



GORDIC®

Informační systémy pro státní správu,
samosprávu a bankovníctví

GINIS 3.72

Modul GFE

Návrhář formátů sestav

Základní příručka uživatele
Od verze 4.72

Modul GFE

Návrhář formátů sestav

Základní příručka uživatele
Od verze 4.72

GINIS - Návrhář formátů sestav

Žádná část této publikace nesmí být publikována a šířena žádným způsobem a v žádné podobě bez výslovného souhlasu společnosti GORDIC spol. s r.o.

Značky GORDIC®, GINIS®, UCR® jsou ochrannými známkami a pobírají ochranu dle příslušných právních předpisů.

Vydal: GORDIC spol. s r.o., Erbenova 4, 586 01 Jihlava; I : 47 90 37 83.

Tisk: V Jihlavě 2013.

Obsah

Část 1 Úvod, obecné zásady práce v modulu	8
Část 2 Popis práce v aplikaci	11
2.1 Vstupní okno	11
2.2 Práce se sestavou	12
2.2.1 Načtení sestavy	12
2.2.2 Uložení sestavy	13
2.2.3 Použití existujících datových struktur	14
2.3 Okna	16
2.3.1 Startovací stránka	16
2.4 Práce s databází GINIS	16
2.4.1 Přihlášení do modulu	17
2.4.2 Načtení formátu z databáze	18
2.4.3 Uložení do databáze	19
Část 3 Reporter	22
3.1 Soubor s daty sestavy	22
3.2 Soubor s popisem struktury dat sestavy	23
3.3 Soubor s předpisem pro formátování sestavy	23
3.4 Základy	24
3.5 Stránkování	25
3.6 Číslování stran	25
3.7 Stránkový součet	26
3.8 Rozdělení textu do více řádků	26
3.9 Skriptování	26
3.10 Datové typy	29
3.10.1 Řetězec	29
3.10.2 Číslo	30
3.10.3 Datum a čas	32
3.11 Podmíněné zobrazení	33
3.12 Třídění dat	34
3.13 Počítané proměnné	34
3.14 Agregace	35
3.15 Skupinování	36
3.16 Sdílení stylů	37
3.17 Obrázky	37
3.18 Nastavení velikosti stránky pomocí paper-setting	38
3.19 Nastavení výchozího zobrazení	39
3.20 Velikosti	39
3.21 Tabulky	39
3.22 Grafy	40

3.23	Výkazy a formuláře (GRF)	45
3.23.1	Stránkování v GRF	46
3.24	Soubor s popisem struktury dat sestavy (.xme)	46
3.25	Soubor s předpisem pro formátování sestavy (.alf)	47
3.25.1	Skupina formátování GRR a GRF	49
3.25.2	Skupina formátování MSE a MSW	54
3.25.3	Skupina formátování RTF	54
3.25.4	Seznam speciálních hodnot dostupných ve formátovacím předpisu (.alf)	55
3.26	Pokročilé skriptování	55
3.27	Převodní mosty	56
3.27.1	Do TXT	57
3.27.2	Do HTML	57
3.27.3	Do MHT	57
3.27.4	Do PDF	57
3.27.5	Do XLS (GRR)	57
3.27.6	Do DOC (GRR)	57
3.27.7	Do RTF (GRR)	57
3.27.8	Do XML (GRR, MSW, MSE)	58
3.27.9	Do XLS (nativní – MSE)	58
3.27.10	Do DOC (nativní – MSW, RTF)	58
3.28	Distribuce v systému GINIS	59

Předmluva

Dámy a pánové, vážení uživatelé,

dostává se Vám do rukou příručka pro práci v modulu Návrhář formátů sestav, který je součástí systému GINIS - Gordic Integrovaný Informační Systém.

Tato příručka si klade za cíl poskytnout Vám potřebné informace pro plnohodnotnou práci s modulem, tak aby Vaše práce s programem byla maximálně efektivní a pro Vás příjemná.

Příručka je členěna do několika základních částí:

Úvod a obecné zásady práce v aplikaci

- úvodní část je věnována elementárnímu seznámení s modulem a ukazuje základní vazby v rámci systému GINIS®, dále pak popisuje obecné zásady práce v aplikaci, typy ovládacích prvků a obecnou terminologii.

Popis práce v aplikaci, jak na to

- tato část tvoří jádro celé příručky a v maximální možné míře, při zachováníčitelnosti a orientace v textu charakterizuje, jak se v modulu orientovat a jak s ním efektivně pracovat.

Dodatky a změny

- část obsahuje dodatky, doplnění, změny. Tato část představuje novinky v modulu a slouží pro rychlou orientaci a seznámení se s novými funkcionalitami.

Věříme, že tato příručka Vám bude platným pomocníkem při Vaší práci.

Hodně úspěchů přeje

tým pracovníků GORDIC®.

část



**Úvod, obecné
zásady práce v
modulu**

1 Úvod, obecné zásady práce v modulu

GFE ver. 4.72

Modul **GFE – Návrhář formátů sestav** umožňuje vytváření a úpravy sestav uživatelem. Lze měnit styl výstupů tisknutých nebo prohlížených grafických výstupů. Modul pracuje se dvěma základními soubory, souborem datové struktury a souborem formátu sestavy. Z prvního souboru se bere struktura dat tak, jak jsou uložena v databázi a druhý soubor slouží k vlastnímu popisu vzhledu sestavy. V návrhář sestav je to právě soubor formátu sestavy, který se vytváří nebo upravuje. Modul **GFE** pro některé svoje funkce vyžaduje informace o subjektech organizace, jejich vazbách a vlastnostech definovaných administračními moduly systému GINIS®. Popis obsluhy definice těchto subjektů je nad rámec této příručky a lze ho nalézt v dokumentaci modulů administrace. Následující popis modulu **GFE** je obecný a může obsahovat sekce, které nejsou u daného typu uživatele díky konfigurovatelnosti systému použité.

Modul **GFE** má v systému GINIS® výjimečné postavení. Modul lze použít i bez instalace celého systému nebo na počítačích, kde se systém GINIS® běžně nevyužívá. Lze použít i pro úpravy sestav jiných systémů (např. Gordic G0). Spuštěním modulu v rámci systému GINIS® získáváte možnost získání informací a sestav uložených v databázi systému.

Všechny obrázky oken aplikace zobrazené v této příručce jsou pouze informativní a nemusejí zcela odpovídat prvotnímu zobrazení v aplikaci, které je často ovlivněno nastavením přístupových práv v aplikaci.

Datum poslední aktualizace dokumentace: **29.10.2013**.

část



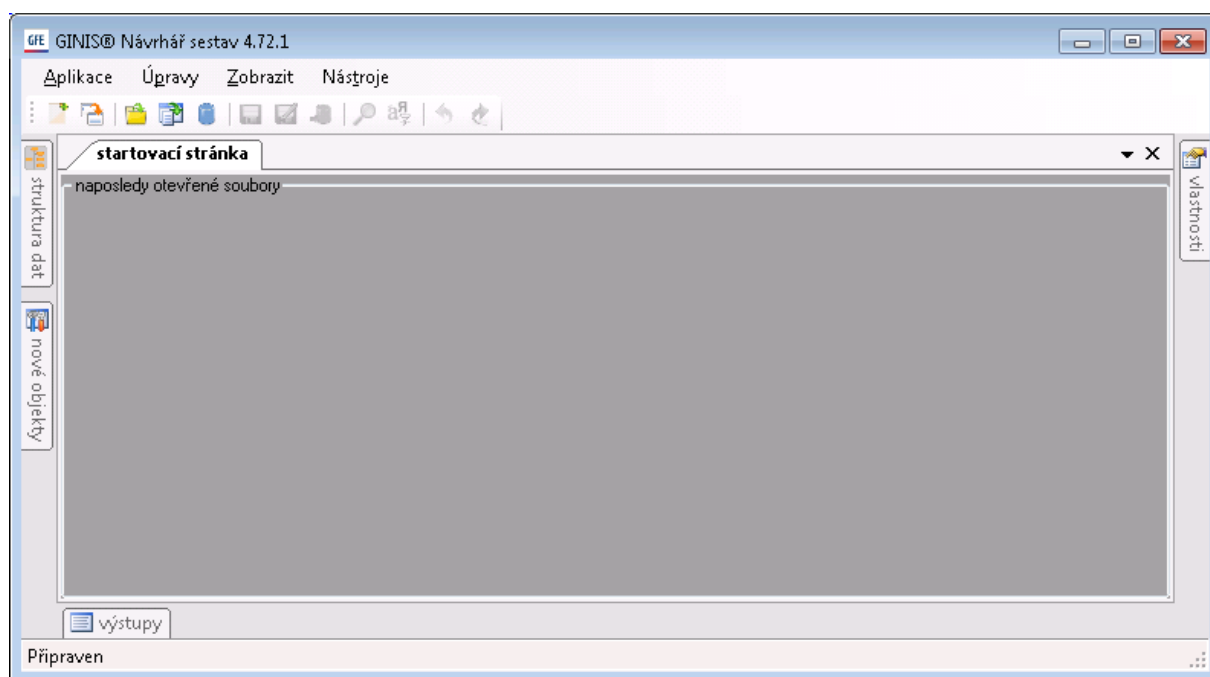
**Popis práce v
aplikaci**

2 Popis práce v aplikaci

Tato část příručky v maximální možné míře, při zachování čitelnosti a orientace v textu charakterizuje, jak se v modulu orientovat a jak s ním efektivně pracovat.

2.1 Vstupní okno

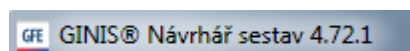
Vstupní okno modulu **GFE** vypadá následovně



Vstupní okno modulu GFE

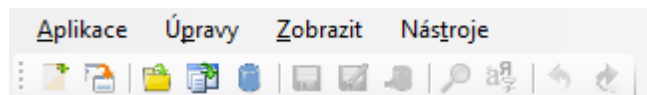
Jak je vidět, okno je rozděleno na několik částí.

První stojí za zmínku samotný popis okna. Stojí v něm označení a verze návrháře, která je aktuálně používaná. Po přihlášení do databáze GINIS se navíc zobrazí informace o přihlášené databázi.



Popisek modulu GFE

Následuje hlavní nabídka (menu) a pruh nástrojů:




Hlavní nabídka aplikace

Informační lišta je umístěna úplně ve spodní části základního okna, je šedě vyplněna a při minimalizaci celého okna lze chytit jeho pravý dolní roh vyznačený šesti šedými tečkami ve tvaru trojúhelníku, a poté kliknutím na levou stranu myši a táhnutím (doprava, doleva, nahoru, dolů) můžeme okno měnit do požadovaných rozměrů. V liště se zobrazují zprávy o průběhu vykonávaných operací.

Připraven

Informační lišta

Po stranách se otevírají [různá okna](#) (např. Struktura dat, Vlastnosti atp.). Tato okna lze tažením přesouvat dle preferencí uživatele. Lze je také pomocí ikony špendlíku  připnout či odepnout.

Při běžném spuštění modulu se otevře také okno [Startovací stránka](#) s rozcestníkem na otevření naposledy upravovaných sestav.

2.2 Práce se sestavou

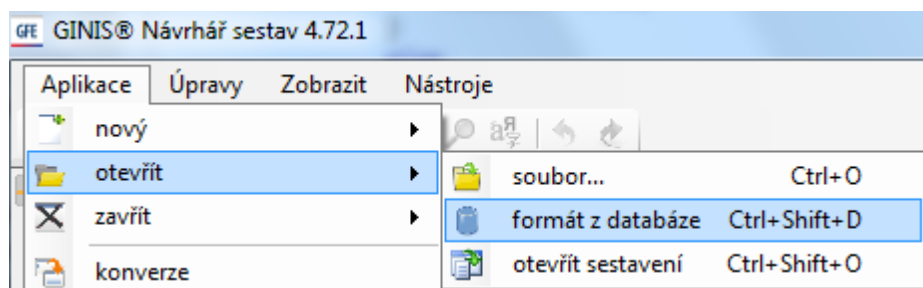
Běžný postup práce v modulu při úpravě existující sestavy je následující:

- [otevření sestavy](#) (načtením ze souboru nebo vyhledáním v databázi **GINIS®**)
- dle typu otevřené sestavy se zobrazí plocha s návrhem sestavy
- samotná úprava sestav
- [uložení sestavy](#) (do souboru nebo do databáze **GINIS®**)
- v případě uložení pod jiným jménem (identifikací) vznikne nový formát zařazený stejně jako původní

2.2.1 Načtení sestavy

Načtení formátu z databáze

- Ve vstupním okně modulu zvolte ikonu lišty  nebo volbu menu **otevřít formát z databáze**

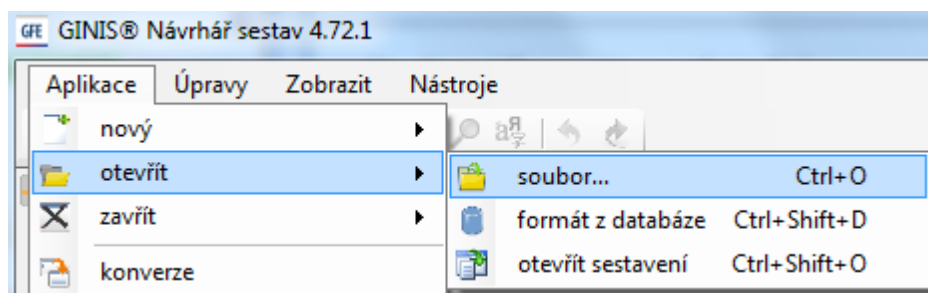


Otevřít formát z databáze

- pokud ještě nejste přihlášení do systému, pak se objeví [okno přihlášení](#)
- po přihlášení se objeví průvodce [načtení sestavy z DB](#)

Otevření formátu ze souboru

- ve vstupním okně modulu zvolte ikonu lišty  nebo volbu menu **otevřít soubor**



Otevřít formát ze souboru


- Otevře se dialogové okno pro vyhledání souboru
- Vyberte formát, který chcete upravovat (soubor .ALF) a klikněte na tlačítko **Otevřít**
- Dojde k automatickému hledání odpovídající struktury. Struktura musí být umístěna ve stejném adresáři jako otevíraný formát.
- Pokud se struktura neotevře, je nutné, před samotnou editací formátu, ji [otevřít](#).

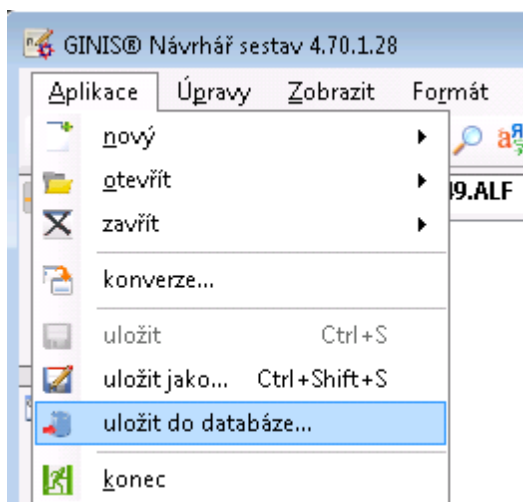
Otevření nedávno otevřeného formátu

- na [Startovací stránce](#) najdete odkaz na požadovaný formát a klikněte na něj

2.2.2 Uložení sestavy

Uložení formátu do databáze


- Ve okně s otevřeným formátem zvolte ikonu lišty  nebo volbu menu **uložit do databáze**

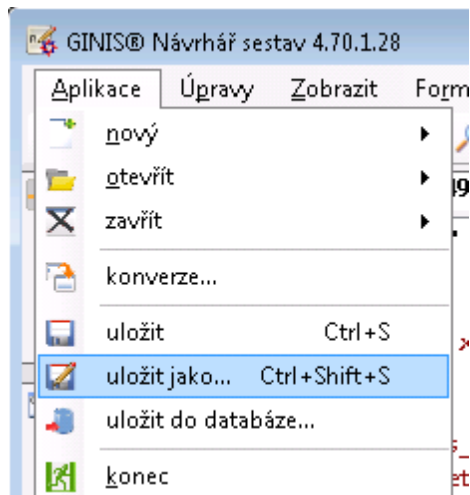


Uložit formát do databáze

- pokud ještě nejste přihlášení do systému, pak se objeví [okno přihlášení](#)
- po přihlášení se objeví [průvodce uložením sestavy z DB](#)

Uložení formátu do souboru

- ve vstupním okně modulu zvolte ikonu lišty  nebo volbu menu **uložit jako**



Uložit formát do souboru

- Otevře se dialogové okno pro zadání názvu souboru
- Do souboru lze uložit i formáty otevřené z databáze. Lze tak přenést formát na jiný počítač či zálohovat rozdělanou práci mimo databázi

Volba Uložit

Volbou Uložit (klávesami Ctrl+S) lze uložit do stejného umístění, z kterého byl formát otevřen.

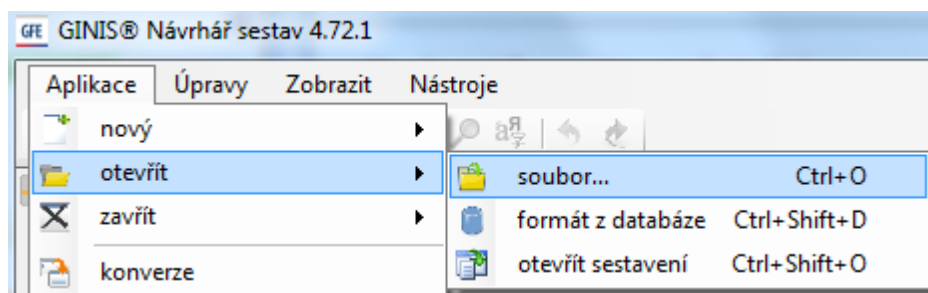
2.2.3 Použití existujících datových struktur

Při otevření formátu z databáze GINIS®, dojde k automatickému otevření struktury. Pokud otevíráte formát ze souboru a nemáte odpovídající strukturu ve stejném adresáři, je nutné ji otevřít samostatně.

Otevření datové struktury

- ve vstupním okně modulu zvolte ikonu lišty  nebo volbu menu **otevřít soubor**

Popis práce v aplikaci



Otevřít formát ze souboru

- Přepněte se na typ souborů struktura (XME)
- Vyberte odpovídající strukturu
- Vybraná struktura se otevře v okně Struktura dat

2.3 Okna

Popis oken modulu.

2.3.1 Startovací stránka

Okno obsahuje rozcestník na otevření naposledy upravovaných sestav.



Plocha startovací stránky

V sekci **Naposledy otevřené soubory** se ukládají naposledy otevřené formáty (či jiné soubory) v posloupnosti od nejdříve otevřené po nejpozději otevřenou tak, aby po opuštění GFE a jeho následném spuštění byla vidět naposledy otevřená místa.

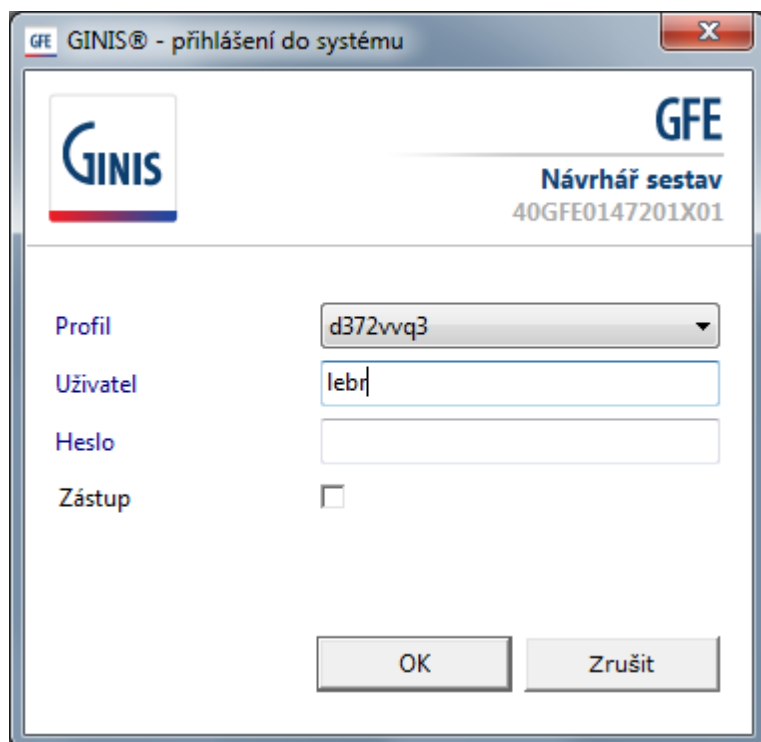
Po najetí myši nad některou z cest se objeví nápovědné okno s informací o tom, kde se nachází sestava a jakého typu

2.4 Práce s databází GINIS

Spuštění modulu v rámci systému **GINIS®** umožňuje využití informací uložených v databázi systému.

Po přihlášení do modulu lze načíst sestavy v databázi již uložené a zpět je do databáze uložit (úprava existujících sestav) nebo ji uložit pod jiným jménem (vytvořit tak novou sestavu na základě již existující). Tímto postupem se odpadne většina práce nutná pro založení nové sestavy.

2.4.1 Přihlášení do modulu



Ukázka přihlašovacího okna

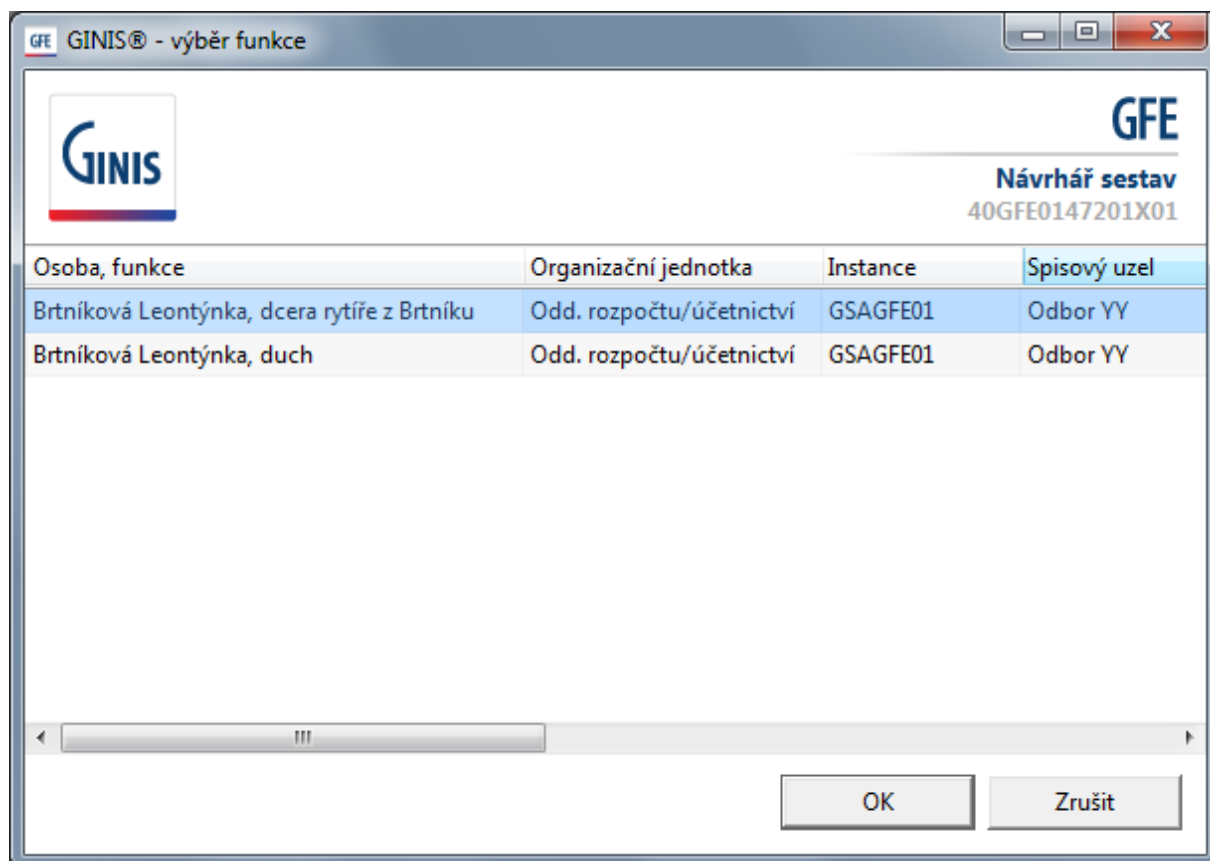
V přihlašovacím okně je třeba zadat název databáze, přihlašovací jméno a heslo:

- Název databáze, do které se přihlašujete, je uveden v poli **Profil**.
- Do pole **Uživatel** запиšte přidělené přihlašovací jméno.
- Stiskněte klávesu **Tab** nebo klikněte myší do pole **Heslo**. Kurzor se přesune do pole **Heslo**.
- Do pole **Heslo** zadejte přidělené heslo (na obrazovce se z bezpečnostních důvodů zobrazují pouze hvězdičky).
- V případě potřeby označte kliknutím myši pole **Zástup** (popis viz níže).
- Stiskněte klávesu **Enter** nebo klikněte myší na tlačítko **OK**.

Pokud se přihlášení nepodaří, ověřte, zda je správně napsáno jméno a heslo. Případné chyby bývají často způsobeny neúmyslným zapnutím klávesy **Caps Lock** nebo záměnou písmen **y** a **z** na klávesnici. Obsahuje-li uživatelské jméno či heslo číslice, může být problém způsoben i vypnutou numerickou klávesnicí.

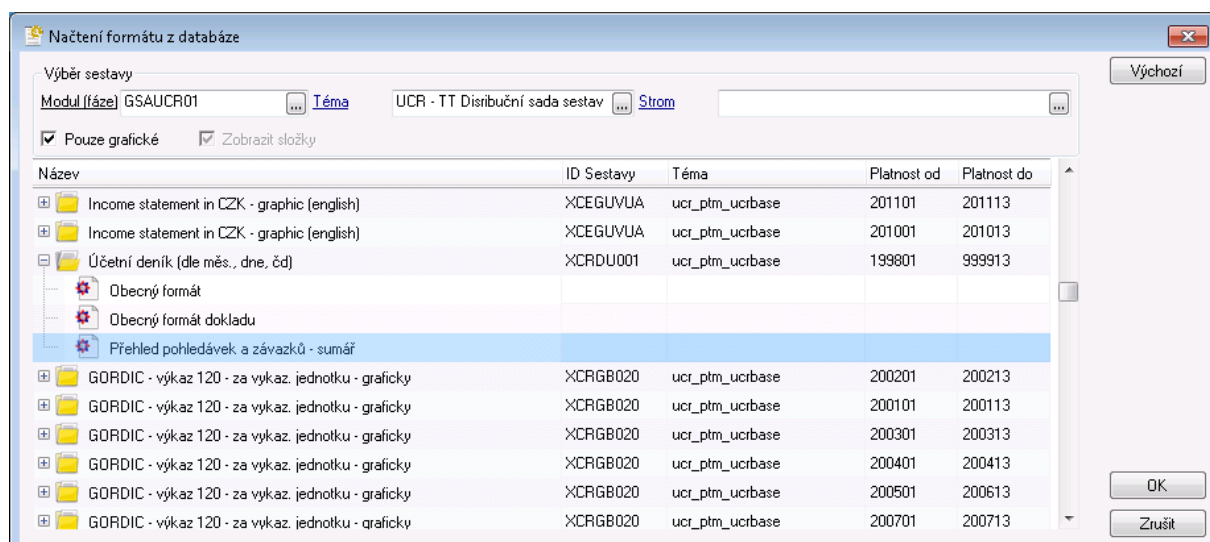
Každý uživatel má v administraci přiřazenou svou funkci (**Funkční místo**) nebo i několik různých funkcí. Kromě toho je možné mu přiřadit právo **zastupovat jiného pracovníka v jeho funkci**. Když se uživatel přihlásí do modulu bez označení pole **Zástup**, může vykonávat činnosti příslušející jeho vlastní funkci (editovat vlastní doklady atd.). Jestliže při přihlášení do modulu zaškrtně pole **Zástup**, může vykonávat činnosti, které má normálně právo vykonávat na základě své funkce jím zastupovaný pracovník (editovat jeho doklady atd.).

V případě, že uživatel má přiděleno více funkčních míst nebo může zastupovat několik pracovníků, musí vybrat, pod kterou funkcí se chce přihlásit do modulu. Výběr příslušné funkce se provede kliknutím myší na příslušný řádek a potvrdí tlačítkem **OK**.



Ukázka výběru funkce

2.4.2 Načtení formátu z databáze



Dialogové okno Načtení z DB

V horní části okna lze zúžit množinu nabízených sestav filtračními políčky. Po vyplnění všech 3 filtračních polí, je možno při zaškrtnutí pole "zobrazit složky" docílit zobrazení stejné nabídky, jako z konkrétního modulu systému. Jinak jsou zobrazeny všechny sestavy

bez hierarchického rozlišení ve složkách. Dvojklikem na záhlaví sloupce lze ještě aktivovat rychlý filtr, např. dle ID nebo názvu sestavy či formátu.

Informativně lze zobrazit i negrafické sestavy, ačkoliv tyto nelze otevřít.

Grafické sestavy, mohou obsahovat více formátů, ty lze zobrazit kliknutím na ikonu plus (případně klávesou Num+). Vybrat k úpravě lze libovolný z formátů.

2.4.3 Uložení do databáze


Dialogové okno Uložení do DB

Uložení formátu do databáze vyžaduje zadání několika doplňujících údajů. Tyto údaje by měly být zapsány pomocí Info sekcí ve formátu samotném, okno před uložením je zobrazí pro kontrolu a umožňuje měnit. Tyto údaje slouží k registraci formátu v databázi a přidružení ke správné sestavě.

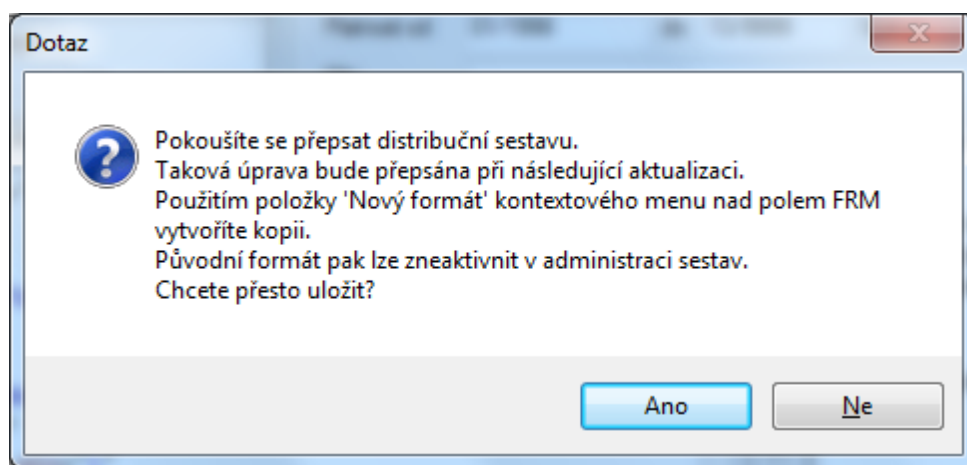
V horní části okna je pole Modul. Slouží ke specifikaci výčtu modulů, ke kterým se sestava nahraje. Je možné vynutit reinstalaci těchto modulů uživatelům (při dalším spuštění modulu se ten automaticky reinstaluje, včetně nově uloženého formátu). Pokud reinstalaci nevynutíte, nahraje se upravený či nový formát jen těm uživatelům, kterým se reinstalace spustí z jiného důvodu (první spuštění modulu atp.). Taktéž je možno formát uložit do souboru a ten nahrát do adresáře FRM příslušného modulu (např. pro otestování).

Ve spodní části okna identifikace samotného formátu (pole FRM) a další údaje sloužící k

prezentaci formátu uživateli. Změna těchto údajů se projeví i změnou odpovídajících info sekcí ve formátu samotném.

Formát je také možno uložit pod jiným jménem (obdoba Uložit jako). Pro uložení je nejdůležitější identifikace (pole FRM), pokud použijete existující identifikace, dojde k přepsání souboru. K tomuto slouží tlačítko , které vytvoří novou a unikátní identifikaci formátu. Identifikace bude začínat licencí vaší databáze - tak poznáte že se nejedná o distribuční formát.

Pokud měníte distribuční formát (formát vytvořený dodavatelem SW), doporučujeme uložit pod jinou identifikací. Pokud opomenete, budete před uložením ještě upozorněni takovýmto dotazem



Dotaz před uložením distribučního formátu

ást



Reporter

3 Reporter

Gordic Reporter - specifikace verze 1.4

Poslední aktualizace 1.11.2012 9:43:00 Martin Aliger

Gordic Reporter slouží jako souhrnné označení několika komponent, jejichž účelem je zprostředkovat grafické zobrazení a tisk sestav. Veškeré prováděné zpracování sestavy v této aplikaci je přitom založeno na existenci minimálně tří speciálních souborů pro každou formátovanou sestavu. Jinak řečeno, aby bylo možno graficky zobrazit a tisknout sestavu prostřednictvím aplikace Gordic Reporter, musí před započítím vlastního formátování na klientském počítači existovat minimálně tři speciální soubory. Jsou to:

- [soubor s daty sestavy](#)
- [soubor s popisem struktury dat sestavy](#)
- [soubor s předpisem pro formátování sestavy](#)

3.1 Soubor s daty sestavy

Soubor s daty sestavy je textový soubor, jehož obsahem jsou vlastní data sestavy. Vytváří ho aplikace po obdržení požadavku na tisk sestavy od uživatele klasicky spuštěním generátoru sestav (pro GINIS typicky ALV, Win&DOS sestavovým nebo výkazovým generátorem apod.)

Formát tohoto souboru připomíná soubor, který vznikne použitím příkazu unload na Informixu. Jednotlivé údaje uvnitř něj jsou členěny do řádků, přičemž každý řádek je považován za samostatný a ucelený záznam, podobně jako řádek dat v databázové tabulce. Tento záznam se dále dělí na jednotlivé datové položky, které jsou v podstatě elementárními částmi dat. Pro srovnání, v analogii databázové tabulky jsou to jednotlivé atributy dat na řádku. Jednotlivé položky dat na řádku jsou přitom odděleny svislítkem. V případě výskytu oddělovače dat (svislítko) nebo oddělovače řádku (enter) či zpětného lomítka v datové položce je nutné použít speciální zápis, viz následující tabulka.

První datová položka na řádku je v souboru s daty sestavy vždy chápána jako identifikátor tzv. oblasti (region) korespondující s označením oblasti v souborech popisujících strukturu dat sestavy a formátování sestavy. Tzn. první datová položka na řádku je v podstatě klíčovým slovem, přes které je vytvořena vazba mezi vlastními daty sestavy a popisem jejich struktury a formátování. Za označením oblasti na začátku každého řádku potom až do konce řádku následují jednotlivé datové položky obsahující vlastní data sestavy v počtu a pořadí přesně odpovídajícím definici oblasti uvedené v příslušném souboru s popisem struktury dat sestavy.

Speciální funkci má mimo to první řádek souboru s daty sestavy. Na tomto řádku musí být uveden jednoznačný identifikátor struktury dat (označovaný jako ix_s_xme, viz dále), ke které se daná data vážou, její verze a sub verze.

Znak	Zápis v datovém souboru
	\
<enter>	\n
\	\\

Příklad souboru s daty sestavy (tři tečky zde zastupují jednotlivé datové údaje) :

```
ixs_xme|0000XME00000|1|0|
faktura|...|...|...|...|
radek_faktury|...|...|...|
radek_faktury|...|...|...|
radek_faktury|...|...|...|
```

```
faktura|...|...|...|...|
radek_faktury|...|...|...|
radek_faktury|...|...|...|
```

3.2 Soubor s popisem struktury dat sestavy

Souboru s popisem struktury dat sestavy je ve formátu XML (viz kapitola [Základy](#)). Jeho primárním významem je specifikace přesného popisu jednotlivých datových položek v souboru s daty sestavy s důrazem na jejich seřazení a hierarchické vztahy mezi nimi. Slouží však rovněž jako výchozí předpis pro návrh formátu sestavy pomocí za tím účelem vytvořené zvláštní aplikace. S využitím XML syntaxe je tedy v souboru s popisem struktury dat popsán počet a vzájemné vztahy jednotlivých typů oblastí dat (region), které se přes svůj název váží na příslušné řádky souboru s daty sestavy generovaného klientskou aplikací. Zároveň pro každou oblast dat definuje počet a názvy jednotlivých datových položek (item), jenž s ohledem na pořadí koresponduje s datovými položkami na příslušném řádku datového souboru. Tak např. první položka definovaná ve struktuře dat se váže k první datové položce následující za označením oblasti na odpovídajícím řádku dat v souboru s daty sestavy, atd.

V položce info (viz příklad) souboru s popisem struktury dat sestavy je povinné uvádět jako atribut identifikátor struktury dat, verzi, sub verzi a název struktury dat, volitelně se zde může uvádět i její popis.

Příklad souboru s popisem struktury dat sestavy:

```
<?xml version="1.0" encoding="windows-1250" ?>
<structure xmlns="http://www.gordic.cz/TR/xme">
  <info ixs_xme="0000XME00000"
    xmeta_ver="1"
    xmeta_subver="0"
    nazev="odb ratelská faktura"
    popis="..."
  />
  <region name="faktura" title="Faktura">
    <item name="nazev_odberatele" title="Název odb ratele" />
    <item name="adresa" title="Adresa" />
    <item name="mesto" title="M sto" />
    <item name="psc" title="PS " />

    <region name="radek_faktury" title=" ádek faktury">
      <item name="cislo_položky" title=" íslo položky" />
      <item name="popis_položky" title="Popis položky" />
      <item name="fakturovana_castka" title="Fakturovaná ástka" />
    </region>
  </region>
</structure>
```

3.3 Soubor s předpisem pro formátování sestavy

Soubor s předpisem pro formátování sestavy je v podstatě popisem grafického výstupu sestavy na obrazovku, tiskárnu či jiné výstupní zařízení. Je abstrakcí nad strukturou dat popsanou v [předchozí kapitole](#). Tzn., neodvolává se již na fyzické rozmístění a uspořádání dat, ale pouze na jejich virtuální identifikaci v rámci příslušné struktury dat. Tato identifikace je pro každou datovou položku unikátně možná specifikací jejího názvu při jejím současném přiřazení do příslušné oblasti dat.

V souboru s předpisem pro formátování sestavy je výstup popsán rovněž s využitím XML syntaxe (viz kapitola [Základy](#)). Formátování rozdělujeme podle tzv. formátovacích skupin. Nejběžnější grafické sestavy využívají skupinu grr (obchodné grafika), existují ale skupiny pro Word, Excel apod. Většina následujícího dokumentu se věnuje právě popisu grr formátovací skupiny, prostor je věnován ale i ostatním skupinám a dalším vlastnostem systému.

Tento předpis obvykle nevzniká vytvořením v textovém editoru (ačkoliv je to možné), ale pomocí k tomu určené aplikace "Návrhář sestav" (GINIS modul GFE).

Příklad souboru s předpisem pro formátování sestavy:

```
<?xml version="1.0" encoding="windows-1250"?>
<format xmlns="http://www.gordic.cz/TR/alf/1.3/" type="grr">
  <info ixm_frm="0000FRM00000"
    nazev="Faktura odb ratelská"
    ixm_alv="0000ALV00000"
    ixm_xme="0000XME00000"
    xmeta_ver="1"
    xmeta_subver_min="0"
    popis="..."
  />
  <region name="faktura">
    <head>
      <style font-underline="true">
        <line> <text value="Faktura odb ratelská"/> </line>
      </style>
    </head>
    <body>
      <line>
        <value-of name="nazev_odberatele"/>
        <value-of name="adresa"/>
        <value-of name="mesto"/>
        <value-of name="psc"/>
      </line>
      <region name="radek_faktury">
        <head>
          <line> <text value="Položky faktury :"/> </line>
        </head>
        <body>
          <line>
            <value-of name="cislo_polozky"/>
            <value-of name="popis_polozky"/>
            <value-of name="fakturovana_castka"/>
          </line>
        </body>
        <foot>
          <line/>
        </foot>
      </region>
    </body>
  </region>
</format>
```

3.4 Základy

Všechny soubory jsou ve formátu XML a používají tzv. atributovou syntax. To znamená, že veškeré údaje jsou zapsány do atributů a musí být v uvozovkách nebo apostrofech. Obsahuje-li údaj uvozovku, je vhodnější jej uzavřít do apostrofů a opačně. Žádný údaj nesmí obsahovat znaky <, > nebo &. Je-li třeba zapsat tyto znaky, je nutné použít zástupných řetězců dle této tabulky:

<	<
>	>
&	&
"	"
'	'
od ádkování	

Soubory musí začínat značkou:

```
<?xml version="1.0" encoding="windows-1250" ?>.
```

Za ní musí následovat kořenová značka (viz tabulky dále). Teprve v této kořenové položce se mohou kombinovat další značky.

3.5 Stránkování

Každá oblast může mít žádnou nebo právě 1 značku `head` a žádnou nebo 1 značku `foot`.

V jedné značce `head` nebo `foot` může být několik řádků s označení `on-each-page` a několik bez tohoto příznaku. Stránková hlavička resp. patička pak bude obsahovat všechny řádky s příznakem (v daném pořadí) a běžná hlavička resp. patička všechny řádky (bez ohledu na příznak).

Existuje ještě speciální možnost příznaku `on-each-page=middle`, která vyjme řádek z běžných hlaviček/patiček. To lze použít např. pro zdůraznění, že se jedná o pokračování z předchozí strany apod.

Stránkovací algoritmus pracuje takto:

- při otevření oblasti v datovém souboru vytvoří a uschová stránkové hlavičky a patičky (tj. řádky z `head` a `foot` s příznakem `on-each-page`) a vytiskne se běžná hlavička
- tisknou se běžné řádky oblasti
- dojde-li během zpracování oblasti na konec stránky, vytiskne se stránková patička a na další stranu stránková hlavička
- na konci oblasti se vytiskne běžná patička
- řádky označené `break-page-before=avoid` nebo `break-page-after=avoid` zabraňují vzniku zlomu stánky před, resp. po dané řádce. Řádky v hlavičce mají (není-li uvedeno jinak) nastaveno potlačení zlomu za a řádky v patičce před řádkem (zaručuje, že hlavička ani patička nebude rozdělena na více stran)

Všimněte si rozdílu mezi stránkovou a běžnou hlavičkou. Řádky označené `on-each-page=true` jsou zařazeny jak do běžné tak do stránkové hlavičky!

Hlavičky a patičky mohou obsahovat datové pole ze všech nadřazených oblastí i z oblasti samotné. Obsahuje-li běžná hlavička datovou položku z dané oblasti, je použito 1. hodnoty z dat. Pro běžnou patičku se použije poslední datová hodnota. Stránkové hlavičky používají 1. hodnotu na příslušné stránce, stránkové patičky poslední hodnotu na příslušné stránce.

Stránkové patičky lze navíc umístit buď hned za poslední tištěný řádek na stránce, nebo těsně před konec stránky (rozhoduje atribut `endpage`). Doporučuji tento atribut nastavit či nenastavit shodně pro všechny řádky z patičky, jinak může dojít k nežádoucím efektům nebo dokonce chybě při zpracování sestavy.

Finta: příznaku `middle` lze použít pro odstranění nadbytečné patičky, např. po tvrdém odstránkování (`break-page-after`). Další možnost je použít pro odstránkování samostatného řádku s výškou `0mm`, který nikdy netvoří samostatnou stranu.

3.6 Číslování stran

Pole `ROOT.PAGE` a `ROOT.PAGES` obsahují číslo strany a celkový počet stran sestavy jako celku. Pozor, tyto pole nelze použít pro výpočty! Lze je ale použít v operacích s řetězci.

Pro hromadné tisky, kde je každý dokument číslován samostatně (např. hromadný tisk faktur), lze použít číslování v rámci oblasti. Pro číslo strany je to funkce `PAGE-NUM`, pro počet stran `PAGE-COUNT`. Ani tyto funkce nelze použít pro aritmetické výpočty.

Příklady:

```
<value-of name="OBJ-1.PAGE-NUM()" />
```

```
<value-of name="" onData="self.value=ZAHHLAVI.PAGE-NUM()+ ' / ' +ZAHHLAVI.PAGE-COUNT()" />
```

3.7 Stránkový součet

Součet hodnot tištěných na konkrétní straně lze zajistit speciální funkcí `PAGE-SUM(hodnota)`. Sčítaná hodnota musí být typu číslo (`number`). Výsledná hodnota funkce nelze použít v dalších aritmetických operacích! Zápis funkce je obdobný ostatním [agregačním funkcím](#).

Příklady:

```
<value-of name="OBJ-1.PAGE-SUM(POC)" />
```

3.8 Rozdělení textu do více řádků

Pokud je zobrazovaný text delší, než je šířka buňky dochází k jeho rozdělení na více řádků, případně jeho zkrácení. Lze řídit několika atributy:

1. `multiline` (víceřádkový text) pokud je `false` tak bude buňka vždy mít jen jednu řádku. Ať je dlouhá jak chce. Při `true` se buňka natáhne na výšku tak, aby se tam text vešel. I kdyby to mělo být deseti řádkové;
2. pokud je nastavena výška řádku na pevnou hodnotu (např. 20mm), pak se udělá pouze tolik řádků, kolik se vejde do této výšky;
3. když už je známý počet řádků, lze ještě nastavit, jak se má zakončit buňka, jejíž text se nevejde celý. Možnosti jsou 3: useknout, zakončit třemi tečkami, nebo celý text vyplnit jedním znakem (výchozí je hvězdička);
4. celé je to ještě rozšířené o to, že v textu samotném může být znak (dvojnák) pro odřádkování;
5. Rozdělení textu do řádků probíhá po slovech. Pokud není v textu žádná mezera, tak se vezme co nejvíce znaků.

Odřádkování v textové buňce v ALFu pomocí tzv. character reference `
`:

```
<text value="první ádek&#10;druhý ádek" />
```

V datech pomocí dvojznaku `\n` takto:

```
POLOZKA|120110|Klávesnice ESCOM\nUS/CZ|ks|700|154.00|
```

3.9 Skriptování

Pro skriptování se využívá jazyka Python, včetně většiny jeho knihoven, možnosti jsou tedy velké. Kromě základních konstrukcí jako smyčky, podmínky apod. má tento jazyk zajímavé možnosti u polí a slovníků. Pozor: veškerý zápis ve skriptech *rozlišuje velká a malá písmena!*

V tabulce specifikace ALF (níže) jsou uvedeny jména atributů, které jsou dostupné ve skriptu. Značka, ve které je skript zpracováván je dostupná pod jménem `self`. Jména atributů se většinou shodují se jmény atributů ve vstupním `alf` souboru, kde znak "-" (pomlčka) je nahrazen "_" (podtříčko). Některé atributy jsou oproti vstupnímu souboru přidány (např. `value` nebo `style`).

Dále se skriptovacího jazyka používá i pro vyhodnocení výrazu – pozor, zde není možné použít příkazy jazyka, jen aritmetiky, funkcí a metod objektů. Je-li zapotřebí složitější konstrukce, je možno vytvořit vlastní funkce (viz [odstavec skript](#)). Více také kapitola [Podmíněné zobrazení](#).

Dále je všude dostupný objekt `format`, který shrnuje několik informací dostupných v celém `alf` (popis viz dále).

Některé atributy jsou pouze pro čtení (označeny R), některé pouze pro zápis (W), většinu lze použít v obou případech (RW).

Skripty rozpoznávají některé základní typy dat:

řetězce (`str`)

čísla (`num`) – pouze několik vestavěných atributů

čísla `Decimal` – takto se předávají data označená `datatype=number` (viz kapitola [typ číslo](#))

datum/čas `Datetime` – předání při `datatype=datetime` (viz kapitola [typ datum a čas](#))

Skript může být vyvolán na některou z událostí. Jsou to atributy některých značek začínající předponou "on".

Skript **onLoad** se zpracuje hned při načtení `alf` souboru do paměti. Z tohoto důvodu je dostupné pouze `self`.

Skript **onData** se zpracovává vždy jednou pro každou hodnotu z datového souboru a to při načítání hodnoty, která tomuto políčku přísluší. Tato hodnota je dostupná jako `self.value`. K dispozici je objekt `self` a odkazy na všechna načtená datová pole z otevřené oblasti i všech nadřazených. K nim je nutno přistupovat vždy přes jméno oblasti (`oblast.hodnota`). Pro tento skript také často lze použít zvláštní syntax značky `value-of` s prázdným atributem `name`. Skript lze použít jak pro změnu načítané hodnoty, tak pro změnu stylu. Nejpoužívanější skript.

Skript **onEnter** se zpracovává vždy jednou po vstupu do oblasti, ve které je značka obsahující tento skript. Zde je vhodné měnit formátování buňky (např. `self.style.width`). Je třeba se však uvědomit, že styl sdílí spolu všechny výskyty této buňky, takže nastavení šířky se bude týkat celého sloupce obsahující tuto buňku. Pokud navíc je oblast vnořena v jiné opakující se oblasti, bude skript vyvolán vícekrát, styl se však uplatní pouze jeden – poslední změněný. K dispozici je i první načtená hodnota `dat`, nicméně se asi nenajde skript, který by ji rozumně využil. Datová pole z nadřazených oblastí se však hodit mohou.

Skript **onPrint** se je shodný s **onData**. Pouze pro zpětnou kompatibilitu. V nově vytvářených sestavách nepoužívat.

Pole `ROOT.PAGE` a `ROOT.PAGES` obsahují během zpracovávání skriptů zástupný řetězec, který bude nahrazen skutečnou hodnotou až bude známa. Tyto pole tak nelze použít pro výpočty, ale lze je použít v operacích s řetězci.

Příklady:

```
<value-of name="CEN" onData="self.value=self.value.format('###,00')"/>
<value-of name="" onData="self.value=info.nazev+' '+info.nazev2"/>
<value-of name="" onData="self.value=OBJ.CEN*100"/>
<text value="pokus" onLoad="self.style.align='center'"/>
<text value="pokus" onEnter="self.style.align=hlavicka.zarovnani"/>
```

Značka **SCRIPT**

Smí být pouze na nejvyšší úrovni (přímo pod `<format>` stejně jako `<info>`). Uvnitř značky jako text lze umístit libovolné skripty, určeno je však především pro uživatelské funkce. Pro zjednodušení zápisu lze použít syntaxe `CDATA`.

Příklad:

```
<script><![CDATA[
def redblue(self):
    if self.value<0:
        self.style.font_color='red'
    else:
        self.style.font_color='blue'
]]></script>
```

Takovou funkci pak lze volat kdekoli ze skriptů, např. takto:

```
<value-of name="CEN" format="# 0,00;# 0,00-" onPrint="redblue(self)" />
```

Všechny datové položky, které jsou třeba uvnitř takové funkce, je nutné si tyto informace předat do funkce parametrem. V jednom bloku `script` je možno definovat i více funkcí případně jiných struktur.

Objekt **FORMAT**:

Objekt `format` obsahuje tyto položky:

<code>ver_major</code>	velké číslo verze alf
<code>ver_minor</code>	malé číslo verze alf
<code>formatting_group</code>	formátovací skupina. Např. "GRR"
<code>Info</code>	seznam info sekcí. Jednotlivé info jsou dostupné pomocí řetězcového indexu např. <code>self.value=format.info['nazev']</code>
<code>Paper_width</code>	Šířka strany nastavená v <code>paper-setting</code>
<code>Paper_height</code>	výška strany nastavená v <code>paper-setting</code>
<code>Paper_margin_top</code>	horní okraj nastavený v <code>paper-setting</code>
<code>Paper_margin_bottom</code>	spodní okraj nastavený v <code>paper-setting</code>
<code>Paper_margin_left</code>	levý okraj nastavený v <code>paper-setting</code>
<code>Paper_margin_right</code>	pravý okraj nastavený v <code>paper-setting</code>

Objekt se rovněž používá ve skriptech pro přístup ke sdíleným stylům podle jména (viz kapitola [Sdílení stylů](#)).

Další postřehy:

Porovnání, zda některý sloupec je určité hodnoty

```
<value-of name="DOD_ICO" onData="if LIK_HLA.DOD_TES == '20': self.value=LIK_HLA.DOD_RC"/>
```

zde jsou nutné apostrofy kolem hodnoty '20' jelikož `DOD_TES` je řetězcová hodnota.

Při `LIK_HLA.DOD_TES==20` dojde k chybě!

Pro porovnání čísel je nutné použít typ `number`

```
<item name="DOD_TES" datatype="number"/>
```

a pak lze psát

```
<value-of name="DOD_ICO" onData="if LIK_HLA.DOD_TES == 20: self.value=LIK_HLA.DOD_RC"/>
```

Porovnání s klausulí `else` (jinak)

```
<value-of name=""
onData="if LIK_HLA.DOD_TES=='20': self.value='R : '&#10;else: self.value='I O: '"/>
```

klausule `else` musí být na následujícím řádku, nelze ji oddělit středníkem.

K tomu lze použít speciální znak `
` nebo použít volání připravené funkce (viz značka [<script>](#))

Při porovnání raději používáme znaménko `>` než `<`. `<` je totiž v XML zakázané a musí se zapisovat jako `<`;

```
<value-of name="DOD_ICO"
onData="if LIK_HLA.DOD_TES &lt; 20: self.value=LIK_HLA.DOD_RC"/>
```

```
<value-of name="DOD_ICO"
  onData="if 20 > LIK_HLA.DOD_TES : self.value=LIK_HLA.DOD_RC"/>
```

Pro porovnání čísel je samozřejmě nutné použít typ `number`.

Více o syntaxi skriptu v kapitole [Pokročilé skriptování](#).

3.10 Datové typy

3.10.1 Řetězec

Řetězcové proměnné jsou všechny, které nemají přiřazen jiný typ. Lze na ně použít základní metody jazyka **Python**. Jsou to hlavně tyto:

Část řetězce (tzv. slice)	Prom[od:do]	VOLIC.PRIJ[0:2]	Číslováno od 0! Znaky od až do-1 (0. a 1.)
	Prom[od:]	VOLIC.PRIJ[0:]	Vše od znaku do konce
	Prom[:do]	VOLIC.PRIJ[:2]	Vše od začátku do znaku
Znak řetězce	Prom[znak]	VOLIC.PRIJ[0]	Jeden znak na určené pozici
find	Prom.find(co)	VOLIC.PRIJ.find('A')	Vrací index 1. výskytu podřetězce nebo -1
lower	Prom.lower()	VOLIC.PRIJ.lower()	Převéde na malá písmena
upper	Prom.upper()	VOLIC.PRIJ.upper()	Převéde na velká písmena
replace	Prom.replace(co,n	VOLIC.PRIJ.replace('A',	Nahradí všechny výskyty podřetězce jiným řetězcem (může být kratší, delší i prázdný)

Velmi zajímavé jsou také dvě funkce pracující s poli řetězců: `split` a `join`. První rozdělí řetězec podle oddělovače na pole řetězců, druhá zase takové pole umí sloučit zpět. Např. rozdělení řetězce obsahující jméno i příjmení (oddělené mezerou) je velmi jednoduché:

```
KRESTNI = VOLIC.JMENO.split()[0]
PRIJMENI = VOLIC.JMENO.split()[1]
```

Více viz [dokumentace jazyka Python](#).

3.10.2 Číslo

Ve struktuře může být položka oblasti definována takto:

```
<item name="CEN" title="Cena" datatype="number"/>
```

V takovém případě bude tato položka chápána číslo. To znamená, že kromě běžných operací (s položkami typu řetězec) jsou k dispozici některé další.

1. taková položka je do skriptů přenášena jako typ `Decimal`. (ať už ze `self.value` nebo z `oblast.jmeno`)

2. u značky `value-of` je možné přidat atribut `format` pro formátování

3. přenos do Excelu jako čísla

Bod 1 má za následek i možnost aritmetických operací bez nutnosti přetypování. Např.

```
<value-of name="" onData="self.value=OBJ.CEN*100"/>
```

Pokud by položka nebyla typu číslo (ale výchozí řetězec) byl by výsledek operace jiný (bylo by to 100 násobné opakování řetězce).

Převod řetězce na číslo je samozřejmě také možný, provede se pomocí konstruktoru `Decimal`(řetězec), např.:

```
<value-of name="" onData="self.value=Decimal('12345.67')*100"/>
```

Formáty pro číselný typ

Pro číselné sloupce je možné využít formátování pomocí atributu `format`, např.:

```
<value-of name="CEN" format="# 0,00;# 0,00-"/>
```

Formátování probíhá až po provedení skriptu **onData** (a před **onPrint**) a může být zcela ekvivalentně nahrazeno tímto skriptem takto:

```
<value-of name="CEN" onData="self.value=self.value.format('# 0,00;# 0,00-')"/>
```

Samotná syntax je inspirována způsobem formátování číselných buněk v `Microsoft Excel` (kódy *nejsou* zcela shodné s MS Excel). Formát může mít tři oblasti oddělené středníkem. První oblast definuje vzhled kladných čísel, druhá je pro čísla záporná, třetí pro nulu. Formát nutně nemusí mít všechny tři oblasti. V případě pouze jedné části se pro zobrazení použije jediná oblast s případným znaménkem vpřed. Pro nulu se používá první oblast i v případě dvou oblastí. Celá část čísla se nikdy nekrátí, desetinná část se uřízne (nezaokrouhlí) podle počtu zástupných znaků.

Jednotlivé oblasti se skládají ze zástupných znaků. Jejich význam je v následující tabulce:

Kód formátu:	Chcete-li zobrazit:
,	oddělení desetinné části čárkou
.	oddělení desetinné části tečkou
#	zastupuje jednu číslici ze vstupního čísla. netiskne zbytečné nuly
0	zastupuje jednu číslici ze vstupního čísla. tiskne i zbytečné nuly
(mezera)	vloží oddělovač tisíců (vždy po 3 místech)
_ (podtřítko)	vloží mezeru o velikosti znaku za _ (ten znak se nevloží)
\ (zpětné lomítko)	ruší význam následujícího znaku

Kód formátu:

Chcete-li zobrazit:

ostatní znaky

normálně se vloží do výstupu

Příklady:

číslo	formát	výsledek
1000000	#,##	1000000
1234567,2	# #,##	1 234 567,2
-123	#,##	-123
34567,2	#,00	34567,20
0,2	#,00	,20
0,02	#0,##	0,02
-123	# 0,00;# 0,00-	123,00-
-123	# 0,00_-;# 0,00-	123,00-

Pro běžné formátování jsou vhodné formáty:

číslo se dvěma des.místy, čárkou, znaménko vpředu	# 0,00
číslo se dvěma des.místy, čárkou, znaménko vzadu	# 0,00_-;# 0,00-
bez oddělení tisíců	0,00
bez desetinných míst, pokud je číslo celé	# 0,##

Dále je možno použít modifikátorů, které se mohou vyskytovat ve všech částech řetězce. Modifikátory mohou měnit nastavení pro formátování čísel (oddělovače, délky apod.) nebo styl buňky (barvu, zarovnání apod.). Modifikátor má syntaxi [jméno=hodnota]. Jméno se může vypustit a pak se jedná o nastavení barvy. Na pořadí modifikátorů nezáleží. Např. :

```
format="[gsep=,][align=center]# ##0.####;[red][gsep=,](# ##0.####)"
```

Způsobí obarvení záporných čísel na červenou, kladná budou zarovnaná na střed a všechna čísla s tisíci oddělené čárkou.

[gsize=...]	počet míst pro oddělení skupin (tisíců) (def. 3)
[gsep=...]	oddělovač skupin (tisíců) (def. mezera)
[dsep=...]	oddělovač desetin (def. je závislý na použitém znaku . nebo ,)
[...]	barva písma (shodné s [font-color=...])
[xxx=...]	nastavení vlastností stylu. pro seznam vlastností viz tabulka pro <style>

Funkce třídy Decimal (ve skriptech)

konstruktor(int)	konverze z integeru
konstruktor(float)	konverze z floatu

<code>konstruktor(string)</code>	konverze ze stringu (využívá <code>reporter</code>)
<code>format(format)</code>	formátování čísla (shodné s atributem <code>format</code>)
<code>spellOut (int rod=0)</code>	převod čísla na text (např. částka slovy); obdoba <code>NumToWords</code> z ALV. Parametrem <code>rod</code> lze nastavit <code>0</code> =ženský (koruna) <code>1</code> =mužský rod (haléř)
<code>toWords(int rod=0)</code>	synonymum k <code>spellOut</code>
<code>round(int mist=0)</code>	zaokrouhlí na daný počet míst (není-li parametr uveden zaokrouhlí na celé číslo)
<code>floor()</code>	vrací nejbližší menší číslo (pro kladná čísla ořízne desetiny)
<code>ceil()</code>	vrací nejbližší vyšší číslo (pro kladná čísla ořízne desetiny a přičte 1)

3.10.3 Datum a čas

Ve struktuře může být položka oblasti definována takto:

```
<item name="DAT" title="Datum sjednání" datatype="datetime"/>
```

V takovém případě bude tato položka chápána datum a čas. To znamená, že kromě běžných operací (s položkami typu řetězec) jsou k dispozici některé další.

1. taková položka je do skriptů přenášena jako typ `Datetime`. (ať už ze `self.value` nebo z `oblast.jmeno`)
2. u značky `value-of` je možné přidat atribut `format` pro formátování
3. do Excelu se přenáší jako datum
4. provádí se základní formátování takových položek pokud není uveden atribut `format (dd.mm.yyyy)`

Formáty pro datum a čas

Pro datumové a časové sloupce je možné využít formátování pomocí atributu `format`, např.:

```
<value-of name="DAT" format="mmm yy"/>
```

Formátování probíhá až po provedení skriptu **onPrint** a může být zcela ekvivalentně nahrazeno tímto skriptem takto:

```
<value-of name="DAT" onPrint="self.value=self.value.format('mmm yy')"/>
```

Samotná syntax je inspirována způsobem formátování datumových buněk v Microsoft Excel (kódy nejsou zcela shodné s MS Excel)

Kód formátu:	Chcete-li zobrazit:
<code>yy</code>	Roky jako čísla od 00 do 99
<code>yyyy</code>	Roky jako čísla od 1900 do 9999
<code>m</code>	Měsíce jako čísla od 1 do 12

<i>Kód formátu:</i>	<i>Chcete-li zobrazit:</i>
mm	Měsíce jako čísla od 01 do 12
mmm	Měsíce jako zkratka textu
mmm	Měsíce jako text leden-prosinec
mmmm	Měsíce jako text ledna-prosince
mmmmmm	Měsíce jako římské číslice od I do XII
d	Dny jako čísla od 1 do 31
dd	Dny jako čísla od 01 do 31
ddd	Dny jako zkratky názvů dnů po-ne
dddd	Dny jako text pondělí-neděle
H	Hodiny jako čísla od 0 do 23
HH	Hodiny jako čísla od 00 do 23
M	Minuty jako čísla od 0 do 59
MM	Minuty jako čísla od 00 do 59
S	Sekundy jako čísla od 0 do 59
SS	Sekundy jako čísla od 00 do 59
ostatní znaky	se normálně přepíše do výstupu

Poznámka: zkratky a texty jsou závislé na operačním systému a nastavení místního prostředí. Tj. na anglických Windows vracejí anglické zkratky!

Funkce třídy `DateTime`

<code>konstruktor(string)</code>	konverze ze stringu (využívá <code>reporter</code>)
<code>konstruktor(tuple)</code>	konverze z vnitřní formy modulu <code>time</code>
<code>format(format)</code>	formátování datumu a času (pro atribut <code>format</code>)

3.11 Podmíněné zobrazení

Zobrazení některých informací (řádků nebo celých oblastí) lze podmínit splněním některé podmínky. Např.

```
<line only-if="OBJ_1.CEN==1000" on-each-page="true">  
<line only-if="OBJ_1.CEN>=1000" on-each-page="true">
```

zobrazí řádku pouze v případě, že cena je rovna (vyšší nebo rovna) 1000 (položka CEN je typu číslo).

Obdobně pro oblasti:

```
<region name="RZP" only-if="OBJ_1.DAT<ROOT.DATE"> (< nutno zapsat jako &lt;)
```

zobrazí oblast RZP pouze v případě, že pole datum objednávky je nižší než datum tisku. Lze

použít i první data z oblasti, ke které se podmínka vztahuje, nicméně ve většině případů se používají data z nadřazených oblastí.

V obou případech je v atributu `only-if` uveden výraz, jehož výslednou hodnotou musí být hodnota `boolean`. Lze použít i funkci. Odkazy na datové položky musí být jako všude ve skriptech uvedeny se jménem oblasti.

Pro oblasti lze použít filtrování dat v dané oblasti. To vyloučí ze zpracování všechny řádky vyhovující podmínce (pozor na obrácený význam podmínky!). Výraz se vyhodnocuje pro každý řádek zvlášť.

```
<region name="SML_POL" filter-out="SML_POL.AN01==''">
```

Více o syntaxi výrazů v kapitole [Pokročilé skriptování](#).

3.12 Třídění dat

Položky oblasti lze setřídít podle některého údaje v této oblasti. Například

```
<region name="OBJ-1" order-by="DAT">
```

setřídí všechny objednávky dle datumu. Třídění lze provádět podle:

- řetězce (typ `string`). Třídění probíhá podle třídícího uspořádání z OS (české třídění u českých Windows). Velikost písmen se neuvažuje (lze ovlivnit modifikátorem – viz. níže)
- čísla (typ `number`). Čísla se třídí dle velikosti v běžném matematickém pojetí
- datumu a času (typ `datetime`). Starší data jsou před novějšími daty

Kritérium je možno rozšířit o další specifikace:

- `<region name="OBJ-1" order-by="DAT DESC">` třídí dle datumu sestupně (obráceně)
- `<region name="OBJ-1" order-by="NAM CASE">` třídí dle názvu s ohledem na velikost písmen

Lze také třídít podle více kritérií. Např.

```
<region name="OBJ-1" order-by="CEN DESC,DAT,NAM">
```

setřídí dle ceny sestupně. Objednávky stejné ceny jsou pak setříděny dle data vzestupně a případné další shody abecedně podle jména.

Upozornění: třídění probíhá na klientské stanici a pro větší množství dat může být velmi pomalé! Pokud tříděná oblast obsahuje ještě podoblasti dochází k dalšímu významném zpomalení. Používejte proto třídění uvážlivě a pouze pro oblasti s rozumným počtem dat.

3.13 Počítané proměnné

```
<variable name="JMENO" value="1+1"/>
<variable name="JMENO" value="1+1" datatype="number"/>
```

`name` určuje jméno definované proměnné. `value` je skript, který se vyhodnotí jako výraz a jeho výsledek je hodnota dané proměnné. Vyhodnocuje se po každém načtení dat. K dispozici jsou všechna načtená datová pole (současná i všechny nadřazené oblasti).

Datový typ není nutno uvádět – zjistí se z výsledku prvního volání výrazu. Pokud je uveden, musí výraz vracet správný typ výsledku.

Proměnnou je možno použít všude tam, kde se používá odkaz na datovou položku. Je vlastně jakousi přidanou datovou položkou. Ve skriptech je nutno ho uvádět plným jménem (se jménem oblasti) stejně jako běžné datové položky.

Příklady:

```
<value-of name="JMENO" />
<line only-if="OBLAST.JMENO==2">
<value-of name=" " onData="self.value = funkce(OBLAST.JMENO)" />
```

3.14 Agregace

Jsou tři druhy agregačních funkcí : průběžné, `total` a `forward`.

Průběžné počítají průběžný počet, součet, apod. Tj. v hlavičce je součet 0, v patičce součet všech, na jednotlivých řádcích je součet předchozích řádků (bez aktuálního).

`total` funkce používají dvou průchodů přes data a zjistí počet, součet apod. již při vstupu do oblasti. Součet pak lze umístit např. v hlavičce či jeho hodnotou podmínit tisk oblasti atd.

`forward` funkce zjišťují počet, součet apod. v podřízených oblastech. Lze např. zjistit počet a součet položek za fakturu atd. Do těchto funkcí se musí uvést jako parametr jméno oblasti a jméno položky (pro count pouze jméno oblasti).

Poznámka: agregační funkce nejsou pravé funkce, spíše jakési pseudofunkce, jelikož jejich parametry nejsou vyhodnocovány, ale použity pro agregaci.

<i>jméno ag. funkce</i>	<i>popis</i>	<i>příklad</i>
ROWNUM	průběžný počet (číslo řádku)	<code><value-of name="ROWNUM" /></code>
SUM	průběžný součet	<code><value-of name="SUM(CENA)" /></code>
MIN	průběžné minimum	<code><value-of name="MIN(CENA)" /></code>
MAX	průběžné maximum	<code><value-of name="MAX(CENA)" /></code>
TOTAL-COUNT	celkový počet	<code><value-of name="TOTAL-COUNT" /></code>
TOTAL-SUM	celkový součet	<code><value-of name="TOTAL-SUM(CENA)" /></code>
TOTAL-MIN	celkové minimum	<code><value-of name="TOTAL-MIN(CENA)" /></code>
TOTAL-MAX	celkové maximum	<code><value-of name="TOTAL-MAX(CENA)" /></code>
FORW-COUNT	dopředný počet	<code><value-of name="FORW-COUNT(ROZPIS)" ,</code>
FORW-SUM	dopředný součet	<code><value-of name="FORW-SUM(ROZPIS.POCI</code>
FORW-MIN	dopředné minimum	<code><value-of name="FORW-MIN(ROZPIS.POCI</code>
FORW-MAX	dopředné maximum	<code><value-of name="FORW-MAX(ROZPIS.POCI</code>

Poznámky:

- `SUM()` v hlavičce je vždy 0
- `ROWNUM` v hlavičce je vždy 1
- `MIN` v hlavičce je vždy $+\infty$ ($+\text{INF}$)
- `MAX` v hlavičce je vždy $-\infty$ ($-\text{INF}$)
- `ROWNUM`, `SUM`, `MIN` a `MAX` mají v patičce stejnou hodnotu jako `TOTAL-COUNT`, `TOTAL-SUM`, `TOTAL-MIN` resp. `TOTAL-MAX`

Existuje také speciální funkce pro stránkový součet `PAGE-SUM`, více viz. kapitola [stránkový](#)

[součet](#).

3.15 Skupinování

Skupiny rozdělují daná data oblasti podle kritéria do skupin. Při změně hodnoty položky kritéria se mění skupina a také všechny podřízené skupiny. Při změně skupiny se tiskne patička staré skupiny a hlavička nové (pokud jsou). Rovněž se počítají všechny agregační funkce za danou skupinu.

Skupiny jsou předřazeny před `<body>`. Samotné tělo je posunuto do oblasti. Je možné je také vnořit do sebe (více skupin). Není možné definovat více skupin vedle sebe (musí být striktně hierarchické).

```
<region name="(jméno oblasti)">
  <head>
    (hlavička oblasti)
  </head>
  <group by="(kritérium)">
    <head>
      (hlavička skupiny)
    </head>
    <body>
      (tělo)
    </body>
    <foot>
      (patička skupiny)
    </foot>
  </group>
  <foot>
    (patička oblasti)
  </foot>
</region>
```

Kritérium je jméno datové položky (nebo proměnné) podle které se skupinuje. Lze skupinovat i za více položek v jedné skupině, položky se oddělují čárkami. Lze skupinovat i podle počítané proměnné.

Jméno oblasti slouží pro odkaz na ní ve `<value-of>` a ve skriptech. Není povinné.

Skupina může obsahovat svojí hlavičku, patičku, další podskupinu nebo tělo. Nemůže mít tělo a současně další skupinu nebo více podskupin.

Ve skupinách je možno použít všech agregačních funkcí stejně jako v bez skupin. Je možné se i odkazovat na funkce z jiných skupin nebo z oblasti samotné (pro odkaz na jiné skupiny je nutno tyto skupiny pojmenovat atributem `name`). Jména skupin musí být jedinečné a nesmí kolidovat ani se jmény oblastí. Je možné i navrhnout sestavy, která má prázdné body a veškeré výpisy probíhají pouze v patičkách/hlavičkách skupin. Taková sestava pak kumuluje data a vypisuje pouze kumulace, nikoli data samotná. Zpracování je pochopitelně delší než výpis řádků samotných.

Příklad:

```
<region name="KARTA" order-by="NKS,SUAUEVIDENCE">
  <group by="NKS">
    <group by="SUAUEVIDENCE">
      <head>
        <line>
          (hlavička skupiny)
        </line>
      </head>
      <body>
        <line>
          (tělo)
        </line>
      </body>
    </group>
  </group>
</region>
```

```

        </line>
    </body>
</foot>
    <line>
        <text value="Celkem za ú et:"/>
        <value-of name="SUAUEVIDENCE"/>
        <value-of name="SUM(MNOZSTVI)" format="# #,000"/>
        <value-of name="SUM(PORIZCENA)" format="# #,00"/>
    </line>
</foot>
</group>
</foot>
    <line>
        <text value="Celkem " />
        <value-of name="SUM(MNOZSTVI)" format="# #,000"/>
        <value-of name="SUM(PORIZCENA)" format="# #,00"/>
    </line>
</foot>
</group>
</region>

```

3.16 Sdílení stylů

Pokud několik buněk (či dokonce několik řádků) sdílí styl takto:

```

<style ...>
    <value-of .../>
    <value-of .../>
    <value-of .../>
</style>

```

je tento jeden styl přiřazen všem buňkám současně. Pokud však takový styl změním skriptem, změna se promítne do všech buněk! Lze tak např. jednoduše změnit celý řádek najednou apod.

Sdílené styly mohou být i hierarchické např.

```

<style id="S1" ...>
    <style ...>
        <value-of .../>
    </style>
    <style ...>
        <value-of .../>
    </style>
</style>

```

Změna stylu `s1` pak změní prezentaci obou datových položek a přesto každá z nich může mít jiné vlastnosti definované v podřazených stylech.

Pojmenování stylů je možné (pomocí atributu `id`) a pomocí jména se pak lze odkazovat ve skriptech na konkrétní styl. Na styly se odkazuje pomocí syntaxe `format.S1.xxx`. Pojmenované styly je také možné vytknout (např. před celý `alf`) a přidružit styl některé buňce pomocí atributu `style`.

```

<style id="S1" ... />
...
<value-of style="S1" .../>

```

3.17 Obrázky

Nastavení velikosti a způsobu ořezávání obrázku

```

<image file="logo.bmp" image-width="35mm" image-height="35mm"/>

```

Pro obrázky vkládané do sestav je vhodné nastavit atributy `image-width` a `image-height` tak, aby obrázek byl vhodně veliký a zapadl do sestavy. Lze také použít speciální hodnoty atributů `auto`, která přizpůsobí obrázek dle velikosti buňky. Pokud atributy nejsou uvedeny, je velikost převzata z fyzické velikosti obrázku a pro různé obrázky pak bude obrázek různě velký i v tisku (nedoporučuje se). Lze zadat pouze jeden z atributů. Obrázky zachovávají poměr stran, pokud nejsou specifikovány oba atributy. Poměr stran se zachovává i pro hodnotu `auto`.

Globální obrázky

```
<image file="logo.bmp" global="true" image-width="35mm" image-height="35mm"/>
```

Pro globální obrázky (vhodné pro logo, erb apod. organizace) je natavení velikosti ještě důležitější. Obrázek se změní pro jiné zákazníky/jiný ORG v aplikaci a rozměry obrázků organizací nemusí být shodné. Zde toto nastavení je v podstatě nutností.

Samotné globální obrázky se neukládají spolu se sestavou ale jejich úložiště je závislé na volající aplikaci. Pro použití globálních obrázků je tedy nutná podpora aplikace (viz příslušní autoři/dokumentace apod.)

3.18 Nastavení velikosti stránky pomocí paper-setting

Pokud je sestava vytvářena na přesně daný formulář, doporučuji všechny rozměry zadávat v mm. V takovém případě, je však také vhodné určit, pro jakou velikost papíru je sestava určena. To lze přímo v .alf pomocí značky

```
<paper-setting paper-width="210mm"/>
```

nebo

```
<paper-setting paper-width="210mm" left-margin="5mm" right-margin="5mm"/>
```

Jsou 3 možnosti:

- bez uvedení značky: prohlížeč použije první grafickou předvolbu na výchozí tiskárně
- pouze rozměr papíru: použije se taková předvolba, s takto velikou stránkou. Velikost tisknutelné oblasti však bude menší a záleží na nastavení předvolby o kolik. Je vhodné použít jako minimální rozměr stránky při zadávání šířek v %.
- i okraje: nalezne se taková předvolba, jejíž papír je žádaného rozměru a okraje upraví se na zadané rozměry. Pozor: tisknutelná plocha je pak o tyto okraje menší! Součet všech šířek by tudíž měl být přesně (nebo méně než) `width - left - right`. Tuto možnost použijte jen ve výjimečných případech. Je možné také nastavit jen některé z okrajů a zbylé nechat na nastavení prohlížeče.

Upozornění: nenastavovat okraje na příliš malé hodnoty! Na některých tiskárnách pak nemusí být sestava tisknutelná. Minimálně doporučuji tyto čísla - `left`, `right`, `top` 5mm ; `bottom` 15mm.

Pro všechny sestavy, kde je nějaký rozměr zadán v mm, doporučuji do záhlaví uvést minimální šířku a výšku stránky, případně i nastavení okrajů. Třetí možnost doporučuji pouze v zakázkových sestavách nebo ojedinělých případech, kde si můžete být jisti, že takové okraje jsou pro danou tiskárnu přípustné. Výchozí nastavení prohlížeče počítá s okraji 10mm, což většina tiskáren zvládá (kromě spodního okraje, který je u některých typů až 25mm). Pro sestavy zadávané v mm, kde součet šířek je nejvýše 190mm je šířka strany 210mm přípustná. Pro širší sestavy je nutno nastavit buď okraje nebo širší papír (např. A4 naležato).

Poznámka: rozměr 0mm je vyhrazen pro chování jako by nebyl zadán. Okraje tudíž musí být větší než 0mm.

Poznámka: U formulářů s pevnou velikostí, kde je rozměr dán doporučuji používat zadání šířky buněk v mm. Není dobré kombinovat mm a %. Často pak dochází k chybě "Šířka sestavy je větší než šířka stránky". Pro běžné sestavy doporučuji naopak všechny rozměry zadávat v % (tak se této chybě zcela vyhnete).

Pro **GRF** formát se navíc může určit počet stran dané sestavy (atribut `page-count`).

3.19 Nastavení výchozího zobrazení

Zobrazení, v němž se sestava zobrazí po otevření, lze ovlivnit pomocí `info` sekce `preferred_output` takto:

```
<info preferred_output="txt"/>
```

Taková sestava pak bude po otevření zobrazena pomocí exportu do textu (ASCII). Do nativního zobrazení může být samozřejmě přepnuta obvyklým způsobem. Lze použít i spolu s jinými exporty (`doc`, `xls`, `html`, ...)

Poznámka: `info` sekce `allowed_output` se již nevyužívá.

3.20 Velikosti

Od verze specifikace **1.4** (prohlížeč **4.2.1.0**) lze zadávat všechny rozměry skupiny **GRR** v několika délkových jednotkách. Rozměr se zadává vždy jako řetězec obsahující číslo a příponu metriky, např. `12mm`. Pro některé vlastnosti nejsou přípustné všechny metriky (např. pro velikost písma `font-size` není přípustná metrika %). Lze zadávat i desetinná čísla s desetinnou tečkou, např. `1.15mm`.

Přípustné metriky jsou shrnuty v následující tabulce:

Metrika	Význam
%	Poměrná část vzhledem k velikosti stránky (většinou k šířce stránky)
mm	Milimetr
pt	1/72 palce; často používané pro velikosti písma, např. <code>12pt</code> odpovídá písmu 12 z Wordu apod.
tw	1/20 pt. <code>1tw</code> je nejmenší jednotka a nelze zadat rozměry s vyšší přesností.

3.21 Tabulky

Vnořené tabulky lze používat od specifikace 1.4. Pomocí nich lze rozvrhnout buňky i jinak než pouze do řádků. Názorná ukázka je nejlepší:

2	DEMOK000E0JN		1	Žádost o konzultaci ve věci jedna dva tři abcd efgh a je to děsivě dlouhý a dlouhý až úplně nejdelší	Králik M Supervi
	03.11.2005 11:11:18	Verlag Für Messepublikationen Thomas Neureuter GmbH, Süskindstrade /4, 81929 München, Německo	0		

Ukázka tabulky

Tabulka je obsah buňky (stejně jako `<text>` nebo `<value-of>`). Obsahuje však více řádků, které obsahují další buňky. Zapisuje se takto:

```
<table>
  <line>
    <style ...>
      <value-of name="..." />
```

```

        </style>
    </line>
    <line>
        <style ...>
            <value-of name="..." />
        </style>
    </line>
</table>

```

Šířky, které se dávají v procentech do buněk vnořených v tabulce jsou vztaženy k šířce tabulky, nikoliv k šířce stránky! Tabulka samotná je stále obsahem, tudíž má svůj `spacing` a `padding`. Pro stejné zarovnání jako běžné buňky je tudíž potřeba tyto rozměry nastavit na nulu a poté opět a původní hodnotu (výchozí hodnota je 5 pro `padding`, 0 pro `spacing`).

```

<style ... spacing="0" padding="0">
    <table>
        <line>
            <style ... spacing="0" padding="5">
                <value-of name="..." />
            </style>
        </line>
    </table>
</style>

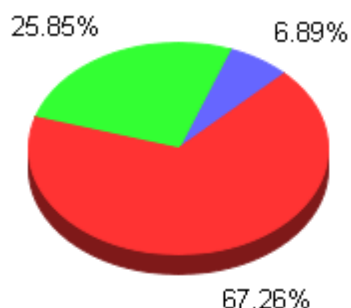
```

3.22 Grafy

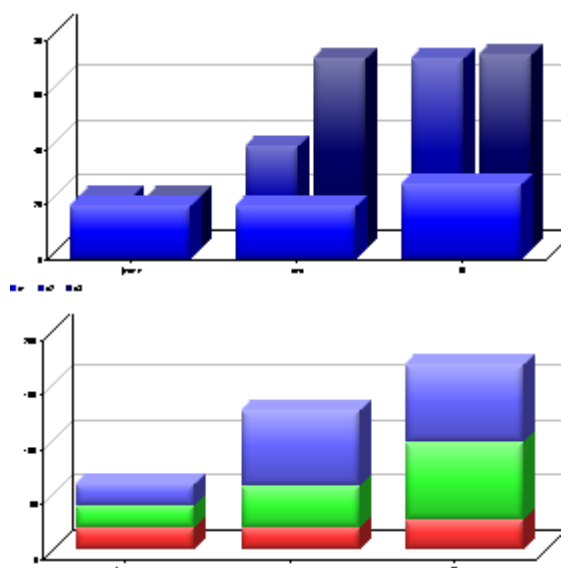
Pro vytvoření grafu je potřeba dvou věcí: datové množiny a typu zobrazení grafu (formátu).

Jsou k dispozici tyto typy grafů:

Výsečový
(koláčový)
`type="pie"`

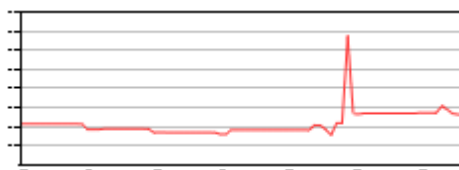


Sloupcový
`type="bar"`



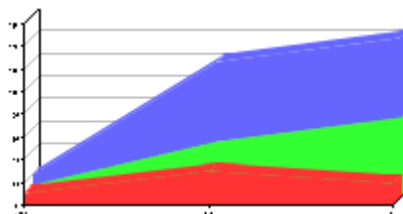
Spojnicový (čárový)

type="line"



Plošný

type="area"



Grafy mimo koláčového lze kombinovat do jednoho konečného grafu. I pro stejný typ, lze do jednoho grafu zakreslit několik hodnot. Ty lze kreslit buď vedle sebe, v prostoru za sebe (v 3d zobrazení) nebo skládat na sebe (pak tvoří jeden sloupec).

Výběr dat pro graf

V případě, že chceme zobrazit několik již vybraných hodnot v rámci jedné oblasti, je situace velmi jednoduchá:

```
<chart ...>
  <data>
    <text value="jeden"/>
    <value-of name="..." />
  </data>
  <data>
    <text value="dva"/>
    <value-of name="..." />
  </data>
</chart>
```

Přímo pod značkou `<chart>` můžeme umístit libovolné množství značek `<data>`, které mohou obsahovat textové obsahy `<text>` a `<value-of>`. První obsah je použit pro osu X, druhý pro 1. sérii, třetí pro 2.sérii atd. Zobrazení více sérií je třeba ovšem uvést ve formátování grafu.

Pro výběr dat z podřízených oblastí píšeme značku `<region>` tak, jak bychom to udělali u standardního výpisu. Je možné použít veškerých možností skupinování, třídění, skriptování, atd. Pro přenos dat do grafu použijeme opět značky `<data>`.

```
<chart ...>
  <region name="JOURNALKARTY_MESIC">
    <body>
      <data>
        <value-of name="..." />
        <value-of name="..." />
      </data>
    </body>
  </region>
</chart>
```

Je možné použít i vícenásobné vnoření oblastí, stejně jako kombinovat oba přístupy.

Vybraná data je také možno měnit, doplnit či mazat pomocí skriptu. Obsah `self.value` je typu seznam (`list`), který je možno měnit. Pro přidání hodnoty je dostupná metoda `add.`:

```
<chart ... onData="self.value.add('A',111)">
```

Formátování grafu

Formátování grafu je možno měnit na několika úrovních:

1. globálně pro celý graf
2. pro danou vrstvu (`layer`)
3. pro skupinu dat (`dataset`)

Jediný povinný údaj je typ grafu, který se uvádí na globální úrovni. Nejsou-li uvedeny žádné vrstvy, vytvoří se právě jedna vrstva s výchozím nastavením. Pro zobrazení více sérií dat, je nutno nastavení vrstev a skupin dat nastavit individuálně.

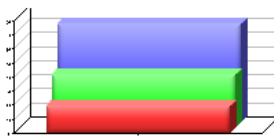
Typ grafu a některé další nastavení na globální úrovni se nastavují jako atributy značky `<chart>`. Např.:

```
<chart type="bar" legend="bottom" color-palette="#00f,#00a,#006">
```

Další atributy jsou společné pro více obsahů (většinou shodné s obrázkem `<image>`).

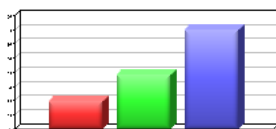
```
image-width="auto" image-height="auto" height="100mm"
```

Pro přidání vrstev se použije značky `<layer>`. Data další vrstvy jsou v grafu znázorněna posunutím v hloubce (pokud je povoleno 3D zobrazení). Např. zde jsou 3 vrstvy:



```
<chart type="bar" image-width="auto" image-height="auto">
  <layer>
    <dataset series="1" name="d1"/>
  </layer>
  <layer>
    <dataset series="3" name="d3"/>
  </layer>
  <layer>
    <dataset series="2" name="d2"/>
  </layer>
  <data>
    <text value="dva"/>
    <text value="20"/>
    <text value="38"/>
    <text value="70"/>
  </data>
</chart>
```

Do vrstev se pak přidávají jednotlivé `datasety`, které odpovídají sériím. Podle typu vrstvy se pak tyto data zobrazí vedle sebe, nad sebou apod.

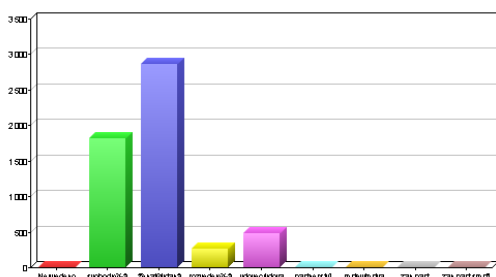


```
<chart type="bar" image-width="auto" image-height="auto">
  <layer type="side">
    <dataset series="1" name="d1"/>
    <dataset series="2" name="d2"/>
    <dataset series="3" name="d3"/>
  </layer>
  <data>
    <text value="dva"/>
    <text value="20"/>
    <text value="38"/>
    <text value="70"/>
  </data>
</chart>
```



```
<chart type="bar" image-width="auto" image-height="auto">
  <layer type="stack">
    <dataset series="1" name="d1"/>
    <dataset series="3" name="d3"/>
  </layer>
  <data>
    <text value="dva"/>
    <text value="20"/>
    <text value="38"/>
    <text value="70"/>
  </data>
</chart>
```

Speciální typ sloupcového grafu, který smí mít pouze jednu množinu dat (`dataset`) a použije barevnou paletu místo k odlišení množin dat, k odlišení jednotlivých hodnot.



```
<chart type="bar">
  <layer type="side-colors" >
```

Pro sloupcové grafy lze ještě volit tvar sloupce:

<code><layer type="side"></code>	
<code><layer type="side" shape="circle"></code>	
<code><layer type="side" shape="bar"></code>	

Atributy grafu

Název atributu	Úroveň	Možné hodnoty / Popis	Výchozí hodnota
<code>image-width</code>	Chart	auto	Doporučuji vždy <code>auto</code>
<code>image-height</code>	Chart	auto	Doporučuji vždy <code>auto</code>
<code>onEnter</code>	Chart		-
<code>onData</code>	Chart		-
<code>only-if</code>	Data		-
<code>type</code>	Chart	pie, bar, line, area	Povinný atribut
<code>group-small</code>	Chart	Text, např. "Ostatní", který bude zobrazen pro součet všech hodnot, které jsou menší než limit.	-
<code>group-limit</code>	Chart	Limit pro skupinování. Lze zadat v % nebo absolutním číslem	5%
<code>type</code>	Layer	stack, side, line, area (nelze pro koláčový graf) <div> <code>stack</code> sloupcový graf skládaný na sebe </div> <div> <code>side</code> sloupcový graf s hodnotami vedle sebe </div> <div> <code>side-colors</code> Jako <code>side</code> s barvami, série </div> <div> <code>line</code> čárový graf </div> <div> <code>area</code> plošný graf </div>	Dle typu grafu
<code>shape</code>	Layer	round, bar, circle (pouze sloupcový)	round
<code>color-palette</code>	Chart	Čárkou oddělený seznam barev	
<code>legend</code>		none, left, top, right, bottom	none
<code>label-layout</code>	Layer	side, circle	side
<code>label</code>	Layer	label, value, percent, label-value, label-percent, none	label-percent

<code>zero-affinity</code>	Chart	Číslo určující zda se má zobrazovat Y osa od nuly. ¹ =ano, 0=ne.	0,8
<code>label-major-step</code>	Chart	Zobrazí každou n-tou položky na X ose	1
<code>label-minor-step</code>	Chart		1
<code>draw3d</code>	layer, chart	true,false	true
<code>color</code>	set, layer, chart	Barva množiny dat	-
<code>name</code>	set, layer, chart	Jméno množiny dat (pro legendu)	-

3.23 Výkazy a formuláře (GRF)

Pro tvorbu výkazů a formulářů, případně jiných sestav, kde nedochází k řádkovému zobrazování předem neznámého množství dat, lze použít formát typu **GRF**.

GRF je na rozdíl od **GRR** absolutně pozicované, tj. nevyskytují se v něm žádné řádky, vnořené tabulky a jiné prvky sloužící pro umístění. Počet stran a rozmístění polí na stránce (stránách) je předem dáno.

Struktura formátovacího souboru skupiny **GRF** je velmi obdobná s **GRR**, kde jsou vynechány všechny `<line>`. Naopak všechny položky musí mít povinný atribut `rect`. Atributy `width` a `height` jsou ignorovány.

Styly položek jsou shodné s **GRR**, pouze nelze použít tyto (méně časté) vlastnosti:

```
spacing
left-spacing
right-spacing
top-spacing
bottom-spacing
inside-border
```

Příklady:

```
<?xml version='1.0' encoding='utf-8' ?>
<format xmlns='http://www.gordic.cz/TR/alf/1.4' type='grf'>

<info ixs_xme='0000ALX0ACHX' />
<info xmeta_ver='1' />
<info xmeta_subver_min='1' />
<info nazev='Bowa zelená obálka' />

<paper-setting paper-width='210mm' paper-height='297mm' page-count='1' />

<region name='HLA'>
  <body>
    <region name='ESU'>
      <body>
        <region name='OBALKA'>
          <body>
            <style top-padding='0' bottom-padding='0' font-face='arial'>
              <value-of name='HLA.ISU_Nazev' rect='2477tw,226tw,10772tw,453tw' />
            </style>
          </body>
        </region>
      </body>
    </region>
  </body>
</region>
```

```

        </region>
    </body>
</region>
</body>
</region>

</format>

```

Z příkladu je zřejmé klasické vnoření oblastí (`region`), klasické záhlaví (`info`, `paper-setting`), formátovací skupina v záhlaví (`grf`) a umístění jedné položky na stránku.

```

<style font-size="8.95pt" font-color="black">
    <text rect="63.69mm,75.11mm,110.24mm,81.1mm">PID Dokumentu</text>
</style>
<style text-orientation="270" font-size="21.65pt">
    <value-of name="ESU.IDDORUCENKY" rect="8090tw,3011tw,8860tw,7546tw" onData="if self.value!='':sel
</style>

```

Zde vidíme textovou položku a příklad datové položky upravené skriptem.

3.23.1 Stránkování v GRF

GRF formát má předem daný počet stran. Pro opakování sestavy pro více dat je určena tzv. hlavní oblast. Je to první oblast ve formátu, která má v sobě nějaké políčka (v příkladu výše je to oblast `OBALKA`). Pro každý výskyt této oblasti v datech je pak do výsledku přidán odpovídající počet stran (v příkladu výše pak pro každou `OBALKA` jedna strana `A4`).

Lze využít i případné vnořené oblasti do této hlavní oblasti, nicméně takové oblasti musí mít uveden vlastní `rect`, v rámci jedné strany a pouze tuto oblast pak lze použít pro vkládání políček z takové oblasti. Pro účely odlišení jednotlivých výskytů těchto podoblastí lze použít atributu `row` (výchozí hodnota 1 se nemusí uvádět). Pole s `row='1'` jsou pak použity pro první výskyt položky v datech, `row='2'` pro druhý atd.

```

<region name="SUB" rect='8940tw,4825tw,10732tw,7052tw'>
    <body>
        <value-of name="a" rect='8940tw,5025tw,10732tw,5152tw' row='1' />
        <value-of name="a" rect='8940tw,5525tw,10732tw,5652tw' row='2' />
        <value-of name="a" rect='8940tw,6025tw,10732tw,6152tw' row='3' />
    </body>
</region>

```

3.24 Soubor s popisem struktury dat sestavy (.xme)

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu
<code>structure*</code>	kořenová značka	<code>xmlns*</code>	<code>http://www.gordic.cz/TR/xme/1.0/</code>		jmenný prostor XML
<code>info*</code>	popis struktury dat	<code>ixs_xme*</code>			identifikátor struktury dat
		<code>xmeta_ver*</code>			verze struktury dat
		<code>xmeta_subver*</code>			subverze struktury dat
		<code>nazev**</code>			název formátu
		<code>poznámka</code>			poznámka

region	sekce	name*			název datové oblasti
		title*			titulek datové oblasti
		description			popis datové oblasti
		preview-count	číslo >= 1	2 nebo 5	počet opakování této oblasti v náhledu během editace
item	datová položka	name			název datové položky
		title			titulek datové položky
		description			popis datové položky
		datatype	string, number, datetime	string	datový typ (pro skripty)
		default-format			formát, který se použije ve formátu, pokud není zadán vlastní. Má smysl pouze pro položky s datatype<>string. Viz value-of/ format v alf.
		preview-value		[name]	hodnota, která se použije v náhledu během editace může obsahovat %d k číslu řádku

legenda k tabulce :

- * povinná položka nebo atribut
- ** povinná položka nebo atribut pro nasazení v systému GINIS

3.25 Soubor s předpisem pro formátování sestavy (.alf)

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu
format*	kořenová značka	xmlns*	http://www.gordic.cz/TR/alf/1.4/		jmenný prostor XML
		type*	grr, msw, mse		skupina formátování
info**	popis formátovacího předpisu	ixs_frm*			identifikátor formátu
		ixs_xme*			identifikátor struktury dat
		xmeta_ver*			verze struktury dat
		xmeta_subver_min*			min. subver. struktury dat
		nazev**			název formátu
		ixs_alv**			identifikátor sestavy

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu
		poznámka			poznámka
		rokmes_od			platnost od
		rokmes_do			platnost do
region	datová oblast	name*			název datové oblasti
		only-if	Výraz		Viz kapitola Podmínění zobrazení
		filter-out	Výraz		Viz kapitola Podmínění zobrazení
head	blok řádků hlavičky oblasti (pouze v <region>)				
body	blok řádků těla oblasti (pouze v <region>)				
foot	blok řádků patičky oblasti (pouze v <region>)				
group	skupina	by*			skupinovací kritérium
		name			jméno oblasti
variable	definice počítané proměnné	name*			jméno
		value*			výraz pro hodnotu
		datatype			datový typ (není-li uveden zjistí se podle výsledku výpočtu)
paper-setting	nastavení stránky	paper-width	*size[mm]	0	šířka papíru***
		paper-height	*size[mm]	0	výška papíru***
		left-margin	*size[mm]	0	levý okraj***
		right-margin	*size[mm]	0	pravý okraj***
		top-margin	*size[mm]	0	horní okraj***
		bottom-margin	*size[mm]	0	spodní okraj***
		page-count	(GRF)	1	počet stran
editor-setting	nastavení editoru – vyhrazeno pro účely návrháře sestav				

legenda k tabulce :

- * povinná položka nebo atribut
- ** povinná položka nebo atribut pro nasazení v systému GINIS
- *** hodnota 0 znamená výchozí nastavení prohlížeče, bez vlivu ze sestavy
- *colors aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple,
- *border-style solid, dotted, dashed, double
- *size kladné číslo bez mezery následované příponou udávající použitou metriku. Viz také [zde](#).
- *npsize Jako *size ale není povolena procentuální metrika (%)

**exact-size* celé kladné číslo v rozmezí od 0 do 25, které udává přesnou velikost v twipech dle pravidla 0 = 0 twipů, 1=1twip, a poté vždy N-1 násobek 10 twipů (tzn. 2 = 10twip , 3 = 20twip , atd.)

**one-character* právě jeden tisknutelný znak

**charset* default, ansi, symbol, oem, easteuropa, mac, baltic, chinesebig5, gb2312, hangul, greek, russian, shiftjis, turkish, vietnamese

3.25.1 Skupina formátování GRR a GRF

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu	Popis chování ve skriptu
line	ádek sestavy (GRR)	on-each-page	true, false, middle	false	ádek na každé stránce	-
		break-page-before	none, always, avoid (true = always, false = none)	none, v patičce avoid	konec strany před ádkem [avoid ver>=1.3]	-
		break-page-after	none, always, avoid (true = always, false = none)	none, v hlavičce avoid	konec strany za ádkem [avoid ver>=1.3]	-
		endpage	true, false	true	ádek stránkové patičky se vytiskne na úplný konec strany (true) nebo hned za text (false). [ver>=1.3]	-
		only-if	Výraz		Viz kapitola Podmíněné zobrazení	-
table	Vnořená tabulka (GRR)				Viz kapitola Tabulky	
chart	Graf				Viz kapitola Grafy	
value-of	datová položka	name*			název datové položky	R (str)
		-			obsahuje hodnotu pole z datového souboru.	value RW (dle typu ve struktuře – str,Decimal, Datetime)
		-			odkaz na styl bučky	style R (obj)
		format			formátování sloupce s typem number nebo datetime provádí se až po onData	-
		onPrint			skript při tisku hodnoty. (pouze pro kompatibilitu)	-
		onEnter			skript při vstupu do oblasti (Regionu) (vždy vyhodnoceno pouze jednou po načtení hodnoty dat) vhodné pro změnu stylu (self.style)	-
		onData			skript při načtení hodnoty datového pole (vždy vyhodnoceno pouze jednou) vhodné pro změnu dat (self.value)	-
		onLoad			skript při načtení formátu	-

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu	Popis chování ve skriptu
		width	*size	poměrná [%]	není-li uvedena je převzata ze stylu	-
		height	*size	poměrná	není-li uvedena je převzata ze stylu	-
		rect	*rect (GRF)		umístění na stránce pro GRF	
		page	1-(page-count) (GRF)		číslo strany pro GRF	
		row	(GRF)		číslo výskytu položky pro GRF	
		style			odkaz na styl (viz sdílení styl)	-
text	Textová hodnota	value*			textová hodnota	RW (str)
		inline hodnota			textová hodnota	-
		-			odkaz na styl buď	style R (obj)
		onLoad			skript při načtení formátu	-
		width	*size	poměrná [%]	není-li uvedena je převzata ze stylu	-
		height	*size	poměrná	není-li uvedena je převzata ze stylu	-
		rect	*rect (GRF)		umístění na stránce pro GRF	
		page	1-(page-count) (GRF)		číslo strany pro GRF	
		row	(GRF)		číslo výskytu položky pro GRF	
		style			odkaz na styl (viz sdílení styl)	-
image	Obrázek	file*			soubor s obrázkem BMP	RW(str)
		global	true, false	false	průznak zda se jedná o globální obrázek (cesta nastavena parametrem image_dir)	RW(num) 0/1
		image-width	*size, auto	none	velikosti obrázku. Může obsahovat hodnotu "auto", pak je velikost určena z velikosti buď. Není-li je velikost závislá na skutečné velikosti obrázku (nedoporučuje se)	image_width RW (str)
		image-height	*size, auto	none		image_height RW (str)
		-			odkaz na styl buď	style R (obj)
		-			Data obrázku kódovaná v Base64	encoded RW (str)
		onLoad			skript při načtení formátu	-
		width	*size	poměrná [%]	není-li uvedena je převzata ze stylu	-
		height	*size	poměrná	není-li uvedena je převzata ze stylu	-
		rect	*rect (GRF)		umístění na stránce pro GRF	
		page	1-(page-count) (GRF)		číslo strany pro GRF	

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu	Popis chování ve skriptu
image-of	Datový obrázek	row	(GRF)		íslo výskytu položky pro GRF	
		style			odkaz na styl (viz sdílení styl)	-
		name			název datové položky	R (str)
		global	true, false	false	příznak zda se jedná o globální obrázek (cesta nastavena parametrem image_dir)	RW(num) 0/1
		image-width	*size, auto	none	velikosti obrázku. Může obsahovat hodnotu "auto",	image_width RW (str)
		image-height	*size, auto	none	pak je velikost určena z velikosti buňky. Nemá-li je velikost závislá na skutečné velikosti obrázku (nedoporučuje se)	image_height RW (str)
		-			odkaz na styl buňky	style R (obj)
		-			Data obrázku kódovaná v Base64	encoded RW (str)
		onPrint			skript při tisku hodnoty. (pouze pro kompatibilitu)	-
		onEnter			skript při vstupu do oblasti (Regionu) (vždy vyhodnoceno pouze jednou po načtení hodnoty dat) vhodné pro změnu stylu (self.style)	-
		onData			skript při načtení hodnoty datového pole (vždy vyhodnoceno pouze jednou) vhodné pro změnu dat (self.value)	-
		onLoad			skript při načtení formátu	-
		width	*size	poměrná [%]	není-li uvedena je převzata ze stylu	-
		height	*size	poměrná	není-li uvedena je převzata ze stylu	-
		rect	*rect (GRF)		umístění na stránce pro GRF	
Style	styl výstupu	page	1-(page-count) (GRF)		íslo strany pro GRF	
		row	(GRF)		íslo výskytu položky pro GRF	
		style			odkaz na styl (viz sdílení styl)	-
		id			jméno stylu (viz sdílení styl)	-
		width	*size	poměrná [%]	šířka buňky. Je-li součet šířek menší než 100% šířky stránky, je poslední buňka na řádku zvětšena tak, aby součet byl 100%. Součet vyšší než 100% znamená chybu!	W (str)

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu	Popis chování ve skriptu
		height	<i>*size</i>	poměrná	výška bučky	W (str)
		font-face	times, arial, courier, custom	times	název písma	font_face RW (str)
		font-name	libovolný et z		název písma, pokud je zvoleno uživatelské písmo <code>font-face=custom</code>	font_name RW (str)
		font-charset	<i>*charset</i>	easturope	kódová stránka zvoleného písma	font_charset RW (num) W(str)
		font-size	1-8, <i>*npsize</i>	2	velikost písma	font_size RW (str)
		font-bold	true, false	false	tučné písmo	font_bold RW (num) 0/1
		font-italic	true, false	false	kurzíva	font_italic RW (num) 0/1
		font-underline	true, false	false	podtržené písmo	font_underline RW (num) 0/1
		font-strikeout	true, false	false	pěškrtnuté písmo	font_strikeout RW (num) 0/1
		font-color	<i>*colors</i>	black	barva písma	font_color RW (str)
		border-width	<i>*exact-size, *npsize</i>	0	tloušťka všech rámečků	border_width W (num)
		left-border-width	<i>*exact-size, *npsize</i>	0	tloušťka levého rámečku	left_border_width RW (num)
		right-border-width	<i>*exact-size, *npsize</i>	0	tloušťka pravého rámečku	right_border_width RW (num)
		top-border-width	<i>*exact-size, *npsize</i>	0	tloušťka horního rámečku	top_border_width RW (num)
		bottom-border-width	<i>*exact-size, *npsize</i>	0	tloušťka dolního rámečku	bottom_border_width RW (num)
		border-color	<i>*colors</i>	black	barva všech rámečků	border_color W (str)
		left-border-color	<i>*colors</i>	black	barva levého rámečku	left_border_color RW (str)
		right-border-color	<i>*colors</i>	black	barva pravého rámečku	right_border_color RW (str)
		top-border-color	<i>*colors</i>	black	barva horního rámečku	top_border_color RW (str)
		bottom-border-color	<i>*colors</i>	black	barva dolního rámečku	bottom_border_color RW (str)
		border-style	<i>*border-style</i>	solid	styl všech rámečků	border_style W (str)
		left-border-style	<i>*border-style</i>	solid	styl levého rámečku	left_border_style RW (str)
		right-border-style	<i>*border-style</i>	solid	styl pravého rámečku	right_border_style RW (str)
		top-border-style	<i>*border-style</i>	solid	styl horního rámečku	top_border_style RW (str)
		bottom-border-style	<i>*border-style</i>	solid	styl dolního rámečku	bottom_border_style RW (str)
		horizontal-align, align	left, right, center, justify	left	horizontální zarovnání textu	horizontal_align, align RW (str)
		vertical-align	top, bottom, center	top	vertikální zarovnání textu	vertical_align RW (str)
		ellipsis-style	three-dots, cut, fill	three-dots	styl zakončení písmene přetečení textu ▪ <i>three-dots</i> přidá na	ellipsis_style RW (str)

Značka (tag)	Popis značky	Atributy	Povolené hodnoty atributu	Defaultní hodnota atributu	Popis atributu	Popis chování ve skriptu
					konec znak ... <ul style="list-style-type: none"> ▪ <i>cut</i> uřízne text bez dalšího zpracování ▪ <i>fill</i> místo textu vypíše n kolik znak <i>ellipsis-char</i> vhodné nastavit i <i>multiline</i> nebo <i>height</i> viz Rozdělení textu do více řádků 	
		<i>ellipsis-char</i>	<i>*one-character</i>	*	znak pro vyplňování (viz <i>ellipsis-style</i>)	<i>ellipsis_char</i> RW (str)
		<i>multiline</i>	true, false	true	více řádkový text viz Rozdělení textu do více řádků	RW (num)
		<i>spacing</i>	<i>*exact-size</i> , <i>*npsize</i> (GRR)	0	odsazení rámečku ve všech směrech	W (num)
		<i>left-spacing</i>	<i>*exact-size</i> , <i>*npsize</i>	0	odsazení rámečku vlevo	<i>left_spacing</i> RW (num)
		<i>right-spacing</i>	<i>*exact-size</i> , <i>*npsize</i>	0	odsazení rámečku vpravo	<i>right_spacing</i> RW (num)
		<i>top-spacing</i>	<i>*exact-size</i> , <i>*npsize</i>	0	odsazení rámečku nahoru	<i>top_spacing</i> RW (num)
		<i>bottom-spacing</i>	<i>*exact-size</i> , <i>*npsize</i>	0	odsazení rámečku dole	<i>bottom_spacing</i> RW (num)
		<i>padding</i>	<i>*exact-size</i> , <i>*npsize</i>	5	odsazení textu ve všech směrech	W (num)
		<i>left-padding</i>	<i>*exact-size</i> , <i>*npsize</i>	5	odsazení textu vlevo	<i>left_padding</i> RW (num)
		<i>right-padding</i>	<i>*exact-size</i> , <i>*npsize</i>	5	odsazení textu vpravo	<i>right_padding</i> RW (num)
		<i>top-padding</i>	<i>*exact-size</i> , <i>*npsize</i>	5	odsazení textu nahoru	<i>top_padding</i> RW (num)
		<i>bottom-padding</i>	<i>*exact-size</i> , <i>*npsize</i>	5	odsazení textu dole	<i>bottom_padding</i> RW (num)
		<i>background-color</i>	<i>*colors</i>	transparent	barva pozadí bučky	<i>background_color</i> RW (str)
		<i>text-orientation</i>	0-360	0	otočení textu	<i>font_orientation</i> RW (num)
		<i>inside-border</i>	true,false (GRR)	false	způsob umístění rámečku v bučce false=rámeček se kreslí na střed rozhraní mezi spacingem a paddingem a nezapočítává se do velikosti bučky true=do velikosti se započte kreslí se dovnitř bučky (sousedící bučky mají každá svůj rámeček)	<i>inside_border</i> RW (num)

legenda k tabulce :

* povinná položka nebo atribut

** povinná položka nebo atribut pro nasazení v systému GINIS

*colors	aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow
*border-style	solid, dotted, dashed, double
*size	kladné číslo bez mezery následované píponou udávající použitou metriku. Viz také zde .
*npsize	Jako *size ale není povolena procentuální metrika (%)
*exact-size	celé kladné číslo v rozmezí od 0 do 25, které udává přesnou velikost v twípech dle pravidla 0 = 0 twip , 1=1twip, a poté vždy N-1 násobek 10 twip (tzn. 2 = 10twip , 3 = 20twip , atd.)
*one-character	práv jeden tisknutelný znak
*charset	default, ansi, symbol, oem, easteurope, mac, baltic, chinesebig5, gb2312, hangul, greek, russian, shiftjis, turkish, vietnamese

3.25.2 Skupina formátování MSE a MSW

Značka (tag)	Popis značky	Atributy	Popis atributu
template	šablona	filename*	název souboru šablony
value-of	datová položka	name*	název položky
		cell	buňka do které se položka přesune (jen pro MSE)
copy-and-fill	kopírování hodnot	from*	z
		to*	do

legenda k tabulce :

* povinná položka nebo atribut

3.25.3 Skupina formátování RTF

Značka (tag)	Popis značky	Atributy	Popis atributu
template	šablona	filename*	název souboru šablony
value-of	datová položka	name*	název položky
		type	RTF field type
		rtf	RTF field spec
		instance	RTF field instance
rtf	kopírování hodnot	from*	z
		to*	do

legenda k tabulce :

* povinná položka nebo atribut

3.25.4 Seznam speciálních hodnot dostupných ve formátovacím předpisu (.alf)

Název hodnoty	Datový Typ	Popis hodnoty
<code>ROOT.PAGE</code>	<code>string</code>	číslo aktuální stránky
<code>ROOT.PAGES</code>	<code>string</code>	počet stránek sestavy
<code>ROOT.DATE</code>	<code>string</code>	datum tisku
<code>ROOT.TIME</code>	<code>string</code>	čas tisku
<code>ROOT.VERSION</code>	<code>number</code>	verze reporteru/prohlížeče (např. 4001005018)
<code>ROOT.NOW</code>	<code>datetime</code>	Datum a čas tisku. Nastavení této hodnoty nastaví i <code>ROOT.DATE</code> a <code>ROOT.TIME</code>

Tyto datové položky lze měnit v datovém souboru. Za povinné informace `ixs_xme|verze|subverze|` se uvede jméno položky = hodnota. Např.

```
ixs_xme|pokus|1|1|NOW=12121212030303|
```

3.26 Pokročilé skriptování

Skripty využívají jazyka **Python** ve verzi **2.4.2**. Množina základních knihoven jsou proti standardní instalaci omezeny, většinu funkcí však lze využít. Dokumentace k Pythonu lze najít online na adrese <http://www.python.org/doc/2.4.2/>. Ve skriptech lze přímo využít všech vlastností a zabudovaných (build-in) funkcí jazyka a [knihovny re](#) pro práci s regulárními výrazy. Navíc jsou přidány dvě rozšiřující knihovny `Decimal` a `Datetime` obsahující stejnojmenné datové typy (viz výše). Některé knihovny, které jsou ve standardní distribuci jsou ze skriptové knihovny vyjmuty (např. knihovny pro vícevláknové zpracování, pro práci se soubory apod.).

Upozornění: od verze **2.4.0** má Python také knihovnu `Decimal`. Ačkoliv některé vlastnosti obou typů jsou stejné, nejedná se o shodné typy. V reporteru je pod jménem `Decima` dostupný jiný typ, s některými jinými funkcemi (např. `spellOut`).

Lze použít i jiných knihoven, je však třeba je pře použitím importovat (příkazem `import`). Dokonce je možné si psát vlastní uživatelské knihovny. Ty pak lze použít např. takto:

```
<value-of name="" onData="import abc;self.value=abc.helloworld()"/>
```

A knihovna samotná vypadá takto:

```
def helloworld(): return "ahoj sv te"
```

Soubor je nutné umístit do souboru `abc.py` a do stejného adresáře jako je skriptová knihovna **g32grs01.dll**. Jednoduché, že?

Některé standardní knihovny, které jsou zařazeny do skriptovací knihovny a mohou být

použity (po importu):

[math](#), [locale](#), [array](#), [time](#), [string](#), [types](#)

Výrazy

Ve výrazech (`only-if`, hodnota počítané proměnné apod.) lze použít totéž co ve výrazu v jazyce Python (např. pravá strana přiřazení). Návratová hodnota výrazu by měla respektovat použití výrazu (např. hodnotu Ano/Ne pro `only-if` podmínky).

Logické výrazy

Shoda	<code>Hodnota1 == Hodnota2</code>
Neshoda	<code>Hodnota1 != Hodnota2</code>
Menší	<code>Hodnota1 < Hodnota2</code> (v XML mimo CDATA se musí zadávat jako <code>Hodnota1 &lt; Hodnota2</code>)
Vetší	<code>Hodnota1 > Hodnota2</code>
Z výčtu	<code>Hodnota in (Vycet1,Vycet2,Vycet3)</code>
V intervalu	<code>Hodnota in range(Pocatek,Konec)</code> , platí pouze pro celá čísla v intervalu!

Pozor na datové typy hodnot. Pokud porovnáváte datovou položku typu řetězec, je nutné porovnávat s řetězcem. Např. `OBLAST.PRIZNAK == '20'`. Porovnání s `20` by nebylo nikdy úspěšné.

Python nemá přímo logický `Ano/Ne` typ. Jako hodnota `Ne` se používá:

- číslo `0`
- prázdný řetězec
- prázdný seznam, ntice, slovník apod.

Podmínky lze skládat pomocí logických operátorů `AND` a `OR`, případně negovat pomocí operátoru `NOT`. Priorita operátorů je přesně dána, ale nedoporučuji se na ni spoléhat a spíše uzavřít do závorek.

Pozor: operátory `AND`, `OR` a `NOT` MUSÍ být malými písmeny!

Příklady:

```
OBJ.CEN>=100 and OBJ.MN>1
(OBJ.CEN>=100) and (OBJ.MN>1)
(OBJ.CEN>=100 and OBJ.MN>1) or (OBJ.MN>100)
```

3.27 Převodní mosty

3.27.1 Do TXT

Most do TXT převádí z formátovací skupiny GRR pole, jejich rozvržení a základní rámování. Šířku stránky lze přizpůsobit parametrem `Ascii_Width`. Parametr `Ascii_Height` ovlivňuje délku stránky pro záhlaví a zápatí. Pokud je nastaven parametr `Ascii_PoorManGraphics` na `false`, tak je pro rámování použito rámečkových znaků v kódování WEC (Gordic), jinak je použita pouze znaků + (plus), - (minus), | (roura)

3.27.2 Do HTML

Most nepřevádí obrázky, nemá žádné stránkování (číslo strany i počet stran je 1), stránkové hlavičky a patičky nejsou zobrazeny. Také spacing buněk je ignorován (padding zachován).

3.27.3 Do MHT

Shodný s [HTML](#), pouze jsou doplněny obrázky.

3.27.4 Do PDF

Využívá knihovny **dynapdf.dll** zakoupenou od 3tí strany. Přenáší veškeré vlastnosti grafických sestav včetně stránkování!

3.27.5 Do XLS (GRR)

Používá k exportu [HTML_most](#), do kterého jsou vloženy speciální značky pro **Excel**. Tento HTML je pak pomocí OLE načten do Excelu (Excel musí být dostupný na počítači kde je most spouštěn). Výsledek je .xls soubor.

- **Excel 97** ignoruje veškeré formátování v HTML, takže barvy, zarovnání, písma jsou ztraceny.
- **Excel 2000** a další nastavení stylu zachovají.

3.27.6 Do DOC (GRR)

Používá [RTF_most](#). Přípona souboru je poté změněna na .doc. Word má omezení na minimální velikost buňky – tj. řádky s hodně malými buňkami (1%) pak mohou být prodlouženy.

- **Word 97** ignoruje spacing buněk. `Padding` musí být ze všech stran stejný. Podpora pouze základních 16 barev.

3.27.7 Do RTF (GRR)

Základní podpora obrázků. Podpora stránkování.

Velikosti písem (1-8) se převedou takto:

- 1 – font 7
- 2 – font 9
- 3 – font 10,5
- 4 – font 13
- 5 – font 16
- 6 – font 22
- 7 – font 33

Obrázky značně zvětšují velikost výsledného RTF. Lze obejít pomocí nastavení zde: <http://support.microsoft.com/kb/224663/en-us> (registrujte se)
 HKEY_CURRENT_USER\Software\Microsoft\Office\xx\Word\Options, (klíč
 ExportPictureWithMetafile, hodnota 0).

3.27.8 Do XML (GRR, MSW, MSE)

Most do XML převádí formáty v libovolné formátovací skupiny (GRR, MSE, MSW) do formátu XML. Pro převod se použijí jména oblastí a datových položek, jak jsou uvedena ve struktuře a značkách `<value-of>`. Tento most ignoruje všechno formátování v `<head>` a `<foot>`! Směrodatné jsou pouze datové pole z `<body>` tj. značky `<value-of>` (nebo přesněji značky které po načtení parserem podporují rozhraní `IDataCacheItem`)

Výsledný XML soubor obsahuje kořenovou značku `<report>` s atributem `created`, kde je datum vytvoření sestavy ve formátu `YYYYMMDD`. Dále jsou jednotlivá data ze vstupního datového souboru ohraničená značkou se jménem shodným s názvem oblasti. Data jsou uspořádána v tom pořadí, v jakém se vyskytují ve formátovacím předpisu (`.alf`)! Pokud některá datová položka není ve formátu uvedena, není uvedena ani ve výsledném XML souboru. Položky, které jsou ve formátu vícekrát v jediném `<body>` se zapíší do XML výstupu taktéž vícekrát. Stejná jména položek jsou změněna pomocí přípony `_n`, kde `n` je číslo výskytu. Obsah datových položek je umístěn jako text mezi značky. Je provedena konverze znaků `<` (menší než), `>` (větší než) a `&` (ampersand) do XML reprezentace těchto znaků. Používá se kódování Windows 1250.

3.27.9 Do XLS (nativní – MSE)

Nahrazuje pole v šabloně patřičnými poli z dat. Při návrhu jsou k dispozici všechny vlastnosti Excelu. Verze Excelu při návrhu a na stroji, kde dochází ke spouštění mostu by měla být shodná (nebo šablonu uložit pro nižší verzi). Je třeba dbát na omezení Excelu, hlavně na omezení délky tabulky (60.000 řádků)

Na jednu věc je třeba dále zvláštní pozor: pokud není položka dat vyplněna (je prázdná), zůstane v buňce ten text, vzorec apod. který je tam umístěn v době návrhu. To lze použít pro různé výpočty, ale je nutné si to uvědomit např. ve sloupci poznámka, která často není vyplněna. Návrhář sestav do buňky umístí název té buňky, tudíž pro nevyplněné se pak všude objeví poznámka (je nutno vymazat nebo vypsát text např. "není k dispozici" apod.)

Zkuste použít exportní most XLS ze skupiny GRR (klasická grafická sestava převedená do Excelu). Lze použít i info sekci `preferred_output`.

3.27.10 Do DOC (nativní – MSW, RTF)

Nahrazuje pole v šabloně patřičnými poli z dat. Pracuje buď v režimu kompatibility nebo v plném režimu. Kompatibilní režim nevyžaduje v `.alf` žádné `<value-of>` ale neumožňuje vkládat pole z jiných oblastí než aktuální (z nadřazených). Návrhář podporuje nový režim od verze **4.2.1.3**.

Verze Wordu při návrhu a na stroji, kde dochází ke spouštění mostu by měla být shodná (nebo šablonu uložit pro nižší verzi).

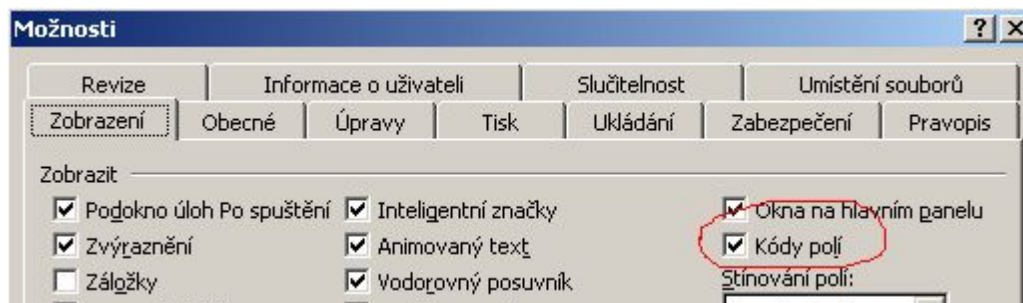
Zkuste použít exportní most DOC ze skupiny GRR (klasická grafická sestava převedená do Wordu). Lze použít i info sekci `preferred_output`.

Problém 1:

namísto hodnot zobrazuje něco takového:

IČO { FORMTEXT }	Zpravodajská jednotka (adresa):
PČZ { FORMTEXT }	{ FORMTEXT }
	{ FORMTEXT }
druh ZZ { FORMTEXT }	{ FORMTEXT }

tak má uživatel ve Wordu nastaveno "zobrazovat kódy polí". Přepnutí do tohoto stavu lze učinit i nechtíc stiskem `Alt-F9`. Naštěstí pomoc je také jednoduchá: opět `Alt-F9` přepne do normálního zobrazení. Lze také vypnout v **Nástroje/Možnosti/Zobrazení**:



3.28 Distribuce v systému GINIS

K tomu, aby byla zabezpečena tato primární funkce aplikace, je nutné pro každou zpracovávanou sestavu zabezpečit přenos několika podpůrných souborů na klientský počítač a zároveň pro ně zaktualizovat příslušné vztahové tabulky v databázi. Veškerá potřebná distribuce souborů a dat je přitom prováděna pomocí standardního reinstalačního mechanismu. Tzn. všechny potřebné soubory pro sestavu jsou zabaleny do klasické distribuční sady a při jejich načítání do databáze proběhne na základě parametrů z nich získaných aktualizace dotčených údajů v databázových tabulkách.

Aplikace **Gordic Reporter** pracuje s následujícími soubory :

1. **Soubor s popisem struktury dat sestavy** - soubor obsahující popis struktury a hierarchie dat, která jsou pro sestavu vybírána z databáze na základě uživatelského požadavku na vytvoření konkrétní sestavy
2. **Soubor s předpisem pro formátování sestavy** - soubor obsahující předpis pro výstupní grafickou vizualizaci dat sestavy
3. **Archivní soubor s podpůrnými soubory pro formátování** - volitelný soubor, který obsahuje podpůrné zdroje pro předpis pro formátování sestavy (tj. např. rastrové obrázky, šablony apod.)
4. **Předpis pro získání dat pro sestavu z databáze** - tradičně je to zkompileovaný tvar předpisu v jazyce **Gordic Basic** známý jako `ALV` soubor. Slouží pro definici algoritmu jehož aplikováním budou pro sestavu vybírána data z databáze, přičemž výsledkem tohoto procesu bude soubor s daty pro zformátování sestavy.

Soubor s popisem struktury dat sestavy

Pro unikátní identifikaci struktury dat slouží autorem přidělený atribut `ixs_xme`. Struktura dat je mimo to verzována pomocí čísla verze a sub verze. Při povyšování subverze, smí být struktura dat pouze rozšiřována, aby se zamezilo konfliktům s jinými sestavami. Při větším zásahu do struktury je nutné povýšit vyšší verzi a tím zamezit spuštění starších sestav. Všechny tři uvedené položky (`ixs_xme`, `xmeta_verze`, `xmeta_subverze`) musí být povinně uváděny jako atributy položky info umístěné bezprostředně v kořenové položce XML popisu struktury dat. Na tomto místě je taktéž nutné uvést název struktury dat a volitelně i poznámku (oba údaje na max. 50 znaků).

Název souboru se strukturou dat je odvozen z `ixs_xme` následujícím způsobem:

```
ixs_xme[1,4]+ixs_xme[8,4]+' .xme'
```

Reakce reinstalátoru v případě výskytu souboru s příponou `xme` v reinstalační sadě se provede aktualizace příslušných tabulek (`ginsxme`, `ginsfil` a přidružených tabulek).

Soubor s předpisem pro formátování sestavy

Jedinečnou identifikaci tohoto souboru zajišťuje jeho autor tím, že mu přidělí identifikátor `ixs_frm`. Dalšími povinnými atributy předpisu pro formát sestavy jsou identifikátor sestavy (`ixs_alv`), identifikace struktury dat v čítně specifikace podporované verze a minimální podporované subverze (`ixs_xme`, `xmeta_verze`, `xmeta_subverze_min`) a název formátu sestavy (max. 50 znaků). Volitelným parametrem je poznámka (také max. 50 znaků). Tyto atributy jsou uváděny v položce info umístěné přímo v kořenové položce obsaženého XML předpisu. Stejná sestava může být zformátována více způsoby na základě více různých předpisů pro formát sestavy.

Název souboru s předpisem pro formátování sestavy je odvozen z `ixs_frm` následujícím způsobem:

```
ixs_frm[1,4]+ixs_frm[8,4]+' .alf'
```

Reakce reinstalátoru v případě výskytu souboru s příponou `alf` v reinstalační sadě je aktualizace příslušných tabulek (`ginsfrm`, `ginvalf`, `ginsfil` a přidružených tabulek).

Archivní soubor s podpůrnými soubory pro formátování

Jedná se o `zip` archiv s dodatečnými soubory pro formátování (bitmapy, šablony).

Název archivního souboru s podpůrnými soubory pro formátování sestavy je odvozen z `ixs_frm` následujícím způsobem:

```
ixs_frm[1,4]+ixs_frm[8,4]+' .zip'
```

Reakce reinstalátoru v případě jeho výskytu v reinstalační sadě je pouze jeho natažení do `ginsfil` a přidružených tabulek.

Předpis pro získání dat pro sestavu z databáze

Tradičně je to zkompileovaný tvar předpisu pro získání dat pro sestavu z databáze v jazyce **Gordic Basic** známý jako `ALV` soubor. Je jednoznačně označen prostřednictvím identifikátoru `ixs_alv`. Ve svém obsahu nese celou řadu atributů, které slouží pro definici sestavy v číselníku sestav `ginsalv` a dále pro její zařazení do stromu sestav. Dále musí

obsahovat několik atributů pro Reporter: identifikátor struktury dat produkované ALV předpisem včetně jejího čísla verze a sub verze (`ixs_xme`, `xmeta_verze`, `xmeta_subverze`), identifikátor přednastaveného formátu sestavy (`ixs_frm`) v rámci všech stromů sestav, a případně též identifikátor přednastaveného formátu sestavy pouze pro daný strom sestav. Posledně jmenovaný atribut slouží k aktualizaci tabulky `ginvalv` (sestavy ve stromu), ostatní budou aktualizovány do tabulky `ginsalv` (Číselník sestav).

Název souboru s předpisem pro získání dat pro sestavu z databáze se odvozuje z `ixs_alv` následujícím způsobem:

```
ixs_alv[1,4]+ixs_alv[8,4]+' .alv'
```