

merbon

DATABASE

Implementační návod

11/2023

Obsah

1	Úvod	4
1.1	Merbon DB	4
1.2	Implementační návod.....	4
1.3	Terminologie.....	4
2	Instalace.....	4
2.1	Merbon Database Adapter.....	5
3	Klíče v databázi.....	6
4	Správa a nastavení.....	7
4.1	Uživatelé	7
5	Konfigurace databáze	9
5.1	Cesty k souborům	10
5.2	Porty	10
5.3	Hloubka logování událostí	11
5.4	Velikosti souborů.....	11
5.5	Ostatní	12
6	API	12
6.1	Základní principy.....	12
6.2	Identifikace proměnných.....	13
6.3	Čtení dat	13
7	Datová pumpa EC - jednorázová	15
7.1	Příklad volání pro import historických dat z RcWare Vision do Merbon DB.....	16
7.2	Příklad volání pro přesun dat z SQL databáze do Merbon DB	16
7.3	Seznam parametrů a jejich použití	16
7.4	Výběr podmnožiny proměnných	18
8	Mazání dat	18
8.1	Kompletní mazání.....	18
8.2	Mazání jedné proměnné přes web	19
8.3	Selektivní mazání - DbTool	19

9	Nastavení zápisu do databáze v SoftPLC.....	21
10	Nastavení zápisu do databáze v Merbon IDE	24

1 Úvod

1.1 Merbon DB

Merbon DB funguje jako služba nebo konzolová aplikace v systémech Windows. Data jsou do ní ukládána z klientských aplikací (Merbon SCADA, PLC, cizí systémy) prostřednictvím otevřeného rozhraní (web service, SOAP).

Díky aplikaci několika druhů komprese dat je dosahováno minimální možné velikosti datových souborů na disku. Navíc je díky použitému způsobu ukládání možno soubory kdykoli za běhu aplikace kopírovat bez narušení konzistence zkopírovaných dat.

Merbon DB je navržena jako vysoce škálovatelná aplikace, která umožňuje ukládat stovky tisíc proměnných/řad dat bez ztráty výkonu. Navíc oproti jiným používaným databázovým řešením (SQL, noSQL) nejsou její hardwarové požadavky nijak vysoké při prakticky konstantním výkonu.

Při rozšiřování systému a přechodu ze standardního souborového ukládání historických dat v RcWare Vision je možné po instalaci databáze tato stará data importovat. Import může probíhat za současného záznamu nových dat, takže nedojde ke ztrátě informací.

Databáze má webové rozhraní pro administraci, správu uživatelů a diagnostiku.

1.2 Implementační návod

Tento implementační návod slouží k popsání postupu nasazení a konfigurace databáze.

1.3 Terminologie

proměnná – entita, představující jeden datový bod. Reprezentace datového bodu v databázi.

datový bod – zdroj dat pro vzorkování. Je to například čidlo teploty, stav čerpadla nebo náměr vodoměru. Jeden (fyzický) měřič může obsahovat více datových bodů, například elektroměr má spotřebu, výkon, 3x napětí ve fázích, 3x výkony fází, kmitočet, účinník atd.

vzorek – záznam obsahu nějaké proměnné k určitému času

řada dat, datová řada – množina vzorků, které se vztahují k jedné proměnné.

2 Instalace

Instalace je popsána v Merbon SCADA Instalačním manuálu. Databáze využívá tyto TCP porty:

- ukládání do databáze je prováděno přes API na portu 9876

- přístup do webového rozhraní databáze pro administraci je na portu 11112
- Merbon Database Adapter: přístup je na portu 8686

Hardwarové požadavky na počítač jsou tyto:

- OS Windows Vista a novější
- Paměť vyhrazená pro Merbon DB min. 1 GB RAM (základ stačí pro desítky tisíc proměnných ukládaných po 1 min.; dále např. pro 1 200 000 proměnných stačí 10 GB RAM), celková paměť musí být větší, aby umožnila normální funkci Windows a dalších programů a služeb
- Procesor: doporučena min. 2 jádra

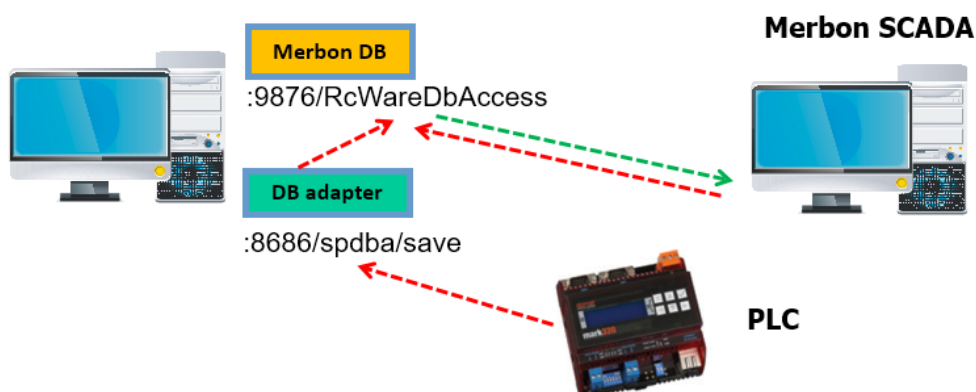
Příklad konfigurace PC pro typické použití:

RAM: 4GB + 1GB pro každých dalších 50 000 proměnných (datových bodů).

Disk: 500 GB + 2GB ročně pro 10 000 proměnných ukládaných každé 3 minuty

2.1 Merbon Database Adapter

Merbon Database Adapter je služba, která vytváří koncový bod pro připojení PLC, pokud tato PLC mají zapisovat historii do databáze. Merbon Database Adapter tato data z PLC přijímá, zpracovává a zapisuje do Merbon DB.



Tok historických dat mezi PLC, databází a Merbon SCADA. Červená – zápis, zelená – čtení.

Merbon Database Adapter nevyžaduje žádnou licenci a může pracovat i na jiném počítači, než je instalována databáze. K tomu obvykle není důvod, může to však vyžadovat bezpečnostní politika (zamezí se tím, aby počítač s databází byl viditelný pro desítky nebo stovky PLC, která do databáze zapisují data – k databázi existuje jen jedno spojení, a to od Merbon Database Adapteru). Výchozí URL cílové databáze je <http://localhost:9876/RcWareDbAccess>, dá se ale změnit v konfiguračním souboru `C:\Apps\Merbon\Database Adapter\Web.config`.

URL Merbon Database Adapteru je [http://\[IP_adresa_servertu\]:8686/spdba/save](http://[IP_adresa_servertu]:8686/spdba/save) - to je také nutné nastavit v konfiguraci PLC, například takto:

Parametry Merbon DB		-
Povoleno	True	■
URL	http://192.168.1.44:8686/spdba/save	■
Interval komunikace [min]	1	■
Uživatelské jméno	rc_user	■
Heslo	rc_user	■
Pozn.		□
Parametry Proxy		+

Uživatel a heslo (zde rc_user / rc_user) musejí být v databázi definovány. Nezapomeňte v PLC dále nastavit

- **výchozí bránu** a případně **adresu DNS serveru** v případě, že databázový server (resp. server, kde běží Merbon DB Adapter) je mimo místní síť
- **identifikaci runtime pro historii** – tento řetězec se v databázi objeví jako jeden z klíčů. Délka Runtime ID by neměla přesáhnout 31 znaků.

3 Klíče v databázi

Klíč v databázi je údaj nebo skupina údajů, které plně postačují k identifikaci položky v databázi (nebo její části). Primární klíč jednoznačně identifikuje položku v tabulce, v níž je definován. V našem případě je položka jednoznačně identifikována množinou „klíčových“ klíčů.

The screenshot shows the 'Mervis DB Administration' interface with a navigation bar containing 'Users', 'Variables', 'Monitoring', 'Permissions', 'Management', and a 'Log out' button. The main heading is 'Edit Variable: f78ef2a7-a479-4312-a681-8b215b3cb477'. Below this, there is a 'Keys' section with an 'Edit Keys' button. A table lists the keys for the variable:

Key Name	Value	Iskey
CommUid	9176	Iskey: false
GUID	9176	Iskey: true
SoftPlcRtId	co2	Iskey: true
VariableName	\$hw\$.SUI905_ActualTemp\$	Iskey: false

At the bottom of the 'Keys' section, there are buttons for 'Cancel', 'Save', 'Graph', and 'Delete variable'.

Například u proměnných zapsaných z Merbon RT ve vlastnostech proměnné (tlačítko „Edit“ u proměnné) vidíme, že proměnná má dva klíče: SoftPlcRtId a GUID.

SoftPlcRtId: identifikace runtime. Každé PLC, které chce zapisovat do databáze, musí mít unikátní RTID – identifikaci runtime. Tato identifikace - textový řetězec – se nastavuje ve vlastnostech PLC v položce *History Runtime Identification*. Jeho délka by neměla přesáhnout 31 znaků.

GUID: identifikátor proměnné v rámci runtime. Při kompilaci dostane každá proměnná (nebo struktura, jako je pole, funkční blok atd.) unikátní GUID (které odpovídá pozici proměnné v paměti).

Při změně aplikačního softwaru (např. přepojení čidla na jiný vstup) v PLC, které zapisuje do databáze, může dojít ke změně hodnot klíčů, a to i primárních. Klienti, kteří data z databáze využívají, musejí toto mít na vědomí a datové řady rekonstruovat (typicky využívat názvu proměnné). Viz též níže – *Nastavení zápisu do databáze v Merbon IDE a v SoftPLC*.

4 Správa a nastavení

Databáze se běžně spravuje pomocí webového rozhraní, které je dostupné na adrese <http://localhost:11112/admin>. Výchozí přístupové údaje jsou tyto:

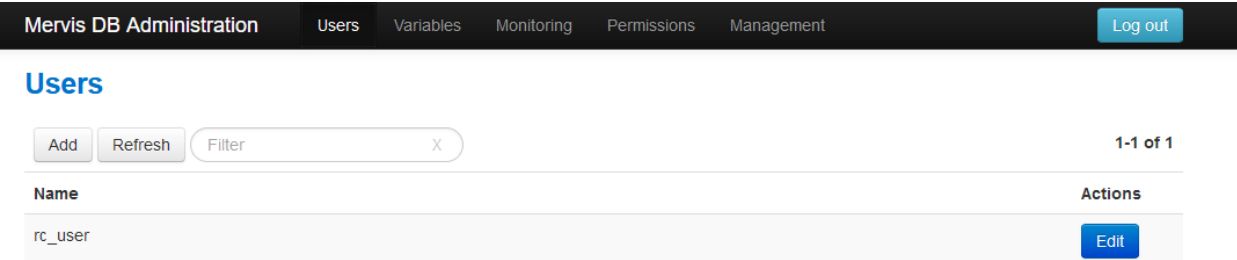
Uživatel: **admin**

Heslo: **rw**

Po přihlášení se otevře webová stránka s menu v horní části.

4.1 Uživatelé

Správa uživatelů je v záložce *Users*:



Name	Actions
rc_user	Edit

Jako první krok je užitečné založit další uživatele, kteří budou využíváni pro čtení a zápis dat – jejich přihlašovací údaje se pak zadávají v klientských systémech, jako jsou PLC, cizí programy atd. Uživatel **admin** by měl být využíván pouze pro správu databáze. Typický uživatel, použitý v příkladech v dokumentaci, je **rc_user** / **rc_user**. Z bezpečnostních důvodů by uživatelé měli mít jiná než výchozí hesla; platí zde všechna běžná pravidla pro bezpečnou tvorbu hesel.

Přihlašovací údaje pro uživatele admin jsou uloženy v konfiguračním souboru jako plain text. Podobně v případně použitém PHP connection stringu jsou přihlašovací údaje do SQL databáze. Proto je nutné v rámci operačního systému zamezit neoprávněným uživatelům přístup do adresáře s konfiguračním souborem.

Tlačítkem *Edit* lze heslo uživateli změnit.

Uživatel jednou založený již nejde smazat, jsou (nebo mohou být) na něj navázány proměnné. Je pouze možné mu změnit heslo.

Záložka *Variables* obsahuje prohlížeč proměnných.

Select	Keys	UserName	Actions
<input type="checkbox"/>	[{"IsKey":false,"Key":"CommUid","Value":"9174"},{"IsKey":true,"Key":"GUID","Value":"9174"}, {"IsKey":true,"Key":"SoftPlcRtId","Value":"co2"}, {"IsKey":false,"Key":"VariableName","Value":"\$hw\$.SUI905_ActualCO2Ppm\$"}]	rc_user	Edit
<input type="checkbox"/>	[{"IsKey":false,"Key":"CommUid","Value":"9177"}, {"IsKey":true,"Key":"GUID","Value":"9177"}, {"IsKey":true,"Key":"SoftPlcRtId","Value":"co2"}, {"IsKey":false,"Key":"VariableName","Value":"\$hw\$.SUI905_ActualRHS\$"}]	rc_user	Edit

Proměnné (datové řady) je možné editovat, zobrazovat v grafu a mazat. Proměnné se dají označit zaškrtnutím *Select* a pak mohou být hromadně smazány. Změny jsou nevratné, proto pozor na to, co děláte. Pro masivnější operace, jako je mazání stovek proměnných, slouží specializované programy. Podrobnosti viz také *Mazání dat*. V případě potřeby kontaktujte technickou podporu na support@domat.cz.

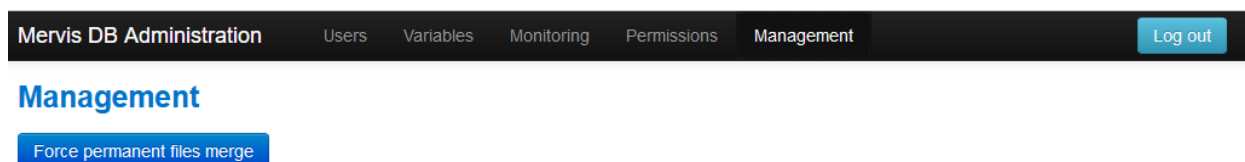
V záložce *Monitoring* je výpis systémových proměnných databáze a dalších užitečných údajů.

Id	Value	Unit
/rcWareDb/instName	MerbonDB	
/rcWareDb/lastSave	01/06/2022 15:50:07	
/rcWareDb/serverStart	01/06/2022 14:50:05	
/rcWareDb/licenseExpiration	01/01/9999 00:00:00	
/rcWareDb/licensedVariableCount	2200	
/rcWareDb/licensedInstanceCount	1	
/rcWareDb/lastStatsUpdate	01/07/2022 08:00:43	
/rcWareDb/memWorkingSetSize	57004032	
/rcWareDb/variableCount	3	
/rcWareDb/userCount	1	

Asi nejdůležitější pro diagnostiku je počet licencovaných proměnných a počet proměnných zapsaných v databázi. Pokud je počet proměnných překročen, databáze je v režimu pouze pro čtení – k dalším zápisům nedochází! Doporučuje se proto pravidelná kontrola, zejména před úpravami v klientských programech (PLC, SCADA), zda nemůže dojít k překročení limitu. V případě potřeby lze licenci včas rozšířit.

Při překročení počtu proměnných není vydáváno žádné explicitní varování, databáze nemá běžné uživatelské rozhraní. Kontrola počtu proměnných by proto měla probíhat vždy před plánovaným rozšířením počtu proměnných (přidání dalších datových řad do databáze) a nadto pravidelně – např. vždy po 6 měsících provozu.

Stránka *Management* obsahuje tlačítko *Force permanent files merge*.



Tato funkce vyvolá spojení permanentních souborů s dočasnými. Jde o „neškodnou“ operaci, kterou lze provést kdykoli – typicky např. před zálohováním databáze nebo přesunem souborů na jiné úložiště. Za běžného provozu ke spojování dochází, když dočasné soubory dosáhnou max. velikosti, zadané v konfiguračním souboru.

5 Konfigurace databáze

Veškeré konfigurační parametry databáze jsou v konfiguračním souboru *ESG.Db.Server.Host.exe.config*, který je v adresáři s databází – výchozí umístění je *C:\Apps\Merbon\Merbon Database*. Parametry se načítají při startu služby, při případných úpravách konfiguračního souboru je pak nutné službu databáze zastavit a spustit (např. pomocí Merbon SCADA instalátoru).

Dále jsou popsány některé vybrané parametry, které mohou být v průběhu provozu databáze upravovány a chod databáze tak optimalizován. Jako vždy platí, že pokud k úpravám není závažný důvod, necháváme parametry na výchozích hodnotách. Parametr nejrychleji nalezneme vyhledáním textu klíče v konfiguračním souboru.

5.1 Cesty k souborům

V konfiguraci jsou k souborům tři cesty. U instalací do 40 – 50 000 datových bodů mohou být všechny soubory ve výchozích cestách, při větším počtu datových bodů můžeme začít systém optimalizovat přesunutím souborů na úložiště s rychlejším přístupem:

InfrastructureDataPath

```
<add key="InfrastructureDataPath" value="c:\apps\merbon\Merbon Database Warehouse" />
```

Cesta k souborům infrastruktury (seznam proměnných, seznam uživatelů, stavové informace). K souborům není přistupováno příliš často a jejich velikost je střední, ale obecně je dobré mít sem rychlý sekvenční přístup pro čtení a zápis – dosáhne se tím rychlejšího startu databáze.

LogsPath

```
<add key="LogsPath" value="c:\apps\merbon\Merbon Database Warehouse" />
```

Zde jsou data v neoptimalizované podobě – nejčerstvější uložená data před uložením do trvalých souborů. Soubory nejsou velké, řádově desítky MB. Jsou ovšem čteny a sekvenčně zapisovány často. Je výhodné, když do tohoto adresáře je rychlý přístup pro čtení a zápis při startu a zastavování databáze.

Navzdory názvu nejde o cestu k záznamům o událostech – viz níže, Soubory s logy.

PermanentFilePath

Data trvale uložená, většinou pouze pro čtení. Velké soubory. K zápisům dochází zřídka. Úložiště by mělo mít rychlý random přístup pro čtení.

```
<add key="PermanentFilePath" value="c:\apps\merbon\Merbon Database Warehouse" />
```

Soubory s logy

Záznamy o systémových událostech databáze, výpisy chyb. Výhodné pro diagnostiku v případě že se databázi např. nepodaří spustit.

```
<add name="circular" type="ESG.Core.Logging.CircularTraceListener, ESG.Core" initializeData="c:\apps\merbon\Merbon Database Logs\rcwaredbserver.log" maxFileSizeKB="100" traceOutputOptions="DateTime"></add>
```

5.2 Porty

TCP porty pro jednotlivé služby by neměl být důvod měnit. Pokud je to nutné, upravíme jejich čísla v těchto řádcích:

9876 – API pro klientské programy, zápis a čtení dat

- ```
<serviceMetadata httpGetEnabled="true" httpGetUrl=http://localhost:9876/RcWareDbAccess
```
- ```
<endpoint contract="ESG.Db.Server.Shared.IHistoryAccess" address="http://localhost:9876/RcWareDbAccess" binding="basicHttpBinding"
```

```
name="HistoryAccess" bindingConfiguration="HistoryAccessHttpBinding"
bindingNamespace="http://dev.rcware.eu/esgdb_HTTP" />
  • <endpoint contract="ESG.Db.Server.Shared.IHistoryAccess"
address="http://localhost:9876/RcWareDbGZipAccess" binding="customBinding"
name="HistoryAccessGZip" bindingConfiguration="HistoryAccessGZipBinding"
bindingNamespace="http://dev.rcware.eu/esgdb_HTTP_GZIP" />
```

(Číslo tohoto portu se v konfiguračním souboru vyskytuje třikrát.)

11111 – monitoring

```
<endpoint contract="ESG.Core.ServerMonitor.Shared.IWebDataRetrieval"
behaviorConfiguration="ServerMonitorWebBehavior"
bindingNamespace="http://dev.rcware.eu/ServerMonitor_REST"
binding="webHttpBinding" bindingConfiguration="WebHttpsBindingConfig"
address="http://localhost:11111/" />
```

11112 – webové rozhraní pro administraci

```
<endpoint
contract="ESG.Db.Server.Engine.Administration.IAdministrationApi"
behaviorConfiguration="ServerMonitorWebBehavior"
bindingNamespace="http://dev.rcware.eu/Administration_REST"
binding="webHttpBinding" bindingConfiguration="WebHttpsBindingConfig"
address="http://localhost:11112/" />
```

5.3 Hloubka logování událostí

```
<add name="sourceSwitch" value="All" />
```

Možné hodnoty jsou *Information*, *Warning*, *Error*, *All*. Pokud databáze běží bez problémů, je vhodné nastavit *Error*. Zamezí se tím vzniku velkých logovacích souborů. Logovací soubory je možné kdykoli smazat.

5.4 Velikosti souborů

Důležitý parametr pro optimalizaci.

Velikost souborů se zadává v mocninách dvou, např.

20 ... 1 048 576 (1 MB)

...

30 ... 1 073 741 824 (1 GB)

Maximum je

32 ... 4 294 967 296 (4 GB)

Každý typ souboru má dva parametry:

...FileSize

Při překročení této velikosti se zakládá další soubor, v názvu má číslo o jednu větší.

...FileGrowth

Skoky, po kterých se alokují data do souborů.

Typy souborů jsou následující:

Users... – soubory s definicí uživatelů, hashovanými hesly atd.

Variables... – seznamy proměnných a jejich metadat

PersLog... – soubory pro rychlé dočasné uložení dat, před uložením do trvalých souborů

PermFileSet... – soubory pro trvalé uložení dat

Perm... – U databází se stovkami tisíc proměnných je lepší mít tyto soubory velké, protože se tím zmenšuje počet indexů a šetří se tak paměť. Např. pro 1 000 000 proměnných by měl být již určitě zvolen 4 GB soubor (hodnota 32). Na druhou stranu zvětšování permanentních souborů způsobuje zpomalování spojování permanentních souborů s logy (buffery přijatých hodnot) – např. k velkému souboru (4 GB) se připojí 64 MB a celé se to uloží, což zabírá výkon PC jako takového.

Index... – indexové soubory. Když se začnou vytvářet indexy .s0, .s1 atd., je vhodné zvětšit velikost indexového souboru.

5.5 Ostatní

MaxDistributionQueueLength

Tento parametr neměňte, při zmenšení hrozí zpomalení odezvy databáze až o tři řády.

Popisy dalších parametrů jsou v komentářích v konfiguračním souboru. Další parametry ale obvykle není nutné měnit.

6 API

API (Application Programming Interface) je rozhraní, které databáze poskytuje pro programování cizích aplikací. V případě Merbon DB jde o popis komunikace mezi cizím systémem, jako je například odečtový systém měřičů nebo vizualizace, a databází Merbon DB. Kompletní dokumentace k API je poskytována samostatně.

API umožňuje ukládat a vyčítat hodnoty typu Double, Boolean, Integer, DateTime, String a Blob. Každá hodnota je uložena se svou časovou značkou, případně s údajem “Good Through” - tedy údajem, do kdy je daná hodnota platná. Dále by měl klient poskytovat s každou ukládanou hodnotou informaci o délce intervalu, ve kterém je hodnota ukládána. Základem API jsou metody **GetData** a **SaveData**, doplněné o speciální metodu **GetParticularData**.

6.1 Základní principy

Při práci s libovolnými textovými řetězci jsou rozlišována malá a velká písmena.

Časové značky, Good Through a časové parametry funkcí jsou vždy v UTC.

Kompletní technický popis API je k dispozici v souboru WSDL.

6.2 Identifikace proměnných

Každý klient, který ukládá data pomocí metody **SaveData**, si sám volí, jakým způsobem unikátně identifikuje hodnotu, resp. její příslušnost k dané proměnné.

Proměnná je identifikována množinou klíčů a hodnot, kde je dále určena množina klíčů, které představují unikátní kombinaci jednoznačně identifikující proměnnou.

Příklad:

HodnotaA

- key1: value1, isKey: **true**
- key2: value2, isKey: false
- key3: **value3**, isKey: **true**
- value: 200
- time stamp: UTC 12.01.2009 10:00:00

HodnotaB

- key1: **value10**, isKey: **true**
- key2: value20, isKey: false
- key3: **value30**, isKey: **true**
- time stamp: UTC 12.01.2009 10:00:00

HodnotaC

- key1: **value10**, isKey: **true**
- key2: value60, isKey: false
- key3: **value30**, isKey: **true**
- time stamp: UTC 12.01.2009 11:00:00

V tomto případě hodnoty (vzorky) patří dvěma různým proměnným - hodnota B a C patří ke stejné proměnné (datovému bodu), neboť “klíčové” klíče mají shodnou hodnotu.

6.3 Čtení dat

Jelikož v Merbon DB může být pro konkrétní proměnnou uloženo mnoho milionů údajů, jsou dotazy na hodnoty limitovány svojí velikostí a pro vyčtení většího počtu dat je potřeba postupovat pomocí zvyšujícího se parametru offset, tak jak je vráceno v odpovědi serveru.

Současně s tím se lze dotazovat na hodnoty více proměnných současně, takže i pro iterování “přes proměnné” je k dispozici samostatný offset parametr.

Popis třídy Valueltem

- "Hvt" - HistoryValueType - typ hodnoty (Double, Blob, String, Int64, NotDefined, ISODateTime, Boolean)
- "Ivl" - Interval - interval, se kterým byla hodnota ukládána
- "Ts" - UtcTimeStamp - okamžik platnosti hodnoty
- "Gt" - GoodThrough - okamžik, do kterého je daná hodnota platná
- "Bv" - BooleanValue
- "Dv" - DoubleValue
- "Iv" - Int64Value
- "Sv" - StringValue
- "BinV" - BlobValue
- "Dtv" - DateTimeValue

Ukázka - příklad v PHP

Příklad je uveden pouze pro ilustraci. Pro implementaci je nutná další dokumentace.

Vyžádejte si ji u technické podpory, support@domat.cz.

```
<?php /* Příklad ukazující vyčítání hodnot z Merbon DB prostřednictvím API v2.0. */
ini_set("soap.wsdl_cache_enabled", "1"); // cachování wsdl - lepší nastavit v php.ini
try
{ // URL - opsat z RC a přidat ?wsdl
  // pouzivame-li souborove wsdl - nahradit vsechny vyskyty db.rcware.eu:9877 za realnou adresu
  // a port aplikace
  $soapClient = new SoapClient('./wsdl/RcWareDbAccess.wsdl.xml', array('trace' => 1,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS)); // trace = 1 - pro debug

  $credentials = array('Name'=>'username', 'Password'=>'password'); // Přístup k databázi

  $utcTZ = new DateTimezone('UTC'); // všechny časy by měly být v UTC

  $utcFrom = new DateTime('-6months', $utcTZ ); // data chceme od tohoto času
  $utcTo = new DateTime('now', $utcTZ ); // do tohoto času
  $valOffset = 0; // začneme od prvního údaje
  $valCount = 10; // budeme postupovat po XX údajích (optimální hodnota ja cca 3000)
  $varOffset = 0;
  $varCount = 10;

  // Merbon DB identifikuje každou proměnnou množinou klíčů a hodnot
  // Celá množina musí být pro každou proměnnou unikátní (nutné při ukládání),
  // ale při čtení se lze dotazovat i neunikátní podmnožinou klíčů a hodnot.
  // Vráceny jsou pak hodnoty všech proměnných, které daná kritéria splňují.
  // mohou si říci o více proměnných najednou, stačí je zde všechny vyjmenovat
  // DPGuid k identifikaci stačí - je unikátní v rámci jednoho projektu.
  $variablesKey = array(array(array('IsKey'=>true, 'Key'=>'DPGuid',
    'Value'=>"2C7F333A-85BA-465B-A7AD-8858B4EBB2F2")));

  echo "Hodnoty požadujeme od: {$utcFrom->format('c')} do: {$utcTo->format('c')}\n\n";

  do // vlastní vyčtení dat ze serveru
  {
    $response = null;
  }
do
```

```
{
    // protistrana bohužel správně neinterpretuje časové zóny, pokud je použit jen parametr 'c'...
    $response = $soapClient->GetData(array('credentials'=>$credentials,
                                          'variablesKey'=>$variablesKey,
                                          'utcFrom'=>$utcFrom->format('Y-m-d\TH:i:s\Z'),
                                          'utcTo'=>$utcTo->format('Y-m-d\TH:i:s\Z'),
                                          'valueOffset'=>$valOffset,
                                          'valueCount'=>$valCount,
                                          'variableCount'=>$varCount,
                                          'variableOffset'=>$varOffset));

    $valOffset = $response->nextValueOffset;

    foreach($response->GetDataResult->Mvr as $varArray)
    {
        foreach($varArray->Vals->I as $val)
        {
            echo print_r($val);
        }
    }
} while($response->nextValueOffset != -1);

$varOffset = $response->nextVariableOffset;

} while ($response->nextVariableOffset != -1);
}
catch (Exception $e)
{
    //print_r($soapClient->__getLastRequest()); //vyžaduje "trace = 1"
    print_r($e);
}
?>
```

Zápis dat a další funkce jsou popsány v elektronické dokumentaci k API.

7 Datová pumpa EC - jednorázová

Datová pumpa je jednoduchý program, který umožňuje **jednorázově** po spuštění kopírovat data mezi různými zdroji. Zdrojem může být

- databáze Merbon DB
- adresář s historickými soubory RcWare Vision nebo Merbon SCADA (soubory .txt)
- obecná SQL databáze

Program se volá z příkazové řádky a jeho parametry určují požadovanou činnost a koncové body pro přenos dat. Je vhodné vytvořit textový soubor .bat s příkazem a v okně s příkazovým řádkem spouštět tento soubor. Všechny parametry se zadávají v rámci jednoho příkazového řádku. Některé parametry jsou uvozeny spojovníkem „ - “.

I když je datová pumpa spuštěna víckrát za sebou se stejnými (nebo překrývajícími se) parametry nad stejnými databázemi, stejná data se v cílové databázi neduplikují, jsou zapsána

pouze jednou. Během kopírování dat do cílové databáze, které může trvat i desítky hodin, může cílová databáze normálně pracovat - přijímat nová data z technologií a poskytovat data dalším klientům.

7.1 Příklad volání pro import historických dat z RcWare Vision do Merbon DB

Nejčastější situace. Používá se v případě, že projekt RcWare Vision s ukládáním do textových souborů je migrován do RcWare Vision s ukládáním do Merbon DB (v podstatě přidání Merbon DB do existující instalace RcWare Vision), nebo při migraci staršího projektu RcWare Vision do prostředí Merbon SCADA s využitím databáze Merbon DB.

```
EsgDbDataPump.exe importType:file directory:"c:\temp\safranka\DATA"  
endPointName:"HistoryAccess" destination:http://localhost:9876/RcWareDbAccess  
destinationUser:rc_user destinationPsw:rc_user from:"2010/1/1 00:00"  
to:"2010/12/31 00:00" historyFileType:all -skipError
```

7.2 Příklad volání pro přesun dat z SQL databáze do Merbon DB

Používá se v případě, že RcWare Vision využíval pro ukládání dat obecnou SQL databázi. Z ní se nyní data kopírují do Merbon DB.

```
EsgDbDataPump.exe importType:sql directory:"c:\RcData\DATA"  
connectionString:"Data Source=local;Integrated  
Security=No;user=rc_user;password=rc_user;" endPointName:"HistoryAccess"  
destination:http://localhost:9876/EsgDbAccess destinationUser:root  
destinationPsw:root from:"1970/1/1" to:"2009/10/8 12:30" -skipError
```

7.3 Seznam parametrů a jejich použití

importType: esgdb, sql, file, datalogger - typ importu dat; podle něj se používají i ostatní parametry

- esgdb – jiná databáze Merbon DB
- sql – SQL databáze, do níž zapisoval data RcWare Vision (vzácný případ)
- file – soubory RcWare Vision nebo Merbon SCADA (nejčastější situace)

source:http://172.16.0.45:9876/EsgDbAccess - adresa zdroje dat

sourceUser:rc_user - uživatel zdroje dat

sourcePsw:rc_user - heslo uživatele zdroje dat

destination:http://localhost:9876/EsgDbAccess - cíl dat

destinationUser:root – uživatel cíle dat

destinationPsw:root – heslo uživatele cíle dat

from:1970/1/1 - od kdy (formát yyyy/m/d)

to:2009/8/1 - do kdy (formát yyyy/m/d)

readBlockCount:10000 - kolik hodnot načítat v jednom dotazu

writeBlockCount:1000 - kolik hodnot ukládat v jednom volání SaveData

skipError - pokračovat pokud server uloží méně dat než mu bylo posláno; pokud není zadáno, tak skončí

forceRepeat - zkouší ukládat data, která se nepodařilo uložit, dokud se to nepodaří. Mezi každým opakováním čeká 30s

readVariableFromSource: - Datová pumpa si (s použitím stejného uživatelského jména a hesla) přečte seznam proměnných z cílové databáze a ze zdrojové databáze naimportuje pouze ty, které v ní již existují. Využívá se pro doplnění databáze staršími vzorky z databáze, která obsahuje i jiné proměnné, které v nové databázi nebudou třeba.

useGetData - použije pro čtení ze zdroje metodu *GetData*, která je podporována jen do verze databáze 3.2 – v novějších už je jen *GetVariableArrayData*

directory: - adresář s daty RcWare Vision, kde se budou hledat textové soubory a soubor *EsgDbImport.inf* s informacemi pro import (při importu pouze některých – ne všech – datových bodů, viz Výběr podmnožiny proměnných)

connectionString: - connectionstring k SQL, odkud se budou RcWare data importovat

historyFileType:all,long,short,hystereze – specifikuje, které soubory se budou importovat:

- all – všechny soubory .tx*
- long – soubory .txl
- short – soubory .txs
- hystereze – soubory .txh

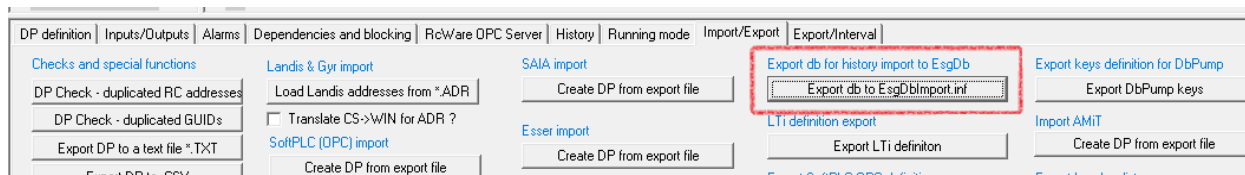
endPointName:”jméno endpoint konfigurace v .config souboru” - kvůli rozpoznání, jakým typem spojení (basicHttpBinding pro SOAP services, netTcpBinding pro trvalé TCP spojení s binárními daty) se má komunikovat, default je basicHttpBinding. HistoryAccess je odkaz do souboru AppConfig, který určuje podrobné parametry spojení.

Další příklad konkrétního použití:

```
D:\_pom\DataPump>EsgDbDataPump.exe importType:file
directory:"c:\data\FVE_KOSORIN" endPointName:"HistoryAccess"
destination:http://localhost:9876/RcWareDbAccess destinationUser:rc_user
destinationPsw:rc_user from:"2010/1/1 00:00" to:"2011/8/15 00:00"
historyFileType:all -skipError
```

7.4 Výběr podmnožiny proměnných

Při importu z textových souborů se importují všechny proměnné (jejichž vzorky vyhovují danému časovému intervalu). Někdy je potřeba určit, které proměnné (datové body) se mají importovat (pro případ, že se mají importovat jen některé řady dat a ne celý obsah historických souborů). To se dělá tak, že se pro příslušný projekt vygeneruje soubor *EsgDbImport.inf*, a to v RcWare Vision v tabulce datových bodů, záložka Import/Export.



Postup je následující:

- odemkněte datový soubor – v RcWare tabulka s datovými body, ikona visacího zámku
- označte (klávesou T, příp. v menu *Označování a výběr*) všechny datové body, které se mají importovat
- dole v tabu Import/Export stiskněte tlačítko *Export db do souboru EsgDbImport.inf*. V adresáři s projektem, kde jsou také soubory s historickými daty, vznikne tento soubor a pumpa si ho po spuštění načte a řídí se podle něj. V případě, že historická data (soubory .tx...) jsou umístěna jinde, soubor *EsgDbImport.inf* je potřeba zkopírovat k nim.

8 Mazání dat

Data by z databáze Merbon DB měla být mazána jen po zralé úvaze a v případě, že to je nutné např. k odstranění již nepotřebných datových řad při hrozícím překročení počtu proměnných.

8.1 Kompletní mazání

Kompletní data všech proměnných lze nejjednodušeji smazat odstraněním souborů

- variables*.*
- plog_*.*
- pfs*.*

v adresáři `C:\Apps\Merbon\Merbon Database Warehouse` (nebo jiném, pokud byl výchozí adresář změněn). Uživatelé a jejich práva zůstanou zachováni. Po startu si databáze vytvoří nové, prázdné soubory.

8.2 Mazání jedné proměnné přes web

Jednoduché smazání celé datové řady pro jednu proměnnou lze provést přes webové rozhraní *Variables – Edit* tlačítkem *Delete variable*. Tato funkce po potvrzení smaže veškeré vzorky, týkající se dané proměnné.



8.3 Selektivní mazání - DbTool

K selektivnímu mazání více proměnných najednou slouží program **DbTool**. Pro vyčtení údajů o všech proměnných je potřeba program spustit s adresou administrace vybrané databáze (tedy TCP port 11112, pokud nebyl v konfiguračním souboru změněn) a jménem/heslem **pro webový přístup do administrace (tedy ne pro běžného uživatele databáze, který zapisuje a čte data)**. Program se musí spustit s parametrem *csvStatsOutput*, což vytvoří soubor se seznamem všech proměnných v databázi a jejich vlastnostmi.

Pozn.: Okno PowerShell otevřeme v adresáři s programem stisknutím Shift a pravým kliknutím myši, pak *Otevřít zde okno PowerShell*.

```
.\ConsoleDBDotNet.exe --db http://localhost:11112/ --user user --password password --csvStatsOutput out.csv
```

U větších databází s desítkami tisíc datových řad může export trvat i několik hodin. Spouštějte tedy vyčítání na počítači, který může nerušeně běžet až do konce procesu, ideálně na tomtéž, kde je instalována databáze.

Takto vytvořený soubor (zde *out.csv*) lze následně použít pro smazání vybraných proměnných z databáze. Otevřeme ho v Excelu (nikoli přímým kliknutím na *.csv* soubor - v Excelu založíme nový prázdný soubor a data importujeme v menu *Data – Z textu/CSV*) a **odstraníme řádky s proměnnými, které chceme v databázi zachovat.**

Zde POZOR, často intuitivně ze souboru odstraňujeme proměnné, které mají být smazány. To je špatně. Odstranit musíme řádky s proměnnými, které mají být zachovány. Vytváříme předpis pro mazání.

Smazány budou tedy ty proměnné, které v souboru zůstanou. Soubor uložíme pod novým jménem, např. *delete.csv*.

V novém souboru je pak nutné provést následující úpravy:

- převzít hlavičku z původního souboru
- pokud je to nutné, nahradit oddělovače středníky „;“ čárkami „“,“

Soubor *delete.csv* pak musí v programu *notepad.exe* vypadat takto:

```
http://localhost:11112/,InnerId,NewestTimestamp,OldestTimestamp,Keys  
http://localhost:11112/,73a93805-f222-46eb-b8d3-e41d6f3db464,06.01.2022 15:50:00,06.01.2022 15:38:09,Comml  
http://localhost:11112/,f78ef2a7-a479-4312-a681-8b215b3cb477,06.01.2022 15:50:00,06.01.2022 15:38:09,Comml
```

Všechny proměnné, které v souboru předaném jako parametr aplikaci zůstanou, budou z databáze smazány. Smazání se provede spuštěním programu s parametrem *csvDeleteInput*:

```
.\ConsoleDBDotNet.exe --db http://localhost:11112/ --user user --  
password password --csvDeleteInput delete.csv
```

Program během chodu vypisuje identifikátory smazaných proměnných. Po skončení běhu jsou všechny proměnné, jejichž záznamy byly v souboru, smazány.

Seznam parametrů:

--db	URL databáze
--user	uživatel databáze s administrátorskými právy
--password	heslo tohoto uživatele
a vždy jeden z parametrů	
--csvStatsOutput	výpis proměnných v databázi do souboru
--csvDeleteInput	smazání proměnných definovaných v souboru.

9 Nastavení zápisu do databáze v SoftPLC

Podporované platformy jsou Windows Runtime a IPLC510. Ukládání dat přímo z PLC do databáze se konfiguruje takto:

U **Windows Runtime** nastavení probíhá pomocí konfigurátoru služby.

RcWare SoftPLC Configuration editor

Basic settings | SoftPLC Proxy settings | **RcWare DB settings** | AlarmServer settings | Advanced settings

Save variables to RcWareDb

RcWareDb credentials

User name: rc_user

Password:

RcWareDb server

URL: http://localhost:9876/RcWareDbAccess

request timeout [ms]: 60000

RcWareDb parameters

Variables to save: OPC

Save interval [sec] (0 means no periodical saving): 60

Use project time in place of value timestamp

SoftPLC RT Id: SplcTest

Values count in one request: 10

File buffer size [kB]: 1000

Buffer files path:

Decimals [-1 means unlimited]: 2

Performance settings | OK | Cancel

Save variables to RcWare DB: Zaškrtněte, pokud se data mají ukládat.

RcWare DB Credentials: Přístupová práva (uživatel musí být v databázi definován).

request timeout: Jak dlouho PLC čeká na odpověď z Merbon DB; po vypršení se spojení vyhodnotí jako nedostupné a data se začnou bufferovat do lokálních souborů.

Variables to save:

- All = všechny proměnné runtime; nepoužívejte;
- Used = „skoro“ všechny proměnné (např. nepoužité vstupy bloku se nebudou vzorkovat) – nepoužívejte;

- OPC = toto je správně, ale v SoftPLC IDE nutno označit požadované proměnné jako OPC!

Save interval: Interval vzorkování hodnot.

Use project time in place of value timestamp: Použije časovou značku z RTC PLC místo času komunikace (zobrazeného v tabulce Variables).

SoftPLC RTID: Tento řetězec je jedním z primárních klíčů a musí být unikátní pro každé PLC.

Values count in one request: Počet proměnných záležití na propustnosti linky. U pomalého připojení hodnotu zmenšete.

File buffer size (kB): PLC má volných asi 10 MB, takže 1MB je OK (bufferování umožní ukládat asi 20 000 vzorků na 1 MB, takže celkový čas bufferování závisí na počtu vzorků a vzorkovacím intervalu).

Buffer files path: Umístění bufferovacích souborů na disku.

Decimals: Počet desetinných míst, na které se zapisují hodnoty typu Real (Double).

U **IPLC510** se zápis do databáze nastavuje pomocí editace konfiguračního souboru `/var/opt/rcware/WinServiceRT/soft_plc.config`. Příslušná část s parametry pro nastavení zapisování do Merbon DB vypadá takto (nejdůležitější parametry jsou vyznačeny tučně):

```
<!-- Whether or not to save data to RcWareDb. -->
<add key="SAVE_DATA_TO_RCWARE_DB" value="true"/>
```

True = bude se zapisovat.

```
<!-- RcWareDb configuration mode (manual/automatic) -->
<add key="RCWARE_DB_AUTOMATIC_CONFIGURATION_MODE" value="false"/>
```

Nechte ve výchozím nastavení – neměnit.

```
<!-- Host of the RcWareDb. -->
<add key="RCWARE_DB_HOST" value="klecany.domat.cz"/>
```

Jméno nebo IP adresa serveru, kde Merbon DB běží. (Pokud se zadává jméno a ne IP adresa, zkontrolujte nastavení DNS v nastavení síťových parametrů PLC.)

```
<!-- Port of the RcWareDb. -->
<add key="RCWARE_DB_PORT" value="9876"/>
```

TCP port Merbon DB serveru. 9876 je výchozí hodnota.

```
<!-- Protocol of the RcWareDb. -->
<add key="RCWARE_DB_PROTOCOL" value="http"/>
```

Nechte ve výchozím nastavení – neměnit; další možností je https, ale musela by tak být nastavena i databáze.

```
<!-- In case of automatic RcWareDb configuration host list is used -->
<add key="RCWARE_DB_URL_LIST" value=""/>
```

Nechte ve výchozím nastavení – neměnit.

```
<!-- Save interval in seconds. -->
<add key="RCWARE_DB_SAVE_INTERVAL" value="300"/>
```

Interval vzorkování hodnot.

```
<!-- Size of file buffer (in kB's) used when there is no connection to RcWareDb. -->
```

```
<add key="RCWARE_DB_FILE_BUFFER_SIZE" value="1000"/>
```

PLC má volných asi 10 MB, takže 1MB je v pořádku (bufferování umožní ukládat asi 20 000 vzorků na 1 MB, tudíž celkový čas bufferování závisí na počtu vzorků a vzorkovacím intervalu).

```
<!-- SoftPlc Rt Id for RcWareDb data identification. -->
```

```
<add key="RCWARE_DB_SOFTPLC_RT_ID" value="FVE_DUBNO"/>
```

Tento řetězec je jedním z primárních klíčů a musí být unikátní pro každé PLC. Viz níže.

```
<!-- Variables that will be saved to RcWareDb. Available choice: All, OPC, Used -->
```

```
<add key="RCWARE_DB_VARIABLES_TO_SAVE" value="OPC"/>
```

- All = všechny proměnné runtime; nepoužívejte
- Used = „skoro“ všechny proměnné (např. nepoužité vstupy bloků se nebudou vzorkovat) – nepoužívejte
- OPC = toto je správně, ale v SoftPLC IDE nutno označit požadované proměnné jako OPC!

```
<!-- Access user name for connecting to RcWareDb. -->
```

```
<add key="RCWARE_DB_USER_NAME" value="rc_user"/>
```

Uživatel pro přístup do databáze.

```
<!-- Access password for connecting to RcWareDb. -->
```

```
<add key="RCWARE_DB_PASSWORD" value="rc_user"/>
```

Heslo tohoto uživatele.

```
<!-- Values count in one request. -->
```

```
<add key="RCWARE_DB_REQUEST_VARIABLES_COUNT" value="100"/>
```

Počet proměnných záleží na propustnosti linky. U pomalého připojení hodnotu zmenšete.

```
<!-- Buffer files path. -->
```

```
<add key="RCWARE_DB_BUFFER_FILES_PATH" value="/tmp"/>
```

Umístění bufferovacích souborů na souborovém systému PLC.

```
<!-- RcWareDb request timeout in ms. -->
```

```
<add key="RCWARE_DB_REQUEST_TIMEOUT" value="120000"/>
```

Jak dlouho PLC čeká na odpověď z Merbon DB; po vypršení se spojení vyhodnotí jako nedostupné a data se začnou bufferovat do souborů.

```
<!-- RcWareDb use RTC time of RT instead of value timestamp. -->
```

```
<add key="RCWARE_DB_USE_RTC_FOR_VALUE_TIMESTAMP" value="false"/>
```

Pokud se hodnota nemění (např. konstanta) – která časová značka se má použít.

Zapsané proměnné v databázi pak vypadají takto:

Variables

Refresh Filter Select all Unselect all Delete selected 1-5 of 5

Select	Keys	UserName	Actions
<input type="checkbox"/>	[{"IsKey":true,"Key":"GUID","Value":"6b6b4f53-1c5e-4e0b-b14b-a232f3a1b860"}, {"IsKey":false,"Key":"SoftPlcProject","Value":"hutska"}, {"IsKey":true,"Key":"SoftPlcRtId","Value":"SpIcTest"}, {"IsKey":false,"Key":"VariableName","Value":"bus.t_TUV"}]	rc_user	Edit
<input type="checkbox"/>	[{"IsKey":true,"Key":"GUID","Value":"89f86aad-5af3-4812-bb4c-2de14067f450"}, {"IsKey":false,"Key":"SoftPlcProject","Value":"hutska"}, {"IsKey":true,"Key":"SoftPlcRtId","Value":"SpIcTest"}, {"IsKey":false,"Key":"VariableName","Value":"bus.t_privod"}]	rc_user	Edit
<input type="checkbox"/>	[{"IsKey":true,"Key":"GUID","Value":"ffa122c3-799e-45dc-81e9-22baf6aa8463"}, {"IsKey":false,"Key":"SoftPlcProject","Value":"hutska"}, {"IsKey":true,"Key":"SoftPlcRtId","Value":"SpIcTest"}, {"IsKey":false,"Key":"VariableName","Value":"bus.t_zpatecka"}]	rc_user	Edit

Proměnné mají dva primární klíče: **GUID** (jedinečný identifikátor proměnné v rámci runtime) a **SoftPLCRtId** (unikátní označení runtime (PLC) v rámci databáze).

Při úpravách projektu v SoftPLC IDE obvykle nedochází ke změně primárních klíčů, GUID se nemění a tedy proměnná existuje po celou dobu – i po případné změně názvu nebo typu; aby došlo k vygenerování nového GUID, proměnná by musela být smazána a znovu založena.

10 Nastavení zápisu do databáze v Merbon IDE

V Merbon IDE je nutné především nastavit obecné parametry pro zápis do databáze:

... Parametry Merbon DB		-
Povoleno	True	■
URL	http://192.168.1.44:8686/spdba/save	■
Interval komunikace [min]	1	■
Uživatelské jméno	rc_user	■
Heslo	rc_user	■
Pozn.		□
... Parametry Proxy		+

Povoleno: True = bude se zapisovat do databáze

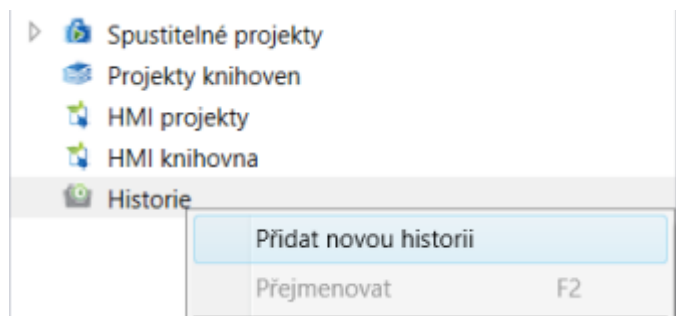
URL: Jméno nebo IP adresa serveru, kde běží Merbon Database Adapter (obvykle též server, na kterém běží Merbon DB). URL je nutné zadat úplně, *http://... :8686/spdba/save*

Interval komunikace: Jak často PLC odesílá data do databáze. Nejde o interval vzorkování! Data se dočasně ukládají v PLC a do databáze se odesílají jednou za čas, aby se Merbon Database Adapter zbytečně nepřetěžoval.

Uživatelské jméno / Heslo: Přístupové údaje uživatele s právy pro zápis (uživatel musí být v databázi definován).

Nezapomeňte také ve Vlastnostech PLC nastavit **Runtime ID** - unikátní označení PLC, které do databáze zapisuje. Obvykle je to název projektu a PLC, např. "ZS_Kolin1_kotelna". **Jde o jeden z primárních klíčů.** Délka textu nesmí přesáhnout 31 znaků.

Dále určíme, které hodnoty a jak často se budou zapisovat. K tomu slouží tzv. Historie. Nezapomeňme, že **pro definici historie musí být sestava přepnuta do Plného módu.**



Ve vlastnostech historického bloku se definuje interval vzorkování dat. Do bloku se pak přiřazují proměnné, které mají být tímto způsobem vzorkovány. V PLC může být více historií s různými frekvencemi vzorkování. Na celkovém počtu vzorkovaných proměnných a frekvenci jejich vzorkování závisí délka historie, bufferovaná v PLC. Interval komunikace s databází musí tedy být několikanásobně kratší, než délka bufferované historie, aby nedošlo ke ztrátě dat. Podrobnosti k nastavení jsou v nápovědě Merbon IDE, *Nastavení ukládání dat do databáze*.

Nastavení je součástí konfigurace PLC. Nakonec je tedy nutné nahrát konfiguraci do PLC.

Při přejmenování nebo změně typu proměnné se změní **CommUid** a s největší pravděpodobností i **GUID** (které je primárním klíčem), což má za následek přerušení datové řady. S rekonstrukcí si musí poradit klient například tím, že proměnné identifikuje podle klíče **VariableName**.