

ProviewR

OPEN SOURCE PROCESS CONTROL



Handbok i Processhistorik

Copyright (C) 2005-2026 SSAB EMEA AB

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

2 Inledning

En allt viktigare del av automationssystemens uppgift är att lagra processdata. Möjligheten att lagra stora mängder data har gett upphov till en rad ny funktionalitet gällande utveckling och optimering av processer, prediktivt underhåll, framtagning av modeller för simulering etc.

Det här dokumentet beskriver lagring av process-historik i ProviewR och hur denna konfigureras.

3 Process data

Historik lagring av processdata innebär att signaler och andra data cykliskt lagras i en databas, och därifrån kan hämtas upp för att visas i kurvor eller analyseras i andra sammanhang, t ex för prediktivt underhåll eller processutveckling.

Lagringen konfigureras med objekten SevHist och SevHistObject. SevHist konfigurerar lagring av ett attribut, och SevHistObject av ett helt objekt. I SevHist-objekten ställer man in hur ofta värdet ska lagras och hur länge det ska sparas i databasen.

Lagringen hanteras av två processer, en klient-process, `rt_sevhistmon`, som samlar ihop data och skickar iväg till en server-process, `sev_server`, som tar emot data och lagrar det i databasen. Klient-processen kan skicka data till flera olika server-processer, och server-processen kan ta emot data från flera olika klient-processer. Klient och server-processen kan gå på samma nod eller i olika noder. För test och felsökning kan man starta en server-process på en process station som lagrar data från den egna noden. För större datamängder med lagring över flera år, skapar man istället en dedikerad lagringsstation som lagrar data från flera noder.

3.1 Klient

`rt_sevhistmon` är klient-processen som samlar ihop process-värden i en node och skickar till servern. Den konfigureras med ett SevHistMonitor objekt i nodhierarkin. Under SevHistMonitor-objektet lägger man SevHistThread-objekt, som innehåller cykeltid och server nod. Varje SevHist och SevHistObject-objekt kopplas till ett thread-objekt och kommer därmed att lagras på den nod, och med den cykeltid, som anges i thread-objektet.

Thread-objektet innehåller även ett ServerThread attribut, med vilket man kan rikta lagringen till en specifik thread i server-processen. Detta kan användas för att fördela lasten mellan olika threads i server-processen.

3.2 Server

Server-processen, `sev_server`, tar emot data från klient-processer och lagrar dem i databasen. Finns inte databasen skapar den databasen med de tabeller som behövs. Sedan sker det en handskadning med klient-processerna som skickar över data på de attribut och objekt som ska lagras. Sedan startar den cykliska överföringen av process data från klient-processerna som lagras i databasen.

Server-processen kan även svara förfrågan om data för ett attribut i ett specifikt tidsintervall.

Data kan lagras i olika typer av databaser, MariaDB (eller MySQL), Sqlite och HDF5.

MariaDB/MySQL

MariaDB är en klon av MySQL som skapades när MySQL övertog av Oracle. De ska vara kompatibla och kan bytas ut mot varandra. I senare Linux releaser är ofta MariaDB installerat. Många

konfigureringsalternativ i ProivewR är fortfarande namnade MySQL men ska även användas för MariaDB.

MariaDB bör användas för större databaser och för databaser för permanent lagring.

Sqlite

Sqlite är en liten, snabb databas som inte kräver installation av en server. Kan användas för mindre databaser för test och felsökning.

HDF5

HDF5 är ett filformat för att lagra stora mängder data. Avsaknad av journalering gör att det inte bör användas för permanent lagring.

Larm och händelser

Lagring av larm och händelser görs med ett SevHistEvent-objekt. Objektet innehåller en SelectList där man anger hierarkier för larm som ska lagras.

4 Konfigurering

Lagringen konfigureras av SevHist och SevHistObject objekt för varje attribut eller objekt som ska lagras. Vidare konfigureras klient-processen med SevHistMonitor och SevHistThread objekt och server-processen med ett SevServer objekt.

4.1 SevHist

Attribut som ska lagras i den historiska databasen konfigureras med SevHist objekt. SevHist objekten placeras i en rot eller sub volym, och attributet anges i Attribute i SevHist objektet. SevHist objektet läggs normalt under det objekt som ska lagras. Om objektet har ett ActualValue attribut, läggs detta automatiskt in i SevHist objektet.

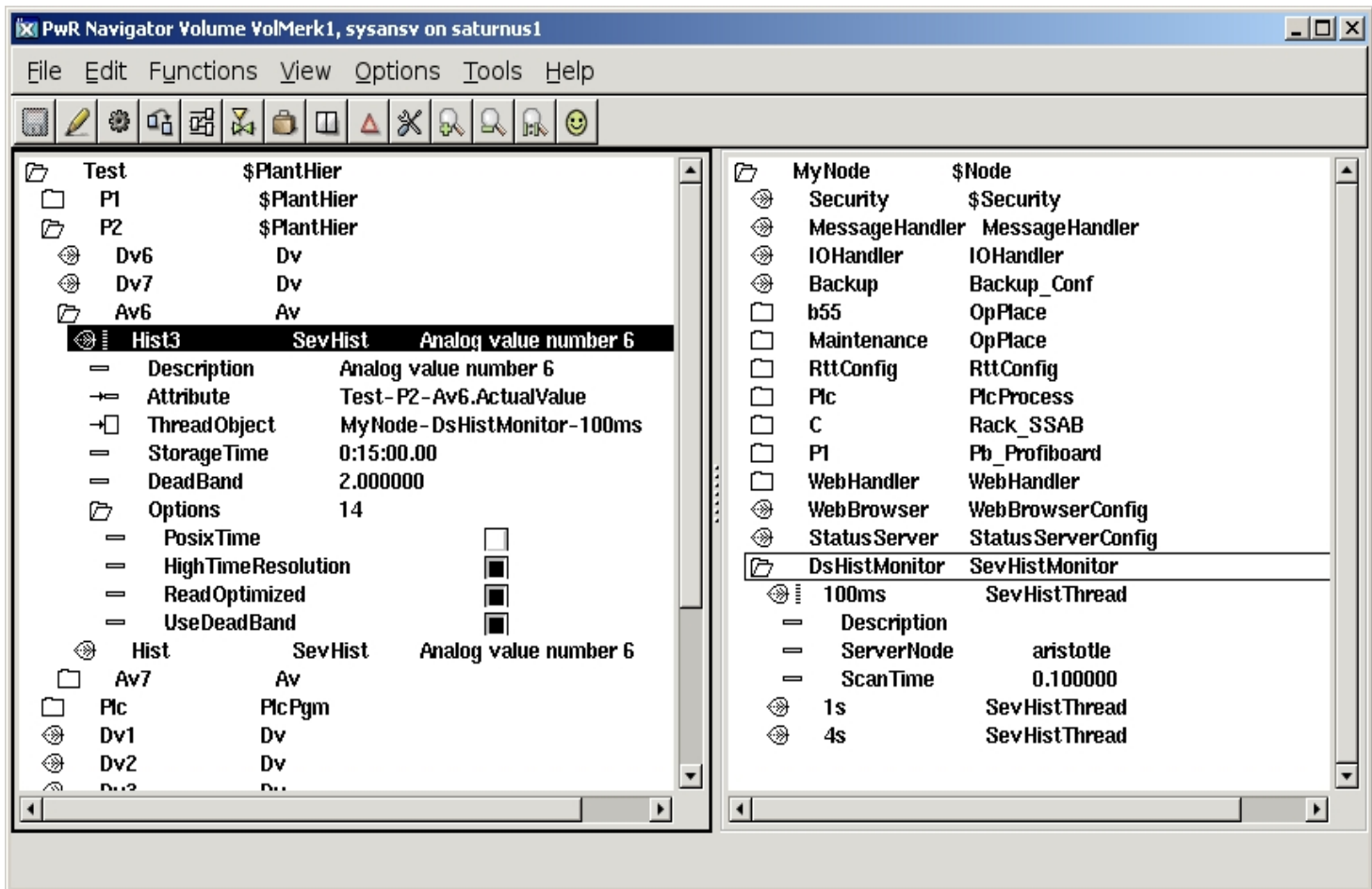


Fig Konfigurering i en rootvolym med SevHist, DsHistMonitor och SevHistThread objekt

4.2 SevHistObject

Till skillnad från SevHist, som lagrar ett enskilt attribut, lagrar SevHistObject samtliga attribut i ett objekt i en tabell. Det rekommenderas att man skapar en specifik klass för användandet, eftersom existerande klasser ofta innehåller attribut som ej behöver lagras.

4.3 Server-trådar

Genom att konfigurera tråddning av serverprocessen kan man avsevärt öka prestanda då lasten fördelas på flera olika trådar.

Tråddning är implementerat för MariaDB/MySQL.

Trådarna konfigureras i klienten genom att ange tråddnummer i SevHistThread-objekten. En tråd kan numreras med ett godtyckligt positivt nummer, och alla SevHistThread-objekt med samma tråddnummer kommer att hanteras av samma tråd. I SevServer objektsbild kan man se belastningen för de olika servertrådarna, och genom att ändra tråddnummer i SevHistThread-objekten se till att ingen servertråd blir överbelastad.

Servertrådar måste även konfigureras för server-processen genom att sätta attributet UseServerThreads i SevServer-objektet till 1.

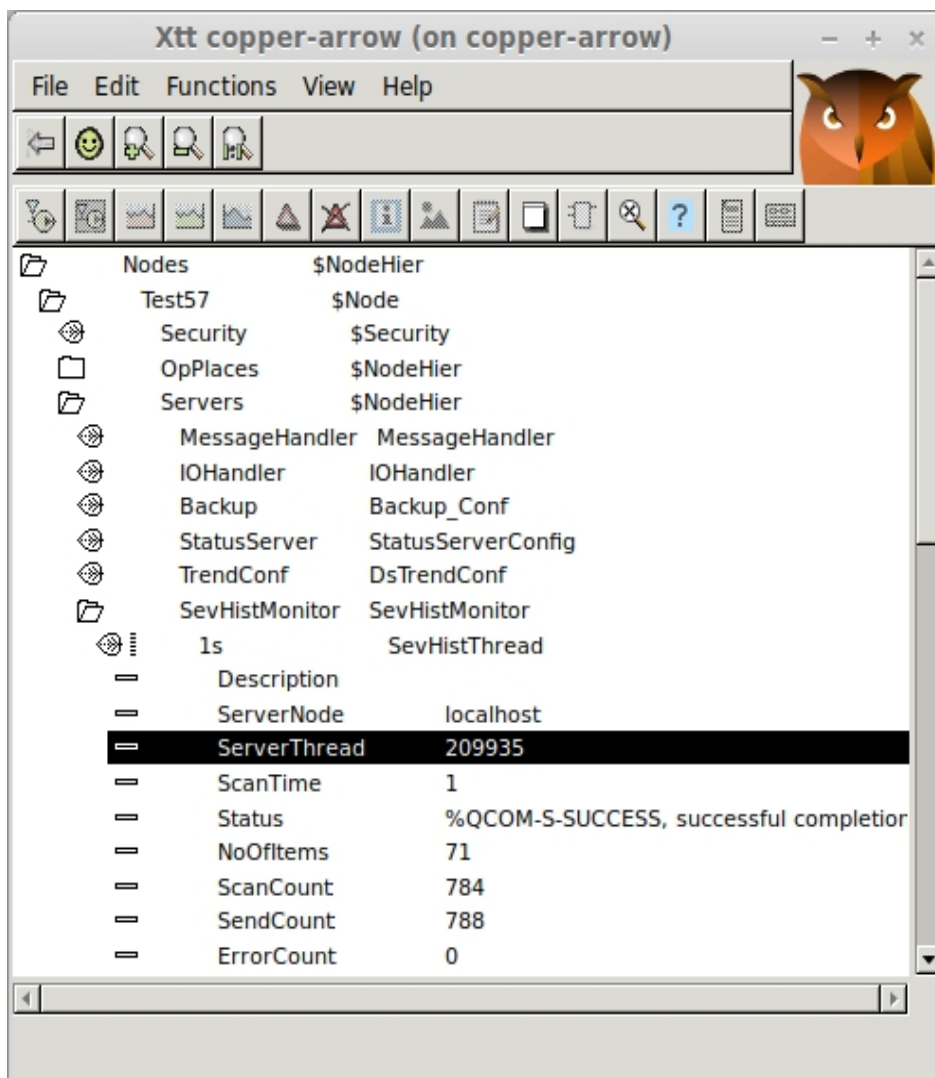


Fig Konfigurering av tråddnummer i SevHistThread-objektet.

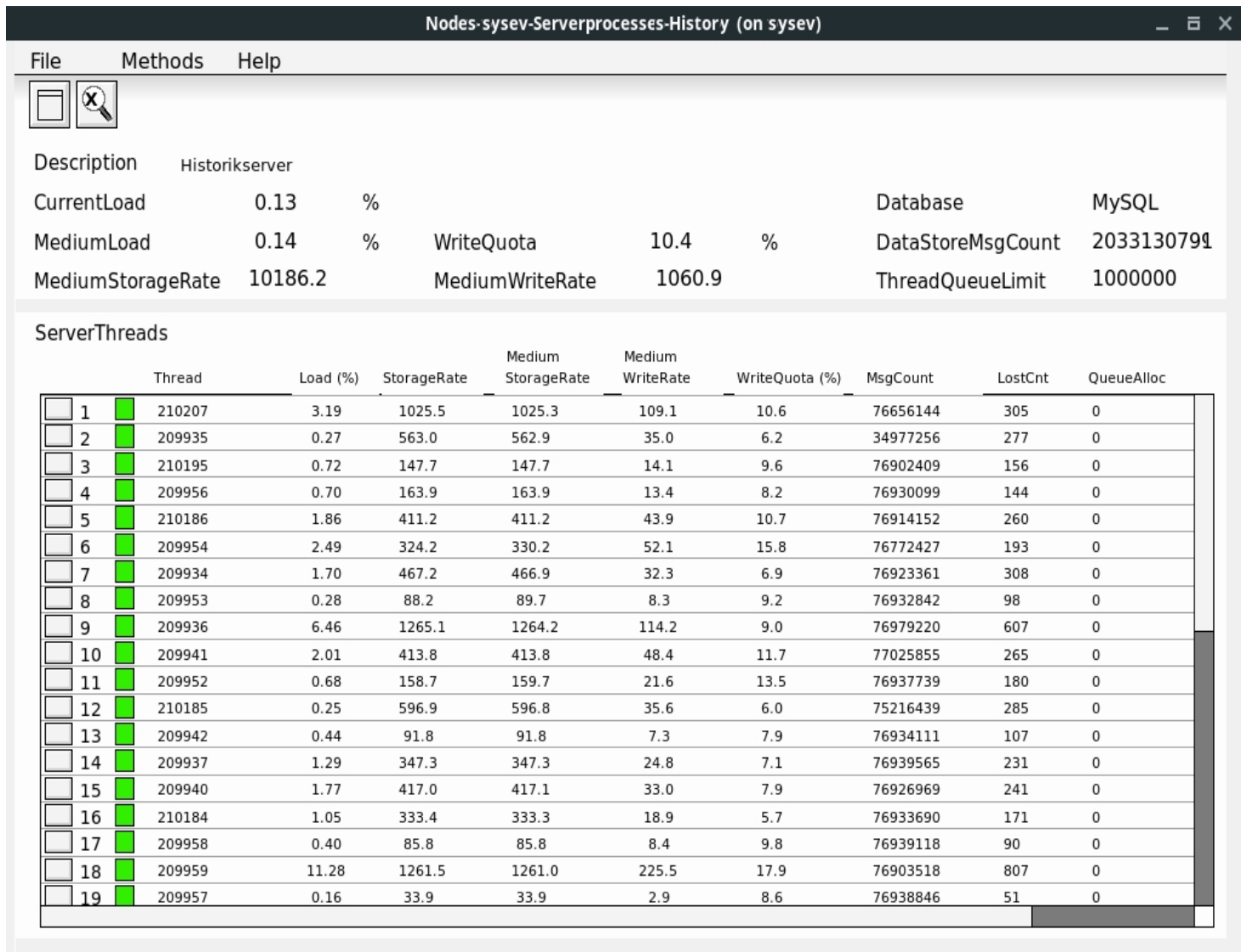


Fig Objektsbild för SevServer objektet som visar servertrådar.

4.4 Dödband

Dödband kan konfigureras på analog, digital och heltals-signaler och innebär att det krävs en viss ändring av värdet för att ett nytt värde ska lagras i databasen. Genom att sätta ett dödband kan diskutrymmet som krävs för att lagra en signal reduceras avsevärt.

Dödband konfigureras genom att sätta Deadband och ReadOptimized i Options för SevHist objektet, och ange storleken på dödbandet i Deadband.

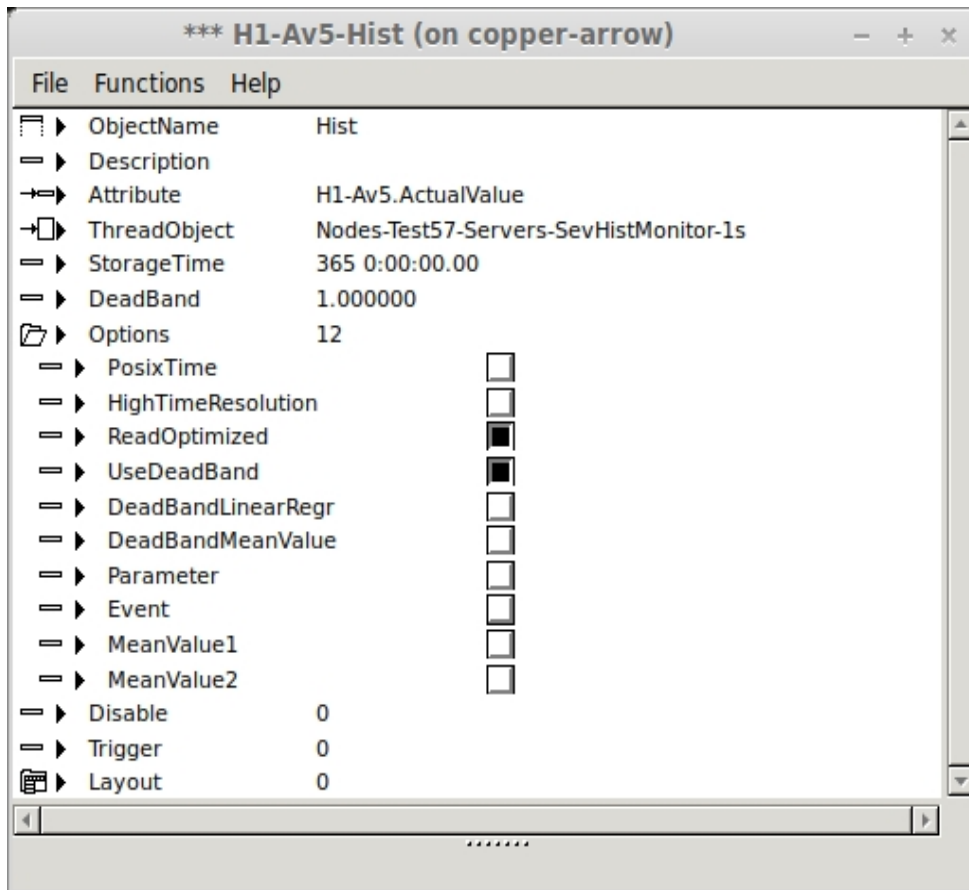


Fig Konfiguration av dÅ¶dband

FÅ¶r dÅ¶dband pÅ¶ digital signal, sÅ¶tt Deadband till 0.5.

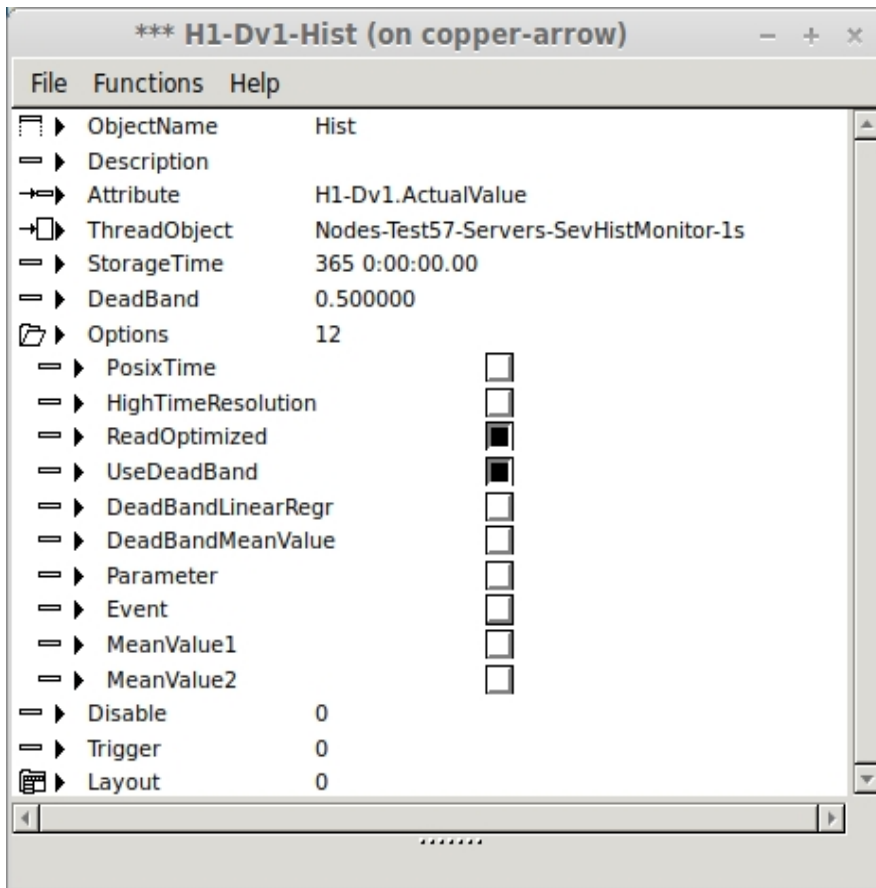


Fig Konfiguration av dÅ¶dband pÅ¶ digital signal

4.5 DÅ¶dband med linjÅ¶ regression

Det hÅ¶r Å¶r ett tvÅ¶dimensionellt dÅ¶dband som Å¶ven fungerar pÅ¶ ramper. Med linjÅ¶ regression berÅ¶knas en rÅ¶tt linje frÅ¶n senast lagrade vÅ¶rde, och sÅ¶ lÅ¶nge inget nytt vÅ¶rde avviker mer Å¶n dÅ¶dbandet frÅ¶n linjen lagras inget nytt vÅ¶rde. Det hÅ¶r reducerar ytterligare det diskutrymme som krÅ¶vs.

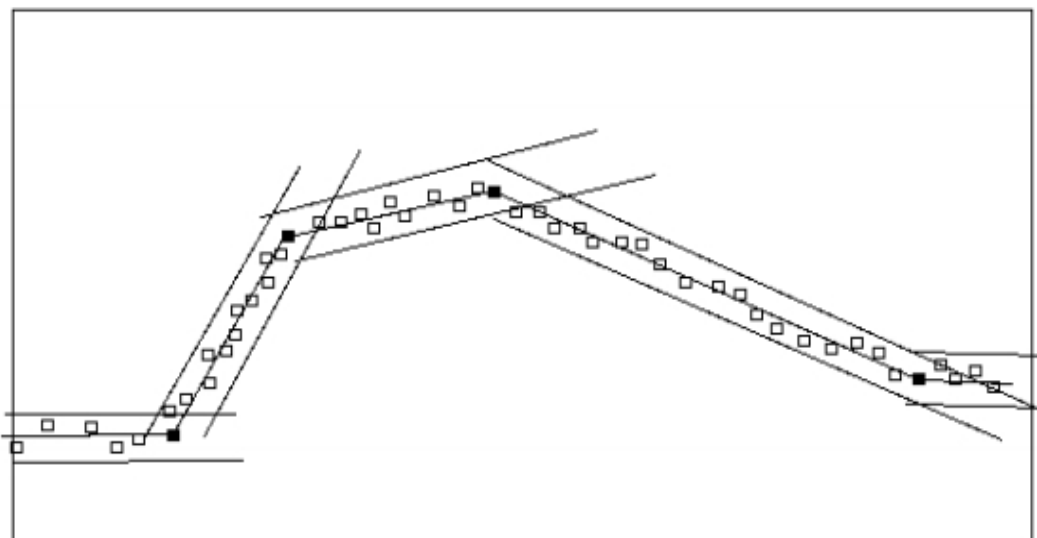


Fig DÅ¶dband med linjÅ¶ regression

Dödband med linjär regression konfigureras genom att Deadband, DeadbandLinearRegr och ReadOptimised sätts i Options i SevHist objektet, och storleken på dödbandet anges i Deadband.

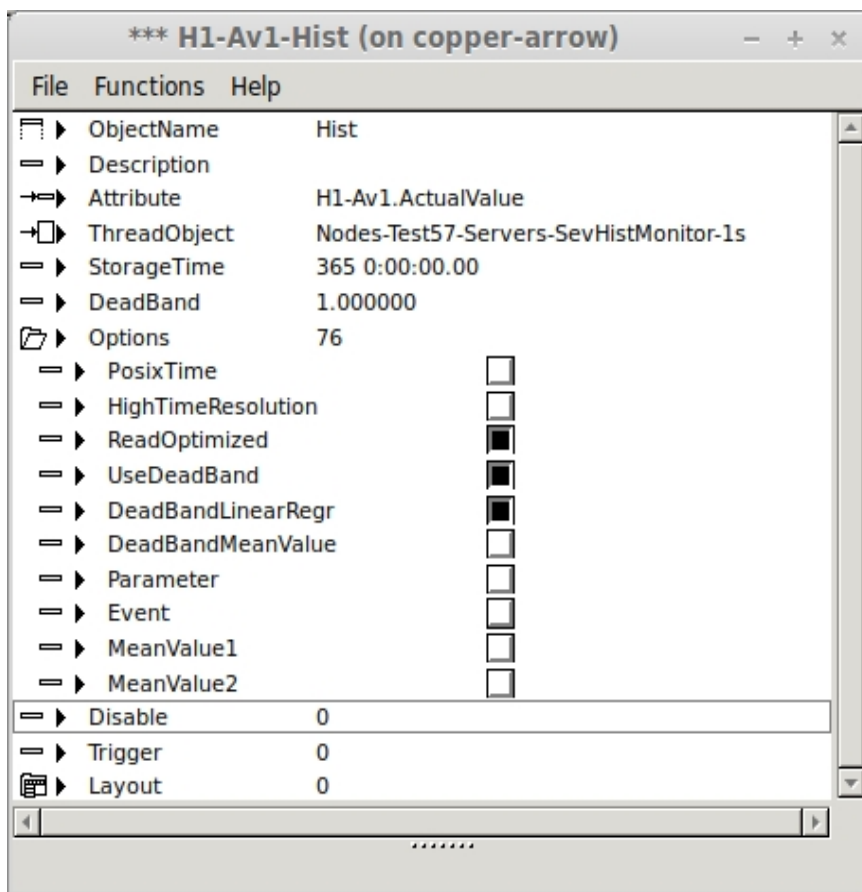


Fig Konfiguration av dödband med linjär regression

4.6 Händelsestyrd logging

Genom att sätta Event i Option i SevHist objektet, kan man styra när lagringen sker. När Trigger i SevHist objektet sätts till 1 skickas aktuellt värde till server processen, Trigger tas ned när värdet är skickat.

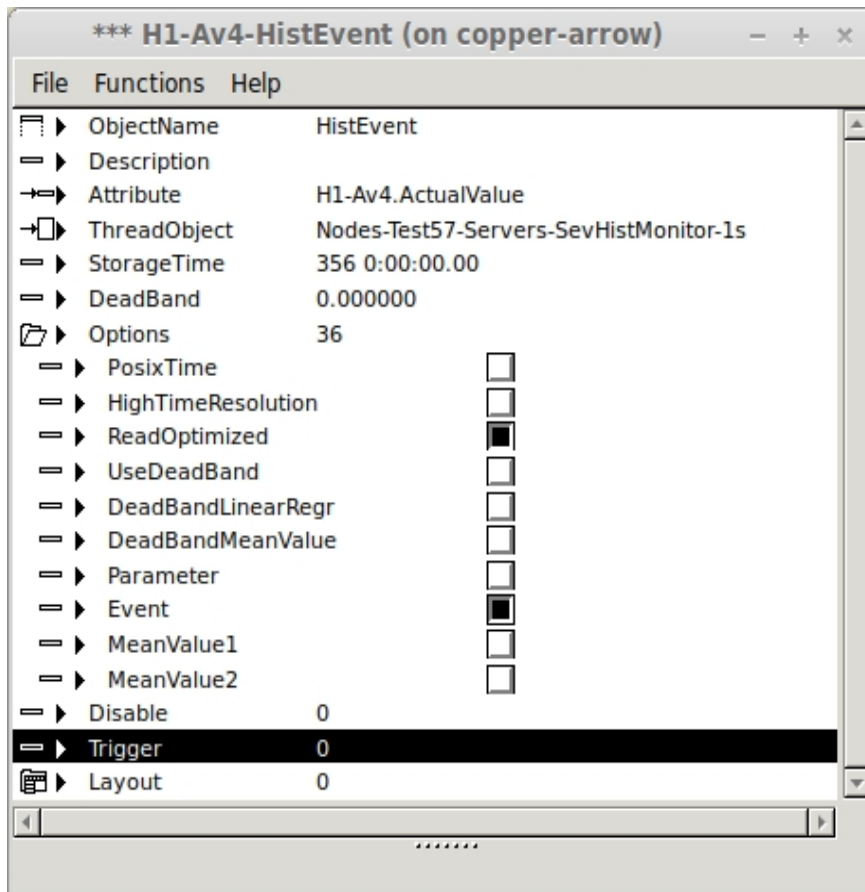


Fig Konfigurering av händelsestyrd lagring

4.7 Medelvärdesberäkning av lagrade signaler

Man kan ange att server-processen ska utföra en medelvärdesberäkning en lagrad signal genom att sätta MeanValue1 eller MeanValue2 i Options i SevHist-objektet.

Servern gör medelvärdesberäkning med två olika tider som anges i SevServer-objektet, attributen MeanValueInterval1 och MeanValueInterval2, och i Options väljer man med vilken av de här tiderna beräkningen ska utföras.

Medelvärdet visas i item-trädet, och kan därifrån refereras från Ge grafer och applikationer med suffixet '.__MeanValue', t ex 'pwrNode-sev-H1-Av1.ActualValue.__MeanValue'.

4.8 Item-träd

Varje lagrad signal i serverns representeras av ett item. Alla item visas i ett item-träd som läggs under pwrNode-sev i realtidsdatabasen. I item-trädet visas signalerna i dess ursprungliga hierarki, och här visas även det sista mottagna värdet. Man kan visa mer information om ett item med Shift+Klick på värdet, eller Shift+Pil höger på tangentbordet.

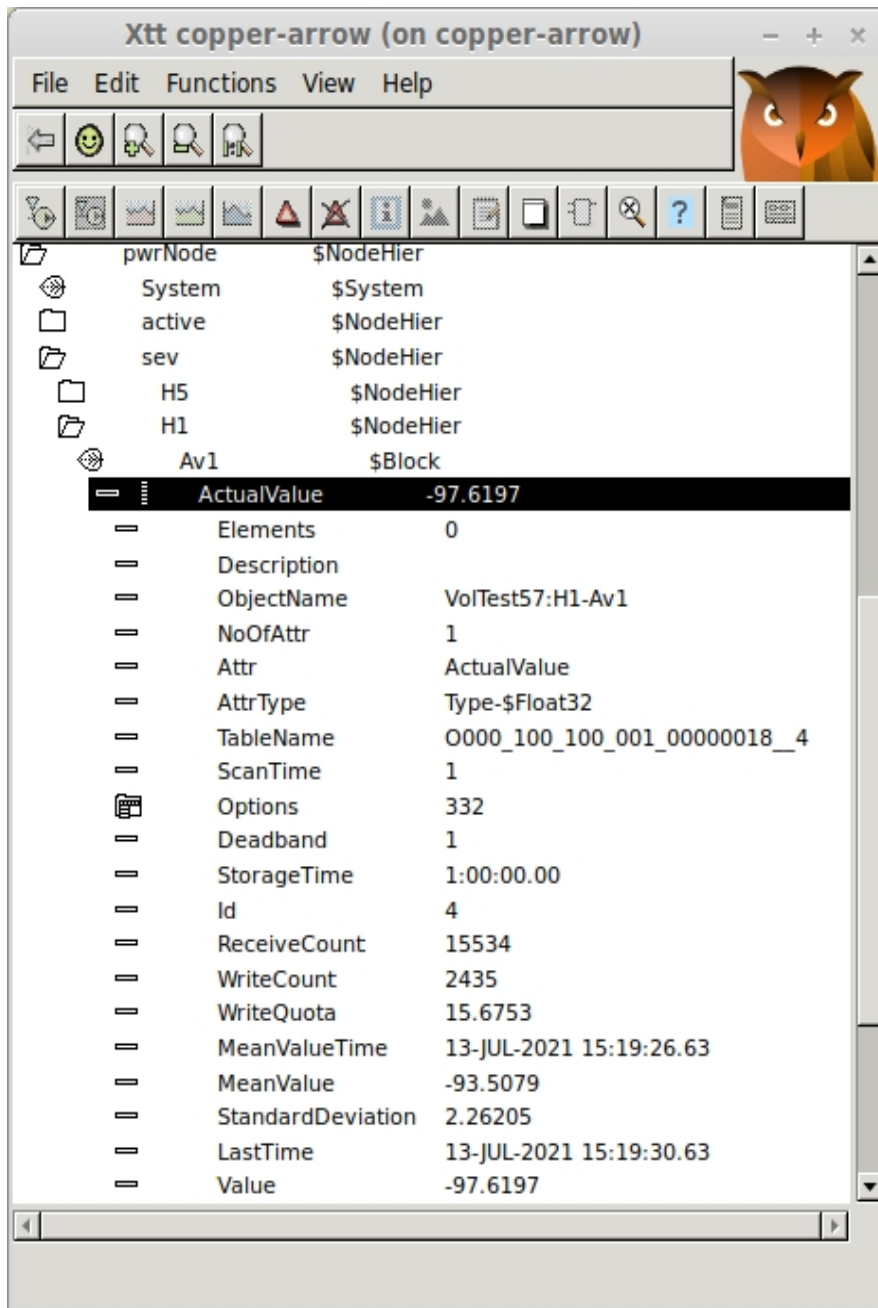


Fig Ett item i item-trÄdet

4.9 Montering av item-trÄdet

ItemtrÄdet byggs upp av objekt av typen \$Block och \$BlockAttribute fÄr att i nÄgon mÄn Äterskapa den ursprungliga hierarkin och objektsstrukturen. Genom att montera hierakierna under pwrNode-sev pÄ top-nivÄn kan man Äterskapa den ursprungliga objekt- och attributnamnen och anvÄnda dessa vid referenser i grafer och applikationer.

Monteringen gÄrs med monteringsobjekt av typen \$MountDynObject eftersom det Är dynamiska objekt som monteras.

NÄr monteringen Är gjord, kan signaler refereras med sitt ursprungliga namn. Det gÄr det enklare att referera till signalerna i grafer och applikationer.

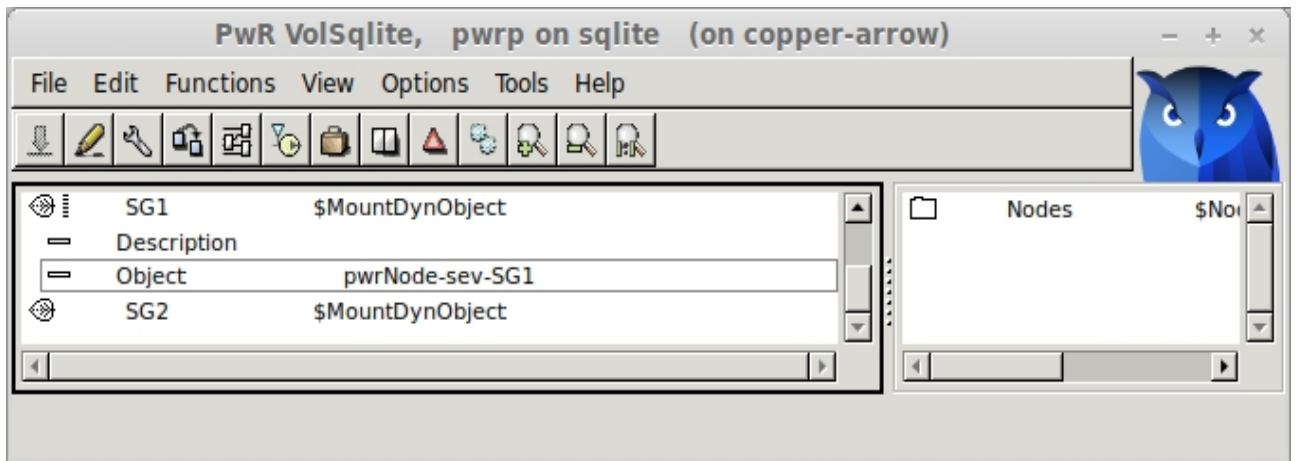


Fig Montering av hierarki i itemtrÅdet

4.10 Referera data i item-trÅdet

Data i ett item kan refereras med suffixet '.___dataname"', dvs medelvÅrdet fÅr Av1 ovan kan refereras med

pwrNode-sev-H1-Av1.ActualValue.___MeanValue

Om det finns en montering pÅ H1 kan pwrNode-sev strykas och referensen gÅrs med

H1-Av1.ActualValue.___MeanValue

Man kan hÅrta vÅrden och lÅgga upp prenumerationer frÅn c++ och Python kod. I Ge grafer lÅgger men pÅ typen som suffix

H1-Av1.ActualValue.___MeanValue##Float32

Exempel pÅ andra vÅrden som kan refereras

H1-Av1.ActualValue

Senast mottagna vÅrde.

H1-Av1.ActualValue.___TableName

Namn pÅ tabell dÅr historiken lagras.

H1-Av1.ActualValue.___StandardDeviation

Standardavvikelse vid medelvÅrdesberÅkningen.

H1-Av1.ActualValue.___LastTime

Tid fÅr senast mottagna vÅrde.

4.11 Plcprogrammering

Itemdata kan Åven hÅrmas upp i plc-programmet. Eftersom data ligger i dynamiska objekt mÅste man anvÅnda GetExt-objekt, t ex GetExtFloat32, GetExtBoolean etc.

I exemplet nedan adderas momentanvÅrden fÅr H1-Av1.ActualValue och medelvÅrden fÅr H1-Av2.ActualValue och lÅggs i H3-Av3 som År ett lokalt objekt i server-noden.

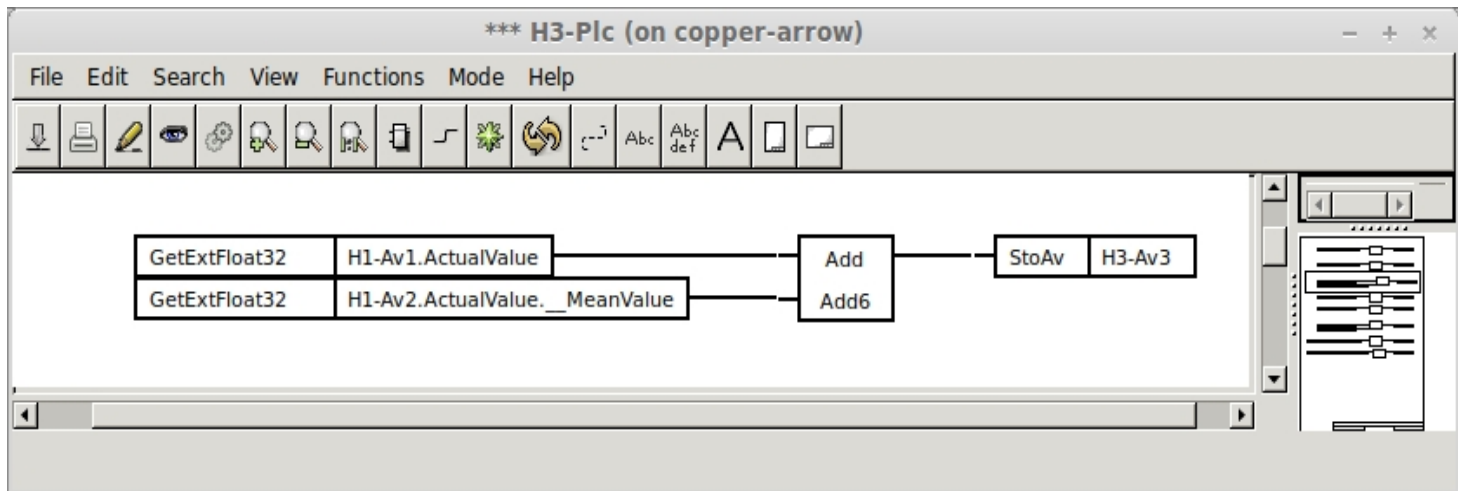


Fig Plc kod med item data

4.12 Sev export

Sev export g r det m jligt att exportera data till sev servern som inte beh ver lagras i historik-databasen, men som  nd  ska vara tillg ngliga f r att presenteras i bilder och rapporter.

Exporten konfigureras med SevExport objekt. Ett SevExport objekt exporterar ett attribut. Liksom f r SevHist objekten  r det rt_sevhistmon som samlar ihop attributen och skickar dem vidare till server noden. I det h r fallet  r mottagarprocessen sev_import som konfigureras med ett SevImportServer objekt i sev server noden. sev_import l gger in v rdet i item-tr det varifr n det kan visas upp i bilder eller anv ndas i plc-programmet.

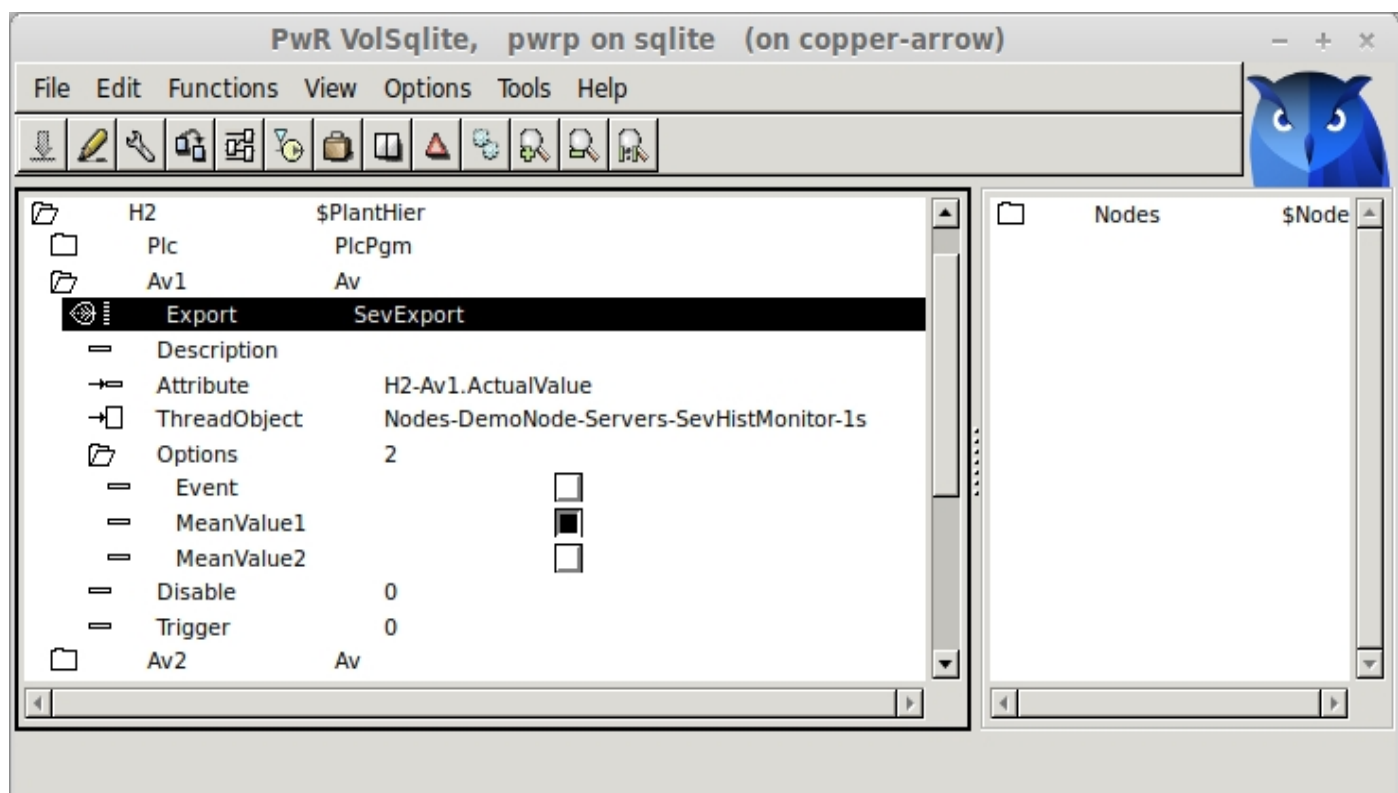


Fig Konfigurering av export med SevExport

Medelvärdesberäkning

Även för exporterade attribut kan man konfigurera en medelvärdesberäkning genom att sätta MeanValue1 eller MeanValue2 i Options i SevExport-objektet.

Servern gör medelvärdesberäkning med två olika tider som anges i SevImportServer-objektet, attributen MeanValueInterval1 och MeanValueInterval2, och i Options väljer man med vilken av de här tiderna beräkningen ska utföras.

Medelvärdet visas i item-trädet, och kan därifrån refereras från Ge grafer och applikationer med suffixet '.__MeanValue', t ex 'pwrNode-sev-H2-Av2.ActualValue.__MeanValue'.

Händelsestyrd export

Genom att sätta Event i Options SevExport objektet kan man styra när exporten av värdet sker. Ävenverförringen gör när Trigger i SevExport objektet sätts till 1. Trigger tas ner automatiskt när Ävenverförringen har utförts.

5 Databasens interna struktur

Databasen namnges 'pwrp__systemnamn', och innehåller tabellerna 'items', 'objectitems', 'objectitemattributes', 'sev_stat' och 'sev_version'. Dessutom finns en tabell för varje attribut eller objekt som lagras.

items

Items tabellen innehåller information från alla SevHist och SevHistEvent som lagras.

Namn	Typ	Beskrivning
id	integer	Identitet.
tablename	varchar	Namn på tabell med historisk data.
vid	integer	Volymsidentitet för lagrat objekt.
oix	integer	Objektindex för lagrat objekt.
oname	string	Objektsnamn.
aname	string	Attributnamn. Om ett SevHistEvent item 'Events'.
uptime	datetime	Starttid.
cretime	datetime	Skapande tid för item.
storagetime	integer	Tid efter vilken data kan raderas i sekunder.
deadband	float	Dödband.
options	integer	Options.
scantime	float	Scantid.
description	string	Beskrivning av lagrat objekt.
vtype	integer	Attributtyp.
vsize	integer	Attributets storlek i bytes.
unit	string	Attributets enhet

objectitems

Objectitems-tabellen innehåller information från alla SevHistObject som lagras.

Namn	Typ	Beskrivning
id	integer	Identitet.
tablename	varchar	Namn på tabell med historisk data.
vid	integer	Volymsidentitet för lagrat objekt.
oix	integer	Objektindex för lagrat objekt.
oname	string	Objektsnamn.
aname	string	Används ej.
uptime	datetime	Starttid.
cretime	datetime	Skapande tid för item.
storagetime	integer	Tid efter vilken data kan raderas i sekunder.
deadband	float	Dödband.
options	integer	Options.
scantime	float	Scantid.
description	string	Beskrivning av lagrat objekt.

objectitemattributes

För att få en fullständig beskrivning av SevHistObject, krävs förutom innehållet i objectitems, information om de attribut som varje objekt innehåller. Informationen lagras i objectitemattributes tabellen med en rad för varje attribut.

Namn	Typ	Beskrivning
tablename	varchar	Namn på tabell med historisk data.
attributename	string	Attributnamn.
attributeidx	integer	Attributindex.
attributetype	integer	Attributtyp.
attributesize	integer	Attributets storlek i bytes.

Historiktabeller för enskilda attribut

Tabeller för lagring av processvärden konfigurerade med SevHist-objekt.

Namn	Typ	Beskrivning
id	integer	Identitet.
time	datetime or integer	Tidpunkt.
ntime	integer	Nanosekunder om hög tidsupplösning är konfigurerad.
value	godtycklig typ	Processvärde.

Historiktabeller för hela objekt

Tabeller för lagring av processvärden för hela objekt konfigurerade med SevHistObject-objekt.

Namn	Typ	Beskrivning
sev__id	integer	Identitet.
sev__time	datetime or integer	Tidpunkt.
sev__ntime	integer	Nanosekunder om hög tidsupplösning är konfigurerad.
'attributenamn1'	godtycklig typ	Processvärde för första attributet i objektet.
'attributenamn2'	godtycklig typ	Processvärde för andra attributet i objektet.
...		

Historiktabeller för larm och händelser

Tabeller för lagring av händelser konfigureras med ett SevHistEvent-objekt.

Namn	Typ	Beskrivning
time	integer	Tid.
ntime	integer	Nanosekunder.
eventtype	integer	Eventtyp.
eventprio	integer	Eventprioritet.
eventid_nix	integer	Event identitet, nix.
eventid_birhtime	integer	Event identitet, birthtime.
eventid_idx	integer	Event identitet, idx.
supobject_vid	integer	Attrref för övervakningsobjektet, vid.
supobject_oix	integer	Attrref för övervakningsobjektet, oix.
supobject_offset	integer	Attrref för övervakningsobjektet, offset.
supobject_size	integer	Attrref för övervakningsobjektet, size.
eventtext	varchar	Eventtext.
eventname	varchar	Eventnamn.
eventstatus	integer	Eventstatus.

sev_stat

sev_stat innehåller statistik. Numera finns denna information även i SevServer-objektet.

Namn	Typ	Beskrivning
current_load	float	Momentanvärde av belastningen i procent.
medium_load	float	Medelvärde av belastningen i procent.

storage_rate	float	Antal lagrade item per sekund.
medium_storage_rate	float	Medelvärde för antal lagrade item.
datastorage_msg_cnt	integer	Antal lagrings-meddelanden sedan startup.
dataget_msg_cnt	integer	Antal förfrågningar om historisk data sedan startup.
items_msg_cnt	integer	Antal meddelanden för items.
eventstore_msg_cnt	integer	Antal lagrings-meddelanden för larm och händelser.

sev_version

Innehåller aktuell version av sev-databasen. Sev versionen räknas upp när databas-strukturen ändras och följer inte versionen för ProviewR utgåvor.

version	integer	Aktuell version.
---------	---------	------------------

6 Databaser

Den mest använda databasen är MariaDB, men det finns även stödd för SQLite och HDF5.

6.1 MariaDB/MySQL

MariaDB är den databas som är mest använd i ProviewR. Den har också full funktionalitet när det gäller databand och server-trådar.

Konfigureringen sker genom att sätta Database till MySQL i SevServer-objektet. Dessutom krävs att man installerar och startar mariadb-server i server-noden. Man måste också skapa användaren pwrp

För MariaDB

```
mysql
MariaDB> create user pwrp@localhost;
MariaDB> grant all privileges on *.* to pwrp@localhost;
```

För MySQL

```
mysql
mysql> grant all privileges on *.* to pwrp@localhost;
```

Den rekommenderade databas-motorn InnoDB är default i senare versioner och ska användas.

För små databaser kan man använda standard-konfigurationen av MariaDB, men för dedicerade databas-noder finns det några inställningar som bör göras i filen /etc/mysql/my.cnf

innodb_file_per_table

Ur underhållssynpunkt är det en fördel att ha varje tabell i en separat fil. Man kan därför återfå diskutrymme för borttagna signaler.

innodb_log_file_size

Transaktioner skrivs först i log-filer, innan de förs in i data-filerna. Med större log-filer kan skrivningen i data-filerna optimeras genom att göras sekventiellt, och andra sidan tar en recovery av databasen längre tid. Storleken på log-filerna bör normalt utökas från default-värdet.

innodb_buffer_pool_size

Minne som inte används av operativsystemet, applikationer eller MariaDB bör allokeras till buffer poolen. Såg att vi har 16 Gb minne, 2 Gb används av operativsystemet, innodb_log_file_size är 0.5 Gb som det bör finnas utrymme för i cachen, ProviewR tar 0.5 Gb, lämnar 1 Gb till förvrigt så återstår 12 Gb som kan konfigureras för innodb_buffer_pool_size.

query_cache_type och query_cache_size

Tabellerna förändras kontinuerligt så det finns ingen anledning att cache resultatet av förfrågningar. Sätt dessa till 0.

Exempel på konfigurationsfil

```
[mysqld]
innodb_log_file_size      = 512M
innodb_buffer_pool_size  = 12G
innodb_file_per_table     = 1
innodb_flush_method       = O_DIRECT
query_cache_type          = 0
query_cache_size          = 0
```

Underhåll och felsökning

Med 'mysql' kan databasen inspekteras och modifieras. Namnet på databasen är 'pwrp__systemname', t ex 'pwrp__test57'. Här är ett exempel på hur man kan titta på items-tabellen och på data för ett enskilt item.

```
> mysql
MariaDB> use pwrp__test57;
```

```
MariaDB> select oname,tablename from items;
```

```
+-----+-----+
| oname                | tablename                |
+-----+-----+
| VolTest57:H5-Av1     | 0000_100_100_001_000000ab__1 |
| VolTest57:H5-Dv1     | 0000_100_100_001_000000b7__2 |
| VolTest57:H5-Dv2     | 0000_100_100_001_000000be__3 |
| VolTest57:H1-Av1     | 0000_100_100_001_00000018__4 |
| VolTest57:H1-Av2     | 0000_100_100_001_0000002e__5 |
| VolTest57:H1-Av3     | 0000_100_100_001_0000009f__6 |
| VolTest57:H1-Dv1     | 0000_100_100_001_00000056__7 |
| VolTest57:H1-Iv1     | 0000_100_100_001_0000005d__8 |
| VolTest57:H1-Iv2     | 0000_100_100_001_0000005e__9 |
| VolTest57:H1-Av4     | 0000_100_100_001_0000013d__10 |
| VolTest57:H18-SevHistEvents | 0000_100_100_001_00000ef3__11 |
| VolTest57:H1-Av5     | 0000_100_100_001_00000f96__12 |
+-----+-----+
```

```
MariaDB> select time,value from 0000_100_100_001_00000018__4 order by id desc limit 10
```

```
+-----+-----+
| time                | value                    |
+-----+-----+
| 2021-07-14 14:21:32 | 83.7721                  |
| 2021-07-14 14:21:26 | 97.9997                  |
| 2021-07-14 14:21:22 | 99.8689                  |
| 2021-07-14 14:21:18 | 95.4521                  |
| 2021-07-14 14:21:13 | 81.552                   |
| 2021-07-14 14:21:07 | 54.4918                  |
| 2021-07-14 14:20:50 | -47.3081                 |
| 2021-07-14 14:20:42 | -83.9247                 |
| 2021-07-14 14:20:36 | -98.0545                 |
| 2021-07-14 14:20:32 | -99.8544                 |
| 2021-07-14 14:20:28 | -95.3687                 |
+-----+-----+
```

```

| 2021-07-14 14:20:23 | -81.3886 |
| 2021-07-14 14:20:17 | -54.2567 |
| 2021-07-14 14:20:02 | 36.1479 |
| 2021-07-14 14:19:53 | 80.5102 |
| 2021-07-14 14:19:47 | 96.7008 |
| 2021-07-14 14:19:43 | 99.9982 |
| 2021-07-14 14:19:42 | 99.8383 |
| 2021-07-14 14:19:38 | 95.2808 |
| 2021-07-14 14:19:33 | 81.2193 |
+-----+-----+

```

6.2 Sqlite

Sqlite kr ver inte installation av n gon ytterligare serverprocess. D remot  r funktionen begr nsad. St rd f r server tr dar och d dband med linj r regression saknas.

Sqlite konfigureras genom att s tta Database i SevServer-objektet till Sqlite.

Databas-filen  ggs p  \$pwrp_db med namnet pwrp__'systemname'.dbsqlite, t ex pwrp__test57.dbsqlite.

Underh ll och fels kning

Databasen kan unders kas med 'sqlite3'.

```

> sqlite3 $pwrp_db/pwrp__test57.dbsqlite
sqlite> select oname,tablename from items;
VolTest57:H1-Av1|0000_001_001_002_0000004a__0
VolTest57:H1-Av2|0000_001_001_002_0000004c__2
VolTest57:H1-Av3|0000_001_001_002_0000004e__3
VolTest57:H1-Av4|0000_001_001_002_00000050__4
VolTest57:H1-Dv1|0000_001_001_002_00000052__5
VolTest57:H1-Iv1|0000_001_001_002_00000054__6
VolTest57:H1-Iv2|0000_001_001_002_00000056__7

sqlite> select time,value from 0000_001_001_002_0000004a__0 order by id desc limit 20
2021-07-09 16:25:10|-26.695
2021-07-09 16:25:09|-20.593
2021-07-09 16:25:08|-14.409
2021-07-09 16:25:07|-8.16822
2021-07-09 16:25:06|-1.89594
2021-07-09 16:25:05|4.38459
2021-07-09 16:25:04|10.6471
2021-07-09 16:25:03|16.8675
2021-07-09 16:25:02|23.0222
2021-07-09 16:25:01|29.0861
2021-07-09 16:25:00|35.0345
2021-07-09 16:24:59|40.8447
2021-07-09 16:24:58|46.4945
2021-07-09 16:24:57|51.9609
2021-07-09 16:24:56|57.2216
2021-07-09 16:24:55|62.2567
2021-07-09 16:24:54|67.0468
2021-07-09 16:24:53|71.5718

```

```
2021-07-09 16:24:52|75.815
2021-07-09 16:24:51|79.7587
sqlite> .quit
```

6.3 HDF5

HDF5 kr ver inte installation av n gon ytterligare serverprocess. D remot  r funktionen begr nsad. St rd f r server tr dar och d dband med linj r regression saknas.

HDF5 konfigureras genom att s tta Database i SevServer-objektet till HDF5.

Databas-filen l ggs p  \$pwrp_db med namnet pwrp__systemname'.hdf5, t ex pwrp__test57.hdf5. Filen inneh ller grupperna 'Dir' och 'Tables', d r 'Dir' inneh ller dataseten 'Cmn', 'Items', 'ObjectItems', 'ObjectItemAttributes' och 'Stat'. Under 'Tables' finns ett dataset f r varje attribut eller objekt som lagras, t ex 'O000_001_001_003_0000004a__0', 'O000_001_001_003_0000004c__1' etc.

Underh ll och fels kning

Det  r m jligt att inspektera data-filen med Python genom att installera python3-h5py.

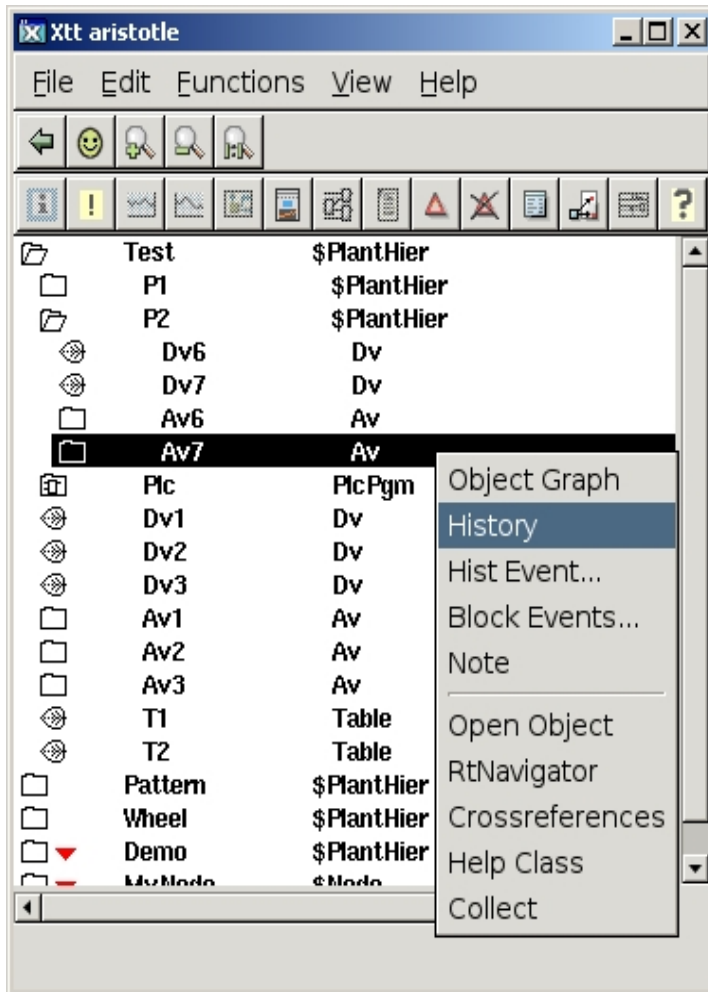
```
> python3
>>> import h5py
>>> f = h5py.File('/usr/pwrp/test57/src/db/pwrp__test57.h5', 'r')
>>> list(f.keys())
['Dir', 'Tables']
>>> list(f['Dir']['Items'])
[(0, b'O000_001_001_003_0000004a__0', 65795, 74, b'VolHdf5:H1-Av1', b'ActualValue',
0, 1625831988, 3600, 1., 76, 1., b'', 98306, 4, b'', 0),
(1, b'O000_001_001_003_0000004c__1', 65795, 76, b'VolHdf5:H1-Av2', b'ActualValue',
0, 1625831988, 3600, 1., 76, 1., b'', 98306, 4, b'', 0)
...
>>> list(f['Tables']['O000_001_001_003_0000004a__0']['Data'])
[(1625835966, 0, -93.87609), (1625835967, 0, -91.52528), (1625835968, 0, -88.812),
(1625835969, 0, -85.74749), (1625835970, 0, -82.343864), (1625835971, 0, -78.61555),
(1625835972, 0, -74.57546), (1625835973, 0, -70.24045), (1625835974, 0, -65.627655),
...]
```

7 Utvinna historisk data

Här visas några exempel på hur man kan utvinna och visa historisk data.

7.1 Xtt

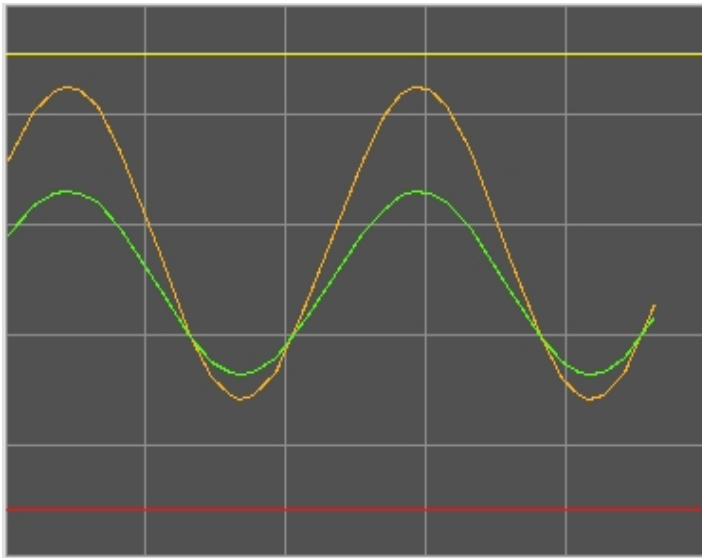
Genom att aktivera Historik-metoden i popupmenyn eller från verktygspanel eller meny i objektbilden får en signal, öppnas ett kurvfönster. Det kan även öppnas med Open Graph-metoden för ett SevHist objekt.



Kurvfönstret kan även öppnas med kommandot 'open history'.

7.2 Ge graph

För Analog/SevHist i Ge editorn, kan man lägga in ett diagram för historik. Diagrammet kan visa två kurvor.



7.3 Sev Analyser

Sev Analyser kan läsa in historik för ett antal items, och visa upp kurvor, scatterplots, skapa modeller med linjär regression och neurala nätverk mm.

Se kapitlet Multivariat analysator nedan.

7.4 Larm och händelseanalysator

Larm och händelseanalysatorn kan läsa in historik för larm och händelser, antal items och visa tabeller och statistik.

Se kapitlet Larm och händelseanalysatorn nedan.

7.5 Python

ProviewR Python runtime-modulen, pwrtr, innehåller funktioner för att hämta historisk data från en sev server.

pwrtr kan exekveras på en nod som har QCom-kontakt med historikservern.

Exempel

`pwrtr.getSevItemData()` hämtar historik för ett attribut. Den returnerar en tuple med tre element: antal mätpunkter, en tuple med värden och en med tider.

```
import pandas as pd
from datetime import datetime
import pwrtr
```

```
pwrtr.init("apl")
```

```
result = pwrtr.getSevItemData( 'localhost', '_00.254.254.204:68',
    'ActualValue', '00:02:00', 'now', 1000)
```

```

for i in range(result[0]):
    print(i, str(result[2][i])[:22], result[1][i])

```

Exempel

`pwrnt.getSevItemsDataFrame()` kan hämta historik för flera attribut, och returnerar historiken i en struktur som kan läggas in i en pandas frame. Den första kolumnen innehåller tiden och övriga kolumner de olika attributens värden.

```

import pandas as pd
import pwrnt

pwrnt.init("appl")

oidlist = []
attrlist = []
isobjectlist = []

# Append first attribute
oidlist.append('_00.254.254.204:68')
attrlist.append('ActualValue')
isobjectlist.append(0)

# Append second attribute
oidlist.append('_00.254.254.204:69')
attrlist.append('ActualValue')
isobjectlist.append(0)

result = pwrnt.getSevItemsDataFrame( 'localhost', oidlist, attrlist,
    isobjectlist, '00:02:00', 'now', 0.5, 1000)
columns = ('time', 'A1', 'A2')
data = pd.DataFrame(data=result)
data.columns = columns
print(data)

```

7.6 Mqtt server

Mqtt server i ProviewR kan svara på förfrågningar om historikdata.

Action "history" hämtar historik för ett attribut och med action "eventhist" hämtas larmhistorik.

Exempel

Nedan följer ett kod-exempel i Python med MQTT modulen `python3-paho-mqtt`. Historiken hämtas från den lokala MQTT servern, topic 'proviewr/server'. Svaret anmodas att skickas till topic 'repl/history', och förfrågan gäller att hämta historik för attributet H1-Av1.ActualValue för de senaste 15 minuterna.

```

#!/usr/bin/python3
#
import paho.mqtt.client as mqtt
import sys
import time

```

```

from datetime import datetime
import json
import matplotlib.pyplot as plt
from datetime import datetime

def on_message(client, userdata, message):
    data = json.loads(str(message.payload.decode("utf-8")))

    # Convert time strings to datetime objects
    t = []
    for dt in data['time']:
        t.append(datetime.strptime(dt+'0000', '%d-%b-%Y %H:%M:%S.%f'))

    # Plot the curve, use drawstyle='steps-pre' for digital signals
    plt.plot(t, data['values'], label='Diff')
    plt.show()

# Connect to MQTT server
client = mqtt.Client('Aristotle')
client.username_pw_set('pwrp', 'pwrp')
client.on_message = on_message
client.connect('localhost')

# Subscribe to reply
client.subscribe("repl/history", 1)

# Send history request
client.publish('proviewr/server', '{"action":"history",' \
'"reply":"repl/history","server":"localhost","object":"H1-Av1",' \
'"attribute":"ActualValue","from":"0:15:0","to":"now","maxrows":2000}')

for i in range (0, 3):
    print("Loop");
    client.loop_start()
    time.sleep(1)
    client.loop_stop()

```

8 Multivariat analysator

Med den multivariata analysatorn är det möjligt att visa och analysera historiska data och loggad data. Det är också möjligt att linarisera och transformera data och applicera verktyg för maskininläring som linjär regression och neurala nätverk, som kan användas i modeller och MPC-regulatorer.

8.1 Dataset

Ett dataset innehåller data ordnad i kolumner och rader. Den första kolumnen är mättidpunkten, och följande kolumner innehåller uppmätt data för processvariabler. Data kan hämtas från en sev server, genereras av loggnings-funktionen i Xtt eller läsas från en csv-fil.

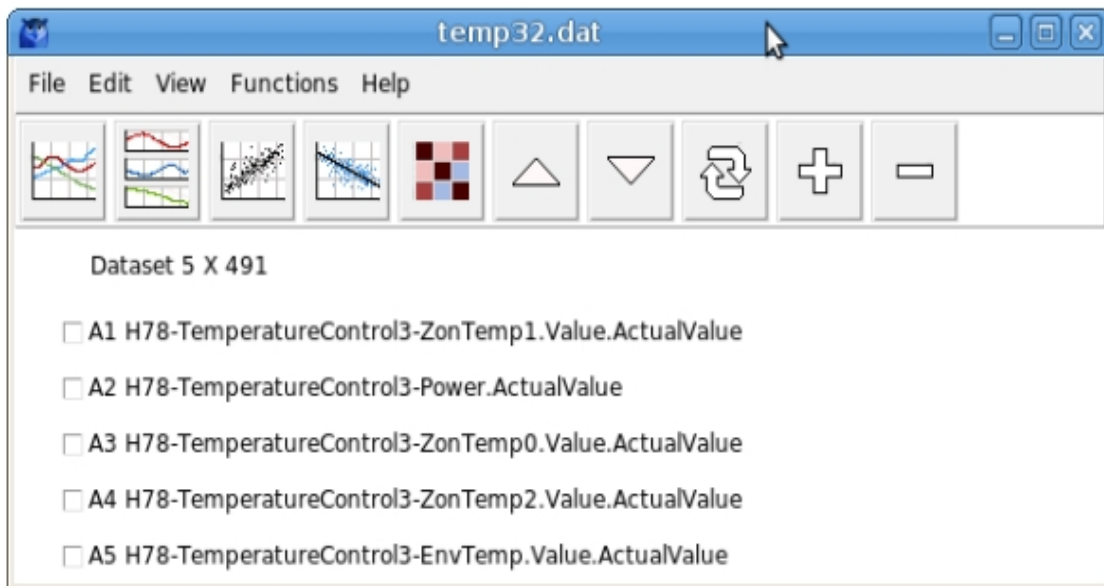


Fig Dataset

Sev server

Data hämtas från en sev server från 'File/Import from server' i menyn. Serverns namn och ett valfritt filter anges. De item som matchar filtret visas och de som ska bli en del av datasetet kan väljas ut. Slutligen matas start och slut-tid in och data hämtas och läggas in i datasetet.

From	<input type="text" value="00:05:00"/>
To	<input type="text" value="now"/>
Interval	<input type="text" value="1.0"/>
Max	<input type="text" value="500"/>

- VolOpg7:H1-Dv1.ActualValue
- VolOpg7:H1-Dv2.ActualValue
- VolOpg7:H1-Dv3.ActualValue
- VolOpg7:H1-Av1.ActualValue
- VolOpg7:H1-Av2.ActualValue
- VolOpg7:H1-Av3.ActualValue
- VolOpg7:H1-Motor1.Motor.TempSensor.Value.ActualValue
- VolOpg7:H1-Motor1.FrequencyConverter.ActSpeed.ActualValue
- VolOpg7:H1-Motor1.FrequencyConverter.RefSpeed.ActualValue
- VolOpg7:H1-Motor1.Contactor.Order.ActualValue

Fig HÅrmta data frÅr en sev server

Xtt loggning

Parametrar samlas in och IÅr ggs in i ett loggnings-entry, och fÅr att fÅr rÅr tidsformat, sÅr sÅr 'Format' till 1. NÅr loggningen År klar, kan analysatorn År ppnas frÅr 'Analyze'-knappen i loggnings-entryt.

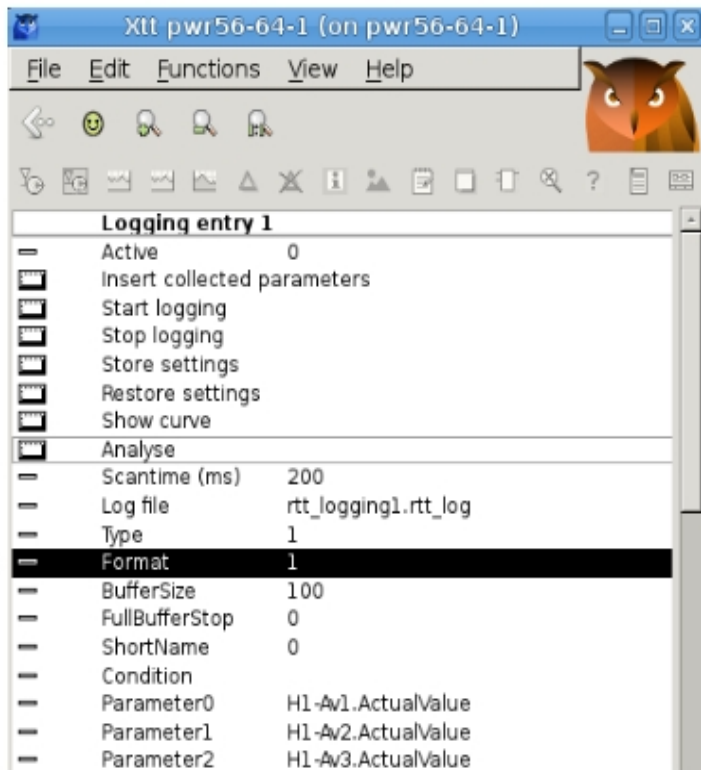


Fig Xtt logging

csv file

Data kan läsas från en csv-fil i formatet som visas nedan. Den första raden är en rubrik-rad med 'Time' och namnet på varje parameter. Följande rader innehåller tid och parametervärden vid denna tid. Filen öppnas från File/Open i menyn.

```
Time,H78-TemperatureControl3-ZonTemp1.Value.ActualValue, H78-TemperatureControl3-
Power.ActualValue, H78-TemperatureControl3-ZonTemp0.Value.ActualValue, H78-Tempera
tureControl3-ZonTemp2.Value.ActualValue, H78-TemperatureControl3-EnvTemp.Value.Ac
tualValue
2019-05-13 09:25:16.11, 130.500320, 0.000032, 190.000000, 180.000000, 21.500000
2019-05-13 09:25:16.62, 130.500320, 0.000032, 190.000000, 180.000000, 21.500000
2019-05-13 09:25:17.12, 130.500320, 0.000032, 190.000000, 180.000000, 21.500000
2019-05-13 09:25:17.62, 130.500320, 0.000032, 190.000000, 180.000000, 21.500000
2019-05-13 09:25:18.12, 130.500320, 0.000032, 190.000000, 180.000000, 21.500000
...
```

8.2 Diagram

Ett antal olika diagram kan visas, t ex sambandsdiagram (scatter plots) som visar sambandet mellan två kolumner, eller correlation heatmap som visar korrelationen mellan kolumner med färger. Mörkare färg är hög korrelation och ljusare färg negativ korrelation medan ljusa toner är låg korrelation.

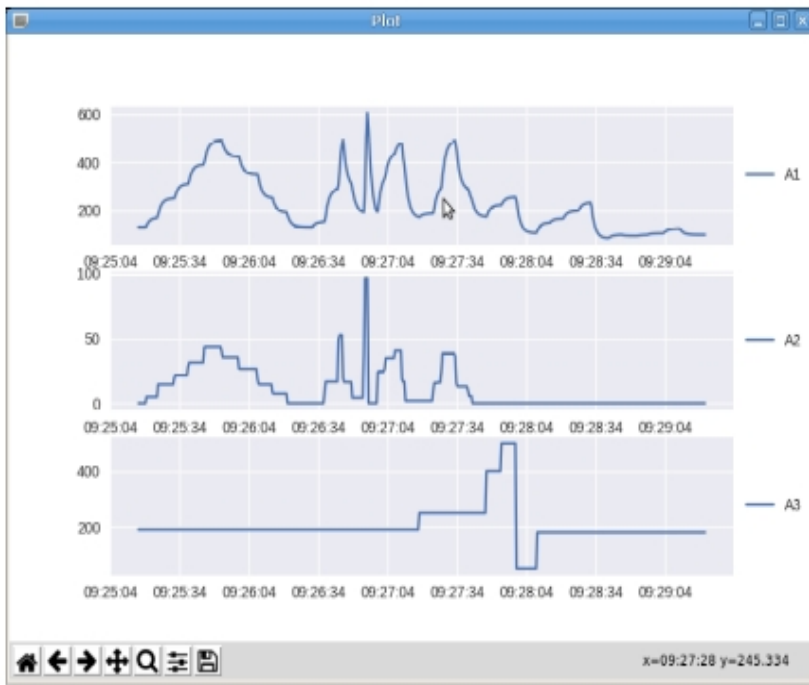


Fig Diagram

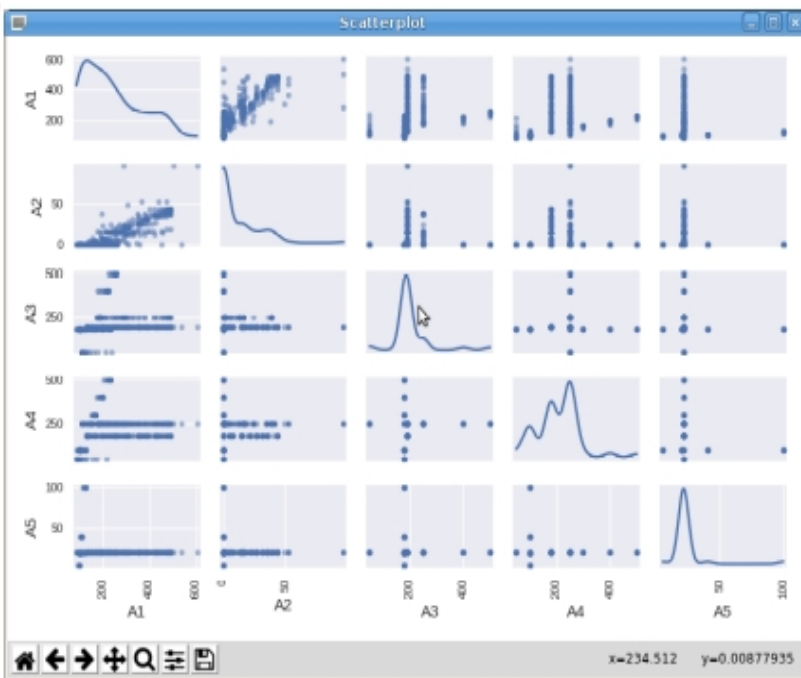


Fig Sambandsdiagram

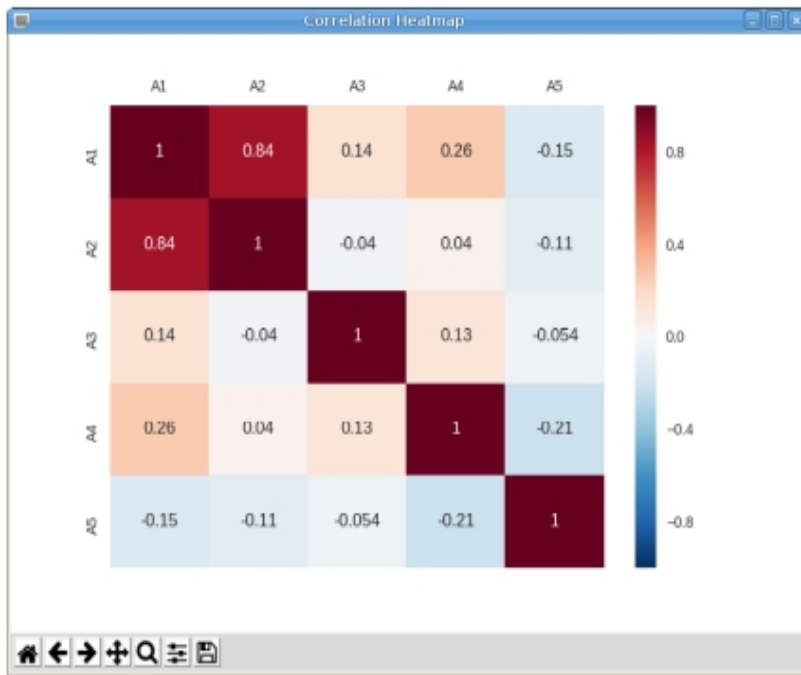


Fig Correlation heatmap

8.3 Editering av data

Datasetet kan editeras med ett antal funktioner

Split	Delat upp datasetet i två dataset.
Clip	Ta ut en del av datasetet.
Join	Sätt i hop två dataset.
Multiply	Skapa ett dataset där det nuvarande setet upprepas ett antal gånger.
Move up och down	Ändra ordning på kolumnerna.

8.4 Transformera data

När man skapar modeller med linjär regression krävs det att kolumnerna i datasetet har linjära beroenden. Ofta är inte det fallet. Nivån i en cylindrisk tank har till exempel inte ett linjärt förhållande till in och utflödet, utan till integralen av in och utflödet. Efter en integration av flödes-kolumnerna kommer det att vara ett linjärt samband och linjär regression kan tillämpas.

Konvertera kolumner

Det finns ett antal funktioner för att transformera data i en kolumn

Norm	Normalisera data.
Square	Beräkna kvadraten för varje rad.
Squareroot	Beräkna kvadratroten för varje rad.
Exp	Exponentialfunktion.
Log	Logaritmisk funktion.
Integral	Tidsintegral.
Curve	Linjär interpolation från en tabell specificerad i en csv-fil med datapunkter,

0,0
30,10
70,90
100,100

Shift

Värdena i kolumnen kommer att skiftas framåt eller bakåt. Antal positioner som värdena ska skiftas anges. Positivt värde skiftar framåt och negativt bakåt.

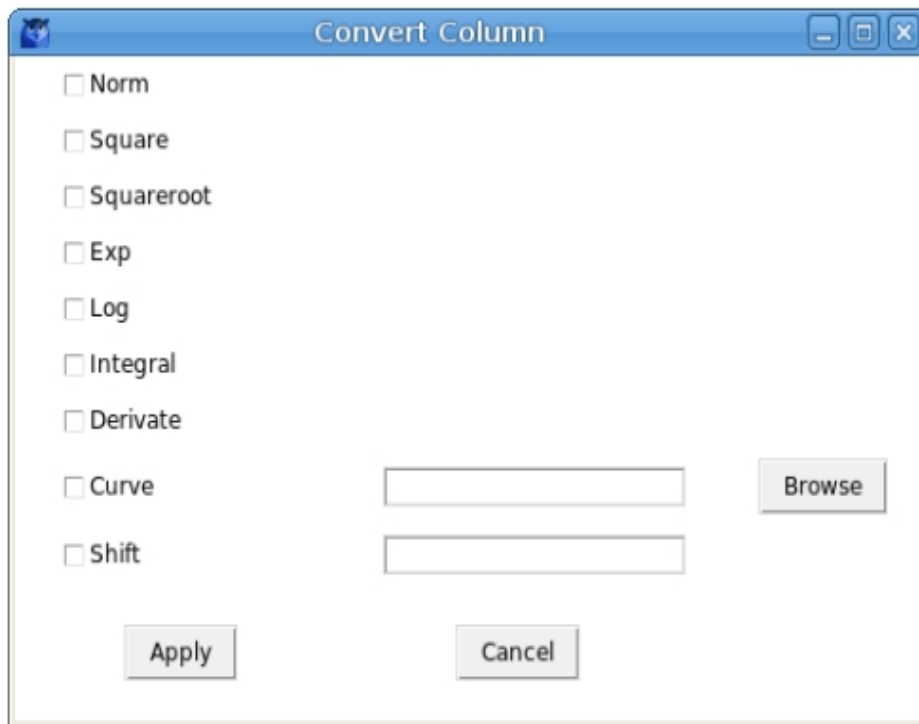


Fig Konvertera kolumn alternativ

Addera kolumn

Addera kolumn kommer i de flesta fall att transformera data i en eller två kolumner och lägga det transformerade datat i en ny kolumn.

Copy	Skapa en kopia av utvald kolumn.
Norm	Normalisera värdena i kolumnen.
Square	Beräkna kvadraten för varje rad.
Squareroot	Beräkna kvadratroten för varje rad.
Exp	Exponentialfunktion.
Log	Logaritmisk funktion.
Integral	Tidsintegral.
Derivate	Tidsderivata.
Add	Addera de två utvalda kolumnerna.
Sub	Subtraktion mellan två utvalda kolumner. Ordningen på kolumnerna i datasetet är av betydelse. Kolumnen med lägre position kommer att subtraheras från den med högre position.
Multiply	Multiplera de två utvalda kolumnerna.
Divide	Division mellan de två utvalda kolumnerna. Kolumnen ned högre position kommer att delas med den med lägre position.
Curve	Linjär interpolation från en tabell specificerad i en csv-fil.
Constant	Skapar en kolumn där varje rad har samma värde.
Shift	Värdena i en kolumn skiftas framåt eller bakåt.



Fig Alternativen för att lägga till kolumn

Formel

Transformationen av ett dataset kan innehålla flera steg. När transformationen är slutförd kan sekvensen lagras som en formel, och senare appliceras på andra matserier på samma parametrar. Formeln sparas med 'File/Save Formula' i menyn, och appliceras med 'File/Apply Formula'.

8.5 Linjär regression

Linjär regression skapar en modell där en parameter, y , kan beräknas från ett antal ingångsvärden $x_1 - x_n$. y antas ha ett linjärt beroende av ingångsvärdena, och formeln är

$$y = a_0 + a_1 * x_1 + a_2 * x_2 + \dots + a_n * x_n$$

där a_0, a_1, \dots, a_n kommer att beräknas vid den linjära regressionen.

Om beroendena inte är linjära måste de först linjäriseras med transformations-verktygen beskrivna ovan. När modellen används i runtime, måste processvärdena genomgå samma transformation innan de används i regressions-modellen.

Lasso och Ridge regression är varianter av linjär regression som också är implementerade.

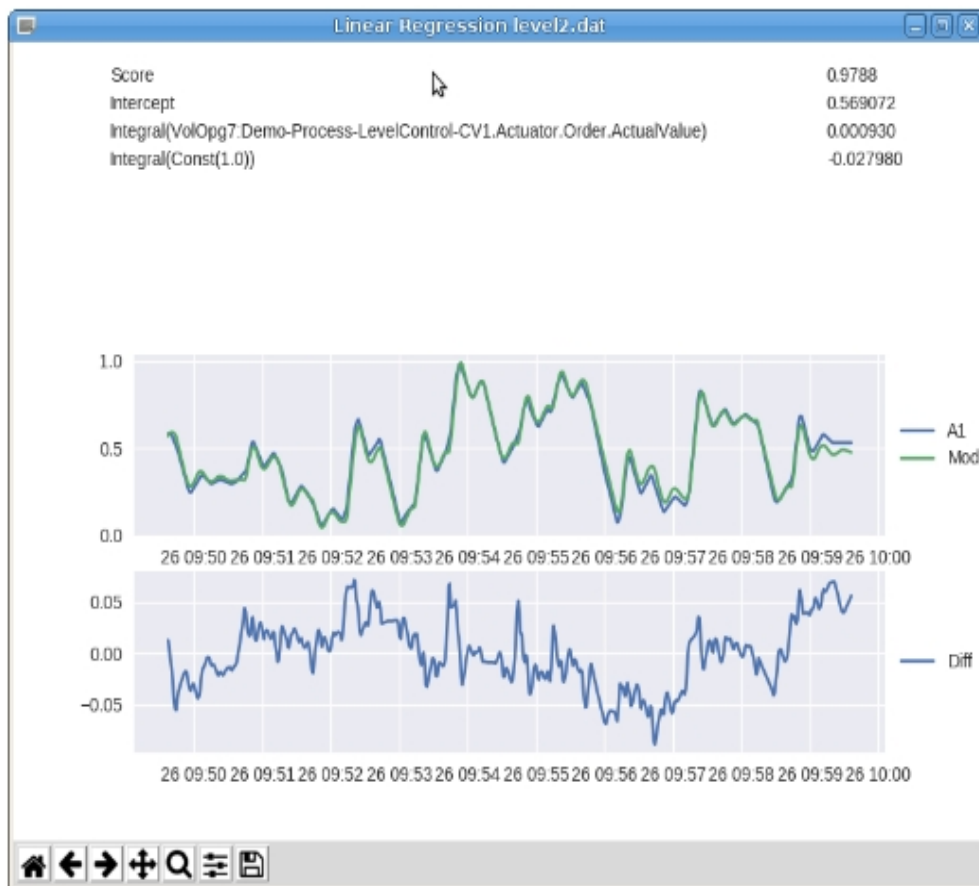


Fig Linear regression

8.6 MLP regressor

MLP (Multi Level Perceptron) är ett neuralt nätverk med ett ingångslager, ett antal dolda lager och ett utgångslager. Varje nod i de dolda och utgångs-lagren är neuroner som använder en icke linjär aktiveringsfunktion. MLP använder en upplärningsteknik som kallas backpropagation.

Innan träningen kan starta, måste ett antal inställningar göras, som antal dolda lager och lagerstorlekar, aktiveringsfunktion etc.

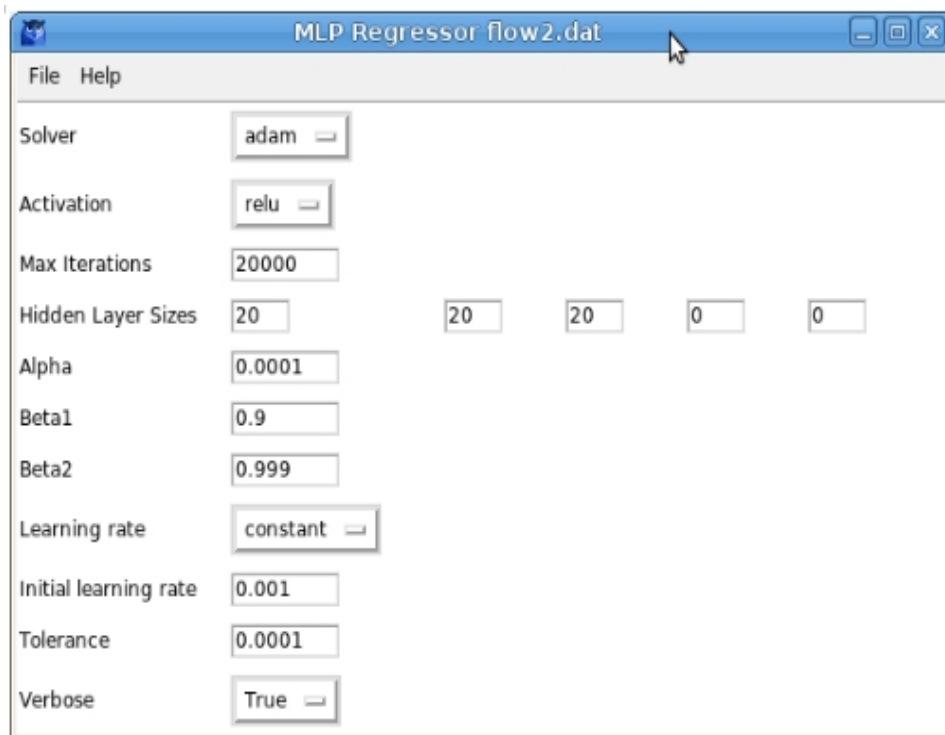


Fig Inställningar för MLP regressor

Från 'File/Create Model' i menyn startas upplärningen och när den är avslutad visas i 'Score' hur väl modellen stämmer överens med uppmätta data. En kurva med beräknade värden plottas också tillsammans med uppmätta värden. Modellen kan sparas på fil från 'File/Export Model' i menyn och sedan användas av MPC regulatoren eller modellobjekt.

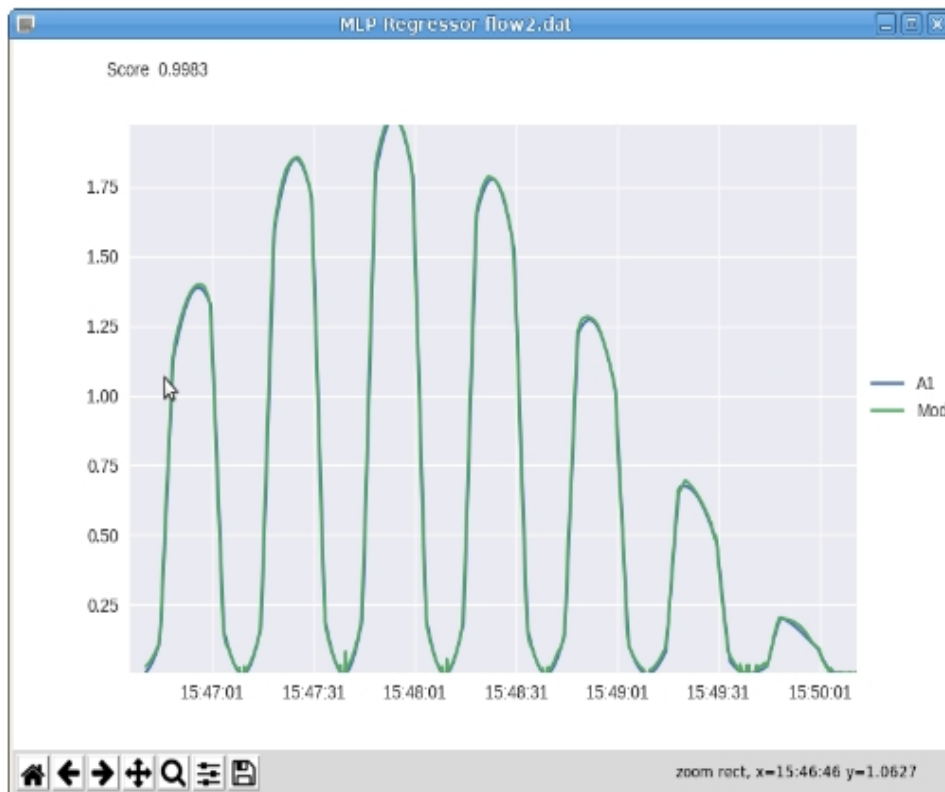
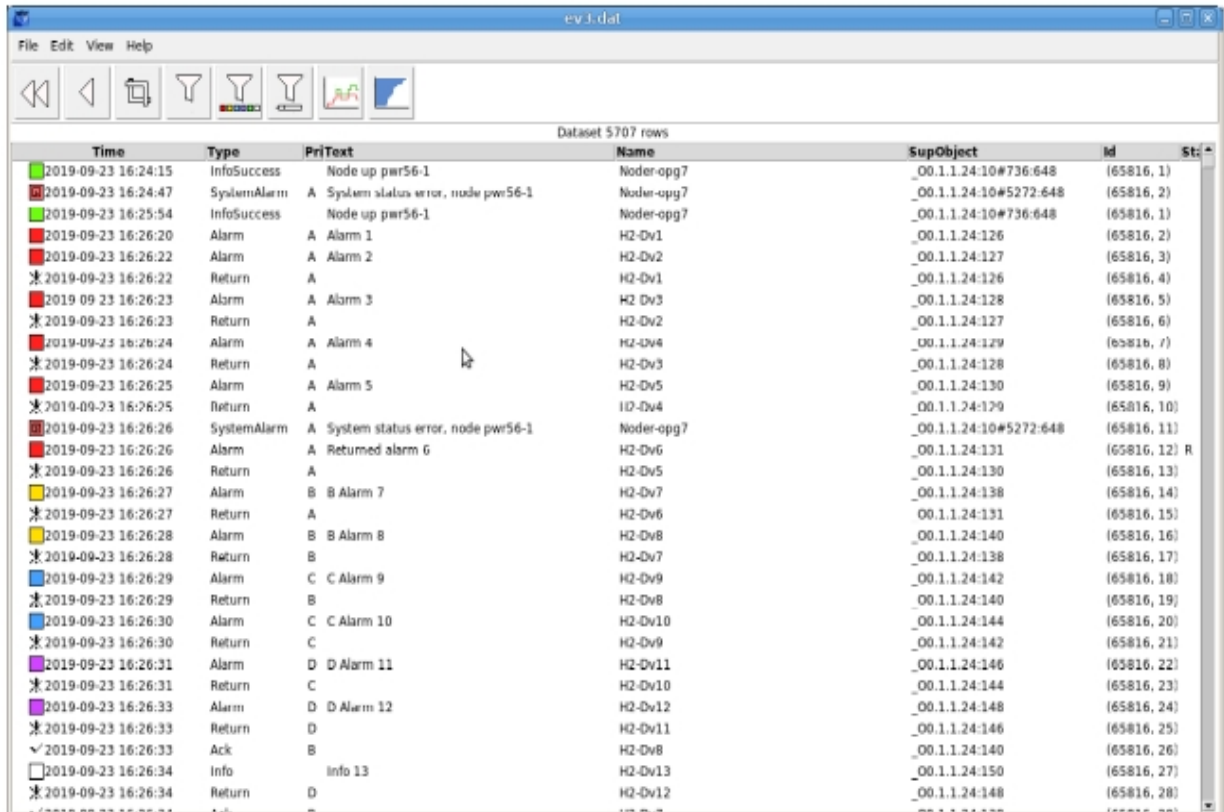


Fig Träningresultat

9 Larm och händelse analysator

Larm och händelse analysatorn kan hämta larm från en sev servern, händelseloggen eller händelselistan, och visa statistik och diagram över larmsituationen. Ett antal filtreringsfunktioner kan användas för att välja ut larm av en speciell typ eller prioritet, eller visa händelser för ett specifikt övervakningsobjekt.



The screenshot shows a software window titled 'ev3.dat' with a menu bar (File, Edit, View, Help) and a toolbar with navigation and filter icons. Below the toolbar, it says 'Dataset 5707 rows'. The main area contains a table with the following columns: Time, Type, PriText, Name, SupObject, Id, and St. The table lists various alarm events with their corresponding times, types (e.g., Alarm, SystemAlarm, InfoSuccess), priority text, names (e.g., Noder-opg7, H2-Dv1), and IDs.

Time	Type	PriText	Name	SupObject	Id	St
2019-09-23 16:24:15	InfoSuccess	Node up pwr56-1	Noder-opg7	_00.1.1.24:10#736:648	(65816, 1)	
2019-09-23 16:24:47	SystemAlarm	A System status error, node pwr56-1	Noder-opg7	_00.1.1.24:10#5272:648	(65816, 2)	
2019-09-23 16:25:54	InfoSuccess	Node up pwr56-1	Noder-opg7	_00.1.1.24:10#736:648	(65816, 1)	
2019-09-23 16:26:20	Alarm	A Alarm 1	H2-Dv1	_00.1.1.24:126	(65816, 2)	
2019-09-23 16:26:22	Alarm	A Alarm 2	H2-Dv2	_00.1.1.24:127	(65816, 3)	
2019-09-23 16:26:22	Return	A	H2-Dv1	_00.1.1.24:126	(65816, 4)	
2019-09-23 16:26:23	Alarm	A Alarm 3	H2-Dv3	_00.1.1.24:128	(65816, 5)	
2019-09-23 16:26:23	Return	A	H2-Dv2	_00.1.1.24:127	(65816, 6)	
2019-09-23 16:26:24	Alarm	A Alarm 4	H2-Dv4	_00.1.1.24:129	(65816, 7)	
2019-09-23 16:26:24	Return	A	H2-Dv3	_00.1.1.24:128	(65816, 8)	
2019-09-23 16:26:25	Alarm	A Alarm 5	H2-Dv5	_00.1.1.24:130	(65816, 9)	
2019-09-23 16:26:25	Return	A	H2-Dv4	_00.1.1.24:129	(65816, 10)	
2019-09-23 16:26:26	SystemAlarm	A System status error, node pwr56-1	Noder-opg7	_00.1.1.24:10#5272:648	(65816, 11)	
2019-09-23 16:26:26	Alarm	A Returned alarm 6	H2-Dv6	_00.1.1.24:131	(65816, 12)	R
2019-09-23 16:26:26	Return	A	H2-Dv5	_00.1.1.24:130	(65816, 13)	
2019-09-23 16:26:27	Alarm	B B Alarm 7	H2-Dv7	_00.1.1.24:138	(65816, 14)	
2019-09-23 16:26:27	Return	A	H2-Dv6	_00.1.1.24:131	(65816, 15)	
2019-09-23 16:26:28	Alarm	B B Alarm 8	H2-Dv8	_00.1.1.24:140	(65816, 16)	
2019-09-23 16:26:28	Return	B	H2-Dv7	_00.1.1.24:138	(65816, 17)	
2019-09-23 16:26:29	Alarm	C C Alarm 9	H2-Dv9	_00.1.1.24:142	(65816, 18)	
2019-09-23 16:26:29	Return	B	H2-Dv8	_00.1.1.24:140	(65816, 19)	
2019-09-23 16:26:30	Alarm	C C Alarm 10	H2-Dv10	_00.1.1.24:144	(65816, 20)	
2019-09-23 16:26:30	Return	C	H2-Dv9	_00.1.1.24:142	(65816, 21)	
2019-09-23 16:26:31	Alarm	D D Alarm 11	H2-Dv11	_00.1.1.24:146	(65816, 22)	
2019-09-23 16:26:31	Return	C	H2-Dv10	_00.1.1.24:144	(65816, 23)	
2019-09-23 16:26:33	Alarm	D D Alarm 12	H2-Dv12	_00.1.1.24:148	(65816, 24)	
2019-09-23 16:26:33	Return	D	H2-Dv11	_00.1.1.24:146	(65816, 25)	
2019-09-23 16:26:33	Ack	B	H2-Dv8	_00.1.1.24:140	(65816, 26)	
2019-09-23 16:26:34	Info	Info 13	H2-Dv13	_00.1.1.24:150	(65816, 27)	
2019-09-23 16:26:34	Return	D	H2-Dv12	_00.1.1.24:148	(65816, 28)	

Fig Larm och händelse analysator

Exempel på diagram är 'Event frequency histogram' som visar de mest frekventa larmen, och 'Not returned alarms' som visar antalet samtidiga larm som funktion av tiden.



Fig Event frequency histogram

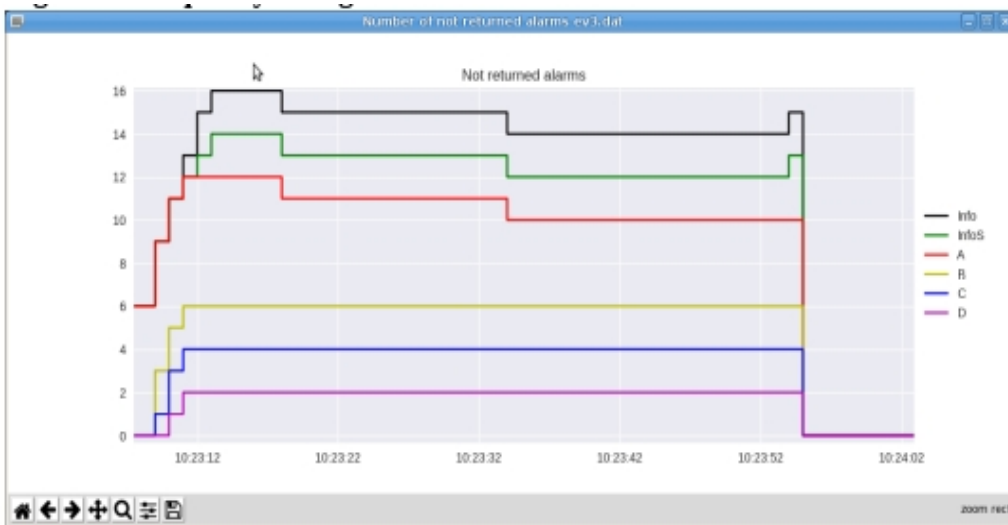


Fig Not returned alarms

10 Konfigurering av en Storage station

En storage station genereras som en process eller operatörsstation genom att installera pwrvt-paketet. Den konfigureras i directoryvolymen med ett NodeConfig objekt och en rotvolym på samma sätt som en process eller operatörsstation.

Kommunikation med processtationer

Tanken är att en storage station betjänar ett avsnitt med flera processtationer som tillhör olika projekt med olika versioner. Därför har normalt inte en storage station någon nathanterarkontakt, utan enbart med QCom-kontakt.

Det konfigureras genom att man lägger in FriendNodeConfig objekt för alla noder som storage stationen ska ha kontakt med, och sätter QComOnly i Connection. Om storage stationen ligger i ett projekt med processtationer, sätter man QComAutoConnectDisable i BusConfig-objektet för att sedan konfigurera även länkarna inom projektet med FriendNodeConfig objekt.

Uppgradering

Att notera vid uppgradering av storage stationer från följande V5.8.

- pwrsev-paketet har utgitt sedan V5.8.0 och pwrvt-paketet ska användas istället.
- Noden ska konfigureras med ett NodeConfig objekt istället för SevNodeConfig.
- Rootvolymen ska konfigureras med ett RootVolumeConfig objekt, och editeras.