

Uživatelská tvorba sestav

Obsah

1 Prerevizity a základní informace.....	2
2 Přístup k databázi.....	2
2.2 Manažer objektů.....	2
2.3 Object collection (kolekce).....	3
2.3.1 Příklad.....	3
2.3.2 Otevření kolekce.....	3
2.3.3 Průchod kolekcí.....	4
2.3.4 Přístup k datům kolekce.....	4
2.3.5 Uvolnění kolekce z paměti.....	4
2.4 Record set.....	4
2.4.1 Příklad.....	5
2.4.2 Průchod recordsetem.....	5
2.4.3 Přístup ke vlastnostem.....	5
2.4.4 Uvolnění recordsetu z paměti.....	5
3 Vytvoření nové sestavy.....	6
3.2 Struktura sestavy.....	7
3.3 Interní tagy.....	8
3.3.1 REPORT.....	9
3.3.1.1 Kolekce.....	9
3.3.1.2 Recordset.....	10
3.3.1.3 Vytvoření zanořeného reportu.....	11
3.3.1.4 Další zajímavosti.....	11
3.3.2 BAND.....	12
3.3.2.1 Popis jednotlivých tříd bandů.....	12
3.3.2.2 Vytvoření nového BANDu.....	13
3.3.2.3 Úpravy existujícího BANDu.....	14
3.3.3 ITEM.....	14
3.3.4 H.....	15
3.3.5 E.....	15
3.3.6 PARAMETERS.....	15
3.3.6.1 Datový typ.....	16
3.3.6.2 Databázový objekt.....	16
3.4 Funkce.....	16
Možnosti ladění.....	18
3.5 Objekt Report.....	18
4 Navázání sestavy na okolí.....	18
4.2 Zobrazení stejných dat jako v aplikaci.....	18
4.3 Seřazení dle stejného sloupce jako v aplikaci.....	19
5 Přílohy.....	21

V systému myWAC je možné uživatelsky definovat tiskové sestavy s pomocí modulu „Uživatelská tvorba sestav“. Pokud máte tento modul zakoupen, je nutné nastavit uživatelům, kteří mají mít možnost tvorby sestav, přístup k zabezpečovací položce „Všeobecné – Tvorba uživatelských sestav“.

1 Prerekvizity a základní informace

Při tvorbě uživatelských sestav (dále jen sestav) se primárně využívá jazyka HTML pro formátování tiskové šablony, jazyka VBScript pro interakci s databází a výpočty (vykonávání na serveru) a základů tvorby dotazů v jazyce SQL. Pro tvorbu sestav je nutné mít alespon základní znalosti těchto jazyků.

2 Přístup k databázi

Přístup k databázi je realizován pomocí jádra systému myWAC, které poskytuje funkce nezávislé na databázi, nad kterými běží myWAC. Jsou poskytovány dva základní přístupy (funkce) pro přístup k DB.

Prvním přístupem je tvorba tzv. object collection (kolekcí objektů, kolekcí), které jsou jednodušší na zápis z pohledu programátora a korektní vzhledem k systému myWAC (výsledek dotazu je v souladu s nastavením práv v systému apod.), tento přístup by měl být a je primárním prostředkem pro komunikaci s DB.

Druhým přístupem je tvorba tzv. record setů (množin záznamů), který lze chápat jako dotazování na nižší úrovni než object collection. Výhodou tohoto přístupu je vyšší rychlost a téměř neomezené možnosti pro tvorbu databázových dotazů (jeho struktura zápisu je obdobná dotazům SQL). Nevýhodou, kvůli které doporučujeme používat převážně kolekce, je nezohledňování práv ve vracených datech a složitější zápis.

Na následujících řádcích budou popsány detailně tyto systémová volání. Pzn. většina systémových volání myWACu je dostupná pomocí metod objektu `waccontext`, který je jedním ze základních objektů jádra.

2.2 Manažer objektů

Abychom byly schopni efektivně využívat uživatelskou tvorbu sestav je nutné mít přístup k datovému modelu systému myWAC. Datový model je možné zobrazit kliknutím na odkaz „Manažer objektů“ v designeru tiskových sestav.

S pomocí tohoto manažera si můžete zjistit detailní informace o objektech, které jsou v systému definovány a vlastnostech, které budete chtít zobrazovat v sestavách.



V levé části jsou zobrazeny názvy objektů, ve kterých je možné vyhledávat pomocí textového pole v horní části, a v pravé části se po kliknutí na některý objekt zobrazí seznam vlastností a základní popis. V případě, že vlastnost je odkazem do nějakého dalšího objektu, tak se ve třetím sloupci název objektu na který vlastnost odkazuje (kliknutím na název se zobrazí definice odkazovaného objektu).

2.3 Object collection (kolekce)

```
WACContext.CreateObjectCollection([classname],[condition],[order by],
[ignore object relations],[linked objects],[top rows])
```

[classname]:string – název objektu, ze kterého budeme při vytváření kolekce vycházet

[condition]:string – podmínka dotazu

[order by]:string – pole, podle kterých chcete řadit

[ignore object relations]:boolean – pokud ano, tak se bude provádět joinování pouze na objekty explicitně uvedené v části [linked objects], v druhém případě se do kolekce přidají všechny objekty, které mají nastavený atribut „preview“ ve správci objektů.

[linked objects]:string – zápis objektů, které chceme k výchozímu objektu připojit (joinovat) struktura:

```
[#|]objekt(vlastnost;vlastnost;...),[#|]objekt(vlastnost;vlastnost;...),...
```

kde nepovinný prefix # indikuje, že se bude joinovat i přes prázdnou hodnotu. Pokud není uveden, tak záznam (který nemá vlastnost přes kterou se joinuje vyplněnu) nebude ve výsledku dotazu zahrnut. Jedná se o ekvivalent leftjoinu

[top rows]:integer – maximální počet řádků, které se vrátí i pokud dotazu vyhovuje více záznamů

2.3.1 Příklad

```
Set col = WACContext.CreateObjectCollection("Good","deleted = 0 and
CenikovaSkupina.RabatC1 > 20","Cislo",true,"CenikovaSkupina(RabatC1)")
```

Vytvoří kolekci objektů Good (zbožových karet), které nejsou smazané a jejichž Marže 1 je větší než 20% seřazených podle čísla. Karty, které nemají nastavenou ceníkovou skupinu nebudou v kolekci zahrnuty.

2.3.2 Otevření kolekce

Aby bylo možné s kolekcí pracovat je nutné ji nejdříve otevřít metodou `openIt`.

2.3.3 Průchod kolekcí

Pro průchod kolekcí se využívá metoda `Movenext`, která způsobí přechod na nový záznam a vlastnost `EOF`, která obsahuje hodnotu `true` v případě, že jsme se již metodou `Movenext` posunuly za poslední záznam kolekce.

Typicky je realizován následujícím kódem

```
col.openIt
while not col.eof
.
.
.
col.movenext
wend
```

2.3.4 Přístup k datům kolekce

Rozlišují se základní dva přístupy – přes objekt a přes kolekci. Pokud chcete přistupovat k vlastnostem výchozí objektu kolekce je nutné použít vlastnost kolekce `object`, která nám vrátí instanci objektu z kolekce a k jeho vlastnostem již můžeme přímo přistupovat (všem vlastnostem). Pokud chceme přistupovat k vlasnosti objektu, který byl do kolekce přidán joinem (spojením), musíme použít tečkovou notaci a ne vlastnost `object`.

Rozdíl v zápisech je tedy následující (v col máme kolekci nad výchozím objektem Good)

```
col.object("CenaNakup")           'k vlastnosti objektu
col("CenikovaSkupina.RabatC1")    'k vlastnosti objektu přidaného spojením
```

Alternativou k joinovanému objektu je využití prvního zápisu
`col.object("CenikovaSkupina").object("RabatC1")`

které je ovšem výrazně výkonově náročnější.

2.3.5 Uvolnění kolekce z paměti

Každá kolekce, která již nebude nadále používána by měla být z paměti uvolněna pomocí systémového volání

```
WACContext.ReleaseCollection(col)
```

jehož jediný parametr je kolekce, kterou chceme uvolnit.

2.4 Record set

```
set rs = WACContext.CreateRecordset("Fields", "Objects", "Condition",
"GroupBy", "Having", "OrderBy", "IgnoreObjectRelations", "Relations")
```

[fields]:string – seznam vlastností objektů, které chceme pomocí recordsetu získat oddělených čárkou

[objects]:string – seznam objektů oddělených čárkou

[condition]:string – podmínka dotazu, SQL syntaxe

[order by]:string – pole, podle kterých chcete řadit

[ignore object relations]:boolean – pokud true, tak jsou joinovány pouze objekty uvedené v části [relations] – využívá se převážně tohoto nastavení (v opačném případě je nutné převést joinovací podmínky do podmínky dotazu)

[relations]:string – zápis objektů, které chceme k výchozímu objektu připojit (joinovat)

struktura: [#|]objekt.vlastnost, [#|]objekt.vlastnost, ...

kde nepovinný prefix # indikuje, že se bude joinovat i přes prázdnou hodnotu. Pokud není uveden, tak záznam (který nemá vlastnost přes kterou se joinuje vyplněnu) nebude ve výsledku dotazu zahrnut. Jedná se o ekvivalent leftjoinu

[top rows]:integer – maximální počet řádků, které se vrátí i pokud dotazu vyhovuje více záznamů

2.4.1 Příklad

```
set rs =
WACContext.CreateRecordset("Good.Cislo,Cenikova_Skupina.RabatC1",
"Good,Cenikova_Skupina", "Good.deleted = 0 and
```

```
Cenikova_Skupina.RabatC1 > 20", "", "", "Good.Cislo", true,
"Good.CenikovaSkupina")
```

Ekvivalent ke kolekci v předchozí části.

2.4.2 Průchod recordsetem

Podobně jako u kolekci, jen není nutné recordset otevírat (je otevřen automaticky při vytvoření recordsetu).

```
while not rs.eof
.
.
.
rs.movenext
wend
```

2.4.3 Přístup ke vlastnostem

U recordsetu na rozdíl o od kolekci je možné získat hodnoty vlastností pouze jedním způsobem a to

```
rs ("Good.Cislo")
rs ("Cenikova_Skupina.RabatC1")
```

nebo je možné využít pořadového indexu (číslováno od nuly).

```
rs(0) 'vrátí Good.Cislo
```

2.4.4 Uvolnění recordsetu z paměti

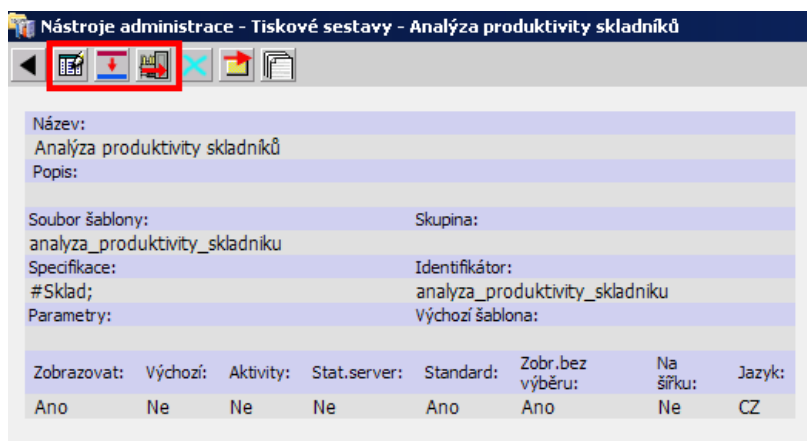
Každý recordset, který již nebude nadále používán by měl být z paměti uvolněna pomocí systémového volání

```
WACContext.ReleaseRecordset(rs)
```

jehož jediný parametr je recordset, který chceme uvolnit.

3 Vytvoření nové sestavy

Novou sestavu je možné vytvořit v „Nástroje administrace – Další nástroje – Tiskové sestavy“ a to buďto jako novou (ikonka „Nová“ v toolbaru) nebo jako kopii stávající (doporučeno).



Kopii stávající vytvoříte kliknutím v detailu existující kliknutím na ikonu



v toolbaru.

Dále je nutné vyplnit formulář:

Nastavení

Název:
Analýza produktivity skladník

Popis:

Soubor šablony: analyza_produkivity_skladniku Skupina:

Specifikace: #Sklad; Identifikátor: analyza_produkivity_skladniku

Parametry: Vchoz šablona:

Zobrazovat:	Vchoz:	Aktivity:	Stat.server:	Standard:	Zobr.bez vbru:	Na šířku:	Jazyk:
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CZ

- Název – název sestavy
- Popis – uřivatelský popis sestavy
- Soubor šablony - soubor šablony je vřdy vygenerovn automaticky (i když jej zmníte)
- Skupina - slouři k označení skupiny šablon pro nř se zvol vchoz šablona
- Specifikace - je ve form #spec; pro odfiltrovn sestav, mře jich bt vce, nelze nastavit vchoz sestavu
- Identifikator - nepouřiv se
- Parametry - připoj se k zadanm parametrm VIEW_Tisk1=1&VIEW_XXX=aaa....
- Vchozi šablona - nepouřiv se
- Zobrazovat - přznak jestli je šablona jako vchozi zobrazena
- Vchoz - aplikuje se pro skupinu
- Aktivity - generovn sestavy udl zznam do logu aktivit
- Stat.server - sestava bude generovna nad statistickm serverem, pokud je definovn
- Standard - sestava je soust standardn sady sestav
- Zobr.bez vyberu - uruje zda sestava um převzít ReportCondition, nebo funguje pouze nad označenmi zznamy
- Na šířku – uruje, ře sestava je na šířku – jen informace pro uřivatele
- Jazyk - uruje v jakm jazyce je sestava, ma vliv na vber vchoz šablony a pořad ve vbru

3.2 Struktura sestavy

Základním jazykem pro tvorbu uživatelských tiskových sestav je jazyk HTML, který je doplněn o řadu interních značek (tagů), sloužících převážně pro přístup k databázovému modelu systému myWAC.

Každá tisková sestava systému myWAC je strukturována do tzv. bandů, což jsou samostatně stojící bloky HTML kódu společně s tagy pro přístup k databázovému modelu, tedy pro přístup k datům objektů, které chceme tisknout. Každý band musí být samostatně stojící, tzn. žádné části bandů se nesmí z pohledu HTML překrývat. Nejčastější realizací bandu v HTML je tabulka, případně DIV. Sestavu je nutné připravovat pro tisk na výšku - **nastavovat šířku bandů na 710px (pro tisk na šířku je doporučena šířka 1040px)!**

Ná následujících řádcích je představena základní struktura sestavy, kterou lze rozšířit o další typy bandů, ale níže uvedené je povinné a každá sestava musí uvedené části obsahovat (krom části STYLE, která je nepovinná a obsahuje standardní CSS styly používané v sestavě). Více ke každé části bude uvedeno dále.

```
<!SCRIPT><%
    --- definice funkcí --
%></SCRIPT>

<STYLE>
    --- styly ---
</STYLE>

<!PARAMETERS>
Name:Název;TYP;výchozí hodnota;operátor;trida objektu;podminka;sort;zobr. pole;
</PARAMETERS>

<BODY MAGRINWIDTH=0 MARGINHEIGHT=0 LEFTMARGIN=0 TOPMARGIN=0>

<!REPORT TITLE="Objednávka Prodej" NAME=OBJEDNAVKA USELINKEDOBJECT=TRUE
LINKEDOBJECTS="Provozovna" BASECLASS=Objednavka_Prodej
VIEW_CONDITION="AND[Objednavka_Prodej] OID = #Objednavka_Prodej#;"
PAGETOPMARGIN=40 PAGELEFTMARGIN=25 PAGEBOTTOMMARGIN=5>

<!BAND NAME=REPORTHEADER CLASS=REPORT_HEADER>
    -- html hlavicky --
</BAND>

<!BAND CLASS=DETAIL>
    -- html pro zobrazovany radek --
</BAND>

<!BAND CLASS=REPORT_FOOTER ALLPAGES=TRUE>
    -- html paticky --
</BAND>

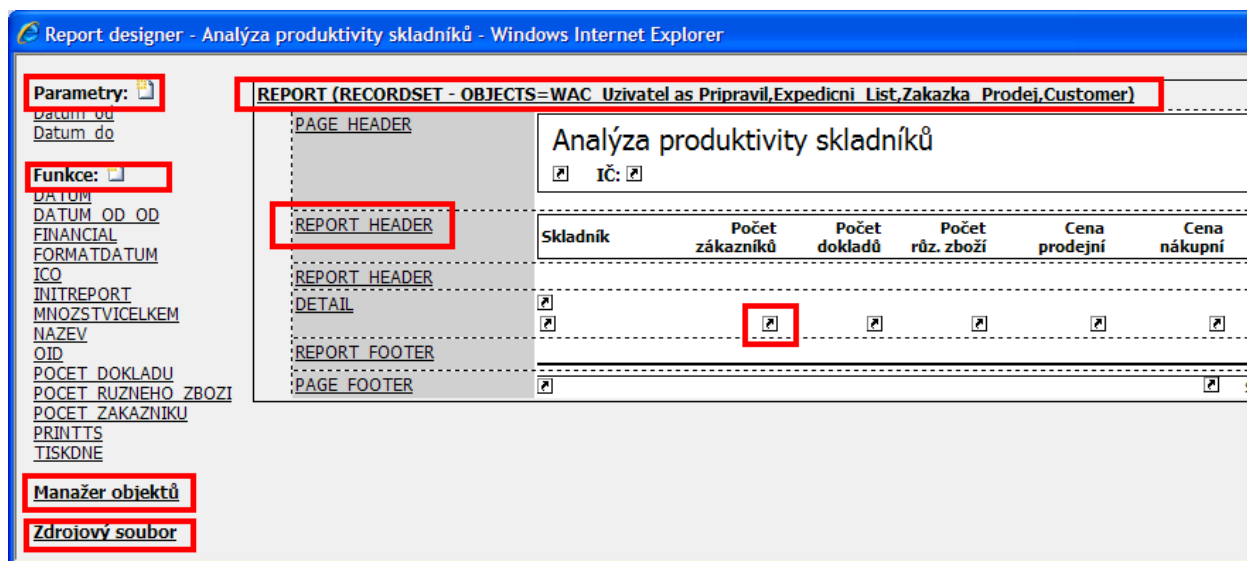
</REPORT>

</BODY>
```

S tímto druhem zápisu se setkáte pouze při pohledu na sestavu pomocí odkazu „Zdrojový soubor“, který je určen pouze pro čtení. Využití tohoto pohledu Vám umožní rychle prohledat sestavu při hledání nějaké vlastnosti nebo programové konstrukce.

Při uživatelské tvorbě sestav se setkáte s přívětivější formou pohledu na sestavu, který lze získat kliknutím na ikonu „designer“





Červeně orámované jsou základní vlastnosti (struktury) sestavy, které jsou rozepsány podrobně dále v textu.

3.3 Interní tagy

Jak si můžete všimnout v příkladu, jazyk HTML je doplněn o řadu dalších interních tagů, které se zapisují ve tvaru `<!NAZEVTAGU parametry>`. Takovými tagy jsou REPORT, BAND, ITEM, H, E a PARAMETERS. Další podkapitoly se budou věnovat jednotlivým tagům a jejich definicím.

S těmito tagy se přímo do kontaktu nedostanete (kromě tagů H a E, které slouží pro zápis exportovaných hodnot), jelikož Vám jsou zprostředkovávány pomocí dialogů designera, ve kterých lze modifikovat veškerá nastavení.

3.3.1 REPORT

Základním kamenem každé tiskové sestavy je tag REPORT. Reporty slouží k prvotnímu přístupu k databázovému modelu a jedná se o přepis systémových volání a dělí se tedy obdobně na report pro vytvoření kolekce (jednodušší zápis, omezené možnosti) a pro vytvoření recordsetu. V této části budou rozepsány atributy, které jsou společné pro obě varianty.

1. PAGESIZE:integer – šířka generované stránky (710px nebo 1040px)
2. PAGEHEIGHT:integer – výška stránky na kterou se bude generovat
3. PAGELEFTMARGIN,
PAGETOPMARGIN,
PAGERIGHTMARGIN,
PAGEBOTTOMMARGIN:integer – nastavení okrajů reportu
4. PRINTEMPTY:bool – tisknou i prázdný report
5. NAME:string – název, nemá žádného praktického významu, pouze pro označení
6. SERVERFORMAT:bool - sestava se bude formátovat na serveru, u všech bandů je třeba zadat atribut BANDHEIGHT
7. ONPRINT:string = volání funkce (pokud vrátí true tiskne se, jinak ne)

8. RELATEDOBJECT: umořňuje vbr polořek pomocí definice vztařench objekt.
9. VIEW_CONDITION:string = podmnka sestavena na zakld parametr, např.
AND[#PARAMETER1#] OID=#PARAMETER1#;

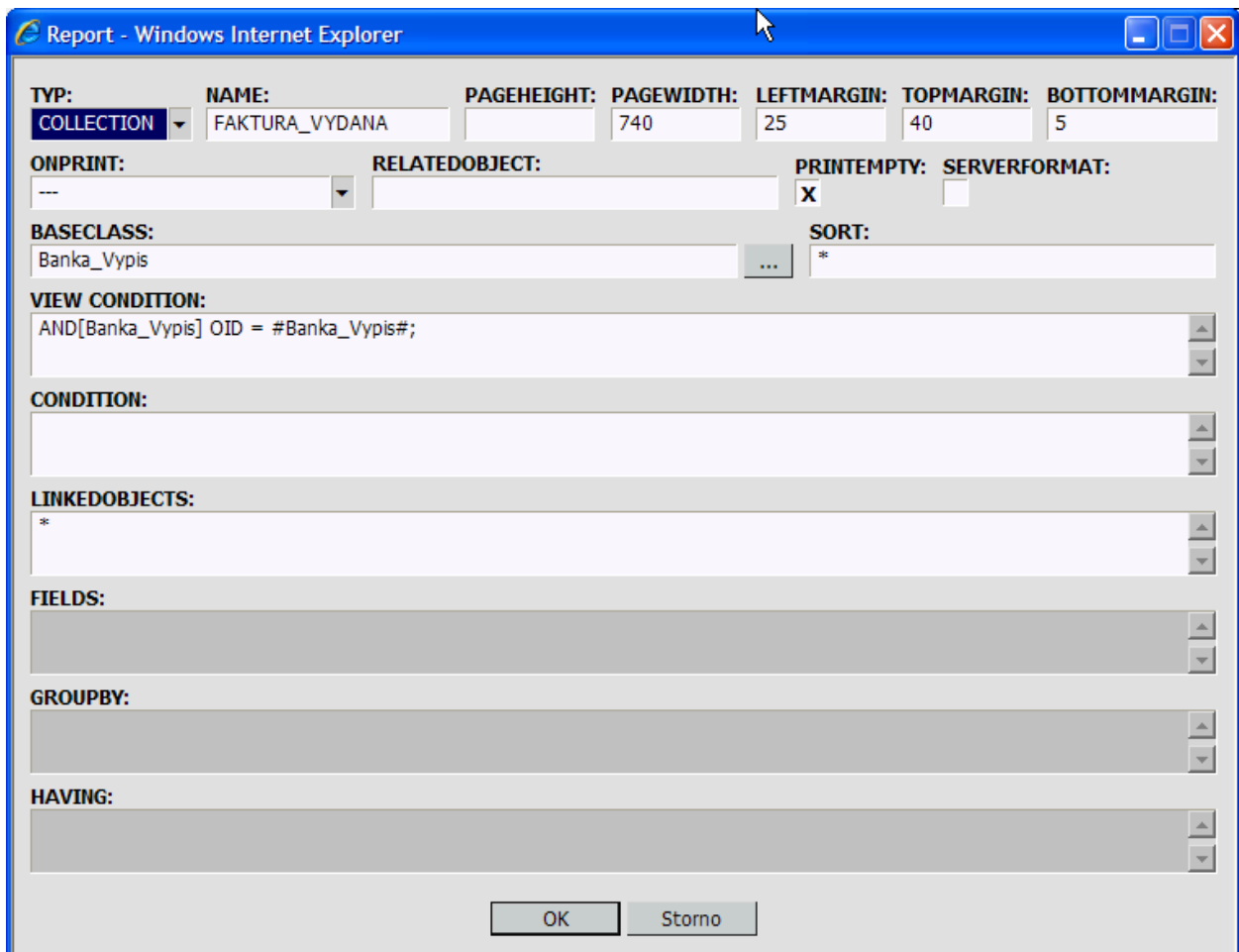
Upozornn: V deklaraci REPORTU (pokud nen editovn přes designer) je třeba překldat vřechny znaky „<“ a „>“ za HTML entity < a >.

Kařd report je realizovn buďto kolekc nebo recordsetem. Tuto volbu provd tvůrce sestavy a odpovd tomu i zpis reportu. Dle budou rozvedeny pouze atributy souvisejc s danm typem.

3.3.1.1 Kolekce

V souladu s definic kolekce jsou i atributy tagu REPORT.

1. BASECLASS:string – vchoz objekt kolekce
2. SORT:string – pole, podle kterho se bude řadit
3. CONDITION:string – podmnka v SQL syntaxi
4. USELINKEDOBJECT:bool - pokud ano, tak jsou připojeny (joinovny) vřechny souvisejc objekty oznaen atributem „Preview“
5. LINKEDOBJECTS:string – definice linkovanch (joinovanch) objekt – syntaxe stejn jako u kolekce



Report - Windows Internet Explorer

TYP: **COLLECTION** NAME: FAKTURA_VYDANA PAGEHEIGHT: PAGEWIDTH: 740 LEFTMARGIN: 25 TOPMARGIN: 40 BOTTOMMARGIN: 5

ONPRINT: --- RELATEDOBJECT: PRINTEMPTY: SERVERFORMAT:

BASECLASS: Banka_Vypis SORT: *

VIEW_CONDITION: AND[Banka_Vypis] OID = #Banka_Vypis#;

CONDITION:

LINKEDOBJECTS: *

FIELDS:

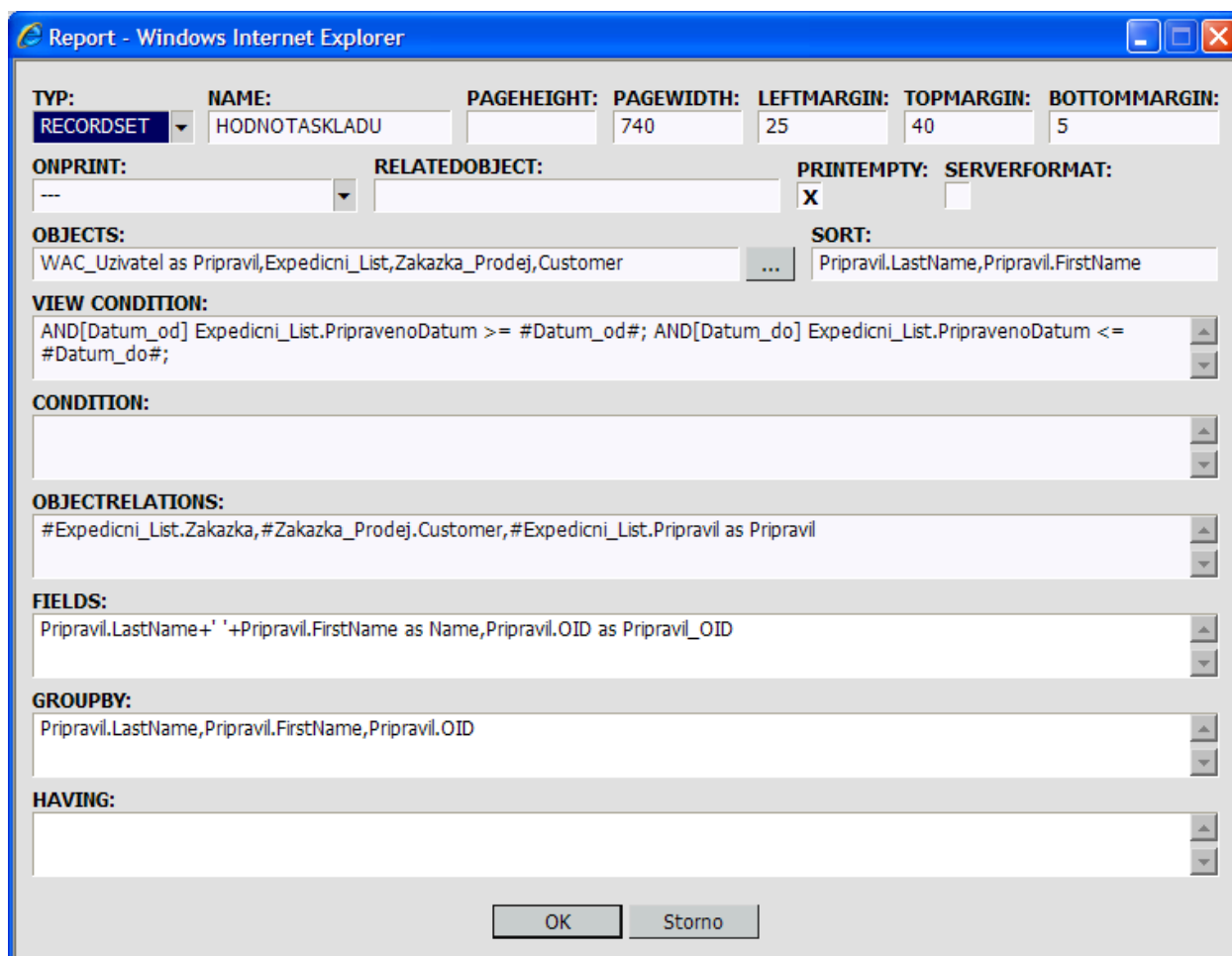
GROUPBY:

HAVING:

OK Storno

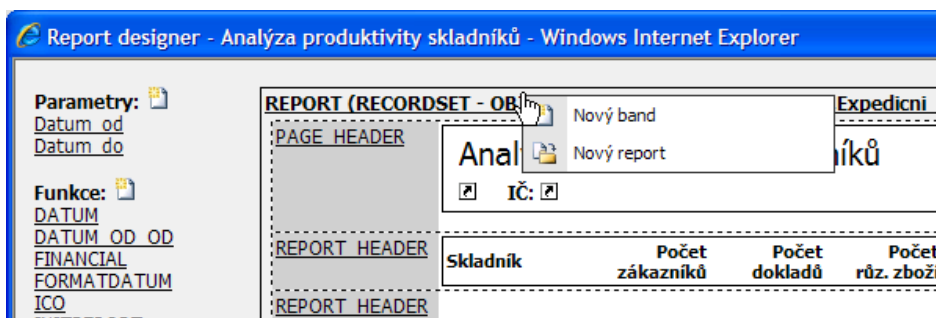
3.3.1.2 Recordset

1. FIELDS:string – pole, které mají být v reportu přístupné
2. OBJECTS:string – všechna použitá pole
3. GROUPBY:string – SQL seskupení
4. HAVING:string – SQL having
5. SORT:string - řazení
6. CONDITION:string – podmínka v SQL syntaxi
7. OBJECTRELATIONS:string – definice spojení (joinů) – syntax stejná jako v recordsetu



3.3.1.3 Vytvoření zanořeného reportu

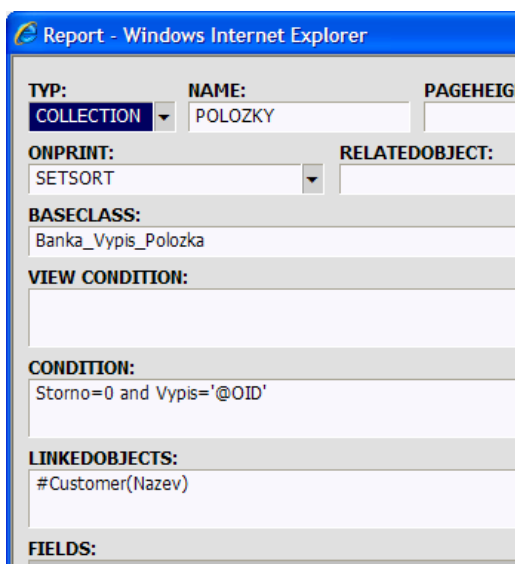
Pokud potřebujete do hlavního reportu zanořit další report (např. pro výpis položek nějakého dokladu), je možné to provést pomocí kontextového menu nad reportem – kliknutím na „Nový report“.



3.3.1.4 Další zajímavosti

Reporty je možné vnořovat do sebe – poté plní vnořený report jakousi obdobu BAND třídy DETAIL, ale s širšími možnostmi.

- @OID v podmínce u zanořeného reportu se nahradí za OID nadřazene kolekce



3.3.2 BAND

Samotný report nám zpřístupní pouze požadovaná data (řádky), ale již je nejsme pomocí něj schopni zobrazit. Pro tento účel se používá tag BAND, jehož význam určuje atribut CLASS.

Základním požadavkem na BAND je nezávislost na okolí, takže každý BAND musí být definován pomocí uzavřeného, konzistentního HTML kódu (např. celé tabulky, ne pouze jejich řádek). Tento požadavek je dám tím, že obsah BANDu je obalován dalším HTML kódem, který je generován při sestavování sestavy.

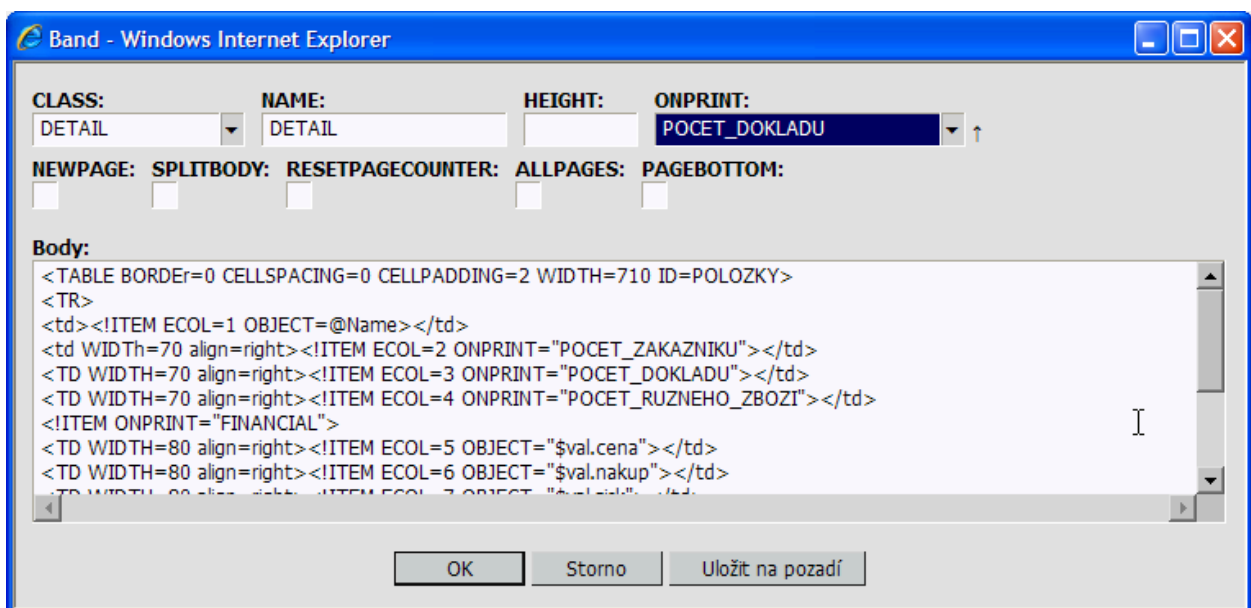
1. NAME – název, slouží pouze pro debugování
2. CLASS – [REPORT_HEADER|REPORT_FOOTER|PAGE_HEADER|PAGE_FOOTER|RECORD_HEADER|RECORD_FOOTER|DETAIL|STATIC]
3. SPLITBODY:boolean - bude rozdělovat obsah bandu na více tiskových stran (není možné definovat kde se rozdělení provede – použití pouze pro dlouhé texty).
4. ONPRINT:string = volání funkce (pokud vrátí true tiskne se, jinak ne)
5. NEWPAGE=ALWAYS - vynutí novou stránku před tímto bandem

6. PAGEBOTTOM:bool – umístí band na konec stránky (nad PAGE_FOOTER), prostor vyplní prázdným DIVem
7. ALLPAGES:bool – (jen pro PAGE_HEADER a PAGE_FOOTER) – band bude renderován na všech stránkách
8. RESETPAGECOUNTER:bool – nastaví počítadlo stránek na „1“
9. BANDHEIGHT:integer - nastaví výšku bandu pro serverové formátování (nutné pro serverside formátování sestavy)

3.3.2.1 Popis jednotlivých tříd bandů

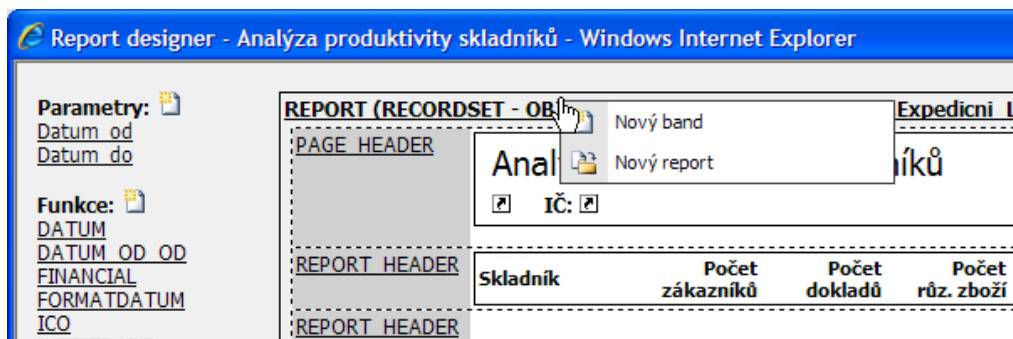
Tučně zobrazené třídy jsou povinné a musí se vyskytovat ve všech reportech (i pokud neobsahují žádný obsah).

- **REPORT_HEADER** – může jich být více, lze použít ALLPAGES, používá se zpravidla pro zobrazení názvu sloupců nad datovou částí
- **REPORT_FOOTER** – podobná funkce jako REPORT_HEADER, pouze se zobrazuje na konci, jinak platí stejná pravidla
- **RECORD_HEADER** – lze použít ALLPAGES, platnost je jen v rámci jednoho záznamu
- **RECORD_FOOTER** – obdoba REPORT_FOOTER, ovšem nad záznamem
- **PAGE_HEADER** - platnost v celem reportu
- **PAGE_FOOTER** – dtto
- **DETAIL** – řídicí band reportu, dá se pracovat s nadřazeným reportem, buildování reportu prochází následující detail bandy případně zanořený report, potom přesun na další řádek
- **STATIC** – není řízen reportem, opakování bandu je řízeno proměnnou Report("BandRepeat"), nutno explicitně nastavit na TRUE.



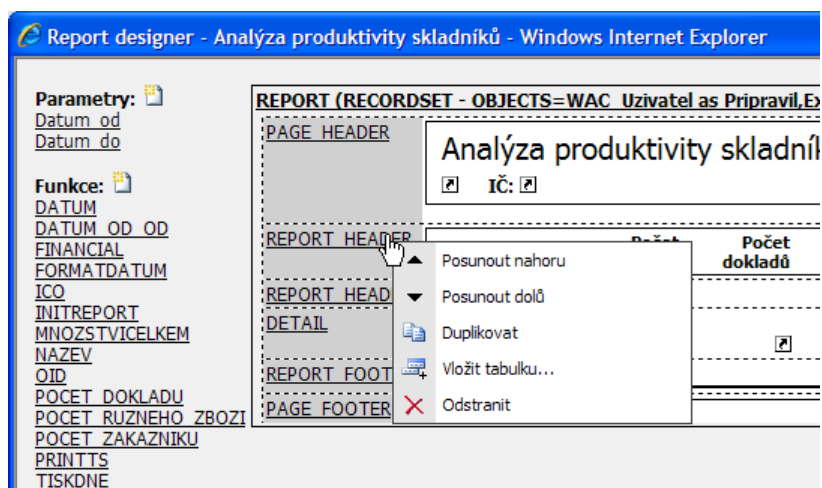
3.3.2.2 Vytvoření nového BANDu

Do reportu je často nutné zařadit další bandy, které nebyly při předgenerování. Toto se provádí v kontextovém menu na reportem, kliknutím na „Nový band“.



3.3.2.3 Úpravy existujícího BANDu

Nad existujícím bandem je možné provádět řadu akcí, které jsou přístupné přes kontextové menu nad názvem bandu.



Kontextové menu nabízí:

1. „Posunout nahoru“ - posune band nahoru v rámci reportu
2. „Posunout dolů“ - posune band dolů v rámci reportu
3. „Duplikovat“ - vytvoří kopii bandu (nový band) a vloží jej nakonec reportu
4. „Vložit tabulku“ - slouží k ulehčení práce při vkládání tabulky (nejčastější obsah), kliknutím se otevře průvodce (o dvou krocích), který vkládá tabulku pro zobrazení základních třech typů obsahu:

- ITEM – vygeneruje do každé buňky prázdný ITEM
- popiska – obsah buněk bude zobrazen standardním textem
- hlavička – obsah buněk bude zobrazen tučně

Tabulka je přidána vždy na konec bandu (pokud již něco obsahuje).

5. „Odstranit“ - odstraní band z reportu

Vlastní kód v jazyce HTML je zobrazen v poli „body“ a uložení bandu je možné dvěma způsoby (tlačítkami):

1. „OK“ - provede uložení a uzavření dialogu
2. „Uložit na pozadí“ nebo kl. zkratka **CTRL+S** – provede uložení na pozadí, po takovém uložení se i obnoví náhled sestavy na pozadí, takže je možné takto rychle zobrazovat úpravy provádné v kódu bandu.

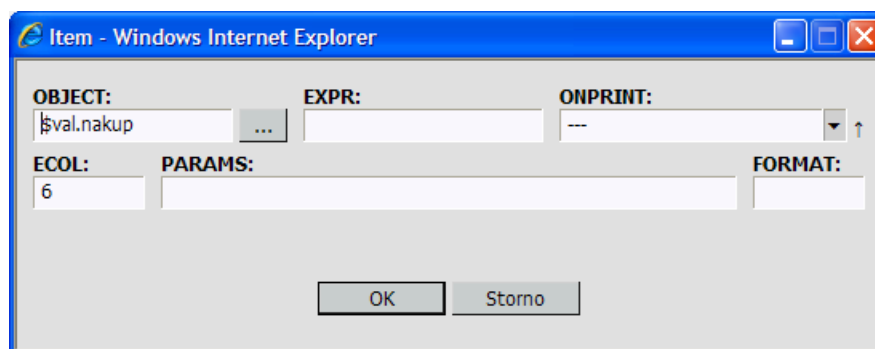
3.3.3 ITEM

Použití pro přímý výpis vlastností z databáze nebo ve spojení s funkcemi k výpisu jejich návratové hodnoty. Jedná se o jediný přístup tisku dynamických dat.

- OBJECT:string – prefix
 - nic – objekt
 - @ - data z kolekce, rychlejší
 - \$ - data z reportu

je možno použít "." notaci

- EXPR:string – matematický výraz
- FORMAT:string – formátování hodnoty
 - číslo - počet desítných míst
 - *- formátování měny (dle Report("Mena"), výchozí nastavení měny)
 - ** - z měny pro celkovou částku
 - \$ - formátování dle nastavení pro množství
 - MĚNA – zkratka
- ONPRINT:string – odkaz na funkci – vytiskne se návratová hodnota
- ECOL:bool – export pole (ekvivalentní exportovaná hodnota), pořadí sloupce v exportu



3.3.4 H

Tag H se používá pro export hodnot hlavičky. Pokud se v sestavě vyskytuje alespoň jeden, tak se povolí tlačítko „Export“ v záhlaví tiskové sestavy a sestava půjde exportovat do CSV.

Používá se v zápisu `<!H1.1></H>`, `<!H2.2></H>`, ..., `<!HX.X></H>`, kde příslušné číslo určuje pořadí v exportovaném souboru. Je možné použít i zápis `<!HX.Y></H>`, které slouží k zopakování hlavičky dále v exportu – podstata je taková, že hlavička je tisknuta pouze

tehdy, pokud část Y (za tečkou) je větší než poslední nejvyšší použitá hodnota při tisku hlavičky (příliš se nepoužívá).

3.3.5 E

Tag E je možné použít při exportu hodnot, které nejsou generované přes ITEM. Použití je podobné jako u H pouze není nutné zadávat tečkovou notaci – jednoduše <!E1></E>

3.3.6 PARAMETERS

Část sestavy uzavřená tagem PARAMETERS sloužící k definici parametru sestavy, které lze zadávat pomocí tlačítka „Parametry“ v záhlaví sestavy. Tyto parametry jsou jediným možným prostředkem, kterým lze ovlivňovat výstup sestavy. Parametry mohou být mnoha datových typů (včetně výčtu) nebo mohou být získávány z databáze (zpravidla číselníky) pomocí zkráceného zápisu kolekce.

3.3.6.1 Datový typ

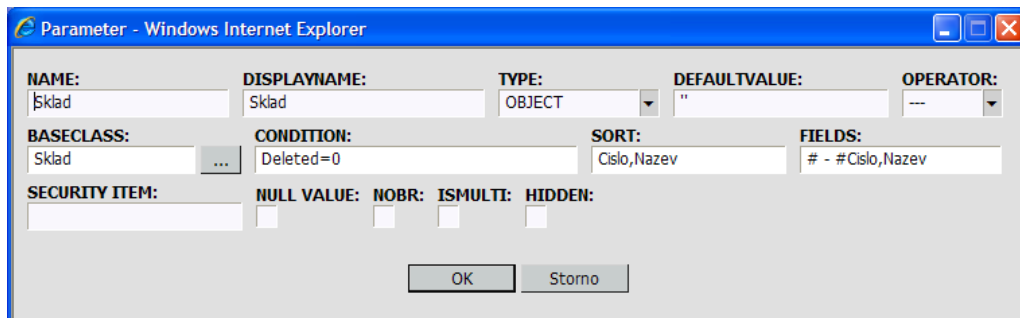
GOOD, OBJECT, STRING, DATE, INTEGER, BOOLEAN, MEMO, ENUM

Pro ENUM je možné zadávat výčet hodnot do pole „VALUES“ následovně:

hodnota:zobrazené_označení#další_hodnota:další_zobrazené_označení#....

3.3.6.2 Databázový objekt

Name:Název;TYP;výchozí hodnota;operátor;trida objektu;podminka;sort;zobrazena pole;

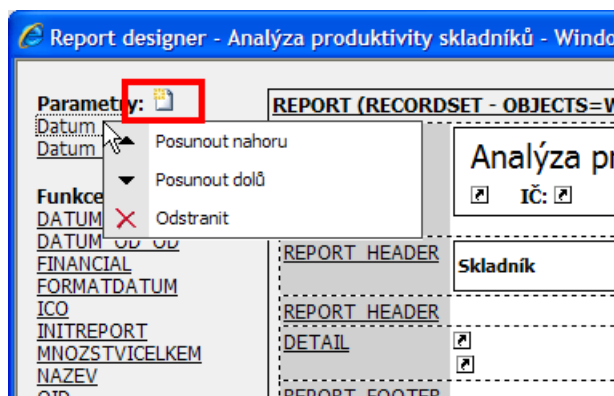


Vytvoření nového parametru lze provést kliknutím na ikonu vedle nápisu „Parametry“:



Poté stačí vyplnit pouze název, displayname, typ a případně další požadované vlastnosti.

Editace se provádí pomocí kontextového menu nad každým parametrem (pravé tlačítko myši), kde lze posunout daný parametr nahoru, dolů nebo odstranit.



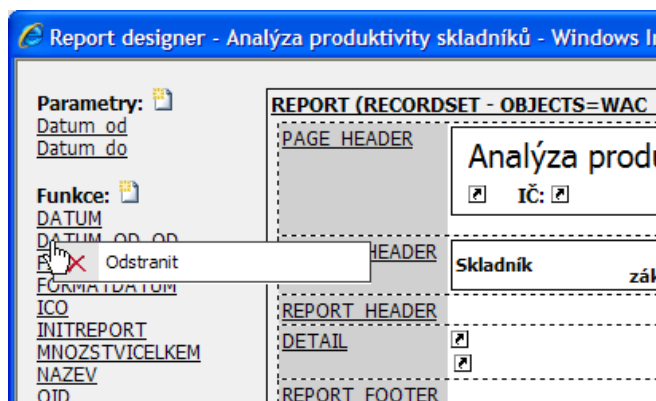
3.4 Funkce

Vytvoření nové funkce lze provést kliknutím na ikonu vedle nápisu „Funkce:“



Poté stačí vyplnit pouze název funkce – musí být tvořen názvem bez diakritiky a meze. Nesmí také začínat číslem.

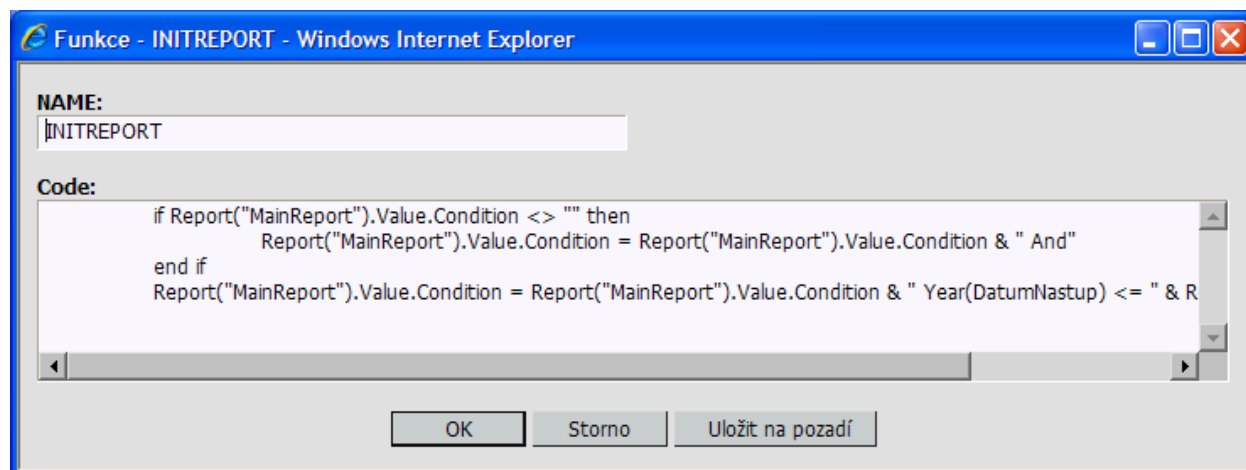
Programový kód se zapisuje (edituje) v poli „Code“. Odstranění funkce se provádí pomocí kontextového menu.



Funkce sestavy jsou uvedené v tagu SCRIPT a psané v jazyce VBScript. Jejich volání je zpravidla realizováno pomocí tagu ITEM a vlastností ONPRINT.

Uložení funkce je možné dvěma způsoby (tlačítkami):

3. „OK“ - provede uložení a uzavření dialogu
4. „Uložit na pozadí“ nebo kl. zkratka **CTRL+S** – provede uložení na pozadí, po takovém uložení si můžete vyzkoušet přegenerovat sestavu zda je vše OK a v případě, že jsou nutné další úpravy není nutné funkci znovu otevírat.



Ve funkcích máme možnost používat řadu předdefinovaných proměnných:

- Report – přístup k vlastnostem sestavy, globální proměnná
- thisObject – thisCollection.Object
- thisCollection – reprezentace aktuálního záznamu v kolekci nebo recordsetu
- SysUtils – řada užitečných funkcí, dostupných jako metody

- WACContext, WACApplication – jádro systému, kolekce, recordsety
- myWAC – přístup k nastavením systému myWAC
- value - hodnota ITEMu před ONPRINTem
- OnPrint – hodnota uložena v této proměnné je vráceno při dokončení funkce ITEMu

Možnosti ladění

Jedinou možností ladění sestav je využití chybového volání jádra, které ukončí vykonávání sestavy a zobrazí ladící informaci. Toto volání je zrealizováno pomocí

```
WACContext.Error.Register waccontext, "text", "detailni info", true, "info"
```

Mimo funkce není možné sestavy jakýmkoliv způsobem ladit.

3.5 Objekt Report

Předdefinovaný objekt Report je reprezentován myWAC kolekcí (asociativní pole, dynamická struktura), pomocí které lze přistupovat a ovlivňovat většinu vlastností reportu (dynamicky z funkcí ovlivňovat nastavení aktuálního REPORTU, předávat hodnoty mezi funkcemi) a jednoduše z něj tisknout pomocí tagu ITEM.

Základní hodnoty kolekce:

- MainReport – hlavní REPORT (nejvyšší v hierarchii)
- CurrentReport - aktuální report
- CurrentBand – aktuální band
- Parameter.XXX - přeepsané parametry sestavy, kde XXX označuje název parametru
- BandRepeat viz. STATIC
- Copyright
- ParameterString – polidštěná hodnota parametru

V případě odkazu na MainReport nebo CurrentReport je v kolekci uložen objekt s vlastnostmi Condition,Sort,...

Příklad odkazu na podmínku hlavního REPORTu:

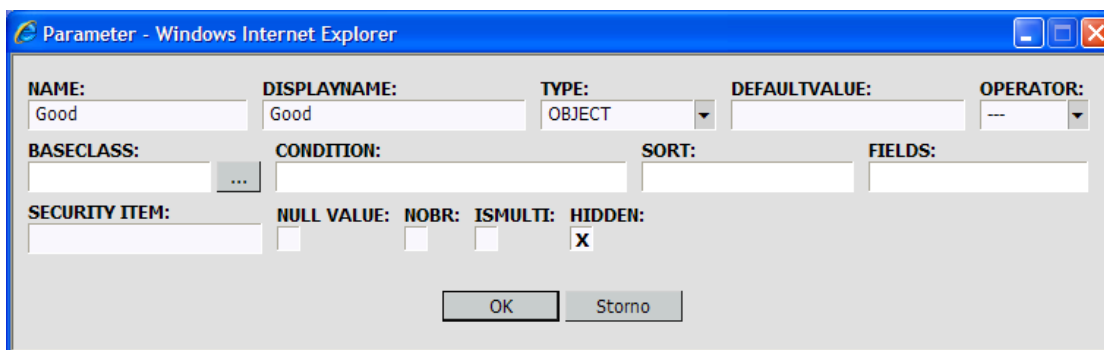
```
Report("MainReport").Value.Condition
```

4 Navázání sestavy na okolí

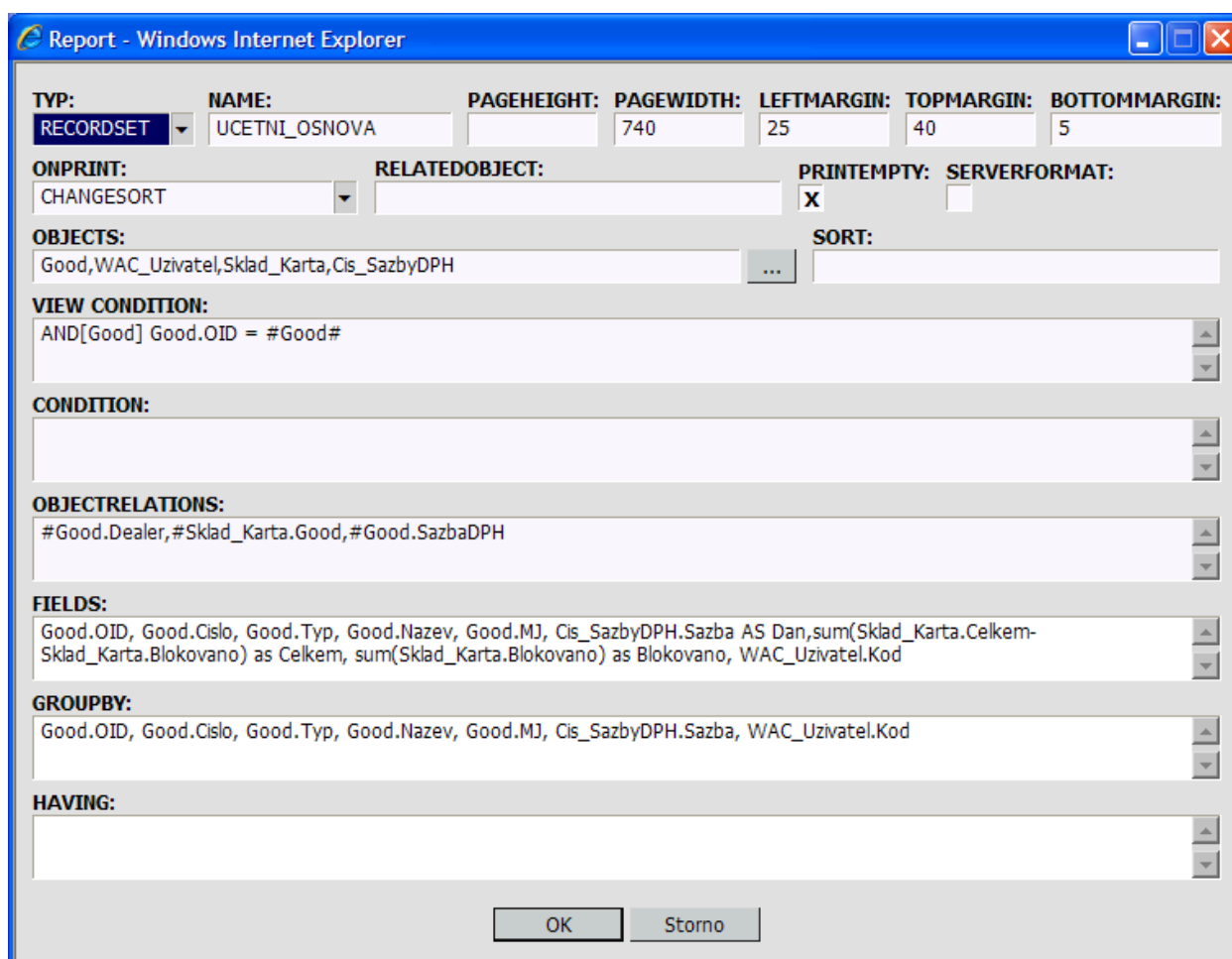
Sestavy systému myWAC jsou schopny z okolního systému přebírat některé vlastnosti – nejčastěji se jedná o zobrazení stejných dat, jako jsou označeny v aplikaci, ze které sestavu voláme nebo sařazení dle stejného sloupce jako v aplikaci.

4.2 Zobrazení stejných dat jako v aplikaci

Tento výběr dat se provádí pomocí speciálního parametru, který se musí jmenovat stejně jako objekt (třída objektů, které jsou označena data v aplikaci – ve skladu se jedná o objekt „Good“, parametr se bude tedy jmenovat Good).



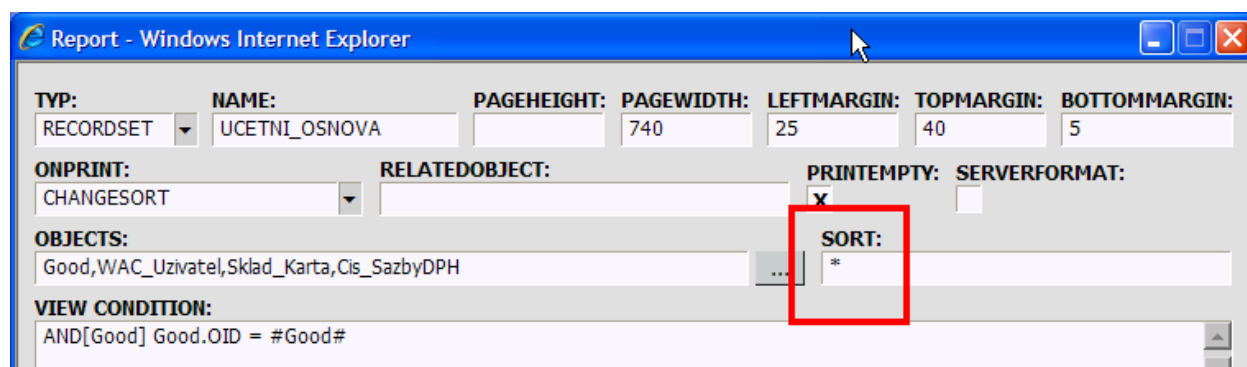
Takový parametr se také označí jako skrytý (HIDDEN), jelikož nemá pro uživatele přímo žádný význam (jeho význam je pouze systémový).



Poslední částí je zobrazení takového parametru ve VIEW CONDITION, která způsobí omezení dat na ta, která jsou vybrána v aplikaci.

4.3 Seřazení dle stejného sloupce jako v aplikaci

Pokud chceme do sestavy předat informaci o sloupci, dle kterého je seřazeno, stačí pouze do hlavního reportu, položky SORT, zapsat „*“. Zde je ale nutné zajistit, že v aplikaci se nebude dát řadit dle žádného jiného sloupce, než které jsou dostupné v reportu – tzn. jsou přidání ve vztazích objektech a podobně.



Report - Windows Internet Explorer

TYP: RECORDSET NAME: UCETNI_OSNOVA PAGEHEIGHT: PAGewidth: 740 LEFTMARGIN: 25 TOPMARGIN: 40 BOTTOMMARGIN: 5

ONPRINT: CHANGESORT RELATEDOBJECT: PRINTEMPTY: SERVERFORMAT:

OBJECTS: Good,WAC_Uzivatel,Sklad_Karta,Cis_SazbyDPH SORT: *

VIEW CONDITION: AND[Good] Good.OID = #Good#

V současné době, kdy dochází k nasazování nového DataGridu do aplikací v myWAC je možné v těchto aplikacích zobrazit téměř libovolný sloupec, který je v nějakém vztahu k aplikaci a tudíž není možné takto předávat informace o seřazeném sloupci, protože není možné zajistit, aby všechny sloupce byly v sestavě k dispozici.

Z tohoto důvodu použití této provázanosti nedoporučujeme.

5 Přílohy

Příloha 1: Přehled nejpoužívanějších HTML tagů

TABLE, TR, TD, DIV, BR, H1, H2, H3, H4, H5, B, I

Příloha 2: Přehled nejpoužívanějších CSS stylů

width, font-size, font-weight, float, color, text-decoration