

*Monitorovací a vizualizační systém ProCop 3.4*

---

# Tipy a triky

## Vývojářská příručka

---



© *ALFA Mikrosystémy, s.r.o.*  
*Ostrava 2012*



# Tipy a triky

## Vývojářská příručka

---

*ALFA Mikrosystémy, s.r.o.*

*Monitorovací a vizualizační systém ProCop 3.4 je specializovaný software pro monitorování technologických procesů, uživatelské řízení technologií, archivaci historických trendů technologických veličin a alarmních stavů.*

*Zahrnuje zkušenosti z předchozích verzí systému, které mají počátky již v roce 1993, kdy byl poprvé nasazen první předchůdce stávajícího monitorovacího systému. Dlouholetou praxí v oboru monitorování technologií, zejména pak v tepelném hospodářství, se podařilo dle zkušeností a požadavků zákazníků a samotných dispečerů vyvinout produkt, který se Vám v tuto chvíli dostává do rukou.*

---

**Copyright © 2012 ALFA Mikrosystémy, s.r.o. Ostrava**

Microsoft, Windows, Windows 7, Windows Vista, Windows XP, Windows 2000 a Internet Explorer jsou registrované obchodní známky Microsoft Corporation, Intel je registrovaná obchodní známka, Pentium je obchodní známka Intel Corporation.  
ProCop je registrovaná obchodní známka firmy ALFA Mikrosystémy, s.r.o.

Vytištěno: listopad 2012

# Obsah

<b>1</b>	<b>Moxa NPort v UDP režimu</b>	<b>7</b>
1.1	Virtuální port vs. UDP režim.....	7
1.2	Nastavení UDP režimu.....	8
	Připojení k převodníku .....	8
	Konfigurace IP adresy .....	9
	Konfigurace UDP režimu .....	11
	Nastavení sériového portu .....	12
	Uložení konfigurace .....	13
1.3	Úpravy projektu pro N-Port.....	13
1.4	Problémy s komunikací.....	15
<b>2</b>	<b>Uživatelský formát kanálů</b>	<b>17</b>
2.1	Nastavovací dialogy.....	18
<b>3</b>	<b>Automatické vlastnosti potrubí</b>	<b>19</b>
<b>4</b>	<b>Ukládání stavových proměnných</b>	<b>21</b>
<b>5</b>	<b>Atributy knihovních bloků</b>	<b>23</b>
5.1	Výroba bloku s atributy.....	23
5.2	Použití bloku s atributy.....	25
<b>6</b>	<b>Parametry modulu PX</b>	<b>29</b>
6.1	Parametry regulátoru.....	31
6.2	Speciální funkce.....	31
6.3	Podrobná nastavení.....	33
<b>7</b>	<b>Export do SQL databáze</b>	<b>37</b>
7.1	Konzola správy ProSQL.....	39
7.2	Definice tabulek.....	40
	Definice struktury tabulky .....	41
	Typy hodnot pro export .....	42
	Datové typy sloupců v SQL .....	43
7.3	Report tabulka.....	43
7.4	Příklad exportu měřičů spotřeb.....	45
<b>8</b>	<b>Provoz v doméne bez lokálního uživ.</b>	<b>49</b>
8.1	Úpravy serveru pro provoz bez uživatele ProCop .....	49
8.2	Úpravy klienta pro provoz bez uživatele ProCop .....	53
	<b>Index</b>	<b>55</b>



# 1 Moxa NPort v UDP režimu

Tato kapitola se zabývá nastavením převodníků MOXA DE-211 a DE-311 v režimu bez ovladače virtuálního sériového portu ve Windows. Převodníky Ethernet / RS232 (RS485,RS422) umožňují využít počítačové sítě k připojení zařízení standardně připojované sériovou linkou přímo k PC s monitorovacím systémem ProCop.

Pomocí převodníků je možné nahradit specializovanou galvanicky oddělenou kabeláž pro RS232, RS422, RS485 standardní počítačovou sítí. Může být využita i stávající síťová infrastruktura po domluvě se správcí dané sítě, kteří musí přidělit příslušný adresní prostor pro PC a převodníky.

## Důležité upozornění!

*Nesmíme zapomenout, že narozdíl od sériové linky bude mít přenos dat přes počítačovou síť určité zpoždění, může docházet ke ztrátám části komunikace v závislosti na zatížení počítačové sítě, kompletním výpadkům sítě a v rozsáhlejších sítích i k záměně pořadí doručovaných dat. Proto NENÍ vhodné používat tento způsob propojení přes rozsáhlé sítě, či dokonce Internet u kritických průmyslových regulací. Nejlépe je volit vlastní nezávislý segment sítě.*

## 1.1 Virtuální port vs. UDP režim

Většina převodníků Ethernet/RS232 umožňuje vyrobít virtuální sériový port přímo v operačním systému počítače.

Aplikace, které umožňují využít pouze standardní sériové linky, mohou snad komunikovat i se vzdálenými zařízeními, aniž by to poznaly. Rozdílem může být pouze delší odezva a větší chybovost linky.

### Problémy s virtuálními porty

Obrovským problémem při používání ovladačů virtuální sériových portů je jejich nespolehlivost při výpadcích či přetížení sítě. Při startu počítače se zavádějí ovladače virtuálních sériových portů, které navazují komunikaci s jednotlivými převodníky. Pokud se daný převodník nehlásí, virtuální port nenaváže TCP/IP spojení a sériový port je zcela nefunkční obvykle až do dalšího restartu počítače. Obdobná situace nastane, dojde-li k delšímu výpadku spojení i při provozu a komunikaci.

#### **Důvody nenavázání, nebo rozvázání komunikace s převodníky a nefunkčnosti komunikace:**

- výpadek komunikace sítě mezi PC a převodníkem
- přetížení některé části sítě mezi PC a převodníkem
- výpadek napájení některého z uzlů sítě
- výpadek napájení převodníku

Uvedené důvody i když bývají jen dočasné, přesto mohou vést k trvalému nenavázání, nebo rozvázání komunikace mezi ovladačem virtuálního sériového portu a převodníku. Opětovné navázání komunikace i když důvod výpadku již pominul, se provede až při dalším restartu PC a zavedením ovladačů.

### UDP Server/Client mode

Pokud je převodník nastaven v režimu **UDP Server/Client**, data jsou přenášena samostatnými UDP pakety, které jsou odesílány přímo monitorovacím systémem ProCop bez návaznosti na ovladač virtuálního sériového portu. Pokud nastane výpadek sítě, napájení a podobně, data pochopitelně nejsou doručena. Po obnovení komunikace na úrovni sítě, jsou UDP pakety opět znovu doručovány. Počty pokusů na doručení paketů a doba čekání na odpověď je nastavitelná přímo v monitorovacím systému ProCop.

#### **Výhody užívání režimu UDP Server/Client:**

- zotavení komunikace monitorovacího systému po zotavení sítě bez restartu PC
- nastavitelný počet pokusů o doručení paketů
- nastavitelný čas čekání na odpověď na daný paket
- spolehlivější komunikace a menší zatížení PC při větším množství převodníků

**⚠ Důležité upozornění!**

*Pokud používáme režim UDP Server/Client, virtuální port pro daný převodník nesmí být na PC definován! Byl-li nakonfigurován již dříve, je vhodné tento virtuální port pomocí nástrojů konfigurace virtuálních portů odstranit.*

## 1.2 Nastavení UDP režimu

Než přistoupíme k samotnému nastavení převodníků MOXA NPort DE-311 (DE-211), budeme potřebovat statické (pevné) IP adresy jednotlivých zařízení.

**Potřebné adresy:**

- **Dispečerské PC:** IP adresa, maska sítě, výchozí brána
- **Každý MOXA NPort:** IP adresa, maska sítě, výchozí brána
- **Komunikační UDP port (např. UDP: 4001)**

**Poznámka**

*Výchozí brány jsou potřeba jen v případě, že se jednotlivé zařízení nacházejí v různých IP sítích. Všechny IP adresy musí být unikátní, pevné, vzájemně na síti dosažitelné. Síť musí být propustná oběma směry z PC na všechny převodníky a z každého převodníku směrem na PC pro UDP pakety na zvoleném portu, implicitně 4001. Port ale může být zvolen libovolně, jak je potřeba.*

**Důležité upozornění!**

*Pokud je na PC používán firewall, je nutné do něj nakonfigurovat výjimku pro příchozí (případně i odcházející) UDP pakety na zvoleném portu (4001). K nastavování brány firewall a dalších konfigurací systému budeme potřebovat příslušná oprávnění systému Windows.*

### 1.2.1 Připojení k převodníku

#### První připojení sériovým kabelem na místě

Při prvním ožívání převodníku MOXA NPort po vybalení a zapojení budeme potřebovat PC se sériovým portem a přímý kabel RS232 s příslušnými redukcemi velikosti konektorů.

**Přepínače na převodníku DE-311 (DE-211) přepneme takto:**

- SW1 - **ON**
- SW2 - **OFF**
- SW3 - **OFF**

Připojíme přímo kabelem k sériovému portu PC. Na PC spustíme libovolný software, který umí emulovat terminály ANSI/VT100 na sériovém portu. Příkladem může být Hyperterminál, dodávaný v příslušenství pro komunikace přímo s Windows, nebo PComm Lite Terminal emulator, dodávaný přímo na CD k převodníkům MOXA NPort (bude jej potřeba nainstalovat). V nastavení emulátoru terminálu zvolíme emulaci terminálu ANSI, nebo VT100.

Otevřeme příslušný sériový port PC, nastavíme rychlost 19200,n,8,1 (19200bps, none = bez parity, 8 bitů v rámci, 1 stop bit) a NPort připojíme k napájení. Po krátké chvíli, by se měl objevit výpis volby režimu terminálu. Pokud se neobjeví, můžeme zkusit párkrát stisknout klávesu **ENTER**. V případě neúspěchu zkontrolujeme kabeláž, přepnutí přepínačů, odpojíme a znovu připojíme napájení převodníku, případně nastavení parametrů sériového portu PC.

**★ Tip!**

*Pokud je již převodník nakonfigurován jako virtuální sériový port a komunikuje, nemusíme k němu chodit, ale je možné jej překonfigurovat vzdáleně přes počítačovou síť.*

#### Vzdálené připojení přes síť, byl-li převodník již dříve konfigurován

Pokud jsme již měli dříve převodník nakonfigurován v síti jako virtuální sériový port, nemusíme k němu chodit s kabelem. Otevřeme správu virtuálních sériových portů, zjistíme poznačíme si jeho IP adresu, síťovou masku, případně výchozí bránu a virtuální sériový port odstraníme.

**Upozornění!**

*Běží-li na PC monitorovací systém, bude nutné přejít do režimu designer, vybrat RS232 cestu příslušného sériového portu a v lokální nabídce, kterou lze otevřít pravým tlačítkem myši zvolit "Zastavit podřizené".*



Nyní můžeme přejít k připojení k vybranému převodníku. Spustíme konzolu příkazového řádku. V nabídce **Start**, v řádku **Spustit** napíšeme "**cmd**" a stiskneme **ENTER**, nebo zvolíme **Start\Všechny programy\Příslušenství\Příkazový řádek**. Otevře se okno s obvykle černým pozadím.

Napíšeme příkaz:

```
telnet 192.168.1.240
```

kde uvedeme správnou IP adresu převodníku získanou z konfigurace virtuálních sériových portů a stiskneme **ENTER**.

*Příkazový řádek a spuštění emulátoru terminálu telnet po síti na IP adresu převodníku*



Pokud je převodník dostupný pro vzdálenou správu, měl by se objevit dotaz na typ terminálu, obdobně jako při přímém připojení přes sériový kabel.

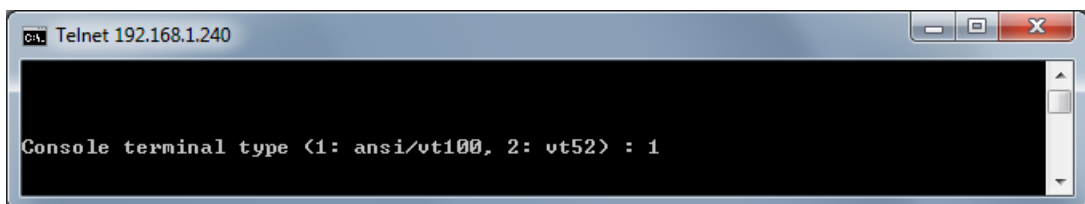
### **Upozornění!**

*Pokud je SW1 přepnut do polohy ON, nelze se připojit terminálem telnet přes počítačovou síť, jen přes sériovou linku přímo u zařízení. Toto platí i opačně, pokud je SW1 v poloze OFF, nelze se připojit přes sériovou linku, pouze vzdáleně přes počítačovou síť.*

## Volba terminálu po připojení (RS232 i telnet)

Po připojení emulátorem terminálu na sériové lince RS232 (hyperterminal, PComm Lite), nebo přes počítačovou síť (telnet) se zobrazí výpis volby terminálu.

*Volba režimu terminálu v programu telnet*



Ponecháme volbu na **1** a stiskneme **ENTER**.

### **Upozornění!**

*Bylo-li dříve do převodníku zadáno heslo pro přístup na konfiguraci, bude nyní vyžadováno. Zadáme jej a stiskneme **ENTER**.*

## 1.2.2 Konfigurace IP adresy

Nejprve poznamenejme, že další kapitoly jsou významově téměř shodné pro konfiguraci přes sériovou linku i síťový terminál telnet. Mohou se lišit zejména v designu zobrazovaných oken dle zvoleného terminálu, nikoliv však významově. Případné významové odlišnosti budou zvláště poznamenány.

### Hlavní nabídka a ovládání

Po připojení **87**, volbě terminálu a případném zadání hesla by se měla zobrazit hlavní nabídka konfigurace převodníku NPort. Po nabídce se pohybujeme šipkami **Vlevo** a **Vpravo**, volbu provedeme klávesou **ENTER**, návrat klávesou **ESC**. Pro přechod mezi zadávacími kolonkami je možné použít klávesu **TAB**, šipky **Nahoru** a **Dolů**, v kolonce pak **Vlevo** a **Vpravo**. Výběr ze seznamu, nebo otevření další podnabídky se provádí klávesou **ENTER**. Úpravy hodnot se provádějí přepisem, klávesy **BackSpace** a **Delete** provádějí vymazání předchozího znaku. Znak je možné přepsat klávesou **Space** (mezerou) na prázdný. Po hlavní nabídce se můžeme pohybovat i volbou zvětšeného písma.

### ⚠ Důležité upozornění!

Veškeré úpravy v konfiguraci se projeví až po uložení a restartu zařízení. Pro zapsání konfigurace zvolíme v hlavní nabídce **Restart** a potvrdíme uložení konfigurace klávesou **ENTER**.

### ★ Tip!

Pokud **nechceme** změny v konfiguraci uložit, zvolíme z hlavní nabídky volbu **Exit**. Případný dotaz na uložení hodnot demontujeme klávesou **ESC**.

## Konfigurace IP Adresy

V hlavní nabídce zvolíme volbu **serverConfig**.

### Konfigurace IP adresy převodníku

```

Telnet 192.168.1.240
MOKA NPort Express U2.6
[serverConfig] OP_mode Serialport Monitor Ping Restart Exit
Config server settings
ESC: back to menu Enter: select

Server Model          DE-311 Rev.2
Server Name           [Imeno naseho prevodniu ]
Serial Number         57288

IP Configuration     [Static IP ]
IP Address            [192.168.1.240 ]
Netmask               [255.255.255.0 ]
Gateway               [192.168.1.1 ]
Ethernet Status       100M/Link
MAC Address           00:90:E8:04:DF:C8

Password              [ ]
  
```

Převodník si můžeme pojmenovat, abychom ho kdykoliv identifikovali. Jméno je viditelné jen v konfiguraci a může být libovolné.

### ✍ Poznámka

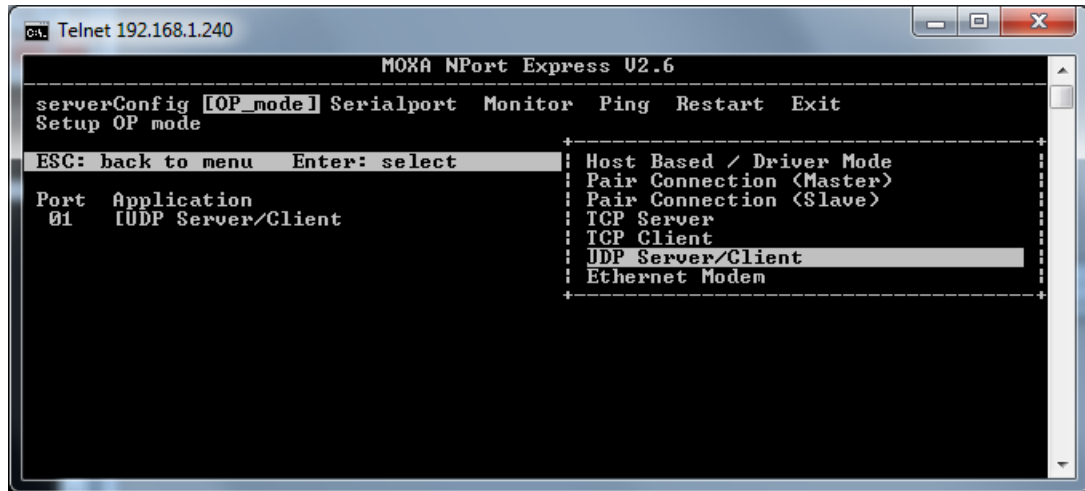
Pokud jsme připojeni přes síť emulátorem terminálu telnet, pravděpodobně již máme správně nastavenou IP adresu. Pokud ji však změníme, promítne se tato změna až při ukončení konfigurace. Nové připojení již budeme muset provést na novou IP adresu. Pokud byla adresa zadána chybně, nebo koliduje s jiným zařízením, budeme muset k převodníku dojet a nastavit správně přes sériovou linku **8**.

Nastavíme IP adresu. Zvolíme z nabídky **Static IP** a zadáme **IP adresu**, masku **Netmask**, případně výchozí bránu **Gateway**. Pro zvýšení bezpečnosti můžeme zadat heslo **Password** pro přístup ke konfiguraci převodníku.

Pokud je vše správně nastaveno, můžeme z nabídky odejít klávesou **ESC**.

### 1.2.3 Konfigurace UDP režimu

V hlavní nabídce zvolíme volbu `OP_mode`. V levé spodní části `Application` vyberme z nabídky `UDP Server/Client`.  
*Výběr režimu převodníku*



Dále bude třeba nastavit parametry přenosu UDP paketů mezi PC dispečinku a převodníkem NPort. V pravé části zvolíme `More setting`, `Select for more setting`.

*Přístup k nastavení parametrů UDP režimu převodníku*



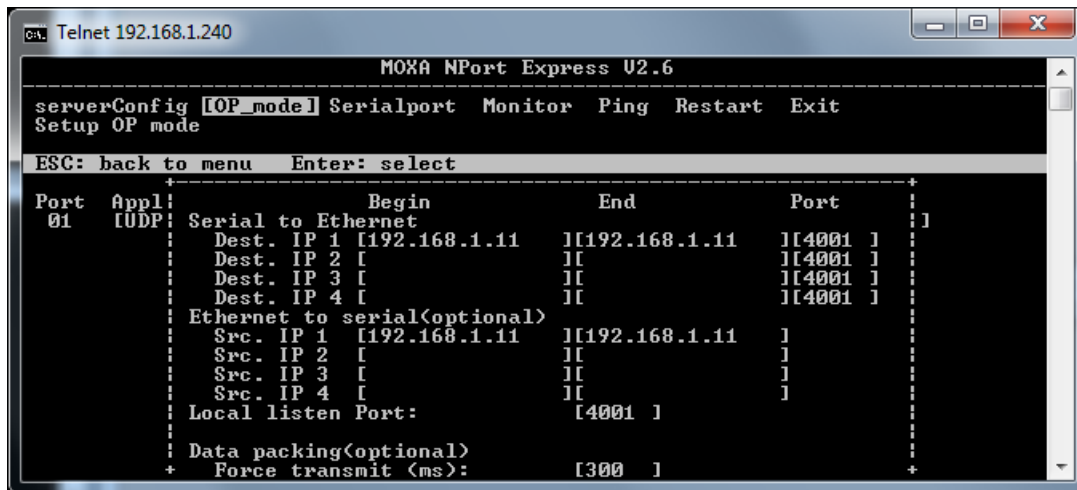
Následující tabulka zobrazuje parametry, které je potřeba nastavit pro PC dispečinku s IP adresou 192.168.1.11. Pokud má PC adresu jinou, zadáme ji místo 192.168.1.11 na všechna čtyři místa. Dále je zde nastaven komunikační UDP port 4001 pro oba směry komunikace. Pokud je potřeba komunikovat po jiném portu, přepíšeme jej ve všech sloupcích.

#### ★ Tip!

*Pokud chceme jednotlivá zařízení rozčlenit podle převodníků, každému z nich nastavíme jiný UDP port, např. 4001, 4002, 4003... a to v obou směrech. To znamená, že při konfiguraci jednoho převodníku použijeme ve všech případech i stejný port, v prvním všude 4001, v druhém všude 4002. Každý port pak bude znamenat vložení nové cesty NPortWay v projektu Monitorovacího systému ProCop.*

Dále nastavíme maximální čas pro odeslání paketu `Force transmit (ms)`. Tento čas udává, po jaké době bude automaticky paket s daty ze sériové linky odeslán. Tento parametr bývá nastaven na 0. Pro různá zařízení je potřeba vyzkoušet, jaký čas bude vhodný. Například pro MBus je vhodné zvolit čas okolo 300ms.

Adresy pro přenos dat v UDP režimu, UDP porty a odesílací čas



Pro vysvětlení uvedme, že se část **Serial to Ethernet** konfiguruje, na jaké až 4 rozsahy IP adres mají být zaslány UDP pakety s definovaným Portem. V našem případě je definován jediný počítač - rozsah právě jedné adresy dispečinku. Část **Ethernet to serial (optional)** definuje, ze kterých počítačů se data přichozí na **Local listen Port** přepošlou na sériovou linku. V našem případě se jedná opět o jediný a to dispečerský počítač. UDP Port nastavíme později v NPort cestě monitorovacího systému ProCop.

Pokud je vše správně nastaveno, můžeme z nabídky odejít klávesou **ESC**.

## 1.2.4 Nastavení sériového portu

Narozdíl od režimu virtuálního portu, kde se komunikační parametry sériové linky zadávaly v příslušné cestě Monitorovacího systému ProCop a přenášely při komunikaci, nyní musíme zadat komunikační parametry přímo do převodníku.

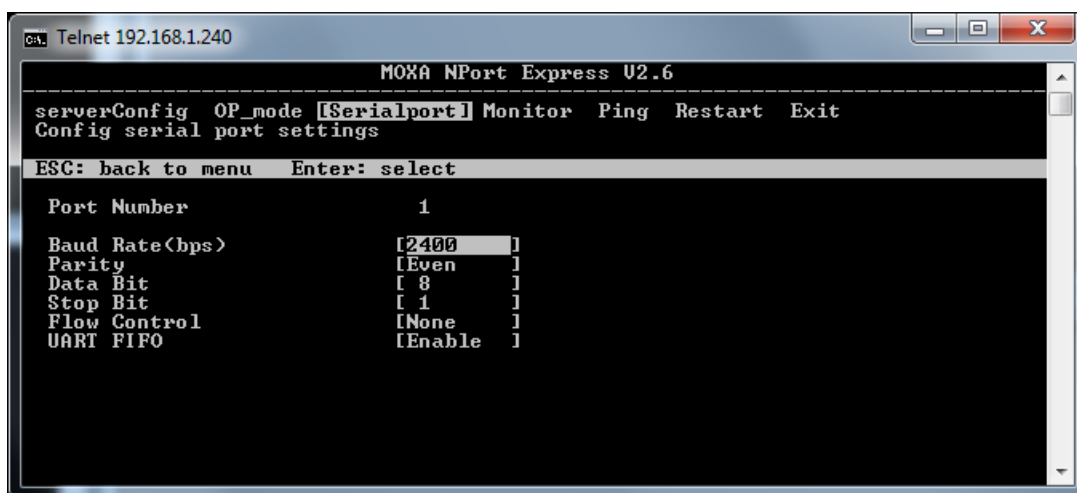
### ★ Tip!

*Pokud předěláváme stávající virtuální port na UDP režim, je vhodné se podívat do Panelu nástrojů ProCop v režimu Designer a ve správci projektu najít příslušnou nahrazovanou cestu RS232, původně připojenou k virtuálnímu portu a komunikační parametry převzít přímo z ní. Parametry nastavované v konfiguratoru virtuálních portů se pro komunikaci Monitorovacím systémem ProCop neuplatňují.*

## Nastavení komunikačních parametrů sériové linky

V hlavní nabídce zvolíme volbu **Serialport**. V kolonce Baud Rate(bps) nastavíme komunikační rychlost, v dalších kolonkách pak paritu, délku rámce, počet stop bitů, řízení toku a používání vyrovnávací paměti na sériovém portu.

*Nastavení komunikačních parametrů pro M-Bus*



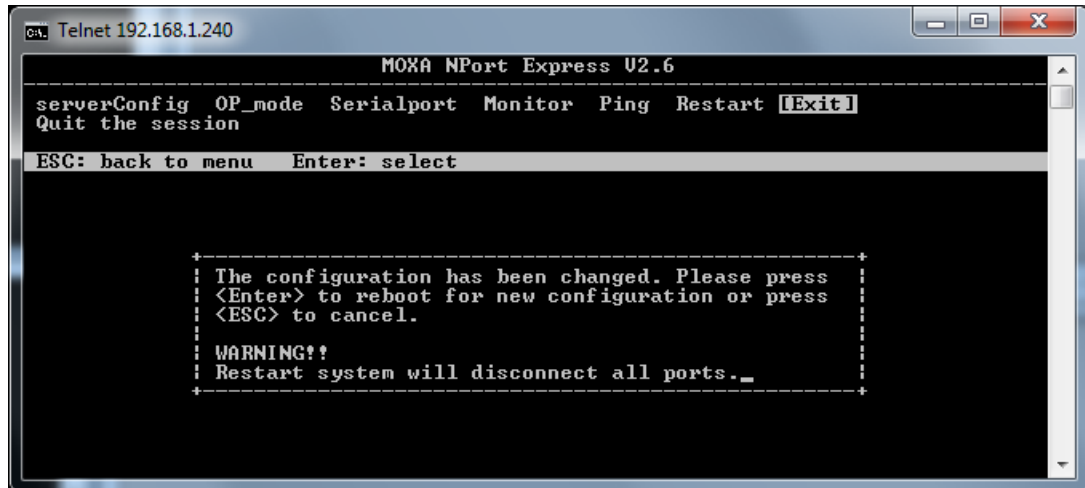
V obrázku jsou nastaveny komunikační parametry obvyklé pro převodník RS232/M-Bus, 2400,e,8,1.

Pokud je vše správně nastaveno, můžeme z nabídky odejít klávesou **ESC**.

## 1.2.5 Uložení konfigurace

Pokud jsme vše nastavili správně, můžeme v hlavní nabídce zvolit volbu **Exit**. Je zobrazena otázka, zda chceme data uložit do EEPROM a převodník restartovat. Pokud jsme si nastavením jisti, stiskneme klávesu **ENTER** pro potvrzení. Pokud chceme všechny změny stornovat, stiskneme klávesu **ESC**.

*Uložení změn nastavení a restart převodníku*



### ▲ Důležité upozornění!

Nastavení ukládáme jen tehdy, když jsme si jisti, že je nastavení IP adresy zařízení správné, nebo jsme je nezměnili. Ostatní nastavení jako **OP\_mode** či **Serialport** lze kdykoliv vzdáleně změnit, ne však, pokud se do zařízení již vzdáleně nedostaneme. V takovém případě můžeme kdykoliv zařízení nakonfigurovat přímo přes port RS232 na místě. Pokud jsme nastavili heslo, bude nadále vyžadováno pro konfiguraci! Raději si jej poznamenejme.

## 1.3 Úpravy projektu pro N-Port

Nyní bude potřeba upravit monitorovací projekt. Narozdíl od virtuálních sériových portů není potřeba mít zvlášť jednotlivé cesty. Všechny převodníky, které mají nastaven shodný UDP port (v našem příkladu 4001) stačí jediná komunikační cesta pro všechna zařízení.

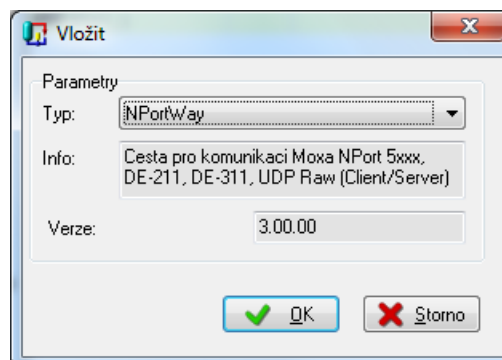
### ★ Tip!

Pokud chceme jednotlivá zařízení rozčlenit podle převodníků, každému z nich nastavíme jiný UDP port (např. 4001, 4002, 4003... Pak bude pro každý původní virtuální sériový port - pro jeden převodník - právě jedna komunikační cesta NPortWay.

### Vložení a nastavení komunikační cesty NPortWay

V Panelu nástrojů ProCop otevřeme v režimu Designer Správu projektu, vybereme Cesty a moduly a stiskneme klávesu **Insert**, nebo symbol zeleného plus. Vybereme NPortWay. Cestu pojmenujeme.

*Vložení komunikační cesty NPortWay*

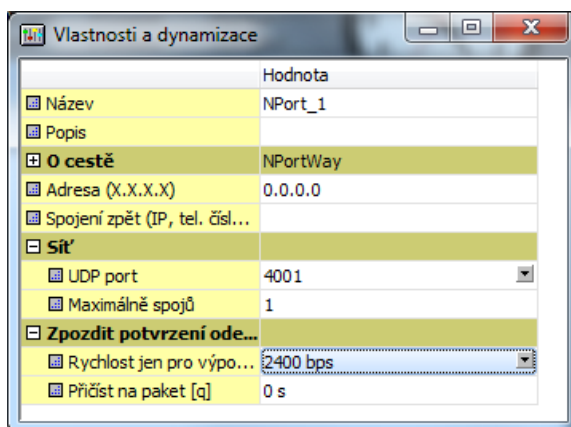


Ve vlastnostech a dynamizacích cesty je adresa PC (0.0.0.0), tu můžeme ponechat, sama se při spuštění načte. V části

**Síť**, v položce **UDP port** nastavíme příslušný port, který jsme nastavili do převodníku. V našem případě 4001. Položka **Maximálně spojů** určuje, kolik zařízení může společně komunikovat současně. Pro zařízení MBus necháme počet na 1.

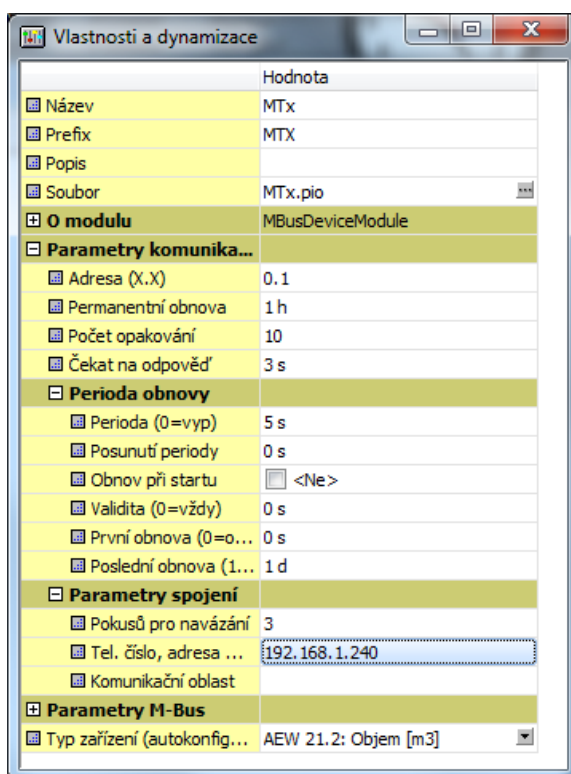
V části **Zpozdít potvrzení odeslání dle rychlosti**, v položce **Rychlost jen pro výpočet** zpoždění uvedeme rychlost komunikace na portu RS232. V našem případě 2400bps. Tato rychlost slouží jen pro výpočet předpokládaného komunikačního zpoždění. Musí být správně nakonfigurována i u sériového portu v převodníku [12].

*Parametry komunikační cesty NPortWay pro MBus.*



Nyní můžeme vložit nové moduly, nebo stávající moduly přesunout z cesty RS232 pod cestu NPortWay. Při přesouvání označíme ve stromu cest a modulů cestu, v pravé části označíme všechny moduly zvolíme **Úpravy/Vyjmut**, nebo klávesovou zkratkou **Ctrl+X**. Vybereme cestu NPortWay a zvolíme **Úpravy/Vložit**, nebo stiskneme klávesy **Ctrl+V**. Označíme moduly, které komunikují přes stejný převodník NPort a zvolíme **Vlastnosti a dynamizace** **Ctrl+ENTER**. V **Parametrech spojení**, položce **Tel. číslo, adresa...** zadáme IP adresu převodníku NPort, přes který jsou dané zařízení připojeny. Můžeme zvýšit **Počet opakování** v parametrech komunikace, například na 10 pokusů.

*IP adresa NPort převodníku v modulu M-Bus.*



V jedné NPort cestě mohou být moduly, které jsou za různými zařízeními. IP adresy v parametru spojení musí odpovídat danému NPortu. Logické adresy (např. M-Bus) zůstávají beze změny.



#### Poznámka

*Pokud jsme přesouvali moduly z cesty RS232 a je cesta již prázdná, můžeme ji zrušit.*

## 1.4 Problémy s komunikací

---

Po spuštění projektu by měla komunikace se zařízeními fungovat. Uvedme pár případů problémů, které mohou nastat a jak je řešit.

### Zařízení nekomunikují

---


Nejprve zkusíme provést test, zda je NPort po síti z PC dosažitelný.


Spustíme konzolu příkazového řádku. V nabídce **Start**, v řádku **Spustit** napíšeme "**cmd**" a stiskneme **ENTER**, nebo zvolíme **Start\Všechny programy\Příslušenství\Příkazový řádek**. Otevře se okno s obvykle černým pozadím. napíšeme příkaz:



```
ping 192.168.1.240
```

kde použijeme místo 192.168.1.240 IP adresu daného převodníku. Pokud převodník odpovídá, vypisují se časy odezvy paketů. Převodník je tedy na síti viditelný.

Problém může být v nastavení Windows Firewall. Otevřeme v Ovládacích panelech jeho správu a definujeme výjimky pro všechny použité UDP porty např. 4001, 4002 ... Zkontrolujeme, že jsou výjimky na firewallu povoleny.

Druhý problém může být chybně konfigurovanou komunikační cestou - zkontrolujeme podle nastavení 


Také může být nevyplněna, nebo chybně vyplněna IP adresa NPort převodníku v komunikačním modulu, zkontrolujeme 

Dále můžou být chybně nastaveny parametry (IP adresy, UDP port ) v převodníku NPort, nebo komunikační parametry na sériovém portu NPort.

Pokud je vše v pořádku, je pravděpodobné, že je problém za NPortem na lince RS232.

### Zařízení komunikují, ale objevují se komunikační chyby

---

Zkontrolujeme parametry komunikačních modulů  v Panelu nástrojů ProCop, zejména **Počet opakování** a **Čekání na odpověď**.

Nebo je síť nespolehlivá, přetížená a podobně.

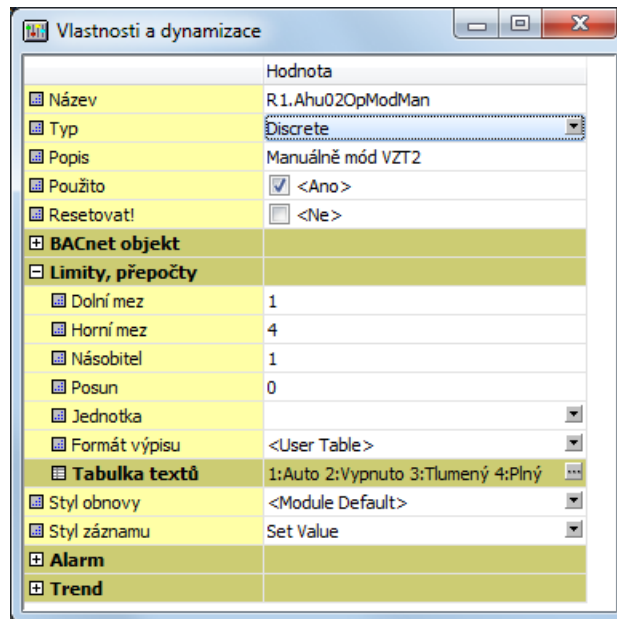




## 2 Uživatelský formát kanálů

Formát výpisu binárních hodnot je od verze ProCop 3.3 nastavitelný obdobně, jako u diskretních hodnot. V tabulce vlastností a dynamizací se nově zobrazuje náhled jednotlivých významů hodnot, zadaných v tabulce.

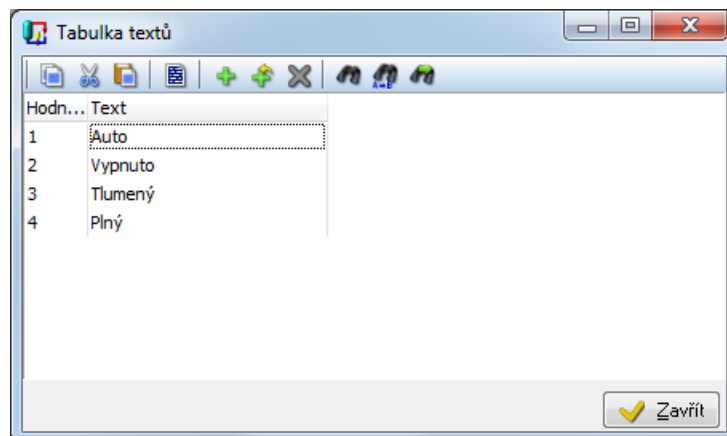
*Uživatelská tabulka a význam hodnot u diskretních (binárních) kanálů*



### Uživatelské formáty binárních a diskretních hodnot

Pokud přednastavené hodnoty nevyhovují potřebám projektu, stačí vybrat volbu <User Table> a nastavit si v tabulce významy jednotlivých hodnot podle potřeby. V tabulce se pro diskretní hodnoty dají zadat hodnoty 0 - 255 a k nim příslušné texty. U binárních hodnot mají význam jen hodnoty 0 a 1. Ostatní jsou ignorovány.

*Uživatelská tabulka a význam hodnot u diskretních (binárních) kanálů*



### Kopírování uživatelských formátů

Kopírování můžeme provádět v tabulce označením a kopírováním řádků. Nově je možné přímo v tabulce vlastností a

dynamizací vybrat řádek **Tabulka textů** a obvyklou kombinací kláves **Ctrl+C** formát zkopírovat a následně vložit do Tabulky textů jiné diskretní, nebo binární proměnné kombinací **Ctrl+V**.


## 2.1 Nastavovací dialogy

Pro diskretní kanály, které mají nastaven formát výpisu na uživatelskou tabulku, nebo přednastavený výčet a pro všechny binární kanály, jsou od ProCop verze 3.3 zobrazovány výběry v nastavovacích dialogích.

*Nastavení hodnoty binárního kanálu - nyní jen výběr hodnoty*

Dialog pro nastavení hodnoty (SetValueDialog) je pro diskretní a binární hodnoty automaticky doplněn přepínači se všemi popsanými hodnotami formátu. Meze a zadávací řádek je zrušen.

*Nastavení hodnoty diskretního kanálu - automaticky vygenerované přepínače*

V nastavovací tabulce hodnot jsou k dispozici vždy jen výběry hodnot pro binární a pro diskretní hodnoty s uživatelskou tabulkou hodnot. Formáty se nastavují přímo u kanálů  a není potřeba je předefinovávat v displeji u konkrétního dialogového okna.

*Nastavení hodnoty v tabulce - diskretní a binární kanál s uživatelským formátem*

## 3 Automatické vlastnosti potrubí

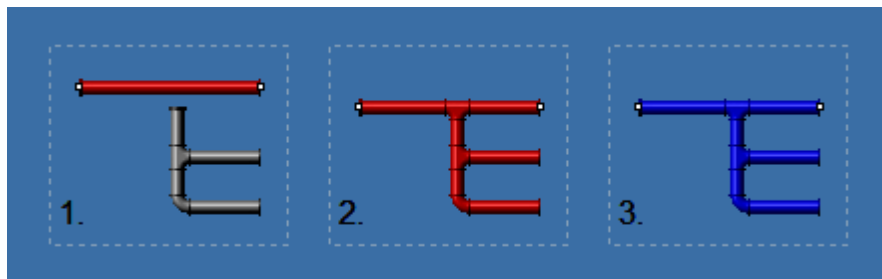
ProCop verze 3.3 disponuje možností nastavit automatickou barvu potrubí (i vzduchotechnického). Potrubí se vykresluje šedě, pokud není napojeno na potrubí jiné, než automatické barvy.

Nastavení automatické barvy potrubí

Vlastnosti potrubí	
Barva	<Automatická>
Šířka	7 bodů

Následující obrázek ve třech krocích zobrazuje, k čemu je možné použít automatické barvy potrubí. V celé síti trubek stačí jediná s danou barvou, například přívodní. Po připojení na ostatní potrubí, všechny navázané s automatickou barvou kopírují barvu připojeného potrubí. Změníme-li barvu jediného potrubí, změní se barva i ostatních připojených.

Označená trubka s pevnou barvou, ostatní automatické, napojení a změna barvy



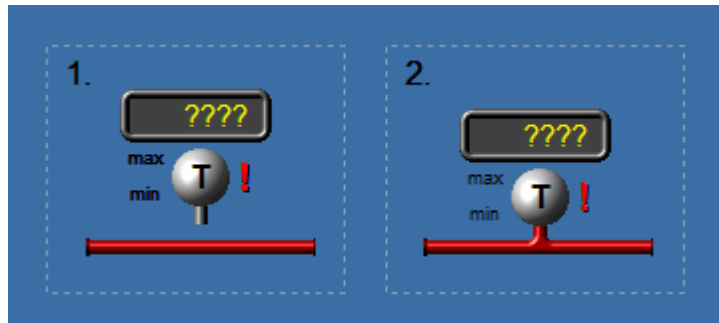
### Napojování potrubí do skupin a knihovních bloků

Výše popsaná vlastnost se hlavně uplatní v kombinaci s knihovními bloky a skupinami. Vložíme-li knihovní blok, nebo vytvoříme-li skupinu s příznakem **Spojovat automaticky**, potrubí se napojí i v rámci knihovního bloku a skupiny, případně přebere barvu potrubí.

Knihovní blok, nebo skupina s příznakem automatického spojování

Příznaky	
Zmrazit entitu	<input type="checkbox"/> <Ne>
Zakázat výběr myši	<input type="checkbox"/> <Ne>
Netisknout	<input type="checkbox"/> <Ne>
Zakázat rotaci	<input type="checkbox"/> <Ne>
Zakázat export	<input type="checkbox"/> <Ne>
Spojovat automaticky	<input checked="" type="checkbox"/> <Ano>
Úroveň přístupu	Trivial
Perioda obnovy [ms]	1000

Podmínkou je automatická barva potrubí v knihovním bloku či skupině a příznak **Spojovat automaticky**. Posuneme-li knihovní blok, či skupinu do blízkosti potrubí, automaticky se napojí a změní svou barvu.

*Napojení knihovního bloku na potrubí*

## 4 Ukládání stavových proměnných

V případě požadavku na ukládání stavových proměnných systému jako je například datum pro export dat zadávané obsluhou je možno využít IO modul MemoryDatabase (paměťová databáze). Tento modul inicializuje hodnoty na hodnotu 0, pokud není zadáno konfiguračně jinak.

Proměnné jsou implicitně ukládány do souborů \*.iod, které jsou považovány za dočasné a jejich formát a obsah může být mezi verzemi systému ProCop změněn. Také může dojít k destrukci jejich obsahu například po výpadku napájení, havárii disku apod.

### **Důležité upozornění!**

*Nedoporučujeme ukládat žádná kritická data do těchto proměnných, při výpadku napájení, nekorektním ukončení aplikace, nebo upgradu systému může dojít k jejich ztrátě!*

Pokud je zapotřebí vyšší bezpečnost pro uložení hodnot stavových proměnných, doporučujeme použít volbu "Ukládat do inicializace (pio)". V tomto případě dojde k přepsání inicializační hodnoty ve zdrojovém souboru projektu typu \*.pio a mělo by následně dojít k archivaci projektu obsluhou tak, aby nemohlo dojít k přepsání stavové proměnné při obnovení projektu ze zálohy (v případě pádu disku apod.).

Pokud je zapotřebí ukládat stavové proměnné s nejvyšší úrovní bezpečnosti, doporučujeme využít externí úložiště jako např. SQL server případně CSV, DBF či INI soubor například na externím zabezpečeném disku. Při využívání stavových proměnných z externích úložišť vysoce doporučujeme kontrolovat stav a validitu dat získaných z těchto úložišť.

### **Poznámka**

*Nastavením vlastnosti "Ukládat do inicializace (pio)" zvýšíte bezpečnost přenosu jejich hodnoty mezi starty systému i při upgrade. Nevyužívejte však pro často se měnící proměnné (skriptem), aby nedocházelo k neustálým modifikacím zdrojového souboru. Vždy je nutné si uvědomit, že pokud i tak dojde ke ztrátě hodnoty, iniciální "tovární" hodnota je 0. Tento stav by měl být pro následné využití považován za bezpečný a s touto logikou je nutné v systému uvažovat.*



## 5 Atributy knihovních bloků

Knihovní bloky jsou již od ProCop verze 3.2 vybaveny funkcí pro definici atributů. Ve verzi 3.3 byly tyto vlastnosti dle zkušeností dále upraveny, aby se lépe a jednodušeji používaly

### K čemu atributy slouží

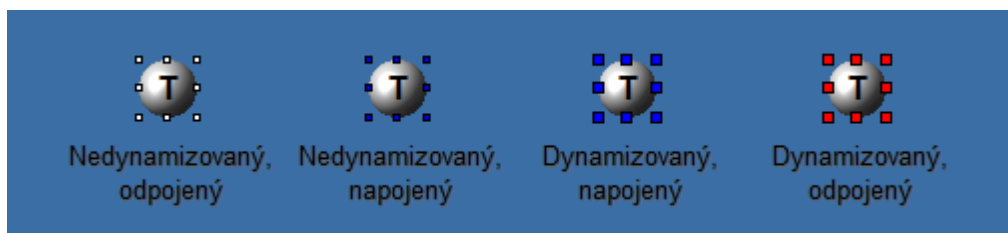
Knihovní bloky po vložení do displeje bylo nutné obvykle oddělit a jednotlivé dynamizované části napojit na požadované proměnné. Nevýhodou je pracnost a odpojení od knihovny. Při úpravě takového bloku v knihovně se změny neprojeví v displejích, jelikož jsou bloky od knihovny odpojeny.

Atributy se snaží tento problém vyřešit. Bloku v knihovně se nadefinují virtuální proměnné - atributy - které se pak v editoru pouze napojí. Nedojde tak k odpojení z knihovny a případné úpravy bloku v knihovně se projeví ve všech instancích v displejích. Navíc je možné atributy vyrobit tak, aby svými jmény přispěly k automatickému napojení ostatních proměnných. K tomuto napojení v displeji slouží [Průvodce napojení atributů](#) **Ctrl+Q**.

### Barevné významy úchopových bodů

Pro informaci o napojení bloků do knihovny se úchopové body body vykreslují modře. Zvětšení úchopových bodů, případně červená barva znázorňuje, že je entita dynamizována a obsahuje odkazy na technologické proměnné.

*Možné barvy a velikosti úchopových bodů: modrá = napojený knihovní blok, zvětšení = dynamizovaná entita*



### 5.1 Výroba bloku s atributy

Pro snadné pochopení bude nejlepší vyrobit knihovní blok s atributy a pak jej použít v displeji.

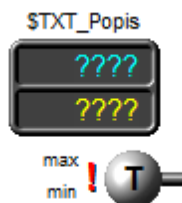
Vybereme nějaký kus z existujícího projektu, nejlépe takový, který se více krát opakuje, ale je pokaždé napojen na různé proměnné. Jako příklad bude použita měřená a žádaná teplota náběhu ve směšovací okruhu MxCrt v regulátoru PX. Nakopírujeme si jej do schránky **Ctrl+C**.

Nejprve otevřeme ve stromu [Knihovnick](#) projektovou knihovnu [Project Library](#). Přidáme knihovní blok například tlačítkem **Insert**, pojmenujeme jej a otevřeme.

#### Definice atributů

Vložíme obsah schránky do knihovního bloku a zvolíme [Úpravy/Převod na atributy](#). Nezapomeneme vybrat případná napojovaná potrubí a změníme jim barvu na automatickou. Celému knihovnímu bloku ve stromu knihovnicka nastavíme příznak [Spojovat automaticky](#), je-li to potřeba.

Knihovní blok po převodu na atributy s automatickou barvou trubky



V nástrojové liště **Vlastnosti** zvolíme **Upravit vlastnosti displeje** (symbol modrého klíče) a ve vlastnostech a dynamizacích zobrazíme **Tabulku atributů**. V tabulce atributů nyní vidíme seznam jednotlivých automaticky vytvořených atributů.

#### Každý atribut má tyto vlastnosti:

- **Název** - jméno atributu, v dynamizacích bloku se použije místo proměnné, na začátku doplněný o znak \$
- **Popis** - textový popis atributu, zobrazovaný při napojování knihovního bloku v průvodci a vlastnostech, měl by přesně definovat význam proměnné
- **Typ** - požadovaný významový typ atributu - pro výběr a zadání slouží tlačítko se třemi tečkami - atributu smí být přiřazena jen:
  - **Proměnná/Hodnota** - technologická proměnná, nebo konstantní hodnota
  - **Proměnná** - technologická proměnná, sufix
  - **Výraz** - libovolný výraz
  - **Text** - libovolný text - průvodce vždy vyžaduje nové vyplnění, nebo potvrzení stávajícího textu
  - **Text implicitní** - libovolný text - průvodce nevyžaduje vyplnění, vždy použije předvolbu, ale je text možno změnit
  - **Text skládaný** - libovolný text - průvodce vždy vyžaduje vyplnění, ale jen jednou pro všechny vnořené bloky s atributem stejného jména
  - **Hodnota** - lze zadat jen konstantu
  - **Seznam** - v editoru zadáme seznam zobrazovaných názvů a reálných hodnot atributu, vhodné pro výběr z entit
  - **Zaškrťovací políčko** - v editoru zadáme popis a hodnotu pro zaškrtnuté a prázdné políčko, vhodné např. pro viditelnost entity
- **Datový typ** - konkrétní datový typ atributu, tedy přiřazené proměnné, výrazu, textu, nebo jakýkoliv typ
- **Předvolba** - implicitně nabízená předvolba hodnoty atributu, prázdná u hlavního atributu, odkazy na příznaky jiných atributů, implicitní text...
- **Hodnota nepoužití** - pokud může být atribut napojen, zde se uvede hodnota, která daný atribut vyřadí tak, aby se v bloku neprojevoval

Tabulka atributů po automatickém převodu atributů

Název	Popis	Typ	Datový typ	Předvolba	Hodnota nepoužití
STXT_Popis	Popis Teplota náběhu	Text impli...	Text	Teplota náběhu	
HysFrPrt	Hystereze protizamrazu	Proměnná...	Analog		0
HysFrPrt_Ack	Hystereze protizamrazu - Ack	Proměnná...	Binary	\$HysFrPrt.Ack	0
HysFrPrt_Status	Hystereze protizamrazu - Status	Proměnná...	Discrete	\$HysFrPrt.Status	0
SpDtr	Určení žádané hodnoty	Proměnná	Analog		
TFl	Teplota náběhu	Proměnná	Analog		
TFl_Ack	Teplota náběhu - Ack	Proměnná...	Binary	\$TFl.Ack	0
TFl_Reliability	Teplota náběhu - Reliability	Proměnná...	Discrete	\$TFl.Reliability	0
TFl_Status	Teplota náběhu - Status	Proměnná...	Discrete	\$TFl.Status	0

Názvy atributů se používají místo proměnných, nebo jejich sufixů v dynamizacích, hodnotách či textech entit. Název uživatelského atributu se uvozuje znakem \$ na začátku, například \$STXT\_Popis.



## Systémové atributy

Vyjma uživatelských atributů jsou k dispozici **předdefinované systémové atributy**. Tyto atributy se mohou použít přímo v dynamizacích, či textech knihovních bloků, nebo rovněž v předvolbách definice uživatelských atributů. Tyto atributy jsou nabízeny v rozbalovacích seznamech (při vkládání proměnné, předvolby atributu příslušného typu) a jsou z obou stran uvozeny znaky `$sys.att$`.

V současnosti jsou k dispozici tyto systémové atributy:

- **\$Display.Name\$** - jméno technologického displeje, ve kterém je blok vložen
- **\$Display.Descr\$** - popis technologického displeje, ve kterém je blok vložen
- **\$Display.Title\$** - popis, nebo jméno technologického displeje, je-li popis prázdný
- **\$Visual.Name\$** - jméno vizuálního projektu
- **\$Visual.Descr\$** - popis vizuálního projektu
- **\$Visual.Title\$** - popis, nebo jméno vizuálního projektu, je-li popis prázdný
- **\$Project.Name\$** - jméno datového projektu
- **\$Project.Descr\$** - popis datového projektu
- **\$Project.Title\$** - popis, nebo jméno datového projektu, je-li popis prázdný



### Poznámka

Je vhodné používat systémové proměnné `$.Title$`, neboť máme zaručeno, že není-li zadán popis, bude doplněno jméno dle daného atributu.

## Použití atributů v dynamizacích knihovního bloku

Máme-li nadefinovány atributy knihovního bloku, nebo automaticky vygenerovány příkazem **Úpravy/Převod na atributy**, můžeme přejít k tvorbě a úpravám dynamizací jednotlivých entit. Převod na atributy automaticky provede nejen vytvoření atributů, ale i úpravy dynamizací všech entit knihovního bloku. Dynamizace je však vhodné zkontrolovat.

Pokud vytváříme dynamizace nové, používáme atributy, či systémové atributy shodně, jako by se jednalo o proměnné. V dialogovém okně pro **Výběr proměnné, funkce a výrazu** lze v rozbalovacím seznamu **Proměnná nebo výraz vybrat** příslušný atribut.

Pro použití atributů zejména typu Proměnná/Hodnota byly definovány dvě nové funkce jazyka Bára:

<i>Funkce</i>	<i>Popis</i>
<b>VarExist( variable ) of binary</b>	Vrací TRUE, pokud proměnná existuje
<b>VarOrConst( ident ) of binary</b>	Vrací TRUE, je-li parametr proměnná či konstanta

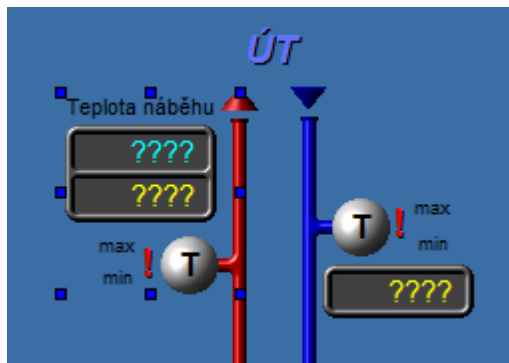
Funkce VarExist vrací logickou hodnotu existence proměnné. Pokud byla do atributu typu Proměnná/Hodnota zadána hodnota (konstanta, např. z parametru hodnota nepoužití), je možné tuto skutečnost v dynamizacích ošetřit, například v lokální nabídce podmínkou viditelnosti daného řádku, v přiřazení hodnoty podmínkou pro provedení přiřazení a podobně, dynamizovat na viditelnost entity dle výsledku funkce a podobně. Obdobně lze využít i funkci VarOrConst pro otestování, zda byla proměnná, nebo hodnota vůbec do atributu zadána.

## 5.2 Použití bloku s atributy

Máme-li vyroben knihovní blok s atributy, můžeme jej zkusit použít v projektu a napojit. Případně zvolíme blok dodávaný v knihovně monitorovacího systému ProCop.

Otevřeme příslušný technologický displej a zvolíme **Kreslení/Vložit blok z knihovny** klávesa **⌘**. Vybereme námi vyrobené nebo dodávaný knihovní blok. Vložíme jej, umístíme jej kam potřebujeme a zkontrolujeme, že se nám potrubí propojilo a má správnou barvu. Pokud ne, blok nemá v knihovně nastaven příznak **Spojovat automaticky**, nebo vnořený blok knihovního bloku nemá tento parametr nastaven. Případně nemá potrubí nastavenou **automatickou barvu**.

*Knihovni blok s atributy po vložení a napojení na technologii*

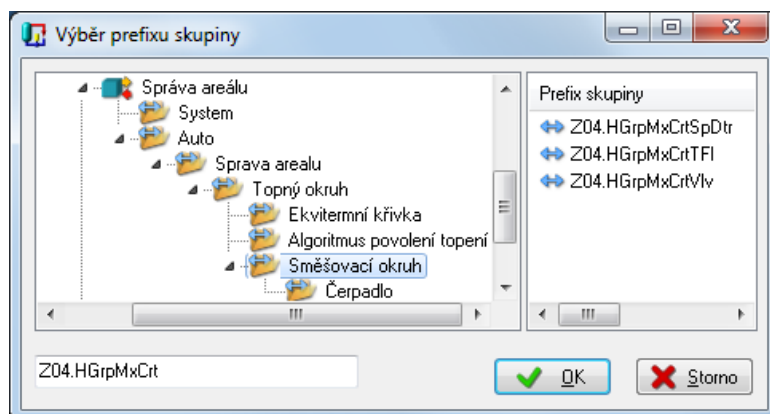


## Průvodce napojení atributů

Po vložení máme dvě možnosti, jak s knihovním blokem s atributy pracovat. Jednodušší je spustit z nabídky [Úpravy/ Průvodce napojení atributů](#) **Ctrl+Q**.

Pokud je knihovni blok nově vložen, nejprve se objeví dotaz na [Výběr prefixu a skupiny](#).

*Výběr prefixu skupiny proměnných pro napojení atributu*



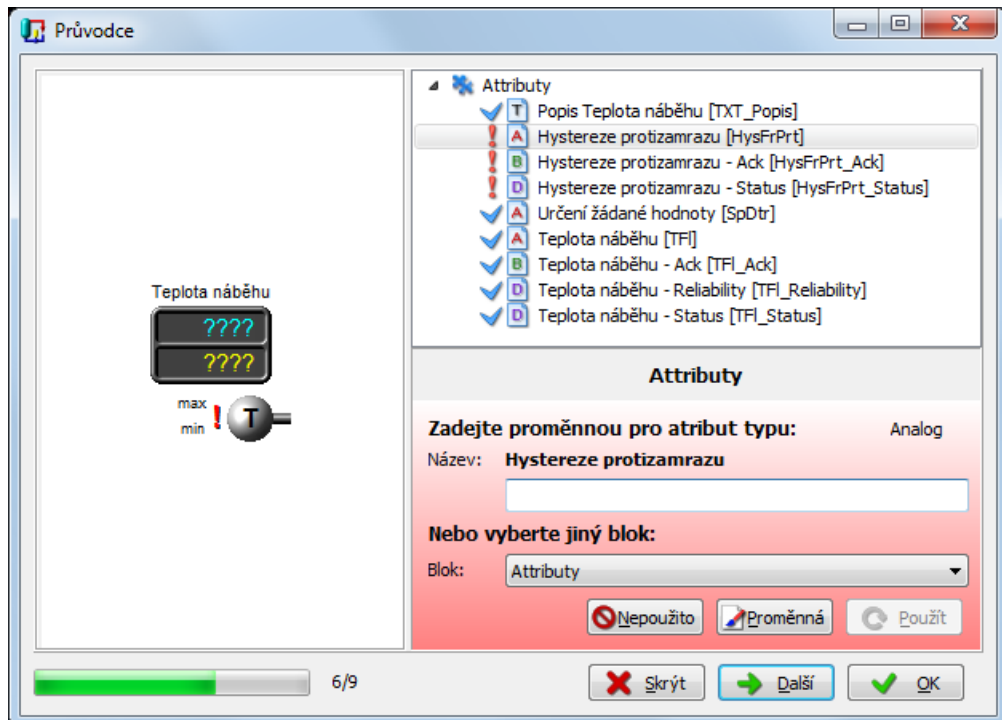
Po regulátoru a skupiny proměnných, na které budeme dále atributy napojovat se automaticky spustí propojování atributů. K zadanému **prefixu** skupiny se připojí postupně **jméno** všech atributů, které byly zadány v knihovním bloku. **Existuje-li proměnná**, vzniklá **spojením vybraného prefixu a jména atributu** (bez znaku \$), **automaticky se proměnná napojí na atribut** a pokračuje se dalším atributem.

Pokud není proměnná nalezena, automatické napojování se zastaví a čeká se na zásah obsluhy.

**Jsou k dispozici tyto možnosti:**

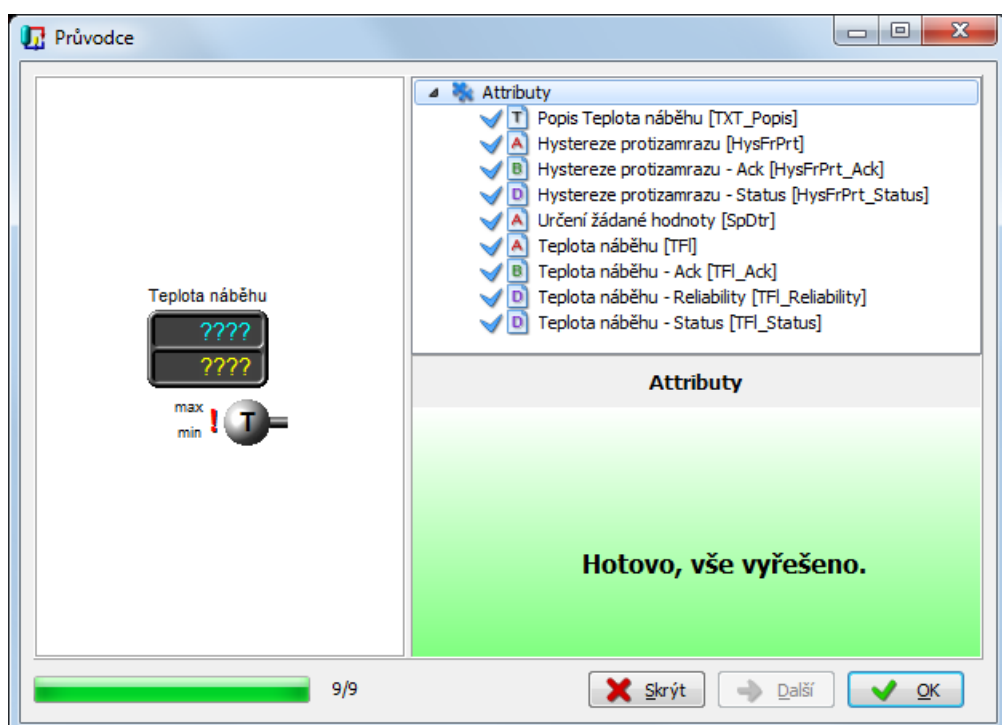
- požadovaná proměnná se jmenuje jinak, nebo je z jiné skupiny - je potřeba zadat či vybrat správnou proměnnou
- daný vložený blok je jiného typu, je možné **Vybrat jiný blok**, byl-li definován (např. 1 či 2 stupňový ventilátor, ...)
- daný podblok není implementován, můžeme jej celý **Skrýt**
- daná proměnná neexistuje, tlačítkem **Nepoužito** vložíme **Hodnotu nepoužití**
- pokud nic nevyhovuje, musím ukončit průvodce, blok oddělit a upravit dynamizovat pole potřeb, již napojené atributy se přepíšou proměnnými

V dané implementaci nepoužitá proměnná, problém vyřešíme tlačítkem nepoužito



Pokud byly napojeny všechny proměnné, přepsány všechno požadované texty a vyřešeny všechny problémy, průvodce úspěšně napojení ohlásí. Procházením po stromu atributů můžeme zkontrolovat jejich napojení, v nejvyšší úrovni lze **změnit celý prefix** napojování bloku.

*Všechny atributy byly napojeny, nebo ošetřeny, knihovní blok je napojen*



## Ruční úpravy napojení atributů

Druhou variantou úprav je ruční napojení jednotlivých atributů ve **Vlastnostech a dynamizacích** klávesy **~**, **Ctrl + Enter** daného knihovního bloku. Napojíme-li jeden atribut, všechny ostatní nenapojené se stejným algoritmem jako v průvodci pokusí napojit na proměnné.



## 6 Parametry modulu PX

---

Parametry komunikace modulu PX jsou obdobné jako u ostatních komunikačních modulů. Výjimkou je chování objektů schopných zasílat změny COV (ChangeOfValue).

	Hodnota
Název	PZ
Prefix	PZ
Popis	
Soubor	pz.pio ...
Zakázáno	Ne
<b>0 modulu</b>	PxModule
<b>Parametry komunikace</b>	
Adresa (hexa)	00200401
Permanentní obnova	1 h
Počet opakování	3
Styl alarmu komunikace	<Comm Alarm>
<b>Perioda obnovy</b>	
Perioda (0=vyp)	5 m
Posunutí periody	0 s
Obnov při startu	<input type="checkbox"/> <Ne>
Validita (0=vždy)	0 s
První obnova (0=od půlnoci)	0 s
Poslední obnova (1d=do půlnoci)	1 d
<b>Parametry regulátoru</b>	
Typ regulátoru	PXC100
Verze PXC	Desigo 4.x
Priorita ovládání	8
COV perioda (0=vyp)	10 m
Hlášení poruch čidel	Sensor Error
<b>Speciální funkce</b>	
Soubor autokonfigurace	Bagetarna_Olomouc_E... ...
Opravit adresy pinů v CSV	<input checked="" type="checkbox"/> <Ano>
Název stanice	AS01
Ignorovat prefixy do úrovně	1
Autogenerace	Autokonfigurace
Rozdíl CSV proti autokonfiguraci	Autodiference
<b>Konverze textů ve skupinách</b>	...
<b>Konverze textů při autokonfiguraci</b>	...
<b>Podrobná nastavení</b>	
Kontrolovat vždy po startu	Ne
Chybu čtení jako událost	Ne
Čist verzi databáze regulátoru	Ano
Převést název objektu na instanci	Ano
Kontrolovat název objektu a instanci	Ne
Opravit chybu akceptace alarmu	Ne
Kontroluj verzi BACnet	Ano
Čti vlastnost AckRequired	Ne
Ignoruj BACnet Error rámce	Ne
Opravuj adresy Pinů za běhu	Ne
Generuj alarm pro resetování	Ano
Generuj půlnoc v Scheduleru	Ano
Manuální režim v Scheduleru	Ano
Čti výjimky ze Scheduleru	Ano
Ukaž seznam poruch v CommonFault	Ano
Ukaž ComnAlarms v CommonFault	Ne
Ukaž definované alarmy v CommonFault	Ne

## 6.1 Parametry regulátoru

### Typ regulátoru

Nastavuje přesný typ regulátoru Desigo PX. Je nutno, aby bylo shodné se skutečným typem regulátoru a kontroluje se za běhu systému. V případě, že typ nebude zadán správně, komunikace se ukončí a bude hlášena porucha komunikace.

### Verze PXC

Určuje, zda je firmware regulátoru ve verzi 2.3x, 4.0 nebo 5.0 (BACnet verze 1.0, 1.5 a 1.10). Kontrola je prováděna v případě zapnutí podrobné volby "Kontroluj verzi BACnet".

### Priorita ovládání

Slouží pro prioritu při ovládání v BACnet vlastnosti PRIORITY\_ARRAY. Po kompatibilitu s Desigo PXC ponechte na implicitní hodnotě 8.

### COV perioda

Protokol BACnet je obecně schopen zasílat změny hodnot z regulátoru do nadřazeného systému (PC). Tuto schopnost podporují i podstanice řady Desigo PX.

Modul při komunikaci požádá regulátor o zaslání změn dat k jednotlivým proměnným s maximální délkou trvání implicitně 30 minut. Po vypršení této doby pak modul vyčte všechna data a následně opět požádá regulátor o registraci a zaslání změn. V periodickém čase obnovy pak modul pouze čte čas regulátoru případně čte proměnné, které mají zakázáno čtení COV nebo regulátor čtení přes COV nepodporuje. Modul u proměnných, kde COV probíhá pak zvyšuje čas obnovy v okamžiku příjmu změny či v době periodické obnovy (simuluje tak příjem dat).

#### **Důležité upozornění!**

*V případě přetížení komunikační linky k regulátoru může být změna zasláná regulátorem ztracena - regulátor se snaží provést doručení max. 3x. Z toho důvodu je nevhodné periodu COV příliš zvyšovat.*

Vypnutí této možnosti lze provést nastavením hodnoty na 0. V tomto případě pak modul čte periodicky data v nastavené periodě modulu respektive v periodách daných stylem obnovy u proměnné (zcela standardním způsobem jako u jiných modulů s komunikací typu dotaz-odpověď).

Potlačení této schopnosti lze provést i u jednotlivých proměnných použitím vestavěného stylu obnovy "NoCOVobjects" případně tvorbou vlastního uživatelského stylu s vypnutou volbou "Používat COV".



#### **Poznámka**

*Tato volba je k dispozici pouze pro modul v PC nikoliv v AlfaBox.*

### Hlášení poruch čidel

Styl alarmů pro hlášení poruch čidel. Porucha je indikována čtením BACnet vlastnosti Reliability.

## 6.2 Speciální funkce

### Soubor autokonfigurace

Soubor autokonfigurace může být typu CSV nebo XLS a je generován pomocí nástroje EDE Export (v Desigo 4.x a 5.x pak volbou nabídky). Soubor poskytuje seznam BACnet objektů, jejich typů popisů a dalších parametrů. V případě použití verze XLS je systém schopen jej přečíst pomocí rozhraní OLE Automation. V tomto případě je nutná instalace MS Excel. Systém pak soubor uloží do rozšířeného souboru CSV, který je možno číst i bez instalovaného systému MS Excel.

Výhoda použití EDE Exportu do XLS spočívá v tom, že v tomto formátu jsou k dispozici další informace oproti exportu do CSV - například texty popisující hodnoty diskrétních stavů. Toto platí pro Desigo XWorks do verze 4.x.

**⚠ Důležité upozornění!**

Nástroj EDE Export verze 4.x nepracuje v některých případech korektně v případě exportu několika podstanic současně do jednoho souboru. Doporučujeme proto provést export pro jednotlivé podstanice do oddělených souborů typu XLS.

Od verze 5.0 jsou programem EDE Export generovány další 3 dodatečné soubory s názvy stejnými jako hlavní soubor rozšířené o přípony -Object-Types, -State-Texts a -Unit-Texts. V tomto případě je funkčnost v případě využití XLS a sady CSV souborů shodná.

**⚠ Důležité upozornění!**

Systém ProCop potřebuje všechny údaje o BACnet databázi v jednom souboru. Dodatečné soubory jsou proto automaticky importovány do hlavního souboru .CSV a dodatečným souborům je přidána nová přípona .MERGED. Kopírujte proto vždy do adresáře modulů všechny soubory (hlavní i dodatečné) společně a najednou !!!

## Opravit adresy pinů v CSV

Volba způsobí opravu chybných adres pinů objektů generovaných EDE Exportem verze 5.0. Adresy budou opraveny při generování souboru CSV z XLS, při spojování souborů s popisy objektů a stupňů nebo při prosté autogeneraci či autodiferenci. Výsledný soubor CSV je použitelný jako zdroj pro systém ProCop starších verzí.

Seznam objektů, u nichž dochází k opravě je uložen v souboru PxOptions.xml, který je uložen v podadresáři Options instalace systému ProCop. Seznam je uložen v sekci "BadPinFunctions" a je možné jej ručně rozšiřovat. Standardně je dodáván soubor s tímto seznamem opravovaných objektů:

```
<BadPinFunctions>
  <Item0>
    <Function t="s">DelOds</Function>
  </Item0>
  <Item1>
    <Function t="s">DelPor</Function>
  </Item1>
  <Item2>
    <Function t="s">Hra</Function>
  </Item2>
  <Item3>
    <Function t="s">KickFnct</Function>
  </Item3>
  <Item4>
    <Function t="s">KickFnct1</Function>
  </Item4>
  <Item5>
    <Function t="s">KickFnct2</Function>
  </Item5>
</BadPinFunctions>
```

## Název stanice

Musí zde být zadán přesný název stanice podle jména v EDE exportu. V opačném případě nebudou žádné proměnné importovány.

## Ignorovat prefixy do hloubky

Umožňuje při tvorbě stromu proměnných ignorovat prvých N úrovní stromu. V případě zbytečně příliš hlubokého stromu proměnných zkuste zvýšit úroveň.

## Autogenerace

Načte data z EDE exportu a vygeneruje z něj databázi regulátoru do podskupiny "Auto".

## Rozdíl CSV proti autokonfiguraci

Volba Autodiference je vhodná v případě, že potřebuje najít větší počet změn po nahrání nového aplikačního softwaru do podstanice. V poměrně rozsáhlých stromech proměnných je poměrně obtížné najít, co aplikační programátor v podstanici doplnil, ubral či změnil.



**★ Tip!**

*Pokud je databáze "vyčištěná" od nepotřebných proměnných, doporučujeme před volbou Autodiference provést volbu Autokonfigurace s původní verzí CSV souboru, která zpět doplní zrušené proměnné.*

Volba Autodiference vygeneruje tři skupiny obsahující změny nové verze softwaru podstanice oproti současné databázi. Před volbou Autodiference je nutné nakopírovat nový soubor vytvořený pomocí nástroje EDE Export. Vytvoří se dočasné skupiny:

- "Removed channels, not use !" - skupina obsahuje proměnné, které v nové verzi EDE exportu chybí
- "Added channels, not use !" - obdobně obsahuje proměnné, které v nové verzi EDE Exportu přibyly
- "Dif. object type, not use !" - skupina obsahuje proměnné, jejichž BACnet typ byl oproti stávající databázi změněn

Skupiny slouží pouze pro orientaci, tak aby bylo zřejmé, co se mezi verzemi software podstanice změnilo, nikoliv pro vizualizaci. V případě spuštění modulu jsou skupiny automaticky smazány.

## Konverze textů ve skupinách

V případě potřeby náhrady standardních textů importovaných z EDE Exportu je k dispozici speciální nástroj, který dokáže generovat dle skupiny a standardního názvu objektu popisy proměnných. Slovník dokáže vkládat zvláštní text namísto zástupného symbolu @ a dále je schopen generovat definici alarmu, trendu a trendy řadit do skupin. Namísto zástupného symbolu je tak možno dynamicky vkládat dle potřeby například krátké názvy jako je TUV3 apod. Seznam a definice konverzního předpisu je možno dle potřeby upravovat či doplňovat.

**★ Tip!**

*Konverzi textů ve skupinách je nutno u každé skupiny v jejich vlastnostech spustit stiskem tlačítka Konverze textů - Spustit. Před spuštěním je vhodné doplnit hodnotu zástupného symbolu @.*

## Konverze textů při autokonfiguraci

System Desigo PX nepoužívá v národní verzi háčky a čárky. Proto má modul implementován jednoduchý překladový slovník, který jednotlivé "ceske" výrazy při importu mění na "české". Slovník je možno doplňovat o vlastní konverzní páry textů dle potřeby.

## 6.3 Podrobná nastavení

### Kontrolovat vždy po startu

Při každém startu monitorování provede kontrolu všech BACnet objektů, jejich typů, instancí (ignoruje uložené hodnoty od minule). Zjistí tak, pokud v PXC někdo od minulého spuštění přehrál SW. Kontrolu je možno vynutit v řízení modulů vyřazením a zařazením do obnovy.

### Chybu čtení jako událost

Pokud nelze přečíst proměnná, generuje událost "Nelze obnovit proměnnou ..." jinak ji pouze znevalidní (do další pravidelné obnovy dat).

### Čist verzi databáze regulátoru

Regulátor PX má uloženo pořadové číslo své databáze, které se po přehrání SW automaticky zvyšuje. Po zvýšení tohoto pořadového čísla (on-line přehrání programátorem za provozu dispečinku) modul automaticky provede kontrolu typů, instancí BACnet objektů apod.

### Převést název objektu na instanci

Nutno nechat nastaveno u regulátorů, které si číslo instance při každé úpravě SW libovolně upraví a mění (např. řada Desigo PX).

---

## Kontrolovat název objektu a instanci

---

V případě vypnutí předchozí volby (u regulátorů, kde je to pevně nastaveno), tak je to alespoň vhodné kontrolovat typ (ochrana proti chybě zadání databáze).

---

## Opravit chybu akceptace alarmu

---

Potlačení chyby implementace kvitování alarmů v regulátorech Delta Controls.

---

## Opravuj adresy Pinů za běhu

---

Koriguje chybu nástroje pro export databází regulátorů EDE Export. Tento nástroj je součástí distribuce systému Desigo XWorks 5.0. Používá se při generování seznamů objektů včetně jejich popisů typů a adres. V případě generování seznamu včetně tzv. Pinů jsou chybně exportovány BACnet adresy pinů. Namísto tečky oddělující pin je exportován apostrof.

Tato volba je implicitně zapnuta pro Desigo 5.0. Volba zkusí po třetím neúspěchu změnit automaticky adresu a zkouší číst znova. Toto velmi zpomaluje komunikaci při kontrole databáze. V případě vypnuté volby a chyby v EDE exportu jsou proměnné s vadnými adresami vyřazeny z obnovy a zalogovány do událostí. Korekce je pak nutno provést manuálně.

---

## Kontrolovat verzi BACnet

---

Čte a kontroluje z Device Objektu verzi BACnet protokolu (jestli je dobře nastavena verze Desigo 2x, 4.x, 5.x)

---

## Čti vlastnost AckRequired

---

Potlačení čtení BACnet vlastnosti AckRequired - tato vlastnost u Desigo 4.0 chybí.

---

## Ignoruj BACnet Error rámce

---

Modul zcela ignoruje příjem chybových rámců protokolu BACnet. Nutné v případě, kdy regulátor generuje nesmyslné chybové rámce (Delta Controls).

---

## Generuj alarm pro resetování

---

V případě rozšířené poruchy u regulátorů řady Desigo PXC je možné zapnout generování alarmu se stylem Alarm Reset. Tento alarm je generován v případě odeznělé (Status=0) a nekvitované poruchy (Ack=1) u rozšířených poruch. Tento stav obvykle blokuje regulaci a je nutno jej odkvitovat.

---

## Generuj půlnoc v Scheduleru

---

Modul u regulátorů Desigo 4 přidává půlnoční záznam (kopie předchozího posledního záznamu) tak, ať se Scheduler chová jako dosud a nevypíná o půlnoci regulaci pokud chybí půlnoční záznam.

---

## Manuální režim v Scheduleru

---

Volba, zda má TSP dialog dovolit manuální ovládání.

---

## Čti výjimky ze Scheduleru

---

Povolí vyčítání Schedule Exceptions a zpřístupní je přes fieldy Except.XXXX v proměnné typu Schedule. Pro ovládání je nutná speciální ovládací obrazovka.

---

## Ukaž seznam poruch v CommonFault

---

Při zapnuté volbě ve fieldech CommonFault.Fault.List[10] a .Fault.Used[10] zpřístupní 10 aktivních alarmů. Seznam alarmů obdobně jako v PXM20 s možností filtrů v následných nastaveních.

Texty u alarmů, které modul najde v databázi vezme z databáze (popis proměnné + názvy 2 nadřazených skupin), ostatní vyčte z popisů objektů z regulátoru.

---

### Ukaž ComnAlarms v CommonFault

---

Implicitně vypnuto, pak nezobrazuje alarmy objektů typů ComnAlarm (protože by byly pravděpodobně hlášeny duplicitně).

### Ukaž definované alarmy v CommonFault

---

Implicitně vypnuto, pak nezobrazuje alarmy objektů které jsou definované v databázi a je nastaveno jejich vyhledávání (pomocí fieldu .Status).



## 7 Export do SQL databáze

### Export do SQL databáze

ProCop od verze 3.4 disponuje možností automatického exportu dat do SQL databází, XLS, XLSX, CSV a jiných formátů, dostupných přes ODBC ovladače Windows. Pro tyto exporty je v monitorovacím systému služba ProSQL, která v naplánovaných časech spouští definované skripty, které vytvářejí, doplňují, nebo modifikují databázi, nebo soubory. Tento návod popisuje definici exportu dat Microsoft SQL databáze.

### Požadavky před realizací

Předpokládáme fungující, běžící monitorovací systém ProCop verze 3.4 a vyšší, který sbírá data z regulátorů, či měřičů spotřeb po libovolných komunikačních cestách. Tato data, nebo některá z nich je potřeba exportovat do SQL databáze, například pro nadřazené systémy jako jsou vyhodnocovací, fakturační či firemní informační systémy. Tyto systémy obvykle pracují nad SQL databází, do které bude potřebné v definovaném formátu dodávat data z monitorovacího systému. Může být také vytvořena specializovaná SQL databáze, která bude sloužit pouze pro výměnu dat mezi monitorovacím systémem ProCop a informačním systémem.

#### Co je potřeba před realizací exportu:

- **Funkční SQL databáze**
- **Název SQL databáze**, případně instance
- **IP adresa**, nebo **doménové jméno počítače**, na kterém SQL databáze běží
- **Uživatelské jméno a heslo** pro čtení i zápis do SQL databáze
- **Monitorovací systém ProCop v. 3.4 a vyšší**, sbírající data z technologie
- **Funkční síťové spojení mezi oběma počítači**, běží-li SQL databáze na jiném počítači
- **Struktura**, definice tabulek a formát dat, v jaké podobě mají být data ukládána

### Instalace Microsoft SQL Server 2008 R2

Instalaci databázového serveru, pokud nepoužíváme existující provedeme podle dokumentace výrobce. V případě použití Microsoft SQL Server 2008 R2 Express Edition je instalace velmi snadná. Stačí stáhnout (v současnosti ze stránek [www.microsoft.com/sqlserver/](http://www.microsoft.com/sqlserver/)) příslušný softwarový balík a nainstalovat do Windows Server 2008 R2, případně Windows 7 Professional. Podle současných licenčních podmínek je možné využívat databázový server ve verzi Express do 10GB dat zcela zdarma.

#### **Důležité upozornění!**

*Licence SQL databází nejsou součástí monitorovacího systému ProCop a řídí se licenčními podmínkami a pravidly konkrétního výrobce.*

#### Uveďme krátký přehled a stručné vysvětlení dále používaného názvosloví, které se při instalaci SQL serveru a při realizaci projektu ProSQL využívá:

- **SQL server (SW)** - softwarový balík, který umožňuje zpracovávat a uchovávat data, přístupný přes SQL (Structured Query Language = strukturovaný dotazovací jazyk), někdy se také hovoří o SQL databázi
- **SQL server (HW)** - počítač, na kterém je provozována SQL databáze
- **Instance** (SQL serveru) - na jednom počítači může být nainstalováno více SQL serverů, tedy více instancí tohoto softwaru, rozlišují se jménem instance
- **Databáze** - v SQL serveru může být více databází, nezávislých úložišť dat, každá s vlastní strukturou a oprávněními

- **Tabulka** - databáze se mimo jiné skládá z tabulek, které mají definovány sloupce a v řádcích se pak ukládají data
- **Sloupec** - vertikální část tabulky, definovaná jménem a datovým typem (případně dalšími parametry), jméno sloupce musí být v rámci tabulky unikátní, nikoliv v rámci databáze; v řádcích se pak v daném sloupci ukládají data definovaného jména a typu
- **Řádek** - horizontální část tabulky, kde jsou ukládána data všech, nebo některých sloupců
- **Buňka** - průnik řádku a sloupce tabulky, hodnota dat daného řádku sloupce
- **Index** - podpůrný mechanismus pro rychlé třídění a vyhledávání v tabulce, obvykle pro daný sloupec
- **Primární index** - index sloupce, kde všechna data musí být vyplněna a musí být unikátní (např. pořadové číslo)
- **Složený index** - index, který je vytvořen nad více sloupci
- **Struktura databáze** - definice struktur všech tabulek, jejich jmen a struktur v dané databázi
- **Struktura tabulky** - definice všech sloupců tabulky, jejich jmen, datových typů a dalších parametrů, indexů a vazeb a omezení
- **Struktura dat** - definice obsahu řádků, jejich identifikace

### ★ Tip!

*Pro správu Microsoft SQL databází je dodáván produkt SQL Server Management Studio, což je velice silný nástroj pro veškerou správu databází a to i vzdáleně.*

## Struktura databáze

Ve stručnosti uvedme, že se jedná o definici tabulek, jejich sloupců, indexů, formátů dat, časů a period ukládání ve kterém mají být data z monitorovacího systému ProCop exportována. Samotnou strukturu tabulek bude nutné nadefinovat v monitorovacím systému ProCop, který ji sám vytvoří v SQL databázi po spuštění exportních úloh. Je potřeba zvážit, zda je vhodné data ukládat do jedné, či více tabulek, zda mají být data v SQL přepisována, nebo přidávána, také množství dat, které bude zpracováváno a tedy i zatížení monitorovacího systému, SQL databáze a její velikost v dlouhodobém měřítku.

### Jaká data je potřeba exportovat do SQL databáze:

- Veškerá nasbíraná data (pozor na obrovský objem dat)
- Jen některá data jako:
  - Teploty a tlaky systémů
  - Provozní hodiny zařízení
  - Hlavní hodnoty měřičů spotřeb

### V jakém formátu mají být data do SQL zapisována a jak často:

- Jedna, nebo více tabulek, například pro každý modul zvlášť
- Struktura tabulek
  - Názvy a SQL typy sloupců
  - Primární indexy
  - Obsah sloupců vkládaných exportem, unikátnost identifikátorů a primárního indexu
- Perioda zápisu dat do SQL, přepisování dat v součinnosti s primárními indexy



### Poznámka:

*Návrh struktury SQL databáze bývá primárním problémem při zprovoznění výměny dat mezi různými informačními systémy. Na návrhu by se měly účastnit společně všechny dotčené strany, které budou data zpracovávat. Při nevhodném návrhu databáze může v dlouhodobém měřítku docházet k velké zátěži SQL databáze, k pomalému přístupu k datům a může být nasbíran obrovský objem špatně zpracovatelných dat.*

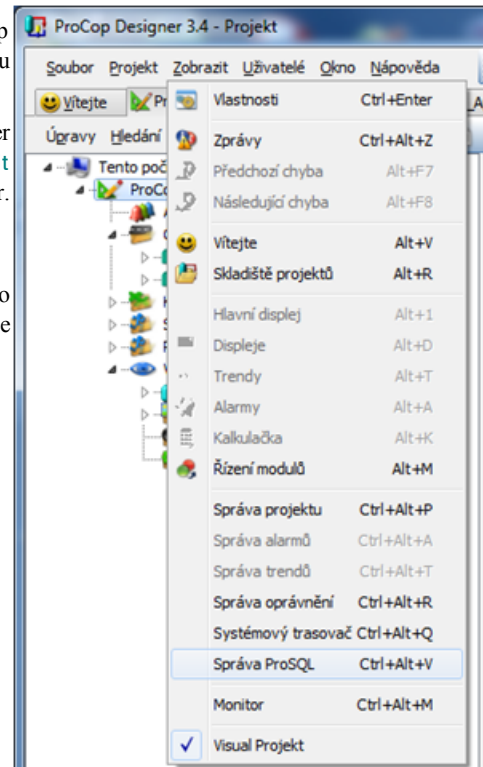
## 7.1 Konzola správy ProSQL

### Konzola správy ProSQL projektu

Samotná realizace exportu je v monitorovacím systému ProCop realizována službou ProSQL, která je součástí monitorovacího systému ProCop od verze 3.4.

Správa projektu ProSQL je součástí návrhové části Designer monitorovacího systému ProCop. Otevřeme ji volbou **Zobrazit \ Správa ProSQL**, nebo kombinací **Ctrl+Alt+V** v režimu designer. Prostředí konzoly je obdobou základního monitorovacího projektu datového serveru, stejně jako jeho obsluha.

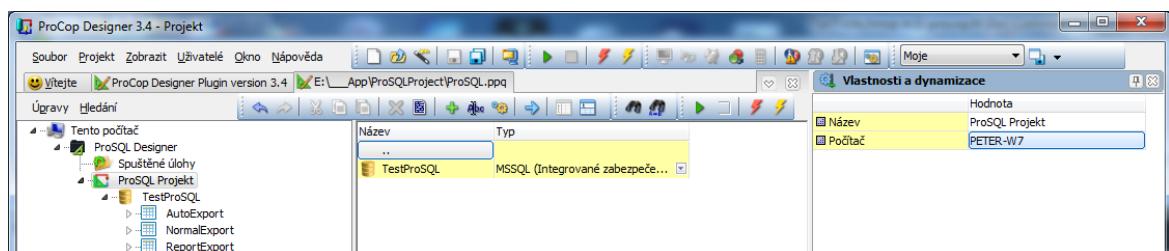
Počítač, na kterém služba ProSQL běží (téměř vždy stejný, jako monitorovací, nejedná se o počítač SQL serveru) a projektu exportu je nastavitelné v designeru.



*Spuštění správy ProSQL*

### Projekt ProSQL

Projektem ProSQL je soubor, obvykle s příponou ".ppq", ve kterém jsou uloženy veškeré potřebné informace k připojení k monitorovacímu systému (datovému serveru ProDat), k SQL databázi, včetně definice struktur tabulek a jejich exportovaných dat. Soubor je možné uložit kdekoliv, nejlépe však do adresáře monitorovacího projektu. Implicitně je projekt ProSQL umístěn v adresáři projektu.



*Správa ProSQL s otevřeným projektem a zobrazenými vlastnostmi připojení k datovému serveru*

Práce s projektem v ProSQL je obdobná, jako s monitorovacím projektem. V levé části okna je zobrazen strom projektu, v pravé tabulkové zobrazení obsahu větve. Projekt lze rozčlenit na více částí, kdyby bylo potřeba data získávat z více datových serverů. Obvykle však stačí vložit jediný projekt a nastavit jeho parametry, viz následující podkapitola. Vlastnosti dané větve či řádku je možné zobrazit pomocí kláves **F4** a **Ctrl+Enter**, v nabídce pak volbou **Zobrazit/Vlastnosti**.

### Parametry připojení k datovému serveru

Pro připojení k datovému serveru je potřeba zadat ve vlastnostech **F4** pouze jméno počítače, na kterém běží datový server monitorovacího systému ProDat. Pro lokální počítač ponecháme jméno prázdné. Oprávnění k přístupu je řešeno na úrovni DCOM práv operačního systému Windows.

Dále je možno zadat jméno projektu, které bude zobrazeno ve stromu ProSQL.

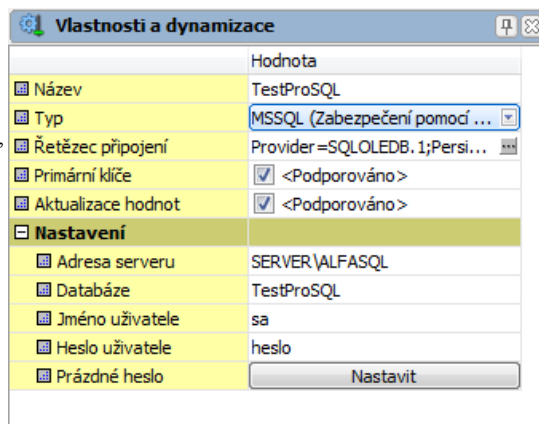
## Parametry připojení k SQL databázi

V projektu je možno přidat SQL databáze, do kterých budou data exportována například klávesou **Insert**. Zrušit připojení je možné např. klávesou **Delete**. Obě operace jsou dostupné i v nástrojové liště.

Parametry připojení k SQL serveru lze definovat ve vlastnostech ( **F4** a **Ctrl+Enter** ).

### Nastavení parametry připojení k SQL databázi:

- **Název** - pojmenování databáze, implicitně shodně se jménem databáze
- **Typ** - výběr typu databáze, Microsoft SQL, MySQL, typ souboru a podobně
- **Řetězec připojení** - konfigurace připojení databáze, je možné jej vytvořit pomocí systémového dialogu, který se otevře tlačítkem **...** u řádku
- **Primární klíče** - definuje, zda databáze podporuje primární klíče (indexaci), soubory indexaci obvykle nepodporují, předvyplní se při přepnutí typu databáze
- **Aktualizace hodnot** - definuje, zda databáze podporuje aktualizaci (přepisování) hodnot, soubory aktualizaci obvykle nepodporují, předvyplní se při přepnutí typu databáze
- **Adresa serveru** - IP adresa, nebo doménové jméno SQL serveru, instance - definuje a sdělí správce SQL databáze
- **Jméno databáze** - jméno SQL databáze - vytvoří a sdělí správce SQL databáze
- **Jméno uživatele** - uživatel SQL databáze s oprávněním pro čtení i zápis - definuje a sdělí správce SQL databáze
- **Heslo uživatele** - heslo uživatele SQL databáze - definuje a sdělí správce SQL databáze



Vlastnosti databáze - připojení k MS SQL databázi



### Poznámka:

Parametry se mohou dle zvoleného typu databáze lišit, jednotlivé parametry je možné zadat i v řetězci připojení, který lze upravit pomocí systémového dialogu.

## 7.2 Definice tabulek

### Definice exportních tabulek

V projektu ProSQL je možné nadefinovat strukturu SQL databáze, jednotlivých tabulek a jejich sloupců. Tyto tabulky jsou zároveň i naplánovanými úlohami, které se dle svých vlastností spouštějí a zapisují data dle potřeby do SQL databáze.

### Vytvoření tabulky - úlohy

Ve stromu v definované databázi je možné vytvořit jednotlivé tabulky se strukturou a vkládanými daty - řádky. Tato tabulka je zároveň naplánovaná úloha, která dle vlastností umožňuje automaticky exportovat data do databáze. Spuštěné exportní úlohy a jejich průběh se zobrazuje ve větvi **Spuštěné úlohy**, je-li projekt spuštěn a naplánovaná úloha běží.


Přidání tabulky možno provést například klávesou **Insert**, zrušit tabulku např. klávesou **Delete**. Obě operace jsou dostupné i v nástrojové liště. Zobrazí se výběr typu tabulky.

### Typy exportních tabulek - úloh:

- **Automatická** - zadáním indexu modulu budou exportována automaticky všechna dostupná aktuální data daného modulu projektu monitorovacího systému (vždy jedna proměnná na řádek), dle definice sloupců tabulky v naplánovaných časech
- **Normal** - lze vytvořit kompletní seznam proměnných modulu, který je možné dále upravovat a promazat, vybraná aktuální data daného modulu projektu monitorovacího systému (vždy jedna proměnná na řádek), dle definice sloupců tabulky v naplánovaných časech
- **Report** - uživatelsky ručně definovaná tabulka s různými proměnnými v řádku, vhodná například pro export dat jednoho modulu v jediném řádku (měřiče spotřeb, provozní hodiny, teploty a tlaky, atp.)

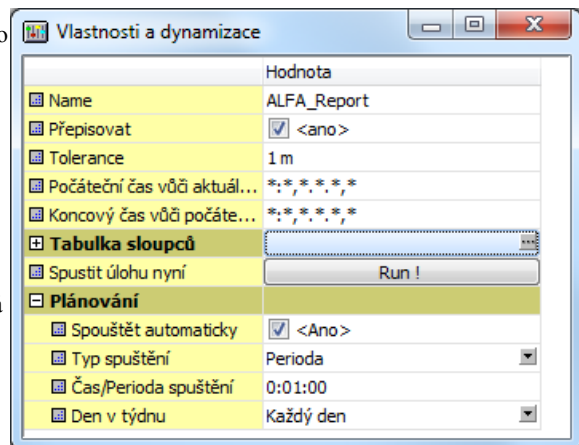


## Vlastnosti tabulky - plánování úlohy

Tabulce lze měnit vlastnosti (  a **Ctrl+Enter**) exportu do SQL databáze a naplánování.

### Vlastnosti tabulky:

- **Jméno** - pevné jméno tabulky v SQL databázi
- **Prefix modulu** - prefix modulu monitorovacího projektu (pouze u typu Auto) - definuje modul, jehož data budou kompletně automaticky exportována
- **Přepisovat** - definuje, zda budou již dříve zapsaná data, která byla nalezena dle primárního indexu, opakovaně přepisována v SQL databázi - obraz modulu, tabulka se nebude v čase zvětšovat
- **Pouze změny** - řádek se zapisuje pouze při změně hlavní hodnoty proměnné, nebo její validitě - pouze u tabulek typu Auto a Normal
- **Počáteční čas vůči aktuálnímu** - jen u tabulky typu Report, podrobněji bude rozebráno dále u tabulky sloupců
- **Koncový čas vůči počátečnímu** - jen u tabulky typu Report, podrobněji bude rozebráno dále u tabulky sloupců
- **Tabulka sloupců** - definuje strukturu tabulky v SQL databázi - podrobněji bude rozebráno dále



Vlastnosti tabulky - plánování úlohy

### Plánování úlohy:


- **Spustit úlohu nyní** - umožní uživatelské okamžité spuštění úlohy - pro vývoj a testování
- **Spouštět úlohu automaticky** - povoluje automatické spuštění dle plánu
- **Typ spuštění** - definuje, zda je úloha spuštěna jen v definovaný čas, nebo pravidelně s níže definovanou periodou
- **Čas/Perioda spuštění** - definuje, nebo jak často, má být úloha automaticky spuštěna
- **Den v týdnu** - definuje, které dny v týdnu má být úloha automaticky spuštěna

## 7.2.1 Definice struktury tabulky

### Definice struktury tabulky

Tabulka v SQL serveru musí mít definovanou pevnou datovou strukturu, tedy jednotlivé sloupce, jejich názvy a typy, primární index. Dále je nutné definovat, jaká data budou z monitorovacího systému exportována do jakých sloupců SQL tabulky. Strukturu a typy dat definujeme ve vlastnostech tabulky v tabulce sloupců.

### Vlastnosti tabulky

Ve vlastnostech tabulky (  a **Ctrl+Enter**) je nutné nejprve nastavit strukturu tabulky v SQL databázi. Dle této struktury budou data exportována do SQL databáze.

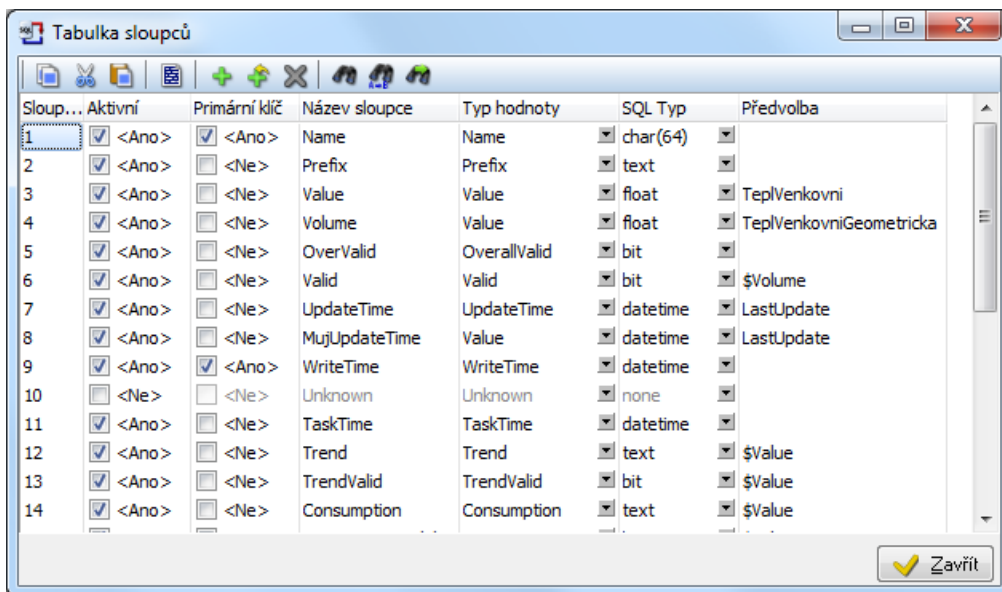
#### **Důležité upozornění!**

*Pokud je provedena významnější změna ve struktuře tabulky a tabulka je již v SQL databázi vytvořena, dojde k jejímu zrušení včetně exportovaných dat! Pro restrukturalizaci existující tabulky je nutné použít nástroje SQL serveru, aby bylo zabráněno ztrátě dat! Druhou možností je změnit jméno tabulky, data s novou strukturou budou ukládána do nové tabulky.*

### Tabulka sloupců

V tabulce sloupců reprezentuje jeden řádek právě jeden sloupec v definici řádků tabulky. Ne všechny definované sloupce jsou exportovány do SQL databáze, záleží to na jejich typu. Vytváření, kopírování a mazání sloupců (tedy řádků) lze provádět standardně v nástrojové liště, lokální nabídce, nebo klávesami **Insert** a **Delete**. Úpravy je možné provádět

myší, přímým psaním, nebo zahájit úpravy klávesou **F2**.



*Definice sloupců tabulky*

Změnu pořadí řádků (sloupců budoucí SQL tabulky) lze provádět pouze změnou pořadového čísla sloupce. Tabulka se okamžitě znovu uspořádá podle zadaného pořadí.

**Každý řádek v tabulce (SQL sloupec) má následující vlastnosti:**

- **Sloupec** - pořadové číslo sloupce - slouží jen pro přesun v tabulce sloupců, který se provede ihned po zadání číselné hodnoty pořadí
- **Aktivní** - definuje, zda je sloupec exportován do SQL databáze - pokud není aktivní, nebude dále zpracováván a kontrolován
- **Primární index** - definuje, zda daný sloupec je součástí primárního indexu SQL tabulky, každá SQL tabulka musí mít alespoň jeden sloupec (nebo kombinaci sloupců) unikátní pro jednoznačné vyhledání, prioritizace sloupců při složeném primárním indexu je definována pořadím
- **Unikátní** - zapíná kontrolu definice sloupce, zda musí být unikátní, kontrola se provádí pouze při překladu projektu
- **Název sloupce** - jméno sloupce v SQL databázi (nesmí obvykle obsahovat jiné, než alfanumerické znaky, bez mezer, dle implementace SQL databáze)
- **Typ hodnoty** - typ hodnoty, která bude z monitorovacího systému ProCop exportována
- **SQL Typ** - SQL datový typ sloupce - data z monitorovacího systému budou konvertována a uložena v tomto typu do tabulky
- **Předvolba** - pokud v definici řádků nebude zadána žádná hodnota, bude použita tato předvolená

## 7.2.2 Typy hodnot pro export

Typy hodnot pro export

Pro definici sloupce je potřeba zadat způsob, jakým bude zadání buňky v řádku interpretováno při exportu do SQL. Tento způsob je potřeba vybrat ve sloupci **Typ hodnot** u každého sloupce při definici tabulky sloupců.

### Typy hodnot pro export z monitorovacího systému ProCop

Jednotlivé sloupce v SQL databázi musí mít nastaven typ hodnoty, který je exportován z monitorovacího systému.

**Typy hodnot exportovaných z monitorovacího systému:**

- **Unknown** - neznámý, není definován, sloupec se neexportuje do SQL databáze
- **Value** - hlavní hodnota v řádku zadané proměnné
- **Valid** - validita v řádku zadané proměnné - znamená, že proměnná byla obcerstvena v intervalu zadaném v monitorovacím projektu
- **OverallValid** - logický součin všech validit proměnných, zadaných v daném řádku (je-li kterákoliv z hodnot v řádku nevalidní, bude nevalidní)

- **UpdateTime** - příznak poslední obnovy hodnoty proměnné v mon. systému (.LastUpdate)
- **TaskTime** - čas spuštění exportní úlohy
- **WriteTime** - čas zápisu řádku do SQL databáze
- **Prefix** - prefix všech použitých proměnných v řádku (sloupec není exportován do SQL databáze, slouží jen aby nebylo nutné definovat stejné prefixy u všech sloupců v daném řádku, bývá zobrazen jako znak @)
- **Trend** - hodnota z trendu proměnné v čase definovaném v parametru úlohy
- **TrendValid** - validita z trendu proměnné v čase definovaném v parametru úlohy
- **Consumption** - rozdíl hodnot trendů v časech definovaném v parametrech úlohy
- **ConsumptionValid** - validita hodnot trendů v časech definovaném v parametrech úlohy
- **TrendBeginTime** - počáteční čas definovaný v parametru úlohy
- **TrendEndTime** - koncový čas definovaný v parametru úlohy
- **UserData** - uživatelská data - libovolná data (text) zapsaná v řádku tabulky exportovaná do SQL databáze, nebere se z monitorovacího systému
- **Time** - definovaná proměnná převedená na čas (datum se nepoužije)
- **Date** - definovaná proměnná převedená na datum (čas je nulový)
- **DateOrNow** - definovaná proměnná převedená na datum (čas je nulový), pokud je datum nedefinováno, zapíše se aktuální datum systému
- **DateTime** - definovaná proměnná převedená na datum a čas
- **DateTimeOrNow** - definovaná proměnná převedená na datum a čas, pokud je datum a čas nedefinováno, zapíše se aktuální datum a čas systému
- **Note** - poznámka - libovolná data (text), zaznamenaný jen v projektu exportu, sloupec se neexportuje do SQL databáze

### 7.2.3 Datové typy sloupců v SQL

Každý sloupec tabulky v SQL databázi musí mít definován typ záznamu (datové reprezentace) hodnot. Tento typ je potřeba vybrat ve sloupci **SQL Typ** u každého sloupce při definici tabulky sloupců.

#### Datové typy sloupců v SQL databázi

Data v jednotlivých sloupcích získaná z monitorovacího systému dle sloupce "Typ hodnot", nebo definovaná v exportním projektu, jsou zaznamenávána v SQL databázi do sloupce zvoleného typu. Data jsou konvertována, je-li to možné do zvoleného SQL datového typu.

##### Typy sloupců v SQL databázi:

- **none** - není definován, data se neexportují do SQL databáze
- **bit** - logický typ, hodnoty 0 a 1 (false, true)
- **tinyint** - 1 bajt, bez znaménka 0-255
- **smallint** - 2 bajty, se znaménkem
- **int** - 4 bajty se znaménkem
- **bigint** - 8 bajtů se znaménkem
- **float** - 4 nebo 8 bajtů dle velikosti, reálné číslo s plovoucí desetinnou čárkou
- **real** - 4 bajty, reálné číslo s plovoucí desetinnou čárkou
- **text** - text libovolné délky
- **char(64)** - text maximálně 64 znaků dlouhý
- **datetime** - datum a čas

##### **Důležité upozornění!**

*Datové typy v SQL databázi se mohou lišit dle konkrétní implementace SQL serveru.*

## 7.3 Report tabulka

### Report Tabulka

V tabulka typu Report je především určena pro export dat z více modulů současně, kde modul má přiřazen jeden či

několik řádků a proměnné daného modulu jsou organizovány ve sloupcích. Z tohoto důvodu byl do definice sloupců tabulky doplněn typ sloupce "Prefix, který se neexportuje do SQL databáze, ale slouží jen jako prefix proměnných definovaných na daném řádku. Není tedy nutné vypisovat do jednotlivých buněk celé jméno proměnné, stačí zadat jméno proměnné bez prefixu a ve sloupci prefix zadat konkrétní prefix řádku. Při exportu je prefix z daného řádku ke všem proměnným na řádku automaticky doplněn. Existují i další metody, jak si ušetřit práci s opisováním stejných textů, ty budou popsány dále.

## Názvy sloupců v definici řádků tabulky

Ve stromu exportního projektu se v tabulce typu Report nachází větve řádky. Ta obsahuje předpis všech řádků, které se pokusí systém exportovat při každém spuštění úlohy. V záhlaví řádků jsou zobrazeny sloupce tak, jak jsou definovány v tabulce sloupců dané SQL tabulky. Jejich názvy jsou však v některých případech označeny znakem '\*' a někdy různě uvozeny.

*Name	/Prefix=@/	Value	Volume	<OverValid>	Valid
IALB1	IALB	<@.TeplVenkovni>	<@.TeplVenkovniGeo...	<>	\$@.TeplVenkovni\$
\$IALB\$	IALB	\$@.TeplVenkovniGeo...	<@.TeplVenkovniGeo...	<>	<\$@.TeplVenkovniGeo...
IALB	\$IALB\$	\$	<@.TeplVenkovniGeo...	<>	<\$@.TeplVenkovniGeo...

UpdateTime	MujUpdateTime	<*WriteTime>	</Unknown/>	<TaskTime>	Trend	TrendValid
<@.LastUpdate>	<@.LastUpdate>	<>	<>	<>	<\$@.Tepl...	<\$@.TeplVenkovni\$
<@.LastUpdate>	<@.LastUpdate>	<>	<>	<>	<\$@.Tepl...	<\$@.TeplVenkovniGe...
<@.LastUpdate>	<@.LastUpdate>	<>	<>	<>	<\$\$\$>	<\$\$\$>

Ukázka definice řádků tabulky typu Report s využitím předvoleb a odkazů

Doplňky k názvu sloupce mají přispět ke snadnější orientaci v tabulce a jsou tyto:

- '\*' - označuje sloupce, které jsou součástí primárního indexu, priorita sloupců je určena pořadím definice v tabulce sloupců
- /Název sloupce/ - takto označený sloupec nebude exportován do SQL databáze, slouží jen k účelům exportu (prefix, poznámka, neznámý typ hodnot)
- /Název sloupce=@/ - Název sloupce=@ poukazuje na to, že sloupec je typu prefix a zobrazení znaku '@' v buňkách znamená, že zde bude automaticky při exportu doplněn obsah sloupce typu Prefix v daném řádku
- <Název sloupce> - uvození jména sloupce do špičatých závorek poukazuje na to, že hodnoty jsou vždy předvolené a do sloupce se při definici řádků nic nezadá (sloupec typu WriteTime - čas zápisu řádku do SQL, TaskTime - čas spuštění úlohy, atp.)

## Vkládání, kopírování a mazání definic řádků

Jednotlivé řádky lze do tabulky přidávat **Insert**, v tabulce posouvat **Alt+Up** **Alt+Down**, kopírovat **Ctrl+C** **Ctrl+V** i mazat **Delete**. Operace jsou rovněž dostupné přes hlavní i lokální nabídku a nástrojovou lištu. Nechtěně provedené operace je možné vrátit zpět **Ctrl+Z**.

Jednotlivé sloupce v řádku je potřeba vyplnit požadovanou hodnotou dle typu sloupce, která bude sloužit k exportu dat do SQL databáze. Pokud je v definici sloupců zadán i sloupec typu Prefix, není potřeba ve sloupcích, které vyžadují zadání proměnné, prefix zadávat.

## Předvolené hodnoty v definici řádků

V definici sloupců tabulky je možné zadat ve sloupci Předvolba libovolný text. Pokud definujeme nový řádek a ponecháme některé sloupce nevyplněné, použije se tato předvolba pro daný sloupec a zobrazí se ve špičatých závorkách, např. <@.TeplVenkovni>. Takto zobrazená hodnota v buňce říká, že daný sloupec nebyl zadán a byla použita předvolba z definice sloupců tabulky a znak '@' nahrazuje text ve sloupci typu Prefix daného řádku.

### ★ Tip!

Pokud budou exportována data z více modulů stejného typu, je vhodné definovat sloupec typu prefix, do kterého se budou uvádět jen prefixy konkrétních modulů pro daný řádek. Sloupce pak mohou být definovány jako jednotlivé proměnné modulu, jejichž jméno bude u jednotlivých modulů stejného typu pravděpodobně shodné. Proto je vhodné jméno proměnné zadat jako předvolbu při definici sloupců. Při vkládání definic řádků do tabulky tak bude potřeba zadat pouze prefix modulu, ostatní sloupce se doplní přednastavenou hodnotou.

## Odkazy mezi sloupci při definici řádků

Při definici řádků může nastat potřeba, aby se dva či více sloupců odkazovaly na stejnou proměnnou. Typický například při použití typu sloupců Value, Valid a UpdateTime, kde z jedné proměnné lze získat hodnotu, její validitu a čas, kdy byla naposledy obcerstvena. Takto definované tři sloupce vyžadují zadání u každého řádku totožně třikrát tutéž proměnnou.

Aby se předešlo zbytečným duplicitám při zadávání, byl zaveden odkaz na hodnotu jiného sloupce ve stejném řádku. Do buňky stačí zadat text `$Název sloupce` a při zadání či změně hodnoty v odkazovaném sloupci se změna automaticky promítne do sloupce s odkazem. Odkazované hodnoty jsou pak uvozeny znaky `:@.TepiVenkovni$`.

Odkaz na hodnotu jiného sloupce stejného řádku lze použít i v předvolbě. Takováto hodnota bude pro přehlednost zobrazena takto: `<$:@.TepiVenkovni$>`.

## Zadání prázdné proměnné

Při kontrole (kompilaci) se kontroluje existence všech proměnných a zadání všech sloupců. Pokud není proměnná, nebo hodnota v buňce zadána, je vyhlášena chyba, aby nedošlo omylem k opomenutí zadání. Pokud však nemá být hodnota daného sloupce a řádku do SQL skutečně exportována (nemá být například použita předvolba), je potřeba zadat namísto prázdné buňky znak '\$', nebo text '\$EMPTY'. Takto vyplněná buňka nebude kontrolována a do SQL databáze nebude zapisována.

### ★ Tip!

*Odkazy na sloupce, lze provádět při definici řádků i v předvolbách při definici sloupců. Pokud je však potřeba do konkrétní buňky zadat jinou proměnnou, než je předvolená, stačí její jméno zadat do buňky. Prefix modulu se doplní automaticky ze sloupce typu Prefix, pokud je zadán.*

## 7.4 Příklad exportu měřičů spotřeb

### Příklad exportu měřičů spotřeb

Typickým příkladem exportu dat do nadřazených informačních a fakturačních systémů přes SQL databázi jsou odečty měřičů spotřeb, jako jsou měřiče tepla, chladu, elektřiny, plynu, vodoměry a mnohé další. Následující příklad uvádí jednu z mnoha možností, jak poskytovat odečtené hodnoty z monitorovacího systému do SQL databáze. Z této databáze může informační systém třetí strany získávat potřebná data.

### Poznámka:

*Konkrétní definice struktury databáze záleží na domluvě s administrátory informačního či fakturačního systému a je velice variabilní.*

### Popis exportu měřičů spotřeb

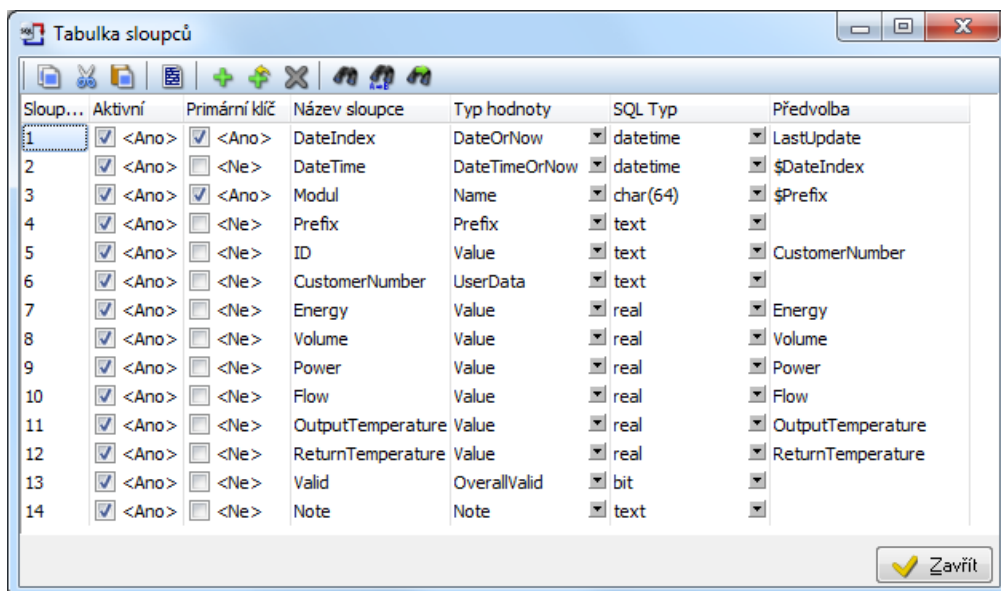
Měřiče spotřeb i různých výrobců a typů měří veličiny jako množství spotřebované energie, objemu média, okamžité veličiny, napětí, teploty a podobně. Pro náš příklad se zaměříme na měřiče tepla (chladu) a vodoměry. Zdefinujeme strukturu databáze, která sloučí rozdílné vlastnosti různých měřičů do jediné tabulky. Tato tabulka - úloha, která jí vyplňuje pak může být periodicky naplánována, například každý den v 6 hodin ráno.

Pro strukturu tabulky tak budeme potřebovat znát exportované veličiny, identifikátor měřiče spotřeb a datum exportovaných hodnot.

V novém projektu si nejprve vytvoříme novou databázi, zadáme jméno počítače SQL serveru, instanci a přihlašovací údaje do databáze. Vyzkoušíme funkčnost připojení v lokální nabídce ve stromu na větví databáze. Zobrazí se jednoduchý prohlížeč SQL databáze, případně chybové hlášení s popisem chyby.

### Struktura tabulky

Do databáze si přidáme definici tabulky typu Report, které zadáme jméno, zdefinujeme čas spuštění na 6 hodin každý den a otevřeme Tabulku sloupců.



Ukázka definice struktury SQL tabulky, jednotlivých sloupců

Vytvoříme jednotlivé sloupce tabulky (zobrazují se jako řádky) a zadefinujeme jim parametry dle následujícího obrázku.

#### Strukturu tabulky a význam sloupců se pokusíme rozebrat důkladněji, sloupec po sloupci:

- **DateIndex** - sloupec je součástí primárního indexu, do kterého se запиše datum, zadané proměnné, předvolba je LastUpdate, tedy poslední obnova proměnné (modulu)
- **DateTime** - datum a čas poslední obnova proměnné (modulu), předvolba je definována jako odkaz na hodnotu ve sloupci DateIndex
- **Modul** - libovolný text jednoznačně identifikující konkrétní měřič tepla - je součástí primárního indexu - předvolba je reference na sloupec prefix
- **Prefix** - sloupec, který nebude v SQL databázi a uvede se do něj prefix příslušného modulu pro ostatní proměnné v řádku, bez předvolby, musí se vždy zadat
- **ID** - hodnota z měřiče - sériové číslo - předvolba proměnná CustomerNumber
- **CustomerNumber** - hodnota zadaná ručně při exportu - kontrolní zákaznické číslo měřičiho místa
- **Energy** - hodnota z měřiče - kumulovaná energie - předvolba: proměnná Energy
- **Volume** - hodnota z měřiče - kumulovaný objem - předvolba: proměnná Volume
- **Power** - hodnota z měřiče - okamžitý výkon - předvolba: proměnná Power
- **Flow** - hodnota z měřiče - okamžitý průtok - předvolba: proměnná Flow
- **OutputTemperature** - hodnota z měřiče - teplota přívodu - předvolba: proměnná OutputTemperature
- **ReturnTemperature** - hodnota z měřiče - teplota zpátečka - předvolba: proměnná ReturnTemperature
- **OverAllValid** - validita všech hodnot z měřiče - vypočtená hodnota ze všech validit hodnot proměnných na řádku
- **Note** - pouze poznámka u každého řádku v definici řádků, neexportuje se do SQL databáze

K primárnímu indexu uvedeme, že se skládá ze dvou sloupců - DateIndex a Modul. Při spuštění úlohy se projdou všechny řádky, vyhodnotí se sloupce dle definice typů a zapíše se do SQL databáze. Takto navržený primární index nám zaručí, že pro každý měřič se k danému datu zapíše jen jediný řádek. Pokud jsou již data k danému datu pro daný modul v databázi zaznamenána, záleží pak na příznaku "Přepisovat" v parametrech úlohy, zda se řádek přepíše novějšími daty, či nikoliv. V případě odečtů měřičů není důvod při vícenásobném spuštění úlohy ve stejném dni data přepisovat.

#### Definice řádků tabulky - měřičů

Máme-li definovanou strukturu databáze, můžeme přejít k definici řádků - měřičů. Každý řádek nám bude reprezentovat jedním měřič, sloupce pak hodnoty z konkrétního měřiče. Následující obrázek ukazuje zadání řádků - měřičů.

*DateIndex	DateTime	*Modul	/Prefix=@/	ID	CustomerNumber	Energy
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$DK10M124\$>	DK10M124	<@.CustomerNumber>	998124	<@.Energy>
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$IOS967M024\$>	IOS967M024	<@.CustomerNumber>	998824	<@.Energy>
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$OS616M068\$>	OS616M068	@.MeasureId	9686	<@.Energy>
<@.LastUpdate>	<\$@.LastUpdate\$>	OS616M068A	OS616M068	<@.CustomerNumber>	6825100	\$
<@.LastUpdate>	<\$@.LastUpdate\$>	OS616M068B	OS616M068	<@.CustomerNumber>	7982395	\$
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$DK5BV110\$>	DK5BV110	<@.CustomerNumber>	9960327	\$
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$IOS967M002\$>	IOS967M002	<@.CustomerNumber>	998821	<@.Energy>
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$IOS967M031\$>	IOS967M031	<@.CustomerNumber>	998631	<@.Energy>
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$IOS967M027\$>	IOS967M027	<@.CustomerNumber>	998627	<@.Energy>
<@.LastUpdate>	<\$@.LastUpdate\$>	<\$IOS967M076\$>	IOS967M076	<@.CustomerNumber>	998626	<@.Energy>

Volume	Power	Flow	OutputTemperature	ReturnTemperature	<Valid>	/Note/
<@.Volume>	<@.Power>	<@.Flow>	<@.OutputTemperature>	<@.ReturnTemperature>	<>	1.
<@.Volume>	<@.Power>	<@.Flow>	<@.OutputTemperature>	<@.ReturnTemperature>	<>	2.
<@.Volume>	<@.Power>	<@.Flow>	@.InputTemperature	@.OutputTemperature	<>	3.
@.A	\$	\$	\$	\$	<>	4.
@.B	\$	\$	\$	\$	<>	5.
@.Inputs	\$	\$	\$	\$	<>	6.
<@.Volume>	<@.Power>	<@.Flow>	<@.OutputTemperature>	<@.ReturnTemperature>	<>	7.
<@.Volume>	<@.Power>	<@.Flow>	<@.OutputTemperature>	<@.ReturnTemperature>	<>	<>
<@.Volume>	<@.Power>	<@.Flow>	<@.OutputTemperature>	<@.ReturnTemperature>	<>	<>
<@.Volume>	<@.Power>	<@.Flow>	<@.OutputTemperature>	<@.ReturnTemperature>	<>	<>

Ukázka definice exportu měřičů do SQL databáze, definice jednotlivých řádků - měřičů

Ve sloupci poznámka **Note** jsou záměrně zapsány čísla řádků pro snazší popis. V tomto sloupci může být uvedeno cokoliv, dále se nezpracovává a do SQL databáze se neexportuje.

V prvním a druhém řádku pro měřič DK10M124 a IOS967M04 stačilo zadat do sloupce prefix uvedené prefixy modulů a kontrolní zákaznické číslo. Jak je ze špičatých závorek a referencí (\$) patrné, ostatní sloupce nebylo potřeba vyplnit, byly použity předvolby.

Na třetím řádku je uveden měřič tepla, který má jiný název proměnné pro výrobní číslo. Proto bylo nutné zadat ještě MeasureId do sloupce **ID**. Dále má rozdílné názvy teplot, proto i tyto bylo potřeba zadat ručně. Ostatní hodnoty jsou nevyplněné (z předvoleb).

Na čtvrtém a pátém řádku jsou ve skutečnosti údaje z měřiče, který je uveden na třetím řádku. Na jeho dva impulsní vstupy A a B jsou připojeny vodoměry. Tyto jsou však měřidla s vlastní identifikací, proto musí být na různých řádcích. Měří pouze objem. Ostatní veličiny nejsou zadány. Při kontrole by však vznikla chyba, že proměnná nebyla zadána. Proto na místo prázdných proměnných musíme uvést znak '\$', nebo text '\$EMPTY'. Všechny tři řádky mají shodný prefix, ale modul musí být unikátní z důvodu primárního indexu. Proto do sloupce Modul je v tomto případě zapsán název modulu s rozšířením a písmeno impulsního vstupu.

Na šestém řádku je umístěn nezávislý měřič s impulsním vstupem a připojeným vodoměrem. Hodnota objemu je v proměnné Inputs, nikoliv Volume, proto je nutné jméno proměnné explicitně do buňky zadat. Ostatní proměnné mají zůstat prázdné, proto je použit znak '\$'.

Od sedmého řádku jsou zase uvedeny standardní měřiče tepla, kde není potřeba zadat nic jiného, než prefix modulu a zákaznické číslo měřiče, jako u prvních dvou.

## Spuštění a kontrola výsledku

Naplánovanou úlohu spustíme. Ta otevře databázi a začne dle definice procházet každý řádek - měřič, jeden po druhém. Hodnoty proměnných získá z monitorovacího systému ProCop dle jména v buňce, předvolené hodnoty či odkazu a prefixu v daném řádku. Pokusí se dle primárního indexu nalézt řádek v databázi se stejným datem a názvem modulu. Pokud jej nenajde, celý řádek do databáze přidá, pokud ano, dle příznaku "**Přepisovat**" zadaném ve vlastnostech úlohy řádek přepíše, nebo vynechá a jde zpracovávat další řádek v definici. Takto postupuje, až projde všechny definované řádky - měřiče.

Průběh exportu můžeme sledovat ve stromu projektu ve větvi "**Spuštěné úlohy**". Až je úloha dokončena, je možné zkontrolovat data v SQL databázi, například jednoduchým prohlížečem, který lze otevřít volbou z lokální nabídky databáze ve stromu projektu.

### **Důležité upozornění!**

Případné chyby vzniklé při běhu úlohy se zapisují do systémového zápisníku Windows, do sekce AlfaLog pod názvem

*aplikace ProSQL.*



## 8 Provoz v doméně bez lokálního uživ.

Komponenty systému ProCop jsou standardně provozovány pod identitou lokálního uživatele ProCop, který je obvykle lokálním administrátorem počítače. Oprávnění tohoto uživatele mohou být snížena až na úroveň skupiny Users (popis konfiguračních kroků není součástí tohoto dokumentu). V případě restriktivních doménových politik je v některých případech požadován provoz bez tohoto uživatele. Změnu konfigurace je nutno v tomto případě provést ručně.

### Podmínky pro úspěšnou konfiguraci

- pro změnu konfigurací jsou nutná administrátorská oprávnění k počítačům
- je provedena instalace systému ProCop typu "Standardní monitorování" na počítači dále označovaném jako server
- volitelně instalace systému ProCop typu "Síťové pracoviště" na počítači označovaném dále jako klient
- na obou počítačích provoz pod stejným nebo různým účtem s oprávněními Domain User
- instalován systém ProCop verze 3.3.8.6 a novější na všech strojích

### Podporované kombinace operačních systémů

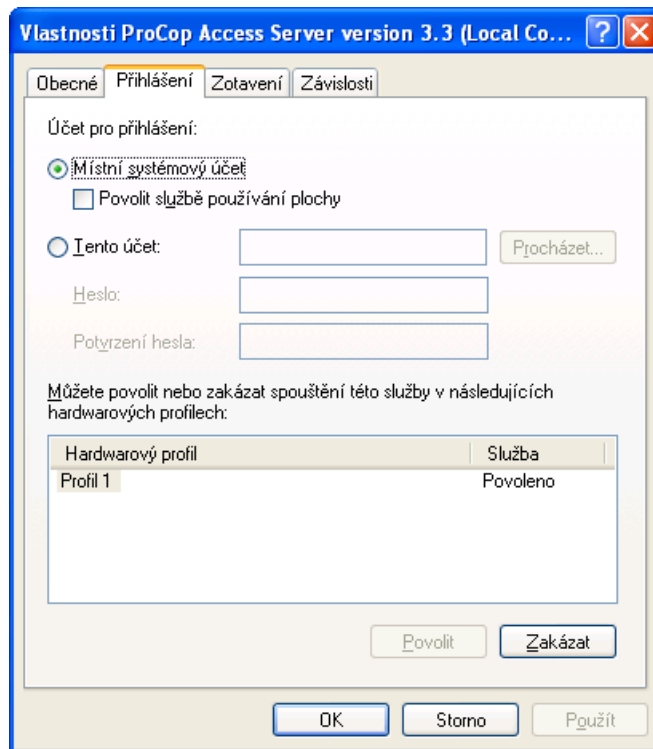
Operační systémy	Varianta 1	Varianta 2	Varianta 3
<b>Server</b>	Windows XP Pro SP 3	Windows 7 Pro a novější	Windows 7 Pro a novější
<b>Klient</b>	Windows XP Pro SP 3	Windows XP Pro SP 3	Windows 7 Pro a novější

## 8.1 Úpravy serveru pro provoz bez uživatele ProCop

### Změna identity služeb systému ProCop

Identita služeb je standardně nastavena na lokálního uživatele ProCop. Pro změnu je nutno kliknout **Start Menu/ Tento počítač** pravým tlačítkem myši volba **Spravovat**. V konzole Správa počítače v položce Služby a aplikace otevřete Služby. Na pravé straně v seznamu služeb najdete ProCop Access Server a volbou v menu **Akce/ Zastavit** zastavíte všechny služby systému ProCop. Volbou **Vlastnosti** pravým tlačítkem myši na kartě **Přihlášení** změňte účet pro přihlášení na Místní systémový účet.

Nastavení identity služeb:



Nastavení je nutné provést pro všechny služby systému ProCop. Po konfiguraci je možné spustit služby obvykle spuštěním služby ProCop Data Server, která díky vazbám spustí i ostatní služby systému ProCop.

## Úpravy zástupců Panelu nástrojů ProCop

Na ploše systému Windows je standardně nainstalován zástupce pro spuštění Panelu nástrojů. Kliknutím pravým tlačítkem myši na zástupci volbou **Vlastnosti** se otevře dialog pro konfiguraci tohoto zástupce. Změníme položku **Cíl** na novou hodnotu (i s uvozovkami) a potvrdíme tlačítkem **OK**:

```
"C:\ProCop 3.3\ProCop.exe" /connect
```

Obdobně v nabídce **Start** podnabídka **Po spuštění** položka **Panel nástrojů ProCop** stiskem pravého tlačítka myši otevřeme lokální nabídku. Volbou **Vlastnosti** upravíme cíl odkazu stejně jako v předchozím případě.

Posledním je zástupce v nabídce **Start** je podnabídka **Monitorovací systém ProCop** položka **Panel nástrojů ProCop**. Cíl změníme stejně jako v předchozích případech. Odkážeme funkčnost všech odkazů.

## Úprava spuštění ohlašování stavu projektu ProTray

Spustím editor registru `regedit.exe` například zadáním tohoto příkazu do příkazové řádky. Najdeme klíč:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

Na pravé straně najdete textovou hodnotu **ProCop Tray**. Dvojitým klikem myši otevřeme editor a nastavíme hodnotu na (bez uvozovek):

```
C:\ProCop 3.3\ProTray.exe
```

Tím jsou všechny úpravy pro lokální monitorování hotovy. Po restartu počítače by mělo všechno fungovat správně. Následně je možno odstranit lokálního uživatele ProCop ze systému Windows.

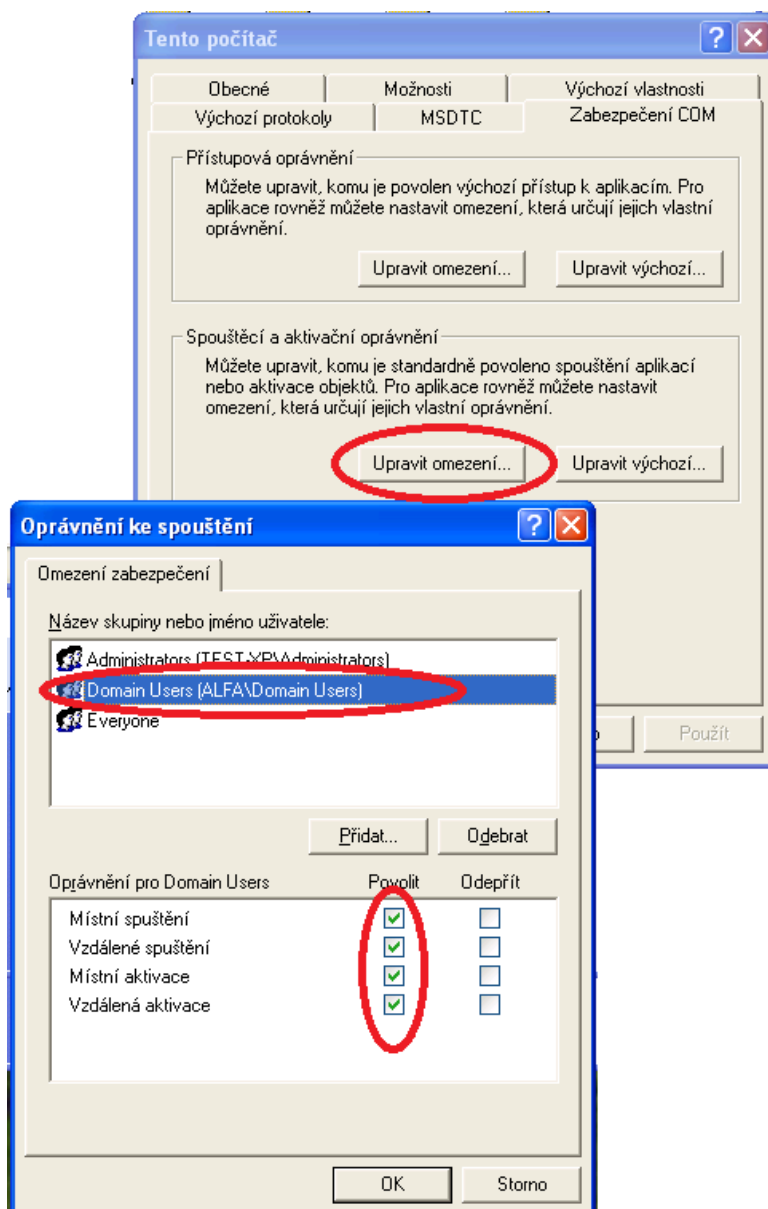
## Konfigurace DCOM pro provoz síťového pracoviště

---

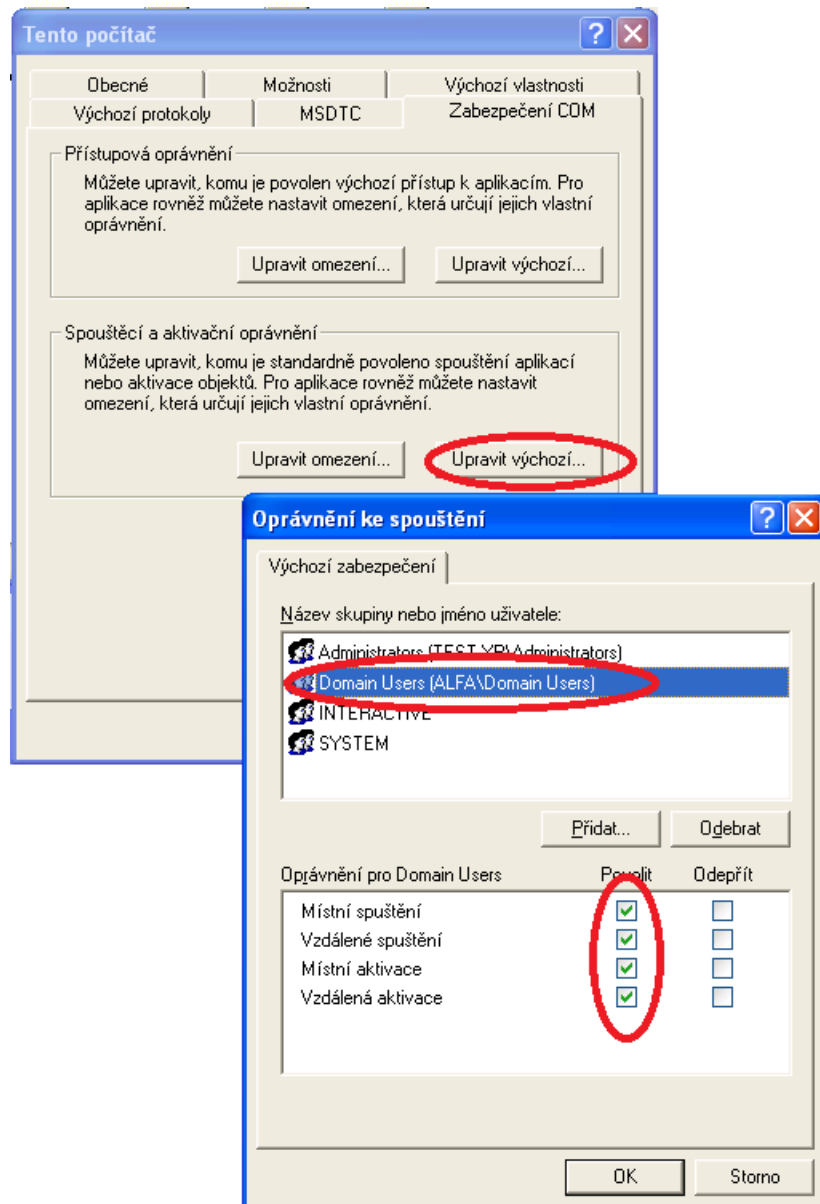
V případě požadavku na provoz se síťovým pracovištěm je nutno nakonfigurovat přístupová oprávnění pro přístup službami DCOM na tento server. Spustíme ovládací konzolu `dcomcnfg.exe` například zadáním příkazu do příkazové řádky Windows. Najdeme ve stromu **Službu komponent** větev **Počítače** položka **Tento počítač**. Pravým tlačítkem otevřeme nabídku a volbou **Vlastnosti** otevřeme dialogové okno. V záložce zabezpečení COM jsou k dispozici tlačítka **Upravit omezení** a **Upravit výchozí**. Provedeme konfiguraci shodně v dialogích pod oběma tlačítky:

- stiskem tlačítka **Přidat** přidáme skupinu uživatelů Domain Users
- následně vybereme přidanou položku v prvním seznamu
- ve sloupci **Povolit** ve druhém seznamu zatrhneme všechny dostupné volby (Místní i vzdálená aktivace i spouštění)
- po restartu je server připraven na připojení síťových pracovišť

*Nastavení omezení pro spouštěcí a aktivační oprávnění:*



*Nastavení výchozí pro spouštěcí a aktivační oprávnění:*



## 8.2 Úpravy klienta pro provoz bez uživatele ProCop

### Úpravy zástupců Panelu nástrojů ProCop

Na ploše systému Windows je nainstalován zástupce pro spuštění Panelu nástrojů. Kliknutím pravým tlačítkem myši na zástupci volbou **Vlastnosti** se otevře dialog pro konfiguraci tohoto zástupce. Změníme položku **Cíl** na novou hodnotu a potvrdíme tlačítkem **OK**:

```
"C:\ProCop 3.3\ProCop.exe" /remote=NAZEV_POCITACE
```

kde NAZEV\_POCITACE je doménové jméno počítače serveru.

Obdobně v nabídce **Start** podnabídka **Po spuštění** položka **Panel nástrojů ProCop** stiskem pravého tlačítka myši otevřeme lokální nabídku. Volbou **Vlastnosti** upravíme cíl odkazu stejně jako v předchozím případě.

Posledním je zástupce v nabídce **Start** podnabídka **Monitorovací systém ProCop** položka **Panel nástrojů ProCop**. Cíl změníme jako v předchozích případech. Odkoušíme funkčnost všech odkazů.



# Index

## - ~ -

~` 25

## - A -

Adresa serveru 39  
Aktivní 41  
Aktualizace hodnot 39  
ANSI 8  
Automatická 40

## - B -

Baud Rate 12  
bigint 43  
bit 43  
Buňka 37

## - C -

Consumption 42  
ConsumptionValid 42

## - D -

databáze 37  
Date 42  
DateOrNow 42  
datetime 42, 43  
DateTimeOrNow 42  
Datové typy 43  
DE-211 7  
DE-311 7  
Definice exportních tabulek 40  
Definice struktury tabulky 41  
Definice tabulek 40  
délka rámce 12  
Dispečerské PC 8

## - E -

EEPROM 13  
Ethernet / RS232 7  
Ethernet to serial 11  
Exit 13

## - F -

Firewall 15  
float 43  
Force transmit 11

## - G -

Gateway 9

## - H -

heslo 9  
Hlavní nabídka 9  
Hyperterminál 8  
char(64) 43

## - I -

Index 37  
Instance 37  
int 43  
IP adresa 8

## - K -

komunikační chyby 15  
komunikační rychlost 12  
Konfigurace IP adresy NPort 9  
Konzola správy ProSQL 39

## - L -

Local listen Port 11  
Logické adresy 13

## - M -

maska sítě 8  
Mbus 11

## - N -

Nastavení sériového portu 12  
Nastavení UDP Server/Client režimu 8  
nastavení Windows Firewall 15  
nastavitelný čas čekání 7  
nastavitelný počet pokusů 7  
Název sloupce 41  
Názvy sloupců v definici řádků tabulky 43  
Netmask 9  
none 43  
Normal 40  
Note 42  
NPort 7  
NPortWay 13

## - O -

Odkazy mezi sloupci 43  
Odkazy mezi sloupci při definici řádků 43

OP\_mode 11  
 OverallValid 42  
 Ovládací panely 15

## - P -

Parametrech spojení 13  
 Parametry modulu PX 29  
 Parametry připojení k datovému serveru 39  
 Parametry připojení k SQL databázi 39  
 Parametry spuštění ProSQL 39  
 parita 12  
 Password 9  
 PComm Lite Terminal emulator 8  
 ping 15  
 Plánování úlohy 40  
 Počet opakování 13  
 počet stop bitů 12  
 Popis exportu měřičů spotřeb 45  
 Požadavky před realizací 37  
 Prefix 42  
 Primární index 37, 41  
 Primární klíče 39  
 Problémy s virtuálními porty 7  
 Projekt ProSQL 39  
 ProSQL 37  
 Předvolba 41  
 Předvolené hodnoty v definici řádků 43  
 Přepínače na převodníku 8  
 Příklad exportu měřičů spotřeb 45  
 PX vlastnosti 29

## - R -

real 43  
 Report 40  
 Report Tabulka 43  
 rozsáhlé sítě 7  
 RS422 7  
 RS485 7  
 rychlost komunikace 13  
 Řádek 37  
 Řešení problémů s komunikací 15  
 Řetězec připojení 39  
 řízení toku 12

## - S -

Serial to Ethernet 11  
 Serialport 12  
 serverConfig 9  
 Síť 13  
 Sloupec 37, 41  
 Složený index 37

smallint 43  
 spolehlivější komunikace 7  
 Spuštění a kontrola výsledku 45  
 SQL 37  
 SQL server 37  
 SQL server (HW) 37  
 SQL server (SW) 37  
 SQL Typ 41  
 Static IP 9  
 Struktura dat 37  
 Struktura databáze 37  
 Struktura tabulky 37  
 SW1 8  
 SW2 8  
 SW3 8

## - T -

Tabulka 37  
 TaskTime 42  
 telnet 8  
 text 43  
 Time 42  
 tinyint 43  
 Trend 42  
 TrendBeginTime 42  
 TrendEndTime 42  
 TrendValid 42  
 Typ hodnoty 41  
 Typy hodnot 42

## - U -

UDP port 8, 11, 13  
 UDP Server/Client 11  
 UDP Server/Client mode 7  
 Uložení konfigurace do převodníku 13  
 Unknown 42  
 UpdateTime 42  
 Úpravy monitorovacího projektu 13  
 UserData 42

## - V -

Valid 42  
 Value 42  
 Virtuální port 7  
 Vkládání, kopírování a mazání definic řádků 43  
 Vlastnosti tabulky 40  
 Vlastnosti tabulky - plánování úlohy 40  
 Volba a konfigurace UDP režimu 11  
 Volba terminálu po připojení 8  
 VT100 8  
 výchozí brána 8



---

výjimky na firewallu 15  
vyrovnávací paměti 12  
Vytvoření tabulky 40  
Vytvoření tabulky - úlohy 40  
Vzdálené připojení přes síť 8

## - W -

Windows Firewall 15  
WriteTime 42

## - Z -

Zadání prázdné proměnné 43  
Zařízení nekomunikují 15  
zotavení komunikace 7