

Komunikační knihovny v prostředí Delphi

ZÁKLADNÍ RADY PŘI PŘECHODU NA
KOMUNIKAČNÍ KNIHOVNY POD DELPHI

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 22.05.2003

Datum posledního uložení dokumentu: 22.05.2003

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.O dokumentu	5
1.1. Revize dokumentu	5
1.2. Účel dokumentu	5
1.3. Rozsah platnosti	5
1.4. Související dokumenty	5
2.Termíny a definice	5
3.Úvod	6
4.Dosud převedené knihovny	6
5.Některé změny při volání metod komunikačních tříd	7
6.Nové metody třídy tChnVirt	8
6.1. ChNumNameClass	8
6.2. ChNameClass	8
6.3. ChGetLastChn	8
6.4. ChGetLastClass	8
7.ChGetNumChn	8
7.1. ChResultStr	8
8.Nové metody třídy tChnCollection	8
8.1. ChNumNameClass	8
8.2. ChNameClass	9

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00		Wi		První vydání.
1.10		Tu	22.05.2003	Úprava dokumentu dle ISO9000.

1.2. Účel dokumentu

Tento dokument popisuje princip použití nových komunikačních knihoven pro jazyk Delphi.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba orientovat se v používání programového vybavení SofCon.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

Firma SofCon s.r.o. převedla některé komunikační knihovny z jazyka Borland Pascal do jazyka Delphi pro jejich snazší použití při vytváření monitorovacích a testovacích programů na úrovni Win32.

Převedení se hlavně týkalo těchto oblastí:

- Jiné používání hardware sériových portů prostřednictvím služeb Windows.
- Převedení objektových typů „object“ na typ třída „class“.
- Používání dlouhých řetězců Delphi.
- Volání standardních funkcí Delphi pro práci s řetězci.
- Přidělení některých šikovných funkcí pro manipulaci s třídními typy „class“.

V nových aplikacích a knihovnách by se **neměly používat** jednotky **Objects** a **NumToStr**.

Typ TObject, který byl definován v jednotce Objects, je již v Delphi definován automaticky s trochu jinými (rozšířenými) vlastnostmi.

Většina funkcí v NumToStr je již implicitně definována v Delphi v jednotce SysUtils.

4. Dosud převedené knihovny

- **UString** - Upraveno dekodování textových parametrů v dlouhých řetězcích Delphi.
- **ChnTypes** - Drobné úpravy konstant.
- **ChnVirt** - Přiděleny funkce *ChNumNameClass*, *ChNameClass*, *ChGetLastChn*, *ChGetLastClass*, *ChGetNumChn*, *ChResultStr* do třídy *tChnVirt*.
- **ChnCom** - Předělána obsluha fyzického sériového portu prostřednictvím služeb Windows.
- **ChnMBox** - Předělána simulace sériové komunikace přes schránky o.s.Retos na simulaci komunikaci přes vnější pomocné buffery (definované na úrovni aplikace). V budoucnu by se tento způsob mohl upravit na komunikaci prostřednictvím zpráv Windows pro možnost simulované komunikace i mezi různými aplikacemi Windows.
- **ChnPrt** - Kromě předělání původního objektového typu *tChnPrt* na třídu *tChnPrt* žádné změny.
- **Xor8** - Kromě předělání původního objektového typu *tXor8* na třídu *tXor8* žádné změny.
- **ChnLecom** - Kromě předělání původního objektového typu *tChnLecom* na třídu *tChnLecom* žádné velké změny.
- **Crc16** - Kromě předělání původního objektového typu *tCrc16* na třídu *tCrc16* žádné změny.
- **Chn** - Kromě předělání původního objektového typu *tChn* na třídu *tChn* žádné změny.
- **Timer** - Objektový typ *tTimer* předělán na třídu *tTimer* a jednotka upravena pro Windows. Jednotka používá standardní komponentu třídy *tTimer* z Delphi.

5. Některé změny při volání metod komunikačních tříd

Inicializace a rušení komunikačního objektu:

V Pascalu se komunikační objekt např. **MyChn** typu **pChnVirt** inicializoval a rušil pro komunikaci přes **ChnCom** následovně:

```
MyChn := New(pChnCom, Init); {vytvoření instance na
objekt}
....
....
Dispose(MyChn, Done); {zrušení instance objektu}
MyChn := nil;
```

V Delphi ale žádný typ **pChnVirt** ani **pChnCom** není, jelikož všechny třídy jsou automaticky dynamické. Takže třída **tChnVirt** v Delphi odpovídá typu **pChnVirt** (ukazatel na objekt) v Pascalu. Proto již nelze vytvořit statický objekt např. typu **tCmd** nebo **tTimer** jako v Pascalu, jelikož v Delphi je tato třída dynamická, tzn. musí se také dynamicky alokovat a rušit. V Delphi dále pracujeme s tímto objektem jako s ukazatelem, ale bez kvalifikátoru „^“.

V Delphi se inicializace a rušení komunikačních objektů (a vlastně všech objektů typu třída) provádí následovně:

```
MyChn := pChnCom.Init; {vytvoření instance na objekt}
....
....
MyChn.Free; {zrušení instance objektu}
MyChn := nil;
```

Pro zrušení instance objektu lze volat kompatibilně i metodu **Done** („**MyChn.Done**“), která metodu **Free** sama volá.

Navazování a rušení komunikačního spojení metodami **ChOpen**, **ChConnect**, **ChDisconnect** a **ChClose** zůstává beze změny.

Stejně tak beze změny zůstává vysílání a příjem zpráv pomocí metod **ChSend**, **ChReceive** a **ChReceiveChar**.

Pozor ale při cyklech dotazujících se na konec provedené akce např.:

```
ChConnect
repeat
  Sts:=ChResult;
until (Sts<>res_Ok) or (ChState=CHS_Connect);
```

Pokud tu program na nějakou dobu zatuhne, zamrzne po tu dobu celá aplikace. Proto je vhodnější pro všechny tyto akce napsat čistě průchozí automat.

6. Nové metody třídy tChnVirt

6.1. ChNumNameClass

```
class function ChNumNameClass(NumName: tChNumName) : TClass; virtual;  
    Vrátí typ třídy komunikačního objektu s číselným jménem NumName.
```

6.2. ChNameClass

```
class function ChNameClass(Name: tChName) : TClass; virtual;  
    Vrátí typ třídy komunikačního objektu se jménem Name.
```

6.3. ChGetLastChn

```
function ChGetLastChn : tChnVirt; virtual;  
    Vrátí ukazatel na poslední komunikační objekt v komunikačním kanálu, který  
    vznikl zřetězením komunikačních objektů.
```

6.4. ChGetLastClass

```
function ChGetLastClass : TClassChnVirt; virtual;  
    Vrátí typ třídy posledního komunikačního objektu v komunikačním kanálu, který  
    vznikl zřetězením komunikačních objektů.
```

7. ChGetNumChn

```
function ChGetNumChn : byte; virtual;  
    Vrátí počet zřetězených komunikačních objektů v komunikačním kanálu. Pokud  
    je kanál tvořen pouze jedním objektem (jedna vrstva) vrátí hodnotu  
    1.
```

7.1. ChResultStr

```
function ChResultStr (Sts : tChResult) : tResultStr; virtual;  
    Vrátí řetězec s popisem případné chyby v parametru Sts.  
    Pokud Sts neobsahuje žádnou chybu, vrátí řetězec 'Ok'. Pokud Sts obsahuje  
    neznámou chybu pro danou vrstvu, vrátí prázdný řetězec.
```

8. Nové metody třídy tChnCollection

8.1. ChNumNameClass

```
function ChNumNameClass(NumName: tChNumName) : TClass;  
    Nalezne v ChnCollection první prvek s daným číselným jménem NumName a  
    vrátí jeho typ třídy. Pokud prvek nenalezne, vrátí třídu TObject.
```


8.2. ChNameClass

```
function ChNameClass(Name: tChName) : TClass;
```

Nalezne v ChnCollection první prvek s daným jménem Name a vrátí jeho typ třídy. Pokud prvek nenalezne, vrátí třídu TObject.