

Syntaxe XML

XML – teorie a praxe značkovacích jazyků (4IZ238)

Jirka Kosek

Poslední modifikace: \$Date: 2009/10/01 19:46:33 \$

Copyright © 2001-2009 Jiří Kosek

Obsah

Základy syntaxe	3
Elementy a atributy	4
Znakový model XML	5
Komentáře	6
Instrukce pro zpracování	7
Sekce CDATA	8
Kontrola správné syntaxe XML dokumentů	9
Entity – fyzická struktura dokumentu	10
Entity	11
Interní textové entity	12
Externí textové entity	13
Externí binární entity	14
Jmenné prostory	15
Jmenné prostory	16
Zkracování zápisu	17
Pár poznámek	18
XML Infoset	19
XML Infoset	20
Další informace	21
Další informace	22

Základy syntaxe

Elementy a atributy	4
Znakový model XML	5
Komentáře	6
Instrukce pro zpracování	7
Sekce CDATA	8
Kontrola správné syntaxe XML dokumentů	9

Elementy a atributy

- element, tag, počáteční tag, ukončovací tag, obsah elementu
- elementy se nesmí křížit
- tagy musí být spárované nebo má element prázdný obsah
- kořenový element – obaluje celý dokument
- atribut, název atributu, hodnota atributu
- názvy elementů a atributů – přesná definice ve specifikaci XML¹
 - začínají písmenem, podtržítkem nebo dvojtečkou
 - další znaky jsou písmena, číslice, tečka, pomlčka, podtržítko, dvojtečka a některé další znaky
- zápis vyhrazených znaků – pomocí entit (< & > ' ")

¹ <http://www.w3.org/TR/REC-xml>

Znakový model XML

- XML dokumenty používají znakovou sadu ISO 10646
- 21bitová znaková sada, v současné době je definováno přes 100 tisíc znaků
- zcela shodné s Unicode
- kódování: UTF-16 a UTF-8
- UTF-16
 - 16bitové slovo přímo obsahuje kód znaku
 - další téměř milion znaků je dostupných pomocí „surrogates“ – 1 znak = dvě 16bitová slova
- UTF-8
 - text může být kódován jako sekvence bajtů
 - 1 znak = 1 až 4 bajty
 - kompatibilní s ASCII

Obrázek 1. Kódování UTF-8

Scalar Value	UTF-16	1st Byte	2nd Byte	3rd Byte	4th Byte
00000000 0xxxxxxx	00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzzyyyy yyxxxxxx	zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
000uuuuu zzzzyyyy yyxxxxxx	110110ww wwzzzzyy 110111yy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

- Where uuuuu = wwwww + 1 (to account for addition of 10000_{16} as in Section 3.7, *Surrogates*).

- standard XML vyžaduje, aby všechny aplikace podporovaly alespoň UTF-8 a UTF-16
- lze použít i jiné kódování, musí se uvést v XML deklaraci

```
<?xml version="1.0" encoding="iso-8859-2"?>
```

musí být první řádka dokumentu

- znak s libovolným kódem z ISO 10646 můžeme zapsat pomocí entity `&#kód;` (*kód* je číslo v desítkové soustavě) nebo `&#xkód;` (*kód* je číslo v šestnáctkové soustavě)

Komentáře

- slouží pro vkládání poznámek nebo pro dočasné vyřazení části dokumentu
- nesmí obsahovat -- a nemohou být navzájem vnořené

```
<!-- ukázkový komentář -->
```

Instrukce pro zpracování

- instrukce pro zpracování = processing instructions
- standardní způsob pro začlenění nestandardních dat
- interpretace závisí na aplikaci:
 - připojování stylu k dokumentu
 - příkazy pro různé preprocesory
 - poslední pozice editace, ruční zlom řádky/stránky

```
<?xml-stylesheet href="styl.css" type="text/css"?>
```

```
<datum>  
  <?php echo Date("d.m.Y")?>  
</datum>
```

```
<?myDTP page-break?>
```

Sekce CDATA

- pro delší úseky dokumentu, kde nechceme interpretovat pro XML významné znaky jako '<', '&', ...
- použití:
 - ukázky XML a HTML kódu
 - zařazování programů přímo do XML dokumentů

```
<výpis>
<![CDATA[
Můžu použít klidně <tag> a nikomu to
nevadí. Ani firmě Son & Brother.
]]>
</výpis>
```

je zcela ekvivalentní s

```
<výpis>
Můžu použít klidně &lt;tag> a nikomu to
nevadí. Ani firmě Son & Brother.
</výpis>
```

- omezení – CDATA sekce nesmí obsahovat]]>

Kontrola správné syntaxe XML dokumentů

- well-formed = správně strukturovaný dokument
 - dokument splňuje základní syntaktická pravidla
 - takový dokument by měla zpracovat každá aplikace s podporou XML
 - lze ověřit pomocí parseru

Entity – fyzická struktura dokumentu

Entity	11
Interní textové entity	12
Externí textové entity	13
Externí binární entity	14

Entity

- jeden XML dokument může být uložen v několika souborech
- entity = části, ze kterých se XML dokument skládá
- druhy entit:
 - interní textové entity
 - externí textové entity
 - externí binární entity
 - parametrické entity (používají se pouze v DTD)
- deklarace entity se provádí buď v externě připojeném DTD nebo přímo uvnitř deklarace typu dokumentu:

```
<!DOCTYPE dokument [  
    deklarace entit  
>  
<dokument>  
    ...  
</dokument>
```

- obecná syntaxe deklarace entity:

```
<!ENTITY název definice>
```

- použití entity v dokumentu se provádí pomocí odkazu na entitu

```
&název;
```

Interní textové entity

- nadefinování entity pro často používané části textu
- entity pro znaky, které nejsou dostupné na klávesnici
- rychlé a konzistentní úpravy dokumentu
- syntaxe:

```
<!ENTITY název "text">
```

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE manuál [
<!ENTITY program "SuperSoft 3.5">
<!ENTITY nbsp    "&#160;">
]>
<manuál>
  <název>&program;</název>
  <podtitulek>Uživatelská příručka</podtitulek>
  <kapitola>
    <název>Úvod</název>
    <para>&program; je nejlepší aplikace ve své
      kategorii. Díky programu &program; můžete
      snadno vařit, prát a&nbsp;dokonce
      i&nbsp;leštit nábytek. Všechny funkce
      programu &program; se ovládají hlasovým
      vstupem.</para>
    ...
  </kapitola>
</manuál>
```

Externí textové entity

- rozdělení XML dokumentu do více souborů
- modularizace dokumentů, možnost opakovaného použití
- lepší správa, možnost editace jednoho dokumentu více uživateli najednou
- **syntaxe:**

```
<!ENTITY název SYSTEM "URI">
```

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE manuál [
<!ENTITY program "SuperSoft 3.5">
<!ENTITY kapitola1 SYSTEM "kap1.xml">
]>
<manuál>
  <název>&program;</název>
  <podtitulek>Uživatelská příručka</podtitulek>
  &kapitola1;
</manuál>
```

```
-----
<?xml version="1.0" encoding="windows-1250"?>
<kapitola>
  <název>Úvod</název>
  <para>&program; je nejlepší aplikace ve své
  kategorii. Díky programu &program; můžete
  snadno vařit, prát a&nbsp;dokonce
  i&nbsp;leštit nábytek. Všechny funkce
  programu &program; se ovládají hlasovým
  vstupem.</para>
  ...
</kapitola>
```

- pokud není entita v kódování UTF-8 nebo UTF-16 musí začínat deklarací kódování
- deklarace kódování je podobná deklaraci XML, ale atribut `version` je nepovinný

Externí binární entity

- nevkládají se přímo do dokumentu
- vytváříme si „jméno“, které označuje externí soubor
- název entity lze použít pouze jako hodnotu atributu speciálního typu
- syntaxe:

```
<!ENTITY název SYSTEM "URI" NDATA "notace">
```

- tato možnost byla převzata z SGML, ale prakticky se moc nepoužívá
- externí soubory se vkládají uvedením jejich jména přímo v nějakém atributu – podobně jako např. obrázky v HTML

Jmenné prostory

Jmenné prostory	16
Zkracování zápisu	17
Pár poznámek	18

Jmenné prostory

- slouží k rozlišení elementů a atributů se shodnými jmény v případech kdy by mohlo dojít ke konfliktům
- aplikace si vybere jen ty části dokumentu, které umí zpracovat
 - kombinování více „sad značek“ dohromady
 - např.: XSLT styly (XSLT instrukce × HTML kód), XHTML stránka s obrázky v SVG, ...
- jména a elementů a atributů se skládají ze dvou částí – ze jmenného prostoru a z lokálního názvu
- jmenné prostory se identifikují pomocí URI adresy, ale nic konkrétního se na ní nevyskytuje, slouží pouze jako identifikátor
- pro zkrácení zápisu se při deklaraci jmenného prostoru vytvoří prefix, který jmenný prostor zastupuje:

```
<prefix:element xmlns:prefix="http://nekde.com/neco"> ...  
</prefix:element>
```

- prefixy je možné použít i u atributů a elementů obsažených v elementu s deklarací

Příklad 1. Dokument se jmennými prostory

```
<ceník:nabídka  
  xmlns:ceník="http://www.ecena.cz/e-cenik"  
  xmlns:bib="http://www.book.org/bibliography">  
<ceník:položka ceník:dph="22%">  
  <ceník:název>  
    <bib:book>  
      <bib:author>Jiří Kosek</bib:author>  
      <bib:title>HTML - tvorba dokonalých  
        WWW stránek</bib:title>  
    </bib:book>  
  </ceník:název>  
  <ceník:cena měna="CZK">259</ceník:cena>  
</ceník:položka>  
</ceník:nabídka>
```

Zkracování zápisu

- implicitní jmenné prostory (prefix je prázdný)

```
<nabídka xmlns="http://www.ecena.cz/e-cenik"
  xmlns:bib="http://www.book.org/bibliography">
  <položka>
    <název>
      <bib:book>
        <bib:author>Jiří Kosek</bib:author>
        <bib:title>HTML - tvorba dokonalých
          WWW stránek</bib:title>
      </bib:book>
    </název>
    <cena měna="CZK">259</cena>
  </položka>
</nabídka>
```

- jmenné prostory se stejným prefixem se navzájem překrývají

```
<nabídka xmlns="http://www.ecena.cz/e-cenik">
  <položka>
    <název>
      <book xmlns="http://www.book.org/bibliography">
        <author>Jiří Kosek</author>
        <title>HTML - tvorba dokonalých
          WWW stránek</title>
      </book>
    </název>
    <cena měna="CZK">259</cena>
  </položka>
</nabídka>
```

Pár poznámek

- prefix + lokální jméno = kvalifikované jméno (QName)
- atribut bez prefixu nepatří přímo do žádného jmenného prostoru:
 - ani když je jeho element v implicitním jmenném prostoru
 - patří tam však „nepřímě“, protože se vždy vztahuje k elementu, u kterého je uvedený

```
<nabídka xmlns="http://www.ecena.cz/e-cenik">
  <položka>
    <název>
      ...
    </název>
    <cena měna="CZK">259</cena>
  </položka>
</nabídka>
```

- atribut `měna` v tomto případě nepatří do jmenného prostoru `http://www.ecena.cz/e-cenik`
- na druhou stranu patří k elementu `cena`, který již do daného jmenného prostoru patří
- specifikace jmenných prostorů doplňuje standard XML
- dokumenty se jmennými prostory nelze validovat oproti DTD bez toho, aby se vytvořil speciální kříženec všech použitých DTD
- jmenný prostor neodpovídá DTD ani XML schématu, nicméně obvykle pro každý jmenný prostor někde existuje popis použitelných elementů a atributů nejčastěji právě ve formě DTD nebo XML schématu
- jmenné prostory se hodně používají v dalších jazycích souvisejících s XML – XML schémata, XSLT, ...

XML Infoset

XML Infoset	20
-------------------	----

XML Infoset

- abstraktní datový model pro XML dokumenty
- ke každému XML dokumentu existuje abstraktní reprezentace v podobě infosetu (stromová reprezentace)
 - místo pojmu strom se používá pojem infoset (information set = množina informací)
 - místo pojmu uzel se používá pojem information item = položka informace
 - každá položka má vlastnosti – dětské položky, svůj obsah, deklarované NS, ...
- dokumenty s DTD mohou „modifikovat“ infoset – např. pomocí defaultní nebo fixní hodnoty atributu
- je využíván v dalších standardech
- PSVI = Post Schema Validation Infoset
 - otypovaný infoset dokumentu
 - využívá se např. v dotazovacích jazycích (XQuery), které potřebují znát typy dat v jednotlivých elementech a attributech
 - ještě se k němu vrátíme u XML schémat

Další informace

Další informace	22
-----------------------	----

Další informace

Povinná četba

- Kapitola o syntaxi XML²

Doporučená četba

- Anotovaná specifikace XML³

² <http://www.kosek.cz/knihy/phpxml/php5xml-ukazka.pdf>

³ <http://www.xml.com/axml/testaxml.htm>