

Integračná příručka

D.Signer/XAdES Java - XML Plugin, v2.0

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.208	Verzia 5

Copyright

Všetky práva vyhradené

Tento dokument je vlastníctvom spoločnosti DITEC, a. s. Žiadna jeho časť sa nesmie akýmkoľvek spôsobom (elektronickým, mechanickým) poskytnúť tretej strane, rozmnožovať, kopírovať, vrátane spätného prevodu do elektronickej podoby, bez písomného povolenia spracovávateľa.

Popisné charakteristiky dokumentu

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Podnázov	D.Signer/XAdES Java - XML Plugin, v2.0	
Ref. číslo	GOV_ZEP.208	Verzia 5

Vypracoval	Vittek Robert	Podpis	Dátum 27. 12. 2022
Preveril		Podpis	Dátum
Schválil		Podpis	Dátum

Formulár	Dokument		
Ref. číslo	Fo 11	Dátum poslednej aktualizácie	Dátum 14. 10. 2005

Akceptované dňa : <Dátum akceptácie>

Za <Objednávateľa>:

Za <Dodávateľa>.::

<Meno zodpovednej osoby>

<Meno zodpovednej osoby >

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.208	Verzia 5

Záznamy o zmenách

Autor	Popis zmien	Dátum	Verzia

Pripomienkovanie a kontrola

Autor	Stanovisko	Dátum	Verzia

Rozdeľovník

	Priezvisko Meno	Firma, Funkcia
Originál		
Kópia		
Kópia		
Kópia		

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.208	Verzia 5

Obsah

1.	Úvod	6
2.	Zoznam použitých skratiek	7
3.	Referencie	8
4.	Formát XML.....	10
5.	Architektúra XML Pluginu	13
5.1.	Postavenie komponentu v rámci prevádzkového prostredia a aplikácie D.Signer/XAdES Java	13
5.2.	Funkčná dekompozícia komponentu	14
6.	Špecifikácia funkčnosti	15
6.1.	Popis činnosti	15
7.	Špecifikácia API.....	16
7.1.	Integračné API pluginu	16
7.1.1.	Popis funkcií a premenných API pluginu	22
7.1.1.1.	createObject (triedy XmlPlugin, XmlPluginApplet).....	22
7.1.1.2.	createObject2 (triedy XmlPlugin, XmlPluginApplet).....	23
7.1.1.3.	createObject (triedy XmlBpPlugin, XmlBpPluginApplet)	23
7.1.1.4.	createObject2 (triedy XmlBpPlugin, XmlBpPluginApplet)	24
7.1.1.5.	getErrorMessage	25
7.1.1.6.	getXMLDataContainer	25
7.1.2.	Trieda XmlDataContainer	25
7.1.2.1.	initialize	25
7.1.2.2.	initialize2.....	25
7.1.2.3.	getErrorMessage	26
7.1.2.4.	getXdcXMLData	26
7.1.2.5.	getXdcIdentifier	26
7.1.2.6.	getXdcVersion	26
7.1.2.7.	getXdcUsedXSD	26
7.1.2.8.	getXdcUsedXSLT	26
7.1.2.9.	getXsdReferenceURI	26
7.1.2.10.	getXsdReferenceTransformAlg	27

Projekt	GOV_ZEP	A3019_002
Dokument	Integračná príručka	
Referencia	GOV_ZEP.208	Verzia 5

7.1.2.11.	getXsdReferenceDigestMethod	27
7.1.2.12.	getXsdReferenceDigestValue	27
7.1.2.13.	getXslReferenceURI	27
7.1.2.14.	getXslReferenceTransformAlg	27
7.1.2.15.	getXslReferenceDigestMethod	27
7.1.2.16.	getXslReferenceDigestValue	28
7.1.2.17.	getXslMediaDestinationTypeDescription.....	28
7.1.2.18.	getXslXSLTLanguage	28
7.1.2.19.	getXslTargetEnvironment.....	28
7.1.2.20.	getXdcIncludeRefs	28
7.1.2.21.	getXdcNamespaceURI.....	29

8. **Návratové kódy XML pluginu30**

1. Úvod

Tento dokument popisuje funkcionálnosť a integračné API komponentu D.Signer/XAdES Java – XML Plugin a tvorí prílohu Integrovačnej príručky aplikácie D.Signer/XAdES Java.

Aplikácia D.Signer/XAdES Java predstavuje riešenie pre vytváranie zaručeného/kvalifikovaného elektronického podpisu (ZEP/KEP) vo formátoch XAdES_ZEP/XAdES_ZEPbp nad množinou rôznych formátov dokumentov, resp. typov dát (XML dokumenty, PDF dokumenty atď.), prípadne nad ľubovoľnou kombináciou podporovaných formátov dát, ktoré spolu vytvárajú tzv. *multipart* dokument. Funkcionálnosť SCA je v rámci aplikácie D.Signer/XAdES Java rozdelená do samostatných modulov, ktoré je možné pri nasadení aplikácie kombinovať podľa požiadaviek zákazníka. Aplikáciu D.Signer/XAdES Java tvorí sada knižníc, ktoré poskytujú pre klientské aplikácie nasledujúce integračné rozhrania:

- Java applet API – umožňuje volanie služieb komponentu D.Signer/XAdES Java priamo z prostredia webového prehliadača,
- Java API – umožňuje volanie služieb komponentu D.Signer/XAdES Java z Java aplikácií bežiacich v JRE.

Aby bolo možné postupne budovať podporu pre ďalšie typy dátových objektov, medzi hlavným modulom D.Signer/XAdES Java a pluginmi bolo takisto navrhnuté abstraktné API, ktoré musí každý plugin implementovať. Hlavný modul komunikuje s jednotlivými pluginmi prostredníctvom tohto rozhrania. Architektúra aplikácie D.Signer/XAdES Java je podrobne popísaná v Integrovačnej príručke D.Signer/XAdES Java.

Každý plugin aplikácie D.Signer/XAdES Java musí pre typ dátového objektu, pre ktorý je určený, definovať triedu, ktorá predstavuje integračné API pluginu. Všeobecné požiadavky na integračné API pluginov, ktoré vyplývajú z architektúry aplikácie D.Signer/XAdES Java, sú definované v Integrovačnej príručke D.Signer/XAdES Java. Trieda integračného API pluginu môže navyše poskytovať svojmu okoliu ďalšie metódy a atribúty, ktoré sú špecifické pre príslušný typ podporovaného dátového objektu.

2. Zoznam použitých skratiek

DHC – Data Hashing Component

DTBS – Data To Be Signed

DTBSF – Data To Be Signed Formatted, resp. komponent Data To Be Signed Formatter

DTBSR – Data To Be Signed Representation

KEP – kvalifikovaný elektronický podpis

NBÚ – Národný bezpečnostný úrad

QSCD – Qualified Signature Creating Device

SAC – Signer's Authentication Component

SCA – Signature Creation Application

SCDev – Signature Creating Device

SCVA – Signature Creation and Validation Application

SDOC – Signed Data Object Composer

SDP – Signer's Document Presentation

SIC – Signer Interaction Component

SLC – Signature Logging Component

SSA – SCDev/SCA Authenticator

SSC – SCDev/SCA Communicator

SSCD – Secure Signature Creating Device

SVA – Signature Validation Application

XML – eXtended Markup Language

XSD – XML Schema Definition

XSL – eXtensible Stylesheet Language

XSLT – XSL Transformation

XAdES – XML Advanced Electronic Signatures

XAdES_ZEP – profil formátu elektronického podpisu XAdES pre ZEP

XAdES_ZEPbp – profil formátu kvalifikovaného elektronického podpisu na báze XAdES baseline profile

ZEP – Zaručený elektronický podpis

3. Referencie

- [1] W3C/IETF Recommendation: "XML-Signature Syntax and Processing" v2002-02-12 (XMLDSIG)
- [2] ETSI TS 101 733 – CMS Advanced Electronic Signatures (CAAdES) v2.2.1
- [3] ETSI TS 101 903 – XML Advanced Electronic Signatures (XAdES) v1.4.2
- [4] RFC 3125 – Electronic Signature Policies
- [5] RFC 3161 – Internet X.509 Public Key Infrastructure Time-Stamp Protocol
- [6] RFC 3279 – Algorithms and Identifiers for the Internet X.509 PKI
- [7] RFC 5280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
- [8] RFC 3548 – The Base16, Base32, and Base64 Data Encodings
- [9] RFC 5652 – Cryptographic Message Syntax (CMS)
- [10] RFC 4051 – Additional XML Security Uniform Resource Identifiers
- [11] Zákon č. 215/2002 Z.z. o elektronickom podpise a o zmene a doplnení niektorých zákonov v znení neskorších predpisov
- [12] Vyhláška NBÚ č. 131/2009 Z.z. o certifikátoch a kvalifikovaných certifikátoch
- [13] Vyhláška NBÚ č. 134/2009 Z.z. o produktoch elektronického podpisu
- [14] Vyhláška NBÚ č. 135/2009 Z.z. o vyhotovení a overovaní elektronického podpisu a časovej pečiatky
- [15] Vyhláška NBÚ č. 136/2009 Z.z. o spôsobe a postupe používania elektronického podpisu v obchodnom styku a administratívnom styku
- [16] NBÚ Formáty certifikátov a kvalifikovaných certifikátov, v4.0 (2014-07-10)
- [17] NBÚ Formáty zoznamu zrušených kvalifikovaných certifikátov, v3.0 (2010-01-17)
- [18] NBÚ Formáty zaručených elektronických podpisov, v3.0 (2009-08-12)
- [19] NBÚ Upresnenia obsahu a formálne špecifikácie formátov dokumentov pre ZEP, v1.0 (2007-07-24)
- [20] Zákon č. 272/2016 Z.z. o dôveryhodných službách
- [21] CWA 14170:2004 E – Security requirements for signature creation applications
- [22] CWA 14171:2004 D/E/F – General guidelines for electronic signature verification
- [23] XMLENC – XML Encryption Syntax and Processing", J. Reagle, D. Eastlake, December 2002. <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>
- [24] Konceptia všeobecného formátu XML podpisu a aplikácie SCVA, DITEC, a.s., 2006

- [25] Profil XAdES_ZEP – formát ZEP na báze XAdES, v1.0, DITEC, a.s., 2008
- [26] Profil XAdES_ZEP – formát ZEP na báze XAdES, v1.1, DITEC, a.s., 2009
- [27] Profil XAdES_ZEP – formát ZEP na báze XAdES, v2.0, DITEC, a.s., 2011
- [28] Profil XAdES_ZEPbp – formát ZEP na báze XAdES baseline profile, v1.0, DITEC, a.s., 2016
- [29] Formát dátových objektov typu XML dokument v rámci profilu XAdES_ZEP, v1.0, DITEC, a.s., 2013
- [30] Formát dátových objektov typu XML dokument v rámci profilu XAdES_ZEP, v2.0, DITEC, a.s., 2013
- [31] Rozhodnutie komisie 2011/130/EU, ktorým sa ustanovujú minimálne požiadavky na cezhraničné spracovanie dokumentov elektronicky podpísaných príslušnými orgánmi v zmysle smernice Európskeho parlamentu a Rady 2006/123/ES o službách na vnútornom trhu
- [32] Rozhodnutie komisie 2014/148/EÚ zo 17. marca 2014, ktorým sa mení rozhodnutie 2011/130/EÚ, ktorým sa ustanovujú minimálne požiadavky na cezhraničné spracovanie dokumentov elektronicky podpísaných príslušnými orgánmi v zmysle smernice Európskeho parlamentu a Rady 2006/123/ES o službách na vnútornom trhu
- [33] Nariadenie Európskeho Parlamentu a Rady EÚ č. 910/2014 z 23. júla 2014 o elektronickej identifikácii a dôveryhodných službách pre elektronické transakcie na vnútornom trhu a o zrušení smernice 1999/93/ES
- [34] Rozhodnutie komisie 2015/1506/EU, ktorým sa ustanovujú špecifikácie týkajúce sa formátov zdokonalených elektronických podpisov a zdokonalených elektronických pečatí, ktoré môžu subjekty verejného sektora uznávať
- [35] ETSI TS 103 171 – Electronic Signatures and Infrastructures (ESI) XAdES Baseline Profile, v2.1.1
- [36] ETSI TS 102 918 – Electronic Signatures and Infrastructures (ESI);. Associated Signature Containers (ASiC), v1.3.1
- [37] ETSI TS 103 174 – Electronic Signatures and Infrastructures (ESI);. ASiC Baseline Profile, v2.2.1
- [38] Zákon č. 305/2013 o elektronickej podobe výkonu pôsobnosti orgánov verejnej moci a o zmene a doplnení niektorých zákonov (zákon o e-Governmente) v znení neskorších predpisov
- [39] Integračná príručka D.Signer/XAdES Java, v2.0, DITEC, a.s., 2016
- [40] Extensible Markup Language (XML) 1.0 (Fifth Edition) – <http://www.w3.org/TR/2008/REC-xml-20081126/>
- [41] Výnos MF SR č. 55/2014 Z.z. o štandardoch pre informačné systémy verejnej správy v znení neskorších predpisov
- [42] Požiadavky na prezentácie XML dokumentov pre podpisovanie, DITEC, a.s., 2015

4. Formát XML

Formát XML v súčasnosti predstavuje rozšírený a podporovaný štandard pre elektronickú komunikáciu a výmenu dát medzi rôznymi systémami v heterogénnych prostrediach, pričom umožňuje jednoznačnú definíciu štruktúry, jednoznačnú interpretáciu obsiahnutých údajov, ako aj ich jednoduché automatické spracovanie.

Jednou z výhod XML formátu elektronického dokumentu je možnosť vyjadrenia štruktúry a definovania údajových typov (jednoduchých aj komplexných) pomocou XML schémy (XSD – <http://www.w3.org/XML/Schema/>).

Na základe definovanej XML schémy je možné automaticky overovať správnosť štruktúry dokumentu. Správna štruktúra XML dokumentu je základnou požiadavkou pre akceptovanie podpísaného dokumentu príjemcom. Správnosť dokumentu umožňuje jeho ďalšie automatizované spracovanie (záruka správnosti štruktúry a typu obsahu) a tiež korektnú vizualizáciu obsahu dokumentu. Preto je vhodné požadovať overenie správnosti štruktúry dokumentu pred samotným podpisom.

Zodpovednosť za vydávanie a zverejňovanie aktuálnych XML schém je na správcovi príslušného komunikačného scenára, v rámci ktorého sa požaduje spracovanie XML dokumentov podpísaných zaručeným/kvalifikovaným elektronickým podpisom. Zodpovednosť za použitie dôveryhodnej a správnej XML schémy je na podpisovateľovi príslušného XML dokumentu.

Z týchto dôvodov je v rámci profilov XAdES_ZEP [25][26][27] potrebné podpisovaný XML dokument opatriť doplňujúcou podpísanou informáciou, ktorá deklaruje XML schému použitú pre overenie správnosti štruktúry podpísaného XML dokumentu pred samotným vytvorením elektronického podpisu.

Overovateľ elektronického podpisu musí overiť použitie správnej XML schémy pre overenie správnosti štruktúry podpísaného XML dokumentu.

Zákon č. 215/2002 Z.z. [11] definuje požiadavku zobrazenia (vizualizácie) podpísaného elektronického dokumentu podpisovateľovi ešte predtým, ako sa spustí procedúra na vyhotovenie zaručeného/kvalifikovaného elektronického podpisu. XML dokument obsahuje štruktúrované dáta, ktoré sú vo väčšine prípadov pre bežného používateľa nečitateľné, preto je vhodné realizovať samotné zobrazenie XML dokumentu podpisovateľovi pomocou transformácie XML dokumentu do čitateľnej formy. Príloha č. 3 Výnosu MF SR o štandardoch pre informačné systémy verejnej správy [41] požaduje, aby povinnou prezentáciou pre podpisovanie a pre iný spôsob autorizácie elektronického formulára (ďalej len „podpisová prezentácia“) bol formát HTML alebo XHTML, a ak ide o elektronické formuláre s viac ako 50 procesnými krokmi, prezentáciou pre podpisovanie môže byť aj formát Plain Text Format (.txt) v kódovaní UTF-8, pričom môžu existovať aj ďalšie podpisové prezentácie v iných formátoch. V prezentačnej schéme sa pre transformáciu dátových prvkov do prezentácie vo formátoch HTML, XHTML alebo TXT používa jazyk XSL Transformations 1.0 (XSLT).

Na vyjadrenie ľubovoľnej transformácie XML dokumentu existuje štandard pre XML transformácie (XSLT – <http://www.w3.org/TR/xslt/>), pomocou ktorého možno definovať pravidlá pre transformáciu XML dokumentu do požadovaného formátu. Možnosti XSLT sú pomerne rozsiahle, preto pravidlá transformácie podpisovaného XML dokumentu musia byť definované tak, aby zaručili zobrazenie úplného obsahu XML dokumentu v zrozumiteľnej forme.

Najjednoduchšou alternatívou pre zobrazenie obsahu XML dokumentu (vzhľadom na technickú realizovateľnosť, aj vzhľadom na dostatočnú vypovedaciu schopnosť) je jednoduchý textový formát (Plain Text). Vizualizácia do TXT umožňuje zároveň realizovať vizualizáciu aj extrémne veľkých XML dokumentov (rádovo desaťtisíce strán), pri ktorých by vizualizácia do HTML, XHTML mohla byť v súčasnosti technicky problematická kvôli pamäťovým nárokom.

Vizualizácia do HTML (XHTML) na druhej strane poskytuje oveľa väčšie možnosti formátovania XML dokumentu a priblíženie sa vzhľadu pôvodného formulára. Pri vizualizácii do HTML je však potrebné zvážiť použitie takých HTML prvkov, ktoré môžu spôsobiť odchýlky v zobrazení výslednej HTML prezentácie XML dokumentu v rôznych prehliadačoch. Takisto je potrebné riešiť možnosť použitia aktívneho kódu vo výslednej HTML prezentácii (javascript, applety), ako aj referencovanie externého obsahu (CSS štýly, obrázky), ktorý nie je súčasťou podpisovaných dát a ktorého zmena môže významne ovplyvniť HTML prezentáciu XML dokumentu podpisovateľovi.

Požiadavky na prezentácie XML dokumentov pre podpisovanie sú mimo rámca tejto špecifikácie a sú definované v samostatnom dokumente [42]. Základom pre úplné a správne zobrazenie dokumentu pred podpisovaním je validácia XML dokumentu vzhľadom na XML schému.

Zodpovednosť za vydávanie a zverejňovanie aktuálnych, úplných a správnych transformačných schém na prezentáciu XML dokumentov pre podpisovanie je na správcovi komunikačného scenára, v rámci ktorého sa požaduje spracovanie XML dokumentov podpísaných zaručeným/kvalifikovaným elektronickým podpisom. Zodpovednosť za použitie dôveryhodnej a správnej prezentácie XML pre podpisovanie je na podpisovateľovi príslušného XML dokumentu.

Podpísaný XML dokument je teda v rámci profilov XAdES_ZEP [25][26][27] potrebné opatriť tiež doplňujúcou podpísanou informáciou, ktorá deklaruje XML prezentáciu použitú pre zobrazenie obsahu podpisovaného XML dokumentu pred samotným vytvorením elektronického podpisu.

Overovateľ elektronického podpisu musí overiť použitie správnej XML prezentácie pre zobrazenie obsahu podpísaného XML dokumentu.

V rámci profil XAdES_ZEPbp [28] sú podpisované XML údaje (elektronického formulára, skr. e-formulára) zahrnuté do štruktúry XMLDataContainer [41], ktorá predstavuje obálku pre spojenie XML údajov e-formulára spolu s XML schémou podpisovaných XML dát e-formulára a jeho prezentačnou schémou do povolených formátov – HTML, XHTML, resp. TXT. Použité schémy môžu byť do štruktúry XMLDataContainer vložené buď priamo, alebo referencované pomocou jednoznačných referencií. Štruktúra XMLDataContainer navyše definuje ďalšie

atribúty jednotlivých spomenutých prvkov (napr. jazyk prezentačnej schémy, cieľové prostredie, pre ktoré je prezentačná schéma určená, atď.), ktorých vyhodnotenie môže byť relevantné pri spracovaní podpísaných XML údajov elektronického formulára.

5. Architektúra XML Pluginu

V rámci tejto kapitoly je popísaná architektúra XML Pluginu pre aplikáciu D.Signer/XAdES Java, ktorá vychádza z dokumentov:

- Konceptia všeobecného formátu XML podpisu a aplikácie SCVA [24]
- CWA14170:2004 E – Security requirements for signature creation applications [21].

5.1. Postavenie komponentu v rámci prevádzkového prostredia a aplikácie D.Signer/XAdES Java

XML Plugin pre aplikáciu D.Signer/XAdES Java je realizovaný ako samostatný komponent, ktorý môže byť nasadený ako súčasť aplikácie D.Signer/XAdES Java v rámci rozsiahlejších systémov, napr. pre elektronickú výmenu dokumentov medzi rôznymi subjektami, v rámci ktorých je potrebné zabezpečiť:

- jednoznačnú identifikáciu pôvodcu dokumentu a neodmietnuteľnosť autorstva,
- integritu (prenášaných) dokumentov.

V rámci aplikácie D.Signer/XAdES Java zabezpečuje XML Plugin činnosti potrebné pre spracovanie a vizualizáciu dát typu XML dokument pred spustením procedúry vytvorenia ZEP/KEP a vytvorenie príslušných XML štruktúr pre formát podpisu v súlade s profilmi XAdES_ZEP/XAdES_ZEPbp.

Komponent XML Plugin poskytuje pre klientské aplikácie nasledujúce integračné rozhrania – API:

- integračné Java API – umožňuje volanie služieb komponentu XML plugin z Java aplikácií,
- integračné Java applet API – umožňuje volanie služieb komponentu XML Plugin priamo z prostredia webového prehliadača.

Pre interakciu s podpisovateľom komponent XML Plugin poskytuje GUI rozhranie, v rámci ktorého je realizované:

- zobrazenie obsahu podpisovaných XML dokumentov,
- zobrazenie obsahu verifikačných údajov pre podpisované XML dokumenty (XML schéma, XSL transformácia, prípadne typ výslednej vizualizácie pre podpisovanie),
- zobrazenie ostatných relevantných parametrov ZEP/KEP (napr. použité algoritmy pre digitálne odtlačky a ich hodnoty)

pred spustením procedúry vytvorenia ZEP/KEP.

Komponent XML Plugin zároveň poskytuje implementáciu abstraktného API rozhrania pre integráciu s aplikáciou D.Signer/XAdES Java, ktoré je definované v rámci dokumentu Integrovaná príručka D.Signer/XAdES Java [39].

Komponent XML Plugin nevykonáva kryptografické operácie ani nekomunikuje s SSCD zariadením. Pre tento účel volá funkcie rozhrania samostatnej knižnice, ktorá takisto tvorí súčasť aplikácie D.Signer/XAdES Java.

5.2. Funkčná dekompozícia komponentu

Vnútoraná architektúra komponentu XML Plugin pre D.Signer/XAdES Java vychádza a je v súlade s funkčným komponentovým modelom dokumentu CWA14170:2004 E – Security requirements for signature creation applications [21].

Z pohľadu funkčného komponentového modelu SCA sú v rámci komponentu XML Plugin pre D.Signer/XAdES Java implementované nasledujúce dôveryhodné komponenty:

- SDP – Signer's Document Presentation Component – zabezpečuje zobrazenie podpisovaných XML dokumentov podpisovateľovi, kontrolu výslednej vizualizácie XML dokumentov na nepovolený obsah,
- SAV – Signature Attributes Viewer – zabezpečuje zobrazenie príslušných verifikačných údajov pre XML dokument a ďalších atribútov vytváraného ZEP/KEP podpisovateľovi,
- DTBSF – Data To Be Signed Formatter – zabezpečuje sformátovanie a transformáciu vstupných XML dokumentov a ďalších parametrov podpisu do kanonickej formy a vytvorenie štruktúry DTBSF,
- SIC – Signer Interaction Component – GUI rozhranie pre vizualizáciu XML dokumentov, príslušných verifikačných údajov a ďalších atribútov ZEP/KEP a pre interakciu medzi podpisovateľom a aplikáciou D.Signer/XAdES Java.

XML Plugin pre D.Signer/XAdES Java obsahuje nasledujúce aplikačne závislé komponenty všeobecnej architektúry SCA:

- SDOC – Signed Data Object Composer – modul pre vytvorenie príslušných XML fragmentov výsledného ZEP/KEP vo formáte XAdES_ZEP, resp. XAdES_ZEPbp zo vstupných XML dokumentov a ďalších vstupných parametrov.

Obrázok funkčnej dekompozície aplikácie D.Signer/XAdES Java na jednotlivé komponenty SCA ako aj pohľad na jednotlivé vrstvy architektúry sa nachádza v dokumente Integrovaná príručka D.Signer/XAdES Java [39], kapitola 6.2.

6. Špecifikácia funkčnosti

6.1. Popis činnosti

Komponent XML Plugin pre aplikáciu D.Signer/XAdES Java zabezpečuje nasledujúce činnosti:

- vytvorenie dátového objektu typu XML dokument pre aplikáciu D.Signer/XAdES Java,
- spracovanie vstupných dátových objektov typu XML dokument, validáciu voči XML schéme a aplikovanie príslušných transformácií pre vytvorenie DTBSF (kanonikalizácia, XSLT),
- vizualizácia XML dokumentu, príslušných verifikačných údajov (XML schémy, XSL transformácie, prípadne typu výslednej vizualizácie pre podpisovanie) a ďalších atribútov vytváraného ZEP/KEP podpisovateľovi,
- podľa typu výslednej vizualizácie zároveň vykoná validáciu výslednej vizualizácie XML dokumentu nasledovne:
 - ⇒ vizualizácia do plain text (TXT) – kontrola na nepovolené znaky (viď povolené znaky pre entitu Char – <http://www.w3.org/TR/2008/REC-xml-20081126/#charsets> [40]),
 - ⇒ vizualizácia do HTML – kontrola nepovolených HTML tagov: applet, script, iframe, link, object, embed, audio, video.
- v prípade, že výsledná vizualizácia obsahuje nepovolený obsah, tak sa XML dokument nezobrazí a aplikácia vypíše upozornenie,
- vytvorenie príslušných fragmentov výslednej štruktúry ZEP/KEP podľa profilu XAdES_ZEP a prílohy Formát dátových objektov typu XML dokument, resp. podľa profilu XAdES_ZEPbp a špecifikácie štruktúry XMLDataContainer a ich poskytnutie aplikácii D.Signer/XAdES Java.

Popis činnosti komponentu v rámci aplikácie D.Signer/XAdES Java je špecifikovaný v rámci dokumentu Integračná príručka D.Signer/XAdES Java [39], kapitola 7.

7. Špecifikácia API

Komponent XML Plugin pre D.Signer/XAdES Java tvorí JAR knižnica, ktorá pre klientské aplikácie poskytuje nasledujúce integračné rozhrania:

- Java API – umožňuje volanie služieb komponentu XML plugin z Java aplikácií,
- Java applet API – umožňuje volanie služieb komponentu XML Plugin priamo z prostredia webového prehliadača.

Princípy návrhu integračných rozhraní Java API a Java applet API sú popísané v rámci integračnej príručky SCA aplikácie D.Signer/XAdES Java [39], kapitoly 8.1.1 a 8.1.2.

XML Plugin definuje v rámci integračného API triedy pre typ dátového objektu XML dokument, ktoré reprezentujú:

- podpísovaný XML dokument,
- verifikačné údaje pre daný XML dokument – XML schéma a XSL transformácia, prípadne typ výslednej vizualizácie pre podpisovanie – a ich atribúty.

XML Plugin pre D.Signer/XAdES Java implementuje abstraktné API IPlugin pre komunikáciu s hlavnou aplikáciou D.Signer/XAdES Java.

V nasledujúcich kapitolách je popísané integračné rozhranie XML Pluginu.

7.1. Integračné API pluginu

XML Plugin pre aplikáciu D.Signer/XAdES Java publikuje pre Java aplikácie nasledujúce rozhranie:

Package:

sk.ditec.zep.dsigner.xades.plugins.xmlplugin

Triedu:

XmlPlugin

Metódy a premenné:

```
public DataObject createObject
(
    String objectId
,   String objectDescription
,   String sourceXml
,   String sourceXsd
,   String namespaceUri
,   String xsdReference
,   String sourceXsl
,   String xslReference
);

public DataObject createObject2
(
    String objectId
,   String objectDescription
,   String sourceXml
,   String sourceXsd
,   String namespaceUri
,   String xsdReference
,   String sourceXsl
,   String xslReference
,   String visualTransformType
);

public static final String VISUAL_TRANSFORM_TXT = "TXT";
public static final String VISUAL_TRANSFORM_HTML = "HTML";

public String getErrorMessage();
```

Package:

sk.ditec.zep.dsigner.xades.bp.plugins.xmlplugin

Triedu:

XmlBpPlugin

Konštanty:

```
public static final String MEDIA_DESTINATION_TYPE_DESC_TXT = "TXT";
public static final String MEDIA_DESTINATION_TYPE_DESC_HTML = "HTML";

public static final String XDC_NAMESPACE_URI_V1_0 =
    "http://data.gov.sk/def/container/xmldatacontainer+xml/1.0";
public static final String XDC_NAMESPACE_URI_V1_1 =
    "http://data.gov.sk/def/container/xmldatacontainer+xml/1.1";
```

Metódy a premenné:

```
public DataBpObject createObject
(
    String objectId
,   String objectDescription
,   String objectFormatIdentifier
,   String xdcXMLData
,   String xdcIdentifier
,   String xdcVersion
,   String xdcUsedXSD
,   String xsdReferenceURI
,   String xdcUsedXSLT
,   String xslReferenceURI
,   String xslMediaDestinationTypeDescription
,   String xslXSLTLanguage
,   String xslTargetEnvironment
,   final boolean xdcIncludeRefs
,   String xdcNamespaceURI
);

public DataBpObject createObject2
(
    String objectId
,   String objectDescription
,   String objectFormatIdentifier
,   String xdcXDCB64
,   String xdcUsedXSD
,   String xdcUsedXSLT
);

public String getErrorMessage();

public XMLDataContainer getXMLDataContainer();
```

Package:

```
sk.ditec.zep.dsigner.xades.bp.plugins.xmlplugin
```

Triedu:

```
XmlDataContainer
```

Metódy a premenné:

```
public int initialize(String xdcDoc);
public int initialize2(String xdcDocB64);
public String getErrorMessage();

public String getXdcXMLData();
public String getXdcIdentifier();
public String getXdcVersion();
public String getXdcUsedXSD();
public String getXdcUsedXSLT();
public String getXsdReferenceURI();
public String getXsdReferenceTransformAlg();
public String getXsdReferenceDigestMethod();
public String getXsdReferenceDigestValue();
public String getXslReferenceURI();
public String getXslReferenceTransformAlg();
public String getXslReferenceDigestMethod();
public String getXslReferenceDigestValue();
public String getXslMediaDestinationTypeDescription();
public String getXslXSLTLanguage();
public String getXslTargetEnvironment();
public boolean getXdcIncludeRefs();
public String getXdcNamespaceURI();
```

XML Plugin pre aplikáciu D.Signer/XAdES Java bude pre webové aplikácie využívajúce Java applet API publikovať nasledujúce rozhranie:

Package:

```
sk.ditec.zep.dsigner.xades.plugins.xmlplugin.applet
```

Triedu:

```
XmlPluginApplet
```

Konštanty:

```
public static final String VISUAL_TRANSFORM_TXT = "TXT";
public static final String VISUAL_TRANSFORM_HTML = "HTML";
```

Metódy a premenné:

```
public boolean createObject
(
    final Object objectId
,
    final Object objectDescription
,
    final Object sourceXml
,
    final Object sourceXsd
,
    final Object namespaceUri
,
    final Object xsdReference
,
    final Object sourceXsl
,
    final Object xslReference
,
    final JSObject callback
);
```

```
public boolean createObject2
(
    final Object objectId
,
    final Object objectDescription
,
    final Object sourceXml
,
    final Object sourceXsd
,
    final Object namespaceUri
,
    final Object xsdReference
,
    final Object sourceXsl
,
    final Object xslReference
,
    final Object visualTransformType
,
    final JSObject callback
);

public boolean getErrorMessage(final JSObject callback);
```

Package:

sk.ditec.zep.dsigner.xades.bp.plugins.xmlplugin.applet

Triedu:

XmlBpPluginApplet

Konštanty:

```
public static final String VISUAL_TRANSFORM_TXT = "TXT";
public static final String VISUAL_TRANSFORM_HTML = "HTML";

public static final String XDC_NAMESPACE_URI_V1_0 =
    "http://data.gov.sk/def/container/xmldatacontainer+xml/1.0";
public static final String XDC_NAMESPACE_URI_V1_1 =
    "http://data.gov.sk/def/container/xmldatacontainer+xml/1.1";
```

Metódy a premenné:

```
public boolean createObject
(
    final Object objectId
    ,
    final Object objectDescription
    ,
    final Object objectFormatIdentifier
    ,
    final Object xdcXMLData
    ,
    final Object xdcIdentifier
    ,
    final Object xdcVersion
    ,
    final Object xdcUsedXSD
    ,
    final Object xsdReferenceURI
    ,
    final Object xdcUsedXSLT
    ,
    final Object xslReferenceURI
    ,
    final Object xslMediaDestinationDescription
    ,
    final Object xslXSLTLanguage
    ,
    final Object xslTargetEnvironment
    ,
    final boolean xdcIncludeRefs
    ,
    final Object xdcNamespaceURI
    ,
    final JSObject callback
);

public boolean createObject2
(
    final Object objectId
    ,
    final Object objectDescription
    ,
    final Object objectFormatIdentifier
    ,
    final Object xdcXDCB64
    ,
    final Object xdcUsedXSD
    ,
    final Object xdcUsedXSLT
    ,
    final JSObject callback
);

public boolean getErrorMessage(final JSObject callback);

public XMLDataContainerAppletWrapper getXMLDataContainer();
```

Package:

```
sk.ditec.zep.dsigner.xades.bp.plugins.xmlplugin.applet
```

Triedu:

```
XMLDataContainerAppletWrapper
```

Metódy a premenné:

```
public boolean initialize(final Object xdcDoc, final JSObject callback);
public boolean initialize2
(      final Object xdcDocB64
,      final JSObject callback
);
public boolean getErrorMessage(final JSObject callback);

public boolean getXdcXMLData(final JSObject callback);
public boolean getXdcIdentifier(final JSObject callback);
public boolean getXdcVersion(final JSObject callback);
public boolean getXdcUsedXSD(final JSObject callback);
public boolean getXdcUsedXSLT(final JSObject callback);
public boolean getXsdReferenceURI(final JSObject callback);
public boolean getXsdReferenceTransformAlg(final JSObject callback);
public boolean getXsdReferenceDigestMethod(final JSObject callback);
public boolean getXsdReferenceDigestValue(final JSObject callback);
public boolean getXslReferenceURI(final JSObject callback);
public boolean getXslReferenceTransformAlg(final JSObject callback);
public boolean getXslReferenceDigestMethod(final JSObject callback);
public boolean getXslReferenceDigestValue(final JSObject callback);
public boolean getXslMediaDestinationTypeDescription
(      final JSObject callback
);
public boolean getXslXSLTLanguage(final JSObject callback);
public boolean getXslTargetEnvironment(final JSObject callback);
public boolean getXdcIncludeRefs(final JSObject callback);
public boolean getXdcNamespaceURI(final JSObject callback);
```

7.1.1. Popis funkcií a premenných API pluginu

7.1.1.1. createObject (triedy XmlPlugin, XmlPluginApplet)

Umožňuje vytvoriť dátový objekt typu XML dokument v1.0 v rámci profilu XAdES_ZEP pre aplikáciu D.Signer/XAdES Java.

Parametre:

objectId – XML Id daného objektu v rámci výslednej XML štruktúry podľa XAdES_ZEP, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník); XML Id musí začínať písmenom alebo podčiarkovníkom,

objectDescription – popis obsahu daného XML objektu, napr: "DPPO 2007",

sourceXml – samotný vstupný XML dokument,

sourceXsd – XML schéma pre vstupný XML dokument,

namespaceUri – namespace URI vstupného XML dokumentu,

xsdReference – URI referencia XML schémy,

sourceXsl – XSL transformácia vstupného XML dokumentu do plain textu,

xslReference – URI referencia XSL transformácie,

callback – vid' dokument [39], kapitola 8.1.2.

Všetky podpisované informácie o dátovom objekte budú pred vytvorením podpisu zobrazené používateľovi a pri overení podpisu budú overené voči referenčným údajom evidovaným v rámci IS spracovateľa elektronického podpisu.

7.1.1.2. createObject2 (triedy XmlPlugin, XmlPluginApplet)

Umožňuje vytvoriť dátový objekt typu XML dokument v2.0 v rámci profilu XAdES_ZEP pre aplikáciu D.Signer/XAdES Java.

Parametre:

objectId – XML Id daného objektu v rámci výslednej XML štruktúry podľa XAdES_ZEP, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník); XML Id musí začínať písmenom alebo podčiarkovníkom,

objectDescription – popis obsahu daného XML objektu, napr: "DPPO 2007",

sourceXml – samotný vstupný XML dokument,

sourceXsd – XML schéma pre vstupný XML dokument,

namespaceUri – namespace URI vstupného XML dokumentu,

xsdReference – URI referencia XML schémy,

sourceXsl – XSL transformácia vstupného XML dokumentu od plain textu, HTML/XHTML,

xslReference – URI referencia XSL transformácie,

visualTransformType – typ výslednej vizualizácie XML dokumentu, povolené hodnoty sú "TXT", "HTML",

callback – vid' dokument [39], kapitola 8.1.2.

Všetky podpisované informácie o dátovom objekte budú pred vytvorením podpisu zobrazené používateľovi a pri overení podpisu budú overené voči referenčným údajom evidovaným v rámci IS spracovateľa elektronického podpisu.

7.1.1.3. createObject (triedy XmlBpPlugin, XmlBpPluginApplet)

Umožňuje vytvoriť dátový objekt typu XML v rámci profilu XAdES_ZEPbp pre aplikáciu D.Signer/XAdES Java. V rámci profilu XAdES_ZEPbp bude podpísaná štruktúra XMLDataContainer [41].

Parametre:

objectId – názov súboru s dátovým objektom v štruktúre XMLDataContainer (odporúča sa názov vrátane prípony .xml); zakázané sú znaky < > : " / \ | ? *,

objectDescription – popis obsahu daného XML objektu, napr: "DPPO 2007", môže byť null

objectFormatIdentifier – jednoznačný identifikátor verzie elektronického formulára, môže byť null

xdcXMLData – samotný vstupný XML dokument; dáta elektronického formulára,

xdcIdentifier – jednoznačný identifikátor elektronického formulára,

xdcVersion – verzia elektronického formulára,

`xdcUsedXSD` – XML schéma elektronického formulára,
`xsdReferenceURI` – URI XML schémy elektronického formulára,
`xdcUsedXSLT` – prezentačná schéma elektronického formulára,
`xslReferenceURI` – URI prezentačnej schémy elektronického formulára,
`xslMediaDestinationTypeDescription` – typ výslednej vizualizácie XML dokumentu, povolené hodnoty sú konštanty `MEDIA_DESTINATION_TYPE_DESC_TXT` a `MEDIA_DESTINATION_TYPE_DESC_HTML`,
`xslXSLTLanguage` – jazyk použitej prezentačnej schémy,
`xslTargetEnvironment` – cieľové prostredie, pre ktoré je prezentačná schéma určená,
`xdcIncludeRefs` – `true`, ak majú byť do štruktúry `XMLDataContainer` zahrnuté referencie schém prostredníctvom elementu `UsedSchemasReferenced`; `false`, ak majú byť do štruktúry `XMLDataContainer` zahrnuté hodnoty schém prostredníctvom elementu `UsedSchemasEmbedded`,
`xdcNamespaceURI` – namespace XML schémy pre štruktúru `XMLDataContainer`, povolené hodnoty `XDC_NAMESPACE_URI_V1_0` a `XDC_NAMESPACE_URI_V1_1`,
`callback` – vid' dokument [39], kapitola 8.1.2.

Všetky podpisované informácie o dátovom objekte budú pred vytvorením podpisu zobrazené používateľovi a pri overení podpisu budú overené voči referenčným údajom evidovaným v rámci IS spracovateľa elektronického podpisu.

7.1.1.4. createObject2 **(triedy XmlBpPlugin, XmlBpPluginApplet)**

Umožňuje vytvoriť dátový objekt typu XML v rámci profilu `XAdES_ZEPbp` pre aplikáciu `D.Signer/XAdES Java`. V rámci profilu `XAdES_ZEPbp` bude podpísaná tá istá štruktúra `XMLDataContainer` [41], ktorá je obsahom vstupného parametra `xdcXDCB64`.

Parametre:

`objectId` – názov súboru s dátovým objektom v štruktúre `XMLDataContainer` (odporúča sa názov vrátane prípony `.xml`); zakázané sú znaky `< > : " / \ | ? *`,
`objectDescription` – popis obsahu daného XML objektu, napr: "DPPO 2007", môže byť `null`
`objectFormatIdentifier` – jednoznačný identifikátor verzie elektronického formulára, môže byť `null`
`xdcXDCB64` – vstupná štruktúra `XMLDataContainer` [41] obsahujúca aj dáta elektronického formulára,
`xdcUsedXSD` – XML schéma elektronického formulára; je potrebná v prípade, ak je vo vstupnej štruktúre `XMLDataContainer` uvedená len referencia XML schémy elektronického formulára; inak môže byť prázdny reťazec,

xdcUsedXSLT – prezentačná schéma elektronického formulára; je potrebná v prípade, ak je vo vstupnej štruktúre XMLDataContainer uvedená len referencia prezentačnej schémy elektronického formulára; inak môže byť prázdny reťazec, callback – vid' dokument [39], kapitola 8.1.2.

Všetky podpisované informácie o dátovom objekte budú pred vytvorením podpisu zobrazené používateľovi a pri overení podpisu budú overené voči referenčným údajom evidovaným v rámci IS spracovateľa elektronického podpisu.

7.1.1.5. getErrorMessage

V prípade výskytu chyby v rámci procesu vytvárania dátového objektu typu XML dokument bude vracať príslušnú chybovú správu uloženú v premennej ErrorMessage.

Parametre:

callback – vid' dokument [39], kapitola 8.1.2

7.1.1.6. getXMLDataContainer

Vráti objekt triedy XMLDataContainer, resp. XMLDataContainerAppletWrapper vytvorený na základe parametrov volania metódy createObject triedy XmlBpPlugin, resp. XmlBpPluginApplet.

7.1.2. Trieda XmlDataContainer

Trieda XmlDataContainer slúži na vyparsovanie informácií z už existujúcej štruktúry XMLDataContainer. Túto triedu je možné využiť napr. v prípade vytvárania viacerých nezávislých elektronických podpisov nad tými istými dátami elektronického formulára na získanie parametrov pre volanie metódy CreateObject triedy XmlBpPlugin.

7.1.2.1. initialize

Inicializácia triedy XmlDataContainer.

Parametre:

string xdcDoc – XML dokument v štruktúre XMLDataContainer ako textový reťazec,

callback – vid' dokument [39], kapitola 8.1.2.

Návratová hodnota:

V prípade úspechu 0; inak chybový kód.

7.1.2.2. initialize2

Inicializácia triedy XmlDataContainer.

Parametre:

string xdcDocB64 – XML dokument v štruktúre XMLDataContainer kódovaný v base64,

callback – vid' dokument [39], kapitola 8.1.2.

Návratová hodnota:

V prípade úspechu 0; inak chybový kód.

7.1.2.3. getErrorMessage

V prípade neúspechu inicializácie triedy XMLDataContainer vráti príslušnú chybovú správu.

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.4. getXdcXMLData

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu XMLData (dáta elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.5. getXdcIdentifier

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu XMLData/Identifier (jednoznačný identifikátor elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.6. getXdcVersion

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu XMLData/Version (verzia elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.7. getXdcUsedXSD

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedXSDEmbedded (XML schéma elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.8. getXdcUsedXSLT

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedPresentationSchemaEmbedded (prezentačná schéma elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.9. getXsdReferenceURI

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedXSDReference (URI XML schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.10. getXsdReferenceTransformAlg

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedXSDReference/TransformAlgorithm (URI použitého kanonikalizačného algoritmu pred výpočtom digitálneho odtlačku XML schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.11. getXsdReferenceDigestMethod

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedXSDReference/DigestMethod (URN OID použitého algoritmu digitálneho odtlačku pre výpočet hodnoty referencie XML schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.12. getXsdReferenceDigestValue

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedXSDReference/DigestValue (hodnota digitálneho odtlačku referencie XML schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.13. getXslReferenceURI

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu elementu UsedPresentationSchemaReference (URI prezentačnej schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.14. getXslReferenceTransformAlg

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaReference/TransformAlgorithm (URI použitého kanonikalizačného algoritmu pred výpočtom digitálneho odtlačku prezentačnej schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.15. getXslReferenceDigestMethod

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaReference/DigestMethod (URN OID použitého

algoritmu digitálneho odtlačku pre výpočet hodnoty referencie prezentačnej schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.16. getXslReferenceDigestValue

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaReference/DigestValue (hodnota digitálneho odtlačku referencie prezentačnej schémy elektronického formulára).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.17. getXslMediaDestinationTypeDescription

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaEmbedded/MediaDestinationTypeDescription, resp. UsedPresentationSchemaReference/MediaDestinationTypeDescription (cieľové prostredie, pre ktoré je prezentačná schéma určená).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.18. getXslXSLTLanguage

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaEmbedded/Language, resp. UsedPresentationSchemaReference/Language (jazyk použitej prezentačnej schémy).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.19. getXslTargetEnvironment

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu UsedPresentationSchemaEmbedded/TargetEnvironment, resp. UsedPresentationSchemaReference/TargetEnvironment (cieľové prostredie, pre ktoré je prezentačná schéma určená).

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.20. getXdcIncludeRefs

V prípade úspechu inicializácie triedy XMLDataContainer vráti true, ak boli do štruktúry XMLDataContainer zahrnuté referencie schém prostredníctvom elementu UsedSchemasReferenced; false, ak boli do štruktúry XMLDataContainer zahrnuté hodnoty schém prostredníctvom elementu UsedSchemasEmbedded.

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

7.1.2.21. getXdcNamespaceURI

V prípade úspechu inicializácie triedy XMLDataContainer vráti hodnotu atribútu xmlns elementu XMLDataContainer.

Parametre:

callback – vid' dokument [39], kapitola 8.1.2.

8. Návrátové kódy XML pluginu

V nasledujúcej tabuľke sú uvedené návratové kódy funkcií initialize a initialize2 tried XMLDataContainer a XMLDataContainerAppletWrapper.

Návratový kód	Popis
0	Inicializácia prebehla úspešne.
-1	Odchytená výnimka v aplikácii. Popis chyby je možné získať pomocou funkcie getErrorMessage.

Ostatné funkcie vrátia v prípade chyby prázdny string, resp. hodnotu Null (v závislosti od typu návratovej hodnoty).