

DSDaxDB

verze 0.1.03 24.8.2020

DSDaxDB

DSDaxDB je program, který umožňuje velmi rychle jednoduchým způsobem vytvořit databázovou aplikaci. Nastavení se provádí v konfiguračních souborech. Pro vytvoření aplikace tedy není třeba nic programovat ani instalovat žádné vývojové prostředí.

Pro vytvoření aplikace je potřeba udělat následující kroky.

1. Vytvořit SQL databázi. DSDax podporuje více typů databází.
2. Nastavit konfigurační soubory aplikace DSDaxDB
3. Vytvořit tiskové formuláře, pokud jsou potřeba tisky.

Velkou výhodou je možnost snadné následné úpravy řešení. Pokud je potřeba nějaké pole přidat nebo upravit, je to možné udělat snadno během velmi krátké doby.

Je možné snadno přidat další tiskový formulář, nebo ho upravit. Vytvářet a upravovat tiskové formuláře lze v programu MS Word, Libre Office nebo pomocí HTML.

DSDaxDB není třeba instalovat. Stačí ho jen uložit na disk nebo dokonce spouštět z přenosného disku.

DSDaxDB je naprogramován v C#. Podporované operační systémy jsou Windows 7 a vyšší. Pro běh je potřeba mít nainstalován NET Framework 4.5. (Ve Windows 10 už je součástí Windows, ve Windows 7 je nutné ho doinstalovat)

DSDaxDB je součástí projektu DSDax. DSDax je vývojové prostředí s vlastním skriptovacím jazykem určeným pro rychlé a efektivní zpracování dat.

Jednou z možností použití DSDaxDB je přístup k datům vytvořených jiným programem. Například účetnictvím nebo nějakým informačním systémem. V takovém případě je ale třeba dávat pozor na změny dat. Pokud nevíte jaká data je možné měnit může dojít k problémům, proto je doporučeno přistupovat k datům jen pro prohlížení.

Licence

DSDaxDB je poskytován zdarma pro komerční i nekomerční účely. Projekty v něm vytvořené můžete libovolně šířit a to zdarma i nebo zpoplatněně.

Jak začít

Nejvhodnější je prohlédnout si vzorové řešení.

Ve vzorovém řešení jsou v konfiguračních souborech komentáře.

SQLDatabáze

Použít můžete následující SQLDatabáze:

SQLite

Firebird - Standard i Embedded (a další kompatibilní - Interbase)

mySQL (a další kompatibilní - MariaDB)

Microsoft SQL

Libovolný zdroj dat přes ODBC

Připravuje se podpora pro:
PostgreSQL
XML ke kterému by se přistupovalo jako k databázi.

DSDax podporuje následující typy polí:
Int, Integer, Varchar, Char, Date, DateTime, Time, Decimal, Boolean

Ostatní typy zpracovává jako Varchar.

Není možné použít typy TEXT, BLOB, MEMO apod.

Pokud databáze neumí pole typu Boolean (True/False), tak použijte pole typu Integer, kde „0“ bude znamenat False a „1“ bude znamenat True. V DSDaxDB v tom případě nastavte typ „IntAsBoolean“ nenastavujte „Boolean“.

Pokud chcete mít možnost provádět editace záznamů, tak každá tabulka musí obsahovat jedinečné pole typu Autoincrement. Doporučen je typ Integer.

Je možné použít například i Varchar. Ale pokud to není nezbytně nutné, používejte typ Integer.

Každopádně je třeba zajistit, aby každý nový záznam dostal hodnotu vyšší než poslední předchozí záznam. A to je třeba zajistit na úrovni databáze.

Postup pro SQLite:

Musí být v tabulce pole, které bude Primary Key a Nastaveno Autoincrement.

Příklad:

```
ID INTEGER PRIMARY KEY AUTOINCREMENT
```

V jiných databázích může být nutné toto řešit například pomocí triggeru.

Pokud nebude zajištěno správné přidělování unikátního ID nebude program pracovat správně.

Poznámka:

DSDaxDB v nesíťové verzi zjišťuje ID nového záznamu pomocí zjištění nejvyšší hodnoty.

Pokud budete nesíťovou verzi provozovat z více stanic nebo více instancí programu může dojít ke kolizi.

Jakou zvolit SQLDatabázi

Nejlépe je zvolit databázi s kterou máte zkušenosti.

Pokud nemáte žádné preference, tak pokud nepočítáte v budoucnosti se síťovým provozováním, tak je vhodnou volbou SQLite. <https://www.sqlite.org>

Je rychlá, není třeba nic instalovat, protože DSDax ji má v sobě integrovánu.

Vzhledem k licenci je možné ji použít zdarma.

Pokud v budoucnu plánujete provoz zároveň z více stanic, tak je dobrá volba Firebird

<https://firebirdsql.org/>

Vzhledem k licenci je možné ji použít zdarma.

V naprosté většině aplikací je nejdůležitější volba, jaký typ databáze použít, protože následná změna je velmi komplikovaná. Vzhledem k tomu, jak je DSDax koncipován, je ke změně databáze nutné změnit jen dva řádky v konfiguraci. Samozřejmě bude nutné převést i data z jedné databáze do druhé.

Výhodou je, pokud nepoužíváte typy polí, které jsou dostupné jen v některých databázích, protože pak není nutné měnit typy polí.

Tiskové formuláře

Tiskové formuláře jsou řešeny tak, aby bylo jejich vytvoření rychlé a snadné.

Pro vytvoření formuláře je třeba vytvořit šablonu formuláře, do které pak bude DSDaxDB doplňovat měnící se údaje.

Jako šablonu je možné použít:

Soubory MSOffice DOCX nebo XLSX.

Soubory OpenOffice(LibreOffice) ODT nebo ODS.

Libovolný textový soubor. Například TXT. Ale i HTML, XML atd.

Při vytvoření šablony je třeba na místo, kam bude při tisku vložen údaj z databáze, vložit značku. Tato značka bude při tisku nahrazena skutečnou hodnotou z databáze.

Pozor:

Některé editory části značky obalují znaky pro změnu fontu apod. A pak se při tisku zobrazí hláška, že nelze najít značku pro pole.

To lze vyřešit tím, že se značka napíše v nějakém editoru, který toto nedělá, (Notepad, PSPad apod.) A do Wordu nebo Libre Office značku překopírovat pomocí Ctrl+C a Ctrl+V. Pak se značka vloží jako celistvý text.

Značka pro pole vypadá takto:

```
{{NAZEV_POLE}}
```

Kde místo „NAZEV_POLE“ vložíme název pole, jehož hodnotu chceme vložit.

Například značka pro pole JMENO bude tedy vypadat takto {{JMENO}}

V případě, že se jedná o typ comboboxdb, tak je možné tisknout pole z rodičovské tabulky spojené s tímto polem přes cizí klíč. Nejprve je nutné definovat příkazem Select dotaz pro získání obsahu polí. Tato definice se provádí v konfiguraci klíčem FieldForPrint.

Značka pro tisk pak vypadá následovně:

```
{{NAZEV_POLE.NAZEV_POLE_V_RODICOVSKE_TABULCE}}
```

Pokud bychom v šabloně potřebovali použít řetězec {{ pak ho musíme zapsat takto: {{{{}}

A obdobně pro řetězec }} použijeme {{{}}

V případě šablon DOCX nebo ODT je doporučeno vložit textové pole a značku vložit do něj.

Výsledný tisk je možné směřovat rovnou na tiskárnu nebo je možné ho zobrazit pro případné úpravy nebo odeslání emailem atd.

Při tisku mohou tisknout i údaje, které nejsou součástí formuláře.

V sekci EDIT nadefinuji pole pro tisk. Mohu buď použít přímou hodnotu nebo dotaz do SQL.

Příklad pro pole POKUS1 je přímá hodnota. Příklad pro pole POKUS2 je odkaz od databáze.

Polí může být neomezené množství.

`PrintField_POKUS1_Value=Pokusná hodnota 1`

`PrintField_POKUS2_ValueFromSQL=select max(CISLO)+1 from FAKTURY`

Na pole se v tiskové šabloně odkazují podobně jako na fyzická pole, ale první znak je >

Příklad:

`{{>POKUS1}} {{>POKUS2}}`

Popis syntaxe souboru .dsdaxdb

Těmito soubory se konfiguruje celý projekt.

Jedná se o modifikaci souboru typu INI.

Od standardního INI se liší takto:

Název klíče musí být od 1. sloupce v řádku (Před názvem nesmí být žádná mezera nebo tabulátor)

Název klíče nesmí obsahovat mezery, rovnítko musí být hned za názvem klíče. Obsah začíná hned za rovnítkem.

Nesmí obsahovat zdánlivě prázdné řádky, které nejsou prázdné, ale obsahují mezery!

Navíc podporuje víceřádkové klíče. Ty se používají ve speciálních případech. Pokud má obsah klíče pokračovat na dalším řádku, tak další řádek **musí** začínat mezerou (tu do obsahu nepočítáme)

Příklad:

`Klíč1=124`

`Klíč2=První Řádek`

`Druhý Řádek`

`Třetí řádek`

`Klíč3=454`

Podporovány jsou následující typy:

String

StringList - Víceřádkový string

Integer

Boolean - Používejte hodnotu 0 pro false a hodnotu 1 pro true

Je možné používat komentáře. Komentář musí být na samostatném řádku. Řádek musí začínat znakem ; (středník)

Soubor musí být v kódování UTF-8. Pokud použijete jiné kódování bude soubor čitelný, ale znaky s diakritikou budou zobrazovány špatně.

Projekt a řešení

Projekt (Project) je zobrazení jedné tabulky nebo jednoho sql dotazu přes více tabulek, popřípadě jednoho formuláře v režimu edit.

Řešení (Solution) zastřešuje jeden nebo více projektů.
Řešení není nutné používat. Vystačíme pouze s projekty.

Nastavení projektu

Projekt v programu DSDaxDB se nastavuje prostřednictvím konfiguračního souboru, který následně otevíráme. V tomto souboru musí být uložen typ a přihlašovací údaje k databázi. Dále tabulka kterou chceme otevřít. Případně další podmínky, dle kterých se výsledný projekt chová.

Sekce [DB]

Nejjednodušší způsob, jak nastavit konfiguraci pro připojení k databázi, je spustit DSDaxDB a v menu Tools/Nástroje vybrat Dialog Database Connection. Po vyplnění údajů se přímo vygeneruje potřebná část pro nastavení. Tu zkopírujte a vložte do konfigurace projektu.

V případě, že používáte nějak nestandardně nastavenou databázi, je nutné nastavit ConnectionString ručně dle dokumentace k databázi.

Pokud sekce [DB] není nastavena v nastavení projektu, tak hledá [DB] v nastavení řešení. (Více hledejte v Popisu řešení)

Sekce [BROWSE]

Toto nastavení řídí zobrazení v režimu Browse (Tabulka s více záznamy pod sebou)

BrowseDisable=True

Pokud uvedeme BrowseDisable=True do sekce BROWSE, tak zakážeme režim Browse a rovnou se otevře záznam v režimu Edit

V tom případě musí! být v sekci EDIT uvedena položka UniqueFieldValue v které bude uvedena hodnota pole UniqueField, podle které se určí, který záznam se má otevřít. Nebo 0 pokud se má záznam přidávat.

UniqueFieldValue=3

Tento režim se používá například pokud chceme jen jednoduše pořizovat nové záznamy a neumožnit obsluhu prohlížet ostatní položky.

Další použití je možné v případě, že chceme jediný záznam tabulky používat konfigurační. Typicky v případě fakturace si můžeme ukládat svoji adresu, která se bude zobrazovat na faktuře.

SQLCommand

Podmínkou je klíč „SQLCommand“

Je to klíč typu StringList, ale podmínkou je mít alespoň jeden řádek. Ten musí obsahovat buď jméno tabulky nebo příkaz Select s tím co chceme zobrazit. Příkaz Select může být i složitý příkaz přes více tabulek.

Před jménem tabulky nebo SqlCommandem je možné uvést název, který se bude případně zobrazovat v menu. Název a příkaz se odděluje středníkem. Název je nepovinný. Pokud by název nebyl použit a SqlCommand obsahoval středník, tak je nutné před SqlCommand dát středník.

Příkaz Select ... musí zobrazit i pole UniqueField. Jinak nebude možné používat režim edit.

V této verzi je výsledek příkazu Select celý nahrán do paměti. Pokud máte velkou databázi je třeba omezit velikost načítaných dat.

První příkaz je příkaz, který se načte automaticky při otevření režimu browse.

Příklad:

SQLCommand=Vše;ADRESAR

Je firma;Select * from ADRESAR where (JE_FIRMA=1);

Není firma;Select * from ADRESAR where (JE_FIRMA=0);

UniqueField

Další povinný klíč je: UniqueField

Musí obsahovat název unikátního pole. (Pole autoincrement, které má primary key)

Příklad:

UniqueField=ID

Dále je možné použít nepovinné klíče:

TableFilter

Filtruje zobrazení. SQLCommand načte požadovaná data a TableFilter umožní při jejich zobrazení použít další filtr. Z hlediska rychlosti a použití paměti je vhodnější data omezit co nejvíce pomocí SQLCommand. A pomocí TableFilter pak jen upravovat zobrazení.

TableFilter má podobný způsob zápisu jako SQLCommand. Hlavní rozdíl je v tom, že nesmíme použít celý příkaz Select, ale jen podmínky. Pokud chceme i možnost zobrazovat bez filtru, tak použijeme podmínku řetězec s jednou mezerou. Viz příklad.

Příklad:

TableFilter=Bez filtru;

Obsahující písmeno a;(NAZEV LIKE '%a%')

Obsahující písmeno b;(NAZEV LIKE '%b%')

Obsahující písmeno a nebo b;(NAZEV LIKE '%a%') or (NAZEV LIKE '%b%')

TableFilterDisable

Zakáže zobrazení editovatelného TableFiltru. V menu bude pořad možné filtry vybírat.

Hodnota 0 nebo 1

Default 0

TableFilterDisable=0

SQLCommandDisable

Zakáže zobrazení editovatelného SQLCommand. V menu bude pořad možné příkazy vybírat vybírat.

Hodnota 0 nebo 1

Default 0

SQLCommandDisable=0

OptionEditEnable

Povolí editovat záznamy v režimu Browse.

V tomto případě musí být SQLCommandem vybrána jen jedna tabulka.

Hodnota 0 nebo 1

Default 0

OptionEditEnable=1

OptionAddRowEnable

Povolí přidat záznamy v režimu Browse.

V tomto případě musí být SQLCommandem vybrána jen jedna tabulka.

Hodnota 0 nebo 1

Default 0

[OptionAddRowEnable=1](#)

OptionDeleteRowEnable

Povolí smazat záznamy v režimu Browse.

V tomto případě musí být SQLCommandem vybrána jen jedna tabulka, nesmí se jednat o dotaz přes více tabulek.

Hodnota 0 nebo 1

Default 1

[OptionDeleteRowEnable=1](#)

OptionFormEditRecordEnable

Povolí otevřít záznam v režimu Edit

Hodnota 0 nebo 1

Default 0

[OptionFormEditRecordEnable=1](#)

OptionFormAddNewRecordEnable

Povolí otevřít nový záznam v režimu Edit

Hodnota 0 nebo 1

Default 1

[OptionFormAddNewRecordEnable=1](#)

DisableViewColumns

Zakáže zobrazení vyjmenovaných sloupců.

Používá se zejména pokud nechcete zobrazit UniqueField v režimu Browse.

[DisableViewColumns=ID](#)

ReadOnlyColumns

Zakáže editaci vyjmenovaných sloupců v režimu Browse.

[ReadOnlyColumns=ID](#)

Column_NAZEV_ForeColor

Barva písmen sloupce.

Místo „NAZEV“ je nutné použít reálný název sloupce.

Povolené hodnoty: black, blue, red, white, yellow, green, gray, magenta, brown

Column_NAZEV_BackColor

Barva pozadí sloupce.

Místo „NAZEV“ je nutné použít reálný název sloupce.

Povolené hodnoty: black, blue, red, white, yellow, green, gray, magenta, brown

Column_NAZEV_Alignment

Zarovnání sloupce. Numerický typ se automaticky zarovnává doprava. Ostatní doleva.

Místo „NAZEV“ je nutné použít reálný název sloupce.

Tímto nastavením je možné nastavit ručně zarovnání.

Hodnoty: Right, Left, Center

FormCaption

Pokud je nastaveno, tak v titulku formuláře zobrazuje tento text.

`FormCaption=Adresář`

Sekce [EDIT]

Toto nastavení řídí zobrazení v režimu Edit. (Jeden záznam na formuláři)

Povinné klíče:

UniqueField

Další povinný klíč je: UniqueField

Musí obsahovat název unikátního pole. (Pole autoincrement, které má primary key)

Pokud ho vynecháme tak použijte hodnotu ze sekce BROWSE

Příklad:

`UniqueField=ID`

Table

Jméno tabulky, kterou chceme editovat.

`Table=ADRESAR`

Dále je možné použít nepovinné klíče:

Field

Obsahuje názvy polí, které se mají zobrazit, oddělené čárkou. Pokud se tento klíč nepoužije tak zobrazuje všechna pole, které obsahuje tabulka.

`Field=NAZEV,JMENO,PRIJMENI,ULICE,MESTO,PSC,TELEFON`

FieldVirtual

Obsahuje názvy polí, které fyzicky v tabulce neexistují. Jedná se například o relace do jiných tabulek. Nastavení těchto polí je pak povinně nutné udělat v nastavení jednotlivých polí!

`FieldVirtual=TYP_ZARIZENI,PRISLUSENSTVI`

ReadOnly

Určuje jestli je formulář jen pro čtení.

0 není readonly - lze editovat

1 je readonly - nelze editovat

2 je readonly v případě editace existujícího záznamu

3 je readonly v případě editace nového záznamu

Pokud je ReadOnly=0 pak je možné nastavovat ReadOnly pro jednotlivá pole. Viz popis u nastavení polí.

Pole UniqueField, pokud je zobrazeno je vždy ReadOnly, nelze to nastavením ovlivnit.

Default hodnota je 0

`ReadOnly=0`

OptionDeleteRecordEnable

Povolí smazat záznam v režimu Edit

Hodnota 0 nebo 1

Default 1

OptionDeleteRecordEnable=1

OptionConfirmSavingBeforeClosing

Jestli se má při zavření okna editace ptát na uložení záznamu nebo má uložit bez dotazu.

Default 0 (může být v dalších verzích změněno)

OptionConfirmSavingBeforeClosing=0

Nastavení jednotlivých polí

Tyto klíče se vztahují k jednotlivým polím.

Každé pole může mít vlastní nastavení. Nastavení pole má klíč s názvem začínajícím slovem Field pak následuje podtržítka pak následuje Název pole pak následuje podtržítka pak následuje určení toho co nastavujeme.

Příklad pro pole jménem „NAZEV“ a pro nastavení typu pole:

Field_NAZEV_Type=varchar(50)

Nastavovat můžeme následující parametry:

LabelPosition

Pozice, kde bude zobrazen popis pole.

Při hodnotě 0,0 bude pozice automaticky vypočítána

Field_NAZEV_LabelPosition=0,0

Position

Pozice, kde bude zobrazeno pole.

Při hodnotě 0,0 bude pozice automaticky vypočítána

Field_NAZEV_Position=0,0

Width

Šířka pole v pixelech

Field_NAZEV_Width=600

Height

Výška pole v pixelech

Má efekt pouze pro pole typu varchar, char nebo textbox

Field_NAZEV_Height=400

NameShow

Zobrazované jméno před polem

Default jméno z databáze

Field_NAZEV_NameShow=Jméno

Type

Typ pole

Pokud není typ pole nastaven, tak se bere výchozí typ z tabulky databáze.

To ale nemusí být vždy správně. DSDaxDB nedokáže některé typy správně odlišit.

Například typ Date od DateTime. Dále některé databáze neumí typ Boolean (True/False) a místo toho se používá Integer kde 0 znamená Ne a 1 znamená Ano. To je ale potřeba ve formuláři nastavit, protože to nelze ze struktury databáze zjistit.

Dále je možné v tabulce zobrazit například pole z jiných tabulek. Viz FieldVirtual.

Field_NAZEV_Type=varchar(50)

Možné typy:

varchar

char

Popřípadě varchar(n) a char(n) kde n je maximální délka pole.

Text. Pokud je zadána maximální hodnota, tak hlídá aby nebyla překročena délka pole.

boolean

Hodnota typu boolean. Logická hodnota True/False Ano/Ne

Způsob ukládání se řídí syntaxí použité databáze.

intboolean

Hodnota typu boolean. Logická hodnota True/False Ano/Ne

Způsob ukládání je vždy 0 nebo 1. Používá se, pokud pole typu integer používáme za logickou hodnotu.

datetime

Hodnota typu datetime

Je možné v SQLdatabázi nastavit pole typu Datetime nebo Varchar.

date

Hodnota typu date

Je možné v SQLdatabázi nastavit pole typu Date, pokud to databáze podporuje nebo Varchar.

time

Hodnota typu time

Je možné v SQLdatabázi nastavit pole typu Time pokud to databáze podporuje nebo Varchar.

comboboxtext

Zobrazí combobox s výběrem hodnoty, která se vloží. Předpoklad je, že v databázi je typ pole char nebo varchar. Ale funguje i na jiné hodnoty, které se dají reprezentovat jako string.

Je třeba použít klíč parameter, který bude obsahovat nabízené hodnoty. Hodnoty jsou odděleny znakem |

Příklad:

Field_TYP_KONTAKTU_Type=comboboxtext

Field_TYP_KONTAKTU_Parameter=Přátelé|Dodavatelé|

comboboxtextedit

Obdoba comboboxtext. Rozdíl je v tom, že hodnota dá i editovat a není omezena na definované hodnoty.

comboboxid

Zobrazí combobox s výběrem textu a ID, který se vloží.

Je třeba použít klíč parameter, který bude obsahovat nabízené hodnoty. ID a Text jsou odděleny středníkem. Hodnoty jsou odděleny znakem |

Do pole se ukládá ID

Příklad:

Field_TYP_KONTAKTU_Type=comboboxid

Field_TYP_KONTAKTU_Parameter=1;Přátelé|2;Dodavatelé|

V tomto případě se v combobox nabízí Přátelé a Dodavatelé. Ale do pole se ukládá 1 nebo 2. Hodnota může být numerická nebo textová.

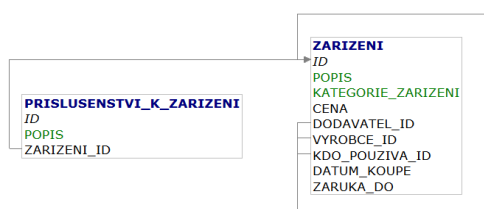
Trocha teorie pro pochopení:

Cizí (Foreign) klíč je pole v dceřinné tabulce, který odkazuje na pole v rodičovské tabulce. V rodičovské tabulce by se mělo jednat o pole s PrimaryKey. (V našem případě je to pole, které nazýváme UniqueField)

Tabulka PRISLUSENSTVI_K_ZARIZENI je dceřinná tabulka

Pole ZARIZENI_ID v tabulce PRISLUSENSTVI_K_ZARIZENI je cizí klíč

Tabulka ZARIZENI je rodičovská tabulka



... FOREIGN KEY (ZARIZENI_ID) REFERENCES ZARIZENI (ID) ON DELETE RESTRICT ON UPDATE RESTRICT ...

comboboxdb

Předpokládá se, že pole je integer. Ale není to podmínka.

Zobrazí combobox s výběrem textu a ID, který se vloží.

Z hlediska cizích klíčů je editovaná tabulka dceřinná tabulka. A editované pole je cizí klíč.

Hodnoty a ID se berou z rodičovské tabulky.

Je třeba použít klíč parameter, který bude obsahovat SQL Select příkaz, pro výběr z rodičovské tabulky.

Select musí vrátit dva sloupce. ID a TEXT. Sloupce musí být v pořadí ID jako první. TEXT jako druhý.

Combobox nabízí názvy, ale do tabulky ukládá ID.

Je doporučeno nadefinovat si foreign (cizí) klíč, pro to aby si databáze hlídala integritu dat.

Pokud je požadavek na tisk polí, je třeba definovat parametr FieldForPrint

Tento parametr musí obsahovat všechny pole, které chceme tisknout s klauzulí where omezující výběr záznamu.

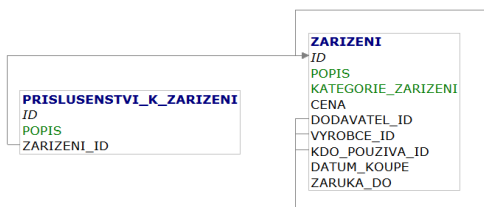
Příklad foreign key: ... FOREIGN KEY (ZARIZENI_ID) REFERENCES ZARIZENI (ID) ON DELETE RESTRICT ON UPDATE RESTRICT ...,

Příklad:

Field_ZARIZENI_ID_Type=comboboxdb0

Field_ZARIZENI_ID_Parameter=select ZARIZENI.ID as ID,POPIS as TEXT from ZARIZENI order by POPIS;

Field_DODAVATEL_ID_FieldForPrint=select * from ADRESAR where ID={0};



comboboxdbnull

Obdoba comboboxdb. Rozdíl je v tom, že je možné uložit i ID=NULL . Pokud chceme zvolit, že ani jedna hodnota není platná. (Záznam nemá rodiče)

comboboxdb0

Obdoba comboboxnull. Rozdíl je v tom, že je možné uložit i ID=0. Pokud chceme zvolit, že ani jedna hodnota není platná.

V tomto případě pozor při vytváření foreign key. Existující klíč by nedovolil vytvoření hodnoty 0. Pokud nemáte nějaký vážný důvod, tak použijte přednostně comboboxdbnull.

relationship_1n

Jedná se o virtuální pole. Název musí být uveden ve FieldVirtual

Jedná se o relaci 1:N do jiné tabulky.

Může být použito jako opak oproti comboboxdb.

Editovaná tabulka je v tomto případě rodičovská.

Parameter1 je SQL příkaz pro výběr nabízených hodnot z dceřinné tabulky. Syntaxe je stejná jako u comboboxdb.

Parameter2 je SQL příkaz pro výběr nabízených hodnot z dceřinné tabulky. SQL příkaz je doplněn omezením where, kde je uveden název pole cizí klíč v dceřinné tabulce dle kterého je relace do rodičovské (editované) tabulky přes pole které označujeme jako UniqueField. V SQL příkazu kde by měla být hodnota UniqueField záznamu který editujeme, nahradíme řetězcem {0}

Parameter3 je název dceřinné tabulky.

Parameter4 je název pole cizí klíč v dceřinné tabulce.

Parameter5 je název pole PrimaryKey v rodičovské (Editované) tabulce. Mělo by být shodné s UniqueField

Příklad editace tabulky ZARIZENI.

Field_PRISLUSENSTVI_Type=Relationship_1N

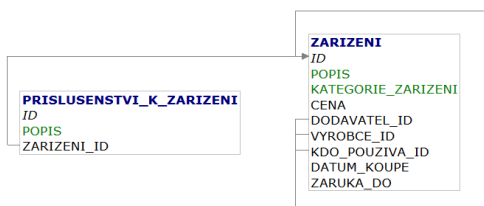
Field_PRISLUSENSTVI_Parameter1=select ID as ID, POPIS as TEXT from PRISLUSENSTVI_K_ZARIZENI order by POPIS;

Field_PRISLUSENSTVI_Parameter2=select ID as ID, POPIS as TEXT from PRISLUSENSTVI_K_ZARIZENI where (ZARIZENI_ID={0});

Field_PRISLUSENSTVI_Parameter3=PRISLUSENSTVI_K_ZARIZENI

Field_PRISLUSENSTVI_Parameter4=ZARIZENI_ID

Field_PRISLUSENSTVI_Parameter5=ID



relationship_mn

Jedná se o virtuální pole. Název musí být uveden ve FieldVirtual

Jedná se o relaci M:N do jiné tabulky pomocí spojovací tabulky.

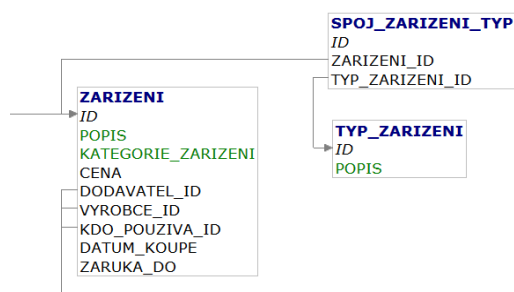
Ve spojovací tabulce musí být jen pole: ID (Autoincrement+PrimaryKey), Pole ID z tabulky kterou editujeme a pole ID z druhé tabulky. Pokud bychom tam měli i jiné pole, tak o jejich obsah přijdeme! DSDax při editaci záznamy ze spojovací tabulky maže a znovu vkládá.

Je doporučeno mít nadefinovány ForeignKey, aby bylo zajištěno, že databáze bude hlídat integraci dat.

Nejjasněji to ukáže níže uvedený příklad.

Námi editovaná tabulka je ZARIZENI. Z hlediska terminologie ForeignKey je tabulka SPOJ_ZARIZENI_TYP rodičovská. Tabulky ZARIZENI a TYP_ZARIZENI jsou dceřinné tabulky.

Editujeme tedy jednu z dceřinných tabulek



Parameter1 je SQL příkaz pro výběr nabízených hodnot z druhé dceřinné tabulky (tedy ne té, kterou editujeme). Syntaxe je stejná jako u comboboxdb.

Parameter2 je SQL příkaz pro výběr nabízených hodnot z druhé dceřinné tabulky prostřednictvím spojovací tabulky. Viz příklad.

V SQL příkazu, kde by měla být hodnota UniqueField záznamu, který editujeme, nahradíme řetězcem {0}

Parameter3 je název spojovací tabulky.

Parameter4 je název pole cizí klíč ve spojovací tabulce vztahující se k námi editované (první dceřinná) tabulce

Parameter5 je název pole cizí klíč ve spojovací tabulce vztahující se k druhé dceřinné tabulce.

```
Field_TYP_ZARIZENI_Type=Relationship_MN
```

```
Field_TYP_ZARIZENI_Parameter1=select ID as ID,POPIS as TEXT from TYP_ZARIZENI order by POPIS;
```

```
Field_TYP_ZARIZENI_Parameter2=select TYP_ZARIZENI.ID as ID, TYP_ZARIZENI.POPIS as TEXT from
TYP_ZARIZENI,SPOJ_ZARIZENI_TYP where (TYP_ZARIZENI.ID=SPOJ_ZARIZENI_TYP.TYP_ZARIZENI_ID) and
(SPOJ_ZARIZENI_TYP.ZARIZENI_ID={0}) order by TYP_ZARIZENI.POPIS;
```

```
Field_TYP_ZARIZENI_Parameter3=SPOJ_ZARIZENI_TYP
```

```
Field_TYP_ZARIZENI_Parameter4=ZARIZENI_ID
```

```
Field_TYP_ZARIZENI_Parameter5=TYP_ZARIZENI_ID
```

Konec popisu TYPE

DefaultValue

Hodnota která se předvyplní při vytváření nového záznamu

Není možné použít zároveň DefaultValue a DefaultValueFromSQL.

Field_NAZEV_DefaultValue=ABC

DefaultValueFromSQL

Hodnota která se předvyplní při vytváření nového záznamu. Uvádíme SQL command, kterým tuto hodnotu chceme zjistit.

Není možné použít zároveň DefaultValue a DefaultValueFromSQL.

Field_CISLO_DefaultValueFromSQL=select max(CISLO)+1 from FAKTURY

ReadOnly

Nastaví pole jen pro čtení. Hodnoty jsou shodné s klíčem ReadOnly v sekci EDIT

Field_NAZEV_ReadOnly=0

Parameter

Parameter1

Parameter2

Parameter3

Parameter4

Parameter5

Upřesňuje chování pro některé typy polí. Popis je uveden u TYPE

Konec Popisu Nastavení jednotlivých polí

Label

Mohu zobrazovat text. Je to obdoba Label pro jednotlivá pole. Ale tento text není závislý na polích. Label může být více řádků. Co řádek to jeden text.

Uvádí se pozice;text;PřípadněColor;PřípadněColorBackground

Label=100,100;Text 1;Red;Blue

100,200;Text 2

CalculateScript

Pro formulář mohu nastavit script, který provádí výpočet a upravuje hodnoty zadaných polí.

Příklad:

CalculateScript=P_PROC=_PROCENTO_DPH/100

_DPH=_CASTKA*P_PROC

_CASTKA_S_DPH=_CASTKA+_DPH

_DATUM_UZP=_DATUM_VYSTAVENI

_DATUM_SPLATNOSTI=DT.Add(_DATUM_VYSTAVENI,"DD",14)

K polím jak pro čtení i pro zápis přistupujeme pomocí proměnných.

Proměnná pro přístup k poli je shodná jako jméno pole, ale na začátku má znak podtržítka.

V názvu pole smí být pouze písmena (bez diakritiky!), číslice a znak podtržítka. Pokud jsou v názvu polí jiné znaky nelze s těmito poli pracovat.

Podporované pro čtení jsou pole všech typů. Pro zápis jsou přístupná pouze pole textová pole, numerická pole a pole typu DATE.

(Je předpoklad, že v následujících verzích se počet typů přístupných i pro zápis bude rozšiřovat.)

CalculateScriptAutomatic

Určuje jestli se script provádí automaticky nebo jen na manuální pokyn

Default je 1 - automaticky

CalculateScriptAutomatic=1

FormCaption

Pokud je nastaveno, tak v titulku formuláře zobrazuje tento text.

Pokud není, tak zkusí použít FormCaption ze sekce BROWSE

FormCaption=Adresář

TISKY

V sekci EDIT je možné uvést odkaz na tiskové šablony.

Tyto šablony jsou následně nabízeny v menu formuláře.

Klíč se jmenuje PrintTemplate a jedná o klíč typu StringList.

Postup je vždy takový, že se ze šablony vygeneruje výsledný soubor ve složce temp, do něj se doplní hodnoty ze záznamu a s tímto souborem se pracuje.

Každý řádek klíče obsahuje název šablony zobrazovaný v menu. Pak následuje středník a za ním cesta a jméno souboru se šablonou. Pak může následovat středník a za ním aplikace, kterou se má výsledný soubor otevřít. Popřípadě parametry, s kterými se bude aplikace spouštět.

Pokud to, co se má se šablonou udělat obsahuje jen název šablony, tak se cílový soubor otevře aplikací, která je příslušně příponě přidružena. Například MSWord pro soubory typu docx.

Pokud řádek obsahuje aplikaci kterou se to má otevřít tak výsledný soubor otevírá touto aplikací. Pokud cesta nebo název aplikace obsahuje mezeru je nutné tento název uzavřít do dvojitých uvozovek. Pozor! Je třeba uzavřít jen cesta a název aplikace. Nikoli parametry, s kterými je volána. V parametrech bude uvedeno {0} . Za tuto sekvenci se dosadí název doplněné šablony.

Místo programu pro otevření. To je třetí parametr v řádku je možné uvést následující příkazy:

>ToPDF

Výsledný soubor převede na PDF a ten PDF otevře. Ve výstupní složce nechává soubory v obou tvarech. PDF i HTML, DOC atd.

Funguje pro soubory html, htm. Pro převod používá Google Chrome nebo MS Edge

Funguje pro soubory doc, docx, xls,xlsx, odt, ods. Pro převod používá Libre Office. Pokud není nainstalován, tak není možné tento příkaz použít.

>ToPrint

>ToPrint,"PrinterName"

Výsledný soubor vytiskne na výchozí tiskárnu.

POZOR: LibreOffice jako výchozí tiskárnu používá poslední vybranou tiskárnu. Pokud tedy pošleme tisk na určenou tiskárnu, tak následující tisk na výchozí tiskárnu je na předchozí vybranou tiskárnu.

Pokud je za čárkou uveden název tiskárny pak tiskne na zadanou tiskárnu

Funguje pro soubory doc, docx, xls,xlsx, odt, ods. Pro tisk používá Libre Office.

>ToPrint pro soubory doc a docx je možné tisknout pomocí MSWord, pokud je MSWord nainstalován jako výchozí program pro otevírání souborů těchto příloh.

```
PrintTemplate=1 Formulář Open;c:\F\Formular1.docx;  
2 Formulář Open LibreOffice;c:\F\Formular1.docx;"C:\Program Files\LibreOffice\program\soffice.exe" "{0}"  
3 Formulář Print LibreOffice;c:\F\Formular1.docx;"C:\Program Files\LibreOffice\program\soffice.exe" -p "{0}"  
4 Formulář Print Word;c:\F\Formular1.docx;"C:\Program Files\MSOffice\Word.exe" -p "{0}" /mFilePrintDefault /mFileExit /q /n  
5 Formulář HTML;c:\F\Formular1.html  
6 Formulář HTML ToPDF;c:\F\Formular1.html;>ToPDF  
7 Faktura docx TO PDF;c:\F\Formular1.docx;>ToPDF  
8 Faktura docx TO Print;c:\F\Formular1.docx;>ToPrint  
9 Faktura docx TO Print PDFCreator;c:\F\Formular1.docx;>ToPrint,PDFCreator
```

- 1.řádek – Otevře se vygenerovaný formulář prostřednictvím aplikace asociované s příponou.
- 2.řádek – Otevře se vygenerovaný formulář prostřednictvím aplikace LibreOffice.
- 3.řádek – Vytiskne se vygenerovaný formulář prostřednictvím aplikace LibreOffice.
- 4.řádek – Vytiskne se vygenerovaný formulář prostřednictvím aplikace MS Word.
- 5.řádek – Otevře se vygenerovaný formulář prostřednictvím aplikace asociované s příponou HTML.
- 6.řádek – Převede se vygenerovaný formulář na PDF a otevře se v PDF.
- 7.řádek – Převede se vygenerovaný formulář na PDF a otevře se v PDF.
Funguje pro soubory html, htm. Pro převod používá Google Chrome nebo MS Edge
Funguje pro soubory doc, docx, xls, xlsx, odt, ods. Pro převod používá Libre Office.
- 8.řádek – Vygenerovaný formulář se vytiskne na výchozí tiskárně.
Funguje pro soubory doc, docx, xls, xlsx, odt, ods. Pro převod používá Libre Office.
- 9.řádek – Vygenerovaný formulář se vytiskne na zadané tiskárně.
Funguje pro soubory doc, docx, xls, xlsx, odt, ods. Pro převod používá Libre Office.

Je třeba správně nastavit cesty k aplikacím. To kde je aplikace uložena se může v každé verzi lišit.

PrintTemplateIdentityField

Standardně se jméno vygenerovaného formuláře vytváří z obsahu UniqueField.

(JménoZdrojovéhoSouboru_JménoTable_UniqueFieldValue)

Pokud toto potřebujeme změnit, tak je možné zadat jména polí z kterých chceme generovat. Je vhodné, aby se jednalo o pole, která jsou unikátní. (Například číslo faktury)

První řetězec před středníkem je pevný řetězec, následují jména polí. (Jedná se vlastně o značku pro tisk. Takže pokud je to pole s odkazem do jiné tabulky, tak ho zadávám s tečkou.

Př.:ODBERATEL_ID.NAZEV)

Můžeme zadat i více polí odděleno středníkem.

PrintTemplateIdentity=Faktura;CISLO;ODBERATEL_ID.NAZEV;

PrintTemplateOutput

Standardně se vygenerovaný formulář vytváří v TEMP.

Pokud chceme výstup nasměrovat jinam, tak zde zadáme výstupní složku.

PrintTemplateOutput=s:\DSDAX\DSDaxDB\Output

OptionDisablePrintTemplateMask

Zakáže nabídku výběru šablony v menu na základě výběru dialogem.

Default=0

OptionDisablePrintTemplateMask=1

Nastavení řešení

Jeden z typů konfiguračního souboru je „řešení“ (Solution) . Řešení je skupina projektů.

Konfigurační soubor řešení musí obsahovat sekci [SOLUTION]

V této sekci musí být klíč name s názvem řešení.

Dále musí obsahovat klíč „project“ typu StringList (Více řádkový string), kde jsou uvedeny projekty patřící k řešení.

Příklad:

[SOLUTION]

name=Adresář

project=Adresář;s:\DSDAX\DSDaxDB\adresar_01.dsdaxdb

Výrobky;s:\DSDAX\DSDaxDB\vyrobky_01.dsdaxdb

>Nastavení:

Výrobky2;s:\DSDAX\DSDaxDB\vyrobky_02.dsdaxdb

Pokud řádek začíná znakem > tak je tento řádek považován za textový řádek. A v přehledu je zobrazen jako text místo tlačítka.

Řešení může obsahovat sekce MAIN a DB.

DisableMenuTools=1

Zakáže Menu Tools v přehledu projektů

Sekce MAIN a DB

Obsah těchto sekcí pokud není uvedeno v projektu tak se použije z řešení.

Pozor pokud otevíráme přímo projekt, tak pak nemá možnost převzít tyto hodnoty a musí být tyto údaje uvedeny v nastavení projektu.

Nastavení cest

V kterémkoli konfiguračním souboru je možné do sekce [MAIN]

uvést

DefaultPath=

Kde uvedeme, od které složky se budou odvozovat odkazy na cesty a soubory

Pokud toto nastavení není uvedeno, tak se jako DefaultPath považuje Pathname ke konfiguračnímu souboru.

DefaultPath=S:\DSDAX\DSDaxDB\

Pokud nastavujeme nějaké soubory nebo složky (PrintTemplate, PrintTemplateOutput atd.) tak můžeme použít relativní cestu k DefaultPath.

Rozšíření oproti standardnímu použití ve Windows: Je povoleno použít na začátku „:\“ pak se doplňuje pouze písmeno disku.

Je možné použít i „\.“ Například: \.\Output.txt Znamená, že od DefaultPath má jít o jednu složku výše.