 575 1. If the implementation is a consumer then it SHALL support the conditions required by the
576 PKCS #11 Baseline Consumer Clause (Section 5.7)
- 577 2. If the implementation is a provider then it SHALL support the conditions required by the
578 PKCS #11 Baseline Provider Clause (Section 5.1)
- 579 3. Supports the following data types [PKCS11_Spec]:
580 b. *CK_HKDF_PARAMS*
- 581 4. Supports the following attributes [PKCS11_Spec]:

- 582 a. None specified
- 583 5. Supports the following objects [PKCS11_Spec]:
- 584 a. CKO_DATA
- 585 b. CKO_SECRET_KEY
- 586 c. CKO_PROFILE with value CKP_HKDF_TLS_TOKEN
- 587 6. Supports the following functions [PKCS11_Spec]:
- 588 a. C_DeriveKey
- 589 7. Supports the following mechanisms:
- 590 a. CKM_HKDF_DATA
- 591 A conformant provider SHALL not reject derive requests based on the plnfo
- 592 value if the following plnfo values are given:
- 593 i. The string L1,L2,"tls iv",0 (L1, L2, 0x74, 0x6c, 0x73, 0x20, 0x69, 0x76, 0x00)
- 594 where L1 is the most significant byte of CKA_VALUE_LEN and L2 is the least
- 595 significant byte of CKA_VALUE_LEN.
- 596 ii. The string L1,L2,"tls quic iv",0 (L1, L2, 0x74, 0x6c, 0x73, 0x20, 0x71, 0x75,
- 597 0x69, 0x63, 0x20, 0x69, 0x76, 0x00) where L1 is the most significant byte of
- 598 CKA_VALUE_LEN and L2 is the least significant byte of CKA_VALUE_LEN.
- 599 A conformant provider MAY accept other values for plnfo.
- 600 8. Supports *Error Handling* [PKCS11_Spec] for any supported object, function or mechanism
- 601 9. Optionally supports any clause within [PKCS11_Spec] that is not listed above
- 602 10. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 603 extensions, conformance clauses) that do not contradict any PKCS #11 requirements.
- 604

605 5.7 Baseline Consumer

606 A PKCS #11 consumer calls a PKCS #11 provider implementation of the PKCS #11 API in order to use

607 the cryptographic functionality from that provider.

608 This profile specifies the most basic functionality that would be expected of a conformant PKCS #11

609 consumer – the ability to consume information via the cryptographic services offered by a provider.

610 An implementation conforms to this specification as a Baseline Consumer if it meets the following

611 conditions:

- 612 1. Supports the conditions required by the *PKCS#11 Consumer Implementation Conformance*
- 613 clauses [PKCS11_Spec]
- 614 2. Supports the following data types [PKCS11_Spec]:
- 615 a. CK_VERSION
- 616 b. CK_INFO
- 617 c. CK_SLOT_ID
- 618 d. CK_SLOT_INFO
- 619 e. CK_TOKEN_INFO
- 620 f. CK_SESSION_HANDLE
- 621 g. CK_USER_TYPE
- 622 h. CK_SESSION_INFO
- 623 i. CK_OBJECT_HANDLE
- 624 j. CK_OBJECT_CLASS
- 625 k. CK_ATTRIBUTE_TYPE
- 626 l. CK_ATTRIBUTE
- 627 m. CK_RV
- 628 n. CK_FUNCTION_LIST
- 629 o. CK_C_INITIALIZE_ARGS
- 630 p. CK_INTERFACE (if C_GetInterfaceList and C_GetInterface is supported)
- 631 3. Supports the following attributes [PKCS11_Spec]:
- 632 a. CKA_CLASS

- 633 *b.* *CKA_VALUE*
- 634 4. Supports the following objects:
- 635 *a.* None specified
- 636 5. Supports the following functions [[PKCS11_Spec](#)]:
- 637 *a.* *C_GetFunctionList* or *C_GetInterfaceList* and *C_GetInterface*
- 638 *b.* *C_Initialize*
- 639 *c.* *C_Finalize*
- 640 *d.* *C_GetInfo*
- 641 *e.* *C_GetSlotList*
- 642 *f.* *C_GetSlotInfo*
- 643 *g.* *C_GetTokenInfo*
- 644 *h.* *C_OpenSession*
- 645 *i.* *C_CloseSession*
- 646 6. Supports the following mechanisms:
- 647 *a.* None specified
- 648 7. Supports *Error Handling* [[PKCS11_Spec](#)] for any supported object, function or mechanism
- 649 8. Optionally supports any clause within [[PKCS11_Spec](#)] that is not listed above
- 650 9. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 651 extensions, conformance clauses) that do not contradict any PKCS #11 requirements

652 5.8 Extended Consumer

653 This profile builds on the PKCS#11 Baseline Consumer profile to add support for mechanism-based
654 usage.

655 An implementation conforms to this specification as an Extended Consumer if it meets the following
656 conditions:

- 657 1. Supports the conditions required by the PKCS11 conformance clauses ([[PKCS11_Spec](#)]
658 Section 7 (PKCS#11 Implementation Conformance))
- 659 2. Supports the conditions required by the PKCS11 Baseline Consumer clauses section 5.7
- 660 3. Supports the following data types [[PKCS11_Spec](#)]:
- 661 *a.* *CK_MECHANISM_TYPE*
- 662 *b.* *CK_MECHANISM*
- 663 4. Supports the following attributes [[PKCS11_Spec](#)]:
- 664 *a.* None specified
- 665 5. Supports the following objects [[PKCS11_Spec](#)]:
- 666 *a.* None specified
- 667 6. Supports the following functions [[PKCS11_Spec](#)]:
- 668 *a.* *C_GetMechanismList*
- 669 *b.* *C_GetMechanismInfo*
- 670 7. Supports the following mechanisms:
- 671 *a.* None specified
- 672 8. Supports *Error Handling* [[PKCS11_Spec](#)] for any supported object, function or mechanism
- 673 9. Optionally supports any clause within [[PKCS11_Spec](#)] that is not listed above
- 674 10. Optionally supports extensions outside the scope of this standard (e.g., vendor defined
- 675 extensions, conformance clauses) that do not contradict any PKCS #11 requirements

676 6 Conformance

677 The baseline provider and consumer profiles provide the most basic functionality that is expected of a
678 conformant PKCS#11 consumer or provider. The complete provider profile defines a PKCS#11 provider
679 that implements the entire specification. A PKCS#11 implementation conformant to this specification (the
680 PKCS#11 Profiles) SHALL meet all the conditions documented in one or more of the following sections.

681 6.1 Baseline Provider Profile Conformance

682 PKCS#11 provider implementations conformant to this profile:

- 683 1. SHALL support [PKCS11_Spec];
- 684 2. SHALL support the Baseline Provider conditions (5.1) and;
- 685 3. SHALL support one or more of the Baseline Provider Mandatory Test Cases (5.1.1).

686 6.2 Complete Provider Profile Conformance

687 PKCS#11 provider implementations conformant to this profile:

- 688 1. SHALL support [PKCS11_Spec];
- 689 2. SHALL support the Complete Provider conditions (5.2) and;
- 690 3. SHALL support all of the provider conformance clauses contained within Conformance (6).

691 6.3 Extended Provider Profile Conformance

692 PKCS#11 provider implementations conformant to this profile:

- 693 1. SHALL support [PKCS11_Spec];
- 694 2. SHALL support the Extended Provider conditions (5.3) and;
- 695 3. SHALL support one or more of the Extended Provider Mandatory Test Cases (5.3.1).

696 6.4 Authentication Token Provider Profile Conformance

697 PKCS#11 provider implementations conformant to this profile:

- 698 1. SHALL support [PKCS11_Spec];
- 699 2. SHALL support the Authentication Token conditions (5.4) and;
- 700 3. SHALL support all of the Authentication Token Provider Mandatory Test Cases (5.4.1).

701 6.5 Public Certificates Token Provider Profile Conformance

702 PKCS#11 provider implementations conformant to this profile:

- 703 1. SHALL support [PKCS11_Spec];
- 704 2. SHALL support the Public Certificates Token conditions (5.5) and;
- 705 3. SHALL support all of the Public Certificates Token Provider Mandatory Test Cases (5.5.1).

706 6.6 HKDF TLS Token Provider Profile Conformance

707 PKCS#11 provider implementations conformant to this profile:

- 708 1. SHALL support [PKCS11_Spec];
- 709 2. SHALL support the HKDF TLS Token conditions (5.6).

710 **6.7 Baseline Consumer Profile Conformance**

711 PKCS#11 consumer implementations conformant to this profile:

- 712 1. SHALL support [[PKCS11_Spec](#)]; and
713 2. SHALL support the Baseline Consumer conditions (5.7).

714 **6.8 Authentication Token Consumer Profile Conformance**

715 PKCS#11 provider implementations conformant to this profile:

- 716 1. SHALL support [[PKCS11_Spec](#)]; and
717 2. SHALL support the Authentication Token conditions (5.4)

718 **6.9 Public Certificates Token Consumer Profile Conformance**

719 PKCS#11 provider implementations conformant to this profile:

- 720 1. SHALL support [[PKCS11_Spec](#)]; and
721 2. SHALL support the Public Certificates Token conditions (5.5)
722

723 **7 PKCS #11 Implementation Conformance**

724 **7.1 PKCS#11 Consumer Implementation Conformance**

725 An implementation is a conforming PKCS#11 Consumer if the implementation meets the conditions
726 specified in one or more consumer profiles specified in **[PKCS11-Prof]**.

727 A PKCS#11 consumer implementation SHALL be a conforming PKCS#11 Consumer.

728 If a PKCS#11 consumer implementation claims support for a particular consumer profile, then the
729 implementation SHALL conform to all normative statements within the clauses specified for that profile
730 and for any subclauses to each of those clauses.

731 **7.2 PKCS#11 Provider Implementation Conformance**

732 An implementation is a conforming PKCS#11 Provider if the implementation meets the conditions
733 specified in one or more provider profiles specified in **[PKCS11-Prof]**.

734 A PKCS#11 provider implementation SHALL be a conforming PKCS#11 Provider.

735

736 Appendix A. References

737 This appendix contains the normative and informative references that are used in this document.
738 While any hyperlinks included in this appendix were valid at the time of publication, OASIS cannot
739 guarantee their long-term validity.

741 A.1 Normative References

742 The following documents are referenced in such a way that some or all of their content constitutes
743 requirements of this document.

745 [PKCS11_Spec]

746 *PKCS #11 Specification Version 3.2*. Edited by Dieter Bong and Greg Scott. Latest stage:
747 <https://docs.oasis-open.org/pkcs11/pkcs11-spec/v3.2/pkcs11-spec-v3.2.html>.

748 [RFC2119]

749 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI
750 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

751 [RFC8174]

752 Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI
753 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

754 [XML]

755 Bray, Tim, et.al. eds, Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26
756 November 2008,
757 <<http://www.w3.org/TR/2008/REC-xml-20081126>>.

759 A.2 Informative References

760 The following referenced documents are not required for the application of this document but may assist
761 the reader with regard to a particular subject area.

763 [RFC3552]

764 Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC
765 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.

766 [XML-SCHEMA]

767 Paul V. Biron, Ashok Malhotra, XML Schema Part 2: Datatypes Second Edition, W3C Recommendation
768 26 November 2008, <<https://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

769

770 **Appendix B. Acknowledgments**

771 **B.1 Special Thanks**

772 Substantial contributions to this document from the following individuals are gratefully acknowledged:

773

774 Ms. Dina Kurktchi-Nimeh, Oracle

775 **B.2 Participants**

776 The following individuals were members of this Technical Committee during the creation of this document
777 and their contributions are gratefully acknowledged:

778

779 Dr. Warren Armstrong, QuintessenceLabs Pty Ltd.

780 Anthony Berglas, Cryptsoft Pty Ltd.

781 Mr. Dieter Bong, Utimaco IS GmbH

782 Hamish Cameron, nCipher

783 Kenli Chong, QuintessenceLabs Pty Ltd.

784 Mr. Justin Corlett, Cryptsoft Pty Ltd.

785 Mr. Tony Cox, TC Logic/Tony Cox

786 Ms. Valerie Bubb Fenwick, Apple

787 Ms. Susan Gleeson, Oracle

788 Tim Hudson, Cryptsoft Pty Ltd. | OpenSSL Software Services Inc.

789 Mr. Gershon Janssen, Reideate

790 Mr. Jakup Jelen, Red Hat

791 Mr. Darren Johnson, THALES

792 Sun-Ho Lee, MDS Intelligence Inc

793 John Leiseboer, QuintessenceLabs Pty Ltd.

794 Mr. John Leser, Oracle

795 Scott Leubner, THALES

796 Scott Marshall, Cryptsoft Pty Ltd.

797 Dr. Michael Markowitz, Information Security Corporation

798 Mr. Darren Moffat, Oracle

799 Mr. Tim Ober, THALES

800 Dr. Florian Poppa, QuintessenceLabs Pty Ltd.

801 Mr. Robert Relyea, Red Hat

802 Mr. Jonathan Schulze-Hewett, Information Security Corporation

803 Mr. Greg Scott, Cryptsoft Pty Ltd.

804 Mr. Martin Shannon, QuintessenceLabs Pty Ltd.

805 Mr. Oscar So, Individual

806 Simo Sorce, Red Hat

807 Mr. Manish Upasani, Utimaco IS GmbH

808 Ms. Magda Zdunkiewicz, Cryptsoft Pty Ltd.

809

Appendix C. Revision History

810

Revision	Date	Editor	Changes Made
WD01	23 Sep-2023	Tim Hudson	Initial Draft
WD02	16 Jul 2024	Tim Hudson	Updated XML encoding approach to avoid use of XML element attributes for structure fields to improve readability of the XML.
WD03	31 Jul 2024	Tim Hudson	Additional typos and cross reference errors and formatting corrected based on review feedback.
WD04	15 Apr 2025	Dieter Bong	Updated link to [PKCS11_Spec]; updated Appendix B Acknowledgements

811

812 Appendix D. Notices

813 Copyright © OASIS Open 2025. All Rights Reserved.

814 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
815 Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website:
816 [<https://www.oasis-open.org/policies-guidelines/ipr/>].

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

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

827 This document and the information contained herein is provided on an "AS IS" basis and OASIS
828 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
829 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
830 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
831 PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT,
832 INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS
833 DOCUMENT OR ANY PART THEREOF.

834 As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards
835 Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

836 [OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
837 necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS
838 TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims
839 in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this
840 deliverable.]

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

846 [OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
847 might be claimed to pertain to the implementation or use of the technology described in this OASIS
848 Standards Final Deliverable or the extent to which any license under such rights might or might not be
849 available; neither does it represent that it has made any effort to identify any such rights. Information on
850 OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS
851 Technical Committee can be found on the OASIS website. Copies of claims of rights made available for
852 publication and any assurances of licenses to be made available, or the result of an attempt made to
853 obtain a general license or permission for the use of such proprietary rights by implementers or users of
854 this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS
855 makes no representation that any information or list of intellectual property rights will at any time be
856 complete, or that any claims in such list are, in fact, Essential Claims.]

857 The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this document, and should be
858 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
859 implementation and use of, documents, while reserving the right to enforce its marks against misleading
860 uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark/> for above guidance.