

# 602XML Form Server

## Příručka návrháře formulářů

<b>Úvod .....</b>	<b>2</b>
602XML Designer .....	2
602XML Filler .....	2
602XML Form Server .....	2
<b>Princip práce s formulářem .....</b>	<b>2</b>
Návrh formuláře .....	2
Instalace formuláře do Form Serveru.....	2
Stažení formuláře uživatelem .....	2
Odeslání vyplněného formuláře do Form Serveru .....	2
Postup zpracování formuláře Form Serverem.....	3
<b>Obsah datové věty 602XML Form Serveru .....</b>	<b>3</b>
Data přihlášeného uživatele .....	3
Informace o podepisujících.....	4
Informace o formuláři.....	4
Nastavení lhůt pro vyřízení formuláře .....	5
<b>URL pro odeslání formuláře do Form Serveru .....</b>	<b>5</b>
Odesílací URL .....	5
Odeslání pomocí tlačítka ve formuláři .....	6
<b>Koloběh XML formulářů .....</b>	<b>9</b>
Obecně.....	9
Datová věta .....	10
Co čte 602XML Form Server z XML dat .....	14
<b>Úpravy formuláře pro koloběh dokumentů .....</b>	<b>14</b>
Informace o podepisujícím uživateli.....	14
Datový číselník .....	15
Zobrazení podpisu.....	17
Vložení a podepsání dat na základě relevance.....	18
Přepínací sekce pro podpisy.....	18
Přepínací sekce pro odeslání schváleného nebo zamítnutého formuláře do archívu .....	19
Přepínací sekce postoupení předvyplněného formuláře prvnímu podepisujícímu .....	19
<b>Možnosti ladění funkce formuláře v 602XML Form Serveru .....</b>	<b>20</b>
Formulář.....	20
602XML Filler .....	20
Form Server.....	21
<b>Appendix A - Linky .....</b>	<b>21</b>

Verze dokumentu z 25.5.2010

## Úvod

Tato příručka je určena pro návrháře 602XML formulářů spolupracujících s programem 602XML Form Server a zabývá se zejména návrhem formulářů s podepisovací cestou. Pro návrh formuláře se používá aplikace 602XML Designer. Pro další text předpokládáme alespoň základní znalost aplikací 602XML Designer, 602XML Filler a 602XML Form Server.

### **602XML Designer**

Aplikace 602XML Designer je určena pro vytváření a úpravy formulářů ve WYSIWYG prostředí. Umožňuje připojení datového zdroje se strukturou XML dat či návrh od ruky, komfortní formátování formuláře, vkládání formulářových prvků a opakovacích sekcí, připojování číselníků a volbu jejich vlastností i vkládání XSL transformací. Návrhář rovněž nastavuje zabezpečení formuláře a určuje pravidla pro podání vyplněného formuláře pomocí aplikace 602XML Filler.

### **602XML Filler**

Aplikace 602XML Filler se nainstaluje na stanici uživatele a slouží pro vyplnění formuláře a jeho odeslání do Form Serveru.

### **602XML Form Server**

602XML Form Server je formulářový server využívající technologie 602XML pro zajištění rychlého a snadného sběru dat a koloběhu formulářů. Je ideálním nástrojem pro centrální správu formulářů, jejich následnou archivaci či uložení rozpracovaných formulářů s případnou informací o průběhu schvalovacích procesů.

## Princip práce s formulářem

Zde uvedme základní postup práce s formulářem v prostředí 602XML a Form Serveru:

### **Návrh formuláře**

Návrhář navrhne v aplikaci 602XML Designer vzhled formuláře, který přiváže vazbami k datové větě. Datová věta bude obsahovat elementy a atributy pro data vyplňovaná do formuláře a také elementy a atributy nutné pro komunikaci s Form Serverem. Na závěr 602XML Designer formulář publikuje, tedy především data uloží do souboru s koncovkou FO nebo ZFO ve formátu XML-FO a formulář elektronicky podepíše.

### **Instalace formuláře do Form Serveru**

Administrátor nainstaluje formulář do Form Serveru (formulář se zapíše do tabulek serveru, u formuláře pro koloběh se definují pravidla podepisovací cesty). Po uvolnění formuláře administrátorem ho uživatelé vidí ve složce a mohou s ním dále pracovat.

### **Stažení formuláře uživatelem**

Uživatel si formulář kliknutím v internetovém prohlížeči otevře v aplikaci 602XML Filler. Před stažením formuláře Form Server předvyplní jemu známé elementy a atributy, tedy např. informace o uživateli, který stahuje formulář, URL adresu kam se bude formulář po vyplnění odesílat a řadu dalších hodnot. Tyto informace mohou sloužit pro řízení vzhledu a jeho dalšího oběhu.

### **Odeslání vyplněného formuláře do Form Serveru**

Uživatel vyplní formulář Fillerem a protokolem http odešle zpět do Form Serveru na jednu z předvyplněných URL (url\_release, url\_forward atd., viz níže).

## Postup zpracování formuláře Form Serverem

- zpracování formuláře probíhá dvěma různými způsoby:

1. formulář určený jen pro sběr dat (nemá žádné podepisovací kroky), po vyplnění uživatelem a odeslání do Form Serveru se data uloží do databáze (u anonymního může odsouhlasit arbitr)

2. formulář pro koloběh má podepisovací kroky, data se standardně ukládají do databáze až když formulář projde úspěšným schvalovacím procesem.

## Obsah datové větvy 602XML Form Serveru

V této kapitole si ukážeme, jak navrhovat formuláře pro použití s programem 602XML Form Server. Nejprve si vyjmenujeme, které údaje Form Server předvyplňuje do formuláře před jeho stažením k uživateli, tj. pokud tyto elementy nalezne v datové větvě, předvyplní je hodnotami.

### Data přihlášeného uživatele

602XML Form Server předvyplní data uživatele, který formulář vyplnil nebo nyní vyplní:

<code>gw_salutation</code>	Oslovení
<code>gw_pre_title</code>	Titul před jménem
<code>gw_name</code>	Jméno
<code>gw_middle_name</code>	Druhé jméno
<code>gw_surname</code>	Příjmení
<code>gw_birth_name</code>	Rodné jméno
<code>gw_post_title</code>	Titul za jménem
<code>gw_fullname</code>	= „surname name pre_title (job)”
<code>gw_id_user</code>	ID uživatele (definuje systém)
<code>gw_602_form_identity</code>	ID z tabulky USERS ve FS (definuje systém)
<code>gw_uid</code>	ID uživatele z FS (nastavuje administrátor)
<code>gw_birth_date</code>	Datum narození
<code>gw_id_card_number</code>	Číslo OP
<code>gw_pin</code>	Rodné číslo
<code>gw_email</code>	E-mail
<code>gw_phone_number</code>	Telefon
<code>gw_mobile_number</code>	Mobilní telefon
<code>gw_fax_number</code>	Fax
<code>gw_web</code>	Webová adresa
<code>gw_job</code>	Funkce, zařazení (např. „Vedoucí IT“)
<code>gw_company</code>	Jméno firmy
<code>gw_department</code>	Odbor, oddělení (např. „Odbor vnitřních záležitostí“)
<code>gw_section</code>	Sekce (další úroveň pod oddělením)
<code>gw_role</code>	Role uživatele
<code>gw_note</code>	Poznámka s libovolnou informací o uživateli
<code>gw_street</code>	Ulice
<code>gw_street_no</code>	Číslo popisné nebo orientační
<code>gw_orientation_no</code>	Číslo orientační

<code>gw_city</code>	Obec
<code>gw_city_district</code>	Část obce
<code>gw_zip_code</code>	PSČ
<code>gw_state</code>	Stát
<code>gw_country</code>	Země
<code>gw_message</code>	Text zprávy od Form Serveru nebo předchozího podepisujícího

### **Informace o podepisujících**

Bližší vysvětlení a seznam odpovídajících atributů je v následující kapitole.

<code>gw_signX</code>	Do atributů toho elementu vloží Form Server informace o předpokládaných podepisujících
<code>gw_fwdX</code>	Do atributů toho elementu vloží Form Server informace o příštím příjemci formuláře
<code>gw_grp_fwdX</code>	Informace umožňující zřízení číselníku pro výběr příštího příjemce formuláře
<code>gw_current_user</code>	Informace o přihlášeném uživateli

### **Informace o formuláři**

Bližší vysvětlení a seznam odpovídajících atributů je v následující kapitole.

<code>gw_sn</code>	Sériové číslo vyplňovaného formuláře generované FS dle zadaných pravidel při instalaci formuláře (dle nastavení ve FS se přiděluje buďto při prvním stažení formuláře nebo po jeho definitivním uložení)
<code>gw_form_name</code>	Změna názvu formuláře s vyplněnými daty – platí pro formuláře uložené v sekci „Rozpracované“ a „Archiv“. Nový název se též fulltextově zaindexuje a lze tak podle něj vyplněný formulář vyhledat.
<code>gw_form_code</code>	Kód formuláře
<code>gw_id_ufrm</code>	Vnitřní ID formuláře generuje Form Server pro jednoznačnou identifikaci formuláře v rámci celého Form Serveru
<code>gw_document_id</code>	Obsahuje identifikátor anonymem vyplněného formuláře. Používá se pro jeho další vyhledání úředníkem.
<code>gw_status</code>	Průběh schvalovacího procesu formuláře (0-v procesu, 1-schválen, 2-zamítnut, 3-v prvním kroku při vyplňování anonymem nebo autorem)
<code>gw_form_status_code</code>	Obsahuje kód aktuálního stavu formuláře (stavy definované administrátorem a nastavené uživatelem nebo Form Serverem)
<code>gw_602_form_identity</code>	Identifikace daného vyplňovaného formuláře, kterou vytváří Form Server a používá ji k detekci změn formuláře, které mohly nastat mezi odesláním formuláře do Filleru a příjmem XML dat zpět.
<code>gw_archive</code>	Příznak o uložení formuláře do archívu
<code>gw_numsigns</code>	Aktuální počet uživatelských podpisů ve formuláři
<code>gw_notify</code>	Příznak zajišťující posílání informačních emailů autorovi formuláře o každém kroku
<code>gw_accept</code>	Pokud je nastaven na 1, Form Server vyhodnotí jako schválení schvalovacího formuláře ještě dříve, než jej podepíše poslední schvalující.
<code>gw_refuse</code>	Pokud je nastaven na 1, Form Server vyhodnotí jako zamítnutí

## schvalovacího formuláře

**Nastavení lhůt pro vyřízení formuláře**

Ve formuláři je možné změnit nastavení avizování lhůty k vyřízení tohoto formuláře. Prázdná hodnota klíče znamená, že se hodnota nemá měnit, text NULL znamená, že se v datázi nastaví hodnota NULL. Časová jednotka údaje se pro danou šablonu formuláře nastavuje v uživatelském prostředí, implicitně jsou to "dny", druhá možnost je "hodiny". V názvu klíče je stále "days" kvůli kompatibilitě se staršími verzemi Form Serveru.

<code>gw_processing_max_days</code>	Maximální počet dnů k vyřízení formuláře
<code>gw_reminder_days</code>	Termín avíza
<code>gw_reminder_frequency_days</code>	Interval odesílání avíza

**URL pro odeslání formuláře do Form Serveru**

Formulář je třeba z aplikace 602XML Filler do Form Serveru odeslat na některou z URL specifikovaných Form Serverem v attributech URL\_xxx.

**Odesílací URL**

Odesílací URL jsou součástí atributů kořenového elementu:

```
<root url_send="..." url_release="...">
```

Tyto URL se obvykle používají v odesílacím profilu formuláře. Použitá URL částečně definuje další postup zpracování formuláře Form Serverem.

- url\_send** Uloží částečně vyplněný XML formulář do 602XML Form Server, později lze formulář stáhnout a dovyplnit. Takto uložený formulář se objeví ve složce „Částečně vyplněné“. XML data zatím nejsou uložena do databázových tabulek. Odesílá se formulář bez podpisu.
- url\_release** XML data z formuláře jsou definitivně uložena do databázových tabulek a ve webovém rozhraní se vyplněný formulář objeví ve složce „Archív“. Odesíláme jen XML data a zaškrťovací pole „Odeslat relevantní data“ nesmí být zaškrtnuté. Neboli odesíláme nerelevantní data, aby se vše z formuláře uložilo do databázových tabulek serveru.
- url\_forward** Postoupení formuláře dalšímu podepisujícímu k podpisu (viz následující kapitola). V závislosti na situaci se posílá podepsaný nebo nepodepsaný formulář.
- url\_back** Vrácení formuláře předchozímu podepisujícímu k přepracování (viz následující kapitola). Odesílá se formulář bez podpisu, Form Server naopak předchozí podpis odstraní.
- url\_base** Pouze základní část URL, používá se při volání uživatelských pluginů Form Serveru.

**Anonymní uživatel** nemá nastaven atribut `url_send`, protože nemůže ukládat rozpracované formuláře. Nemá také nastavený atribut `url_back`, protože může být

pouze prvním podepisujícím. Pokud odešle formulář na `url_release`, může formulář v závislosti na konfiguraci před archivací čekat na schválení arbitrem.

### **Odeslání pomocí tlačítka ve formuláři**

Odeslání formuláře nebo vyplněných dat do Form Serveru se obvykle provede zmáčknutím tlačítka ve formuláři, které vyvolá aplikační událost programu Filler (SendAppEvent). Pomocí parametrů ve zvoleném odesílacím profilu formuláře dojde k odeslání formuláře nebo dat do Form Serveru pomocí protokolu http.

### **Profil**

V odesílacím profilu formuláře se URL obvykle definuje pomocí výrazu `$(Eval())`, kde je argumentem buď přímo atribut předvyplněný Form Serverem:

Příklad: `$(Eval (//@url_forward))`

nebo předvyplněný atribut rozšířený pomocí funkce `concat` (viz dokument o XPATH výrazech, funkcích a aplikačních událostech) o další řetězec:

Příklad: `$(Eval (concat (//@url_send, '&to_tables=true')))`

Výjimku z tohoto postupu tvoří formulář připravený pro anonymního uživatele šířený jinými prostředky než stahováním z Form Serveru, kdy se do URL zapíše konkrétní URL zkopírováním z odpovídající řádky ve vlastnostech již zavedeného formuláře.

Obsahem připojovaných řetězců jsou obvykle tyto parametry:

### **Předčasné zpracování dat do tabulek**

v odesílacím profilu se za URL přidá parametr `to_tables`

Např.: `$(Eval(concat(//@url_send, '&to_tables=true')))`

(použit lze pouze v souvislosti s `url_send`)

Parametr `to_tables` způsobí, že se po přijetí dat Form Serverem provede uložení hodnot do uživatelských SQL tabulek.

### **Předčasné zpracování dat do archívu**

v odesílacím profilu se za URL přidá parametr `to_archive`

Např.: `$(Eval(concat(//@url_forward, '&to_archive=true')))`

(je třeba napřed zajistit, aby data již byla v uživatelských SQL tabulkách, viz. výše. Použit lze pouze v souvislosti s `url_forward`)

### **Volba typu odpovědi Form Serveru po přijetí dat**

Funkce: Po odeslání vyplněného formuláře na URL doplněnou o volání `answer=<param>` vrátí Form Server odpověď v požadovaném tvaru.

Obecně může mít proměnná `answer` hodnoty:

=**info** - implicitní, Form Server vrací textovou zprávu do okna Filleru

=**xml** - Form Server vrátí celou datovou větu, Filler jí načte a data vloží do zobrazeného formuláře - načtení vrácených XML dat do formuláře musí být nastaveno v odesílacím profilu.

=**form** - Form Server vrátí do Filleru kompletní formulář, což má význam především při odesílání podepsaného formuláře, protože je tím umožněno uživateli zobrazit jím odeslaný již podepsaný formulář k dalšímu zpracování, např. odeslání emailem, tisku atd.

Proměnná answer se týká jen odesílacích profilů url\_release a url\_forward u postoupení předvyplněného formuláře prvním podepisujícím.

Příklad použití v URL: `$(Eval(concat(//@url_release,'&answer=xml')))`

## Posílání a postupování formuláře v koloběhu

Obecně platí: v prvním kroku je další předvyplnitel nebo první podepisující uveden v gw\_fwd1, podepisující pro druhý krok je v gw\_fwd2. Formulář může ponechat tyto elementy beze změny, tady tak, jak je vyplnil Form Server, nebo může jejich hodnotu měnit podle nabídnutých hodnot v gw\_grp\_fwdX nebo do nich může zapsat ID jiného existujícího uživatele.

1) **Základní koloběh** s předvyplňujícími a podepisujícími uvedenými v pravidlech jednotlivých kroků.

Odesílá se na URL: `$(Eval(//@url_forward))`

- když se v prvním kroku pošle na url\_forward celý podepsaný formulář, pošle se uživateli uvedenému v gw\_fwd2. Obsah gw\_fwd2 se může ve formuláři měnit podle uživatelů uvedených v použitém pravidlu pro krok 2 (tedy ve struktuře/číselníku gw\_grp\_fwd2)

- když se v prvním kroku pošle na url\_forward celý nepodepsaný formulář, pošle se podle pravidel na gw\_fwd1. To znamená, že (bez zásahu formuláře do hodnoty v gw\_fwd1) předvyplňující posílá formulář prvním podepisujícímu, resp. první podepisující vrácí nepodepsaný formulář předvyplňujícímu.

Pokud se odešle formulář jinému uživateli, pro kterého tedy Form Server podle pravidel nenajde novou hodnotu pro gw\_fwd1, nedovolí Form Server takovému uživateli formulář stáhnout.

Z toho plyne, že i pro uživatele, u kterého při návrhu koloběhu předpokládáme, že formulář podepíše, musíme založit pravidlo, aby např. mohl formulář postoupit bez podpisu např. sám sobě, jinak mu Form Server nedovolí formulář stáhnout k vyplnění.

- postoupit tedy lze uživatelům uvedeným v pravidlech tohoto kroku. Výjimku tvoří situace, kde se formulář do daného kroku dostal přeskočením podepisovacích kroků, pak musí být příjemce postupovaného formuláře uveden mezi příjemci v přeskokovacích pravidlech.

2) **Koloběh s forward=true** (od verze 2.5.3)

Při tomto typu postoupení celého nepodepsaného formuláře dojde v prvním kroku k běžnému postoupení jinému uživateli podle proměnné gw\_fwd1. Příjemce musí být uveden v pravidlech prvního kroku. Při druhém postoupení v prvním kroku tímto způsobem tedy nedojde k automatickému návratu autorovi.

Odesílá se typicky na URL sestavenou takto:

```
$(Eval(concat(//@url_forward,'&forward=true')))
```

V dalších krocích nemá parametr forward=true žádný význam, provede se postoupení stejným způsobem jako bez tohoto parametru.

Pozn1.: První uživatel, který stáhne formulář, nemusí mít vyplněné gw\_fwd2, tedy nenajde se žádný příjemce podle pravidel pro druhý krok. Kdyby pak odeslal podepsaný formulář do 2.kroku, ohlásí se chyba. Podstatné je, že se mu podaří formulář stáhnout k vyplnění.

### 3) Opakované postupování podepsaného formuláře stále v prvním kroku

Od verze 2.5.5 přibyla možnost postupování formuláře v prvním kroku s podepisováním. Cílem tohoto postupu je umožnit podepisování formuláře předem neznámým počtem podepisujících nebo velkým počtem podepisujících, kdy není možné/reálné vydefinovat podpisové kroky standardním způsobem.

K tomuto účelu byly doplněny do URL klíče forward=sign a forward=removesign, které tvoří spolu s forward=true a standardním odesláním na url\_forward kompletní sadu nástrojů.

Typy odeslání formuláře na URL předvyplněnou v elementu url\_forward:

bez parametru	s podpisem	přechod do dalšího kroku
bez parametru použití je zpět k předvyplňujícímu	bez podpisu	postoupení mezi předvypl. a podep. v 1.kroku, druhé
forward=true	bez podpisu	běžné postoupení jinému uživateli
forward=sign	s podpisem	postoupení dalšímu stále v prvním kroku
forward=removesign	bez podpisu	odejme podpis a vrátí předchozímu podep. (pouze v rámci 1.kroku)

Vždy se odesílá "kompletní formulář včetně vyplněných dat".

V rámci prvního kroku se formulář postoupí uživateli, jehož ID je v elementu gw\_fwd1, do druhého kroku uživateli, jehož ID je v elementu gw\_fwd2 atd.

Upozorníme, že uživatelské podpisy vložené do formuláře v rámci postupování formuláře v prvním kroku se NEZAPOČÍTÁVAJÍ do počtů podpisů v rámci standardních podepisovacích kroků (gw\_numsigns, sign\_level atd.)

### Použití přeskokovacích pravidel v koloběhu formuláře

Přeskočit podepisovací krok lze až s podepsaným formulářem, tj. ve všech podepisovacích krocích kromě prvního. Po podpisu odešle uživatel (podepisující) z aplikace 602XML Filler podepsaný XSL-FO formulář na URL url\_forward, do něhož doplní URL proměnné **rule\_set** s názvem přeskokovacího pravidla a **jump\_to** s nominálním číslem podepisovacího kroku, do kterého se má přeskočit. Např.

```
$(Eval(concat(//@url_forward,'&jump_to=4&rule_set=jump_2_4')))
```

přeskočí do čtvrtého kroku (jump\_to=4) a použije přeskokovací pravidlo se jménem "jump\_2\_4".

Přeskakování podepisovacích kroků je implementováno až ve Form Serveru od verze 2.5.3. Bližší popis najdete v nápovědě k Form Serveru, viz [www.602.cz](http://www.602.cz).

### Tlačítko a aplikační událost

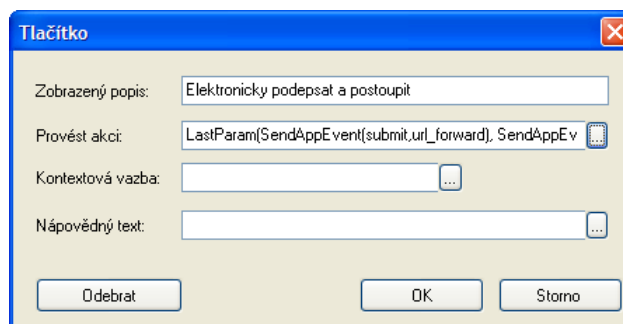
Tlačítko pro odeslání může obsahovat jednoduché volání akce programu Filler:

Příklad: `SendAppEvent (submit, profil_forward)`

nebo skupiny akcí:

Příklad - po odeslání do Form Serveru zavřít aplikaci Filler:

`LastParam (SendAppEvent (submit, profil_forward) , SendAppEvent (closeapp) )`



## Koloběh XML formulářů

### Obecně

Koloběh formulářů lze rozdělit dle praxe na dvě úlohy:

**Předem stanovená trasa schvalování (podepisování) formuláře** – u formuláře je předem pevně stanoveno, kdo jej podepisuje v druhém, třetím atd. kroku.

Množina uživatelů podepisujících formulář je nastavena v pravidlech na serveru a uživatelé během podepisování formuláře tuto cestu nemohou měnit.

**Trasa s volbou dalšího podepisujícího** - v každém kroku podepisování může být na serveru nastavena množina uživatelů, takže podepisující uživatel si může vybrat z datového číselníku komu formulář postoupí k následujícímu podepisovacímu kroku.

Formulář nabývá během schvalovacího procesu následujících stavů:

1. Formulář je ve schvalování, ještě nemá všechny podpisy.
2. Formulář byl zamítnut (libovolný účastník jej může během schvalovacího procesu zamítnout).
3. Formulář je schválený (podepsal jej poslední podepisující).

Formuláře se během schvalování mohou vyskytovat v následujících složkách:

- **Moje k vyřízení** – V této složce se zobrazují formuláře, které má uživatel právě podepsat nebo jinak zpracovat. Současně zde vidí i své rozpracované a dočasně uložené formuláře. Nadřízení (vedoucí skupin) zde vidí pouze formuláře, které mají podepsat, nevidí formuláře k podpisu svých podřízených.
- **V procesu** – V této složce se uživatelům zobrazují všechny rozpracované formuláře, ke kterým má přístupové právo. Zeleně podbarvené jsou ty, které čekají na jeho zpracování, žlutě podbarvené jsou ty, které může převzít, ostatní jsou podbarveny standardně bíle.

- **Archív** – Do archívu se dostane formulář poté, kdy byla jeho XML data uložena do databázových tabulek 602XML Form Serveru. V případě formulářů určených pro sběr dat, dochází k ukládání do archívu po správném vyplnění formuláře a odeslání dat. Formuláře, které musí projít schvalovacím procesem, se archivují po finálním schválení nebo zamítnutí formuláře. Vyplněný formulář v archívu vidí ti, kteří jej podepsali, a ti, kteří jej do archívu ukládali. Nadřízení vidí formuláře svých podřízených.

Podle výše uvedených pravidel vidí formulář účastník schvalovacího procesu. Dále může existovat tzv. Správce skupiny, který vidí formuláře všech uživatelů ze své skupiny. Administrátor 602XML Form Serveru vidí vše.

V 602XML Form Serveru je u každého formuláře k dispozici:

- Popis, čeho se formulář týká resp. co se v něm schvaluje.
- Seznam uživatelů, kteří se účastní schvalovacího procesu, s poznámkou u těch, kteří jej již schvalovali.
- Kdo jej odeslal, kdo jej právě vyřizuje a kdo jej vrátil.
- Zpráva přečtená z libovolného XML elementu datové větvy použita pro informaci při postupování nebo vracení formuláře.

Stáhnout formulář a zobrazit si jej v aplikaci 602XML Filler může každý podle svých přístupových práv, ale odeslat jej s podpisem do systému může jen ten, kdo je právě na řadě podle pravidel schvalovacího procesu.

### Postupování a převzetí formulářů

602XML Form Server umožňuje uživateli **postoupit** nepodepsaný formulář jinému uživateli, který má právo jej „vhodit“ do schvalovacího procesu (podepisovat jako první). Tato funkcionality slouží k tomu, aby jeden uživatel mohl založit formulář, předvyplnit jej daty a poté postoupit jinému uživateli k podpisu. Od Form Serveru verze 2.5.3 je možné postoupit formulář ve všech podepisovacích krocích. **Převzetí** formuláře uživatelem je proces, kdy uživatel může převzít formulář čekající na zpracování jiným uživatelem a sám ho podepsat a tím pokračovat ve schvalovacím koloběhu.

### Zastupování při podpisu formuláře

602XML Form Server řeší problematiku zastupování v případě nepřítomnosti uživatele. U každého uživatele se ukládá informace o době jeho nepřítomnosti.

Pro celý formulář nebo pro každý podepisovací krok zvlášť může být nastaven k uživateli jeho zástupce v případě nepřítomnosti. Pokud Form Server zjistí, že uživatel, kterému postupuje formulář k podpisu, není přítomen, pokusí se najít zástupce. Pokud se mu to podaří, uvidí zástupce ve složce k podpisu formulář, kterým se má zabývat. V případě neúspěchu (zástupce není definován), formulář pošle do složky k podpisu původnímu uživateli. Při ukončení nepřítomnosti uživatele přesune Form Server formuláře dosud nezpracované zástupcem zpět původnímu uživateli.

### Datová věta

Koloběh formulářů pracuje s libovolným formulářem, kterému je nutné upravit datovou větu. Ta bude obsahovat níže uvedené tagy a atributy:

```
<root url_send="" url_forward="" url_back="" url_release="">
  <gw_numsigs></gw_numsigs>
  <gw_status></gw_status>
  <gw_notify></gw_notify>
  <gw_accept></gw_accept>
  <gw_refuse></gw_refuse>
  <gw_message></xg_message>
  <gw_signX sign_level="a" signed="b"
```

```

pre_title="" name="" surname="" post_title="" email=""
phone_number="" mobil_number="" fax_number="" street=""
street_no="" zip="" city="" job="" department="" section=""
note="" />
<gw_fwdX>25</gw_fwdX>
<gw_grp_fwdX>
  <gw_fwd_itemX id="" fullname=""
    pre_title="" name="" surname="" post_title="" email=""
    phone_number="" mobil_number="" fax_number="" street=""
    street_no="" zip="" city="" job="" department="" section=""
    note="" />
</gw_grp_fwdX>
</root>

```

, kde X je číslo nominálního kroku, tedy hodnoty 1 až maximální počet podpisů ve formuláři. Tagy vyznačené zelenou barvou nemusí být použity.

### Při prvním volání

```

<root url_send="http://webserver/save.php"
  url_forward="http://webserver/save_forward.php"
  url_back=""
  url_release="">

```

- K URL přibude parametr ...?id=ID\_DOKUMENTU, kde bude informace o skutečném ID dokumentu v databázi a informace o tom, zda dokument otevřel správný uživatel s právem odeslat data na server.

```
<gw_numsigns>0</gw_numsigns>
```

- Do tohoto tagu Form Server nastaví číslo odpovídající počtu uživatelských podpisů ve formuláři. Toto číslo nabývá hodnot 0 až maximální počet podpisů ve formuláři. Při přeskakování podepisovacích kroků bude počet podpisů menší nominální počet kroků.

```
<gw_status>0</gw_status>
```

- Nastaví se dle stavu formuláře:
  - 0 – formulář je ve schvalovacím procesu
  - 1 – formulář je schválen
  - 2 – formulář je zamítnut
  - 3 – v prvním kroku při vyplňování anonymem nebo autorem

```
<gw_archive>0</gw_archive>
```

- Form Server nastaví na 1, pokud byla XML data uložena do archívu, jinak nastaví 0.

```
<gw_message></gw_message>
```

- Tag se zprávou od Form Serveru nebo pro předchozího podepisujícího uživatele. Při přechodu do dalšího podepisovacího kroku Form Server tento tag vymaže.

XML data obsahující informaci o uživateli podepisující formulář ve všech krocích:

```

<gw_signX sign_level="a" signed="b"
  pre_title="" name="" surname="" post_title="" email=""
  phone_number="" mobil_number="" fax_number="" street=""
  street_no="" zip="" city="" job="" department="" section=""
  note="" />

```

- Kde X odpovídají nominálním číslům kroků. Začíná od 1 (číslo 1 obsahuje data autora). Tagů `gw_signX` je tolik, kolik je nadefinovaných nominálních podepisovacích kroků ve Form Serveru.
- Zde budou v prvním volání atributy u `gw_sign1` nastaveny takto:  
`sign_level="1" signed="0"`

602XML Form Server vloží dva datové číselníky se seznamem uživatelů.

```
<gw_fwdX>25</gw_fwdX>
```

- V tomto tagu je ID uživatele, který dostává formulář v daném kroku jako první.

```
<gw_grp_fwdX>
```

```
<gw_fwd_itemX id="11" fullname="Vonasek Karel Ing. (vedoucí IT)"
  pre_title="" name="" surname="" post_title="" email=""
  phone_number="" mobil_number="" fax_number="" street=""
  street_no="" zip="" city="" job="" department="" section=""
  note="" />
```

```
<gw_fwd_itemX id="25" fullname="Navratil Jan Bc. (zástupce
vedoucího
  IT)" pre_title="" name="" surname="" post_title="" email=""
  phone_number="" mobil_number="" fax_number="" street=""
  street_no="" zip="" city="" job="" department="" section=""
  note="" />
```

```
</gw_grp_fwdX>
```

Kde X odpovídají nominálním číslům kroků:

- X=1 – datový číselník obsahuje seznam uživatelů, kteří mohou formulář vhodit do schvalovacího procesu, neboli, kteří jej mohou podepsat jako první.
- X=2 – datový číselník obsahuje seznam uživatelů, kteří mohou formulář podepsat jako druzí.

#### V každém kroku (včetně postupování nebo vracení)

```
<root url_send="http://webserver/save.php"
  url_forward="http://webserver/save_forward.php"
  url_back="http://webserver/save_back.php">
```

```
<gw_numsigns>N</gw_numsigns>
```

- Do tohoto tagu Form Server nastaví číslo odpovídající počtu uživatelských podpisů ve formuláři. Toto číslo nabývá hodnot 0 až maximální počet podpisů ve formuláři. Při přeskokování podepisovacích kroků bude počet podpisů menší než nominální počet kroků.
- `<gw_status>0</gw_status>`

Nastaví se dle stavu formuláře:

- 0 – formulář je ve schvalovacím procesu
- 1 – formulář je schválen
- 2 – formulář je zamítnut
- 3 – v prvním kroku při vyplňování anonymem nebo autorem

```
<gw_archive>0</gw_archive>
```

- Form Server nastaví na 1, pokud byla XML data uložena ve FS (archivována), jinak 0.

```
<gw_message></gw_message>
```

- Tag se zprávou od Form Serveru nebo pro předchozího podepisujícího uživatele. Při přechodu do dalšího podepisovacího kroku Form Server tento tag vymaže.

```
<gw_signX sign_level="a" signed="b"
  pre_title="" name="" surname="" post_title="" email=""
  phone_number="" mobil_number="" fax_number="" street=""
street_no=""
  zip="" city="" job="" department="" section="" note="" />
```

- Form Server nastaví gw\_signX pro všechny nominální kroky podle následujících pravidel:
  - v již provedených krocích nastaví atributy sign\_level na počet podpisů ve formuláři před přidáním podpisu v daném nominálním kroku a signed na 1.
  - u uživatele, který právě stahuje formulář a bude jej podepisovat, nastaví všechny atributy s informací o uživateli a sign\_level na počet podpisů ve formuláři před podepsáním tímto uživatelem a signed na 0.
  - U zbývajících podpisů nastaví všechny atributy s informací o uživateli a sign\_level na „" a signed na „" .
- Atribut sign\_level obsahuje skutečné číslo podpisu ve formuláři, které nemusí souhlasit s nominálním číslem kroku.
- Atribut signed má tři hodnoty:
  - „" – tento podpis není aktivní
  - 0 – tento podpis právě bude podepisovat
  - 1 – tento podpis již podepsal

Form Server vloží datový číselník se seznamem uživatelů, kteří mohou podepisovat v následujícím kroku. Z tohoto číselníku si uživatel, který právě stahuje formulář, může vybrat z následujícího:

```
<gw_fwdY>id_fwd</gw_fwdY>
```

- ID uživatele, který dostává formulář v následujícím kroku jako první (implicitní hodnota datového číselníku).

```
<gw_grp_fwdY>
```

```
<gw_fwd_itemY id="11" fullname="Vonasek Karel Ing. (vedoucí IT)"
  pre_title="" name="" surname="" post_title="" email=""
  phone_number="" mobil_number="" fax_number="" street=""
street_no="" zip="" city="" job="" department="" section=""
  note="" />
```

```
<gw_fwd_itemY id="25" fullname="Navratil Jan Bc. (zástupce
vedoucího
```

```
IT)" pre_title="" name="" surname="" post_title="" email=""
  phone_number="" mobil_number="" fax_number="" street=""
street_no="" zip="" city="" job="" department="" section=""
  note="" />
```

```
</gw_grp_fwdY>
```

**V posledním kroku před uložením dat do archívu**

```
<root url_release="http://webserver/release.php"
  url_send=""
  url_forward=""
  url_back="">
  <gw_numsigs>N</gw_numsigs>
  <gw_status>1</gw_status>
```

```
<gw_message></xg_message>
<gw_signX ...
```

### Co čte 602XML Form Server z XML dat

#### Po prvním podpisu (podpis autora):

```
<gw_notify>0</gw_notify>
```

- Pokud si autor zaškrtně příslušnou volbu, dostane e-mailem zprávu o každém pohybu formuláře – tedy pokaždé, když jej někdo podepsal a odeslal na server.

#### V každém kroku:

```
<gw_refuse>0</gw_refuse>
```

- Pokud formulář nastaví na 1, Form Server vyhodnotí jako zamítnutí schvalovacího formuláře.

```
<gw_accept>0</gw_accept>
```

- Pokud formulář nastaví na 1, Form Server vyhodnotí jako schválení schvalovacího formuláře ještě dříve, než jej podepíše poslední schvalující.

```
<gw_message></xg_message>
```

- Tag se zprávou pro předchozího podepisujícího uživatele.

```
<gw_fwdY>25</gw_fwdY>
```

- ID uživatele, který bude podepisovat formulář v následujícím kroku. Tento element může být nastaven formulářem nebo Form Serverem (záleží na autorovi formuláře). Y je nominální číslo kroku.

## Úpravy formuláře pro koloběh dokumentů

V této kapitole si postupně ukážeme, jak upravit formulář v aplikaci 602XML Designer pro jeho nasazení do koloběhu dokumentů, který 602XML Form Server poskytuje.

### Informace o podepisujícím uživateli

Datová vazba elementu `signX` vypadá takto:

```
gw_signX
  sX.id
  sX.pre_title
  sX.name
  sX.surname
  sX.post_title
  sX.email
  sX.phone_number
  sX.mobil_number
  sX.fax_number
  sX.street
  sX.street_no
  sX.zip
  sX.city
  sX.job
  sX.department
  sX.section
  sX.note
```

Navíc jsou zde hodnoty všech tagů, které Form Server používá pro přednastavené hodnoty (viz kapitola Data přihlášeného uživatele na straně 3), ale bez předpony „gw\_“.

### Datový číselník

Datová vazba datového číselníku:

```

gw_fwdX
gw_fwd_itemX
  fwdX.id
  fwdX.fullname           = „surname name pre_title (job)“
  fwdX.pre_title
  fwdX.name
  fwdX.surname
  fwdX.post_title
  fwdX.email
  fwdX.phone_number
  fwdX.mobil_number
  fwdX.fax_number
  fwdX.street
  fwdX.street_no
  fwdX.zip
  fwdX.city
  fwdX.job
  fwdX.department
  fwdX.section
  fwdX.note

```

Navíc jsou zde hodnoty všech elementů, které Form Server používá pro přednastavené hodnoty (viz kapitola Data přihlášeného uživatele na straně 3), ale bez předpony „gw\_“.

Příklad:

Uživatel si má vybrat dalšího podepisujícího z komba (=rozbalovacího seznamu), do kterého Form Server naplní jména a ID ostatních příjemců. Form Server tedy naplní strukturu gw\_grp\_fwdX, Filler naplní tato data do číselníku a uživatel z komba vybere podle příjmení uživatele, jehož ID se vloží do gw\_fwdX.

V 602XML Designeru vytvoříme číselník, nazveme jej např. forward a navážeme na element gw\_fwdX. Následně doplníme atributy číselníku:

- form602indexlist gw\_fwdX
- form602listname forward
- form602bind gw\_fwdX
- form602ct ilb
- valuexp @id
- textxp @fullname
- form602reference //d:gw\_fwd\_itemX

Název	Hodnota
form602indexlist	gw_fwd2
form602listname	forward
form602bind	gw_fwd2
form602ct	ilb
valuexp	@id
textxp	@fullname
	...

Příklad:

Z číselníku je také možné vybrat "skryté", tedy pomocí XPath výrazu zapsaného do pole Výpočet ve vazbě.

Atribut naplňovaný hodnotou z datového číselníku bude mít například následující výpočet:

Název: **sX.surname**

Vazební výraz: @surname

Výpočet:

```
((//d:gw_grp_fwdX/d:gw_fwd_itemX [@id = //d:gw_fwdX])/@surname
```

kde **x** odpovídají nominálním číslům kroků a nabývá hodnot 1 až maximální počet podpisů. Závislost musí být nastavena na gw\_fwd**X**.

Příklad:

Element naplňovaný hodnotou z datového číselníku bude mít například následující výpočet:

Název: mesto

Vazební výraz: /d:root/d:mesto

Výpočet:

```
((//d:gw_grp_fwdX/d:gw_fwd_itemX [@id = //d:gw_fwdX])/@city
```

## Podpisy ve formuláři

Při práci s formuláři je řada akcí závislých na elektronických podpisech ve formuláři. Níže popsaná problematika se týká především parciálně podepsaných formulářů, kde v každém podpisovém kroku jsou na základě relevancí (výraz zapsaný v datové vazbě) do datové věty zařazeny pouze relevantní elementy a formulář je pak opatřen elektronickým podpisem. Při další práci s formulářem bývá třeba správně zobrazovat dříve podepsaná data a jméno vlastníka odpovídajícího podpisu.

### Typy podpisů ve formuláři

Formulář publikovaný programem 602XML Designer vždy obsahuje elektronický podpis vlastníka licence a může obsahovat i elektronický podpis vydavatele formuláře. Dále pak může obsahovat (parciální) podpisy jednotlivých uživatelů 602XML Form Serveru. Každý podpis má své pořadové číslo (=úroveň v terminologii Designeru). Budeme tedy hovořit o **celkovém počtu podpisů** ve formuláři (tedy včetně licenčního a vydavatelského) nebo o **počtu uživatelských podpisů** (pouze podpisy uživatelů Form Serveru ve vztahu k podepisovacím krokům).

### Počty podepisovacích kroků ve Form Serveru a odpovídající podpisy ve formuláři

V 602XML Form Serveru se pro daný formulář definují podepisovací kroky. Provedení podepisovacího kroku uskutečňuje uživatel podepsáním formuláře svým elektronickým podpisem. Počty kroků jsou nominální a reálné:

**Nominální počet kroků** odpovídá počtu kroků nadefinovaných v konfiguraci Form Serveru.

**Nominální číslo kroku** je pořadí kroku podle kroků nadefinovaných v konfiguraci Form Serveru.

**Reálný počet kroků** je počet reálně provedených kroků a je menší nebo roven nominálnímu počtu kroků, menší je v případě přeskočení kroku(ů).

Pozn.: Podpisy tedy ve formuláři přibývají "jeden za druhým", ale kroky se mohou přeskakovat. Navíc formulář může a nemusí obsahovat podpis vydavatele. Postup pro stanovení vztahu podepsaných dat, podpisu uživatele a případně podepisovacího kroku uvádíme níže.

Pro práci s podpisy ve formuláři jsou k dispozici dva typy funkcí a informací:

## 1) Funkce programu 602XML Filler pro práci s elektronickými podpisy ve formuláři

Tyto funkce jsou uvedeny v dokumentu "XPath výrazy, funkce a aplikační události.pdf" (dostupný na [www.602.cz](http://www.602.cz)), především v části "1.11 Popis funkcí rozšíření 602XML modulu XMLcore pro formulářové objekty". Zde uvedeme pouze jejich výčet a základní vlastnosti:

- GetNumSigns vrací aktuální počet podpisů ve formuláři
- SignLevel() vrací TRUE pokud aktuální počet podpisů ve formuláři odpovídá parametru.
- PartSignLevel vrací TRUE pokud aktuální počet podpisů ve formuláři odpovídá parametru, přičemž ignoruje elektronický podpis návrháře formuláře (PUBLISHER), pokud je ve formuláři obsažen.
- GetSignProperty vrací konkrétní informace o elektronických podpisech, které jsou k dokumentu připojeny.

Příklad:

```
GetSignProperty(GetFieldSignerIndex(//*[ @form602bind="supervisor" ]), SignSubject, 2.5.4.3)
```

Příklad syntaxe z form602extern:

```
GetSignProperty(GetFieldSignerIndex({//*[ @form602bind="v_Zprava1" ]}), SignSubject, 2.5.4.3)
```

## 2) Informace o elektronických podpisech ve formuláři dodané 602XML Form Serverem

Element gw\_signX obsahuje informace o uživatelích podepisujících formulář. Mj. se v něm nalézají atributy sign\_level a signed:

+ sign\_levelX - počet podpisů ve formuláři v kroku X v okamžiku před podpisem v daném kroku (na začátku 1 nebo 2) , kde X je nominální číslo kroku

+ atribut signed má tři hodnoty:

- „" - tento podpis není aktivní
- 0 - tento podpis bude právě podepisovat
- 1 - tento podpis již podepsal

Při otevření čisté šablony formuláře nepodepsané vydavatelem je v gw\_sign1 atribut sign\_level=1 a signed=0. Přeskočený krok má sign\_level=0 a signed="". A např. po skoku z prvního do čtvrtého kroku je pro gw\_sign4 atribut sign\_level=2 a signed=0.

- gw\_numsigns - aktuální počet uživatelských podpisů ve formuláři

### Zobrazení podpisu

Elektronický podpis vytvoří uživatel 602XML Form Serveru svým certifikátem. Při zobrazení jména uživatele tedy může autor formuláře zobrazit jméno uživatele 602XML Form Serveru, který formulář podepsal, nebo jméno majitele certifikátu použitého v podpisu. Z bezpečnostního hlediska je správné pracovat s certifikátem:

K zobrazení celého jména podepsaného uživatele použijeme následující výraz:

```
GetSignProperty(Y, SignSubject, 2.5.4.3)
```

, kde Y je číslo podpisu ve formuláři. Toto číslo může být pořadové číslo uživatelského podpisu + 1 nebo 2 (formulář podepsán vydavatelem). Tento postup bude pracovat správně pouze za předpokladu očekávaného počtu podpisů ve formuláři - přeskočení kroku nebo přidání podpisu vydavatele funkci naruší. Poslední parametr funkce je Object

Identifikátor podle normy X.509, 2.5.4.3 je cn (jméno), 2.5.4.5 je sériové číslo certifikátu, 2.5.4.6 je c (země) atd.

Vhodnější varianta výrazu:

```
GetSignProperty(GetFieldSignerIndex({/*[@form602bind="element"
]}), SignSubject, 2.5.4.3)
```

, kde `element` je libovolný element podepsaný daným podpisem (na základě *Relevance*). Tento výraz zobrazí jméno vlastníka certifikátu, kterým je podepsaná část formuláře obsahující uvedený element.

### Vložení a podepsání dat na základě relevance

Vložení výrazu do položky *Relevance* zajistí autor formuláře, že element odpovídající aktuální vazbě bude vložen do podepisované datové věty pouze v případě platnosti vloženého výrazu. Naopak pomocí výrazu v položce *Pouze pro čtení* zajistí, že v bude tento element zobrazen pouze pro čtení. V praxi se tedy tyto dva výrazy obvykle negují.

V položce *Relevance* můžeme použít následujících výrazů:

```
"SignLevel(X) "
```

, tedy s pevným číslem podpisu ve formuláři,

nebo lépe:

```
"SignLevel(DataXPath(//d:gw_signX/@sign_level)) "
```

kde `X` je číslo podpisu a `sign_level` je počet vložených podpisů v daném podp.kroku. Při použití tohoto výrazu se stane formulář nezávislým na případném podpisu vydavatele i na případném přeskočení kroku.

Do položky *Pouze ke čtení*, vložíme adekvátně následující výrazy:

```
"Not(SignLevel(X)) "
```

nebo

```
"Not(SignLevel(DataXPath(//d:gw_signX/@sign_level))) "
```

Vlastnosti datové vazby	
Název:	gw_fwd2 <span style="float:right">Datový typ: <input type="text"/></span>
Vazební výraz:	/d:root/d:gw_fwd2 <input type="button" value="..."/>
Výpočet:	<input type="text"/> <input type="button" value="..."/>
Editovatelnost:	<input type="text"/> <input type="button" value="..."/>
Pouze pro čtení:	"Not(SignLevel(DataXPath(//d:gw_sign1/@sign_level)))" <input type="button" value="..."/>
Povinnost:	true() <input type="button" value="..."/>
Relevance:	"SignLevel(DataXPath(//d:gw_sign1/@sign_level))" <input type="button" value="..."/>
Závislost:	gw_sign1 system <input type="button" value="..."/>
Omezení:	<input type="text"/> <input type="button" value="..."/>

### Přepínací sekce pro podpisy

Pokud chceme na základě počtu vložených uživatelských podpisů zobrazovat jen některé části formuláře, můžeme to provést pomocí přepínacích sekcí a použijeme pomocné elementy nazvané např. `signX`, které níže uvedeným výpočtem nabývají tří hodnot:

- 0 – `SignLevel X` ještě není na řadě (není aktuální)
- 1 – `SignLevel X` je právě na řadě – bude nyní podepisovat
- 2 – `SignLevel X` již podepsal

Výpočet pro `signX` vypadá takto:

```
(//d:gw_signX/@sign_level > 0) * (
(//d:gw_signX/@signed=0) + 2 * (//d:gw_signX/@signed > 0))
```

Atribut `sign_level` udává počet podpisů ve formuláři před přidáním podpisu v daném nominálním kroku, `signed` udává stav tohoto podpisu:

Zopakujeme, že atribut `signed` má tři hodnoty:

- „“ – tento podpis není aktivní
- 0 – tento podpis bude právě podepisovat
- 1 – tento podpis již podepsal

### ***Přepínací sekce pro odeslání schváleného nebo zamítnutého formuláře do archívu***

Vytvoříme si pomocný element, např. `to_archive`, který se bude počítat dle následujícího vzorce:

```
((//d:gw_status >0) and (//d:gw_status <3) and
(//d:gw_archive = 0)) + (//d:gw_archive = 1)*2
```

Hodnota elementu `gw_status` je přednastavena Form Serverem a nabývá tří hodnot:

- 0 – formulář je ve schvalovacím procesu
- 1 – formulář je schválen
- 2 – formulář je zamítnut

Hodnota elementu `gw_archive` je přednastavena Form Serverem a nabývá dvou hodnot:

- 0 – XML data z formuláře ještě nejsou uložena v archívu
- 1 – XML data jsou již v archívu uložena

Pomocný element `to_archive` bude nabývat tří hodnot:

- 0 – ještě se nebude ukládat do archívu
- 1 – formulář je schválen nebo zamítnut a je připraven k uložení XML dat do archívu
- 2 – XML data z formuláře jsou již v archívu uložena

### ***Přepínací sekce postoupení předvyplněného formuláře prvnímu podepisujícímu***

Pokud formulář ještě nebyl postoupen, má element `gw_status` hodnotu **3**, viz výše.

Vytvoříme si tedy pomocný element, např. `sign0`, jehož hodnota se bude počítat například podle následujícího vzorce

```
((//d:gw_status!=3)*1
```

Tento výpočet nabývá hodnot:

- 0 - formulář ještě nebyl postoupen
- 1 - formulář již byl postoupen prvnímu podepisujícímu

Jestliže používáme ve Form Serveru postupování formuláře, musí být pro první podpis (pro `sign1`) použit následující vzorec, který počítá s elementem `gw_status`:

```
((//d:gw_sign1/@sign_level>0)*((//d:gw_sign1/@signed=0)+2*(//d:gw_sign1/@signed > 0))*(//d:gw_status!=3)
```

Tento výpočet pak nabývá hodnot:

- 0 - formuláře ještě nebyl postoupen

- 1 - formulář má podepsat první podepisující (ten, kterému byl postoupen)
- 2 - formulář je podepsán prvním podepisujícím

## Možnosti ladění funkce formuláře v 602XML Form Serveru

Méně zkušenému autorovi formuláře nebo správci Form Serveru se může stát, že právě dokončený formulář nebude po první instalaci do 602XML Form Serveru pracovat správně a bude třeba hledat příčinu. Zde popíšeme prostředky, které jsou pro ladění k dispozici. Možnosti jsou v úpravě formuláře, použití ladicí funkce aplikace 602XML Filler a nastavení vyššího stupně výpisů (loglevel) Form Serveru.

### Formulář

- přidání kontrolních výpisů do formuláře

Nejsnadnější možností je přidání kontrolních výpisů některých elementů přímo do formuláře - zde se při práci s formulářem zobrazí hodnoty, podle kterých se např. přepínají sekce nebo které odcházejí zpět do Form Serveru jako součást datové věty formuláře.

- kontrola obsahu XML souboru

Další možností je prohlížení obsahu formuláře jako XML dat. Ve formuláři jsou vidět elementy zobrazované na obrazovce a jednotlivé datové věty. Ve spodní části XML souboru snadno najdeme datovou větu formuláře a za ní i datovou větu připojenou Form Serverem sloužící k předvyplnění hodnot v datové větě formuláře Fillerem - zde právě bývá nejčastěji problém, např. překlep ve jménech elementů, kvůli kterému se neprovede předvyplnění, případně chybná/neodpovídající struktura vět. Jak na to? Formulář je při stažení z Form Serveru ve tvaru ZFO. Abychom ho mohli prohlížet jako XML data, je třeba koncovku přepsat na ZIP a soubor otevřít vhodným dekomprimačním programem. V tomto archívu bude soubor form.fo. Tento soubor vykopírujeme na disk a změníme jeho koncovku na XML. Potom je možné pouhým poklepáním soubor otevřít v programu pro přehledné zobrazení XML dat, kterým obvykle bude internetový prohlížeč, např. MSIE.

### 602XML Filler

602XML Filler umožňuje zapnutí ladicí konzole (Ctrl-F5, resp. pro plugin Ctrl-0 (klávesa é/0, nikoliv 0 na numerické klávesnici)). Tato klávesová zkratka otevře ve spodní části okna programu podokno pro výpis zpráv programu.

- **výpis datové věty** (Ctrl-Shift-D kdykoliv z prostředí programu 602XML Filler nebo v pluginu z menu vyvolaného pravým tlačítkem myši v ladicí konzoli)

Pomocí uvedené klávesové zkratky otevřeme okno, kde je přehledným způsobem zobrazena XML datová věta a v ní obsažené hodnoty vypovídající o stavu formuláře v aplikaci 602XML Filler.

Ve Windows, kde není nainstalovaný vývojářský licenční certifikát užívaný pro provoz 602XML Designer, je pouze možné v menu vyvolaném pravým tlačítkem myši v okně konzole zapnout logování provozu programu do souboru a konfigurovat výpisy o práci programu.

Po nainstalování licenčního certifikátu se zpřístupní ještě další volby menu:

- řádka "**vyhodnocení výrazu**" umožňuje přímo zapisovat XPath výrazy pracující s aktuálními daty v datové větě formuláře a po odklepnutí dojde ke zpracování a zobrazení výsledku. Tím je možné si vyzkoušet správný tvar výrazu nebo najít chybu v datech

- řádka "**spouštění aplikací**" slouží ke spuštění událostí 602XML Filleru, např. odeslání dat podle odesílacího profilu.

## Form Server

- nastavení vyššího režimu logování (loglevel) Form Serveru

V souboru config (bez koncovky) lze mj. nastavit úroveň výpisů serveru do log souboru.

V poznámce u tohoto klíče je uveden výčet možných hodnot a jejich významu:

```
define('XMLGW_LOGFILE_LOGLEVEL',2); // 4=debug, 3=detailed, 2=extended,  
1=normal, 0=none
```

Zbytečné zvýšení úrovně vede k nepřehlednosti výpisu, doporučujeme s tímto nastavením pracovat uvážlivě.

## Apendix A - Linky

Na formulář se lze odkazovat několika různými URL, tzv. linky. Link na formulář můžeme uvést na webové stránce, v obsahu informačního e-mailu, v jiném formuláři atd. Přístupová práva k šabloně formuláře řeší Form Server, nikoliv aplikace, kde je link uveden.

Uvedme zde varianty linků a vlastnosti:

### Linky na šablonu formuláře

- Link pro vyplnění šablony **přihlášeným uživatelem** Form Serveru je uveden ve Správě formulářů, ve větvi Informace.

Např.: /login.php?goto=fill&id=45

- Link pro vyplnění šablony formuláře **anonymním uživatelem** je uveden ve Správě formulářů, ve větvi Anonymní uživatel. Na tomto místě je také uvedena URL *url\_release platné pro celý formulář*, kterou je třeba doplnit do formuláře v případě, že bude k vyplňování distribuován jinou cestou než stažením z Form Serveru (tedy e-mailem, stažením z webu nebo FTP atd.)

Např.: /aforms.php?action=fill&id\_form=45

Tyto linky jsou nezávislé na verzi šablony formuláře.

### Univerzální odkaz na formulář (rozpracovaný/archiv)

Link má tento tvar:

/login.php?goto=ushow&id=ID , kde ID je ID částečně vyplněného formuláře (id\_ufrm)

Tento link ukazuje na částečně nebo zcela vyplněný formulář. Pro stažení formuláře uživateli postačí práva přístupu, není nutné, aby uživatel měl v daném okamžiku možnost pokračovat ve vyplňování či jiném zpracování tohoto formuláře.