



XPath výrazy, funkce a aplikační události v 602XML

Verze dokumentu:

Vztaženo k aplikacím 602XML Filler a 602XML Designer:

Datum uveřejnění:

2.0

2.06

28.02.07

Obsah

1	Úvod	3
2	XPath – úvod	3
3	XPath – terminologie	3
3.1	Uzly	3
3.2	Vztahy uzlů	3
4	XPath – syntax	4
4.1	Výběr uzlů	4
4.2	Identifikátory osy	4
4.3	Testy uzlu a predikáty	5
4.4	Zástupné znaky	5
4.5	Operátory	6
5	XPath – funkce	6
5.1	Popis standardních funkcí XPath	6
5.2	Popis funkcí 602XML rozšíření jazyka XPath	8
6	Funkce a aplikační události v rozšíření 602XML	11
6.1	Aplikační událostí rozšíření 602XML	17
7	Použití XPath v aplikacích 602XML	20
8	Literatura	20

1 Úvod

Tento dokument stručně popisuje principy jazyka XPath verze 1.0, jeho syntax, možné operace a funkce, a to včetně rozšíření 602XML. Dokument by měl přispět ke zjednodušení návrhu složitějších konstrukcí v XML formulářích pomocí aplikace 602XML Designer.

2 XPath – úvod

XPath je jazykem, který stojí na pozadí téměř veškeré práce s XML. Slouží k adresování částí XML dokumentu. Pomocí výrazů jazyka XPath je možno vyhledávat informace v XML dokumentu a dále pohybovat se mezi elementy a atributy v tomto dokumentu. XPath je součástí standardu XSLT podle specifikace W3C.

Pomocí XPath výrazů je tedy možno vybírat množiny uzlů (node-sets) v XML dokumentu. XPath výrazy jsou konstruovány obdobným způsobem, jakým jsou adresovány soubory a adresáře v tradičním počítačovém systému souborů. XPath specifikace obsahuje přes 100 zabudovaných funkcí – funkce pro práci s řetězci, čísla, datovými a časovými údaji, funkce pro manipulaci s uzly a prvky QName (kvalifikované jméno), práci se sekvencemi, logickými hodnotami a další.

Podrobněji se s jednotlivými součástmi jazyka XPath seznámíme v následujících kapitolách.

3 XPath – terminologie

3.1 Uzly

Specifikace XPath definuje 7 druhů uzlů – elementy, atributy, texty, jmenné prostory (namespace), předpisy (processing-instruction), komentáře (comment) a uzly dokumentů (root). Všechny XML dokumenty jsou chápány jako stromy uzlů. Kořen takového stromu je nazýván uzlem dokumentu (tzv. kořenový uzel).

3.2 Vztahy uzlů

Mezi uzly ve stromu příslušného XML dokumentu definujeme jednotlivé vztahy, které usnadňují orientaci a práci s XML dokumentem pomocí XPath. Role v těchto vztazích jsou – rodič (parent) neboli přímý předek, dítě (child) neboli přímý potomek, sourozenec (sibling), předek (ancestor) a potomek (descendant).

Příklad:

```
        </dsig:SignatureProperty>
    </dsig:SignatureProperties>
</dsig:Object>
<dsig:Object>
  <root>
    <adresa>
      <ulice>Havrdova 7/1258</ulice>
      <psc>12589</psc>
      <mesto>Praha</mesto>
      <stat>Česká republika</stat>
    </adresa>
  </root>
</dsig:Object>
</dsig:Signature>
```

V tomto příkladu je element *adresa* rodičem (a zároveň předchůdcem) elementů *ulice*, *psc*, *mesto* a *stat*. Elementy *ulice*, *psc*, *mesto* a *stat* jsou dětmi (a zároveň potomky) elementu *adresa*, zároveň jsou mezi sebou sourozenci. Element *root* je předchůdcem elementů *ulice*, *psc*, *mesto* a *stat*, a tyto jsou naopak jeho potomky.

Význam těchto vztahů bude objasněn v následující kapitole, která se zabývá vlastní syntaxí jazyka XPath pro výběr uzlů a vyhodnocování výrazů.

4 XPath – syntax

4.1 Výběr uzlů

Nejdůležitější konstrukcí jazyka XPath je zřejmě konstrukce cesty ke XML uzlu (Location Path). Cesta obsahuje vždy obsahuje několik *kroků*, přičemž každý z nich sestává z *identifikátoru osy*, *testu uzlu* a tzv. *podmínky (predikátu)*. Povinný je pouze test uzlu. Jednotlivé kroky XPath výrazu se pak spojují lomítky, vyhodnocují se zleva doprava a vždy se vychází z předchozích výsledků. Na začátku zpracování celého výrazu se vychází z aktuálního uzlu. Výsledkem může obecně být libovolný typ uzlu, tedy nejen element nebo atribut, ale třeba i textový uzel.

Při vyhodnocování cesty se nejdříve podle tzv. identifikátoru osy určí uzly, které se budou v tomto kroku výrazu zpracovávat. Dle výchozí osy (*child::*) jsou to uzly podřízené aktuálnímu uzlu. Množina vybraných uzlů se omezí podle testu uzlu, kterým může být třeba název elementu, a nakonec se ještě zohlední dodatečné podmínky. Uzly, které v tomto procesu obstály, postupují do dalšího kroku. Ten se pak bude vyhodnocovat postupně pro všechny uzly, které do něj postoupily. Takto se projde celý XPath výraz. Výsledkem výběru je pak množina uzlů, která prošla vyhodnocením.

Stejně jako u adresářové struktury jednotlivé kroky XPath výrazu oddělujeme pomocí lomítka (/). Pokud je lomítko na začátku výrazu, znamená to, že výraz není vztážen k aktuálnímu elementu, ale počítá se od kořene dokumentu. Dvě lomítka bezprostředně za sebou (//) slouží k překonání víceúrovňové struktury. Pokud jsou dvě lomítka na začátku výrazu, opět se jedná o absolutní cestu od kořene dokumentu, takže lze snadno vybrat libovolný element odkudkoli.

Příklady jednoduchých XPath výrazů pro výběr uzlů z příkladového souboru „*XPath sample.xml*“ :

XPath výraz	Aktuální uzel	Vyhodnocení
<code>/root/clovek/jmeno</code>	jakýkoliv	Všechny elementy <i>jmeno</i> , které jsou přímými potomky elementu <i>clovek</i> , jež je přímým potomkem elementu <i>root</i>
<code>/root</code>	jakýkoliv	Kořenový element dokumentu
<code>telefon/domu</code>	clovek	Všechny elementy <i>domu</i> , které jsou přímými potomky elementu <i>telefon</i> , obsaženého v elementu <i>clovek</i>
<code>//domu</code>	jakýkoliv	Všechny elementy <i>domu</i> v jakémkoliv kontextu
<code>root//domu</code>	kořen dokumentu	Všechny elementy <i>domu</i> , i když nejsou přímými potomky elementu <i>root</i>

4.2 Identifikátory osy

Testu uzlu může předcházet takzvaný *identifikátor osy*. Ten určuje směr procházení XML dokumentu, tedy říká, kterým směrem se uzly k vyhodnocení budou vybírat. Pokud identifikátor osy nevedeme, použije se výchozí osa *child::*, a dále tedy postupují pouze potomci aktuálního uzlu.

Identifikátor	Uzly k vyhodnocení
<code>child::</code>	Přímí potomci aktuálního uzlu.
<code>descendant::</code>	Všichni potomci aktuálního uzlu.
<code>descendant-or-self::</code>	Aktuální uzel a všichni potomci.
<code>self::</code>	Aktuální uzel.
<code>ancestor-or-self::</code>	Aktuální uzel a všichni jeho předci.
<code>ancestor::</code>	Všichni předci aktuálního uzlu.
<code>parent::</code>	Rodič aktuálního uzlu.
<code>following::</code>	Všechny uzly, které jsou v toku XML dokumentu za aktuálním uzlem.
<code>preceding::</code>	Všechny uzly, které jsou v toku XML dokumentu před aktuálním uzlem.
<code>following-sibling::</code>	Všichni následující sourozenci aktuálního uzlu.
<code>preceding-sibling::</code>	Všichni předcházející sourozenci aktuálního uzlu.
<code>attribute::</code>	Atributy aktuálního uzlu.
<code>namespace::</code>	Deklarované jmenné prostory

4.3 Testy uzlu a predikáty

Test uzlu dále vymezuje množinu uzlů, která byla určena identifikátorem osy. Rozlišujeme dva základní typy testů uzlu, a to *uzel určený názvem* a *uzel určený typem*. Test uzlu nazývaný *cílený test předpisu* (Targeted Processing Instruction Test) v 602XML nevyužíváme.

Uzel určený názvem je základním testem uzlu. Vybere všechny XML elementy s daným názvem. Tento test zapíšeme jednoduchým zápisem testovaného názvu elementu.

Uzel určený typem bere ohled na typ uzlu, a vybírá pouze určený typ. Tento test se zapisuje pomocí určení typu uzlu následovaným prázdnými kulatými závorkami. K dispozici jsou tyto volby: `comment()`, `text()`, `processing-instruction()` a `node()`.

Do hranatých závorek se dále zapisují další podmínky (predikáty), které zužují výsledky předchozího vyhodnocení. Tyto podmínky se používají k nalezení specifického uzlu nebo uzlu, který obsahuje danou hodnotu. Pokud místo podmínky napíšeme pouze celé číslo, vybere podmínka pouze uzel umístěný na pozici s daným pořadovým číslem.

Příklady testů uzlů v XPath výrazech pro výběr uzlů z příkladového souboru „XPath sample.xml“

XPath výraz	Aktuální uzel	Vyhodnocení
<code>jmeno</code>	<code>clovek</code>	Všechny podřízené elementy s daným názvem
<code>root</code>	kořen dokumentu	Kořenový element dokumentu
<code>//text()</code>	jakýkoliv	Vybere všechny textové uzly v dokumentu.
<code>//clovek[@id="3"]</code>	jakýkoliv	Všechny elementy <i>clovek</i> , jejichž atribut <i>id</i> má hodnotu 3.
<code>*[2]</code>	jakýkoliv	V pořadí druhý element podřízený aktuálnímu.

4.4 Zástupné znaky

Jelikož některé konstrukce nacházejí použití v jazyce XPath velice často, existují pro ně zkrácené tvary a zástupné znaky. Velmi doporučujeme tyto zkrácené tvary používat, protože zvyšují přehlednost celého XPath výrazu a usnadňují tak práci s těmito výrazy. Mezi tyto znaky patří:

- **Hvězdička (*)** - hvězdička použitá jako test uzlu vybírá všechny poskytnuté elementy.
- **Tečka (.)** - tečka nahrazuje odkaz na sebe sama. Je náhražkou celého kroku, takže k němu již nelze připojit podmínky.
- **Dvě tečky (..)** - dvě tečky (stejně jako v adresářové cestě) umožní pohyb o jednu úroveň výše. Jsou opět náhražkou celého kroku, takže k němu již nelze připojit podmínky.
- **Zavináč (@)** - Zavináč nahrazuje identifikátor osy `attribute::`. Jako test uzlu pak obvykle poslouží název atributu.
- **Kombinace (@*)** - tato kombinace použitá jako test uzlu vybírá všechny poskytnuté atributy
- **Příkaz node()** - jako test uzlu vybírá všechny uzly (bez ohledu na jejich typ)

Příklady použití zástupných znaků v XPath výrazech pro výběr uzlů z příkladového souboru „XPath sample.xml“ :

XPath výraz	Aktuální uzel	Vyhodnocení
<code>.</code>	jakýkoliv	Vybere aktuální uzel
<code>./*</code>	jakýkoliv	Vybere všechny přímé potomky aktuálního uzlu
<code>..</code>	jakýkoliv	Vrací rodiče aktuálního uzlu
<code>../*</code>	jakýkoliv	Vybere všechny sourozence aktuálního uzlu včetně jeho samotného
<code>../*[@*=3]</code>	jakýkoliv	Vybere všechny sourozence aktuálního uzlu včetně jeho samotného, pokud mají libovolný svůj atribut roven 3

4.5 Operátory

Výrazy XPath vždy vrací buď množinu uzlů, řetězec, logickou hodnotu nebo číslo. Pro operace s návratovými hodnotami se v XPath používají operátory obvyklé téměř ve všech programovacích jazycích. Využití nacházejí především v podmínkách, a při použití v 602XML také při nastavování vlastností datových vazeb, kde se provádí řada výpočtů a je třeba vyhodnotit a zpracovat velké množství XPath výrazů. Pořadí vyhodnocení operací je možné klasickým způsobem upravovat závorkami.

Operand	Popis	Příklad	Navracená hodnota
	Spojí dvě množiny uzlů	//domu //prace	Vrátí množinu uzlů se jmény <i>domu</i> a <i>prace</i>
+	Sčítání	8 + 4	12
-	Odčítání	8 - 4	4
*	Násobení	8 * 4	32
div	Dělení	8 div 4	2
=	Rovnost	id = 2	TRUE, pokud id je 2, FALSE jinak
!=	Nerovnost	id != 2	TRUE, pokud id není 2, FALSE jinak
<	Menší než	id < 2	TRUE, pokud id je menší než 2, FALSE jinak
<=	Menší než nebo rovno	id <= 2	TRUE, pokud id je menší nebo rovno 2, FALSE jinak
>	Větší než	id > 2	TRUE, pokud id je větší než 2, FALSE jinak
=>	Větší než nebo rovno	id => 2	TRUE, pokud id je větší nebo rovno 2, FALSE jinak
or	Logické OR	id = 2 or id = 3	TRUE, pokud id je 2 nebo 3, FALSE jinak
and	Logické AND	id < 4 and id > 2	TRUE pokud je id menší než 4 a větší než 2, FALSE jinak
not()	Logický zápor	not(id = 2)	TRUE, pokud id je různé od 2, FALSE jinak
mod	Zbytek po dělení (modulo)	5 mod 2	1

5 XPath – funkce

Specifikace XPath obsahuje celou řadu užitečných funkcí, které lze využít především ke tvorbě podmínek a výrazů pro výpočty. V této kapitole popíšeme jak funkce náležející přímo do standardu specifikace XPath, tak funkce použitelné v aplikacích 602XML při konstrukci XPath výrazů jako rozšíření 602XML. Funkce popsané v této kapitole je možno použít ke konstrukci výrazů ve vlastnostech datových vazeb.

5.1 Popis standardních funkcí XPath

V této kapitole budou popsány XPath funkce, které jsou vždy přítomny v knihovně funkcí implementace XPath, která je využívána pro vyhodnocení výrazů.

Každá z těchto funkcí je popsána tzv. funkčním prototypem, který specifikuje typ návratové hodnoty, jméno funkce a typy argumentů funkce. Pokud je typ argumentu následován otazníkem, znamená to, že použití argumentu je volitelné; v opačném případě je argument pro funkci povinný.

Funkce pro práci s uzly:

Funkce	Popis
<i>number last()</i>	Vrací počet v uzlů v právě zpracovávaném seznamu uzlů
<i>number position()</i>	Vrací aktuální pozici uzlu, který je právě zpracováván
<i>number count(node-set)</i>	Vrací počet uzlů v argumentu <i>node-set</i>
<i>node-set id(object)</i>	Vrací elementy vybrané na základě jejich unikátního ID
<i>string local-name(node-set?)</i>	Vrací lokální jméno aktuálního uzlu nebo prvního uzlu v zvolené množině uzlů – bez prefixu jmenného prostoru
<i>string namespace-uri(node-set?)</i>	Vrací URI jmenného prostoru aktuálního uzlu nebo prvního uzlu ve zvolené množině uzlů
<i>string name(node-set?)</i>	Vrací kvalifikované jméno aktuálního uzlu nebo prvního uzlu v zvolené množině uzlů

Funkce pro práci s řetězci:

Funkce	Popis
<i>string</i> string (<i>object</i> ?)	Konvertuje zvolený objekt na objekt typu řetězec. Popis konverze různých typů objektů je popsán na těchto stránkách
<i>string</i> concat (<i>string</i> , <i>string</i> , ...)	Vrací řetězec vzniklý zřetěžením všech argumentů
<i>boolean</i> starts-with (<i>string</i> , <i>string</i>)	Vrací TRUE, pokud první řetězec začíná druhým řetězcem, FALSE jinak
<i>boolean</i> contains (<i>string</i> , <i>string</i>)	Vrací TRUE, pokud první řetězec obsahuje druhý řetězec, FALSE jinak
<i>string</i> substring-before (<i>string</i> , <i>string</i>)	Vrací podřetězec prvního argumentu, který předchází výskytu druhého argumentu v prvním řetězci, nebo prázdný řetězec
<i>string</i> substring-after (<i>string</i> , <i>string</i>)	Vrací podřetězec prvního argumentu, který následuje po výskytu druhého argumentu v prvním řetězci, nebo prázdný řetězec
<i>string</i> substring (<i>string</i> , <i>number</i> , <i>number</i> ?)	Vrací podřetězec prvního argumentu, začínající na pozici specifikované druhým argumentem, o délce rovné třetímu argumentu. Pokud je třetí argument vynechán, je vrácen podřetězec začínající na pozici dle druhého argumentu až po konec původního řetězce
<i>number</i> string-length (<i>string</i> ?)	Vrací počet znaků v řetězci. Pokud je argument vynechán, vrací počet znaků v názvu aktuálního uzlu převedenému na řetězec
<i>string</i> normalize-space (<i>string</i> ?)	Vrací řetězec z nějž jsou odstraněny tzv. whitespaces (bílé znaky). Pokud je argument vynechán, funkce se opět vztahuje k aktuálnímu uzlu převedenému na řetězec
<i>string</i> translate (<i>string</i> , <i>string</i> , <i>string</i>)	Vrací první řetězec, který má znaky z druhého argumentu nahrazeny znaky z odpovídajících pozic řetězce v třetím argumentu

Booleovské funkce:

Funkce	Popis
<i>boolean</i> boolean (<i>object</i>)	Konvertuje zvolený objekt na objekt typu boolean. Popis konverze různých typů objektů je popsán na těchto stránkách
<i>boolean</i> not (<i>boolean</i>)	Vrací TRUE pokud je argument funkce roven FALSE, a naopak
<i>boolean</i> true ()	Vrací TRUE
<i>boolean</i> false ()	Vrací FALSE
<i>boolean</i> lang (<i>string</i>)	Vrací TRUE, pokud je jazyk kontextového uzlu (specifikován atributem xml:lang) je shodným s jazykem specifikovaným v argumentu funkce, nebo je jeho podjazykem.

Číselné funkce:

Funkce	Popis
<i>number</i> number (<i>object</i> ?)	Konvertuje zvolený objekt na objekt typu number. Popis konverze různých typů objektů je popsán na těchto stránkách
<i>number</i> sum (<i>node-set</i>)	Vrací součet hodnot množiny uzlů v argumentu. V případě textových hodnot jsou řetězce převedeny na čísla
<i>number</i> floor (<i>number</i>)	Vrací nejvyšší celé číslo, které je nižší než argument
<i>number</i> ceiling (<i>number</i>)	Vrací nejnižší celé číslo, které je větší než argument
<i>number</i> round (<i>number</i>)	Vrací nejbližší celé číslo

V konkrétním prostředí (například XSLT) mohou k funkcím popsaným výše být doplněny ještě další funkce. Navíc je možné (jak je tomu také v aplikacích 602XML) přidat k výchozím funkcím i vlastní doprogramované rutiny.

5.2 Popis funkcí 602XML rozšíření jazyka XPath

Funkcionalitu, která je nad rámec možností výpočetního modelu XPath jazyka (implementovaného v komponentě MSXML4), je možné integrovat do formuláře pomocí funkcí tzv. 602XML rozšíření. Při používání funkcí 602XML rozšíření XPath je nutno brát v úvahu několik faktů:

- Vyhodnocování funkcí 602XML rozšíření neprobíhá přímo na úrovni XML parseru MSXML4, který 602XML využívá, ale přes základní vrstvu XMLCore, což může vést ke zpomalení funkčnosti.
- Funkce 602XML rozšíření je nutno uzavírat do uvozovek, jinak nebudou fungovat!
- Závorkovací konvence – pokud je potřeba, aby nebyla textová hodnota předaná funkcí rozšíření zpracovávána interním XML parserem, je potřeba tuto hodnotu uzavřít do složených závorek. Je nutno dodržovat zejména pokud předáváme funkci jako parametr řetězec složený z více jednotlivých řetězců (např. výraz `{../d:delitel div ../d:delenec}`)

Přehled funkcí 602XML rozšíření specifikace XPath použitelných pro výpočty vlastností datových vazeb je v následující tabulce:

Funkce	Popis	Příklady volání
DataTransform (<i>string, string, [string, string]</i>)	Funkce volá transformaci, kdy na jejím vstupu je dokument nebo jeho část určená referenčním uzlem a transformovaný dokument je vrácen jako návratová hodnota. První parametr specifikuje XSL stylesheet, který se má použít, druhý parametr pak referenční XPath výraz pro určení uzlu. Poslední dva (nepovinné) parametry specifikují tzv. mode, resp. modeuri parametr XSLT transformace (pokud je v transformaci použit) – více o parametrech mode a modeuri viz. http://www.w3.org/TR/xslt#data-model	DataTransform(rc, ...) DataTrasform(pocet_dni, ../d:doba_trvani)
StrCat (<i>string, string, ...</i>)	Funkce spojuje jednotlivé parametry a vytváří jeden řetězec jako návratovou hodnotu.	StrCat(DataXPath(..d:jmeno),{ }, DataXPath(..d:prijmeni))
StrCmp (<i>string, string</i>)	Funkce porovná dva řetězce předané jako parametry a vrací hodnotu TRUE v případě shody, FALSE jinak.	StrCmp(DataXPath(..d:jmeno), DataXPath(..d:skolitel/d:jmeno))
Round (<i>string, string</i>)	Funkce vrací zaokrouhlenou vstupní hodnotu. První parametr specifikuje původní hodnotu. Druhý parametr specifikuje počet desetinných míst použitých pro zaokrouhlení.	Round(DataXPath(..d:cislo), 2)
And (<i>string, string</i>)	Logické AND. Vrací TRUE, pokud jsou oba argumenty rovny TRUE, FALSE jinak.	And(DataXPath(..d:bool1), DataXPath(..d:bool2))
Or (<i>string, string</i>)	Logické OR. Vrací TRUE, pokud je alespoň jeden argument roven TRUE, FALSE jinak.	Or(DataXPath(..d:bool1), DataXPath(..d:bool2))
Not (<i>string</i>)	Logický zápor. Vrací TRUE, pokud je argument roven FALSE, a naopak.	Not(DataXPath(..d:bool1))

Funkce	Popis	Příklady volání
CallDll (<i>string</i> , <i>string</i> , <i>string</i> , <i>string</i> , ...)	Umožňuje volat metodu z externí DLL knihovny. První argument je cesta k DLL knihovně, druhý je jméno volané metody. Třetí parametr definuje typ volání metody; může nabývat hodnot typu <i>hwnd_ppStr_XpStr</i> , kde X je počet parametrů (tedy např. <i>hwnd_ppStr_2pStr</i> pro 2 parametrovou metodu). Dalšími argumenty funkce jsou pak jednotlivé parametry. Návrátová hodnota funkce je typu <i>string</i> .	CallDll(c:_sample\test.dll, _RunFunction@12, hwnd_ppStr_pStr, GetFormDataXML())
If (<i>string</i> , <i>string</i> , <i>string</i>)	Podle vyhodnocení logické hodnoty v prvním argumentu provede příslušnou akci. Druhý parametr specifikuje akci odpovídající hodnotě TRUE, třetí hodnotě FALSE.	If(DataXPath({../d:polozka1}), spravne, chyba)
IsEmpty (<i>string</i>)	Funkce vrací TRUE v případě, že vstupní parametr je prázdný, FALSE jinak.	IsEmpty(DataXPath({../d:sest}))
CmdLineParam ()	Funkce vrací jako <i>string</i> text příkazové řádky použité pro spuštění Filleru.	
CurrentUserName ()	Funkce vrací uživatelské jméno přihlášeného uživatele do systému.	
CurrentFileName ()	Funkce vrací název resp. cestu k otevřenému formuláři.	
CurrentDate (<i>format</i>)	Funkce vrací aktuální systémové datum počítače. Pokud je zadán parametr pro formát datumu, je systémové datum přeformátováno do požadovaného tvaru.	CurrentDate(ddd)
CurrentTime (<i>format</i>)	Funkce vrací aktuální systémový čas počítače. Pokud je zadán parametr pro formát času, je systémové čas přeformátován do požadovaného tvaru.	CurrentTime(HH:mm)
CurrentTimeUTC (<i>format</i>)	Funkce vrací aktuální UTC čas (Coordinated universal time). Pokud je zadán parametr pro formát času, je hodnota přeformátována do požadovaného tvaru.	CurrentTimeUTC(HH:mm)
FormID ()	Funkce vrací ID otevřeného formuláře.	
FormCreationDate ()	Funkce vrací datum posledního uložení aktuálního formuláře. Toto datum je uloženo uvnitř každého formuláře a je identifikováno jako „Datum vytvoření formuláře“.	
GetListItemData (<i>listbox</i> , <i>dataXPath</i> , <i>atribut</i> , <i>element</i>)	Funkce vrací datovou hodnotu vybrané položky z vícesloupcového položkového seznamu. První argument je XPath adresa formulářového prvku, který představuje položkový seznam. Druhý argument je relativní XPath adresa datové hodnoty vybrané položky. Třetí, resp. čtvrtý argument specifikuje atribut, resp. element položkového seznamu ze kterého má pocházet navrácená hodnota.	GetListItemData(mesic, //d:mesic, @zkratka) GetListItemData(mesic, //d:mesic, fm:zkratka)

Funkce	Popis	Příklady volání
LastParam (<i>param, param, ...</i>)	Funkce, která umožňuje sekvenčně vyhodnotit skupinu funkcí zadaných jako parametr. Návrátová hodnota funkce LastParam odpovídá návratové hodnotě poslední volané funkce. Argumenty této funkce jsou jednotlivé funkce v pořadí tak, jak mají být vyhodnoceny.	LastParam(DataTransform(rc ...), DataTrasform(pocet_dni, ../d:doba_trvani))
GetNumSigns ()	Funkce vrací počet platných elektronických podpisů v dokumentu	
GetEnvVariable (<i>string</i>)	Funkce vrací textovou hodnotu systémové proměnné specifikované parametrem.	GetEnvVariable(PROGRAMFILES)
GetSignProperty (<i>signIndex, signParam, signOfOID</i>)	Funkce vrací konkrétní informace o elektronických podpisech, které jsou k dokumentu připojeny. Prvním parametrem je index podpisu, ze kterého se informace budou extrahovat. Druhým parametrem je pole certifikátu, ze kterého se budou informace extrahovat. Povolené hodnoty tohoto parametru jsou { SignSubject, SignIssuer, SignNotBefore, SignNotAfter, SignTime, SignComment}. Třetím parametrem je OID označení dané položky (Object identifier).	GetSignProperty(3, SignSubject, 2.5.4.3) GetSignProperty(GetFieldSignerIndex(//*[@form602bind="supervisor"], SignSubject, 2.5.4.3)
DataXPath (<i>string</i>)	Vyhodnotí datový XPath výraz v argumentu.	DataXPath(..d:ks * ../d:cena)
GetDataStr (<i>XPath</i>)	Vrací z formuláře kompletní XML data vyjma datové položky specifikované v atributu.	GetDataStr(@hash)
GetHash (<i>string, method</i>)	Vrací kontrolní součet XML dat ve formuláři. Prvním parametrem jsou XML data, která mají být použita (většinou výsledek funkce GetDataStr). Druhým parametrem je metoda použitá k výpočtu kontrolního součtu (MD5 nebo SHA1)	GetHash(GetDataStr(@hash), MD5)
SignLevel (<i>string</i>)	Vrací TRUE pokud aktuální úroveň parciálního podpisu odpovídá parametru. Jinak vrací FALSE.	SignLevel(1) SignLevel(DataXPath(..d:signs))
PartSignLevel (<i>string</i>)	Vrací TRUE pokud aktuální úroveň parciálního podpisu odpovídá parametru, přičemž ignoruje elektronický podpis návrháře formuláře (PUBLISHER), pokud je ve formuláři obsažen. Jinak vrací FALSE.	PartSignLevel(3)
GetFormParam (<i>XPath</i>)	Vrací hodnotu formulářového parametru podle XPath výrazu specifikovaného parametrem.	GetFormParam(//muj_parametr)
GetNumPages ()	Vrátí hodnotu čísla stránky, na níž vede datová vazba, ve které se volá	

6 Funkce a aplikační události v rozšíření 602XML

Sada rozšíření 602XML neobsahuje jen funkce použitelné ve vlastnostech datových vazeb, ale také funkce použitelné ve formulářové části formuláře. Jedná se např. o funkce volatelné po stisku tlačítka, funkce napojitelné na některé atributy formulářových polí atd. Tyto funkce jsou popsány v následující tabulce.

Funkce	Popis	Příklady volání
GetAttribute (<i>AttrName</i> , <i>FormXPath?</i>)	Funkce vrací hodnotu atributu podřízenému elementu, který je specifikován formulářovým XPath výrazem. POZOR! Jedná se pouze o formulářové prvky ze jmenného prostoru FO!	GetAttribute(form602state,*)
SetAttribute (<i>AttrName</i> , <i>Value</i>)	Funkce nastavuje hodnotu atributu nad nímž je volána na hodnotu určenou druhým parametrem.	SetAttribute(border-style, buttonpushed)
StrCat (<i>string</i> , <i>string</i> , ...)	Funkce spojuje jednotlivé parametry a vytváří jeden řetězec jako návratovou hodnotu.	StrCat(jmeno: , DataXPath(//d:prijmeni))
CallDll (<i>dllName</i> , <i>procName</i> , <i>procType</i> , <i>param</i> , ...)	Umožňuje volat metodu z externí DLL knihovny. První argument je cesta k DLL knihovně, druhý je jméno volané metody. Třetí parametr definuje typ volání metody; může nabývat hodnot typu <i>hwnd_ppStr_XpStr</i> , kde X je počet parametrů (tedy např. <i>hwnd_ppStr_2pStr</i> pro 2 parametrou metodu). Dalšími argumenty funkce jsou pak jednotlivé parametry. Návratová hodnota funkce je typu string.	CallDll(c:_sample\test.dll, _RunFunction@12, hwnd_ppStr_pStr, GetFormDataXML())
SaveData (<i>fileName</i> , <i>user?</i> , <i>password?</i> , <i>proxyuser?</i> , <i>proxypwd?</i>)	Funkce provede export dat a podle parametru <i>fileName</i> exportovaná data zpracuje. Parametr <i>fileName</i> obsahuje cíl, kam mají být data uložena popř. odeslána. Může to být buď cesta k souboru, do kterého mají být data uložena, nebo URL adresa v případě, kdy mají být exportovaná data předána serveru pomocí http požadavku. Parametry <i>user</i> a <i>password</i> specifikují autentifikaci uživatele pro přístup ke složce na HTTP serveru. Parametry <i>proxyuser</i> a <i>proxypwd</i> specifikují autentifikaci uživatele pro přístup přes PROXY server.	SaveData(c:\temp.xml)
Save (<i>fileName?</i>)	Funkce uloží formulář do aktuálního souboru. Pomocí nepovinného parametru je možné definovat alternativní soubor pro uložení formuláře. Parametr <i>fileName</i> obsahuje cestu a název souboru pro uložení formuláře.	Save(c:\temp.fo)
SaveSigned (<i>fileName?</i>)	Funkce uloží formulář s elektronickým podpisem do aktuálního souboru. Pomocí parametru je možné definovat alternativní soubor pro uložení formuláře. Parametr <i>fileName</i> obsahuje cestu a název souboru pro uložení formuláře.	SaveSigned(c:\temp.fo)
CleanForm ()	Funkce provede vyčištění všech formulářových polí a nastavení jejich výchozích hodnot.	
Recalc ()	Funkce provede přepočítání všech automaticky doplňovaných hodnot ve formuláři.	
CheckData ()	Funkce provede kontrolu korektnosti vyplnění všech formulářových polí.	

Funkce	Popis	Příklady volání
If (<i>condition, action1, action2</i>)	Funkce provede příslušnou akci podle logické hodnoty parametru condition. Prvním parametrem je tedy logická hodnota, která bude na počátku vyhodnocena. Parametr action1 reprezentuje akci, která bude provedena, pokud condition je TRUE, parametr action2 reprezentuje akci, která bude provedena, pokud condition je FALSE.	If(DataXPath({../d:polozka1}), spravne, chyba)
IsEmpty (<i>param</i>)	Funkce vrací TRUE v případě, že vstupní parametr není prázdný řetězec, v opačném případě vrací FALSE.	IsEmpty(DataXPath({../d:sest}))
RefreshElements (<i>XPath</i>)	Funkce vyvolá akci grafického přeformátování oblasti dokumentu určené XPath výrazem v parametru funkce.	RefreshElements(//fo:block[6])
SaveAs (<i>filename</i>)	Funkce uloží formulář do souboru a vrací cestu k souboru jako návratovou hodnotu. Parametr filename obsahuje cestu a název souboru.	SaveAs(c:\temp.fo)
SaveAsSigned (<i>filename</i>)	Funkce uloží formulář s elektronickým podpisem do souboru a vrací cestu k souboru jako návratovou hodnotu. Parametr filename obsahuje cestu a název souboru.	SaveAsSigned(c:\temp.fo)
GetTmpFileName (<i>filename?</i>)	Funkce vrátí cestu k adresáři, kam jsou umísťovány dočasné soubory. V případě, že byl jako parametr funkce předán název souboru, bude vrácena cesta včetně názvu dočasného souboru.	
Is_Changed ()	Funkce vrací logickou hodnotu true v případě, že byl dokument změněn, v opačném případě vrací false.	
StrCmp (<i>string, string</i>)	Funkce porovná dva řetězce předané jako parametry a vrací hodnotu TRUE v případě shody, FALSE jinak.	StrCmp(DataXPath(..:jmeno), DataXPath(..:skolitel/d:jmeno))
Stop (<i>state</i>)	Funkce provede break (ukončí vyhodnocování výrazů, např. Spojených pomocí funkce LastParam) v případě, že parametrem break je logická hodnota TRUE.	Stop(DataXPath(//d:bool1))
Not (<i>string</i>)	Logický zápor. Vrací TRUE, pokud je argument roven FALSE, a naopak.	Not(DataXPath(..:d:bool1))
And (<i>string, string</i>)	Logické AND. Vrací TRUE, pokud jsou oba argumenty rovny TRUE, FALSE jinak.	And(DataXPath(..:d:bool1), DataXPath(..:d:bool2))
Or (<i>string, string</i>)	Logické OR. Vrací TRUE, pokud je alespoň jeden argument roven TRUE, FALSE jinak.	Or(DataXPath(..:d:bool1), DataXPath(..:d:bool2))
Transform (<i>stylesheetname, refxpath, mode, modeuri</i>)	Funkce volá transformaci, kdy na jejím vstupu je dokument nebo jeho část určená referenčním uzlem a transformovaný dokument je vrácen jako návratová hodnota. První parametr specifikuje XSL stylesheet, který se má použít, druhý parametr pak referenční XPath výraz pro určení uzlu. Poslední dva parametry specifikují tzv. mode, resp. modeuri parametr XSLT transformace (pokud je v transformaci použit) – více o parametrech mode a modeuri viz. např. http://www.w3.org/TR/xslt#data-model	Trasform(pocet_dni, .)
GetEnvVariable (<i>string</i>)	Funkci vrací hodnotu systémové proměnné specifikované parametrem.	GetEnvVariable(PROGRAM FILES)

Funkce	Popis	Příklady volání
GetSaveFileNameDialog (<i>type?</i> , <i>name?</i>)	Funkce zobrazí systémový dialog pro výběr souboru pro uložení na disk. Návrátová hodnota je cesta a název vybraného souboru a nebo prázdný řetězec v případě, že byl dialog stornován. První parametr nepovinně specifikuje typ souboru pro uložení. Druhý parametr specifikuje jméno.	
LoadFormData (<i>type?</i> , <i>name?</i>)	Funkce importuje do formuláře data zvolená pomocí parametrů. První parametr nepovinně specifikuje typ souboru s daty. Druhý parametr pak specifikuje jméno souboru.	
InsertSection (<i>XPath</i> , <i>data</i>)	Funkce vloží do formuláře další instanci opakovací sekce. Prvním parametrem je formulářový Xpath výraz, který specifikuje pozici opakovací sekce, která má být přidána. Druhý parametr umožňuje automaticky naplnit do sekce data.	InsertSection(//fo:table[5]//fo:table-row[2])
RemoveSection (<i>XPath</i>)	Funkce odebere z formuláře jednu instanci opakovací sekce. Parametrem je Xpath výraz, který určuje pozici sekce.	RemoveSection(//fo:table[5]//fo:table-row[2])
GetOpenFileNameDialog (<i>type?</i> , <i>name?</i>)	Funkce zobrazí systémový dialog pro výběr souboru pro otevření. Návrátová hodnota je cesta a název vybraného souboru a nebo prázdný řetězec v případě, že byl dialog stornován. První parametr nepovinně specifikuje typ souboru pro uložení. Druhý parametr nepovinně specifikuje jméno.	GetOpenFileNameDialog(fo)
Message_Box (<i>message</i> , <i>title</i>)	Funkce zobrazí potvrzovací dialog se zprávou. Při potvrzení dialogu je návratová hodnota funkce true, při stornování dialogu false. Prvním parametrem je text zprávy, druhým parametrem je titulek dialogu.	MessageBox(Ahoj, Zpráva)
QueryTextDialog (<i>Title</i> , <i>Comment</i>)	Funkce zobrazí dotazovací dialog. Jako návratová hodnota je vrácen string vložený do editačního pole dialogu popř. prázdná hodnota v případě stornování dialogu, Prvním parametrem je titulek dialogu, druhým parametrem je text zobrazený v editačním poli.	QueryTextDialog(Dotaz, {Chcete opravdu...})
CurrentUserName ()	Funkce vrací uživatelské jméno přihlášeného uživatele do systému.	
CurrentFileName ()	Funkce vrací název resp. cestu k otevřenému formuláři.	
CurrentTime (<i>format</i>)	Funkce vrací aktuální systémový čas počítače. Pokud je zadán parametr pro formát času, je systémové čas přeformátován do požadovaného tvaru.	CurrentTime(HH:mm)
CurrentTimeUTC (<i>format</i>)	Funkce vrací aktuální UTC čas (Coordinated universal time). Pokud je zadán parametr pro formát času, je systémové čas přeformátován do požadovaného tvaru.	CurrentTimeUTC(HH:mm)
CurrentDate (<i>format</i>)	Funkce vrací aktuální systémové datum počítače. Pokud je zadán parametr pro formát datumu, je systémové datum přeformátováno do požadovaného tvaru.	CurrentDate(ddd)
FormID ()	Funkce vrací ID otevřeného formuláře.	

Funkce	Popis	Příklady volání
FormCreationDate()	Funkce vrací datum posledního uložení aktuálního formuláře. Toto datum je uloženo uvnitř každého formuláře a je identifikováno jako „Datum vytvoření formuláře“.	
SectionNo()	Funkce vrací číslo opakovací sekce, na kterou je její volání navázáno.	
RowNo()	Funkce vrací číslo řádku tabulky, na kterou je její volání navázáno.	
FormatNumb(<i>value</i>, <i>format</i>)	Funkce vrací přeformátovanou hodnotu. Prvním parametrem je původní hodnota. Druhým parametrem je maska, na kterou se má původní hodnota naformátovat!	FormatNumb(DataXPath(//d:vysledek) ,#.00)
IntAdd(<i>int</i>, <i>int</i> (, ...))	Funkce vrací výsledek celočíselného součtu parametrů.	IntAdd(SectionNo(),2)
IntMult(<i>int</i>, <i>int</i> (, ...))	Funkce vrací výsledek celočíselného násobení parametrů.	IntMult(FormID(),3)
CellNo()	Funkce vrací číslo buňky tabulky, na kterou je její volání vázáno.	
GetFormParam(<i>XPath</i>)	Vrací hodnotu formulářového parametru podle XPath výrazu specifikovaného parametrem. Formulářové parametry jsou obsahem elementů/atributů, které jsou uzavřeny v elementu dsig:object[3] v XML struktuře formuláře.	GetFormParam(//muj_parametr)
XpathExpr(<i>param</i>, <i>relat?</i>)	Funkce vyhodnotí XPath výraz a výsledek vrátí jako návratovou hodnotu. Výsledkem vyhodnocení může být textová hodnota (hodnota elementu apod.), uzel (node) popř. sada uzlů (nodeset). V případě, že vyhodnocením výrazu je uzel potažmo sada uzlů je návratová hodnota funkce textový obsah elementů. Při vyhodnocování výrazu je možné zadat referenční uzel, ke kterému bude zadáný XPath vztahován. Prvním parametrem je XPath výraz, který má být vyhodnocen. Druhým parametrem je nepovinná specifikace referenčního uzlu.	XpathExpr(//*['@*='path'])
GetListItemData(<i>listbox</i>, <i>dataXPath</i>, <i>atribut</i>, <i>element</i>)	Funkce vrací datovou hodnotu vybrané položky z vícesloupcového položkového seznamu. První argument je XPath adresa formulářového prvku, který představuje položkový seznam. Druhý argument je relativní XPath adresa datové hodnoty vybrané položky. Třetí, resp. čtvrtý argument specifikuje atribut, resp. element položkového seznamu ze kterého má pocházet navrácená hodnota.	GetListItemData(mesic, //d:mesic, @zkratka) GetListItemData(mesic, //d:mesic, fm:zkratka)
LastParam(<i>param</i>, <i>param</i> (, ...))	Funkce umožňuje zřetězní několika akcí za sebe, např. Po stisknutí tlačítka. Návratová hodnota je identická s návratovou hodnotou poslední akce.	LastParam(InsertSection(), SaveSigned())
GetNumSigns()	Funkce vrací počet platných elektronických podpisů v dokumentu.	

Funkce	Popis	Příklady volání
GetSignProperty (<i>signIndex</i> , <i>signParam</i> , <i>signOfOID</i>)	Funkce vrací konkrétní informace o elektronických podpisech, které jsou k dokumentu připojeny. Prvním parametrem je index podpisu, ze kterého se informace budou extrahovat. Druhým parametrem je pole certifikátu, ze kterého se budou informace extrahovat. Povolené hodnoty tohoto parametru jsou { SignSubject, SignIssuer, SignNotBefore, SignNotAfter, SignTime, SignComment }. Třetím parametrem je OID označení dané položky (Object identifier), resp. formát zobrazení získané hodnoty.	GetSignProperty(3, SignSubject, 2.5.4.3) GetSignProperty(GetFieldSignerIndex(//*[@form602bind="supervizor"]), SignSubject, 2.5.4.3)
GetFormDataXML (<i>sign?</i>)	Funkce vrací XML data z vyplněného formuláře. Data mohou být volitelně podepsána elektronickým podpisem. Nepovinný parametr – v případě, že odpovídá hodnotě „signed“ je po zavolání funkce vyvolán dialog pro elektronický podpis a funkce tak vrátí data XML, která jsou elektronicky podepsána.	
GetFormXML (<i>sign?</i>)	Funkce vrací XML textový zdroj formuláře jako návratovou hodnotu.	
Print ()	Funkce vyvolá dialog pro tisk formuláře.	
GetDataStr (<i>XPath</i>)	Vrací z formuláře kompletní XML data vyjma datové položky specifikované v atributu.	GetDataStr(DataXPath(// @hash))
GetHash (<i>string</i> , <i>method</i>)	Vrací kontrolní součet XML dat ve formuláři. Prvním parametrem jsou XML data, která mají být použita (většinou výsledek funkce GetDataStr). Druhým parametrem je metoda použitá k výpočtu kontrolního součtu (MD5 nebo SHA1).	GetHash(GetDataStr(DataXPath(@hash)), MD5)
ActivateField (<i>XPath</i> , <i>reference?</i>)	Funkce způsobí přidělení focusu formulářovému poli specifikovanému parametrem XPath.	ActivateField(/fo:root/fo:page - sequence[1]/fo:flow[1]/fo:block[4]/fo:inline[1])
SetValueToData (<i>value</i>)	Funkce nastaví datovou hodnotu do datové položky určené parametrem form602bindref, který přísluší formulářovému prvku, na něž je volání této funkce napojeno. Parametre value určuje hodnotu, která bude do dat vložena.	SetValueToData(ano)
GetCurrentDataXPath (<i>bindXPath?</i>)	Funkce vrací datový XPath aktuálního prvku. Pokud je specifikován parametr, pak vrací relativní datový XPath prvku specifikovaného parametrem vzhledem k aktuálnímu prvku.	
SetMultipleDataTo (<i>XPath</i> , <i>data</i> , <i>XPath</i> , <i>data</i> ,...)	Funkce nastaví datové hodnoty do příslušných prvků datové věty určeným pomocí datových XPath výrazů. Je možno specifikovat libovolné množství dvojic parametrů. První z každé dvojice parametrů udává název specifikující položku datové vazby, druhý z každé dvojice parametrů udává hodnotu, která má být do položky vložena.	SetMultipleDataTo(a,1,b,2,c,3)
SetListDataTo (<i>node</i> , <i>bindXPath</i>)	Funkce vloží data z číselníku, který je tvořen položkami datové věty a nad kterým je volána, do datové položky specifikované datovou vazbou. Prvním parametrem je datová položka číselníku, která má být přenesena do datové vazby. Druhým parametrem je XPath výraz datové vazby, do níž má být hodnota vložena.	SetListDataTo(d:ciselnik, jmeno_uzivatele)

Funkce	Popis	Příklady volání
Switch (<i>DataXPath, defvalue, value, switchvalue, value, switchvalue, ...</i>)	Funkce slouží k přepínání hodnot datového switche, nad nímž je volána, na základě jiných hodnot zvolených ve formuláři. Prvním parametrem specifikuje aktuální hodnotu datového přepínače, na základě které se má přepnout hodnota datového přepínače. Druhým parametrem je výchozí hodnota datového přepínače. Dále následují dvojice parametrů specifikující interpretaci jednotlivých hodnot – první parametr z každé dvojice specifikuje původní hodnotu datové položky a druhá hodnota specifikuje hodnotu, jež bude předána datovému přepínači při vybrání této položky.	Switch(DataXPath(.),vychozi moznost,3,c,2,b,1,a)
DataXPath (<i>XPath</i>)	Funkce vrací vyhodnocení výrazu XPath specifikovaného parametrem.	DataXPath(//@hash)
SubmitData (<i>profile</i>)	Funkce slouží k odeslání dat pomocí odesílacího profilu specifikovaného parametrem.	SubmitData(http_send)
GetFieldSignerIndex (<i>fieldXPath</i>)	Funkce vrací index parciálního podpisu formulářové části specifikované XPath výrazem specifikujícím daný formulářový prvek resp. prvky.	GetFieldSignerIndex(//*[@form602bind="admin"])
Round (<i>string, digits</i>)	Funkce vrací zaokrouhlenou vstupní hodnotu. První parametr specifikuje původní hodnotu. Druhý parametr specifikuje počet desetinných míst použitých pro zaokrouhlení.	Round(DataXPath(..:d:cislo), 2)
SendAppEvent (<i>event, param, ...</i>)	Funkce provádí tzv. aplikační události. Každá aplikační událost je specifikována prvním parametrem. Následující argumenty jsou pak vlastními parametry každé události. Podrobný popis aplikačních událostí viz. následující kapitola.	SendAppEvent(printpdf)
PageNo ()	Funkce vrací číslo stránky, na které leží element nad nímž je volána.	
XSLCheck (<i>xsltname</i>)	Funkce provede XSLT transformaci, jejíž jméno je specifikováno prvním parametrem, pomocí kterého se zadává jméno transformace. V případě chyby vrací funkce XPath výraz, jinak vrací prázdný string. Zároveň označí červeně pole s chybou. Pro nastavení fokusu na toto pole je nutno volat ji ve spojení s funkcí ActivateField.	XSLCheck(muj_datacheck)
SystemDataChanged ()	Funkce vyvolává přepočítání těch datových položek, které závisí na psuedovazbě „system“	
TransformDataBranche (<i>stylesheetname, bindname1, bindname2</i>)	Funkce spustí XSLT transformaci specifikovanou prvním parametrem. Tato XSLT transformace je spuštěna nad tou částí datové věty, jež je obsažena v datové položce odpovídající vazbě specifikované v parametru <i>bindname1</i> . Výsledek transformace je poté vložen do části datové věty specifikované datovou vazbou zadanou v parametru <i>bindname2</i> .	TransformDataBranche(moje_transformace, opak_sekce_zadatel, rodne_cislo_seznam)

6.1 Aplikační události rozšíření 602XML

Aplikační události představují speciální skupiny funkcí rozšíření aplikací 602XML. Představují sadu událostí (event), které lze předat aplikaci 602XML Filler, která je poté zpracovává. Výhodou je možnost stornování celé operace v případě jejího stornování uživatelem a další.

Přehled aplikačních událostí rozšíření 602XML je v následující tabulce:

Výraz	Parametry	Akce
SendAppEvent (<i>printpdf</i> , [<i>typedataXPath</i> , <i>ruledataXPath</i> , <i>filedataXPath</i>])	typedataXPath - datový XPath - určuje typ tisku: '0'...etOptPrintAll (výchozí) '1'...etOptPrintNoFields '2'...etOptPrintOnlyFields ruledataXPath - '1' - datový XPath - použití pravidla pro export do pdf (výchozí nastavení je NEPOUŽÍVAT) filedataXPath – datový XPath – pro přímou specifikaci výstupního PDF souboru	Tisknout formulář do PDF
SendAppEvent (<i>print</i> , [<i>typedataXPath</i> , <i>ruledataXPath</i>])	typedataXPath - datový XPath - určuje typ tisku '0'...etOptPrintAll (výchozí) '1'...etOptPrintNoFields '2'...etOptPrintOnlyFields ruledataXPath - '1' - datový XPath - použití pravidla pro tisk (výchozí nastavení je NEPOUŽÍVAT)	Tisknout formulář
SendAppEvent (<i>checkdata</i> , [<i>infoincomplete</i>])	Infoincomplete – nepovinný parametr, pokud je specifikován, dojde při špatném vyplnění formuláře k dotazu, zda se má pokračovat v provádění dalších akcí	Zkontrolovat vyplnění formuláře
SendAppEvent (<i>recalc</i>)		Přepočítat hodnoty ve formuláři
SendAppEvent (<i>submit</i> , <i>submit_name</i>)	submit_name – název odesílacího profilu specifikovaný ve formuláři	Odeslat data
SendAppEvent (<i>submitpart</i> , <i>submit_name</i> , <i>sendDataXPath</i> , <i>reponseDataXPath</i>)	submitName – název odesílacího profilu specifikovaný ve formuláři sendDataXPath - datový XPath, jméno souboru pro typ responseDataXPath – datový XPath, pro uložení odpovědi	Odeslat pouze část datové věty
SendAppEvent (<i>addsect</i> , <i>formXPath</i> , <i>number</i>)	formXPath – formulářový XPath do opakovací sekce number – počet sekcí pro přidání	Přidat opakovací sekci
SendAppEvent (<i>delsect</i> , <i>formXPath</i>)	formXPath – formulářový XPath vedoucí do opakovací sekce	Odstranit opakovací sekci
SendAppEvent (<i>fillsect</i> , <i>formXPath</i> , [<i>fileDataXPath</i> , <i>dlgFormats</i> , <i>xsltName</i>])	formXPath – formulářový XPath pro určení opakovací sekce fileDataXPath – pro získání cesty na soubor s daty dlgFormats - konstanta pro dialog LoadData (volá se, je-li druhý parametr prázdný) 'form', 'xml', 'fo', 'zfo', 'p7m' (výchozí jsou všechny typy) xsltName - jméno transformace k	Naplnit opakovací sekci daty

Výraz	Parametry	Akce
	provedení před načtením	
SendAppEvent (<i>loadformdata</i> , [<i>fileDataXPath</i> , <i>destDataXPath</i> , <i>dlgFormats</i> , <i>xsltName</i>])	fileDataXPath - datový XPath - pro získání cesty na soubor s daty destDataXPath - datový XPath - určuje kam se data nahrávají dlgFormats - konstanta pro dialog LoadData (volá se, je-li fileDataXPath prázdný) 'form', 'xml', 'fo', 'zfo', 'p7m' (výchozí jsou všechny) xsltName - jméno transformace k provedení před načtením	Načíst data do formuláře
SendAppEvent (<i>openform</i> , [<i>fileDataXPath</i>])	fileDataXPath - datový XPath pro získání cesty na soubor k otevření	Otevřít soubor s formulářem
SendAppEvent (<i>saveform</i> , [<i>format</i> , <i>fileDataXPath</i> , <i>dlg</i>])	format - konstanta - formát souboru. Výchozím formátem je 'fo' fileDataXPath - datový XPath - výsledkem je název souboru k uložení dlg - pokud je specifikován tento string, dojde k zobrazení dialogu pro výběr adresáře pro uložení	Uložit formulář
SendAppEvent (<i>saveformsign</i> , [<i>format</i> , <i>fileDataXPath</i> , <i>dlg</i>])	format - konstanta - formát souboru. Výchozím formátem je 'fo'. fileDataXPath - datový XPath - výsledkem je název souboru k uložení dlg - pokud je specifikován tento string, dojde k zobrazení dialogu pro výběr adresáře pro uložení	Uložit formulář s elektronickým podpisem
SendAppEvent (<i>closeapp</i> , [<i>closeType</i>])	closeType - konstanta; je-li '0', pak nedojde k dotazu na uložení	Zavřít aplikaci
SendAppEvent (<i>openbindata</i> , <i>bindName</i>)	bindName - název datové vazby s binární přílohou !POZOR! Pro otevření souboru z přílohy je nutné, aby v datové větě formuláře byly přítomny i prvky obsahující název a příponu souboru přílohy!	Otevřít soubor z přílohy
SendAppEvent (<i>setdato</i> , <i>destFormXPath</i> , <i>string</i>)	destFormXPath - formulářový XPath pro určení cílového pole string - data, která se do pole nastaví	Nastavit hodnotu do formulářového prvku
SendAppEvent (<i>setevaldata</i> , <i>bindName</i> , <i>destDataXPath</i> , <i>XPathexpr</i>)	bindName - jméno vazby do dat destDataXPath - ABSOLUTNÍ datový XPath (cíl) XPathexpr - data, která se do pole nastaví, přičemž je zde možno provést vyhodnocení výrazu	Nastavit hodnotu do prvku datové věty na základě vyhodnocení XPath výrazu
SendAppEvent (<i>setdatavalue</i> , <i>bindName</i> , <i>destDataXPath</i> , <i>string</i>)	bindName - jméno vazby do dat destDataXPath - ABSOLUTNÍ datový XPath (cíl) string - data, která se do pole nastaví	Nastavit hodnotu do prvku datové věty
SendAppEvent (<i>messagebox</i> , <i>string</i> , [<i>dlgType</i>])	string - zobrazený text dlgType - počet tlačítek v MB (default='1')	Zobrazit dialog se zprávou
SendAppEvent (<i>gotofield</i> , <i>formXPath</i>)	formXPath - formulářový XPath určující, kam se posunout	Přejít na formulářové pole
SendAppEvent (<i>startfiller</i> , <i>fileDataXPath</i> , <i>xmldataDataXPath</i>)	fileDataXPath - datový XPath výraz specifikující formulář, který má být v nové	Otevřít novou instanci 602XML Filler

Výraz	Parametry	Akce
	instanci aplikaci načten xmldataDataXPath – datový XPath výraz specifikující soubor s XML daty, která se mají naimportovat do formuláře	
SendAppEvent(formtohtml)		Provést tisk formuláře jako HTML stránky
SendAppEvent(printparam, [printerName, FromPage, ToPage, Copies])	printerName – nepovinný parametr, specifikuje výběr tiskárny FromPage – nepovinný parametr, specifikuje počáteční stranu pro tisk ToPage – nepovinný parametr, specifikuje poslední stranu pro tisk Copies – nepovinný parametr, specifikuje počet vytisknutých kopií	Provést přímý tisk formuláře (bez nutnosti výběru tiskárny)
SendAppEvent(openfiller, fileDataXPath, xmldataDataXPath)	fileDataXPath – datový XPath výraz specifikující formulář, který má být v nové instanci aplikaci načten xmldataDataXPath – datový XPath výraz specifikující soubor s XML daty, která se mají naimportovat do formuláře	Otevřít novou instanci 602XML Filler v modálním okně.
SendAppEvent(loaddata, submit_name)	submit_name – název odesílacího profilu specifikovaný ve formuláři	Načíst data do formuláře pomocí daného profilu
SendAppEvent(signclick, constant, string, size)	constant – konstanta, jedná se o řetězec GetAttribute(form602state) jež je vyhodnocen pomocí XMLcore string – specifikuje sekvenci akcí prováděných, pokud je konstanta „none“ size – udává velikost použité ikony	Slouží k podpoře elementu pečeť pro parciální podepisování formuláře
SendAppEvent(csvtoxml, fileDataXPath1, fileDataXPath2)	fileDataXPath1 – cesta ke vstupnímu souboru fileDataXPath2 – cesta k výstupnímu souboru	Převést vstupní CSV soubor do formátu XML
SendAppEvent(openfillerfrombuffer, foxslbuffer, xmlbuffer, [readonly])	foxslbuffer – buffer s XSL-FO souborem xmlbuffer – buffer s XML daty readonly – nepovinný parametr, specifikuje, zda je zobrazený formulář editovatelný či ne	Otevřít novou instanci 602XML Filler pouze s částí formuláře danou vstupními soubory
SendAppEvent(printpreview)		Zobrazit náhled tisku s formulářem
SendAppEvent(deletefile, fileDataXPath)	fileDataXPath – datový XPath výraz specifikující soubor, který má být smazán	Smazat zadaný formulář
SendAppEvent(savebindata, bindname, [fileDataXPath, expDataXPath])	bindname – název vazby s BASE64 daty fileDataXPath – datový XPath výraz, nepovinně specifikuje soubor, do kterého mají být data zapsána expDataXPath – datový XPath výraz, nepovinně specifikuje příponu souboru do kterého mají být data vloženy.	Naplnit automaticky datový soubor binárními daty
SendAppEvent(openlink, linkdataXPath)	linkdataXPath – datový XPath výraz specifikující URL nebo lokální adresu cíle	Přejít na adresu odkazu
SendAppEvent(viewdata, relevance)	relevance – nepovinný parametr, pokud je roven nule, budou zobrazena i nerelevantní data	Zobrazit XML data
SendAppEvent(savebase64datatofile, binDataXPath, fileDataXPath, extDataXPath)	binDataXPath – datový XPath specifikující položku obsahující binární data	Uložit binární data uložené v položce datové věty do

Výraz	Parametry	Akce
	<p>fileDataXPath – nepovinný parametr specifikující jméno cílového souboru. Pokud není uveden, otevře se SaveAs dialog.</p> <p>extDataXPath – nepovinný parametr umožňující nastavení povolených typů souborů pro uložení pomocí SaveAs dialogu.</p>	souboru

7 Použití XPath v aplikacích 602XML

XPath výrazy (jak obecně definované normou, tak přístupné díky rozšíření 602XML) je možno používat v aplikacích 602XML k mnoha úkonům a na mnoha místech. Patří mezi ně:

- Vlastnosti datových vazeb nastavované v 602XML Designer - ve většině případů (kromě pole **Výpočet**) je předpokládán **logický výraz**, tj. takový, který vrací hodnotu TRUE nebo FALSE, podle ní je daná vlastnost aktivní či nikoliv. V poli **Výpočet** se uvádí **obecný XPath výraz**, který bude vyhodnocen a hodnota následně přenesena do pole. Výpočetní model provádí operace nad datovou větou formuláře, tzn. operandem logického i obecného XPath výrazu může být libovolná část datové věty formuláře. Pokud v poli **Výpočet** bude vložen logický XPath výraz, tj. vracející hodnotu TRUE nebo FALSE, bude příslušné pole nabývat právě jedné z uvedených hodnot.
- Napojování akcí a metod, které využívají XPath výrazy, na formulářová tlačítka.
- Nastavení hodnot některých atributů formulářových polí
- Výběr formátovacích elementů grafické podoby formuláře pomocí zapsání XPath cesty na spodní lištu aplikace 602XML Designer
- Výběr položek datové věty pomocí zapsání absolutní XPath cesty na lištu pod panelem s datovou větou formuláře

8 Literatura

- [1] XML Path Language (XPath)
Version 1.0
W3C Recommendation 16 November 1999
<http://www.w3.org/TR/xpath>
- [2] XPath Tutorial and Samples
<http://www.w3schools.com/xpath/default.asp>
- [3] Web Software602 pro vývojáře
<http://www.602.cz/dev>