

Popis rozhraní RKS Data Designer pro OLE Automation

© Radim Kunz Software

Program

procedure	Tile;	Srovná okna do dlaždic.
procedure	Cascade;	Srovná okna do kaskady.
function	LoadModelFromFile(filename:string):variant;	Načte model ze souboru.
function	NewModel:variant;	Vytvoří nový prázdný model.
procedure	Exit;	Ukončení programu.
Function	VerMajor:Integer;	Verze programu, před des. čárkou
Function	VerMinor:Integer;	Verze programu, po des. čárce
procedure	Show;	Zobrazení programu.
procedure	Hide;	Ukrytí programu

Model

function	EntCount:integer;	Počet entit v modelu
function	Ent(i:integer):variant;	Entita (0..Entcount-1). Parametrem je pořadí.
function	DatTyp(atrid:longint):string;	Funkce pro zjištění datového typu atributu. Ve vstupu je identifikátor atributu atr.oid, ve výstupu textový řetězec s typem i délkou v formátu SQL. Pracuje s normálními typy i s uživatelskými typy ze slovníku.
function	AtribCount:integer;	Počet atributů v modelu.
function	Atrib(i:integer):Variant;	Atribut. (0..AtrCount-1). Parametrem je pořadí.
function	RelCount:integer;	Počet relací v modelu
function	Rel(i:integer):Variant;	Relace (0..RelCount-1). Parametrem je pořadí.
function	DictCount:integer;	Počet položek ve slovníku
function	DictIt(i:integer):variant;	Položka slovníku (0..DictCount-1). Parametrem je pořadí.
function	TypeCount:integer;	Počet datových typů.
function	Typelt(i:integer):variant;	Datový typ (0..TypeCount-1). Parametrem je pořadí.
function	IndexCount:integer;	Počet indexů.
function	Index(i:integer):variant;	Index (0..IndexCount-1). Parametrem je pořadí.
function	IndexItemsCount:integer;	Počet položek indexů.
function	IndexItem(i:integer):Variant;	Položka indexu (0..IndexItemsCount-1). Parametrem je pořadí.
function	FindEntOID(oid:longint):variant;	Vyhledá entitu podle jejího identifikátoru.
function	FindAtrOID(oid:longint):variant;	Vyhledá atribut podle jeho identifikátoru.
function	FindRelOID(oid:longint):variant;	Vyhledá relaci podle jejího jejího identifikátoru.
function	FindDictItOID(oid:longint):variant;	Vyhledá slovníkovou položku podle jejího identifikátoru.
function	FindTypeltOID(oid:longint):variant;	Vyhledá typ podle jeho identifikátoru.
function	FindIndexOID(oid:longint):variant;	Vyhledá index podle jeho identifikátoru.
function	FindIndexItOID(oid:longint):variant;	Vyhledá položku indexu podle jejího identifikátoru.
function	ModelFileName:string;	Název souboru
procedure	SaveToFile(fname:string);	Uložení modelu do souboru.
procedure	Save;	Uložení modelu do stávajícího souboru
procedure	ModelClose;	Uzavření modelu
function	Changed:boolean;	Když byla nějaká změna ?

Entita

function	name:string;	Název entity.
-----------------	---------------------	---------------

function	OID:longint;	Identifikátor entity.
function	EntTablename:string;	Název tabulky
function	entdependent:wordbool;	Zda je entita závislá.

Atribut

function	name:string;	Název atributu
function	OID:longint;	Identifikátor atributu
function	AtrPK:WordBool;	Zda je atribut primárním klíčem.
function	AtrFK:WordBool;	Zda je atribut cizím klíčem.
function	oidEnt:longint;	Identifikátor entity, ke které atribut patří. Navazuje na Model.Ent(n)
function	oidFKRel:longint;	Pokud je FK, identifikátor relace, pomocí které atribut migroval.
	Návazuje na Model.Rel(n)	
function	oidFKAtr:longint;	Pokud je FK, identifikátor atributu, ze kterého atribut migroval.
	Navazuje na Model.Atrib(n)	
function	oidFKEnt:longint;	Pokud je FK, identifikátor entity, ze které atribut migroval. Navazuje na Model.Ent(n)
	na Model.Ent(n)	
function	oidType:longint	Identifikátor typu. Navazuje na Model.TypeIt(n).
Function	oidDict:longint	Identifikátor slovníkové položky. Navazuje na Model.DictIt(n). Pokud v této položce není 0, pak je pro atribut platný tento slovníkový typ, a hodnota v oidType je irelevantní.
function	AtrLength:longint;	Délka datového typu.
function	AtrDecimal:longint;	Počet des. míst datového typu.
function	AtrNotNull:WordBool;	Zda je atribut NOT NULL
function	AtrRoleName:string;	Rolename
function	AtrColName:string;	Název sloupce
function	AtrDefault:string;	Defaultní hodnota
function	AtrCheck:string;	Check hodnota
function	AtrDef:string;	Def1
function	AtrDef2:string;	Def2
function	AtrDef3:string;	Def3, zatím se nepoužívá
function	AtrDescription:string;	Poznámka

Relace

function	name:string;	Název relace
function	OID:longint;	Identifikátor relace
function	OidEntParent:longint;	Identifikátor parent entity. Navazuje na Model.Ent(n).
function	OidEntChild:longint;	Identifikátor child entity. Navazuje na Model.Ent(n).
function	RelType:integer;	Typ relace, 0-Identifikační, 1-Nidentifikační
function	RelDescription:string;	Poznámka
function	RelParcParent:integer;	Parcialita na straně parent, 0-Povinný, 1-Nepovinný
function	RelParcChild:integer;	Parcialita na straně child, hodnoty jako výše.
function	RelRefParentUpdate:integer;	Referenční integrita Parent Update 0-Žádná, 1-Set Null, 2-Restrict, 3-Cascade
function	RelRefParentInsert:integer;	Referenční integrita Parent Insert, hodnoty jako výše.
function	RelRefParentDelete:integer;	Referenční integrita Parent Delete, hodnoty jako výše.
function	RelRefChildUpdate:integer;	Referenční integrita Child Update, hodnoty jako výše.
function	RelRefChildInsert:integer;	Referenční integrita Child Insert, hodnoty jako výše.
function	RelRefChildDelete:integer;	Referenční integrita Child Delete, hodnoty jako výše.

Index

function	name:string;	Název indexu.
function	OID:longint;	Identifikátor indexu.
function	oidEnt:longint;	Identifikátor entity, ke které patří index. Navazuje na Model.Ent(n).

function	IndUnique:wordbool;	Zda je index Unique.
function	IndDescend:wordbool;	Zda je index sestupný

IndexIt

Položka indexu, je to vlastně spojení mezi indexem a atributem.

function	name:string;	Název položky indexu
function	OID:longint;	Identifikátor položky indexu.
function	oidInd:longint;	Identifikátor indexu, ke kterému položka patří. Navazuje na Model.Index(n).
function	oidAtr:longint;	Identifikátor atributu, ze kterého položka vznikla. Navazuje na Model.Atrib(n).
function	IndItDescend:wordbool;	Zda je položka indexu sestupná.

DictIt

Položka ze slovníku uživatelských typu.

function	name:string;	Název slovníkové položky.
function	OID:Longint;	Identifikátor slovníkové položky.
function	OidType:longint;	Identifikátor datového typu slovníkové položky. Navazuje na Model.TypeIt(n).
function	DictLength:longint;	Délka datového typu.
function	DictDecimal:longint;	Počet des. míst datového typu.
function	DictDefault:string;	Defaultní hodnota.
function	DictCheck:string;	Check hotnota.
function	DictDescription:string;	Poznámka.

TypIt

Typy, které aktuální databáze podporuje.

function	name:string;	Název datového typu
function	OID:Longint;	Identifikátor datového typu.
function	NamePh:String;	Fyzický název typu. Většinou je stejný jako name , ale ne vždycky.
function	LengthExist:WordBool;	Zda datový typ vyžaduje zadání délky.
function	DecimalExist:WordBool;	Zda datový typ vyžaduje zadání počtu des. míst.
function	LengthMax:longint;	Maximální délka datového typu.
function	LengthMin:longint;	Minimální délka datového typu.
function	LengthDefault:longint;	Přednastavená délka datového typu.
function	DecimalMax:longint;	Maximální počet des. míst.
function	DecimalMin:longint;	Minimální počet des. míst.
function	DecimalDefault:longint;	Přednastavený počet des. míst.

Program RKScase se musí zaregistrovat jako OLE Automation Server. Registruje se spuštěním programu s parametrem **rkscase.exe /regserver**. Pokud se bude RKS Data Designer odinstalovávat, bylo by rozumné před odinstalací program odregistrovat příkazem **rkscase /unregserver**

V Delphi 2,3

Program se jako OLE Server spouští:

```
Var x:Variant;
```

```
x := CreateOleObject('rkscase.command');
```

Pokud u kterékoliv proměnné typu Variant víte, že
1 použijete ji pro jinou proměnnou
2 v programu opustíte oblast rozsahu platnosti proměnné
pak pro jistotu proveďte `promenna := unassigned`

Tenhle příklad je napsaný pro Delphi 2, ale obdobně by měl fungovat i ve VB, dokonce to pracuje i s VBA v MSWord97, prostě všude, kde jde naprogramovat řadič OLE Automation.

```
uses OleAuto;
```

```
var   prog:variant;      (* Program CASE *)
      model:variant;    (* Model *)
      i:integer;
      atr:variant;      (* Atribut *)
      ent:variant;      (* Entita *)
begin
prog := CreateOleObject('rkscase.command');
(* spuštění programu jako OLE Automation Serveru *)

model := prog.LoadModelFromFile('c:\model.dmr');
(* načtení modelu *)
(* Soubor musí existovat, to si musíte ošetřit sami *)
(* Pokud neexistuje, v proměnné model je hodnota Unassigned *)

prog.show;
(* zobrazení programu *)

for i := 0 to model.EntCount-1 do
  begin
  writeln(model.ent(i).name);
  (* Vypis všech atributů v modelu *)
  end;
for i := 0 to model.AtribCount-1 do
  begin
  atr := model.Atrib(i);
  (* Prochází všechny atributy po řadě *)

  ent := model.FindEntOid(atr.oidEnt);
  (* K atributu najde příslušnou entitu *)

  writeln(format('%s %s', [atr.name, ent.name]));
  (* A vypise *)
  end;

atr := UnAssigned;
ent := UnAssigned;
(* Radeji udelejte na konci tohle, je to bezpecnejsi *)
(* Tyka se asi jenom Delphi *)

model.ModelClose;
(* Uzavření modelu *)

model := UnAssigned;
(* Totéž jako u atr a ent *)

prog.Exit;
```

(* Ukončení OLE Automation Serveru *)

end;