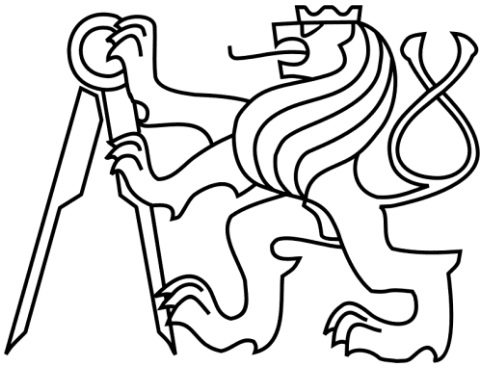
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



Bakalářská práce

# Analýza vizuální integrity digitálně podepsaného PDF dokumentu

Ondřej Suchý

Vedoucí práce: doc. Ing. Zdeněk Kouba, CSc.

Studijní program: Otevřená informatika (bakalářský)
Obor: Softwarové systémy

3. 1. 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Ondřej Suchý**

Studijní program: Otevřená informatika (bakalářský)
Obor: Softwarové systémy

Název tématu: **Analýza vizuální integrity digitálně podepsaného PDF dokumentu**

Pokyny pro vypracování:

1) Seznamte se se specifikací formátu PDF.
2) Nalezněte konstrukce, které mohou způsobovat odlišné zobrazení digitálně podepsaného dokumentu.
3) Vytvořte testovací PDF dokumenty, které budou demonsnrovat jednotlivé konstrukce nalezené v bodě 2).
4) Navrhněte a implementujte program, který analyzuje PDF dokument a vyhledá v něm konstrukce specifikované v bodě 2) a upozorní přehlednou formou uživatele na jejich výskyt.
5) Funkci programu ověřte na dokumentech vytvořených v bodě 3).
6) Do textu bakalářské práce zařaďte specifikaci konstrukcí nalezených v bodě 2).

Seznam odborné literatury:

[1] PDF Reference, sixth edition, Adobe Portable Document Format, ver. 1.7, http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_1-7.pdf, Adobe Systems Incorporated, 2006, citováno 22.11.2013
[2] Digital Signiture User Guide for Acrobat 9.0 and Adobe Reader 9.0, http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/acrobat_digsig_userguide_90.pdf, Adobe Systems Incorporated, Nov 2008, citováno 22.11.2013

Vedoucí: doc.Ing. Zdeněk Kouba, CSc.

Platnost zadání: do konce letního semestru 2014/2015


doc. Ing. Filip Železný, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 25. 2. 2014

## Poděkování

V první řadě bych chtěl poděkovat doc. Ing. Zdeňku Koubovi, CSc. za vedení mé bakalářské práce a celého projektu zaměřeného na možnosti kontroly vizuální integrity PDF dokumentů, za cenné rady a konstruktivní kritiku. Dále děkuji společnosti Software602 a.s. za poskytnuté podklady k formátu PDF, zapůjčení potřebného softwaru a neobvyklé příklady PDF dokumentů a v neposlední řadě bych rád poděkoval své rodině za poskytnutou podporu a motivaci.

## Abstract

The focus of this thesis is a visual integrity of digitally signed documents in PDF format, i.e. properties of PDF document, which can cause inconsistent visual appearance of such document depending on external factors. Firstly there is a brief description of inner structure of the PDF documents and the explanation of main principles of digital signatures. Secondly there is a result of the analysis, i.e. the description of document properties, which could have an effect on the visual integrity of digitally signed document. Subsequently there is a description of a utility which can detect document properties that can have an impact on document visual integrity and warn the user.



## Key words:

Portable document format, digital signature, visual integrity



## Abstrakt

Tato práce se zaměřuje na vizuální integritu digitálně podepsaných dokumentů ve formátu PDF, tj. vlastnosti PDF dokumentů, které mohou způsobit nekonzistentní zobrazení v závislosti na externích faktorech. Nejdříve je čtenáři nastíněna vnitřní struktura PDF dokumentů a princip fungování elektronického podpisu. Následuje výsledek samotné analýzy, kde jsou popsány jednotlivé vlastnosti, které by mohly mít vliv na vizuální integritu digitálně podepsaného dokumentu. Dále je zde popsán nástroj naprogramovaný v rámci této práce, který má za úkol vyhledávat dané problémové vlastnosti, které by mohli ovlivňovat vizuální integritu dokumentu, a varovat před nimi uživatele.



## Klíčová slova:

Formát PDF, elektronický podpis, vizuální integrita

## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a že jsem uvedl veškeré informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 3. 1. 2015 .....

Ondřej Suchý

# Obsah

# Seznam tabulek

# Seznam příkladů

# 1 Úvod

Portable Document Format (PDF, česky přenosný formát dokumentu) je všeobecně přijímaný standard pro dokumenty s fixním formátováním, který umožňuje konzistentní zobrazení dokumentů nezávisle na hardwaru, softwaru a operačním systémů, a tím uživatelům významně zjednodušuje sdílení dokumentů napříč různými platformami. Dokument by měl být zobrazen stejně na stolním počítači s velkým monitorem, na přenosném počítači s malým displejem, tabletu i mobilním telefonu. Konzistence zobrazení je vykoupena tím, že vzhled a rozložení stránky není při čtení přizpůsobeno jednotlivým platformám na rozdíl od například HTML formátu, u něhož může být rozložení stánky přizpůsobeno displeji použitého zařízení nebo velikosti okna použitého programu.

Dokumenty v tomto formátu se neomezují na zobrazení statických stránek, ale mohou obsahovat i různé dynamické prvky od JavaScriptů, přes multimédia až po externí obsah načítaný například z internetu. Ovšem dynamické a externí prvky mohou být zobrazeny nekonzistentně v závislosti na mnoha faktorech. Odlišný vzhled dokumentu může být způsoben například nedostupností použitých fontů na daném počítači, různým přiblížením dokumentu nebo aplikací, která je použita k prohlížení nebo tisku. Dokument může být jinak zobrazen na monitoru počítače a jinak vytištěn. Vzhled dokumentu se může měnit i v závislosti na čase nebo v průběhu vyplňování polí dokumentu.

Formát PDF je dnes používán například i pro různé právní dokumenty, jako jsou smlouvy případně dokumenty pro komunikaci s úřady. Takto podepsané dokumenty mohou být snadno přenášeny například pomocí elektronické pošty. V tomto případě by fakt, že dokument je jinak zobrazen na monitoru a jinak vytištěn, mohl vést k problémům. Elektronický podpis pomocí kryptografických nástrojů garantuje pravost dat, tj. že data nebyla cestou ke čtenáři změněna, zároveň slouží i k autentizaci, lze s jeho pomocí ověřit identitu autora. Ovšem samotné zajištění pravosti binární podoby dokumentu nemusí být dostatečné, vzhled elektronicky podepsaného dokumentu se může měnit, část dokumentu může být překryta jiným prvkem, použití jiného písma by mohlo teoreticky vést až ke změně textu. Z tohoto důvodu některé aplikace jako například Adobe Acrobat provádějí před podepsáním kontrolu dokumentů a uživatele varují před možnou nekonzistencí při zobrazení dokumentu.

Přestože je formát PDF definován v otevřené specifikaci [1], která popisuje syntaxi a zobrazení PDF dokumentů, specifikace prvků ovlivňujících vizuální integritu v oficiálně dostupných dokumentech popsána není. Popis problémových vlastností je jeden z cílů této práce spolu s demonstrací provedení kontroly vizuální integrity ukázkovým programem, který na základě této analýzy vznikl. Samotná analýza problémových prvků probíhala v rámci projektu pro společnost Software602 a za podpory projektu Akcelerace, který zastřešuje hlavní město Praha.

## 1.1 Terminologická poznámka

Je předpokládáno, že typickým čtenářem této práce je programátor, a proto se nesnažím o překládání anglických termínů do češtiny za každou cenu. Tam, kde výstižnosti textu prospěje běžně programátory používaný výraz převzatý z angličtiny, pro který neexistuje ustálený český překlad, zpráva používá tento anglicismus, byť by to byl výraz slangový. Příkladem může být slovo *escapování* namísto mnohoznačného českého ekvivalentu *kódování* nebo *slajd* namísto nejednoznačných českých překladů *snímek* nebo *průsvitka*.

# 2 Portable Document Format

V této kapitole je čtenář stručně uveden do specifikace formátu PDF a je zde popsán princip fungování elektronického podpisu.

V prvních částech je vysvětlena použitá znaková sada PDF dokumentů, jejich syntaxe na úrovni objektů a struktura dokumentu jako celku. Je zde uveden stručný popis jednotlivých částí dokumentu. Tento popis formátu PDF přímo vychází z oficiální specifikace PDF verze 1.7[1].

V poslední sekci jsou čtenáři velmi stručně vysvětleny základní pojmy jako hash, asymetrické šifrování, princip fungování elektronických podpisů a jejich aplikace v PDF dokumentech.

## 2.1 Historie formátu PDF

Formát PDF byl vyvinut společností Adobe na začátku devadesátých let, zpočátku šlo o uzavřený formát a aplikace pro zápis i čtení PDF souborů nebyly volně dostupné. Nejenom to způsobilo, že rozšíření formátu bylo zpočátku pozvolné. Kromě nedostupnosti aplikací pro práci s dokumenty v tomto formátu, byl jeho nástup zpomalen i mnoha dalšími faktory. Zobrazení PDF dokumentů bylo pro tehdejší počítače poměrně náročné a vykreslování stránek mohlo na méně výkonných strojích probíhat velmi pomalu. Rané verze neobsahovaly podporu hypertextových odkazů, což omezovalo možnosti použití PDF v prostředí internetu.

Formát PDF před verzí 1.2 nepodporoval deflate kompresi obsahu datových proudů (streamů) ani linearizaci struktury dokumentů, tj. možnost zapisovat PDF dokumenty tak, aby mohly být zobrazeny stránky na začátku částečně staženého dokumentu, dokument bez linearizace musí být čten aplikací od konce souboru, kde se nachází tabulka referencí. Tato tabulka, tzv. cross-reference table, popisuje polohu jednotlivých objektů v rámci PDF dokumentu. Tabulka byla umístěna až za samotné objekty, aby byl zjednodušen zápis PDF dokumentu a bylo ho možné snadno provést při jediném průchodu dat, ze kterých byl PDF soubor generován. Od PDF verze 1.5 byla přidána podpora pro tzv. cross-reference stream, který může být na rozdíl od cross-reference table komprimovaný, a umožňuje odkazovat na objekty zkomprimované uvnitř object streamů, čímž může být větší část dokumentu komprimována. Chybějící linearizace a omezené možnosti komprese dokumentů bez cross-reference streamů v prvních verzí zpomalovaly zobrazení dokumentů stahovaných z internetu prostřednictvím tehdejšího připojení.

Později ovšem popularita formátu PDF strmě rostla i díky tomu, že Adobe se rozhodlo svoji aplikaci pro čtení PDF dokumentů – Acrobat Reader, v pozdějších verzích nazvaný Adobe Reader, nabízet ke stažení zdarma. PDF se stalo standardem pro výměnu dokumentů s fixním formátováním a jeho podpora se zlepšovala i v aplikacích mimo rodinu Adobe, například kancelářský balík Open Office nabízí export do PDF v základu bez nutnosti instalace doplňků už od verze 1.1, která vyšla již v roce 2003[10].

V roce 2008 formát PDF prošel procesem standardizace organizace ISO (International Organization for Standardization – Mezinárodní organizace pro normalizaci) a tím se z tohoto formátu stala oficiální norma ISO 32000-1:2008[1].

Formát PDF byl vyvíjen s ohledem na zpětnou i dopřednou kompatibilitu. Do jeho specifikace je možné snadno přidávat nové vlastnosti, klíče a objekty, které nejsou ve verzi formátu PDF používaného aplikací definovány, mají být aplikací, která soubor načítá, ignorovány. To umožňuje definovat různé proprietární rozšíření tohoto formátu tak, že v aplikacích bez tohoto rozšíření je možné soubor zobrazit, i když bez dané proprietární vlastnosti.

## 2.2 Struktura PDF souborů

PDF dokument je obvykle tvořený jediným souborem, který obsahuje vše potřebné pro své zobrazení. Výjimkou jsou dokumenty, kde je použitý externí obsah, jako například systémové fonty nebo obrázky načítané z internetu. Standard PDF umožňuje kompresi jak pomocí obecných algoritmů (Deflate, LZW) tak i specializovaných algoritmů pro kompresi obrazu (JPEG, JPEG2000). Jak již bylo zmíněno, PDF dokumenty jsou obvykle čteny odzadu s výjimkou linearizovaných souborů. Tento přístup usnadňuje zápis souborů. Tabulka nebo stream s offsety ukazujícími na umístění jednotlivých objektů rámci souboru (tj. cross-reference table nebo cross-reference stream) je umístěn na konec souboru. Tím je umožněno generování PDF souboru jedním průchodem, a to může být výhodné pro zařízení, která mají málo operační paměti a nemohou používat dočasné soubory.

Grafické funkce a vzhled stránek formátu PDF vychází z programovacího jazyka PostScript. Jde o tzv. Adobe imaging model, který je použitý pro vysokoúrovňové a na zařízení nezávislé generování stránek dokumentu. Základní grafické objekty formátu PDF jsou objekty textové, obrázkové a objekty popisující cestu. Objekty popisující cestu jsou tvořeny sekvencí propojených a oddělených bodů, úseček a křivek, s jejichž pomocí jsou popsány tvary a jejich pozice. Textové objekty (text objects) popisují jeden nebo více znaků a jsou popsány v datových strukturách nazývaných fonty (fonts). Obrázkové objekty (image objects) jsou obdélníkové bitmapy, obvykle jsou použity pro fotografie. Pro zaznamenání grafů, plánů a vektorové grafiky obecně je lepší použít popis pomocí objektů popisujících cestu. Dosáhneme tím hladkého zobrazení na všech typech zařízení, jejichž rozlišení se pohybuje od 75 až 110 pixelů na palec u počítačových monitorů po 2400 pixelů u fotografických tiskáren.

Soubory ve formátu PDF mohou být považovány za sekvenci bajtů, kde je každý bajt tvořen 8 bity. Přestože je možné zajistit, aby soubor obsahoval pouze bajty ze základní sedmibitové tabulky znaků ASCII, tj. znaky jejichž číselná reprezentace bez znaménka je menší než 128, není to doporučeno. Důvodem je větší velikost a možná chyba při přenosu těchto souborů. PDF soubory by vždy měly být přenášeny jako binární soubory, neměly by být měněny kvůli konvencím použité platformy či znakové sady. Převod mezi znakovými sadami či převod znaků značících konce řádků by mohl dokument poškodit. Proto je doporučeno, aby ihned po hlavičce dokumentu byl řádek s alespoň čtyřmi binárními znaky, tj. znaky jejichž číselná hodnota bez znaménka je větší než 127. Některé systémy analyzují začátek souboru, a pokud by znaky mimo základní tabulku ANSII nebyly přítomny, soubor by mohl být považován za textový, což by mohlo vést až k poškození souboru během přenosu. Formát PDF je case sensitive, tj. malá a velká písmena, např. „A“ a „a“ jsou považovány za dva odlišné znaky.

[1] http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502

Znaky použité v PDF souborech jsou rozděleny do třech skupin:

- Běžné znaky (regular characters)
- Oddělovací znaky (delimiter characters)
- White-space znaky (White-space characters)

White-space znaky se nacházejí mezi jednotlivými objekty v PDF souboru. S více po sobě jdoucími white-space znaky je aplikací, která čte PDF soubor, nakládáno jako s jedním white-space znakem, s výjimkou znaků v komentářích, řetězcích a datových proudech (streams).

**Tabulka 2.1 – White-space znaky**

| Zkratka | Jméno | Decimálně | Hexadecimálně |
|---|---|---|---|
| **NUL** | Null | 0 | 00 |
| **HT** | Tabulátor | 9 | 09 |
| **LF** | Line feed | 10 | 0A |
| **FF** | Form feed | 12 | 0C |
| **CR** | Carrige return | 13 | 0D |
| **SP** | Mezera | 32 | 20 |

Oddělovací znaky (delimiter characters) oddělují jednotlivé objekty v PDF souboru, pomocí nichž je vyznačen začátek a konec jednotlivých objektů.

**Tabulka 2.2 – Oddělovací znaky**

| Znak | Decimálně | Hexadecimálně | Použití |
|---|---|---|---|
| **(** | 40 | 28 | začátek stringu |
| **)** | 41 | 29 | konec stringu |
| **<** | 60 | 3C | začátek slovníku/ hexastringu |
| **>** | 62 | 3E | konec slovníku/ hexastringu |
| **[** | 91 | 5B | začátek pole |
| **]** | 93 | 5D | konec pole |
| **{** | 123 | 7B | |
| **}** | 125 | 7D | |
| **/** | 47 | 2F | names |
| **%** | 37 | 25 | komentáře |

Všechny ostatní osmibitové znaky kromě white-space a oddělovacích znaků spadají do skupiny normálních znaků.

### 2.2.1 Objekty

Tělo souborů ve formátu PDF je tvořeno jednotlivými objekty, rozlišujeme následující typy objektů:

- **Číselné objekty** (numbers) mohou být celočíselné nebo reálné, v souboru jsou zapsány jako znaky v desetinné formě.
- **Řetězce** (strings) mohou sloužit pro uchování textů nebo třeba formátovaných dat (např. čas, datum).
- **Name objekty** jsou nedělitelné unikátní identifikátory.
- **Boolean objekty** mohou mít pouze dvě hodnoty `true` a `false` (pravda a nepravda).
- **Pole** (arrays) jsou jednorozměrné kolekce po sobě jdoucích objektů. Pole mohou být heterogenní, tj. každá položka v poli může být objekt libovolného typu včetně vnořených polí, na

rozdíl od mnoha programovacích jazyků, kde pole mohou obsahovat pouze položky stejného typu.

- **Slovníky** (dictionaries) obsahují dvojice klíč-hodnota, kde klíč je vždy name objekt a hodnota může být libovolného typu včetně vnořeného slovníku.
- **Proudové objekty** (streams) obsahují data nebo vložené objekty. Mohou být komprimované, nejčastěji pomocí algoritmu deflate.
- **Null objekty** jsou objekty, které mají jinou hodnotu než všechny ostatní objekty.
- **Nepřímé objekty** (Indirect objects) slouží jako reference na objekt, který se nachází na jiném místě v PDF souboru.

#### *2.2.1.1 Číselné objekty*

Formát PDF nabízí dva typy číselných objektů (numeric objects), celočíselné (integer objects) a reálné (real objects), ve specifikaci může být kromě konkrétních typů (integer/real) definován typ obecně jako number. Specifikace PDF doporučuje v aplikacích, které čtou PDF soubory, pracovat s reálnými čísly pomocí fixed-point reprezentace, tj. reprezentace s daným počtem desetinných míst.

Čísla integer jsou zapsána jako jedno nebo více číslic zapsaných v desetinné soustavě. Před číslem může být přítomno znaménko. Pokud je číslo větší než dovoluje implementace dané architektury pro typ integer, tak je číslo převedeno na typ real.

**Příklad 2.1 – Čísla typu integer**

```
-2960 1454 0 13680
```

**Příklad 2.2 – Čísla typu real**

```
683.138 -3.14 +258.96 86. -.036 0.0
```

Pokud reálné číslo překračuje maximální nebo minimální velikost danou implementačním limitem pro reálná čísla v dané aplikaci, tak je vyvolána výjimka. Číslo typu integer je v případě potřeby převedeno na číslo typu real, tj. není nutné psát např. „`32.0`“, ale stačí napsat celé číslo „`32`“, i když je ve specifikaci uvedeno, že daná položka má být typu real.

#### *2.2.1.2 Řetězce*

Řetězce (strings) jsou série po sobě jdoucích znaků. Maximální délka řetězce v content streamu podléhá implementačnímu limitu, pokud je soubor otevřen v prohlížeči založeném na 32 bitové architektuře, je maximální délka řetězce 32 767 bajtů[2]. Řetězce se dají použít pro skladování různých údajů, například časů, dat nebo otisků (hashes). Rozlišujeme dva typy řetězců literal a hexadecimal strings.

##### 2.2.1.2.1 Literal strings

**Literal strings** jsou zapsané jako série znaků uzavřené v kulatých závorkách „`()`“. Mohou obsahovat libovolný znak, kromě nespárovaných kulatých závorek a zpětného lomítka. Znak zpětného lomítka slouží k escapování, tj. ke kódování řetězcových literálů, jako jsou například symbol nového řádku, nespárované kulaté závorky, znak zpětného lomítka a netisknutelné ASCII znaky. Znak nebo znaky, které následují za zpětným lomítkem, určují význam escapovací sekvence.

[2] Tabulka C1 Apendix C ve [1]

**Tabulka 2.3 – Escapování řetězců pomocí zpětného lomítka**

| Escapovací sekvence | Význam |
|---|---|
| **\n** | Nový řádek/Line feed (LF) |
| **\r** | Přesun na začátek řádku/Carrige return (CR) |
| **\t** | Horizontální tabulátor (HT) |
| **\b** | Backspace (BS) |
| **\f** | Form feed (FF) |
| **\(** | Levá závorka |
| **\)** | Pravá závorka |
| **** | Zpětné lomítko |
| **\ooo** | Libovolný znak, kde **ooo** je číselný zápis znaku v osmičkové soustavě, je možné zapsat pouze jeden nebo dvě osmičkové číslice, pokud znak za osmičkovou číslicí není osmičkové číslo. Pokud escapovaný znak je následován osmičkovým číslem, tak je nutné použít tři osmičkové číslice. |

### Příklad 2.3 – Řetězce (literal strings)

```
(Jednoduchy retezec)
(Retezec se (sparovanymi) zavorkami)
(Se specialnimi znaky *!&}^% nasledovany prazdnym retezcem)
()
(Retezec se symbolem
noveho radku)
(Retezec se symbolem\nnoveho radku)
(Predchozi 2 retezce jsou stejne)
(Retezec se zpetnym lomitkem \\ a tabulatorem \t)
(Escapovany symbol plus /053)
```

Pokud je řetězec příliš dlouhý je možné ho napsat na více řádků a použít zpětné lomítko k tomu, aby znak nového řádku nebyl do řetězce zahrnut.

```
(Toto je \
jednoradkovy retezec.)
```

#### 2.2.1.2.2 Hexadecimal strings

**Hexadecimal strings** jsou uzavřené ve špičatých závorkách, tj. symboly menší/větší než „<>“. Používají se pro zapsání binárních dat uvnitř PDF souborů. Jde o sekvenci znaků `0-9` a `A-F` nebo `a-f`.

### Příklad 2.4 – Hexadecimal string

```
<A9F0B6E39FC1BC12D6>
```

Každý bajt je zapsán pomocí dvou hexadecimálních číslic. White-space znaky jsou ignorovány. Pokud sekvence obsahuje lichý počet znaků, předpokládá se, že chybějící znak je 0. Tj. `<A1B>` je přečteno stejně jako `<A1B0>`.

#### *2.2.1.3 Name objekty*

Name objekty (jmenné objekty) jsou nedělitelné sekvence znaků, nemají vnitřní strukturu, tudíž nelze pracovat s jednotlivými znaky v name objektu. Tyto objekty nejsou určeny k prezentaci uživatelům PDF prohlížeče, slouží jako identifikátory. Name objekty, které se skládají ze stejných znaků, jsou považovány za stejné objekty, tj. jsou jednoznačně definované.

Name objekt je vždy uvozen lomítkem („/“), které ovšem není součástí samotného jména. Hned po lomítku může následovat jeden nebo více běžných znaků, tj. v samotném name objektu nesmí být white-space a oddělovací znaky. Name objekt může být také samotné lomítko. U name objektů je rozlišována velikost písmen, tj. name objekty jsou case sensitive. `/Name` a `/name` jsou považovány za dva různé objekty.

Maximální délka name objektu podléhá implementačnímu limitu. Specifikace formátu PDF doporučuje nepřekračovat délku 127 bajtů[3], s kódovanými znaky se zachází jako s jedním znakem tj. kódovaný znak mezery („`#20`“) je brán jako jeden znak/bajt.

Následuje několik příkladů validních name objektů:

**Příklad 2.5 – Name objekty**

```
/Name123
/DostDlouheJmenoJmenehoObjektu
/LzePouzit;Ruzne_Znaky-Mimo***OddelovaciAWhiteSpace
/1.2
/$$
/@pattern
/.notdef
/
```

V posledním řádku tohoto příkladu je vidět, že samotný znak lomítka je také platný name objekt.

Od verze PDF 1.2 je možné do name objektů zapsat libovolný znak kromě znaku null (tj. znak s číselnou hodnotou 0). Znaky s číselnou hodnotou menší než 33 (!) nebo naopak větší než 126 (~) je doporučeno zapisovat jako dvě hexadecimální číslice uvozené znakem křížku („#“), tímto způsobem lze zapsat i oddělovací a white-space znaky.

**Tabulka 2.4 – Zápis speciálních znaků v name objektech**

| Zápis | Význam |
|---|---|
| **/Jmeno#20s#20mezerou** | Jmeno s mezerou |
| **/KodovanyKrizek#23** | KodovanyKrizek# |
| **/nameObjectSe#28#29Zavorkami** | nameObjectSe()Zavorkami |
| **/KodovanyBeznyZnak#43** | KodovanyBeznyZnakC |

[3] Apendix C v referenci [1]

#### *2.2.1.4 Boolean objekty*

Boolean objekty mohou mít pouze hodnotu „**true**“ a „**false**“.

#### *2.2.1.5 Pole*

Pole je skupina po sobě jdoucích prvků. Může obsahovat prvky různých typů, tj. může být heterogenní. Pole v PDF jsou jednorozměrná, ale je možné je do sebe zanořovat, a tím zapsat pole, která mají více rozměrů.

**Příklad 2.6 – Pole**

```
[/NameObject 123 1.423 true (Retezec)]
```

#### *2.2.1.6 Slovníky*

Slovníky jsou hlavní stavební kameny PDF dokumentů. Slovník je uveden dvojicí symbolů menší než „<<“ a zakončen dvojicí symbolů větší než „>>“. Jsou používány pro strukturovaný zápis informací o prvcích, které tvoří PDF dokument, od jednotlivých stránek přes prvky formulářů a obrázky až po informace o fontech. Slovník je vlastně asociativní tabulka, která obsahuje páry klíč-hodnota. Klíč musí být name objekt. Hodnota může být libovolného typu, včetně vnořeného slovníku. Pokud má nějaká položka hodnotu `null`, tak je to stejné jako, kdyby daná položka ve slovníku nebyla přítomna vůbec. Žádné dvě položky ve slovníku by neměly mít stejný klíč.

**Příklad 2.7 – Slovník**

```
<< /Type /Priklad
   /Subtype /PrikladSlovniku
   /DesetinaPolozka 3.14
   /CelociselnaPolozka 10
   /Retezec (Obsah retezce)
   /VnorenySlovnik <</Polozka1 true
                     /Polozka2 (test)
                   >>
>>
```

Hodnota položky s klíčem `/Type` obvykle určuje, jaký typ objektu daný slovník popisuje. Pro podrobnější rozlišení jednotlivých objektů může být použita položka s klíčem `/Subtype`, které je často zkracováno na `/S`. Hodnota položek `/Type` a `/Subtype` je vždy name objekt.

#### *2.2.1.7 Streams*

Stream objekty jsou tvořeny slovníkem a samotnými daty streamu. Slovník popisuje data streamu, jejich délku a použité filtry. Data streamu jsou uzavřená mezi slova „`stream`“ a „`endstream`“.

Stream může obsahovat žádný, jeden nebo více bajtů. Po klíčovém slově `stream` by měl následovat znak konce řádku buď `CR` a `LF` nebo pouze `LF`, použití samotného `CR` není dovoleno.[4] Stream musí být indirect object, tj. nemůže být vložen do jiného objektu, ale musí být dostupný pomocí čísla objektu, přímo v těle PDF dokumentu. Délka streamů na rozdíl od stringů není omezena, proto je doporučeno streamy načítat postupně.

[4] Pokud by byl použitý samotný znak [CR] a steam by začínal znakem [LF], pak nebylo možné určit, zda znak [LF] patří do stringu, nebo značí konec řádku.

**Tabulka 2.5 – Položky stream slovníků**

| Klíč | Typ hodnoty | Použití |
|---|---|---|
| **Length** | integer | Počet bajtů mezi klíčovým slovem „`stream`[LF]“ nebo „`stream`[CR][LF]“ a klíčovým slovem „`endstream`“, jde o velikost zakódovaných dat. Konec řádku, který musí následovat po klíčovém slově „stream“, není součástí streamu a nezapočítává se do jeho délky. |
| **Filter** | name, pole | Filtr, který je aplikovaný na data streamu, například komprese dat nebo šifrování. Pokud je použito více filtrů, měly by být seřazeny v takovém pořadí, v jakém mají být aplikovány. |
| **DecodeParms** | slovník, pole | Slovník parametrů, které jsou potřeba pro použití filtru, nebo pole slovníků parametrů k použitým filtrům. Pokud nějaký filtr nemá parametr, nebo má všechny parametry nastaveny na výchozí hodnoty je možné tuto položku vynechat. |
| **F** | specifikace souboru | Soubor, který obsahuje data streamu. Data mezi `stream` a `endstream` jsou při použití této položky ignorována. |
| **FFilter** | name, pole | Filtr nebo filtry, které mají být použity při zpracovávání externích streamů. |
| **FDecodeParms** | slovník, pole | Parametr nebo parametry použité při zpracování externích streamů. |
| **DL** | integer | Dekódovaná délka streamu v bajtech, jde pouze o orientační hodnotu, při použití některých filtrů může být problém velikost dekódovaných dat předem odhadnout. |

**Příklad 2.8 - Stream**

```
2 0 obj
<</Length 64>>stream
  BT
  /F1 24 Tf
  1 0 0 1 250 390 Tm
  (Hello World)Tj
  ET
endstream
endobj
```

#### *2.2.1.8 Null objekty*

Null objekty se do PDF souborů zapisují klíčovým slovem `null`. Liší se od kteréhokoli objektu jiného typu, než je null. Nepřímý objekt, který nebyl v souboru nalezen, je považovaný na null objekt. Pokud má nějaká položka ve slovníku hodnotu null, tak je ignorována, tj. je s ní nakládáno stejně, jakoby nebyla ve slovníku vůbec přítomna.

#### *2.2.1.9 Nepřímé objekty*

Nepřímé objekty/Indirect objects jsou objekty, které mají jedinečný identifikátor objektu. Tento identifikátor může být použitý pro odkazování na objekt například z pole nebo může být použitý jako hodnota položky slovníku. Opakem nepřímých objektů jsou objekty zapsané přímo v mateřském objektu, které jedinečný identifikátor nemají.

Zápis nepřímého objektu začíná objektovým a generačním číslem následovaným klíčovým slovem „obj“, za ním se nachází samotný objekt a po něm následuje klíčové slovo „endobj“. **Číslo objektu (object number)** je kladné celé číslo. Objekty zapsané v PDF souboru mohou být číslovány v libovolném pořadí, ale často jsou čísla přidělována postupně od nejmenšího po největší. **Generační číslo (generation number)** značí verzi objektu. Generační číslo je vždy nezáporné. Pokud jde o nový soubor, za který nebyla připsána žádná nová verze dokumentu, tak mají všechny nepřímé objekty generační číslo 0. Při aktualizaci souboru mohou být na konec připsány nové objekty s generačním číslem vyšším než nula. Tomuto způsobu aktualizace se říká inkrementační update. Změny jsou připsány na konec souboru a původní verze souboru zůstává zachována.

Číslo objektu a generační číslo spolu tvoří jednoznačný identifikátor objektů v rámci PDF souboru.

**Příklad 2.9 – Nepřímé objekty**

```
...
6 0 obj
<</Metadata 7 0 R/Pages 5 0 R/Type/Catalog>>
endobj
7 0 obj
<</Length 3112/Subtype/XML/Type/Metadata>>stream
...metadata dokumentu v XML...
endstream
endobj
5 0 obj
<</Count 2/Kids[8 0 R 1 0 R]/MediaBox[0 0 612 792]/Type/Pages>>
endobj
1 0 obj
<</Contents 2 0 R/Parent 5 0 R/Resources 3 0 R/Type/Page>>
endobj
...
```

Příklad 2.9 ukazuje použití nepřímých objektů, Objekt `6 0` je kořenový element dokumentu, tzv. Documents catalog, odkazuje na objekt 7 0, který obsahuje metadata, a objekt 5 0, který zase odkazuje na jednotlivé stránky, objekt `8 0` (který v této ukázce není) a `1 0`. Objekt `1 0`, je odkazuje zpět na mateřský objekt položkou `/Parent`.

### 2.2.2 Komentáře

Znak %, který není v řetězci nebo streamu, uvozuje komentář. Všechny znaky za znakem procenta až do konce řádky tvoří komentář. Celý komentář je aplikací pracující s PDF soubory považován za jeden white-space znak.

Komentáře nemají v PDF souboru žádný sémantický význam. Výjimkou jsou speciální komentáře „%PDF−n.m“ na začátku souboru a „%%EOF“ na jeho konci. Písmena „n.m“ značí verzi PDF souboru. Běžné komentáře nemusí být při editaci zachovány.

**Příklad 2.10 – Komentář**

```
1 0 obj
(Toto je string)%Toto je komentar{/% stale komentar
endobj
```

## 2.3 Vnitřní uspořádání PDF souborů

Soubory ve formátu PDF se v nejjednodušším případě skládají ze čtyř částí: hlavičky, těla, tabulky referencí a traileru. Když je soubor aktualizován, tak se počet částí může zvýšit. Nová data mohou být připsána na konec souboru, tím je zachována původní verze. Zachování původní verze souboru je výhodné například v případě, kdy jsou k dokumentu připsány nějaké poznámky nebo jsou provedeny změny v dokumentu. Uživatel má stále možnost si zobrazit původní verzi, tj. tu která byla podepsána, a ověřit platnost podpisu v této verzi. Tento způsob aktualizace dokumentu se označuje jako incremental update.

Místo tabulky referencí (cross-reference table) může být od verze PDF 1.5 použitý stream s referencemi (cross-reference stream), který lze na rozdíl od tabulky komprimovat a může odkazovat na objekty uvnitř tzv. object streamů, to jsou streamy, ve kterých jsou vnořeny další objekty.

### 2.3.1 Linearizované PDF soubory

Speciálním případem jsou linearizované soubory, které nemusí být čteny odzadu, ovšem jejich struktura je složitější. Jsou uspořádány tak, aby bylo možné zobrazit stránky na začátku dokumentu co nejrychleji, i když je dokument teprve stahován.

### 2.3.2 Hlavička dokumentu

Hlavička dokumentu se nachází na jeho prvním řádku, obsahuje číslo verze specifikace, které dokument odpovídá. Příklad 2.11 ukazuje hlavičku aktuální verze PDF 1.7.

**Příklad 2.11 – Hlavička dokumentu**

```
%PDF-1.7
```

Díky zpětné kompatibilitě mohou aplikace podporující PDF verze 1.7 číst i soubory, které podléhají libovolné starší verzi specifikace formátu PDF, tj. verze 1.0, 1.1, 1.2, 1.3, 1.4, 1.5 a 1.6.

Od PDF verze 1.4 je možné do document's catalog dictionary vložit položku s klíčem `/Version`, která přepisuje verzi uvedenou v hlavičce dokumentu. To umožňuje provést inkrementální update dokumentu na vyšší verzi, než je verze původního dokumentu.

### 2.3.3 Tělo dokumentu

Tělo dokumentu se skládá z jednotlivých nepřímých objektů. Od PDF verze 1.5 mohou být nepřímé objekty vloženy do object streamů, čímž je umožněno je komprimovat, případně šifrovat.

#### *2.3.3.1 Object Streams*

Object stream může obsahovat většinu PDF objektů, s výjimkou dalších object streamů, objektů s generačním číslem větším než 1, slovníku s informacemi o šifrování dokumentu (viz položka `/Encrypt` ) a objektu, na který odkazuje položka `/Length`, jenž se nachází ve slovníku libovolného object streamu.

**Tabulka 2.6 – Položky specifické pro object stream dictionary**

| Klíč | Typ hodnoty | Použití |
|---|---|---|
| **Type** | name | Typ objektu, v object streamu musí být „`ObjStm`“. |
| **N** | integer | Počet objektů, které object stream obsahuje. |
| **First** | integer | Offset v dekódovaném streamu, na kterém se nachází na první objekt ve streamu. |
| **Extends** | integer | Reference na object stream, který je rozšiřován daným object streamem. Streamy mohou tvořit kolekci, jde o acyklický graf object streamů provázaných referencemi. |

Je vhodné do streamu dávat omezený počet objektů, které spolu souvisí, například jsou na stejné straně. Pokud by v object streamu bylo příliš mnoho objektů, které spolu nemají nic společného, mohl by se celý stream dekomprimovat kvůli jedinému objektu a ostatní objekty v streamu by byly dekomprimovány zbytečně a to by vedlo ke zbytečnému snížení výkonu. Pokud je ovšem souvisejících objektů mnoho nebo chceme později nějaké objekty přidat, je možné využít volitelnou položku `/Extends` v object stream dictionary.

Data uvnitř object streamu začínají páry celých čísel, kde první číslo z páru je vždy object number objektu uvnitř streamu a druhý člen páru je offset začátku daného objektu relativně k prvnímu objektu ve streamu, tj. offsetu udaném v položce `/First`.

Potom následují samotné PDF objekty, které nejsou uvozeny číslem objektu a klíčovým slovem `obj` a nejsou ukončeny pomocí `endobj` jako objekty mimo object stream. Díky absenci generačních čísel v object streamu musí tyto objekty mít toto číslo rovno nule. To samé platí pro samotný object stream. Nové verze object streamů a objektů v nich vnořených musí používat nová objektová čísla, nelze použít ta původní s vyšším generačním číslem.

**Příklad 2.12 – Object stream**

```
<</Filter/FlateDecode
  /First 5
  /Length 68
  /N 1
  /Type/ObjStm
>>stream
...komprimovaná data...
endstream
```

### 2.3.4 Cross-Reference Table

Tabulka referencí obsahuje informace o poloze nepřímých objektů v rámci dokumentu. Tím je umožněno k jednotlivým objektům přistupovat bez nutnosti načtení celého dokumentu.

Tabulka referencí je rozdělena na sekce. Soubor, který není linearizovaný a nebyl aktualizován, obsahuje pouze jednu sekci. Při každé aktualizaci souboru je přidána jedna sekce. Každá sekce je uvozena klíčovým slovem „`xref`“, po něm následuje jedna nebo více podsekcí.

První řádek podsekce obsahuje dvě čísla oddělená mezerou. První značí objektové číslo prvního objektu, který je součástí dané podsekce a druhé číslo je počet objektů v dané podsekci. Například „`32 3`“ značí, že první položka v této podsekci je objekt s objektovým číslem 32 a celkem jsou v této podsekci 3 objekty, tj. objekty 32, 33, 34. Tato podsekce by mohla vypadat například takto:

```
32 3
0000001000 00003 n
0000001500 00000 n
0000002300 00000 n
```

První řádka, jak už bylo zmíněno, značí, že na následujících třech řádcích se budou nacházet objekty s objektovými čísly 32, 33 a 34. První část druhé řádky (`0000001000`) obsahuje offset, kde začíná daný objekt. V tomto případě objekt s objektovým číslem 32 začíná na 1000 bajtu daného souboru. V druhé části (`00003`) položky každé subsekce je zapsané generační číslo objektu, v tomto případě je generační číslo 3, to značí, že toto číslo objektu již bylo použito. V poslední části každého záznamu v tabulce referencí je písmeno `n` nebo `f`. Písmeno `n` značí, že jde o právě používaný objekt a písmeno `f` označuje nepoužívaný objekt. Na třetím řádku je objekt 33, který začíná na offsetu 1500 a má generační číslo 0, tudíž je objektové číslo 33 použito poprvé. Na posledním řádku je objekt 34, který začíná na offsetu 2300 a generační číslo je také 0.

Pokud je záznam volný, tj. ve třetí části je použito písmeno `f`, tak první část není offset na začátku objektu, ale jde o objektové číslo příštího nepoužitého objektu. Nepoužívané objekty v PDF souboru tvoří spojový seznam, každý volný záznam obsahuje číslo příštího volného záznamu.

**Příklad 2.13 – Cross-reference table**

```
xref
0 7
0000000000 65535 f
0000000017 00000 n
0000000112 00000 n
0000000236 00000 n
0000000309 00000 n
0000000403 00000 n
0000000502 00000 n
```

První položka, tj. položka s objektovým číslem 0, je vždy volná a má generační číslo 65 535. Poslední volná položka, která je vlastně koncem tohoto spojového seznamu, odkazuje právě na první položku s číslem 0. Pokud je první část položky objektu 0 rovna „0000000000“, pak to značí, že v dokumentu není žádný volný objekt.

### 2.3.5 Trailer

Trailer slouží k tomu, aby aplikace, která čte PDF dokument, mohla rychle najít cross-reference tabulku/stream a důležité objekty, jako jsou trailer předchozí verze souboru před updatem, catalog dictionary, který odkazuje na ostatní objekty, které obsahují jednotlivé stránky a jejich obsah, případně slovník potřebný pro otevření šifrovaného dokumentu a další.

**Tabulka 2.7 – Položky stream slovníků**

| Klíč | Typ hodnoty | Použití |
|---|---|---|
| **Size** | integer | Celkové množství záznamů v cross-reference table. Hodnota musí být zapsána přímo ve slovníku, nemůže jít o nepřímý objekt. |
| **Prev** | integer | Offset, který určuje vzdálenost předchozí cross-referenc table od začátku souboru. |
| **Root** | slovník | Nepřímý objekt, který odkazuje na document's catalog dictionary. |
| **Encrypt** | slovník | Slovník, který obsahuje informace o šifrování, které bylo v dokumentu použito. Pokud tato položka není přítomna, znamená to, že doument není šifrovaný. |
| **Info** | slovník | Reference na Document Information Dictionary, obsahuje metadata daného dokumentu, jako jsou například informace o autorovy, jméno dokumentu, datum vytvoření/modifikace atd. |
| **ID** | pole | Pole dvou byte-stringů, které jednoznačně identifikují daný dokument pomocí tvz. file identifiers. |

**Příklad 2.14 – Trailer dokumentu**

```
trailer
<<
  /Size 32
  /Root 11 0 R
  /Info 30 0 R
>>
startxref
653434
%%EOF
```

### 2.3.6 Cross-reference stream

Cross-reference streamy se dají použít místo cross-reference table. Kromě toho, že mohou být komprimované, navíc mohou popisovat polohu objektů, které se nalézají uvnitř object streamů.

**Tabulka 2.8 – Položky specifické pro cross-reference stream**

| Klíč | Typ hodnoty | Použití |
|---|---|---|
| **Type** | name | Typ objektu, musí být `/XRef`. |
| **Size** | integer | Podobné položce `/Size` v trailer dictionary. Číslo o jedna větší než největší číslo v dané sekci nebo v daném update PDF dokumentu. |
| **Index** | pole | Pole, které obsahuje páry hodnot, první hodnota je první objektové číslo v podsekci a druhá hodnota je počet položek |

| | | |
|---|---|---|
| | | v dané podsekci. |
| **Prev** | integer | Offset odkazující na předchozí cross-reference stream. |
| **W** | pole | Pole `/W` určuje velikost jednotlivých polí v každé položce cross-reference streamu. Například `/W[1 2 0]` znamená, že první položka má velikost 1 bajt, druhá 2 bajty a třetí není přítomna, tudíž musí být použita výchozí hodnota. |

Položky v cross-reference streamu mají v PDF verze 1.5 vždy tři položky. Pokud nějaká položka chybí, tj. má velikost 0, tak je použita výchozí hodnota. Mohou být zapsány třemi způsoby, které jsou popsány níže:

- **Typ 0**
  - 1. pole
    - Typ položky, pro tento typ musí být nastaveno na 0. Používá se pro spojový seznam volných objektů. Jde o analogii položek `f` v cross-reference table.
  - 2. pole
    - Objektové číslo následujícího volného objektu.
  - 3. pole
    - Generační číslo objektu, které bude mít objekt, pokud bude dané objektové číslo znovu použito
- **Typ 1**
  - 1. pole
    - Typ položky, pro tento typ musí být nastaveno na 1. Typ 1 se používá pro objektová čísla, které jsou zrovna používána, ale samotná data objektu nejsou komprimována uvnitř object streamů. Jde o alternativu `n` položek v cross-reference table. Pokud má první pole velikost 0, tj. není v cross-reference stream přítomno, pak mají všechny položky typ 1.
  - 2. pole
    - Offset daného objektu od začátku PDF souboru.
  - 3. pole
    - Generační číslo objektu, výchozí hodnota je nula.
- **Typ 2**
  - 1. pole
    - Typ položky, pro tento typ musí mít hodnotu 2.
  - 2. pole
    - Objektové číslo object streamu ve kterém je daný objekt uložený, object streamy mají generační číslo vždy 0.
  - 3. pole
    - Index daného objektu v rámci object streamu.

## 2.4 Elektronický podpis PDF dokumentu

### 2.4.1 Úvod do problematiky

V této části je čtenáři stručně vysvětleno několik pojmů z oblasti kryptografie, které jsou důležité pro pochopení problematiky elektronického podepisování PDF dokumentů.

#### *2.4.1.1 Hash*

Hašování je jeden ze základních prvků elektronického podpisu. Princip hašování spočívá ve vytvoření relativně krátkého haše (otisku) pevné délky z libovolně dlouhých vstupních dat.

Hash se snaží data, z nichž vznikl, jednoznačně popisovat. Bohužel z principu není možné, aby existoval hash pevné délky pro každý libovolně dlouhý vstup. Musí existovat více vstupů, které jsou odlišné, ale jejich hash je stejný. Tomuto stavu se říká kolize. U dobrého hašovacího algoritmu musí být nalezení druhého vstupu k libovolnému vstupu nebo nalezení dvou libovolných vstupů, které dávají stejný hash tak výpočetně náročné, aby nebylo možné kolize v rozumném čase nalézt.

Druhou vlastností, kterou požadujeme po hašovacích algoritmech je to, aby byly jednosměrné, tj. je možné poměrně rychle spočítat hash ze vstupních dat, ale naopak získání původních dat z hashe musí být v rozumném čase neproveditelné.

Mezi hašovací funkce patří například velmi oblíbená MD5 [11], bohužel pro tuto funkci je znám postup nalezení kolizní funkce ve velmi krátkém čase [12], a proto se již nehodí pro použití při digitálním podepisování dokumentů a další kryptografické účely. Ovšem stále jde o používaný způsob kontroly integrity dat při přenosu například po síti, v tomto případě jde ovšem pouze o detekci chyb během přenosu. MD5 zpracovává vstupní data po blocích o 512 bitech a výsledný hash má délku 128 bitů.

Nástupcem hašovací funkce MD5 se stal algoritmus SHA-1, ovšem i u něho se objevily slabiny, které by mohli vést k prolomení v blízké budoucnosti [13]. Proto je doporučeno již dnes upřednostňovat hašovací funkce z rodiny algoritmů SHA-2.

Algoritmy z rodiny SHA-2 produkují haše s délkou 224 až 512 bitů a používají bloky o velikosti 512 nebo 1024 bitů. Struktura jednotlivých hašovacích algoritmů je velmi podobná, odlišují se především v délce výsledného haše.

#### *2.4.1.2 Šifrování*

Šifrování slouží k zabezpečení zpráv před neoprávněným čtením. Šifrovací algoritmy dělíme na symetrické a asymetrické.

Symetrické šifrování je založeno na použití jediného tajného klíče, který by měl znát pouze odesilatel a příjemce. Tento klíč je použitý pro šifrování i dešifrování. Problémové je zahájení komunikace, kdy si účastníci komunikace musí tajný klíč vyměnit. Výhodou symetrického šifrování je to, že bývá výrazně rychlejší než níže popsané šifrování asymetrické, které při komunikaci používá právě pro výměnu klíčů a potom se přejde na šifrování symetrické. Příklady symetrických šifer jsou například RC4, 3DES nebo AES.

Asymetrické šifrování je založené na použití dvou klíčů. Veřejný klíč může být volně dostupný a slouží k zašifrování zprávy. Naopak soukromý klíč se nesmí dostat k nikomu jinému než k příjemci dané zprávy. V praxi si odesilatel vyžádá od příjemce veřejný klíč, s tím zašifruje zprávu a pošle ji příjemci. Jenom příjemce má soukromý klíč a může si danou zprávu přečíst. Asymetrické šifrování je používané nejen pro šifrování, ale používá se i pro elektronický podpis. Nejznámější asymetrická šifra je pravděpodobně algoritmus RSA, který je založený na předpokladu, že rozložení velmi velkého čísla na součin prvočísel je velmi těžký problém, přestože násobení dvou velkých prvočísel je velmi lehká úloha.

### 2.4.2 Princip elektronického podpisu

Elektronický podpis je vlastně opakem asymetrického šifrování. Z dat dokumentu je vytvořen hash, ten se zašifruje privátním klíčem odesilatele a tím vznikne samotný podpis. Veřejný klíč nutný pro ověření podpisu je součástí certifikátu. Při verifikaci je z dokumentu stejným způsobem vytvořen hash a ten je porovnán s výstupem vzniklým po dešifrování podpisu pomocí veřejného klíče z certifikátu podepisujícího. Pokud se rovnají, je zaručeno, že dokument nebyl od podepsání změněn. Podpis je svázaný s konkrétní zprávou, verifikace proběhne úspěšně pouze nad určitými daty. Pokud je zpráva úspěšně verifikována nad veřejným klíčem konkrétního uživatele, tak je jisté, že k tvorbě podpisu byl použit soukromý klíč daného uživatele, ke kterému by měl mít přístup pouze on.

**Zaručený elektronický podpis** brání padělání zprávy, zajišťuje pravost, neměnitelnost zprávy a brání použití stejného podpisu pro jinou zprávu, ovšem není možné ověřit, že odesilatel uvedený v certifikátu opravdu existuje, údaje uvedené v certifikátu mohou být smyšlené nebo může certifikát obsahovat údaje osoby bez jejího vědomí. Tím pádem není zaručena nepopiratelnost, tj. uživatel může popřít podepsání zprávy, přestože jsou v certifikátu jeho údaje.

Problém nepopiratelnosti řeší **uznávaný elektronický podpis**, který na rozdíl od zaručeného zajišťuje skutečnou identitu držitele tohoto podpisu. Uznávaný certifikát je vydáván kvalifikovanými poskytovateli certifikačních služeb, v České republice mají nyní (v roce 2014) oprávnění vydávat elektronický podpis pouze 3 subjekty [14], mezi nimiž je nejznámější a největší státní podnik Česká pošta a její certifikační autorita PostSignum[5]. Uznávaný elektronický podpis je jako jediný druh elektronického podpisu uznáván orgány veřejné moci v České republice.

### 2.4.3 Podepisování PDF dokumentu

Dokumenty ve formátu PDF podporují několik typů elektronických podpisů od těch založených na čistě matematických principech, tj. použití haše/otisku dokumentu zašifrovaného asymetrickou šifrou až po způsoby biometrické identifikace pomocí ručně psaného podpisu, otisku prstu nebo skenu sítnice.

Podpora pro digitální podpisy je ve formátu PDF přítomna již od verze PDF 1.3, ovšem specifické formy autentifikace jsou implementovány pomocí plug-in modulu signature handler, který je doporučeno registrovat u Adobe. Informace o podpisu jsou obsaženy v tvz. signature dictionary.

[5] http://www.postsignum.cz/

Existují dva způsoby počítání hešů v PDF dokumentu. Prvním je otisk počítaný z určitého rozsahu bajtů v daném dokumentu, tzv. byte range digest. Rozsah je definován v položce `/ByteRange` v signature dictionary. Rozsah je obvykle definován tak, aby zahrnoval celý dokument, včetně samotného signature dictionary, vyjma položky `/Contents` v signature dictionary, která obsahuje hodnotu samotného podpisu. Pokud je použitý tento způsob, pak je vyžadováno, aby všechny položky v signature dictionary byly přímé objekty.

Druhým způsobem je otisk, který je počítaný pomocí selektivního procházení části stromu daného dokumentu v paměti počítače. Objekt, od kterého se s procházením části stromové struktury dokumentu začne, je specifikován položkou `/Data` v signature reference dictionary, další podrobnosti o způsobu a parametrech procházení jsou také specifikovány v tomto slovníku. Signature reference nemusí být v souboru jen jeden, těchto slovníků může být v souboru i více a odkazy na ně jsou umístěny v poli v položce `/Reference` v signature dictionary.

# 3 Obecné informace k provedené analýze

## 3.1 Definice PDF/SigQ

Následující tabulka popisuje skupiny vizuální integrity tak, jak jsou rozlišovány aplikací Adobe Acrobat, Adobe Reader a v dokumentu Digital Signiture User Guide for Acrobat 9.0 and Adobe Reader 9.0 [3].

| PDF/SigQ varianta | Popis |
|---|---|
| - | **Nevyhovující dokument (not compliant)** obsahuje externí závislosti (obrázky nebo multimédia, která jsou mimo PDF soubor) nebo TrueType fonty.<br><br>Před zobrazením režimu náhledu se objeví chybový dialog s textem: „*V tomto dokumentu je bohatý obsah, který nelze spolehlivě potlačit pomocí zobrazení náhledu dokumentu. Klepněte na Pokračovat, chcete-li přesto zobrazit náhled tohoto dokumentu*.“ Po stisknutí tlačítka pokračovat se objeví režim náhledu podpisu, kde je v horní liště zobrazena zpráva: „*Jste v režimu náhledu podpisu. Dokument obsahuje některé konstrukty, které způsobují dynamické chování nebo externí závislosti, a může se zobrazit nekonzistentně. Zkontrolujte to prosím před podepsáním u autora dokumentu*.“ |
| **PDF/SigQ-1B** | Dokument **obsahuje interaktivní obsah**, který může být potlačen v režimu náhledu (viewing mode), interaktivní obsah, který nelze potlačit není povolen.<br><br>V režimu náhledu se nahoře v liště zobrazí zpráva: *„Jste v režimu náhledu podpisu. Dokument obsahuje některé konstrukty, jako jsou například pole formulářů, multimédia nebo JavaScript, což může ovlivnit jeho vzhled. Zkontrolujte prosím dokument před tím, než ho podepíšete*.“ |
| **PDF/SigQ-1A** | **Žádný obsah s externími závislostmi** a **žádný interaktivní obsah**.<br><br>V režimu náhledu se nahoře v liště zobrazí zpráva: „*Jste v režimu náhledu podpisu. Dokument neobsahuje žádný dynamický obsah ani externí závislosti. Zkontrolujte prosím dokument před tím, než ho podepíšete*.“ |

Při výchozím nastavení registrů se nezobrazují chyby **2013** a **2014**. Ovšem po aktivaci zobrazování této chyby (podrobnosti viz popis postupu vyvolání těchto chyb) se zobrazuje u většiny vytvořených dokumentů. Takže se může stát, že na jednom počítači bude dokument **SigQ 1-A** nebo **SigQ 1-B** a na jiném stroji to bude **nevyhovující dokument.**

## 3.2 Inkrementální update

Inkrementálním updatem míníme situaci, kdy se obsah dokumentu změní po jeho podpisu. PDF prohlížeč musí tuto situaci rozpoznat a uživatele důrazně varovat, že aktuální obsah dokumentu neodpovídá jeho stavu v okamžiku podpisu.

Acrobat na tuto situaci upozorňuje varováním v modré liště v horní části aplikace (Acrobat X Pro), kde je napsáno:

„Podepsáno a všechny podpisy jsou platné, avšak byly zjištěny nepodepsané změny po posledním podpisu.“

Jsou zde pochybnosti, zda je varování Acrobatu dostatečně důrazné a zda laický uživatel chápe nebezpečnost dané situace.

**Soubor:** Comment_1002/comment_FreeText_1002sign.pdf

V podepisovaném PDF dokumentu je uložen pouze odkaz na externí Acrobat FDF (Forms Data Format) soubor, z něhož se mají data do podepisovaného PDF importovat. FDF soubor není nikterak chráněn před modifikací.

V určitých situacích (setkali jsme se s tím při importu dat z externího dokumentu) Acrobat informuje uživatele, že se dokument od podpisu změnil, v modrém pruhu nad zobrazeným dokumentem).

„Nejméně jeden podpis vyžaduje ověření.“

A teprve, když provedeme ověření volbou „Ověřit všechny podpisy“ na panelu podpisů, varování se změní na:

„Podepsáno a všechny podpisy jsou platné, avšak byly zjištěny nepodepsané změny po posledním podpisu.“

Příkladem je formulář `Hidden_actions_1001_1008/importForm_podvod_1008_1001.pdf` spolu s datovým souborem `importForm_1008_1001_data.fdf`.

## 3.3 Zvláštní chování editovaných PDF souborů

Pokus o podpis (zobrazení v režimu náhledu) před uložením vede k **růstu velikosti souboru**. Příklad ve složce „zvetseni_souboru_v rezimu_nahledu“. Při porovnání vnitřní struktury dokumentů je vidět, že dokument byl Acrobatem modifikován. Přidají se do něj písma, přestože dokument byl pouze zobrazen v režimu náhledu, nebyl podepsán a nic se v jeho obsahu nezměnilo.

Vícenásobná editace PDF souboru vede k tomu, že objekty z původní verze souboru zůstávají zachovány, změněné objekty jsou připsány na konec souboru. Proto při procházení stromu objektů v PDF souboru se musí určit, které objekty jsou použity a které ne. V souboru mohou zůstat objekty, které by měly vyvolat varování. Pokud ale tyto objekty nejsou používány, tak varování nevyvolávají, přesto, že se v souboru nalézají.

Objekty, které v souboru zůstaly z předchozí editace lze v Acrobatu odstranit použitím možnosti: Nástroje -> Ochrana -> Odstranit skryté informace -> V levém panelu vybereme pouze možnost: „Odstraněný nebo oříznutý obsah“ -> Stiskneme tlačítko „Odstranit“.

# 4 Analýza jednotlivých problémových konstrukcí

Analýza jednotlivých případů byla částečně vypracována v rámci projektu na základě smlouvy o dílo mezi Software 602 a.s. jakožto odběratelem a ČVUT FEL jakožto dodavatelem, jehož jsem se zúčastnil jako spoluřešitel spolu s vedoucím mé bakalářské práce. Výstupem tohoto projektu je technická zpráva 78[2], která byla oponována zadavatelem. V rámci projektu jsem se podílel na vytipování problematických konstrukcí, na kterých se daná problematická konstrukce projevuje, sestavil jsem PDF soubory s projevem daného problému a snažil jsem se odhalit a zanalyzovat odpovídající syntaktické úseky.

V následujících sekcích této kapitoly jsou představeny jednotlivé konstrukce, které mohou vést k nekonzistentnímu zobrazení daných dokumentů, jejich přítomnost v dokumentu ovšem nemusí nutně znamenat, že daný dokument bude zobrazen nekonzistentně.

V této části dokumentu jsou z důvodu přehlednosti jednotlivé problémové vlastnosti PDF dokumentů popsány pomocí tabulky. Tabulka 4.1 složí jako vzor, všech ostatních tabulek, které z důvodu přehlednosti dodržují jednotný formát.

**Tabulka 4.1 – Vzorová tabulka**

| | |
|---|---|
| **SigQ kód:** | **XXXX** PDF/SigQ level – Pokud tabulka nepojednává přímo o chybě 2013, 2014, tak tyto dvě chyby nejsou uvedeny. |
| **Odkaz:** | [1] kapitola X.X.X nadpis, strana XXX kde je daná vlastnost zmíněna |
| **Klíč:** | `/XXX/YYY` (Required/Optional) |
| **Popis:** | Klíč `/XXX` s hodnotou `/YYY` označuje... popis klíče nebo dvojce klíč-hodnota, jenž danou vlastnost definuje. |
| **Očekávaný projev:** | Projev vlastnosti, který byl při pokusu očekáván. |
| **Soubor:** | `cesta_k_ukázkovému/souboru.pdf` |
| **Postup:** | panel Nástroje -> XXX -> XXX -> popis postupu, jak chybu v Acrobatu vyvolat, byl použit Acrobat verze 10.1.8 na Win7 64bit |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód XXXX. |
| **Pozorování:** | Soubory s Tlačítky vyvolávají kód XXXX. |
| **Důvod:** | Proč jsou ve zprávě zobrazeny kódy XXXX a YYYY |
| **Posouzení bezpečnosti:** | Podrobnější analýza možnosti zneužití příslušné vlastnosti PDF dokumentu. |

Každý problém je v této části předveden na příkladu, jako je ukázkový příklad níže. Příklad vždy obsahuje objekt nebo objekty, kde je popisovaná vlastnost definována.

### Příklad 4.1 – Vzorový příklad

```
1 0 obj
<</Místo, kde je v ukázkovém PDF souboru definována daná problémová
vlastnost>>
endobj
```

## 4.1 JavaScript

### 4.1.1 JavaScript – Trigger Events

Všeobecný popis je v kapitole 8.5.2 Trigger Events na str. 649, tabulky 8.44,8.45, 8.46, 8.47.

| Tabulka 4.2 – JavaScript, Trigger Event /WC | |
|---|---|
| **SigQ kód:** | **1000** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.2 JavaScript, strana 651 |
| **Klíč:** | **/WC** (Optional) v document catalog's additional-actions dictionary, samotný skript je v **/JS** a je specifikován klíčem a hodnotou **/S/JavaScript** (Required) |
| **Popis:** | Klíč **/WC** (will close), značí JavaScriptovu akci dokumentu, která se vykoná před zavřením souboru. |
| **Očekávaný projev:** | Zobrazení kódů 1000 nebo 1008 ve zprávě. |
| **Soubor:** | JavaScript_1000/JavaScript_alertWhileClosing_1000.pdf |
| **Postup:** | panel Nástroje -> JavaScript -> Nastavit akce dokumentu Dokument se zavře -> Upravit -> napíšeme tam nějaký JavaScriptový kód, např. „app.alert("Zaviram se...");“ |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1000. |
| **Pozorování:** | Soubor s nastavenou akcí před zavřením souboru, vyvolává zprávu s kódem 1000. |
| **Důvod:** | Dokument obsahuje JavaScript, to je skrytá akce, tudíž kód 1000. **Trigger WC (will close) vyvolává pouze kód 1000, na rozdíl od ostatních trigger eventů.** |
| **Posouzení bezpečnosti:** | JavaScript Trigger Events **jsou velmi nebezpečné**, spouštěný JavaScript může změnit obsah zobrazeného dokumentu. |

**Příklad 4.2 – JavaScript, Trigger Event /WC**

```
8 0 obj
<</AA<</WC 18 0 R>>/Metadata 1 0 R/PageLayout/OneColumn/Pages 5 0
R/Type/Catalog>> endobj
18 0 obj
<</JS(app.alert\("Zaviram se..."\);)/S/JavaScript>> endobj
```

| Tabulka 4.3 – JavaScript, Trigger Events /WS, /DS, /WP a /DP | |
|---|---|
| **SigQ kód:** | **1000**, **1008** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.3.3 JavaScript, strana 651 |
| **Klíče:** | **/WS, /DS, /WP, /DP** v document catalog's additional-actions dictionary samotný skript je v **/JS** a je specifikován klíčem a hodnotou **/S/JavaScript** |
| **Popis:** | Klíč **/WS** (will save), značí JavaScriptovu akci dokumentu, která se vykoná **před** uložením souboru.<br>Klíč **/DS** (did save), značí JavaScriptovu akci dokumentu, která se vykoná **po** uložení souboru.<br>Klíč **/WP** (will print), značí JavaScriptovu akci dokumentu, která se vykoná **před** tiskem.<br>Klíč **/DP** (did print), značí JavaScriptovu akci dokumentu, která se vykoná **po** tisku. |
| **Očekávaný projev:** | Varování s kódem 1000 nebo 1008 |
| **Soubor:** | JavaScript_1000/JavaScript_alertWhileClosing_1000.pdf |

| | |
|---|---|
| **Postup:** | panel Nástroje -> JavaScript -> Nastavit akce dokumentu -> Dokument se zavře -> Upravit -> napíšeme tam nějaký JavaScriptový kód, např. „app.alert("Zaviram se...");“ |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny kódy 1000 a 1008 |
| **Pozorování:** | Soubor s nastavenou akcí před nebo po tisku, vyvolává kódy 1000 a1008. |
| **Důvod:** | Dokument obsahuje JavaScript, to je skrytá akce, tudíž kódy 1000 a 1008. |
| **Posouzení bezpečnosti:** | JavaScript Trigger Events **jsou velmi nebezpečné**, spouštěný JavaScript může změnit obsah zobrazeného dokumentu. |

**Příklad 4.3 – JavaScript, Trigger Event /WS**

```
9 0 obj
<</AA<</WS 6 0 R>>/Metadata 1 0 R/PageLayout/OneColumn
/Pages 5 0 R/Type/Catalog>>
endobj

6 0 obj
<</JS (app.alert\("Bude se ukládat..."\);)
/S /JavaScript>>
endobj
```

**Příklad 4.4 – JavaScript, Trigger Event /DS**

```
9 0 obj
<</AA<</DS 19 0 R>>/Metadata 1 0 R/PageLayout/OneColumn
/Pages 5 0 R/Type/Catalog>>
endobj

19 0 obj
<</JS(app.alert\("Dokument se uložil."\);)/S/JavaScript>>
endobj
```

**Příklad 4.5 – JavaScript, Trigger Event /WP**

```
9 0 obj
<</AA<</WP 6 0 R>>/Metadata 1 0 R/PageLayout/OneColumn
/Pages 5 0 R/Type/Catalog>>
endobj

6 0 obj
<</JS (app.alert\("Bude se tisknout..."\);)/S /JavaScript>>
endobj
```

**Příklad 4.6 – JavaScript, Trigger Event /DP**

```
9 0 obj
<</AA<</DP 24 0 R>>/AcroForm 19 0 R/Metadata 1 0 R
/Outlines 20 0 R/PageLayout/OneColumn/Pages 5 0 R/Type/Catalog>>
endobj

24 0 obj
<</JS(app.alert\("Dokument se vytisknul..."\);)/S/JavaScript>>
endobj
```

### 4.1.2 JavaScript actions

**Tabulka 4.4 – JavaScript actions**

| | |
|---|---|
| **SigQ kód:** | **1001** a **1008** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.6.4 Form Actions -> JavaScript Actions, strana 709 |
| **Klíč:** | **/S/JavaScript** a **/JS** |
| **Popis:** | Klíč **/S** s hodnotou **/ JavaScript** označuje objekt JavaScriptu. Klíč **/S** je požadovaný (Required). |
| **Očekávaný projev:** | **1001** a **1008** |
| **Soubor:** | Hidden_actions_1001_1008/button_1001_1008JavaScript_pres_button.pdf |
| **Postup:** | panel Nástroje -> Obsah- > přidat Tlačítko, pak Vybrat obsah -> vybereme tlačítko -> v kontextové nabídce vybereme položku „Vlastnosti“ -> panel Akce -> Vybraná akce: Spustit JavaScript |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny kódy 1001 a 1008. |
| **Pozorování:** | Soubory s Tlačítky vyvolávají zprávu s kódy 1001 a 1008. |
| **Důvod:** | Tlačítko způsobí zobrazení kódu 1001, protože vizuální vzhled tlačítka se může měnit na základě externích proměnných. To, že dokument obsahuje JavaScript způsobí, že ve zprávě je uveden kód 1008 s typem akce (action type) JavaScript. |
| **Posouzení bezpečnosti:** | JavaScript actions **jsou velmi nebezpečné**, spouštěný JavaScript může změnit obsah zobrazeného dokumentu. |

**Příklad 4.7 – JavaScript actions**

```
18 0 obj
<</A 29 0 R/AP<</N 23 0 R>>/DA(/HeBo 12 Tf 0 g)/F 4/FT/Btn
/Ff 65536/MK<</BG[0.75293]/CA(Button)>>/P 9 0 R
/Rect[212.825 739.122 284.825 759.122]
/Subtype/Widget/T(button)/Type/Annot>>
endobj

29 0 obj
<</JS(app.alert\("JavaScript spusten..."\);)/S/JavaScript>>
endobj
```

## 4.2 Annotations

Specifikace formátu PDF [1] kapitola 8.4.2 Annotation Flags, tab. 8.16 na str. 608. Tag **/F** obsahuje parametr, kterým je celé číslo (rozsah 32 bitů) zapsáno v textové podobě. Význam je přiřazen jednotlivým bitům tohoto čísla, bity jsou číslovány od 1, jednička značí nejméně významný bit. Např. **/F 4** reprezentuje čtveřici bajtů, z nichž 3 významnější jsou nulové a nejméně významný bajt má podobu 0000 0100. Má tedy nastaven bit číslo 3.

Týká se to všech anotací, to jest:

- Widget annotations 4.2.1
- Free Text annotation 4.2.2
- Multimédia 4.2.3
- Pop-up Anotace 4.2.4

Vliv nastavení těchto příznaků z hlediska bezpečnosti je dále v této sekci ilustrován na případu Widget annotace. V Acrobatu je možné nastavit pouze 4 kombinace bitů, které jsou dokumentovány na příkladech 1 až 4. Bezpečný je Widget, který má /F 4.

### Příklad 4.8 – Viditelná anotace

Tag **/F** s hodnotou **4**
32 bitový integer: `0000 0000 0000 0000 0000 0000 0000 0100`
Nastaven je bit 3, který se jmenuje Print.
Widget se zobrazí na monitoru a je vytištěn.

```
21 0 obj
<<
/AP <<
/N 26 0 R
>>
/DA (/HeBo 12 Tf 0 g)
/F 4
/FT /Btn
/Ff 65536
/MK <<
/BG [ 0.75293 ]
/CA (Pokus...)
>>
/P 14 0 R
/Rect [ 64.8267 736.298 136.827 756.298 ]
/Subtype /Widget
/T (... binarní data... )
/Type /Annot
>>
endobj
```

**Soubor:** Non_signature_form_fields_1001/Form_1001_bezpecny_Viditelne_decompressed.pdf

### Příklad 4.9 – Viditelná, ale netištěná anotace

Chybí tag **/F**, tím pádem Print není nastaveno.
Widget se zobrazí na monitoru, ale netiskne se.

```
21 0 obj
<<
/AP <<
/N 26 0 R
>>
/DA (/HeBo 12 Tf 0 g)
/FT /Btn
/Ff 65536
/MK <<
/BG [ 0.75293 ]
/CA (Pokus...)
>>
/P 14 0 R
/Rect [ 64.8267 736.298 136.827 756.298 ]
/Subtype /Widget
/T (...binarni data...)
/Type /Annot
>>
endobj
```

**Soubor:**
Non_signature_form_fields_1001/Form_1001_nebezpecny_ViditelneNetistene_decompressed.pdf

### Příklad 4.10 – Skrytá anotace

Tag **/F** s hodnotou **6**
32 bitový integer: 0000 0000 0000 0000 0000 0000 0000 0**11**0
Nastaven je bit **2**, který se jmenuje **Hidden**, a bit **3**, který se jmenuje **Print**.
Widget se nezobrazí na monitoru a netiskne se.

```
21 0 obj
<<
/AP <<
/N 26 0 R
>>
/DA (/HeBo 12 Tf 0 g)
/F 6
/FT /Btn
/Ff 65536
/MK <<
/BG [ 0.75293 ]
/CA (Pokus...)
>>
/P 14 0 R
```

```
/Rect [ 64.8267 736.298 136.827 756.298 ]
/Subtype /Widget
/T (...binarni data...)
/Type /Annot
>>
endobj
```

**Soubor:** Non_signature_form_fields_1001 /Form_1001_nebezpecny_Skryte_decompressed.pdf

### Příklad 4.11 – Skrytá, ale tištěná anotace

Tag **/F** s hodnotou **36**
32 bitový integer: 0000 0000 0000 0000 0000 0000 00**1**0 0**1**00
Nastaven je bit **3**, který se jmenuje **Print**, a bit **6**, který se jmenuje **No View**.
Widget se nezobrazí, ale vytiskne se.

```
21 0 obj
<<
/AP <<
/N 26 0 R
>>
/DA (/HeBo 12 Tf 0 g)
/F 36
/FT /Btn
/Ff 65536
/MK <<
/BG [ 0.75293 ]
/CA (Pokus...)
>>
/P 14 0 R
/Rect [ 64.8267 736.298 136.827 756.298 ]
/Subtype /Widget
/T (...binární data...)
/Type /Annot
>>
endobj
```

**Soubor:**
Non_signature_form_fields_1001/Form_1001_nebezpecny_SkryteTistene_decompressed.pdf

## 4.2.1 Widget annotations

**Tabulka 4.5 – Widget annotations**

| | |
|---|---|
| **SigQ kód:** | **1001** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.4.5 Widget annotation, strana 640 |
| **Klíč:** | **/Subtype/Widget** (Required) |
| **Popis:** | Klíč **/Subtype** a hodnotou **/Widget** označuje widget anotace, které reprezentují pole a interakci s uživatelem. |
| **Očekávaný** | |

| | |
|---|---|
| **projev:** | |
| **Soubor:** | Výše uvedený příklad se vztahuje k souboru:<br>`non_signature_form_fields_1001/button_1001.pdf`<br>Příklady dalších projevů téhož problému jsou v ostatních souborech téže složky. |
| **Postup:** | panel Nástroje -> Obsah -> Tlačítko<br>nebo<br>Vytvořit -> Formulář PDF... -> ...panel vpravo Úlohy -> Přidat nové pole ... |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny chyby 1000 |
| **Pozorování:** | Soubory s tlačítky a dalšími formulářovými poli, seznamy a rozbalovacími nabídkami, tj. jiná pole formuláře než pole podpisu/non-signature form fields, vyvolávají chybu 1001. |
| **Důvod:** | Vizuální vzhled widget annotations se může měnit v závislosti na externích proměnných. |
| **Posouzení bezpečnosti:** | Widget anotace tedy kromě rizika spojeného s anotačními příznaky, popsanými v úvodu sekce 4.2 Annotations, nepředstavuje další bezpečnostní riziko, viz příklady 1 až 4 v sekci 4.2. |

**Příklad 4.12 – Widget annotations**

```
18 0 obj
<</AP<</N 23 0 R>>/DA(/HeBo 12 Tf 0 g)/F 4/FT/Btn
/Ff 65536/MK<</BG[0.75293]/CA(Button)>>
/P 9 0 R
/Rect[212.825 739.122 284.825 759.122]
/Subtype/Widget/T(button)/Type/Annot>>
endobj
```

### 4.2.2 Free Text annotation

**Tabulka 4.6 – Free Text annotation**

| | |
|---|---|
| **SigQ kód:** | **1002** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.4.5 FreeText annotation, strana 624 |
| **Klíč:** | **`/Subtype/FreeText`** |
| **Popis:** | Klíč **`/Subtype`** s hodnotou **`/FreeText`** označuje anotaci, která se zobrazuje přímo na stránce. |
| **Očekávaný projev:** | `1002` |
| **Soubor:** | `Comments_1002/comment_FreeText_1002.pdf` |
| **Postup:** | panel Poznámka -> Kreslená označení -> Přidat textový rámeček |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1002 a 1000 |
| **Pozorování:** | Soubory, které obsahují Free Text anotace/textové rámečky vyvolávají zprávu s kódem 1002. |
| **Důvod:** | Vizuální vzhled poznámek se může měnit na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Text anotace může svým vzhledem splynout s obsahem dokumentu, ale je součástí podpisu a proto jeho změna po podpisu odhalena – viz *3.2 Inkrementální update*. FreeText anotace tedy kromě rizika spojeného s anotačními příznaky, popsanými v úvodu sekce 4.2, nepředstavuje další bezpečnostní riziko.<br><br>**Soubor:** `Comment_1002/comment_FreeText_1002sign.pdf` |

**Příklad 4.13 – Free Text annotation**

```
22 0 obj
<</AP<</N 23 0 R>>/C[1.0 1.0 1.0]
/Contents(Text)/CreationDate(D:20131213130221+01'00')
/DA(1 0 0 rg /Helv 12 Tf)
/DS(font: Helvetica,sans-serif 12.0pt; text-align:left;
color:#FF0000 )
/F 4/M(D:20131213130239+01'00')
/NM(5ce50bd5-f834-4569-8f6b-dd2a36a1fa33)/P 9 0 R
/RC(<?xml version="1.0"?><body xmlns="http://www.w3.org/1999/xhtml"
xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"
xfa:APIVersion="Acrobat:10.1.8" xfa:spec="2.0.2"  style="font-
size:12.0pt;text-align:left;color:#FF0000;font-weight:normal;font-
style:norma\
l;font-family:Helvetica,sans-serif;font-stretch:normal"><p
dir="ltr"><span style="font-
family:Helvetica">Text</span></p></body>)
/Rect[174.305 566.607 453.193 709.627]
/Subj(Textovy ramecek)/Subtype/FreeText
/T(Ondra-ProBook)/Type/Annot>>
endobj
```

### 4.2.3 Multimédia

Přidání SWF souboru, Přidání Zvuku, Přidání Videa, SWF souboru a 3D

Obsah -> Multimédia -> Video, Zvuk a SWF je vloženo v anotaci, která má podtyp „**/Subtype/RichMedia**“, ovšem 3D anotace je přímo „**/Subtype /3D**“.

#### Posouzení bezpečnosti: Vložená multimédia

Všechny typy multimédií jsou speciálním případem anotace a vztahuje se k nim tudíž bezpečnostní riziko spojené s anotačními příznaky, popsanými v úvodu sekce 4.2 Annotations.

Pokud je dokument vytvořený v Acrobatu verze X, tak se vložená multimédia automaticky vkládají přímo do PDF souboru, a je použit tag **/EF** ([1] kapitola 3.10.3 Embedded File Streams). V takovém případě nevidíme (s výjimkou dvou níže uvedených případů) možnost, aby byl dokument po podpisu zobrazen odlišně – snad až na odlišný vzhled ovládacích prvků přehrávače.

Vidíme pouze dvě problematické situace:

- Typem multimédia je Flash (viz. sekce 0). Obsah může být naprogramován tak, že se v různých situacích chová různě, doporučujeme nedovolit podepisovat dokumenty obsahující Flash.
- V případě, že multimediální obsah není obsažen uvnitř podepisovaného PDF dokumentu, ale používá se externí soubor. Takový dokument sice není možné v Acrobatu vytvořit, ale syntakticky přípustný je. V takovém případě ve FileSpecification dictionary odpovídajícím danému multimediálnímu obsahu chybí tag **/EF**.

#### *4.2.3.1 Vložený zvuk*

**Tabulka 4.7 – Vložený zvuk**

| | |
|---|---|
| **SigQ kód:** | **1002** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 9.2 Sounds, strana 782 |
| **Klíč:** | **/Subtype/Sound** |
| **Popis:** | Klíč **/Subtype** s hodnotou **/Sound** označuje zvuk vložený v anotaci, která obsahuje zvuk, typ této anotace je **/Subtype/RichMedia**. |
| **Očekávaný projev:** | 1002 |
| **Soubor:** | Comments_1002/zvuk_1002.pdf |
| **Postup:** | panel Nástroje -> Obsah -> Multimédia -> Zvuk... |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1002. |
| **Pozorování:** | Soubory se zvuky vyvolávají zprávu s kódem 1002. |
| **Důvod:** | Vložená multimédia jsou brána jako poznámky a vizuální vzhled poznámek se může měnit na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny typy multimédií na začátku sekce 4.2.3 Multimédia. |

**Příklad 4.14 – Vložený zvuk**

```
21 0 obj
<</Instances 22 0 R/Subtype/Sound>>
endobj

22 0 obj
[23 0 R]
endobj

23 0 obj
<</Asset 24 0 R/Params 25 0 R>>
endobj

24 0 obj
<</EF<</F 27
0 R>>/F(AudioPlayer.swf)/Type/Filespec/UF(AudioPlayer.swf)>>
endobj

25 0 obj
<</Binding 26 0
R/FlashVars(source=zvuk.mp3&autoPlay=true&volume=1.00)>>
endobj
```

#### *4.2.3.2 Vložené video*

**Tabulka 4.8 – Vložené video**

| | |
|---|---|
| **SigQ kód:** | **1002** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 9.3 Movies, strana 784 |
| **Klíč:** | **/Subtype/Video** |
| **Popis:** | Klíč **/ Subtype** s hodnotou **/ Video** označuje video v anotaci, typ této anotace je **/Subtype/RichMedia.** |

| | |
|---|---|
| **Očekávaný projev:** | 1002 |
| **Soubor:** | Comment_1002/Multimedia_video_1002.pdf |
| **Postup:** | panel Nástroje -> Obsah -> Multimédia -> Video |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1002. |
| **Pozorování:** | Soubory s videem vyvolávají kód 1002. |
| **Důvod:** | Vložená multimédia jsou brána jako poznámky a vizuální vzhled poznámek se může měnit na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny typy multimédií na začátku sekce 4.2.3 Multimédia. |

**Příklad 4.15 – Vložené video**

```
21 0 obj
<</Instances 22 0 R/Subtype/Video>>
endobj

22 0 obj
[23 0 R]
endobj

23 0 obj
<</Asset 24 0 R/Params 25 0 R>>
endobj

24 0 obj
<</EF<</F 27 0
R>>/F(VideoPlayer.swf)/Type/Filespec/UF(VideoPlayer.swf)>>
endobj

25 0 obj
<</Binding 26 0
R/FlashVars(source=big_buck_bunny_small.mp4&skin=SkinOverAllNoFullNo
Caption.swf&skinAutoHide=true&skinBackgroundColor=0x5F5F5F&skinBackg
roundAlpha=0.75&volume=1.00)>>
endobj
```

#### *4.2.3.3 Vložený Flash*

**Tabulka 4.9 – Vložený Flash**

| | |
|---|---|
| **SigQ kód:** | **1002** PDF/SigQ-1B |
| **Odkaz:** | **/Subtype/Flash** není v dokumentaci [1] zmíněn, zmínka o SWF je na straně 1123 |
| **Klíč:** | **/Subtype/Flash** |
| **Popis:** | Klíč **/Subtype** s hodnotou **/Flash** označuje vložený SWF objekt v anotaci, typ této anotace je **/Subtype/RichMedia.** |
| **Očekávaný projev:** | 1002 |
| **Příklad:** | |
| **Soubor:** | Comment_1002/comment_SWF_1002.pdf |
| **Postup:** | panel Nástroje -> Obsah -> Multimédia -> SWF... |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uvedena chyba 1002. |

| | |
|---|---|
| **Pozorování:** | Soubory s tlačítky vyvolávají chybu 1002. |
| **Důvod:** | Vložená multimédia jsou brána jako poznámky a vizuální vzhled poznámek se může měnit na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny typy multimédií na začátku sekce 4.2.3 Multimédia. |

**Příklad 4.16 – Vložený Flash**

```
21 0 obj
<</Instances 22 0 R/Subtype/Flash>>
endobj

22 0 obj
[23 0 R]
endobj

23 0 obj
</Asset 24 0 R/Params 25 0 R>>
endobj

24 0 obj
<</EF<</F 27 0 R>>/F(Car-speakers-590x90.swf)/Type/Filespec/UF(Car-
speakers-590x90.swf)>>
endobj

27 0 obj
<</DL 116887/Length 116887/Params<</CheckSum<995BB44DF3D6B31D9422DDB
9F3F78B7B>/CreationDate(D:20131213005425+01'00')/ModDate(D:200902261
42630+01'00')/Size 116887>>/Subtype/application#2Fx-shockwave-
flash>>
stream
...data flashového souboru...
```

#### *4.2.3.4 3D Anotace*

**Tabulka 4.10 – 3D Anotace**

| | |
|---|---|
| **SigQ kód:** | **1002** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 9.5.1 3D Annotation, strana 791 |
| **Klíč:** | **/Subtype /3D** |
| **Popis:** | Klíč **/Subtype** s hodnotou **/3D** označuje typ anotace, která obsahuje 3D obsah. |
| **Očekávaný projev:** | 1002 |
| **Soubor:** | Comments_1002/comments_3D.pdf |
| **Postup:** | panel Nástroje -> Obsah-> Multimédia -> 3D... |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uvedena chyba 1002. |
| **Pozorování:** | Soubory s 3D anotacemi vyvolávají zprávu s kódem 1002. |
| **Důvod:** | Vložená multimédia jsou brána jako poznámky a vizuální vzhled poznámek se může měnit na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny typy multimédií na začátku sekce 4.2.3 Multimedia. |

**Příklad 4.17 – 3D Anotace**

```
19 0 obj
<<
/3DA <</Style /Embedded>>
/3DD 20 0 R
/3DV 21 0 R
/AP <</N 22 0 R>>
/BS <</S /S/Type /Border/W 0>>
/Border [ 0 0 0 ]
/Contents (3D Model)
/F 68
/NM (3D1)
/P 9 0 R
/Rect [ 174.184 537.672 503.07 726.753 ]
/Subtype /3D
/Type /Annot
>>
endobj
```

### 4.2.4 Pop-up Anotace

**Tabulka 4.11 – Pop-up Anotace**

| | |
|---|---|
| **SigQ kód:** | **1002** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.4.5 Pop-up Annotations, strana 637 |
| **Klíč:** | **/Subtype/Popup** |
| **Popis:** | Klíč **/Subtype** s hodnotou **/Popup** označuje poznámku, která zobrazuje text v pop-up okně. |
| **Očekávaný projev:** | 1002 |
| **Soubor:** | Comments_1002/commnet_pop_up_1002.pdf |
| **Postup:** | panel Poznámka-> Anotace -> Přidat lístek s poznámkou |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uvedena chyba 1002. |
| **Pozorování:** | Soubory s Pop-up anotacemi (Poznámky, Sicky notes) vyvolávají chybu 1002. |
| **Důvod:** | Pop-up anotace jsou poznámky a vizuální vzhled poznámek se může měnit na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Vzhled Pop-up anotací se může měnit, v Adobe Acrobatu X a Adobe Readeru X jsou dobře odlišené od samotného obsahu dokumentu (vypadají jako žluté nalepovací lístečky na dokumenty), nepředpokládáme proto, že by to uživatele významným způsobem mátlo. |

**Příklad 4.18 – Pop-up Anotace**

```
20 0 obj
<</AP<</N 22 0 R>>/C[1.0 1.0 0.0]
/Contents(Toto je poznámka.)/CreationDate(D:20131213112121+01'00')
/F 28/M(D:20131213112131+01'00')
/NM(89841106-7734-4b18-9c76-d99731c80d05)
/Name/Comment/P 9 0 R/Popup 21 0 R
```

```
/RC(<?xml version="1.0"?><body xmlns="http://www.w3.org/1999/xhtml"
xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"
xfa:APIVersion="Acrobat:10.1.8" xfa:spec="2.0.2" ><p dir="ltr"><span
dir="ltr" style="font-size:10.0pt;text-
align:left;color:#000000;font-w\
eight:normal;font-style:normal">Toto je
pozn&#225;mka.</span></p></body>)
/Rect[138.336 775.962 156.336 793.962]
/Subj(Lístek s poznámkou)/Subtype/Text/T(Ondra-ProBook)/Type/Annot>>
endobj

21 0 obj
<</F 28/Open true/Parent 20 0 R/Rect[595.32 673.962 775.32
793.962]/Subtype/Popup/Type/Annot>>
endobj
```

## 4.3 Položky nabídek/Named actions

**Tabulka 4.12 – Položky nabídek/Named actions**

| | |
|---|---|
| **SigQ kód:** | **1003**, 1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.1 Action Dictionaries, strana 647<br>[1] Named Actions, strana 666, tab. 8.61, tab. 8.62 |
| **Klíč:** | **/S/Named** |
| **Popis:** | Klíč **/S** s hodnotou **/Named** v action dictionary (**/A**) |
| **Očekávaný projev:** | 1003 |
| **Soubor:** | named_actions/named_actions_1001_1003.pdf |
| **Postup:** | panel Nástroje -> Obsah- > přidat Tlačítko, pak Vybrat obsah -> vybereme tlačítko -> v kontextové nabídce vybereme položku „Vlastnosti“ -> panel Akce -> Vybraná akce: Provést příkaz z nabídky -> Soubor -> Zavřít |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny chyby 1001 a 1003. |
| **Pozorování:** | Soubory s Tlačítky, které spouští pojmenované akce vyvolávají chyby 1001 a 1003. |
| **Důvod:** | Named actions mohou spouštět příkazy z nabídky bez toho, aby o tom uživatel věděl, proto kód 1003.<br><br>Kód 1001 je způsoben tlačítkem, popsáno v tomto dokumentu kapitola 4.2 Widget annotations. |
| **Posouzení bezpečnosti:** | Výchozí akce uvedené v tabulce 8.61 Named actions na straně 666 v dokumentaci PDF [1] **jsou bezpečné**, protože slouží pouze k navigaci v aktuálním dokumentu. Výchozí akce v tabulce 8.61 mají jména (**/N**): **/N/NextPage**; **/N/PrevPage**; **/N/FirstPage**; **/N/LastPage** a jsou ve slovníku Named action (tag subtype:**/S/Named** ).<br>Chování dokumentu by tak bylo závislé na tom, zda použitá aplikace, popř. její verze, danou akci podporuje a dokument by tak byl **nepřenosný** (viz poznámka na straně 666 v [1]). |

**Příklad 4.19 – Položky nabídek/Named actions**

```
18 0 obj
<</A 33 0 R/AP<</N 23 0 R>>/DA(/HeBo 12 Tf 0 g)/F 4/FT/Btn
/Ff 65536/MK<</BG[0.75293]/CA(Button)>>/P 9 0 R
/Rect[212.825 739.122 284.825 759.122]/Subtype/Widget
/T(button)/Type/Annot>>
endobj

33 0 obj
<</N/Close/S/Named>>
endobj
```

## 4.4 Prezentace

| Tabulka 4.13 – Položky nabídek/Named actions | |
|---|---|
| **SigQ kód:** | **4000** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 8.3.3 Presentations, strana 598 |
| **Klíče:** | **/Dur** a **/Trans** |
| **Popis:** | Uvedené klíče se mohou vyskytovat v prezentacích ve formátu PDF. Tag **/Dur** definuje dobu trvání slajdu a tag **/Trans** způsob přechodu mezi slajdy. |
| **Očekávaný projev:** | Chybovou hlášku 1004 se ale nepodařilo reprodukovat. |
| **Soubor:** | Presentation_4000/ presentation_4000.pdf |
| **Postup:** | Nástroje -> Zpracovaní dokumentu -> Přechody stránek |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uvedena chyba 4000. |
| **Pozorování:** | Prezentace byla vytvořena v aplikaci PowerPoint, v Acrobatu ji lze bez problémů přehrát. Při pokusu o podpis však hlásí: *„V tomto dokumentu je bohatý obsah, který nelze spolehlivě potlačit pomocí zobrazení náhledu dokumentu. Klepněte na Pokračovat, chcete-li přesto zobrazit náhled tohoto dokumentu.*“ Pokračujeme-li, hlásí Acrobat chybu 4000 (Nerozpoznaný obsah PDF). |
| **Důvod:** | Důvod zobrazení chyby 4000 je neznámý, protože příklad byl vytvořen zásuvným modulem Acrobatu pro MS PowerPoint a nebyl nijak ručně editován. Přesto je tato prezentace, která byla vygenerována generátorem od Adobe, který je součástí Acrobatu X Pro, zobrazuje chybu 4000 – Unrecognized content. |
| **Posouzení bezpečnosti:** | Prezentaci s chybou, která by měla ve zprávě SigQ kód 1004, se nám nepodařilo vytvořit. Klíče **/Trans** a **/Dur** považujeme za **relativně bezpečné.** Projevují se pouze v režimu „Na celou obrazovku“ (obdoba slide-show v PowerPointu). Přišli jsme na jediný případ, kdy by mohl být uživatel zmaten. Příklad 4.20 ukazuje, že pokud by totiž byl čas trvání stránky (/Dur) nastaven na 0, stránka se v případě rychlého přechodu (např. Transition dictionary /S/R) vůbec nezobrazí (nebo nepostřehnutelně problikne), ačkoliv ji při podpisu viděl. Mimo režimu „Na celou obrazovku“ totiž viditelná je.<br><br>**Soubor:** Presentation_4000/presentation_4000_ZeroDur.pdf |

**Příklad 4.20 – Nebezpečná prezentace**

```
50 0 obj
<</Contents 52 0 R/CropBox[0.0 0.0 720.0 540.0]/Dur 0.0
```

```
/MediaBox[0.0 0.0 720.0 540.0]/Parent 46 0 R
/Resources<</ColorSpace<</CS0 55 0 R>>/Font<</TT0 57 0 R>>>>
/Rotate 0/StructParents 0/Trans 61 0 R/Type/Page>>
endobj

61 0 obj
<</Curve/Easy/Directional/BiDir/Dm/V/S/R>>
endobj
```

**Poznámka:** Specifikace [1] kapitola **9.4 Alternate Presentations** (str. 786) popisuje alternativní prezentace, které ovšem nemusí PDF prohlížeče podporovat. Ve verzi PDF 1.5 a pozdějších je jediným typem alternativních prezentací „Slideshow" (/Type/Slideshow/S/Embedded). V Acrobatu X Pro ani jinými prostředky se nám nepodařilo takovouto alternativní prezentaci vytvořit.

## 4.5 XFA-based documents

**Tabulka 4.14 – XFA-based documents**

| | |
|---|---|
| **SigQ kód:** | **1005** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 8.6.7 XFA Forms, strana 722 |
| **Klíč:** | **/XFA** |
| **Popis:** | Klíč **/XFA** označuje XFA interactive form dictionary. V některých případech mají XFA dokumenty **/NeedsRendering true**, ale ne vždy např. RadioCheckBox_XFA_1005_2014_1001_1008.pdf<br>HSS-Foot-and-Ankle-Registration-Forms-v3.pdf<br>birth-certificate-application-eform-tt-sample_1005_2014_1001_1008.pdf<br>**/NeedsRendering true** neobsahují, přitom jde o XFA formuláře. |
| **Očekávaný projev:** | 1005 |
| **Soubor:** | XFA_Form_1005/*.pdf |
| **Zdroje:** | http://www.windjack.com/DownLoads/RadioCheckBox_XFA.pdf<br>http://forums.adobe.com/thread/496087<br>http://smarte-forms.com/wp/wp-content/uploads/2011/03/birth-certificate-application-eform-tt-sample.pdf |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1005. |
| **Pozorování:** | Soubory s XFA formuláři vyvolávají kód 1005. |
| **Posouzení bezpečnosti:** | Vzhled a chování dokumentů založených na XFA formulářích se může měnit. Narozdíl od Acro Formulářů jsou totiž podepsaná pouze data nikoliv vizuální podoba formuláře, která se vždy dynamicky generuje.<br><br>[3] stránka 130, tabulka 13: *„XFA-based (dynamic forms) documents are not allowed since such forms could alter the document's appearance or behavior."*<br><br>[6] stránka 7*: „The form author uses LiveCycle Designer ES to lay out the form and create an XFA or XML Data Package (XDP) file to persist the template. By default LiveCycle Designer 8.0.1 or later saves the form as a dynamic PDF file that is compatible with 8.0.1 or later.*<br>*Tip: To enable a form to be signed or certified, that form must be rendered to PDF on the server or client. If the form is always being generated on the fly with data and is completely dynamic, it conflicts with the "what you see is what you sign" and "what you signed is what you saw" principles that are the core tenets of a* |

*corner-to-corner digital signature. This is quite different from the expectation surrounding a data signature in which the data alone is being signed—not the visual representation of the form."*

**Příklad 4.21 – XFA-based documents**

```
41 0 obj
<<
/DA (/Helv 0 Tf 0 g )
/DR <<
/Font <<
/Helv 18 0 R
/MyriadPro-Black 19 0 R
/MyriadPro-Regular 20 0 R
>>
>>
/Fields [ ]
/XFA [ (preamble) 1 0 R (config) 2 0 R (template) 3 0 R (localeSet)
4 0 R (form) 5 0 R (datasets) 6 0 R (xmpmeta) 7 0 R (xfdf) 8 0 R
(postamble) 9 0 R ]
>>
endobj
```

## 4.6 Dokument odkazuje na externí PDF dokument

**Nepodařilo se v Acrobatu reprodukovat.**

**Tabulka 4.15 – Dokument odkazuje na externí PDF dokument**

| | |
|---|---|
| **SigQ kód:** | **předpokládaný kód 1006** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Embedded Go-To Actions strana 655 |
| **Klíč:** | **/S/GoToE** , který odkazuje na externí obsah **/F (someFile.pdf)** |
| **Popis:** | Klíč **/S** s hodnotou **/ GoToE** označuje Embedded Go-To action a ty nesmí odkazovat na externí PDF soubor (tag **/F**) na internetu, v souborovém systému nebo na síti. |
| **Očekávaný projev:** | Zpráva s kódem 1006 |
| **Důvod:** | Embedded Go-To akce nesmí odkazovat na externí struktury, mimo strukturu PDF dokumentu, externí obsah by mohl být změněn. |
| **Posouzení bezpečnosti:** | Uživatel by měl být varován, pokud dokument odkazuje na externí PDF soubory například na Internetu, v souborovém systému počítače nebo na síti. Dokument nemá žádnou kontrolu nad takovýmto linkovaným obsahem, který může být změněn. |

**Příklad 4.22 – XFA-based documents**

```
4 0 obj % Link to an embedded file in an external document
<< /Type /Action
/S /GoToE
/D (Chapter 1)
/F (someFile.pdf)
/T << /R /C
/N (Embedded document) >>
>>
endobj
```

Převzato ze specifikace [1] příklad 8.2, straně 658.

## 4.7 Nepovolené typy anotací

**Nepodařilo se v Acrobatu reprodukovat.**

**Tabulka 4.16 – Nepovolené typy anotací**

| | |
|---|---|
| **SigQ kód:** | **předpokládaný kód 1007** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.4 Annotations, strana 604 |
| **Klíč:** | **/Type/Annot**, **/Subtype/**XXYY, kdy XXYY je subtyp z tab. 8.20 na straně 615 |
| **Popis:** | Jedno nebo více formulářových polí má být nějak spojeno s 3D objekty, přílohami, multimédii nebo dalšími dynamickými objekty. |
| **Očekávaný projev:** | Zpráva s kódem 1007 |
| **Posouzení bezpečnosti:** | Nemáme dostatek informací o této chybě pro posouzení bezpečnosti. |

## 4.8 Nepovolené typy akcí

Tabulka typů akcí/Action types, z [1] kapitola 8.5.3, tabulka 8.48, strana 653

| Typ akce | Povoleno | Popis |
|---|---|---|
| **GoTo** | ANO | Zobrazí místo v prohlíženém dokumentu. |
| **GoToR** | NE | Zobrazí místo v jiném dokumentu. |
| **GoToE** | ANO[1] | Zobrazí místo ve vloženém dokumentu. |
| **Launch** | NE | Spustí nějakou cizí aplikaci, obvykle k otevření souboru. |
| **Thread** | ANO | Začne číst určitý článek. |
| **URI** | NE | Jde na určité URI (jednotný identifikátor zdroje). |
| **Sound** | NE | Přehraje zvuk. |
| **Movie** | NE | Přehraje video. |
| **Hide** | ANO | Nastaví „Hidden flag“ u anotace. |
| **Named** | NE[2] | Vykoná nějakou akci předdefinovanou v PDF prohlížeči. |
| **SubmitForm** | ANO | Pošle data na určité URL. |
| **ResetForm** | ANO | Nastaví hodnoty polí na jejich výchozí hodnoty. |
| **ImportData** | NE | Importuje hodnoty polí formuláře ze souboru. |
| **JavaScript** | NE | Spustí vykonávání JavaScriptu. |
| **SetOCGState** | NE | Nastaví stav OCG (optional content groups) |
| **Rendition** | Nevíme | Ovládá přehrávání multimediálního obsahu. |
| **Trans** | NE[4] | Updatuje zobrazení dokumentu použitím „transition dictionary“ |
| **GoTo3DView** | NE | Nastaví pohled v 3D anotaci. |
| **RichMediaExecute**[3] | NE | Provede předem definovanou akci s multimédii. |

1. Kvůli vloženému souboru dokument obsahuje JavaScript a ten vyvolává kód 1008
2. Named actions vyvolávají kód 1003, popsány v kapitole 4.3 Položky nabídek/Named actions, ale nezpůsobují kód 1008.
3. V dokumentaci [1] není zmíněno, nalezeno náhodou při tvorbě příkladů.
4. Tag **`/Trans`**, zmíněný v kapitole 4.4, vyvolával společně s tagem **`/Dur`** chybu 4000.

### 4.8.1 Nepovolený typ akce: URI

**Tabulka 4.17 – Nepovolený typ akce: URI**

| | |
|---|---|
| **SigQ kód:** | **1008,** 1002/1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> URI Actions, strana 662 |
| **Klíč:** | **`/S/URI`** |
| **Popis:** | Klíč **`/S`** s hodnotou **`/URI`** označuje URI action. |
| **Očekávaný projev:** | 1008 – Akce URI |
| **Soubor:** | `Hidden_actions_ 1001_1008/vazba_URI_1008_1002.pdf` |
| **Postup:** | panel Nástroje -> Obsah -> Vazba -> Vyplnit nějaké URI |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s vazbou vyvolávají kód 1008. |
| **Důvod:** | Akce URI otevře webovou stránku, tato akce nemusí být uživateli známa, proto jde o skrytou akci, kód 1008. |
| **Posouzení bezpečnosti:** | Považujeme za bezpečné, ale uživatel musí počítat s tím, že obsah odkazované stránky se může změnit, popř. stránka nemusí v budoucnu existovat.<br><br>Plný Acrobat 10 nedovoluje odskok na takové URL, ledaže by byl v jeho konfiguraci explicitně povolen přístup na daný webový server (seznamu důvěryhodných webů v předvolbách Správce práv) nebo byl povolen přístup na všechna URL. Acrobat |

| | |
|---|---|
| | Reader 9.4 se nejprve explicitně dotáže, zda uživatel skutečně chce k dané stránce přistoupit. |

**Příklad 4.23 – Nepovolený typ akce: URI**

```
20 0 obj
<</S/URI/URI(www.fel.cvut.cz)>>
endobj
```

### 4.8.2 Nepovolený typ akce: Sound

**Tabulka 4.18 – Nepovolený typ akce: Sound**

| | |
|---|---|
| **SigQ kód:** | **1008**, 1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Sound Actions, strana 663 |
| **Klíč:** | **/S/Sound** |
| **Popis:** | Klíč **/S** s hodnotou **/Sound** označuje Sound action. |
| **Očekávaný projev:** | 1008 – Akce Sound |
| **Soubor:** | Hidden_actions_1001_1008/actionSound_1008_1001.pdf |
| **Postup:** | panel Nástroje -> Obsah -> Tlačítko<br>Všechny vlastnosti/Vlastnosti -> Akce -> Vybraná akce: Přehrát zvuk |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s vazbou vyvolávají kód 1008. |
| **Důvod:** | Akce Sound zahájí přehrávání předem definovaného zvuku, tato akce nemusí být uživateli známa, proto jde o skrytou akci, kód 1008.<br><br>Kód 1001 je způsoben tlačítkem, popsáno v tomto dokumentu sekce 4.2 Annotations. |
| **Posouzení bezpečnosti:** | Nepovažujeme za nebezpečné. |

**Příklad 4.24 – Nepovolený typ akce: Sound**

```
27 0 obj
<<
/S /Sound
/Sound 16 0 R
>>
endobj
```

### 4.8.3 Nepovolený typ akce: Movie

**Nepodařilo se replikovat**, nejspíše se používalo ve starších verzích Acrobatu. Údaje o této chybě jsou pouze z dokumentace. Očekáváno podobné chování jako v akci popsané v sekci 4.8.4 Nepovolený typ akce: RichMediaExecute.

**Tabulka 4.19 – Nepovolený typ akce: Movie**

| | |
|---|---|
| **SigQ kód:** | předpokládané kódy **1008**, 1001, 1002 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> MovieActions, strana 664 |
| **Klíč:** | **/S/Movie** |
| **Popis:** | Klíč **/S** s hodnotou **/Movie** označuje Movie action. |

| | |
|---|---|
| **Posouzení bezpečnosti:** | Nepovažujeme za nebezpečné. |

### 4.8.4 Nepovolený typ akce: RichMediaExecute

**Tabulka 4.20 – Nepovolený typ akce: RichMediaExecute**

| | |
|---|---|
| **SigQ kód:** | **1008**, 1001, 1002 PDF/SigQ-1B |
| **Odkaz:** | Tento podtyp v dokumentaci [1] není zmíněn, ale jde o podtyp Action type [1] kapitola 8.5.3 Action Types, strana 652 |
| **Klíč:** | **`/S/RichMediaExecute`** |
| **Popis:** | Klíč **`/S`** s hodnotou **`/RichMediaExecute`** označuje akci, která spustí přehrávání multimédia. |
| **Očekávaný projev:** | 1008 – Akce Movie -> Předpoklad se nenaplnil, Acrobat X Pro používá akci RichMediaExecute. |
| **Soubor:** | `Hidden_actions_1001_1008/ prehrajVideo_RichMediaExecute_1002_1001_1008.pdf` |
| **Postup:** | Použijeme příklad s vloženým videem: `Multimedia_video_1002.pdf` panel Nástroje -> Obsah -> Tlačítko<br>Všechny vlastnosti/Vlastnosti -> Akce -> Vybraná akce: Multimediální operace (Acrobat 9 a novější) -> Přidat -> např. Spustit |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s action type RichMediaExecute vyvolávají kód 1008. |
| **Důvod:** | Akce RichMediaExecute provede předem definovanou akci s multimédii, tato akce nemusí být uživateli známa, proto jde o skrytou akci, která je označená kódem 1008.<br><br>Kód 1002 je způsoben anotací, ve které je video umístěno viz sekce 4.2.3.2 Vložené video.<br><br>Kód 1001 je způsoben tlačítkem, popsáno v tomto dokumentu sekce 4.2 Annotations. |
| **Posouzení bezpečnosti:** | V dokumentaci není specifikováno. Experimentálně jsme našli jediný typ – Video, které nepovažujeme za nebezpečné (Flash má svůj vlastní typ), otázkou je, jaké další typy multimédií mohou do této kategorie spadat. |

**Příklad 4.25 – Nepovolený typ akce: RichMediaExecute**

```
47 0 obj
<</CMD<</C(multimedia_play)>>/S/RichMediaExecute/TA 19 0 R>>
endobj
```

### 4.8.5 Nepovolený typ akce: Launch

**Tabulka 4.21 – Nepovolený typ akce: Launch**

| | |
|---|---|
| **SigQ kód:** | **1008,** 1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Launch Actions, strana 662 |
| **Klíč:** | **`/S/Launch`** |
| **Popis:** | Klíč **`/S`** s hodnotou **`/Launch`** označuje launch action. |
| **Očekávaný projev:** | 1008 – Akce Launch |
| **Soubor:** | `Hidden_actions_1001_1008/button_1001_1008_Launch.pdf` |

| | |
|---|---|
| **Postup:** | panel Nástroje -> Obsah -> Tlačítko<br>Kliknout pravým tlačítkem myši na právě vytvořené tlačítko v PDF dokumentu -> Vlastnosti -> Akce -> Otevřít soubor |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s vazbou vyvolávají kód 1008 – Nepovolený typ akce: Launch |
| **Důvod:** | Launch actions otevře nějaký soubor, tato akce nemusí být uživateli známa, proto jde o skrytou akci. |
| **Posouzení bezpečnosti:** | Akce Launch spustí externí aplikaci, obvykle za účelem otevření souboru nebo tisku dokumentu, tato akce **je nebezpečná**, PDF dokument nemá žádnou kontrolu nad spouštěnou aplikací ani nad otvíraným či tištěným dokumentem. |

**Příklad 4.26 – Nepovolený typ akce: Launch**

```
18 0 obj
<</A 21 0 R/AP<</N 22 0 R>>/DA(/HeBo 12 Tf 0 g)/F 4/FT/Btn/Ff
65536/MK<</BG[0.75293]>>/P 9 0 R/Rect[205.245 777.606 277.245
797.606]/Subtype/Widget/T(Test)/Type/Annot>>
endobj

21 0 obj
<</F 23 0 R/NewWindow false/S/Launch>>
endobj

23 0 obj
<</F(URI_1008_1002.pdf)/Type/Filespec/UF(URI_1008_1002.pdf)>>
endobj
```

### 4.8.6 Nepovolený typ akce: GoToR

**Tabulka 4.22 – Nepovolený typ akce: GoToR**

| | |
|---|---|
| **SigQ kód:** | **1008,** 1002 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Remote Go-To Actions, strana 655 |
| **Klíč:** | **/S/GoToR** |
| **Popis:** | Klíč **/S** s hodnotou **/GoToR** označuje Remote Go-To Action. |
| **Očekávaný projev:** | Otevření externího dokumentu, chyba 1008. |
| **Soubor:** | `Hidden_actions_1001_1008/1008_GoToR.pdf` |
| **Postup:** | panel Nástroje -> Obsah -> Vazba -> Vybrat objekt-> Vlastnosti -> Akce -> Vybraná akce: Jít na zobrazení stránky -> Kliknutím nastavit vazbu na nějaký externí dokument. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s Remote Go-To Actions vyvolávají kód 1008. |
| **Důvod:** | Akce Remote Go-To skočí na zadané místo v externím dokumentu, tato akce nemusí být uživateli známa, proto jde o skrytou akci. Nad obsahem externího dokumentu nemáme kontrolu. |
| **Posouzení bezpečnosti:** | Viz výše. |

**Příklad 4.27 – Nepovolený typ akce: GoToR**

```
25 0 obj
<</A 26 0 R/BS<</S/S/Type/Border/W 1>>/Border[0 0
1]/H/I/Rect[67.0544 749.793 171.426
819.18]/Subtype/Link/Type/Annot>>
endobj
26 0 obj
<</D[0/FitH 844]/F 27 0 R/S/GoToR>>
endobj
27 0 obj
<</F(empty – kopie \(4\).pdf)/Type/Filespec/UF(empty … kopie
\(4\).pdf)>>
endobj
```

### 4.8.7 Nepovolený type akce: ImportData

**Tabulka 4.23 – Nepovolený type akce: ImportData**

| | |
|---|---|
| **SigQ kód:** | **1008<<ImportData>>,** 1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Import-Data Actions, strana 708 |
| **Klíč:** | **/S/ImportData** |
| **Popis:** | Klíč **/S** s hodnotou **/ImportData** označuje Import-Data Actions. Tato akce nastaví formulářová pole na předem uložené hodnoty, které načte ze souboru specifikovaného klíčem /F. |
| **Očekávaný projev:** | Akce importování dat formuláře by měla vyvolat chybu 1008. |
| **Soubor:** | Hidden_actions_1001_1008/importForm_1008_1001.pdf |
| **Postup:** | panel Nástroje -> Formuláře -> Vytvořit, do nového formuláře přidáme Textové pole a Tlačítko.<br>Pak klikneme na „Ukončit úpravy formuláře“.<br>Do textového pole napíšeme nějaký text a pomocí „panel Nástroje -> Formuláře -> Další volby pro formulář -> Správa dat formuláře -> Exportovat data“ uložíme vyplněnou hodnotu formuláře do FDF souboru.<br>panel Nástroje -> Obsah -> Vybrat objekt, pak kliknout pravým tlačítkem myši na vytvořené tlačítko v PDF dokumentu -> Vlastnosti -> Akce -> Vybraná akce: Importovat data formuláře -> Přidat a vybereme námi vytvořený FDF soubor. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s Import-Data Actions vyvolávají kód 1008. |
| **Důvod:** | Akce ImportData nastaví formulářová pole na předem uložené hodnoty, tato akce nemusí být uživateli známa, proto jde o skrytou akci. |
| **Posouzení bezpečnosti:** | V podepisovaném PDF dokumentu je uložen pouze odkaz na externí Acrobat FDF (Forms Data Format) soubor, z něhož se mají data do podepisovaného PDF importovat. FDF soubor není nikterak chráněn před modifikací.<br><br>Import dat vyvolaný po podepsání dokumentu nepředstavuje bezpečnostní riziko, neboť na změnu obsahu dokumentu po podpisu je uživatel upozorněn – viz sekce *3.2 Inkrementální update*.<br><br>Jiné (větší) nebezpečí spočívá v tom, že by formulář vyvolal import dat před podpisem bez uživatelova vědomí. Uživatel by v takovém případě podepisoval formulář s přesvědčením, že podpisuje jím ručně vyplněná data, ve skutečnosti by však |

| | |
|---|---|
| | podepisoval data importovaná.<br><br>Příkladem je formulář `Hidden_actions_1001_1008/importForm_podvod_1008_1001.pdf` spolu s datovým souborem `importForm_1008_1001_data.fdf`.<br><br>Uživatel vyplní hodnotu políčka určitou hodnotou a poté při vyplňování jiné položky v dolní části dokumentu (první políčko už uživatel nemusí vidět) nevědomky spustí import dat z uvedeného datového souboru, čehož si, v případě, že je již na jiné stránce, nevšimne. Podepisuje tedy jiná data, než jaká vyplnil. |

**Příklad 4.28 – Nepovolený type akce: ImportData**

```
28 0 obj
<<
/F 29 0 R
/S /ImportData
>>
endobj
```

### 4.8.8 Nepovolený type akce: SetOCGState

**Tabulka 4.24 – Nepovolený type akce: SetOCGState**

| | |
|---|---|
| **SigQ kód:** | **1008<<SetOCGState>>,** 1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Set-OCG-State Actions, strana 667 |
| **Klíč:** | **`/S/SetOCGState`** |
| **Popis:** | Klíč **`/S`** s hodnotou **`/SetOCGState`** označuje Set-OCG-State Actions. Tato akce nastaví předem uložené nastavení viditelnosti jednotlivých vrstev. Viditelnost jednotlivých vrstev je specifikována v hodnotě klíče /State, kterou je pole hodnot z množiny /ON, /OFF, /Toggle (zobrazené vrstvy, skryté vrstvy a vrstvy, jejichž viditelnost se má invertovat). Každá tato hodnota (pokud se v poli vůbec vyskytuje) je následována žádným (jako v případě hodnoty /OFF v níže uvedeném příkladu) nebo více odkazy na objekty typu /OCG, na které se dané nastavení viditelnosti aplikuje.<br>V níže uvedeném příkladu pole s klíčem /State:<br>• neobsahuje žádnou hodnotu /Toggle, u žádné vrstvy tedy není požadováno invertování viditelnosti<br>• obsahuje hodnotu /OFF, ale protože ta neodkazuje na žádné objekty typu /OCG, není žádná vrstva skrytá<br>• obsahuje hodnotu /ON, která specifikuje, že objekty 24, 25 a 44 mají být viditelné. |
| **Očekávaný projev:** | Akce nastavení uložené viditelnost vrstev by měla vyvolat chybu 1008. |
| **Soubor:** | `Hidden_actions_1001_1008/vrstvy_SetOCGState_1008_1001.pdf` |
| **Postup:** | Na liště vlevo vybereme panel vrstvy a nastavíme stav viditelnosti vrstev tak, jak ho chceme uložit.<br>panel Nástroje -> Obsah -> Tlačítko<br>Kliknout pravým tlačítkem myši na právě vytvořené tlačítko v PDF dokumentu -> Vlastnosti -> Akce -> Vybraná akce: Nastavit viditelnost vrstvy -> Přidat... |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008. |
| **Pozorování:** | Soubory s Set-OCG-State Actions vyvolávají kód 1008. |

| | |
|---|---|
| **Důvod:** | Akce Set-OCG-State nastaví předem uložené nastavení viditelnost jednotlivých vrstev v dokumentu, tato akce nemusí být uživateli známa, proto jde o skrytou akci. |
| **Posouzení bezpečnosti:** | To, co se na úrovni GUI prohlížeče (PDF viewer) nazývá vrstvami, je ve specifikaci nazýváno OCG (Optional Content Groups). Akce SetOCGState nastaví viditelnost vrstev (OCG).<br><br>Akce SetOCGState by se dala zneužít například ke schování části dokumentu, která je v určité vrstvě. Uživatel si toho nemusí všimnout (změna se stala na jiné stránce, nebo jde jen o malou změnu).<br><br>Například v dokumentu by mohl být text nějaké smlouvy, uživatel si dokument prostuduje, vyplní nějaké údaje o své firmě (při vyplňování nějakého políčka se spustí akce, která skryje důležitý prvek, který je umístěný v nějaké vrstvě), uživatel dokument podepíše, uloží. Dokument se skrytou důležitou větou je podepsán. |

**Příklad 4.29 – Nepovolený type akce: SetOCGState**

```
52 0 obj
<</A 56 0 R/AP<</N 57 0 R>>/DA(/HeBo 12 Tf 0 g)/F 4/FT/Btn/Ff
65536/MK<</BG[0.75293]/CA(SetOCGState)>>/P 13 0 R/Rect[33.4966
427.199 174.351 465.808]/Subtype/Widget/T(SetOCGState)/Type/Annot>>
endobj

56 0 obj
<</PreserveRB false/S/SetOCGState/State[/OFF/ON 24 0 R 25 0 R 44 0
R]>>
endobj

24 0 obj
<<
/Name (skla)
/Type /OCG
>>
endobj

25 0 obj
<<
/Name ()
/Type /OCG
>>
endobj

44 0 obj
<</Name(odlesky)/Type/OCG>>
endobj
```

### 4.8.9 Nepovolený typ akce: GoTo3DView

**Tabulka 4.25 – Nepovolený type akce: GoTo3DView**

| | |
|---|---|
| **SigQ kód:** | **1008<<GoTo3DView>>,** 1001 PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Go-To-3D-View Actions, strana 670 |
| **Klíč:** | **`/S/GoTo3DView`** |
| **Popis:** | Klíč **`/S`** s hodnotou **`/GoTo3DView`** odkazuje na 3D anotaci a specifikuje pohled, který má tato 3D anotace použít. |
| **Očekávaný projev:** | Chyba 1008 – Go-To-3D-View akce. |
| **Soubor:** | `Hidden_actions_1001_1008/Hidden_action_GoTo3DView_1008_1002.pdf` |
| **Postup:** | • Vezmeme dokument, který slouží jako příklad k 3D anotacím: Comment_1002/comments_3D_1002_1008GoTo3DView.pdf<br>• Vložíme tlačítko -> Nástroje -> Obsah -> Tlačítko<br>• panel Nástroje -> Obsah -> Vybrat objekt (Vybereme tlačítko) -> Vlastnosti -> Akce -> Vybraná akce: Jít na zobrazení 3D/Multimédia ->Vybereme danou 3D anotaci |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1008, nepovolený typ akce: GoTo3DView. |
| **Pozorování:** | Soubory s Go-To-3D-View Actions vyvolávají kód 1008. |
| **Důvod:** | Akce Go-To-3D-View Actions změní pohled na pohled 3D anotaci, tato akce nemusí být uživateli známa, proto jde o skrytou akci. |
| **Posouzení bezpečnosti:** | Uživatel si prohlédne dokument s 3D anotací a souhlasí s ním. Pokud se při vyplnění jiného pole spustí akce nastavení pohledu 3D anotace, uživatel nevědomky podepíše jiný pohled, než předpokládal. Změny pohledu si uživatel u vícestránkového dokumentu nemusí vůbec všimnout. |

**Příklad 4.30 – Nepovolený type akce: GoTo3DView**

```
25 0 obj
<<
/A 62 0 R
/AP <<
/N 15 0 R
>>
/DA (/HeBo 12 Tf 0 g)
/F 4
/FT /Btn
/Ff 65536
/MK <<
/BG [ 0.75293 ]
/CA (Predchazejici pohled)
>>
/P 12 0 R
/Rect [ 158.015 767.262 336.015 795.857 ]
/Subtype /Widget
/T (Predchazejici pohled)
/Type /Annot
>>
endobj
```

```
62 0 obj
<<
/S /GoTo3DView
/TA 23 0 R
/V /P
>>
endobj
```

## 4.9 Povolené typy akcí

Povolenými akcemi nazýváme akce, u nichž Acrobat při pokusu o podpis nehlásí (na rozdíl od *disallowed* akcí) žádnou chybu. Při jejich analýze jsme však u některých z nich nalezli situace, kdy by nebezpečné být mohly. Detaily jsou diskutovány u každé z akcí.

### 4.9.1 Povolený typ akce: Go-To

| Tabulka 4.26 – Povolený typ akce: Go-To | |
|---|---|
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Go-To Actions, str. 654 |
| **Soubor:** | `AllowedActions/loremIpsumGoTo.pdf` |
| **Posouzení bezpečnosti:** | Nepovažujeme za nebezpečnou. |

**Příklad 4.31 – Povolený type akce: GoTo**

```
75 0 obj
<<
/D [ 11 0 R /FitH 797 ]
/S /GoTo
>>
endobj
```

### 4.9.2 Povolený typ akce: Embedded Go-To Actions

| Tabulka 4.27 – Povolený typ akce: Embedded Go-To Actions | |
|---|---|
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Embedded Go-To Actions, str. 655 |
| **Soubor:** | `AllowedActions/loremIpsumGoToE.pdf` |
| **Posouzení bezpečnosti:** | Pokud odkazuje na vložený (child) dokument nebo naopak z vloženého dokumentu na rodičovský (parent) či odkaz na sourozenecký (sibling) dokument, nepovažujeme odkaz za nebezpečný. Všechny tyto dokumenty totiž jsou vloženy v podepisovaném dokumentu. Z příkladů uvedených ve specifikaci [1] na str. 658 vyplývá, že může jít i o odskok na PDF soubor vložený (embedded) do externího PDF souboru. Tuto situaci za nebezpečnou považujeme. |

**Příklad 4.32 – Povolený type akce: Embedded Go-To Actions**

```
37 0 obj
<<
/D [ 6 /FitH 763 ]
/S /GoToE
/T <<
```

```
/N (...binární data...)
/R /C
>>
>>
endobj
```

### 4.9.3 Povolený typ akce: Hide Actions

| Tabulka 4.28 – Povolený typ akce: Hide Actions | |
|---|---|
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Hide Actions, str. 665 |
| **Soubor:** | `AllowedActions/actionHide_1001_2013_2014.pdf` |
| **Posouzení bezpečnosti:** | Ačkoli Acrobat uvedenou akci nepovažuje za nebezpečnou a nikterak před ní nevaruje, domníváme se, že by povolena být neměla nebo alespoň by na ni měl být při podpisu uživatel upozorněn.<br><br>Zobrazení a skrytí určitého formulářového pole by se dalo zneužít ke schování uživatelem vyplněného pole případně i jeho nahrazením jiným polem.<br><br>K problematické situaci by mohlo dojít tehdy, kdyby se dvě políčka formuláře překrývala. Pokud by byla vyvolána akce Hide na „vrchní“ políčko, objevilo by se ve formuláři „spodní políčko, které uživatel při vyplňování neviděl. Následně by tak podepisoval obsah, o němž by měl jinou představu.<br><br>**Soubor:** Hidden_actions_1001_1008/Form_podvod_HideAction.pdf |

**Příklad 4.33 – Povolený typ akce: Hide Actions**

```
24 0 obj
<<
/A 28 0 R
/AP <<
/N 14 0 R
>>
/DA (/HeBo 12 Tf 0 g)
/F 4
/FT /Btn
/Ff 65536
/MK <<
/BG [ 0.75293 ]
/CA (zneviditelniTlacitko)
>>
/P 12 0 R
/Rect [ 47.2296 721.047 230.598 742.796 ]
/Subtype /Widget
/T (zveviditelniTlacitko)
/Type /Annot
>>
endobj
```

```
28 0 obj
<<
/S /Hide
/T (testovaciTlacitko)
>>
endobj
```

### 4.9.4 Povolený typ akce: Submit-Form

| Tabulka 4.29 – Povolený typ akce: Submit-Form | |
|---|---|
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Submit-Form Actions, str. 703 |
| **Soubor:** | `AllowedActions/submitForm.pdf` |
| **Posouzení bezpečnosti:** | Nepovažujeme za nebezpečnou. |

**Příklad 4.34 – Povolený typ akce: Submit-Form**

```
28 0 obj
<<
/F <<
/F (www.fel.cvut.cz)
/FS /URL
>>
/Flags 704
/S /SubmitForm
>>
endobj
```

### 4.9.5 Povolený typ akce: Reset-Form

| Tabulka 4.30 – Povolený typ akce: Reset-Form | |
|---|---|
| **Odkaz:** | [1] kapitola 8.5.3 Action Types -> Reset-Form Actions, str. 707 |
| **Soubor:** | `AllowedActions/resetForm.pdf` |
| **Posouzení bezpečnosti:** | Na Akci Reset Acrobat nijak neupozorní, považuje ji zřejmě za bezpečnou. Domníváme se však, že by neměla být povolena nebo alespoň by na ni měl být při podpisu uživatel upozorněn. Šlo by ji zneužít k nastavení určitých polí do výchozích hodnot, například při editování polí na konci dokumentu nebo při stisku tlačítka „Odeslat“.<br><br>Pokud je pole (v závěru rozsáhlého formuláře) spojeno s akcí Reset, bude uživatelem již vyplněná hodnota jiného pole (na počátku rozsáhlého formuláře) přepsána na hodnotu výchozí. Protože toto pole je mimo rozsah obrazovky, uživatel si změny nevšimne a podepíše dokument s vědomím, že podepisuje jím vyplněnou (nikoliv výchozí) hodnotu daného pole.<br><br>**Soubor:** `Hidden_actions_1001_1008/Form_podvod_ResetAction.pdf` |

**Příklad 4.35 – Povolený typ akce: Reset-Form**

```
28 0 obj
```

```
<<
/Fields [ ]
/Flags 1
/S /ResetForm
>>
endobj
```

## 4.10 Vrstvy

**Tabulka 4.31 – Vrstvy**

| | |
|---|---|
| **SigQ kód:** | **1009** PDF/SigQ-1B |
| **Odkaz:** | [1] Kapitola 4.10 Optional Content Groups, str. 364 |
| **Klíč:** | **/ExportState, /PrintState** nebo **/ViewState** v **/Usage** v Optional Content dictionary (**/Type/OCG**) nastaveno na **/ON** nebo **/OFF** |
| **Popis:** | Pokud Klíč **Usage** v **Optional Content dictionary** specifikuje (ne)viditelnost vrstvy za nějaké situace, to znamená, že **ExportState**, **PrintState** nebo **ViewState** jsou nastaveny na **ON** nebo **OFF**, stačí když je nastavena jen jedna situace, pak dokument nesplňuje normu SigQ-1A. |
| **Očekávaný projev:** | Chybová hláška 1009 |
| **Soubor:** | Layers_1009/vrstvy_1009.pdf |
| **Postup:** | Vlevo zobrazíme panel „Vrstvy“, vybereme nějakou vrstvu, pravým tlačítkem zobrazíme kontextovou nabídku, zvolíme možnost „Vlastnosti...“, tam nastavíme alespoň u jedné položky viditelnost jinou než „Viditelná/Tiskne/Exportuje, když je viditelná“. (Např. Viditelnost: Vždy neviditelná, Tisk: Vždy se tiskne, Export: Exportuje se, když je viditelná, toto je nastaveni vrstvy odlesky z příkladu). |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 1009. |
| **Pozorování:** | Tento soubor Acrobatem bez problémů zobrazen. V režimu náhledu je zobrazeno varování: „Tento dokument obsahuje některé konstrukty, jako jsou například pole formulářů, multimédia nebo JavaScript, což může ovlivnit jeho vzhled“, ve zprávě je uvedena chyba 1009 (Obsah dokumentu je rozdělen do vrstev, které mohou být za běhu bez upozornění zobrazeny nebo skryty). |
| **Důvod:** | Pokud se v dokumentu nacházení vrstvy, které mají nastavené, kdy se mají exportovat, tisknout a zobrazit, tak může uživatel něco jiného vidět na monitoru a něco jiného tisknout. |
| **Posouzení bezpečnosti:** | Pokud je v Optional Content Group specifikována viditelnost vrstvy za nějaké situace (klíče: **/ExportState, /PrintState** nebo **/ViewState**), tak se zobrazení vrstev nechová tak, jak si je nastaví uživatel v panelu na „Vrstvy“, který je na levé straně, ale jejich viditelnost se řídí nastavením výše zmíněných klíčů. Při exportu, tisku nebo při příštím zobrazení může být zobrazeno něco jiného než to, co uživatel očekával, že teď podepisuje.<br>Za jedinou přípustnou kombinaci tedy považujeme situaci, kdy žádný z výše uvedených klíčů není specifikován. Acrobat v takové situaci slovník /Usage do dokumentu vůbec nezařazuje. |

**Příklad 4.36 – Vrstvy**

```
44 0 obj
<<
/Name(odlesky)/Type/OCG
/Usage<</Print<</PrintState/ON>>/View<</ViewState/OFF>>>>
>>
endobj
```

**Ve složce** Layers_1009 **jsou soubory se všemi kombinacemi** `ExportState`**,** `PrintState` **a** `ViewState`**. Tabulka 4.32 shrnuje výsledek pokusů.**

Soubory ve složce jsou pojmenovány podle jednotné konvence, která zohledňuje nastavení OCG. Například jméno souboru `vrstvy_1009_E-_Pon_Voff_1009_decompressed.pdf` odpovídá situaci, kdy `ExportState` není nastaven (viz pomlčka za písmenem „E“ v názvu), `PrintState` je nastavený na „ON“ a `ViewState` na „OFF“.

| Tabulka 4.32 – Výsledek pokusu s vrstvami | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **ExportState** | - | ON | - | - | OFF | - | - | ON | - |
| **PrintState** | - | - | ON | - | - | OFF | - | ON | ON |
| **ViewState** | - | - | - | ON | - | - | OFF | - | ON |
| **Kód** | - | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **ExportState** | ON | OFF | - | OFF | ON | OFF | ON | OFF | - |
| **PrintState** | - | OFF | OFF | - | ON | OFF | OFF | ON | ON |
| **ViewState** | ON | - | OFF | OFF | ON | OFF | - | - | OFF |
| **Kód** | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **ExportState** | - | ON | OFF | OFF | ON | ON | OFF | OFF | ON |
| **PrintState** | OFF | - | - | ON | OFF | ON | OFF | ON | OFF |
| **ViewState** | ON | OFF | ON | ON | ON | OFF | ON | OFF | OFF |
| **Kód** | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 | 1009 |

## 4.11 PostScript XObjects

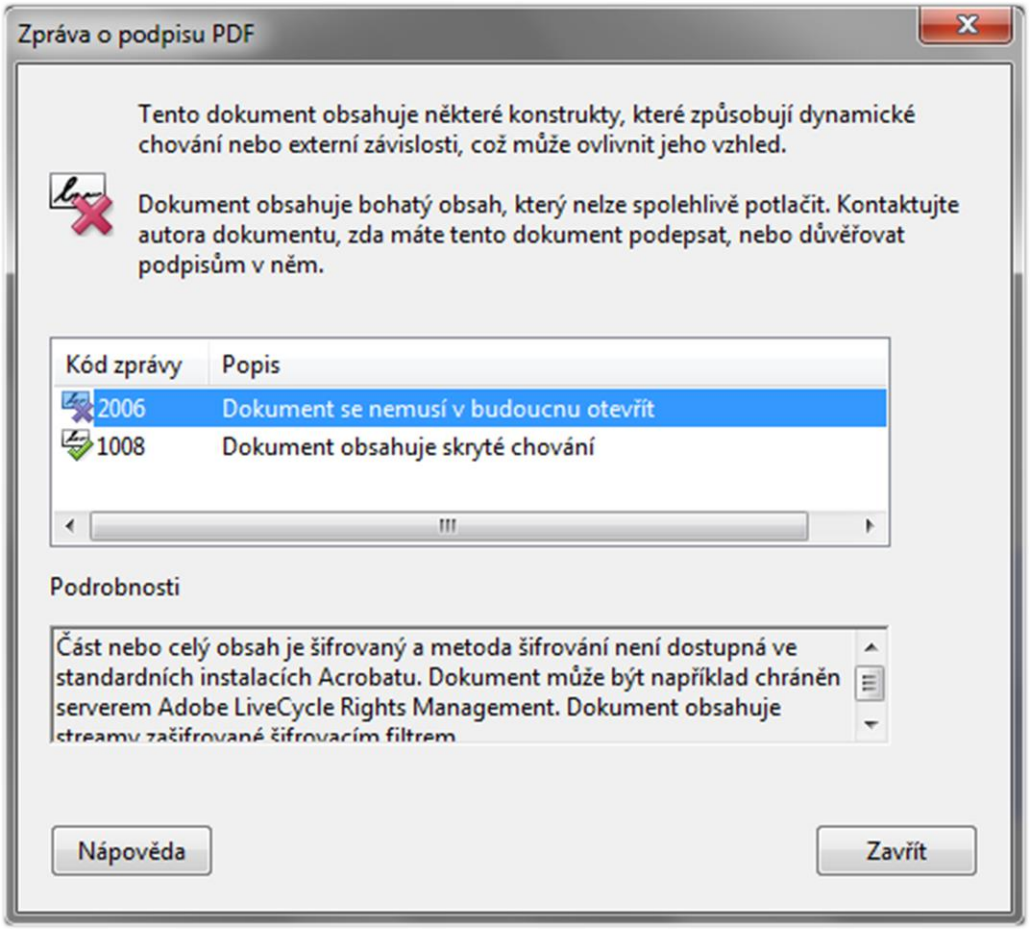
| Tabulka 4.33 – PostScript XObjects | |
|---|---|
| **SigQ kód:** | **2004** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 4.7.1 PostScript XObjects, strana 333 |
| **Klíč:** | **`/Subtype /PS`** nebo **`/Subtype2 /PS`** |
| **Popis:** | Klíč **`/Subtype`** nebo **`/Subtype2`** s hodnotou **`/PS`** v XObject dictionary označuje Postscript XObject. Od PDF verze 1.4 není důvod používat tyto objekty, bude odstraněno v budoucích verzích PDF. |
| **Očekávaný projev:** | Chybová hláška 2004 |
| **Soubor:** | `PostScript_XObject_2004/PDFTestFile_2004_4002_4001_1002.pdf` |
| **Zdroj:** | `http://www.planetpdf.com/forumarchive/160134.asp` |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 2004. |
| **Pozorování:** | Soubory s PostScript XObject vyvolávají kód 2004. |
| **Důvod:** | PostScript je ve své podstatě skriptovací jazyk, a proto vizuální elementy se mohou |

| | |
|---|---|
| | měnit podle externích proměnných. Například logo může měnit barvu podle času nebo úrovně zvětšení. |
| **Posouzení bezpečnosti:** | **Nebezpečné**, zdůvodnění výše. |

**Příklad 4.37 – PostScript XObjects**

```
13 0 obj
<<
/Type /XObject
/Subtype /Form
/Subtype2 /PS
/BBox [0.0000 0.0000 191.9996 275.9998]
/Length 1502032
>>
```

## 4.12 Nestandardní šifrování



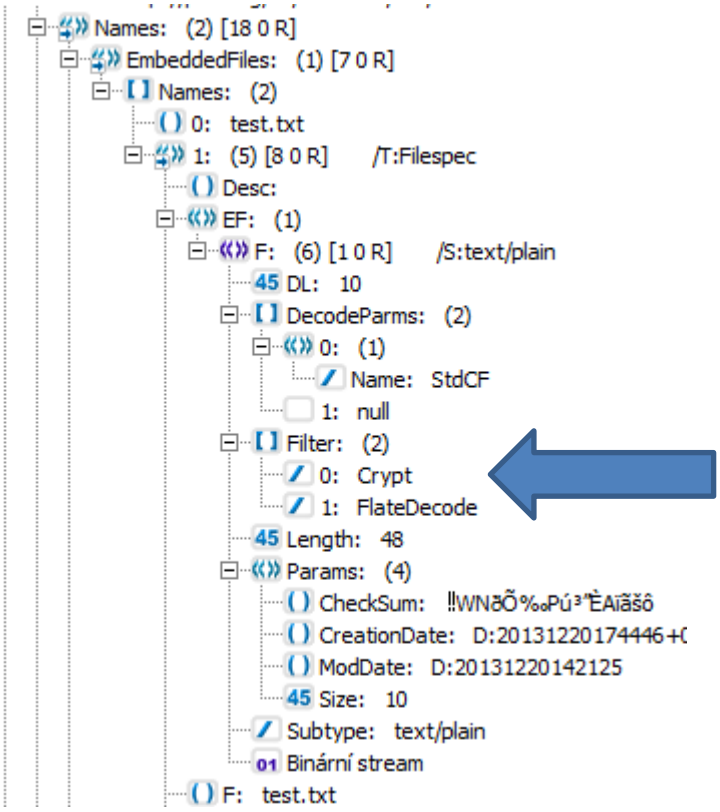

**Obrázek 1 – Zpráva o podpisu PDF v aplikaci Adobe Acrobat 10**

| Tabulka 4.34 – Nestandardní šifrování | |
|---|---|
| **SigQ kód:** | **2006** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 3.5 Encryption, strana 115 |
| **Klíč:** | **`/Filter[/Crypt...]`** |
| **Popis:** | Klíč **`/Filter`**, který obsahuje hodnotou **`/Crypt`** označuje šifrované streamy. Jeden stream může mít více filtrů např. **`/Filter[/Crypt/FlateDecode]`** |
| **Očekávaný projev:** | Chybová hláška 2006 |
| **Soubor:** | `sifrovani_2006/empty_priloha_123456_2006.pdf` |
| **Postup:** | panel Nástroje -> Obsah -> Přiložit soubor, Ochrana- > Zašifrovat -> 2 Zašifrovat pomocí hesla -> Kompatibilita: Acrobat X a novější, Zašifrovat pouze přiložené soubory, soubor se zašifruje až po uložení. |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny kódy 1008 a 2006. |
| **Pozorování:** | Soubory se šifrovanou přílohou vyvolávají zprávu s kódem 2006. |
| **Posouzení bezpečnosti:** | Použití nestandartního šifrování **je celkem bezpečné**. Šifrované vložené soubory se nejspíše v Acrobatu, který dané šifrování nepodporuje, vůbec nezobrazí, nebo se při otvírání souboru zobrazí chyba. Nevidíme zde možnost zneužití dokumentu či zmatení uživatele, které by spočívalo v odlišnosti obsahu/vzhledu dokumentu před a po podpisu. |

### Příklad 4.38 – Nestandardní šifrování

```
1 0 obj
<</DL 10/DecodeParms[<</Name/StdCF>>null]
/Filter[/Crypt/FlateDecode]/Length 48
/Params<</CheckSum(WNđŐ‹ PúłŽČAďăťô)
/CreationDate(D:20131220174446+01'00')
/ModDate(D:20131220142125)/Size 10>>
/Subtype/text#2Fplain>>
stream
...binární data...
endstream
endobj

11 0 obj
<</CF<</StdCF<</AuthEvent/EFOpen/CFM/AESV3/Length 32>>>>
/EFF/StdCF/EncryptMetadata false/Filter/Standard/Length
256/O(...binární data...
```




**Obrázek 2 – Procházení objektů uvnitř PDF aplikaci Adobe Acrobat 10**

## 4.13 Interpolované obrázky

**Tabulka 4.35 – Interpolované obrázky**

| | |
|---|---|
| **SigQ kód:** | **2007** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] Kapitola 4.8.4 Image Dictionaries, strana 340, 346 |
| **Klíč:** | **`/Interpolate true`** v Image dictionary |
| **Popis:** | Pokud je klíč **`/Interpolate`** v Image dictionary **`true`**, tak dokument nesplňuje normu SigQ, interpolace obrázků není dovolena. |
| **Očekávaný projev:** | Zpráva s kódem 2007 |
| **Soubor:** | `Image_interpolation_2007/iterpolate_2007.pdf` |
| **Zdroj:** | http://www.pdfscripting.com/public/FreeStuff/PDFSamples/BouncingButton.pdf **(Antivir Avira považuje tento soubor za nebezpečný**, nejspíš kvůli příliš bohatému JavaScriptu – tlačítko „utíká“ před myší), soubor byl upraven, JavaScript kompletně odstraněn, aby byla izolována pouze chyba 2007.) |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 2007. |
| **Pozorování:** | Tento soubor Acrobatem bez problémů zobrazen. Při pokusu o podpis však hlásí: „*V tomto dokumentu je bohatý obsah, který nelze spolehlivě potlačit pomocí zobrazení náhledu dokumentu. Klepněte na Pokračovat, chcete-li přesto zobrazit náhled tohoto dokumentu.*“ Pokračujeme-li, v režimu náhledu Acrobat zobrazí chybu 2007 (Autor dokumentu povolil interpolace obrazu, interpolace obrazu není povolena) |
| **Důvod:** | Interpolace obrázků není dovolena. |
| **Posouzení** | Interpolace obrázků není podle normy SigQ dovolena, ovšem praktické riziko zneužití |

| | |
|---|---|
| **bezpečnosti:** | považujeme za nízké.<br><br>Ve specifikaci formátu PDF není specifikováno, jaký algoritmus by měl být pro interpolaci použitý (poznámka na straně 347, ve specifikaci [1]). Výsledek interpolace tedy může být závislý na algoritmu, který provádí interpolaci v použitém PDF prohlížeči. Interpolace obrázku by mohla způsobit např. chybné zobrazení tenkých čar nebo ostrých hran, které se interpolací změní v postupně přecházející hrany.<br><br>Dle specifikace [1], se tato interpolace projevuje tehdy, když je rozlišení zdrojového obrázku výrazně menší než rozlišení zobrazovacího zařízení.<br><br>Interpolace může způsobovat tyto nežádoucí artefakty: aliasing („zubatý obrázek“), bluring („rozmazaný obrázek“), edge halo („obrysy okolo hran“), záleží na použitém algoritmu, jak výrazné tyto obrazové artefakty budou. Možnost praktického zneužití této vlastnosti pro podvržení jiného obsahu při podpisu, než jaký bude zobrazen později, jsme nenašli. |

**Příklad 4.39 – Nestandardní šifrování**

```
14 0 obj
<</BitsPerComponent 8/ColorSpace 17 0 R/Filter/JPXDecode
/Height 157/Intent/RelativeColorimetric/Interpolate true
/Length 4722/Subtype/Image/Type/XObject/Width 121>>
```

## 4.14 Problémové položky v Extended graphics state dictionary

Odkaz na slovník typu **/ExtGState** se nachází například v objektech typu:

- Resource dictionary, který odkazován tagem **/Resources** např. z objektů:
  - Stránka (**/Type/Page**)
    - Specifikace [1] kapitola 3.6.2 Page Tree, strana 144, tab. 3.27
    - Příklad: extended_graphics_state/extGState01.pdf
  - Typ XObject a podtyp Formulář (**/Type/XObject**, **/Subtype/Form**)
    - Specifikace [1] kapitola 4.9.1 Form Dictionaries, strana 358, tab. 4.45
    - Příklad: extended_graphisc_state/extGState02.pdf
- Resource dictionary (specifikace [1] strana 154, tab. 3.30)
- Operand **gs** ((specifikace [1] kapitola 4.3.3 Graphics State Operators, strana 219, tab. 4.7)
- Type 2 pattern dictionary, specifikace [1] kap. 4.6.3 Shading Patterns, strana 302, tab. 4.26

Tyto objekty obsahují klíč **/ExtGState** a ten obsahuje odkaz na extended graphics state dictionary, který může volitelně obsahovat položku určující typ slovníku, která je nastavená ExtGState(**/Type/ExtGState**).

**Příklad 4.40 – Extended graphics state**

```
115 0 obj
<<
/First 10
/Length 458
/N 2
```

```
/Type /ObjStm
>>
stream
4 0 41 60 <</Count 2/Kids[41 0 R 67 0 R]/Resources 42 0
R/Type/Pages>><</ArtBox[0.0 0.0 595.28 841.89]/BleedBox[0.0 0.0
595.28 841.89]/Contents 96 0 R/CropBox[0 0 595.28 841.89]/Group 40 0
R/MediaBox[0 0 595.28 841.89]/Parent 4 0
R/Resources<<GS0 101 0 R>>/Font<</T1_0 12 0 R/T1_1 11 0
R/T1_2 13 0 R/T1_3 15 0 R/T1_4 17 0 R/T1_5 14 0 R/T1_6 16 0
R>>/ProcSet[/PDF/Text]/XObject<</Fm0 97 0 R>>>>/Rotate 0/TrimBox[0.0
0.0 595.28 841.89]/Type/Page>>
endstream
endobj

101 0 obj
<</CA 1.0/Type/ExtGState/ca 1.0>>
endobj
```

### 4.14.1 Klíč /TR

**Tabulka 4.36 – Klíč /TR**

| | |
|---|---|
| **SigQ kód:** | **2009,** 4000, 1008 nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 4.3.4, Graphics State Parameter Dictionaries, tab. 4.8, str. 220, transfer function popsána na straně 484, kapitola 6.3. Ukázka použití tagu TR na straně 224 v [1]. |
| **Klíč:** | **/TR** v Extended graphics state dictionary |
| **Popis:** | Pokud je klíč **/TR** v Extended graphics state dictionary, tak dokument nesplňuje normu SigQ, použití transfer function, která interpretuje a nahrazuje barvy, v Extended graphics state dictionary není dovoleno. |
| **Očekávaný projev:** | Zpráva s kódem 2009 |
| **Soubor:** | `extended_graphics_state/extGState_TR_2009.pdf` |
| **Zdroj:** | Tento soubor byl vytvořen přímou editací obsahu PDF dokumentu, tag /CA byl nahrazen tagem /TR, tím sice vznikla chyba neznámého obsahu PDF (kód 4000) dokumentu, ale podařilo se tím vyvolat chybu 2009 |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uvedeny kódy 2009, 4000 a 1008. |
| **Pozorování:** | Pouhá přítomnost tagu `/TR` (transfer function) v Extended graphics state dictionary (`/ExtGState`), i když není transfer function správně zapsána, vede k chybě 2009; |
| **Důvod:** | Klíč **/TR** by neměl být přítomen v Extended graphics state dictionary. |
| **Posouzení bezpečnosti:** | Pokud dokument používá transfer function (**/TR**), která interpretuje a nahrazuje barvy, bylo by možné barvu tisku (černou) nahradit barvou pozadí (bílou) a tím změnit význam obsahu dokumentu (Specifikace [3], strana 130, tab. 14) |

**Příklad 4.41 – Klíč /TR**

```
101 0 obj
<</TR 1.0/Type/ExtGState/TR 1.0>>
endobj
```

### 4.14.2 Klíč /TR2

**Nepodařilo se nám tuto situaci navodit**, vycházíme proto z dokumentace, kde je zmíněna podobnost s funkcí TR a popsán rozdíl mezi nimi.

**Tabulka 4.37 – Klíč /TR2**

| | |
|---|---|
| **SigQ kód:** | **2010,** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 4.3.4, Graphics State Parameter Dictionaries, tab. 4.8, str. 220, transfer function popsána na straně 484, kapitola 6.3<br>[3] strana 130, tab. 14 |
| **Klíč:** | **/TR2** v Extended graphics state dictionary, není nastavený na hodnotu default |
| **Popis:** | Pokud je klíč **/TR2** v Extended graphics state dictionary a má jinou hodnotu než „default“, tak dokument nesplňuje normu SigQ, použití transfer function, která interpretuje a nahrazuje barvy, v Extended graphics state dictionary není dovoleno. |
| **Očekávaný projev:** | Zpráva s kódem 2010 |
| **Pozorování:** | Samotný tag **/TR2** s odkazem nestačí k navození chyby 2010, protipříklad extGState_TR2_4000.pdf |
| **Posouzení bezpečnosti:** | Jediné, čím se položka TR2 liší od položky TR, je možnost specifikovat hodnotu „default“. To znamená, že se pro interpretaci barev použije výchozí transformační funkce. Domníváme se, že z hlediska bezpečnosti je použití výchozí transformační funkce ekvivalentní situaci, kdy by tag /TR2 nebyl specifikován vůbec a hodnota „default“ nepředstavuje z hlediska bezpečnosti žádný problém. Je-li však uveden odkaz na transformační funkci, tak platí pro položku TR2 platí totéž, co bylo uvedeno pro položku TR. |

### 4.14.3 Klíč /FL

**Tabulka 4.38 – Klíč /FL**

| | |
|---|---|
| **SigQ kód:** | **2011, 4000, 1008** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 4.3.4, Graphics State Parameter Dictionaries, tab. 4.8, str. 220, transfer function popsána na straně 484, kapitola 6.3 |
| **Klíč:** | **/FL** v Extended graphics state dictionary |
| **Popis:** | Pokud je klíč **/FL** v Extended graphics state dictionary, tak dokument nesplňuje normu SigQ, použití flatness tolerance v Extended graphics state dictionary není dovoleno. |
| **Očekávaný projev:** | Zpráva s kódem 2010 |
| **Soubor:** | `extended_graphics_state/extGState_FL_2009.pdf` |
| **Zdroj:** | Tento soubor byl vytvořen přímou editací obsahu PDF dokumentu, tag /CA byl nahrazen tagem /FL, tím sice vznikla chyba neznámého obsahu PDF (kód 4000) dokumentu, ale podařilo se tím vyvolat chybu 2009 |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny kódy 2009, 4000 a 1008. |
| **Pozorování:** | Pouhá přítomnost tagu `/FL` (transfer function) v Extended graphics state dictionary (`/ExtGState`) vede k chybě 2011. |

| | |
|---|---|
| **Důvod:** | Klíč **/FL** by neměl být přítomen v Extended graphics state dictionary. |
| **Posouzení bezpečnosti:** | Flatness tolerance je maximální přípustná chyba při aproximování spojitých křivek po částech lineárními úseky.<br><br>Vzhled křivky by se proto mohl při použití různých PDF prohlížečů lišit – mohla by být viditelná nahrazení křivky rovnými úseky. Specifikace [1], kap. 6.5.1 Flatness Tolerance, str. 509 – poslední řádek nad obrázkem 6.6 říká „..., the result is unpredictable“. Z tohoto důvodu Acrobat zařazuje dokument obsahující tuto vlastnost do SigQ-1B.<br><br>Z hlediska bezpečnosti jsme přesvědčeni, že výsledkem může být jisté zkreslení obrázků, nikoliv však změna smyslu obsahu dokumentu a proto nepovažujeme tuto vlastnost za nebezpečnou. |

**Příklad 4.42 – Klíč /FL**

```
101 0 obj
<</FL 1.0/Type/ExtGState/FL 1.0>>
endobj
```

## 4.15 Image XObject alternate version

**Tabulka 4.39 – Image XObject alternate version**

| | |
|---|---|
| **SigQ kód:** | **2012** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 4.8.4 Image Dictionaries -> Alternate Images, str. 347 |
| **Klíč:** | **/Alternates XX X R** v objektu typu XObject a podtypu Image (/Subtype/Image/Type/XObject) |
| **Popis:** | Tag **/Alternates XX X R** v objektu typu XObject a podtypu Image (/Subtype/Image/Type/XObject), je volitelný a odkazuje na pole alternativních obrázků (alternate image dictionaries). Alternate images umožňují zahrnout do jediného PDF dokumentu více verzí obrázku, např. pro obrazovku a pro tisk, tyto obrázky se mohou lišit např. v rozlišení nebo v color space. |
| **Očekávaný projev:** | Zpráva s kódem 2012 |
| **Soubor:** | Složka alternateImage_2012_3XXX, soubory:<br>• /alternateImage_2012_4003_2013.pdf (DefaultForPrinting true)<br>• /alternateImage_2012_4003_2013_ DefaultForPrinting_false.pdf (DefaultForPrinting false) |
| **Zdroj:** | Nepodařilo se vytvořit alternativní obrázek pomocí Acrobatu, příklad musel být vytvořen manuální úpravou PDF dokumentu. V Acrobatu jsme do Mini.pdf dokumentu přidali 2 obrázky, pomocí „Nástroje-> Stránky -> Vložit ze souboru -> Soubory typu: Všechny soubory (*.*)“ . Do prvního obrázku (10 0 obj) jsme manuálně připsali tag (**/Alternates 17 0 R**), za objekt (10 0 obj) jsme přidali objekt (17 0 obj) a z něj vede odkaz do druhého obrázku (15 0 obj). |
| **Chování:** | V režimu náhledu podpisu jsou ve zprávě uvedeny kódy 2012, 4003 a 2013.<br><br>Na monitoru je na první stránce bílý čtverec, na druhé stránce žlutý čtverec. Onen bílý čtverec má však jako alternativu nastaven čtverec z druhé stránky (tedy žlutý čtverec). Druhá stránka není pro analýzu jevu podstatná, je použita pouze jako kontejner alternativního obrázku.<br><br>**/DefaultForPrinting true** (první z příkladů): |

| | |
|---|---|
| | Na monitoru není na první stránce čtverec zobrazen, protože je bílý, zatímco na tiskárně je na první stránce zobrazen žlutý čtverec.<br><br>**/DefaultForPrinting false** (druhý z příkladů):<br>Na monitoru je zobrazení stejné jako v předešlém případě, na tiskárně se však od předchozího případu liší – zobrazení je stejné jako na monitoru, tedy na první stránce bílý čtverec, na druhé stránce žlutý čtverec.<br><br>Důvodem odlišnosti obou případů je to, že ačkoliv v obou případech má bílý čtverec z první stránky nastavenu jako alternativu žlutý čtverec z druhé stránky, ve druhém případě odkazuje tag /Alternates na slovník, ve kterém je klíč DefaultFor Printing s hodnotou false (na rozdíl od prvního případu, který tam má true). Při tisku se proto alternativní obrázek nepoužije. |
| **Pozorování:** | Pouhá přítomnost klíče **`/Alternates`** (alternativních obrázků) vede k chybě 2012. |
| **Důvod:** | Alternativní verze obrázků nejsou dovoleny, protože při použití alternativních verzí se vzhled obrázků může lišit např. na monitoru a při tisku. |
| **Posouzení bezpečnosti:** | Z popisu chování je zřejmé, že použití **`/Alternates`** v kombinaci s **`/DefaultForPrinting true`** je nebezpečné, protože způsobuje změnu dokumentu v závislosti na tom, zda je dokument zobrazen na monitoru nebo je vytištěn. |

**Příklad 4.43 – Image XObject alternate version**

```
10 0 obj
<</BitsPerComponent 8/ColorSpace/DeviceRGB/Filter/DCTDecode/Height
19/Alternates 17 0 R/Length 661/Metadata 11 0
R/Name/X/Subtype/Image/Type/XObject/Width 19>>stream
...data obrázku...
endstream
endobj

17 0 obj
[<< /Image 15 0 R/DefaultForPrinting true>>]
endobj

15 0 obj
<</BitsPerComponent 8/ColorSpace/DeviceRGB/Filter/DCTDecode/Height
19/Length 665/Metadata 16 0 R
/Name/X/Subtype/Image/Type/XObject/Width 19>>
stream
...data jiného obrázku, který byl použitý jako alternativní...
endstream
endobj
```

## 4.16 Non-embedded fonts

**Ve výchozím nastavení se tato chyba nezobrazuje, ovšem po zapnutí zobrazení chyb 2013 a 2014 v registrech se zobrazuje alespoň jedna z těchto chyb u většiny vytvořených dokumentů.**

**Tabulka 4.40 – Non-embedded fonts**

| | |
|---|---|
| **SigQ kód:** | **2013** nevyhovuje PDF/SigQ (pokud je potlačení chyby v registrech vypnuto, ve výchozím nastavení může být PDF/SigQ-1A nebo PDF/SigQ-1B ) |
| **Odkaz:** | [1] kapitola 5.8 Embedded Font Programs, strana 465, 743<br>[4] 5.4.3.7 Controlling LegalPDF and PDF Signature Report Warnings, str. 136 |
| **Klíč:** | ve font descriptoru (**`/Type/FontDescriptor`**) není **`/FontFile`**, **`/FontFile2`** ani **`/FontFile3`** |
| **Popis:** | Pokud ve font descriptoru (**/Type/FontDescriptor**) není **/FontFile**, **/FontFile2** ani **/FontFile3**, tak to znamená, že tam není vloženo písmo, při zobrazení se použije písmo ze systému, pokud písmo v systému není, tak se použije nějaké podobné písmo. |
| **Očekávaný projev:** | Zpráva s kódem 2013 |
| **Soubor:** | `non-embedded_fonts_2013/non-embedded_font_Vladimit_Script.pdf`<br><br>**protipříklad:**<br>`non-embedded_font_Backadder_noError.pdf` |
| **Postup:** | Napsat dokument ve Wordu, použít nějaké nezákladní písmo (Vladimir Script), záložka Acrobat -> Předvolby -> nastavení převodu -> tlačítko další nastavení -> písma -> Vybrat použitý font a přesunout ho do listu „Nikdy nevkládat“ -> na závěr převést do PDF pomocí možnosti Vytvořit PDF na záložce Acrobat.<br><br>**Zobrazení chybové hlášky 2013 je ve výchozím nastavení registrů potlačeno**. Aby se chyba projevila, je nutné **v registrech** pro klíč „**HKEY_CURRENT_USER\Software\Adobe\Adobe Acrobat\10.0\Security\cDigSig**“ přidat položku s názvem „**bEnNonEmbFontLegPDFWarn**“ a její hodnotu nastavit na „**1**“. Toto nastavení funguje na Win7 64bit, Acrobat verze 10.1.8<br><br>Po restartu Acrobatu (nebo systému) se při pokusu o podpis se projeví výše uvedená chyba 2013, která se ve výchozím nastavení registrů nezobrazuje. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 2013. |
| **Pozorování:** | Acrobat při výše uvedeném nastavení registrů při pokusu o podpis hlásí: *„V tomto dokumentu je bohatý obsah, který nelze spolehlivě potlačit pomocí zobrazení náhledu dokumentu. Klepněte na Pokračovat, chcete-li přesto zobrazit náhled tohoto dokumentu.“* Pokračujeme-li, hlásí Acrobat kód 2013 (Vzhled text se může změnit bez upozornění). |
| **Důvod** | Dokument obsahuje písma, která nejsou vložená. Když se dokument otevře v systému, kde nejsou tato písma, tak je Acrobat nahradí jinými a to může způsobit změnu vzhledu. Nebo by někdo mohl tato písma v systému nahradit upravenou verzí. |
| **Posouzení bezpečnosti:** | Viz výše kolonka „Důvod“. |

**Příklad 4.44 – Non-embedded fonts**

```
23 0 obj
<</Ascent 896/CapHeight 674/Descent -386/Flags 6/FontBBox[-205 -386
1426 896]/FontFamily(Blackadder
ITC)/FontName/BlackadderITC/FontStretch/Normal/FontWeight
400/ItalicAngle 0/StemV 84/Type/FontDescriptor/XHeight 303>>
endobj
```

**Příklad 4.45 – Non-embedded fonts, protipříklad, který chybu nevyvolává**

```
32 0 obj
<</Ascent 896/CIDSet 33 0 R/CapHeight 674/Descent -386/Flags
6/FontBBox[-205 -386 1426 896]/FontFamily(Blackadder
ITC)/FontFile2 34 0 R/FontName/IWWDYX+BlackadderITC/FontStretch/Norm
al/FontWeight 400/ItalicAngle 0/StemV 84/Type/FontDescriptor/XHeight
303>>
endobj
```

## 4.17 TrueType and TrueType based OpenType

**Ve výchozím nastavení se tato chyba nezobrazuje, ovšem po zapnutí zobrazení této chyby v registrech se zobrazuje u většiny vytvořených dokumentů.**

**Tabulka 4.41 – TrueType and TrueType based OpenType**

| | |
|---|---|
| **SigQ kód:** | **2014** nevyhovuje PDF/SigQ (pokud je potlačení chyby v registrech vypnuto, ve výchozím nastavení může být PDF/SigQ-1A nebo PDF/SigQ-1B ) |
| **Odkaz:** | [1] kapitola 5.5.2 TrueType Fonts, strana 418, příklad 5.8 str. 419<br>[4] 5.4.3.7 Controlling LegalPDF and PDF Signature Report Warnings, str. 136 |
| **Klíč:** | **`/Subtype/TrueType`** |
| **Popis:** | Klíč **`/Subtype`** s hodnotou **`/TrueType`** označuje písmo typu TrueType. **OpenType, které je založené na TrueType** má stejný subtype viz přiklad ve složce „TrueType_font_2014/OpenType based TrueType“ písmo z http://www.ceskefonty.cz/ceske-fonty/promocyja. |
| **Očekávaný projev:** | |
| **Soubor:** | `TrueType_font_2014/Testovaci text03_2014.pdf` |
| **Postup:** | Napsat dokument ve Wordu, použít nějaké TrueType písmo (např. DejaVu Sans Light), záložka Acrobat -> Předvolby -> nastavení převodu -> tlačítko další nastavení -> písma -> Vybrat použitý font a přesunout ho do listu „Vždy vkládat“ -> na závěr převést do PDF pomocí možnosti Vytvořit PDF na záložce Acrobat.<br><br>**Zobrazení chybové hlášky 2013 je ve výchozím nastavení registrů potlačeno**. Aby se chyba projevila, je nutné **v registrech** pro klíč „**HKEY_CURRENT_USER\Software\Adobe\Adobe Acrobat\10.0\Security\cDigSig**“ přidat položku s názvem „**bTrueTypeFontPDFSigQWarn**“ a její hodnotu nastavit na „**1**“. Toto nastavení funguje na Win7 64bit, Acrobat verze 10.1.8<br><br>Po restartu Acrobatu (nebo systému) se při pokusu o podpis projeví výše uvedená chyba 2014, která se ve výchozím nastavení registrů nezobrazuje. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 2014. |

| | |
|---|---|
| **Pozorování:** | Acrobat při výše uvedeném nastavení registrů při pokusu o podpis hlásí: *„V tomto dokumentu je bohatý obsah, který nelze spolehlivě potlačit pomocí zobrazení náhledu dokumentu. Klepněte na Pokračovat, chcete-li přesto zobrazit náhled tohoto dokumentu.“* Pokračujeme-li, hlásí Acrobat kód 2014 (Vzhled text se může změnit bez upozornění). |
| **Důvod:** | Dokument používá písma TrueType nebo OpenType založená na TrueType. Tato písma nejsou povolená, protože jsou to programy a mohou měnit vzhled na základě externích proměnných. |
| **Posouzení bezpečnosti:** | Viz výše kolonka „Důvod“. |

**Příklad 4.46 – TrueType and TrueType based OpenType**

```
18 0 obj
<<
/BaseFont /JCATYV+DejaVuSans-ExtraLight
/Encoding /WinAnsiEncoding
/FirstChar 32
/FontDescriptor 17 0 R
/LastChar 120
/Subtype /TrueType
/ToUnicode 14 0 R
/Type /Font
/Widths [ 318 0 0 0 0 0 0 0 0 0 0 0 0 318 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 611 0 0 0 0 0 0 0
0 0 0 0 0 613 0 550 0 615 0 0 0 278 0 0 0 0 0 612 0 0 0 521 392 0
592 0 592 ]
>>
endobj
```

## 4.18 External XObjects

**Nepodařilo se v Acrobatu reprodukovat.**

**Tabulka 4.42 – External XObjects**

| | |
|---|---|
| **SigQ kód:** | **předpokládaný kód 3000** nevyhovuje PDF/SigQ |
| **Odkaz:** | **Možná** jde o [1] kapitola 4.9.3 Reference XObjects, strana 361 |
| **Klíč:** | **Možná** položka **`/Ref`** ve form dictionary |
| **Popis:** | Reference XObject v dokumentu, který odkazuje (containing document), je form XObject, který obsahuje volitelnou položku **`/Ref`** ve svém form dictionary. Tento form XObject slouží jako proxy, která může být zobrazena nebo vytištěna místo importovaného obsahu. Proxy může obsahovat importovaný obrázek v nízkém rozlišení, textový popis, který se vztahuje k obrázku, šedý box, který se má zobrazit místo obrázku, nebo nějaký podobný nahrazující prvek. Aplikace, která nerozpozná položku **Ref**, jednoduše zobrazí nebo vytiskne proxy jako obyčejný XObject. |
| **Očekávaný projev:** | Zpráva s kódem 3000 |
| **Důvod:** | Dokument odkazuje na obrázky, které nejsou v PDF souboru. Externí XObjects nejsou dovoleny. |
| **Posouzení** | Uživatel by měl být důrazně varován před podpisem dokumentu. Aplikace by ho mě- |

| | |
|---|---|
| **bezpečnosti:** | la upozornit, že dokument se může měnit, a že podepsání dokumentu není doporučeno. Zobrazení obsahu dokumentu, kde jsou některé obrázky externí, je z hlediska visuální integrity nebezpečné. Nevíme, zda se použije proxy nebo zda bude obsah importován z externího zdroje, který se může měnit. |

## 4.19 OPI Alternate images

**Tabulka 4.43 – OPI Alternate images**

| | |
|---|---|
| **SigQ kód:** | **3003** PDF/SigQ-1B |
| **Odkaz:** | [1] kapitola 10.10.6 Open Prepress Interface (OPI), strana 978 a 342 (tag /OPI v image dictionary)<br>[1] kapitola 3.10.2 File Specification Dictionaries, strana 182, tab. 3.41 |
| **Klíč:** | **`/OPI`** |
| **Popis:** | Tag **`/OPI`** s referencí na **OPI version dictionary** ([1] tab. 10.54 str. 979) v image dictionary způsobuje chybu 3003. |
| **Očekávaný projev:** | Zpráva s kódem 3001 a 3003 |
| **Soubor:** | `extAlternateImageOPI_3003_3002_2012_4003_4000_2013.pdf` |
| **Zdroj:** | Tento soubor byl vytvořen přímou editací obsahu PDF dokumentu. Byl přidán tag /OPI do objektu 10 0 obj a byly vytvořeny objekty 19 0 obj a 20 0 obj. |
| **Chování:** | Chybový kód 3001 a 3003 mají ve specifikaci [3] identický popis, proto jsme očekávali jeden z nich nebo oba dva.<br>V režimu náhledu podpisu je ve zprávě uveden ve skutečnosti kód 3003, což je v souladu s očekáváním. |
| **Pozorování:** | Soubory s OPI dictionary vyvolávají chybu 3003. |
| **Důvod:** | Před tiskem je dokument zpracován OPI serverem, který nahradí proxies (obrázky v nízkém rozlišení) obrázky v plném rozlišení. |
| **Posouzení bezpečnosti:** | Nahrazení obrázku OPI serverem znamená zobrazení obsahu dokumentu, jehož část je nahrazena externím obsahem, což je z hlediska visuální integrity nebezpečné. |

### Příklad 4.47 – OPI Alternate images

```
10 0 obj
<</BitsPerComponent 8/ColorSpace/DeviceRGB/Filter/DCTDecode
/Height 19/Alternates 17 0 R/Length 661
/Metadata 11 0 R/Name/X/Subtype/Image/Type/XObject
/Width 19/OPI<</2.0 19 0 R>>>>stream...

19 0 obj
<</Version 2.0/F 20 0 R>>
endobj

20 0 obj
<</Type/FileSpec/F(smallImg2.jpg)>>
endobj
```

## 4.20 External streams

**Tabulka 4.44 – External streams**

| | |
|---|---|
| **SigQ kód:** | **3002** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 4.8.4 Image Dictionaries -> Alternate Images, str. 347<br>[1] kapitola 3.10.2 File Specification Dictionaries, str. 182<br>[1] kapitola 3.10.4 URL Specification, str. 188 |
| **Klíč:** | **/Alternates XY 0 R**<br><br>XY 0 obj<br>[<< **/Image AB 0 R**/DefaultForPrinting true>>]<br>endobj<br><br>AB 0 obj<br><< /Type **/XObject**/Subtype **/Image**/Width XXXX/Height YYYY/ColorSpace /DeviceRGB/BitsPerComponent 8**/Length 0/F**<br><< **/FS /URL**<br>**/F** (http://www.myserver.mycorp.com/images/image.jpg)<br>>><br>/FFilter /DCTDecode>> stream<br>endstream |
| **Popis:** | Položka **/Alternates** odkazuje na pole, které odkazuje na pole, které v sobě má alternate image dictionary, které odkazují na objekty typu **/XObject** a podtypu **/Image**, který načítá obsah streamu z externího zdroje, například z URL adresy (**/FS/URL**), další možnosti specifikace externích souborů popsány v [1] tabulka 3.41 (klíče: **/F**, **/UF**, **/DOS**, **/Mac**, **/Unix**) |
| **Očekávaný projev:** | Zpráva s kódem 3002 |
| **Soubor:** | alternate-Image_2012_3XXX/extAlternateImage_3002_2012_4003_2013.pdf |
| **Zdroj:** | Nepodařilo se replikovat v Acrobatu, soubor manuálně upraven, podle příkladu 4.29 na straně 348. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 3002. |
| **Pozorování:** | Soubory s externími streamy vyvolávají chybu 3002. |
| **Důvod:** | Obsah externích streamů se může měnit. |
| **Posouzení bezpečnosti:** | Obsah streamu se čte z externího zdroje, není tedy součástí podepsaného dokumentu, což je z hlediska visuální integrity nebezpečné. |

### Příklad 4.48 – External streams

```
10 0 obj
<</BitsPerComponent 8/ColorSpace/DeviceRGB/Filter/DCTDecode/Height
19/Alternates 17 0 R/Length 661/Metadata 11 0
R/Name/X/Subtype/Image/Type/XObject/Width 19>>stream
...binarní data obrázku...
endstream
endobj

17 0 obj
[<< /Image 18 0 R/DefaultForPrinting true>>]
endobj
```

```
18 0 obj
<< /Type /XObject/Subtype /Image/Width 1000/Height 2000/ColorSpace
/DeviceRGB/BitsPerComponent 8/Length 0/F << /FS /URL
/F (http://www.myserver.mycorp.com/images/exttest.jpg)
>>
/FFilter /DCTDecode>>
stream
endstream
endobj
```

## 4.21 Unrecognized PDF content

### Posouzení bezpečnosti: Unrecognized PDF content

Nerozpoznaný nebo neznámý obsah se může v závislosti na prohlížeči zobrazit jinak. V novějších verzích se může zobrazit, ve starších verzích může být ignorován nebo chybně vykreslen. Uživatel by měl být varován před podpisem dokumentu, který obsahuje nerozpoznaný obsah.

### 4.21.1 Odhalení chyb typu „Unrecognized PDF content"

Narazili jsme na více důvodů, které vedou k chybě 4000. K odhalení chyb tohoto typu, by se muselo zkontrolovat:

- **A.** Zda všechny hodnoty všech klíčů ve všech objektech mají správný typ, plus jaká jsou pravidla pro nedefinované klíče ve všech objektech, položka type v tabulkách „**Entries in XXXX dictionary**". Tabulka typů používaných v PDF souborech tab. 3.31 v kapitole 3.8 Common Data Structures na straně 155, obsahuje odkazy s podrobnostmi o každém typu.
- **B.** Zda všechny objekty mají všechny povinné (**Required**) atributy. Povinné atributy jsou definovány ve specifikaci [1] ve všech tabulkách, které popisují různé slovníky (dictionary), takové tabulky se jmenují „**Entries in XXYY dictionary**", kde XXYY je typ daného slovníku. V těchto tabulkách je ve sloupci „**Value**" na začátku v závorce uvedeno, zda je hodnota „**Required**" nebo „**Optional**", případně za jakých podmínek je „**Required**".

### 4.21.2 Příklady chyb „Unrecognized PDF content"

#### A. Špatný typ hodnoty klíče

**Tabulka 4.45 – Chyba „Unrecognized PDF content" – Špatný typ hodnoty klíče 1**

| | |
|---|---|
| **SigQ kód:** | **4000** nevyhovuje PDF/SigQ |
| **Odkaz:** | Klíč v příkladu je popsán v [1] kapitola 3.9.1 Type 0 (Sampled) Functions str. 170.<br>Všeobecně ve všech tabulkách musí k určitému klíči/key sedět jeho typ/type.<br>[1] kapitola 3.8 Common Data Structures, tab. 3.31, str. 155. |
| **Klíč:** | **Žádný konkrétní klíč.** |
| **Popis:** | Hodnota klíče/key neodpovídá typu/type. Viz. příklad: Měl by tam klíč /Size<br>s hodnotou typu Array, ale místo toho má tag /Size hodnotu typu Integer. |
| **Očekávaný projev:** | Zpráva s kódem 4000 |
| **Soubor:** | `Unrecognized_PDF_content_4000/pokus_TR2_4000_4003_2013.pd` |

| | |
|---|---|
| | f<br><br>**Protipříklad:**<br>Unrecognized_PDF_content_4000/pokus_TR2_4003_2013.pdf |
| **Zdroj:** | Příklad získán od zadavatele. Protipříklad byl opraven přímou editací PDF souboru. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 4000. |
| **Pozorování:** | Soubory, ve kterých je nějaká hodnota, jejichž typ neodpovídá typu, který by hodnota pro daný klíč podle dokumentace měla mít, vyvolávají chybu 4000. |
| **Důvod:** | Nerozpoznaný obsah muže být vlastností budoucích verzí Acrobatu, soubor by se mohl v různých verzích Acrobatu zobrazit různě. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny příklady chyby 4000 na začátku sekce 4.21 nadpis: Posouzení bezpečnosti: Unrecognized PDF content. |

**Příklad 4.49 – Chyba „Unrecognized PDF content“ – Špatný typ hodnoty klíče 1**

```
30 0 obj
<</Type /ExtGState/SA true/TR 31 0 R>>
endobj

31 0 obj
<</FunctionType 0/Domain [ 0.0 1.0 ]/Range [ 0.0 1.0 ]
/Size 2/BitsPerSample 8/Length 7/Filter /ASCIIHexDecode>>
stream
01 00 >
endstream
endobj
```

**Příklad 4.50 – Oprava Chyby „Unrecognized PDF content“ – Špatný typ hodnoty klíče 1**

```
31 0 obj
<</FunctionType 0/Domain [ 0.0 1.0 ]/Range [ 0.0 1.0 ]
/Size [2]/BitsPerSample 8/Length 7/Filter /ASCIIHexDecode>>
stream
01 00 >
endstream
endobj
```

**Tabulka 4.46 – Chyba „Unrecognized PDF content“ – Špatný typ hodnoty klíče 2**

| | |
|---|---|
| **SigQ kód:** | **4000** nevyhovuje PDF/SigQ |
| **Odkaz:** | Ve Specifikaci formátu PDF [1] kapitola 10.2.1Document Information Dictionary, strana 843. |
| **Popis:** | Objekt „16 0 obj“ je typu document information dictionary, může mít položky zmíněně v tabulce 10.2 na str. 844, pokud jsou tam položky, které v tabulce nejsou, tak **hodnota asociovaná s klíčem, který není specifikovaný v tabulce 10.2, musí být typu „text string“.** Viz předposlední odstavec na straně 843, (*The value associated with any key not specifically mentioned in Table 10.2 must be a text string.)* |
| **Soubor:** | Unrecognized_PDF_content_4000/03_PDFA_err_4000.pdf<br><br>**Protipříklad/Oprava:**<br>Unrecognized_PDF_content_4000/03_PDFA_err_4000_repaired.p |

| | |
|---|---|
| | df |
| **Zdroj:** | Příklad získán od zadavatele. Protipříklad byl opraven přímou editací PDF souboru. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny příklady chyby 4000 na začátku sekce 4.21 nadpis: Posouzení bezpečnosti: Unrecognized PDF content. |

**Příklad 4.51 – Chyba „Unrecognized PDF content“ – Špatný typ hodnoty klíče 2**

```
16 0 obj
</Title <
<FEFF004E006F007600FD0020006F0062006A0065006B00740020002D00200054006
500780074006F007600FD00200064006F006B0075006D0065006E0074002E0074007
800740020002D00200050006F007A006E00E1006D006B006F007600FD00200062006
C006F006B>
/Producer (Amyuni PDF Converter version 4.0.0.9)
/Creator
<FEFF0053006F006600740077006100720065003600300032002000500072006900
E00740032005000440046002000380002E0030002E00300039002E003100300032003
3>
/CreationDate (D:20091215094718+02'00')
/ModDate (D:20091215094720+02'00')
/obj <<
/Type /Pages
/Count 0
/Kids []
 >>
/endobj 7
/OutputIntent /OutputIntent
/S /GTS_PDFA1
/OutputConditionIdentifier (Custom)
/Info (Unknown)
/Filter /FlateDecode
/N 3
>>
endobj
```

**Příklad 4.52 – Oprava Chyby „Unrecognized PDF content“ – Špatný typ hodnoty klíče 2**

```
16 0 obj
<<
/Title
<FEFF004E006F007600FD0020006F0062006A0065006B00740020002D00200054006
500780074006F007600FD00200064006F006B0075006D0065006E0074002E0074007
800740020002D00200050006F007A006E00E1006D006B006F007600FD00200062006
C006F006B>
/Producer (Amyuni PDF Converter version 4.0.0.9)
/Creator
<FEFF0053006F006600740077006100720065003600300032002000500072006900
E00740032005000440046002000380002E0030002E00300039002E003100300032003
3>
/CreationDate (D:20091215094718+02'00')
```

```
/ModDate (D:20091215094720+02'00')
/OutputIntent (OutputIntent)
/S (GTS_PDFA1)
/OutputConditionIdentifier (Custom)
/Info (Unknown)
/Filter (FlateDecode)
/N (3)
>>
endobj
```

| Tabulka 4.47 – Chyba „Unrecognized PDF content“ – Špatný typ hodnoty klíče 3 | |
|---|---|
| **SigQ kód:** | **4000** nevyhovuje PDF/SigQ |
| **Popis:** | Nesprávný typ hodnoty asociovaný s klíčem, 23 0 obj, 3.7.2 Resource Dictionaries, str. 153, v tabulce 3.30 není tag /Encoding uveden, takže by v resource dictionary neměl být, po jeho odstranění je dokument bez chyby. |
| **Soubor:** | Unrecognized_PDF_content_4000/04_PDFA_err_4000.pdf<br><br>**Protipříklad/Oprava:**<br>Unrecognized_PDF_content_4000/04_PDFA_err_4000_repaired.pdf |
| **Zdroj:** | Příklad získán od zadavatele. Protipříklad byl opraven přímou editací PDF souboru. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny příklady chyby 4000 na začátku sekce 4.21 nadpis: Posouzení bezpečnosti: Unrecognized PDF content. |

**Příklad 4.53 – Chyba „Unrecognized PDF content“ – Špatný typ hodnoty klíče 3**

```
23 0 obj
<</Encoding/WinAnsiEncoding
/Font<</F9 15 0 R/ZaDb 30 0 R/Helv 29 0 R>>>>
endobj
```

### B. Chybějící povinné atributy

| Tabulka 4.48 – Chyba „Unrecognized PDF content“ – Chybějící povinné atributy | |
|---|---|
| **SigQ kód:** | **4000** nevyhovuje PDF/SigQ |
| **Popis:** | Chybí klíče **/Subtype** a **/Type** v **6 0 obj,** tyto klíče jsou u Metadat povinné/Required viz tab. 10.3 ve Specifikaci formátu PDF Additional entries in a metadata stream dictionary na straně 846. |
| **Protipříklad /Oprava:** | 6 0 obj<br><</Type/Metadata/Subtype/XML/Length 7 0 R>>stream<br>...data... |
| **Soubor:** | Unrecognized_PDF_content_4000/01_PDFA_err_4000.pdf<br><br>**Protipříklad/Oprava:**<br>Unrecognized_PDF_content_4000/01_PDFA_err_4000_repaired.pdf |
| **Zdroj:** | Příklad získán od zadavatele. Protipříklad byl opraven přímou editací PDF souboru. |
| **Posouzení bezpečnosti:** | Uvedeno společně pro všechny příklady chyby 4000 na začátku sekce 4.21 nadpis: Posouzení bezpečnosti: Unrecognized PDF content. |

**Příklad 4.54 – Chyba „Unrecognized PDF content“ – Chybějící povinné atributy**

```
11 0 obj
<<
/Type /Catalog
/Pages 31 0 R
/AcroForm 1 0 R
/PageMode /UseNone
/Lang (1)
/OutputIntents 5 0 R
/Metadata 6 0 R
/MarkInfo <</Marked true>>
/StructTreeRoot 8 0 R
>>
endobj

6 0 obj
<<
/Length 7 0 R
>>
stream
<?xpacket begin='' id='W5M0MpCehiHzreSzNTczkc9d'?>
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
xmlns:iX='http://ns.adobe.com/iX/1.0/'>
<rdf:Description about='' xmlns='http://purl.org/dc/elements/1.1/'
xmlns:dc='http://purl.org/dc/elements/1.1/'>
<dc:title><rdf:Alt><rdf:li xml:lang="x-
default"></rdf:li></rdf:Alt></dc:title>
<dc:creator><rdf:Seq><rdf:li></rdf:li></rdf:Seq></dc:creator>
</rdf:Description>
<rdf:Description about='' xmlns='http://ns.adobe.com/pdf/1.3/'
xmlns:pdf='http://ns.adobe.com/pdf/1.3/'>
<pdf:Producer>Print2PDF 7.0</pdf:Producer>
</rdf:Description>
<rdf:Description about='' xmlns='http://ns.adobe.com/xap/1.0/'
xmlns:xap='http://ns.adobe.com/xap/1.0/'>
<xap:CreatorTool>Software602 Print2PDF 8.0.09.0227</xap:CreatorTool>
<xap:MetadataDate>2009-04-15T07:33:17+02:00</xap:MetadataDate>
<xap:CreateDate>2009-04-15T08:33:04+02:00</xap:CreateDate>
<xap:ModifyDate>2009-04-15T07:33:17+02:00</xap:ModifyDate>
</rdf:Description>
<rdf:Description rdf:about=''
xmlns:pdfaid='http://www.aiim.org/pdfa/ns/id/'>
<pdfaid:part>1</pdfaid:part>
<pdfaid:conformance>A</pdfaid:conformance>
</rdf:Description>
</rdf:RDF>
<?xpacket end='r'?>
```

```
endstream
endobj
```

## 4.22 Unrecognized drawing operator

**Tabulka 4.49 – Unrecognized drawing operator**

| | |
|---|---|
| **SigQ kód:** | **4001** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola str. 196 |
| **Klíč:** | **V Content streamu je jiný operátor než ten zmíněný v tabulce 4.1 na str. 196.** |
| **Popis:** | V Content streamu je jiný operátor než ten zmíněný v tabulce 4.1 na str. 196, může být závislé na verzi Acrobatu, v novější verzi Acrobatu může tento operátor být podporován, ale v současné verzi se zobrazí chyba. |
| **Očekávaný projev:** | Zpráva s kódem 4001 |
| **Soubor:** | `Uncategorized_warnings_4XXX/ 01_err_4001_decompressed.pdf`<br><br>**Protipříklad:**<br>`01_err_4001_OK_repaired_decompressed.pdf` |
| **Zdroj:** | Nepodařilo se replikovat v Acrobatu, soubor manuálně upraven, nahrazením operátoru „Tm“, neexistujícím operátorem „Tn“. |
| **Chování:** | V režimu náhledu podpisu je ve zprávě uveden kód 4001. |
| **Pozorování:** | Soubory s neexistujícími kreslícími operátory vyvolávají chybu 4001. |
| **Důvod:** | Chování těchto operátorů není v současné verzi Acrobatu definováno, soubor by se na různých verzích prohlížečů formátu PDF mohl zobrazit různě. |
| **Posouzení bezpečnosti:** | Uživatel by měl být varován před podpisem dokumentu, který obsahuje neznámý operátor, protože neznámý kreslící operátor se může v závislosti na použitém PDF prohlížeči zobrazit pokaždé jinak. V novějších verzích se může zobrazit očekávaný obsah, ve starších verzích může být operátor ignorován nebo může dojít k jinému nepředvídatelnému chování při zobrazení dokumentu. |

### Příklad 4.55 – Unrecognized drawing operator

```
10 0 obj
<<
/Length 6935
>>
stream
1 w
/F9 9.00 Tf
BT
1 0 0 1 284.15 806.40 Tm
-0.020 Tc <0004000500060007000 8> Tj
ET
/F9 9.00 Tf
BT
1 0 0 1 21.25 797.40 Tn
<00090007000A00030007000B000C000B000D000E000F000700100011001200060001
3000B0014000B00150007001100060016000D0017000E> Tj
ET
```

## 4.23 Malformed drawing instruction

Tabulka všech operátorů s odkazy na podrobnosti o každém z nich je ve specifikaci v kapitole Appendix A na straně 985.

### Posouzení bezpečnosti: Malformed drawing instruction

Uživatel by měl být varován před podpisem dokumentu, který obsahuje kreslící operátor s chybnou syntaxí. Chybná syntaxe může způsobit nekonzistentní zobrazení dokumentu v různých PDF prohlížečích.

### 4.23.1 Příklady chyb „Malformed drawing instruction“

**Tabulka 4.50 – Chyba „Malformed drawing instruction“ 1**

| | |
|---|---|
| **SigQ kód:** | **4002** nevyhovuje PDF/SigQ |
| **Popis:** | chyba je v 37 0 obj, offset 51209, řádka 672 |
| **Řešení:** | Odebrání operátorů **/Tx a BMC** problém vyřeší, nebo musíme přidat EMC. BMC zahajuje marked-content sequence která musí být ukončena tagem EMC.<br><br>**Obecně:**<br>Musíme zkontrolovat, zda je správné párování a zanoření tagů BMC, BDC, EMC, BT a ET. Viz Specifikace formátu PDF [1] str. 851 |
| **Soubor:** | `Unrecognized_PDF_content_4000/01_err_4002_decompressed.pdf`<br>`Unrecognized_PDF_content_4000/01_err_4002_repaired_decompressed.pdf` |
| **Posouzení bezpečnosti:** | Uvedeno společně pro příklady chyby 4002 na začátku sekce 4.21 nadpis: Posouzení bezpečnosti: Malformed drawing instruction. |

**Příklad 4.56 – Chyba „Malformed drawing instruction“ 1**

```
37 0 obj
<<
/Type /XObject
/Subtype /Form
/BBox [ 0 0 30 31 ]
/Resources << /ProcSet [/PDF /Text /ImageB /ImageC] /Font << /F1 36
0 R >> >>
/Length 4992
>>
stream
/Tx BMC q
30 0 0 31 0 0 cm
BI
/W   128 /H   128 /BPC 8 /CS /RGB /F /DCT
ID ... data...
```

**Tabulka 4.51 – Chyba „Malformed drawing instruction“ 2**

| | |
|---|---|
| **SigQ kód:** | **4002** nevyhovuje PDF/SigQ |
| **Odkaz:** | [1] kapitola 4.1 figure 4.1 na straně 197 a tab. 4.1 na str. 196. |
| **Popis:** | Chyby jsou v 4 0 obj. |

| | |
|---|---|
| | První výskyt chyby je operátor **g** na 5 místě ve streamu (**1 g**), před ním je operátor **re** (**5.76 835.92 583.92 -830.64 re**), ovšem podle figure 4.1 by po operátoru **re** měl následovat Path construction operator, všechny chyby byly způsobeny tak, že operátory **g** nebo **rg** následovaly po operátorech **re**, **h** nebo dalších z kategorie **Path construction**. |
| **Řešení:** | Zkontrolovat dodržovaní pravidel znázorněných a popsaných v kapitole 4.1 figure 4.1 na straně 197 a tab. 4.1 na str. 196. |
| **Soubor:** | Unrecognized_PDF_content_4000/02_err_4002_decompressed.pdf Unrecognized_PDF_content_4000/02_err_4002_repaired_decompressed.pdf |
| **Posouzení bezpečnosti:** | Uvedeno společně pro příklady chyby 4002 na začátku sekce 4.21 nadpis: Posouzení bezpečnosti: Malformed drawing instruction. |

**Příklad 4.57 – Chyba „Malformed drawing instruction“ 2**

```
4 0 obj
<<
/Length 74707
>>
stream
q
1 0 0 1 0 0 cm
n
5.76 835.92 583.92 -830.64 re
1 g
f*
n
17.04 833.04 114.24 -69.36 re
f*
n
17.04 745.92 114.24 -589.92 re
f*
n
```

## 4.24 Nerozpoznaný obsah 4003

**Tabulka 4.52 – Nerozpoznaný obsah 4003**

| | |
|---|---|
| **SigQ kód:** | **4003** nevyhovuje PDF/SigQ |
| **Odkaz:** | Chyba 4003 není ve specifikaci formátu PDF [1] zmíněna.<br>Položka **/Prev** je zdokumentována ve Specifikaci [1] na straně 97, tab. 3.13<br>[1] kap. 3.4.3 Cross-Reference Table, str. 93 |
| **Klíč:** | **/Prev položky v Cross-Reference Table (xref) odkazují na špatné offsety.** |
| **Popis:** | Offsety referencí objektů v PDF souboru v tabulce **Cross-Reference Table (xref)** nebo v položce **/Prev** neodpovídají skutečné pozici/offsetu začátku objektů v PDF souboru.<br><br>Offset začátku objektu se počítá jako počet bajtů od začátku PDF souboru do místa před začátkem čísla objektu. Například offset objektu **10 0** bude počet bajtů od začátku souboru do místa před bajty, které lze v ANSII zobrazit jako řetězec „**10 0 obj**“, kterým objekt **10 0** začíná. Tedy po konci předchozího objektu „**endobj**+ znaky odřádkování CR LF“. |

| | |
|---|---|
| **Očekávaný projev:** | Zpráva s kódem 4003 |
| **Soubor:** | `Unrecognized_PDF_content_4003/ 01_err_4003_decompressed.pdf` |
| **Zdroj:** | Nepodařilo se replikovat v Acrobatu, soubory manuálně upraveny. |
| **Chování:** | U některých souborů se pouze v režimu náhledu podpisu zobrazí chyba 4003, některé soubory nejdou vůbec otevřít, Acrobat hlásí, že soubor je poškozen. |
| **Pozorování:** | Soubory nesedícími indexy vyvolávají chybu 4003, případně je nelze otevřít. |
| **Důvod:** | Offsety nesedí, pasování může dávat různé výsledky v různých programech. |
| **Posouzení bezpečnosti:** | Uživatel by měl být varován před podpisem dokumentu, jehož přímé odkazy (tj. odkazy pomocí offsetů) nejsou platné. Chyba v přímých odkazech může způsobit například to, že soubor nebude vůbec otevřen, nebo se při jeho otevření objeví chybové okno. Chování a zobrazení takového souboru se může v různých PDF prohlížečích lišit. |

# 5 Nástroj pro detekci prvků ovlivňujících vizuální integritu

V této kapitole je popsána implementace demonstrační aplikace, která byla pojmenována PDF VIA (PDF Visual Integrity Analysis) pro detekci prvků, které by mohly ovlivňovat vizuální integritu. Tím se ověřila možnost převedení prvků zjištěných při analýze do praxe.

## 5.1 Použité technologie a architektura

Prototyp aplikace je postaven na technologii Java SE, jde o robustní a ověřenou multiplatformní technologii, která se hodí pro vývoj prototypu. Vývoj a testování probíhalo na platformě MS Windows, ovšem díky použité technologii by neměl být problém aplikaci upravit nebo bez úprav provozovat na ostatních platformách, které mají k dispozici interpret Javy.

Pro snadný návrh uživatelského rozhraní byl použitý vestavěný nástroj pro tvorbu grafického uživatelského rozhraní ve vývojovém prostředí NetBeans (tvz. project Matisse), který patří mezi nejlepší volně dostupné nástroje na tvorbu grafického uživatelského rozhraní.

Pro rozbalení komprimovaných streamů aplikace využívá Apache PDFBox[6], to je open source Java knihovna, která umožňuje práci s PDF dokumenty. Kromě asistence při rozbalování streamů, je ovšem celé načítání PDF souborů prováděno přímo aplikací. Pokud bylo nutné zbavit se závislosti na knihovně PDFBox, tak by stačilo implementovat vlastní modul na rozbalování streamů.

Aplikace je rozdělená na tři hlavní komponenty: model, view a controller. Model poskytuje vrstvu pro práci s PDF soubory, jejich parsování a načítání jednotlivých objektů do paměti. Controller prochází strom objektů v PDF dokumentu a vyhledává podezřelé konstrukce. View pouze zobrazí nalezené problémy.

## 5.2 Funkcionalita

Prototyp implementované demonstrační aplikace poskytuje tuto funkcionalitu:

- Umožňuje uživateli zvolit PDF soubor, který má být analyzován, přetažením souboru na formulář aplikace nebo vybráním v dialogu.
- Načte cross-reference tabulku nebo cross-reference stream.
- Prochází poslední verzi zvoleného souboru od kořenového objektu (tj. položka `/Root` v trailer dictionary), postupně jsou procházeny všechny slovníky a pole, které jsou použité v poslední verzi souboru, a vyhledává v nich nebezpečné konstrukce.
- Výsledky jsou potom představeny uživateli v přehledné tabulce i s umístěním problematické konstrukce v souboru. Umístění je zapsáno jako cesta, kterou se lze k problémovému prvku dostat od kořenového objektu daného dokumentu.

[6] https://pdfbox.apache.org/

Zatím je podporováno vyhledávání následujících problémových konstrukcí:

- Trigger events
- JavaScriptů
- Widgetů
- FreeTextových anotací
- Multimedií
- Pop-up anotací
- Položek nabídek/Named actions
- XFA dokumentů
- Nepovolených typů akcí
- Vlastní nastavení viditelnosti vrstev
- Klíčů /TR a /FL
- Písem typu TrueType

Analyzované soubory nesmí obsahovat chyby a neplatné offsety, jinak analýza nejspíše selže, nebo nebude provedena nad problémovými objekty.

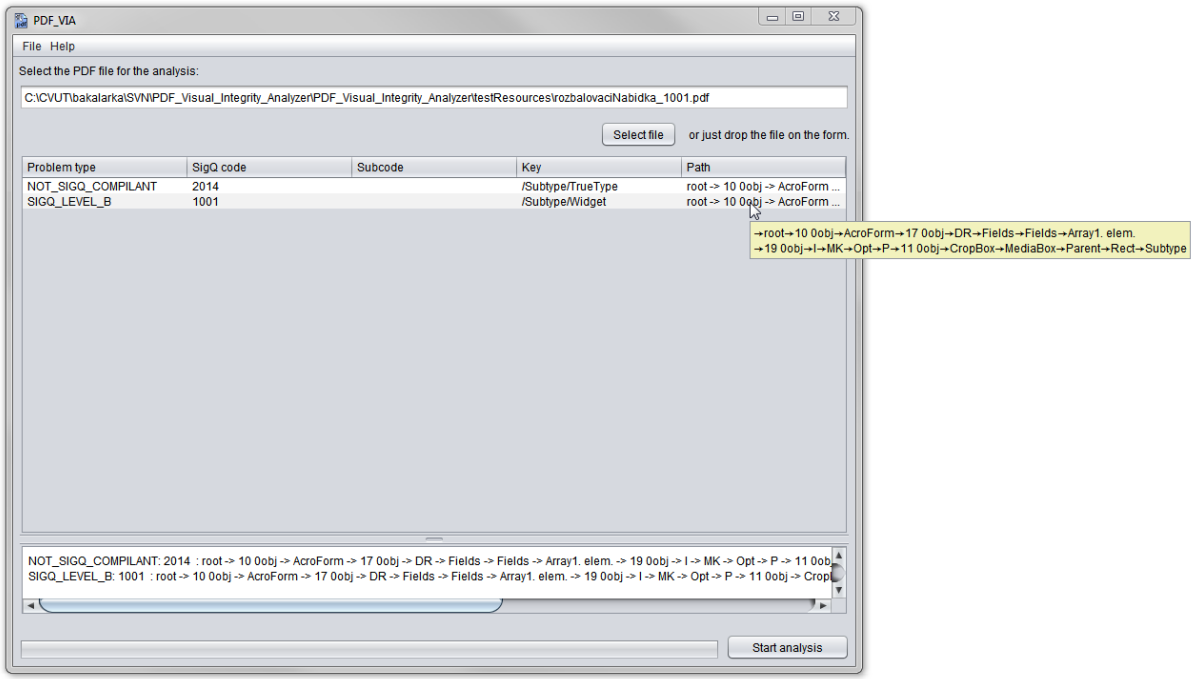
Aplikace byla úspěšně testována na níže uvedeném seznamu vybraných PDF dokumentů, které neobsahují chyby v offsetech a mají problémové vlastnosti, které současná verze aplikace podporuje. Soubory jsou uloženy na přiloženém CD.

**Tabulka 5.1 – Testované PDF dokumenty**

| Název | Velikost |
|---|---|
| 1008_GoToR.pdf | 57.7 kB |
| 1008_Launch_1001_2013_2014.pdf | 12.7 kB |
| 2009-08-05 Form 1 (app media method)_1005_1008.pdf | 146 kB |
| 2009-08-05 Form 2 (app media method)_1005_1008.pdf | 145 kB |
| button_1001.pdf | 12.8 kB |
| comment_FreeText_1002.pdf | 14.5 kB |
| comment_pop_up_1002.pdf | 12.0 kB |
| comment_SWF_1002.pdf | 254 kB |
| comments_3D_1001_1008GoTo3DView.pdf | 26.5 kB |
| comments_3D_1002.pdf | 14.3 kB |
| extGState_FL_2011.pdf | 595 kB |
| extGState_TR_2009.pdf | 595 kB |
| Hidden_action_GoTo3DView_1008_1001.pdf | 16.4 kB |
| importForm_1008_1001.pdf | 68.2 kB |
| interpolateOld_2007.pdf | 13.8 kB |
| JavaScript_alertAfterPrint_1000_1008.pdf | 12.1 kB |
| link_1008_URI.pdf | 297 kB |
| Mini.pdf | 775 B |
| Multimedia_video_1002.pdf | 843 kB |
| Multimedia_zvuk_1002_2014-TrueType_signed.pdf | 327 kB |

| | |
|---|---|
| Multimedia_zvuk_1002.pdf | 296 kB |
| radio_list_selectBox_button_1001.pdf | 9.8 kB |
| radioButton_1001.pdf | 10.6 kB |
| rozbalovaciNabidka_1001.pdf | 7.7 kB |
| seznam_1001.pdf | 7.5 kB |
| Testovaci text03_2014.pdf | 14.2 kB |
| textove_pole_1001.pdf | 7.3 kB |
| vrstvy_1009_E-_P-_Voff_1009.pdf | 69.0 kB |
| vrstvy_1009_E-_P-_Von_1009.pdf | 69.0 kB |
| vrstvy_1009_E-_Poff_V-_1009.pdf | 69.0 kB |
| vrstvy_1009_E-_Pon_V-_1009.pdf | 69.0 kB |
| vrstvy_1009_Eoff_P-_V-_1009.pdf | 69.0 kB |
| vrstvy_1009_Eoff_Poff_Voff_1009.pdf | 69.1 kB |
| vrstvy_1009_Eon_P-_V-_1009.pdf | 69.0 kB |
| vrstvy_1009_Eon_Pon_Von_1009.pdf | 73.3 kB |
| vrstvy_SetOCGState_1008_1001.pdf | 196 kB |
| zasrtavaci_pole_1001.pdf | 8.5 kB |

## 5.3 Práce s aplikací




**Obrázek 3 - Okno demonstrační aplikace PDF VIA**

Práce s aplikací PDF VIA je velmi jednoduchá, stačí přetáhnout PDF dokument, který chceme analyzovat nad okno aplikace a stisknout tlačítko „Start analysis“. Po chvíli jsou v tabulce zobrazeny výsledky analýzy. Při najetí myší na určité políčko se zobrazí celý obsah daného pole.

# 6 Závěr

Práce se zabývá problematikou vizuální integrity elektronicky podepsaných PDF dokumentů. Přestože je formát PDF otevřený a jeho specifikace je volně dostupná, prvky, které ovlivňují vizuální integritu, v ní popsány nejsou. Cílem této práce bylo problémové prvky identifikovat a implementovat jednoduchou demonstrační aplikaci, která tyto prvky dokumentu nalezne.

Na začátku práce na tomto projektu jsem se seznámil se specifikací formátu PDF. Tato specifikace je velmi obsáhlá, tudíž nebylo možné se v části práce popisující formát PDF zabývat všemi aspekty. Proto byly vybrány pouze části, které jsou důležité k pochopení analytické části práce, popisují jednotlivé objekty a základní strukturu PDF souborů.

Ve 4. kapitole jsou popsány výsledky analýzy, jejímž cílem bylo identifikovat jednotlivé prvky, které by mohly způsobit nekonzistenci při zobrazení dokumentů. Pokud je daná konstrukce v dokumentu přítomna, tak to ještě neznamená, že daný dokument bude nutně zobrazen nekonzistentně, ovšem přítomnost prvků označených v analýze za nebezpečné by mohla být zneužita a proto bych doporučil tyto prvky nepoužívat, pokud to není nutné. Podařilo se dohledat většinu prvků, které při podpisu vyvolávají varování o možném problému vizuální integrity i několik potencionálně zneužitelných vlastností, které varování v této aplikaci nevyvolávají. U všech nalezených příkladů byla identifikována konstrukce, která hlášení s největší pravděpodobností vyvolala. U některých příkladů byl jako důkaz uveden i protipříklad, který zkoumanou chybu nevyvolává a liší se od příkladu právě ve zkoumané konstrukci.

Jedním z hlavních cílů této práce bylo ověřit možnost realizace jednoduché a prakticky použitelné aplikace, která vyhledává v PDF dokumentu problémové konstrukce. Proto je v 5. kapitole práce představena jednoduchá demonstrační aplikace PDF VIA pro kontrolu prvků porušujících vizuální integritu. Aplikace načte tabulku referencí, případně cross-reference stream, pak započne samotná analýza. Postupuje se od kořenového objektu, postupně prochází všechny slovníky a pole, které jsou použité v poslední verzi souboru, a vyhledává v nich nebezpečné konstrukce. Výsledky jsou potom představeny uživateli v přehledné tabulce i s umístěním problematické konstrukce v souboru.

## 6.1 Možná zlepšení

V analytické části byly objeveny možné chyby, které se nedařilo navodit ani v Adobe Acrobatu X Pro ani přímou úpravou souborů. V další části projektu by bylo možné použít ostatní verze Adobe Acrobatu a snažit se vytvořit příklady, které ve verzi 10 vytvořit nešly, případně věnovat další čas práci na manuální tvorbě chybějících příkladů a jejich rozboru.

V současné implementaci aplikace PDF VIA by bylo možné přidat podporu detekce pro další problémové konstrukce, které současná verze aplikace není schopná detekovat. Případní uživatelé by jistě ocenili, kdyby se v budoucích verzích objevila funkce, která by pro běžného uživatele srozumitelněji specifikovala polohu nalezeného problému. Zatím není implementována podpora pro načítání PDF souborů, ve kterých se objekty nenacházejí na správném offsetu udaném v cross-reference tabulce nebo streamu.

# Literatura


[1] PDF Reference, sixth edition, Adobe Portable Document Format, ver. 1.7,
Adobe Systems Incorporated, 2006,
<http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_1-7.pdf>, citováno 22. 11. 2013

[2] Analýza prvků, které ovlivňují vizuální integritu digitálně podepsaného PDF dokumentu,
Kouba Z., Suchý O., Oponovaná technická zpráva č. GL 234/14, Zdadavatel Software 602 a.s.,
ČVUT FEL, Praha 2074

[3] Digital Signiture User Guide for Acrobat 9.0 and Adobe Reader 9.0,
Adobe Systems Incorporated, Nov 2008,
<http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/acrobat_digsig_userguide_90.pdf>, citováno 22. 11. 2013

[4] Digital Signatures & Rights Management in the Acrobat Family of Products,
Adobe Systems Incorporated, 29. 3. 2010,
<http://www.adobe.com/content/dam/Adobe/en/devnet/reader/pdfs/acrobat_reader_security_9x.pdf >, citováno 22. 11. 2013

[5] Digital Signatures in XFA Documents,
Adobe Systems Incorporated, 23. 5. 2007,
<https://wwwimages2.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/digisig_in_XFA.pdf>, citováno 25. 3. 2014

[6] Adobe® LiveCycle® Digital Signatures ES
Best practices to streamline your business, Technical Guide, Adobe®, 19. 6. 2008,
<http://www.adobe.com/content/dam/Adobe/en/devnet/livecycle/pdfs/lc_digitalsig_wp_ue.pdf> ,citováno 25. 3. 2014

[7] Zpracování inteligentních formulářů pomocí tenkého klienta,
Pavel Stolař,
<https://dip.felk.cvut.cz/browse/pdfcache/stolapav_2012dipl.pdf>, citováno 14. 12. 2014

[8] Portable Document Format,
<http://en.wikipedia.org/wiki/Portable_Document_Format>, citováno 14. 12. 2014

[9] The history of PDF,
Laurens Leurs, 9. 8. 2013
<http://www.prepressure.com/pdf/basics/history>, citováno 14. 12. 2014

[10] OpenOffice.org 1.1 Features
<http://www.openoffice.org/dev_docs/features/1.1/index.html>, citováno 20. 11. 2014

[11] The MD5 Message-Digest Algorithm,
MIT Laboratory for Computer Science and RSA Data Security, Inc., Duben 1992,
<http://tools.ietf.org/html/rfc1321>, citováno 12. 12. 2014

[12] How to Break MD5 and Other Hash Functions,
Shandong University, Jinan,
<http://merlot.usc.edu/csac-f06/papers/Wang05a.pdf>, citováno 12. 12. 2014

[13] Finding Collisions in the Full SHA-1,
<http://people.csail.mit.edu/yiqun/SHA1AttackProceedingVersion.pdf>,
citováno 12. 12. 2014

[14] Přehled kvalifikovaných poskytovatelů certifikačních služeb a jejich kvalifikovaných služeb
<http://www.mvcr.cz/clanek/prehled-kvalifikovanych-poskytovatelu-certifikacnich-sluzeb-a-jejich-kvalifikovanych-sluzeb.aspx>, citováno 12. 12. 2014

# Příloha A Obsah přiloženého CD

- Složka “PDF_Visual_Integrity_Analyzer“
  - Tato složka obsahuje zdrojový kód demonstrační aplikace PDF VIA.
- Složka „zdokumentovane_priklady“
  - Příklady získané v analytické části.
- Soubor “suchyon6_bp_Analyza_vizualni_integrity_pdf_2015.pdf“.
  - Tato práce ve formátu PDF.

# Příloha B Seznam SigQ kódů

Níže uvedená tabulka vychází ze seznamu kódů uvedených v dokumentu [3] v tabulkách Table 13 až Table 16.

| Kód | Text zprávy | Význam |
|---|---|---|
| 1000 | Document contains hidden behavior | The document contains hidden actions that may not be intended or known by the end user. Actions include JavaScript actions (document open, save, etc.), playing multimedia, executing a menu item, and so on. |
| | *Dokument obsahuje skryté chování* | *Tento dokument obsahuje skryté akce, které nemusí být určeny nebo být známy koncovému uživateli. Akce zahrnují akce JavaScriptu (otevření, uložení dokumentu atd.), přehrávání multimédií, provádění položek nabídek atd.* |
| 1001 | Comment or form field may silently change | The document contains non-signature form fields. Such fields' visual appearances may change based on external variables. |
| | *Poznámky nebo pole formuláře se mohou změnit bez upozornění* | *Tento dokument obsahuje jiná pole formuláře než pole podpisu. Vizuální vzhled takových polí se může změnit na základě externích proměnných.* |
| 1002 | Comment or form field may silently change | The document contains comments. Comments' visual appearances may change based on external variables. |
| | *Poznámky nebo pole formuláře se mohou změnit bez upozornění* | *Tento dokument obsahuje poznámky. Vizuální vzhled poznámek se může změnit na základě externích proměnných.* |
| 1003 | Document may silently launch menu items | The document contains named actions that may launch menu items without the user's knowledge. |
| | *Dokument může bez upozornění spouštět položky nabídek* | *Dokument obsahuje pojmenované akce, které mohou spouštět položky nabídek bez vědomí uživatele.* |
| 1004 | Presentation elements may change appearance | Presentations are not allowed since a presentation may contain animations or other elements that may change document appearance or behavior. |
| 1005 | The document contains a dynamic form | XFA-based (dynamic forms) documents are not allowed since such forms could alter the document’s appearance or behavior. |
| | *Tento dokument obsahuje formulář založený na XFA* | *Tento dokument je založený na XFA a proto může na pozadí změnit vzhled nebo chování dokumentu.* |
| 1006 | Document contains links to external PDFs | The document links to external PDFs on the Internet, file system, or network and it has no control over the nature of that linked content. Embedded Go-To actions must not refer to external hierarchies. |
| 1007 | Comment or form field may silently change | Disallowed annot type: <annot type>. One or more form fields are associated with a 3D object, file attachment, mul- |

| | | |
|---|---|---|
| | | timedia, or other dynamic objects. |
| 1008 | Document contains hidden behavior | Disallowed action type: <action type>. The document contains hidden actions that may not be intended or known by the end user. Actions include JavaScript actions (document open, save, etc.), playing multimedia, executing a menu item, and so on. |
| | *Dokument obsahuje skryté chování* | *Nepovolený typ akce: <typ akce>. Tento dokument obsahuje skryté akce, které nemusí být určeny nebo být známy koncovému uživateli. Akce zahrnují akce JavaScriptu (otevření, uložení dokumentu atd.), přehrávání multimédií, provádění položek nabídek atd.* |
| 1009 | Document contains hidden behavior | The document's content is divided into layers that can be silently displayed or hidden on the fly. |
| | *Dokument obsahuje skryté chování* | *Obsah dokumentu je rozdělen do vrstev, které mohou být za běhu bez upozornění zobrazeny nebo skryty.* |
| 2004 | Page content may silently change | Visual elements may change based on external variables. For example, a logo may change color based on time or zoom level. No postscript XObjects allowed. |
| | *Obsah stránky se může změnit bez upozornění* | *Vizuální elementy se mohou měnit podle externích proměnných. Například logo může změnit barvu podle času nebo úrovně zvětšení. Nejsou povolené postscriptové XObjects.* |
| 2006 | Document may not open in the future | Some or all of the content is encrypted and the encryption method is not available in standard Acrobat installations. For example, the document may be protected by the Adobe Policy Server. Document contain streams encrypted using crypt filter. |
| | *Dokument se nemusí v budoucnu otevřít* | *Část nebo celý obsah je šifrovaný a metoda šifrování není dostupná ve standardních instalacích Acrobatu. Dokument může být například chráněn serverem Adobe LiveCycle Rights Management. Dokument obsahuje streamy zašifrované šifrovacím filtrem.* |
| 2007 | Page content may silently change | The document author has enabled image interpolation. No image interpolation is allowed. |
| | *Obsah stránky se může změnit bez upozornění* | *Autor dokumentu povolil interpolace obrazu, interpolace obrazu není povolena.* |
| 2009 | | The document uses a PDF transfer function that interprets and replaces color. For example, it could replace black with white. Extended graphic state should not use the TR key. |
| 2010 | | The document uses a PDF transfer function that interprets and replaces color. For example, it could replace black with white. If present, the extended graphic state's TR2 key must be set to default. |
| 2011 | | The document's extended graphic state uses the FL key. The key is a number that indicates how much flatness tolerance should exist when drawing objects. Content may display differently from Acrobat to other applications. |

| | | |
|---|---|---|
| | | *Rozšířený grafický stav dokumentu používá klíč FL. Klíč je číslo, které označuje kolik tolerance slučování by mělo existovat při vykreslování objektů. Obsah se může zobrazovat jinak v Acrobatu a jinak v jiných aplikacích.* |
| 2012 | | Image XObject must not contain an alternate version. |
| | | *Formulářový XObject nesmí obsahovat alternativní verze.* |
| 2013 | Text appearance may silently change | Document contains non-embedded fonts. When the document opens on a system that does not have the requisite fonts, Acrobat will replace them with some other font.<br><br>Users should always turn on font-related warnings.<br>The non-embedded fonts warning is turned off by default. It can be turned on by setting the DigSig\bEnNonEmbFontLegPDFWarn preference to true. The disallowed font type warning is also turned off by default and can be turned on by setting the DigSig\bTrueTypeFontPDFSigQWarn preference to true. |
| | *Vzhled text se může změnit bez upozornění* | *Dokument obsahuje písma, která nejsou vložená. Když se dokument otevře v systému, který nemá nutná písma, Acrobat je nahradí jinými písmy.* |
| 2014 | Text appearance may silently change | Disallowed font type: <font type>. True type and TrueType-based OpenType fonts are not allowed because they are programs and may change the document's appearance based on external variables. |
| | *Vzhled text se může změnit bez upozornění* | *Tento dokument používá písma TrueType. Písma TrueType a OpenType založené na TrueType nejsou povolené, protože jsou to programy a mohou změnit vzhled dokumentu podle externích proměnných.* |
| 3000 | Document links to external content | Document links to images not in the PDF. No external XObjects allowed. |
| | *Vazby dokumentu na externí obsah* | |
| 3001 | | Document links to images not in the PDF that are used as alternates. For example, an alternate, high resolution images might be specified for printing. Images must not contain an OPI alternate version. |
| 3002 | | Document contains external streams. The author has flagged some PDF bytes as a stream which may get data from an external source. |
| | | *Dokument obsahuje externí streamy. Autor označil některé byty v PDF jako stream, který může získávat data z externího zdroje.* |
| 3003 | | Document links to images not in the PDF that are used as alternates. For example, an alternate, high resolution images might be specified for printing. Form XObject must not contain an OPI alternate version. |

| | | |
|---|---|---|
| | | *Dokument obsahuje vazby na obrazy použité jako alternativy, které nejsou v PDF. Například mohou být určeny alternativní obrazy ve vyšším rozlišení pro tisk. Obrazy nesmí obsahovat alternativní verze OPI.* |
| 4000 | Unrecognized PDF content | Unrecognized PDF content: The document contains PDF content or custom content not supported by the current version of Acrobat. The document may have been created by a later version of Acrobat. |
| | *Nerozpoznaný obsah PDF* | *Nerozpoznaný obsah PDF: V dokumentu je obsah PDF nebo uživatelský obsah, který aktuální verze Acrobatu nepodporuje. Zdá se, že tento dokument mohl být vytvořen novější verzí Acrobatu.* |
| 4001 | Page content may silently change | Unrecognized drawing operator: The document contains PDF content or custom content not supported by the current version of Acrobat. The document may have been created by a later version of Acrobat. |
| | *Nerozpoznaný obsah PDF* | *Nerozpoznaný obsah PDF: V dokumentu je obsah PDF nebo uživatelský obsah, který aktuální verze Acrobatu nepodporuje.* |
| 4002 | Page content may silently change | Malformed drawing instructions: Syntax error. Page content violates the grammar for page content definition. For example, the instruction might specify drawing a square but the syntax for doing it is incorrect. |
| | *Obsah PDF obsahuje chyby* | *Nesprávné kreslící instrukce: Syntaktická chyba. Obsah stránky porušuje gramatiku definice obsahu stránky. Například instrukce může určovat kreslení čtverce, ale syntaxe je nesprávná.* |
| 4003 | – | – (V dokumentaci není tento SigQ kód zmíněn.) |
| | *Nerozpoznaný obsah PDF* | *Tento soubor PDF obsahuje data, která nebyla zpracována. Taková data mohou změnit vizuální vzhled tohoto souboru, když se tento soubor zobrazí v jiné aplikaci.* |