

ProviewR

OPEN SOURCE PROCESS CONTROL



Handbok för Operatörer

Copyright (C) 2005-2025 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

ProviewR Handbok för Operatörer är främst avsedd för personer som kommer i kontakt med gående proviewsystem i sitt dagliga arbete. Det kan vara

- operatörer som övervakar och sköter en process.
- underhållspersonal som felsöker anläggningen.
- processutvecklare som hämtar information om processen.
- systemansvariga som underhåller och felsöker systemet.

Funktioner för alla dessa grupper finns tillgängliga i den ordinare operatörmiljön, och vem som tillhör in viss grupp bestäms av de privilegier som en viss användare är tilldelad.

3 Allmänt om ProviewR

Vad är ProviewR

ProviewR är ett modernt, kraftfullt och generellt processtysystem. Det innehåller alla de funktioner som normalt behövs för att styra och övervaka en process: sekvensstyrning, reglering, datainsamling, kommunikation, larmhantering, HMI/SCADA, historisk datalagring mm.

ProviewR är ett distribuerat system, dvs det består av ett antal datorer (noder) förbundna i ett nätverk. Noderna kan vara av typen processtation, operatörsstation eller lagringsstation.

3.1 Stationer

Processtation

En processtation samlar in mätdata från olika givare i processen, det kan vara analoga mätdata som temperaturen, flöden, nivåer eller digital mätdata från t ex fotoceller eller tryckvakter. I processtationen exekveras ett styrprogram, som utifrån mätdata beräknar styrdata för processen och skickas ut till motorer och ställdon som i sin tur påverkar processen.

Processtationer innehåller ofta en speciell hårdvara för att läsa in måtvärden och ställa ut styrdata, men detta kan även ske över nätverket mha protokoll som Modbus/TCP eller Profinet.

Operatörstation

Operatörsstationens uppgift är att svara för gränssnittet mellan operatör och process. Operatören övervakar processen genom processbilder, som visar läget i processen genom att data presenteras i form av staplar, kurvor, indikatorer mm. Operatören kan påverka processen t ex genom att mata in data, eller trycka på knappar i processbilden.

Lagringsstation

Vissa mätdata måste lagras för att man ska kunna se förändringar och trender, eller för att man ska kunna gå tillbaka och analysera läget i processen vid en viss tidpunkt. Lagringstiden kan variera från någon timme till flera år. Lagringen sker på lagringsstationer, som har tillräckligt med diskutrymme för att kunna lagra data under lång tid, och som även har backupfunktioner så att data inte förloras vid t ex ett diskhaveri.

Man brukar även tala om nivå 2 system, dvs system av överordnad karaktär som innehåller t ex materialplaneringsfunktioner, tunga beräkningar och liknande. Dessa ingår i kategorin processtationer.

Utvecklingsstation

På nätverket kan även finnas utvecklingstationer, på vilka process- operatörs- och lagringstationerna konfigureras och programmeras. På utvecklingsstationerna finns det verktyg för att rita processbilder, programmera sekvenser, logikschema och reglerkretsar, mm. Efter en ändring i konfigurationen för en station, laddas det nya konfigurationen ner till stationen via nätverket.

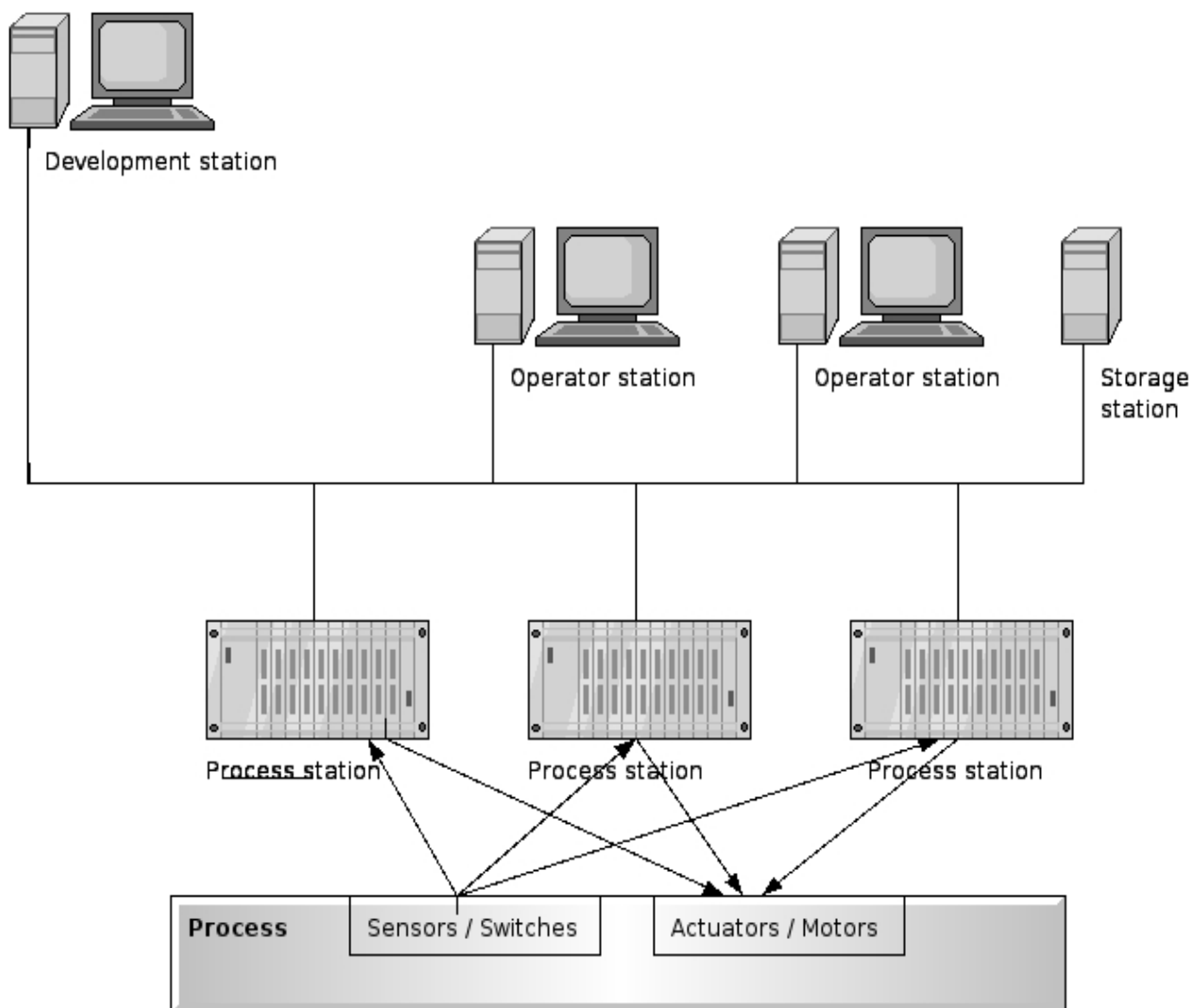


Fig Komponenter i ett ProviewR system

3.2 Objekt

Objektträd

Ett centralt begrepp i ProviewR är objekt. Givare, ventiler, motorer, regulatorer etc, representeras av objekt i en databas. Objektet är ordnade i en trädsstruktur där olika hierarkinivåer i trädet motsvarar delar i anläggningen och processen. Om vi betraktar en ugn kan den översta nivån representera ugnen. Ugnen består av 4 värningszoner som var och en representeras av objekt på nästa nivå i trädet. Varje zon består av två värmeelement och två temperaturgivare som finns ytterligare en nivå ner. Här finns även några objekt som inte

har någon fysisk motsvarighet i anläggningen, en temperatur regulator och ett styrprogram för zonen.

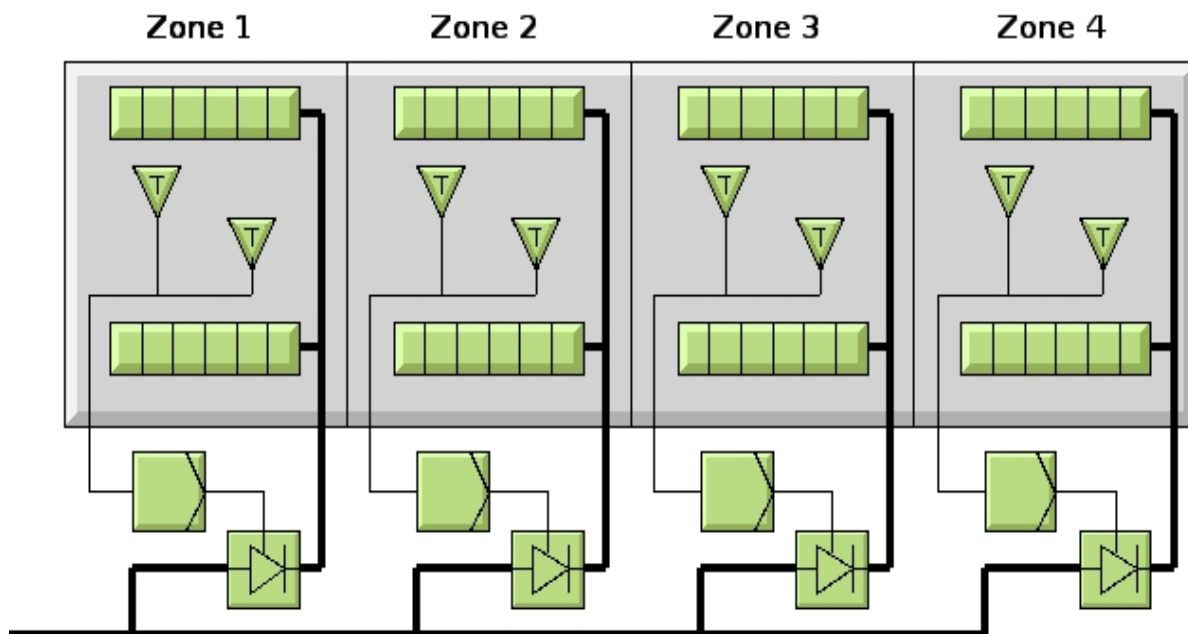


Fig Ugn med 4 zoner

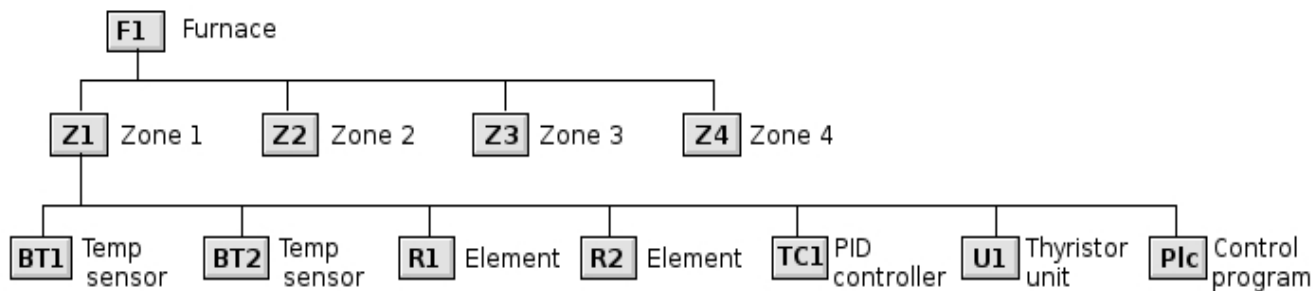


Fig Objekt träd för ugnen

Notera att många objekt har ett ganska kryptiskt namn som följer en beteckningsstandard, men till varje objekt finns en utförligare beskrivning som ofta visas tillsammans med objektet.

Låt oss titta lite närmare på ett objekt, t ex ett av temperaturgivarobjekten. Objektet visas i en översiktssbild med objektets grafiska symbol, en trekant med bokstaven T.

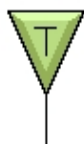


Fig Grafisk symbol för temperaturgivare

Objektsbild

Om man klickar på symbolen öppnas objektets objektsbild. Från objektsbilden kan man se objekts egenskaper. Det mest intressanta för en temperaturgivare är naturligtvis den

uppmätta temperaturen, som visas både i siffror och i form av en stapel (termometer), Se 'Fig Objektsbild för temperaturgivare' nedan. Temperaturgivar-objektet innehåller även fyra larmgränser, höghög, hög, låg och låglåg. Nivån på larmgränserna kan sättas från objektsbilden, och man kan även lägga in hysteres på dem. Med hjälp av checkboxar kan man också avaktivera larmgränser.

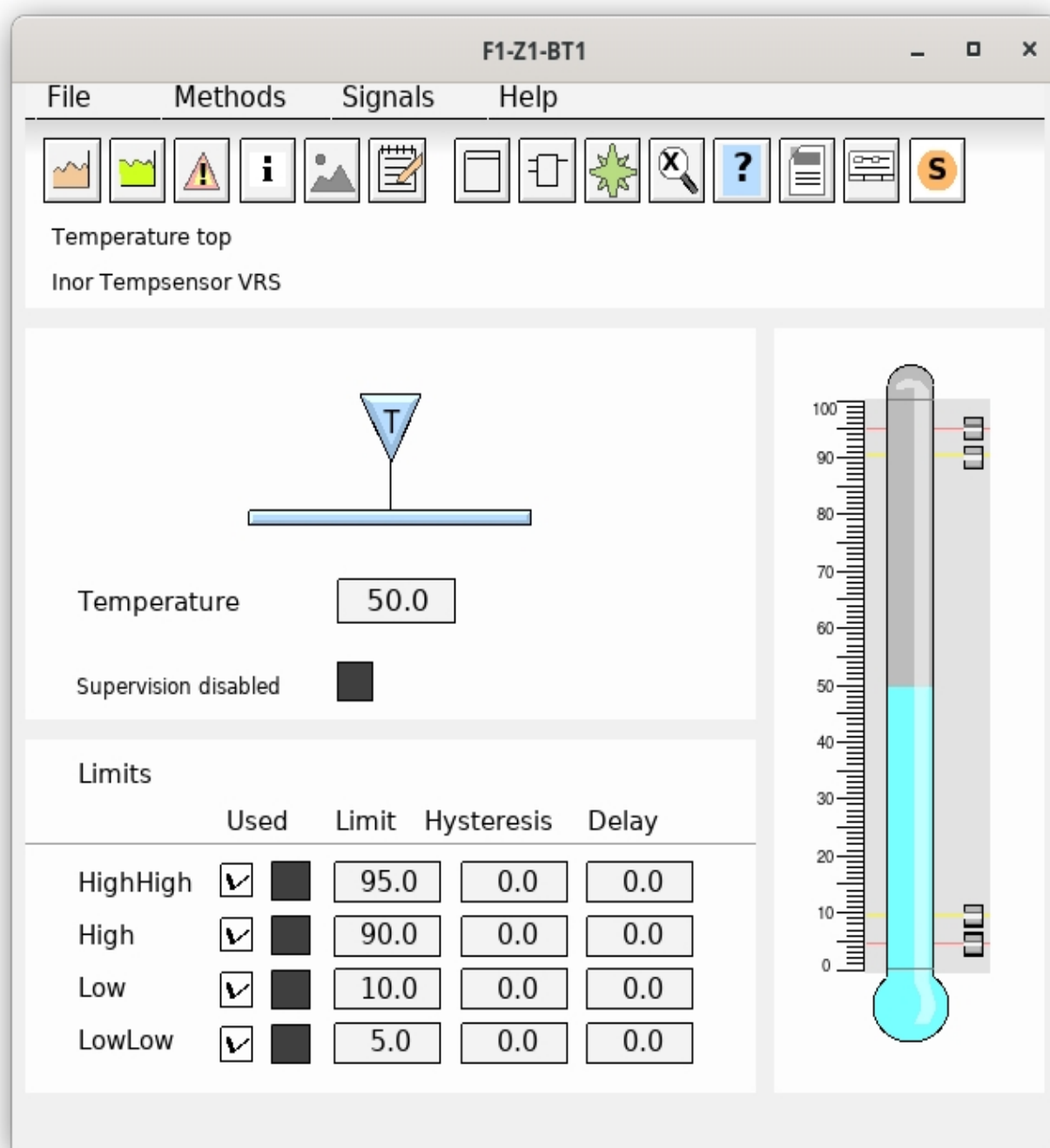


Fig Objektsbild för temperaturgivare

Metoder

I överkanten på objektsbilden finns ett antal tryckknappar som aktiverar objektets metoder. Mha av metoderna kan man hämta upp all information som finns om objektet i styrsystemet. Metoderna kan även aktiveras från objektsbildens meny, eller genom att högerklicka på den grafiska symbolen i översikt bilden, då en popupmeny med metoderna visas. Faktum är att var man än stöter på objektet, i larmlistor, i bilder, i plc koden etc, kan man genom att

högerklicka på objektet öppna popupmenyn med objektets metoder. Här följer en lista på olika metoder, men vilka som är aktuella för ett specifikt objekt beror på vilken typ av objekt det är, och hur det är konfigurerat.

Metod	Beskrivning
Objekt Bild	Öppna Objektsbilden.
Trend	Visa en trendkurva.
Historik	Hämta historiska data från en lagringsstation och visa i en kurva.
Historisk lista	Visa historisk lista med larm och händelser för objektet.
Blockera händelser	Blockera larm och händelser.
Hjälp	Visa en hjälptext för objektet.
Notera	Skriv in en anteckning för objektet. Anteckningen visas i objektsbilden.
Öppna objekt	Visa objektets datainnehåll.
RtNavigatör	Visa objektet i objektstrådet.
Korsreferenser	Visa var objektet förekommer i plc-kod och bilder.
Hjälp klass	Visa hjälp för den här objekttypen.
Datablad	Visa databladet för den komponent i anläggningen som objektet motsvarar.
Kopplingsschema	Visa elektriskt kopplingschema för objektet.

4 Operatörsplatsen

4.1 Starta operatörmiljön

Vanligtvis är inloggingen av arbetsstationen konfigurerad så att operatörmiljön startar automatiskt. Vid starten anges ett operatörsplatsobjekt där operatörsplatsens egenskaper finns konfigurerade.

Ibland vill man starta på annat sätt, t ex från ett terminalfönster.

Operatörmiljön startas med programmet `rt_xtt`, och man anger ett OpPlace objekt som argument. Kommandot för att starta med OpPlace objektet `Nodes-OpgNode-b55` är

```
> rt_xtt Nodes-OpgNode-b55
```

Det finns även ett antal optioner som man kan skicka med som argument

- l Språk. Ett av följande språk anges:
en_us engelska (default).
sv_se svenska.
de_de tyska.
fr_fr franska.
- q Fönstret med licenstexten visas ej vid uppstart.
- c Knapp för att stänga operatörmiljön visas i operatörsfönstret.
- u Startar med ett opplace objekt med samma namn som den linuxanvändare man är inloggad som.
- s Visar en lista på tillgängliga operatörsplatsobjekt.

4.2 Operatörsfönstret

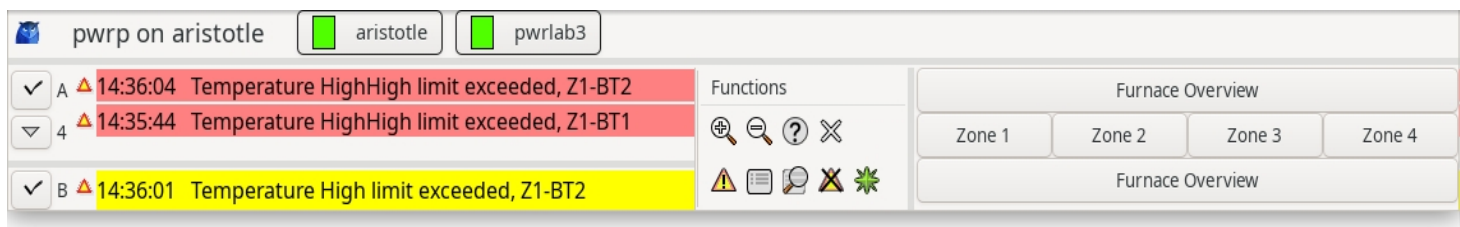


Fig Operatörsfönster

Operatörsfönstret lägger sig längst upp på bildskärmen och saknar titelrad och knappar för att ikonisera och ta bort fönstret. Det är uppdelat i tre delar, till vänster visas aktuella larm och meddelanden, i mitten finns en meny och knappar för olika basfunktioner, och till vänster knappar som är konfigurerade för just den här operatörsplatsen.

Statusfält

Den översta raden är ett statusfält. Här visas vem som är inloggad och på vilken nod. För den aktuella noden, och för alla noder som det finns nod-övervakning på, visas en indikator för nodens systemstatus. Grön innebär att allt fungerar som det ska, gult indikerar varning, rött fel och blinkande rött fatalt fel. Om man klickar på en nod öppnas status-bilden för noden.

Larm och meddelanden

I vänstra delen av operatörsfönstret visas larm och meddelanden.

Larmen är uppdelade i fyra prioritetsnivåer A, B, C och D, där A har högst prioritet och D lägst. A-larm är röd-makerade, B-larm gul, C-larm blå och D-larm violett. Dessutom finns kategorin Info meddelande som markeras med grönt.

I det övre fältet visas de senaste okvitterade A-larmen. Beroende på operatörsfönstrets storlek i höjdlängd, visas 2 till 5 larm. Här finns även en knapp markerad med en bock för att kvittera A-larm.

I den undre fältet visas de senaste okvitterade B, C och D-larmen, samt info meddelanden. Dessa larm visas i prioritetsordning snarare än tidsordning, dvs om det finns okvitterade B-larm visas det senaste av dessa. Endast om det inte finns några okvitterade B-larm visas eventuella C-larm etc. Även här finns en kvittensknapp, som man kvitterar B, C, D-larmen och info meddelandena med.

Ett larm är rådande om larmvillkoret fortfarande är uppfyllt. Detta markeras med en varningstriangel före larmtexten.

Meny och knappar för att öppna basfunktioner

I mittendelen av operatörfönstret finns en meny, och knappar för att öppna basfunktioner.

Meny

Här följer en beskrivning på de olika menyalternativen i Functions menyn.

Functions/Alarm/AlarmList	Öppnar larmlistan med rådande eller okvitterade larm.
Functions/Alarm/EventList	Öppnar händelselistan, en lista på de senaste larmen och händelser.
Functions/Alarm/EventLog	Öppnar händelseloggen, där man kan visa lagrade händelser.
Functions/Alarm/BlockList	Öppnar listan över blockerade larm.
Functions/Curves/Trends	Visar en lista på alla trendkurvor.
Functions/Curves/Fast	Visar en lista på alla snabbkurvor.
Functions/Curves/Process History	Visar en lista på process historik kurvor.
Functions/Process Graphics	Visar en lista på alla processbilder som är definierade med ett XtGraph objekt.
Functions/Navigator	Öppnar navigatorn.
Functions/View/Zoom in	Ökar textstorleken i larmtexterna.
Functions/View/Zoom out	Minskar textstorleken i larmtexterna.
Functions/User/Switch User	Öppnar inloggningsfönster för inloggning som annan användare.
Functions/User/Show User	Visar den användare som är inloggad.
Functions/User/Return	Återgår till den ursprungliga användaren.
Functions/Help/Project	Visar hjälptexter för projektet.
Functions/Help/Overview	Visar en översikt över hjälptexter.
Functions/Help/Operator Window	Visar hjälp för operatörsfönstret.
Functions/Help/About ProviewR	Visar info om ProviewR, t ex version av installerat paket.
Functions/Close	Stänger ner operatörsplatsen.

Tryckknappar

Under menyn finns en verktygspanel med ett antal tryckknappar som fungerar som genvägar till menyalternativ:

- zooma in och ut.
- visa hjälptexten för projektet.
- avslutas.
- visa larmlista.
- visa händelselista.
- visa händelselog.
- visa lista över blockerade larm.
- öppna navigatorn.

Funktionsknappar

Till höger finns ett antal knappar som kan användas för att öppna bilder eller exekvera olika order eller kommandon. Hur knapparna används beror på operatörsplatsens konfigurering.

5 Larm och händelser

5.1 Allmänt om larm och händelser

Larm

Larm skickas ut till operatören när någonting inträffar som operatören måste uppmärksamma. Det kan t ex vara en temperatur som går över en gräns, eller att man har matat in ett felaktigt värde i en bild. I ProviewR finns speciella övervakningsobjekt som övervakar signaler och genererar larm.

Larm visas i larmlistan.

Prioritet

Larmen är grupperade i fyra prioritetsnivåer, A, B, C och D, där A har högst prioritet och D lägst. Hur man använder prioriteterna beror på hur system är konfigurerat. Det är vanligt att larmprioriteterna används på följande sätt.

A-larm markeras med rött. De har högst prioritet och markerar att ett allvarlig fel i anläggningen har inträffat som omgående bör åtgärdas. Ofta innebär det att driften av anläggningen stoppas och inte kan startas så länge larmen är rådande.

B-larm markeras med gult. De har lite lägre prioritet och innebär ett fel som snarast bör åtgärdas, men att driften kan fortsätta ytterligare en tid.

C-larm markeras med blått och D-larm med lila. De markerar mindre fel som inte är akuta. I många system är det tillräckligt med två larmnivåer, och man använder då enbart A och B-larm.

Larmtext

Larm innehåller en text, som visas i larm och händelselistor. Denna text är enradig och maximalt 80 tecken lång. Det finns även utrymme för en längre text, en mertext, som t ex kan innehålla ytterligare förklaring till larmets orsaker eller hur det bör åtgärdas. Mertexten visas i larm och händelselista om man placerar markören på larmtexten.

Kvittering

Ett larm måste kvitteras av operatören. Så länge larmet är okvitterat, ligger det kvar i larmlistan och ger upphov till en ljudsignal. Om ett larm visas på flera operatörsplatser räcker det med att larmet kvitteras från en av operatörsplatserna.

Blockering

Om ett larm inte relevant under en period, kan man blockera detta. Man kan t ex blockera larmen från en anläggningsdel som är tagen ur drift. Blockeringen kan ske för ett enskilt

larmobjekt, eller för en hierarki. Blockering är en metod för ett objekt, och utförs från ett blockeringsfönster som kan öppnas från popupmenyn för objektet. För att en användare ska kunna blockera måste han ha privilegiet RtEvents.

Blockerade objekt visas i en blockeringslista.

Meddelanden

Info-meddelanden är en grupp som har samma funktion som larm. De har lägre prioritet än larm och markeras med grönt.

Händelser

En händelse genereras på samma sätt som larm, med hjälp av övervakningsobjekt. Även larm genererar händelser. Aktivering av ett larm räknas som en händelse, och även när larm tillståndet försvinner och när larmet kvitteras.

Händelserna lagras i en händelselista, som innehåller de senaste händelserna. De lagras också i en händelselogg, där man kan gå tillbaka och titta på händelser en tid tillbaka, och även se statistik på händelser. För händelseloggen finns en sökfunktion, där man kan söka på händelser med olika sökbegrepp, t ex tid, händelsetyp eller händelsetext.

Urvalslista

Operatörsplatsen har en urvalslista innehåller en lista på anläggningsdelar i objektsträdet. Enbart larm som tillhör dessa anläggningsdelar kommer att visas i larm och händelselistan, övriga larm och händelser filtreras bort.

5.2 Larmlista

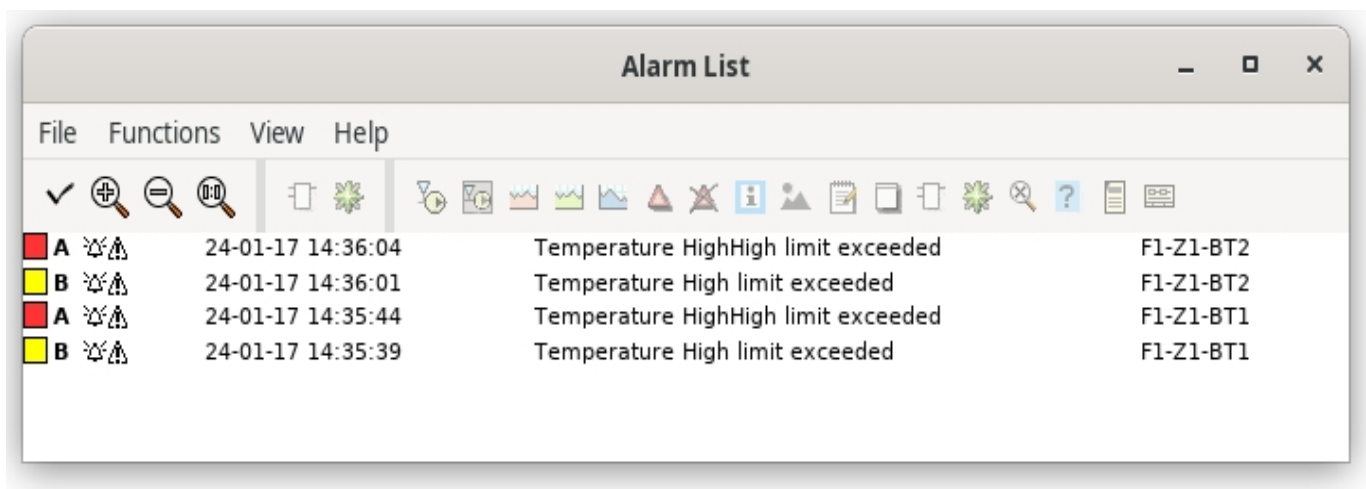


Fig Larmlista

Larmlistan öppnas genom att aktivera knappen "Larmlista" i operatörfönstret. Man kan också öppna larmlistan från menyn i navigatorn, Alarm/Alarm List, och med xtt kommandot 'show alarmlist'.

I larmlistan visas rådande och okvitterade larm. Okvitterade larm markeras med en klocka

och rådande med en varningstriangel. Endast larm från de anläggningsdelar som är angivna i urvalslistan visas.

Mertexten för att larm visas om man placerar markören på larmtexten.

Larm kan kvitteras genom att aktivera Functions/Acknowledge (Ctrl+K) i menyn.

Om man högerklickar på en larmtext, visas metoderna för larmobjektet. Det gör att man enkelt kan visa objektsbilden eller se var i plckoden larmet genereras.

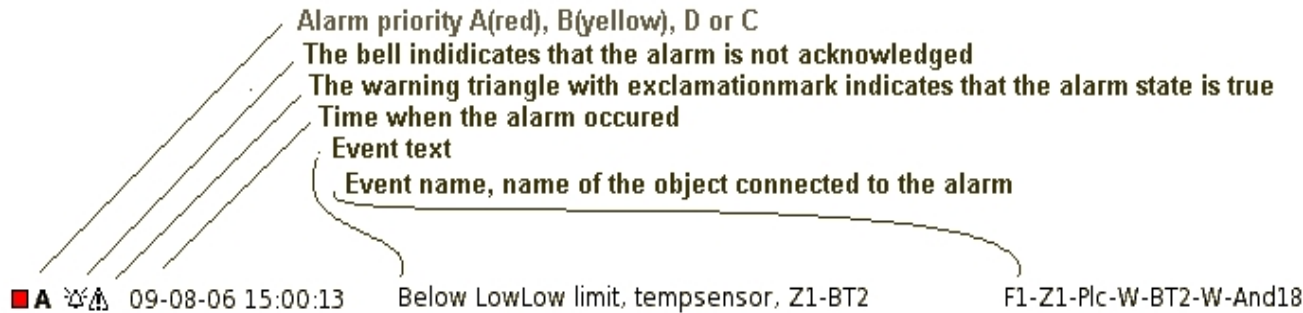


Fig Larmtext

5.3 Händelselista

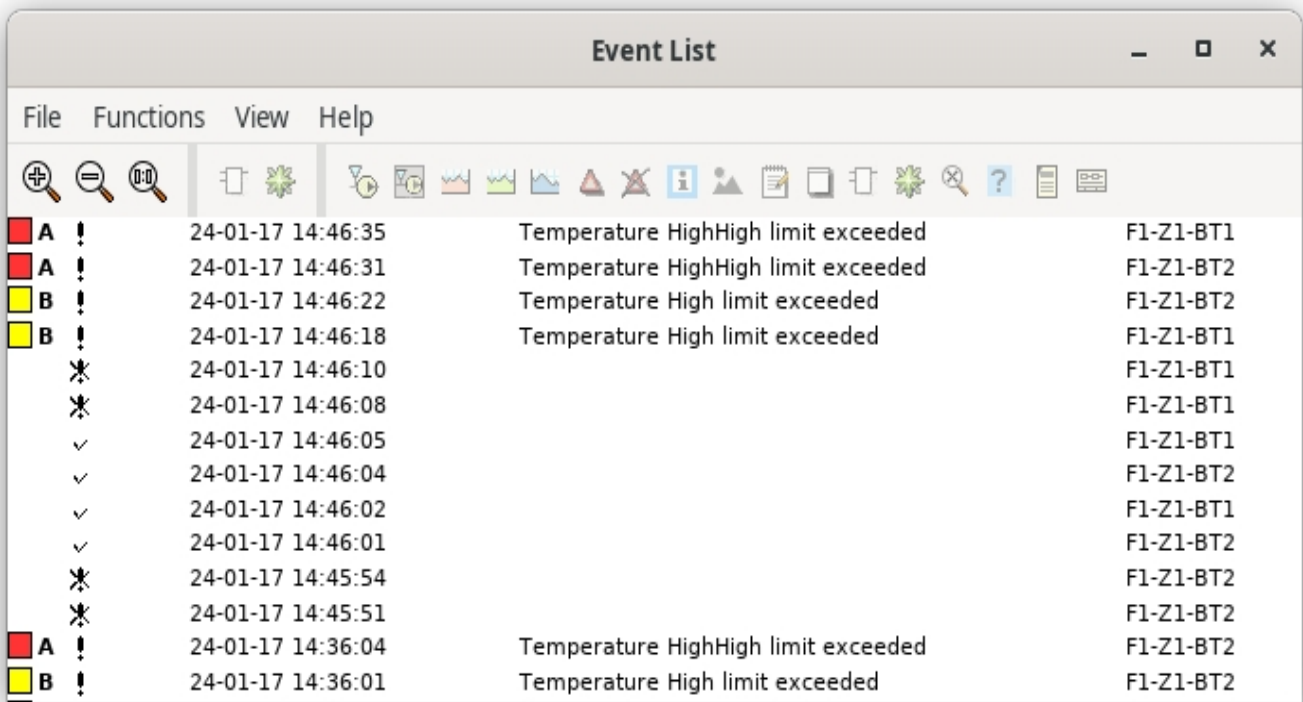


Fig Händelselista

Händelselistan öppnas från knappen "Händelselista" i operatörsfönstret. Man kan även starta den från menyn i navigatören eller med xtt kommandot 'show eventlist'.

Händelser från de anläggningsdelar som finns angivna i urvalslistan visas i händelselistan. Antalet händelser konfigureras i User objektet, och när den maximala antalet händelser är

uppnått, kastas gamla händelser när nya anländer.

För varje händelse visas

- Färgindikering för larm prioritet.
- Händelse typ.
- Händelse tid.
- Händelse text.
- Händelse objekt (objekt/signal namn).

Mertexten för en händelse visas när markören placeras på händelsetexten.

Om man högerklickar på en händelsetext, visas metoderna för händelseobjektet. Det gör att man enkelt kan visa objektbilden eller se var i koden händelsen genereras.

5.4 Händelselogg

Händelselistan är av begränsad längd och visar endast de senaste händelserna. Alla händelser lagras emellertid in en databas. Med den historiska händelselistan, eller händelseloggen, är det möjligt att titta på händelser för en tid tillbaka.

Eftersom antalet lagrade händelser ofta uppgår till fler tusen, söker man efter de händelser man är intresserad av med olika sökvillkor. De händelser som passar in på sökvillkoren visas i en lista.

Sökvillkor

Dialogen för händelseloggen visas i figuren Händelselogg nedan.

Tid

På översta raden kan man ange ett tidsintervall för sökningen. Ställ in önskat intervall med optionmenyn till höger. Väljer man 'Time' anger man starttid och stoptid i inmatningsfälten.

Händelsetyp

Här kan man markera händlestyper. Om ingen type är markerad, tas alla typer med i sökningen.

- | | |
|-----------|---|
| - Active | Händelsen att ett larm aktiveras. |
| - Message | Ett info meddelande. |
| - Return | Ett larm återgår från att ha varit rådande. |
| - Ack | Ett larm kvitteras. |

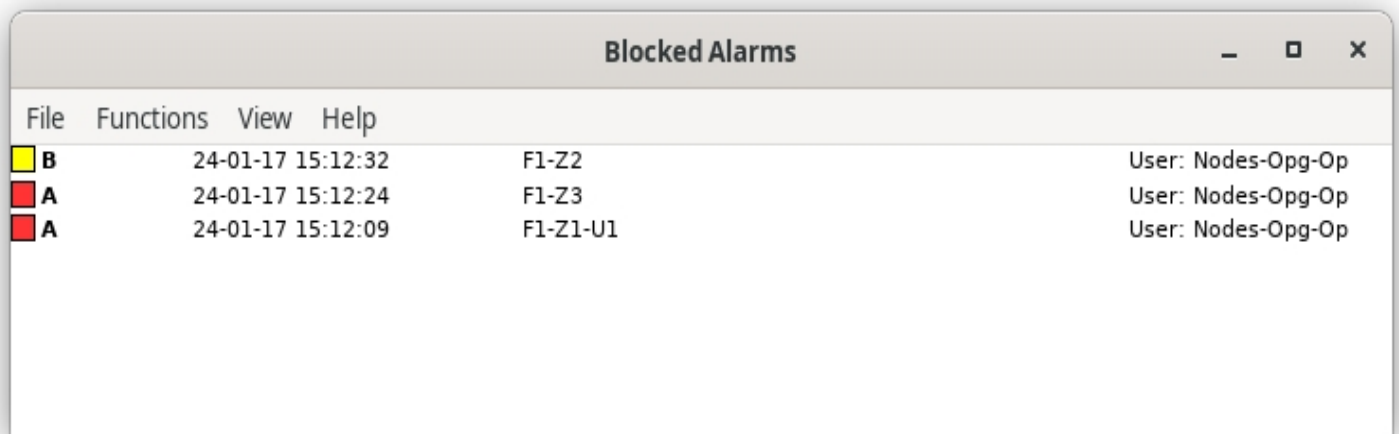
Prioritet

Här kan man markera att man vill söka på larm med en viss prioritet. Om ingen prioritet anges utförs sökningen på alla prioriteter.

Händelsenamn

Namn på det objekt som händelsen är kopplad till. I namnet kan ingå wildcard (*), t ex F1-Z1* söker på alla event under hierarkin F1-Z1. Om inte något objektsnamn anges, söks på alla objekt.

5.5 Blockeringslista



Blocked Alarms			
File	Functions	View	Help
 B	24-01-17 15:12:32	F1-Z2	User: Nodes-Opg-Op
 A	24-01-17 15:12:24	F1-Z3	User: Nodes-Opg-Op
 A	24-01-17 15:12:09	F1-Z1-U1	User: Nodes-Opg-Op

Fig Blockeringslista

Listan över blockerade larm öppnas från menyn i navigatören, Alarm/Blocked Alarms, eller med xtt kommandot 'show blocklist'.

För varje blockerat objekt visas:

- Färgindikering som markerar den prioritetsnivå som har blockerats.
- Objektsnamn.
- Användare som har utfört blockeringen.

En blockering tas bort genom att blockeringen markeras och Functions/Remove Blocking i menyn aktiveras.

Blockering av larm sker genom 'Block Events' metoden för ett objekt.

5.6 Blockera larm

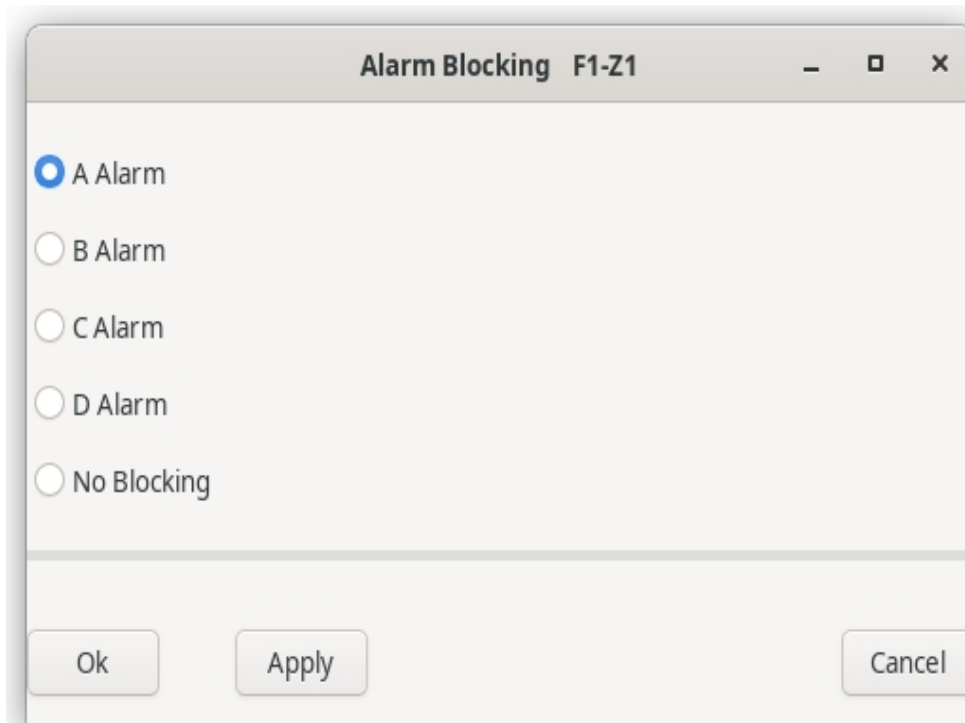


Fig Dialog för larmblockering

Larmblockering aktiveras genom 'Block Events' metoden, dvs från popup-menyn för ett objekt, eller med xtt kommandot 'eventlist block'.

Fönstret används för att blockera larm för ett objekt. Alla larm med den specificerade prioriteten, och med lägre prioritet, blockeras för objektet och underliggande objekt.

Prioriteten specificeras med radio-knappar. Man kan även ta bort en blockering med knappen 'No Blocking'.

Endast användare med privilegiet 'RtEvent' eller 'System' är auktoriserade att blockera larm.

6 Kurvor

6.1 Trendkurvor

Trendkurvor öppnas med 'Trend' metoden i t ex popumenyn för ett objekt. De kan även öppnas med Xtt kommandot 'open trend'. Trendkurvor konfigureras med DsTrend objekt, och om flera kurvor ska visas i samma kurvfönster, med PlotGroup objekt.

En trendkurva är en signal, t ex av typen temperatur, tryck eller flöde, vars värde lagras med ett visst tidsintervall. En trendkurva har plats för 478 värden, och hur lång tid tillbaka värdena kan lagras beror på hur ofta lagringen sker. Med ett nytt värde per sekund kan ca 8 minuter lagras, med nytt värde varje minut blir det ca 8 timmar. Värdena lagras i RAM-minnet och vid en omstart försvinner alla lagrade data.

Trendkurvorna visas i ett kurvfönster. Ett kurvfönster kan visa upp till 20 olika kurvor.

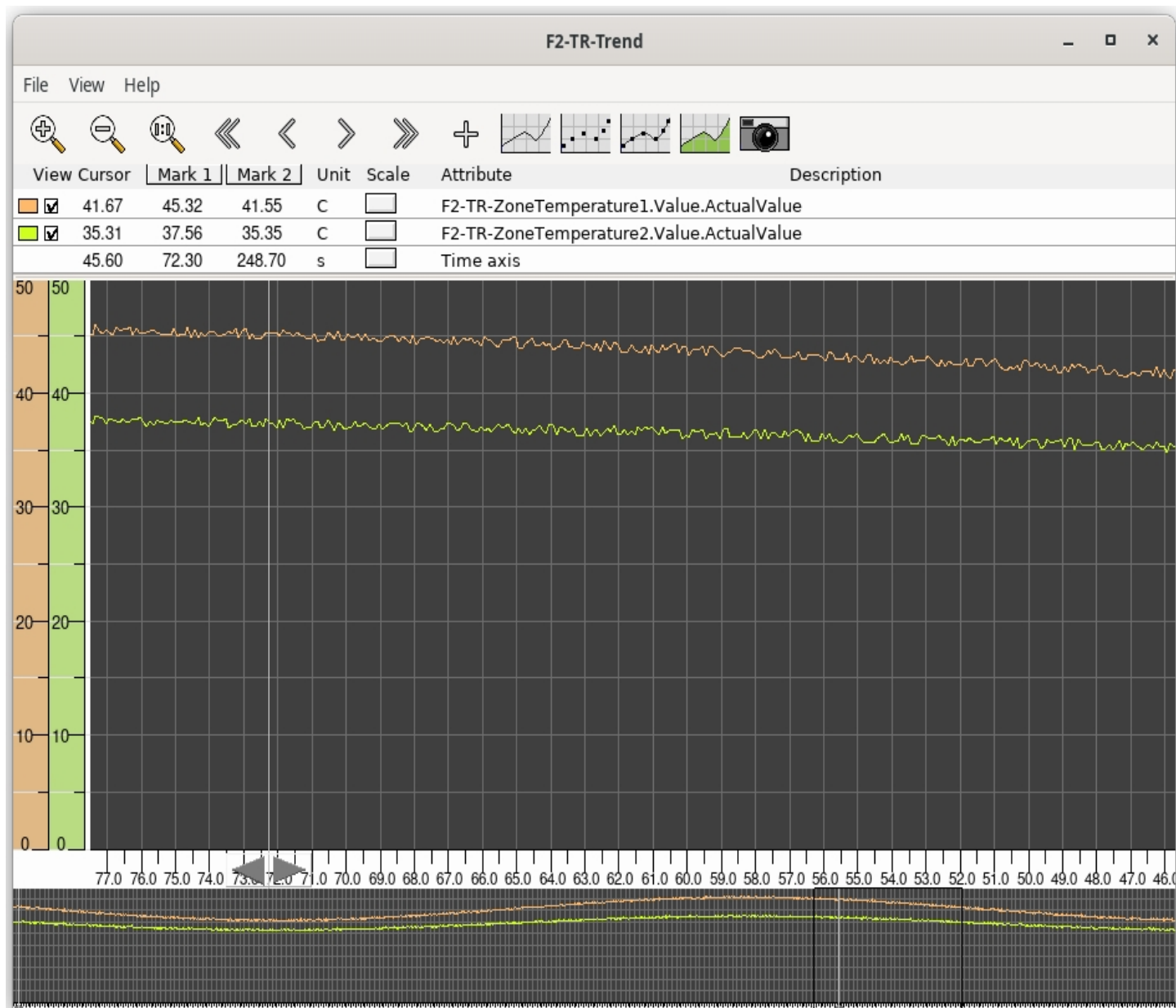


Fig Trendkurvor

I kurvfönstrets övre del visas en tabell över de kurvor som visas. Första kolumnen är en färgmarkering för att identifiera den korresponderande kurvan.

Beskrivning av kolumner i tabellen

View	En checkbox med vilken man kan välja om en kurva ska visas eller inte.
Cursor	Visar värdet för kurvan vid den tidpunkt där cursorn är placerad.
Mark1	Visar värdet för kurvan vid den tidpunkt där den första markören är placerad.
Mark2	Visar värdet för kurvan vid den tidpunkt där den andra markören är placerad.
Unit	Visar enhet för kurvan.
Scale	Från 'Scale' kan man ställa in skal-området i y-led (Scale för raden Time axis ställer om i x-led).
Attribute	Anger det attribut som trendkurvan visar.
Description	Beskrivning av attributet.

Navigering

Verktygspanelen innehåller knappar för att zooma in och ut, och för att flytta visningen åt vänster eller höger.

Man kan även navigera med navigationsrutan i fönstrets undre del. Den del av kurvan som visas markeras med en svart fyrkant. Genom att dra med MB1 kan man flytta den i x-led, och genom att dra med mittenknappen, kan man zooma in eller ut.

Navigering med tangentbordet sker med pil tangenterna. Pil upp och ner ändrar zoomningsgraden och pil vänster och höger flyttar visningen åt höger resp vänster.

Markörer

Det finns två markörer i fönstret som kan positioneras till specifika tider. Kurv-värdet för positionen visas i tabellen. Markörerna kan flyttas med cursorn, eller genom att klicka med MB1 (första markören) eller Ctrl+MB1 (andra markören) i kurv-fönstret.

Snapshot

För vissa trend-kurvor är det möjligt att ta en ögonblicksbild av trenden för att undersöka den noggrannare eller spara den till fil. Ögonblicksbilden aktiveras med kamera-knappen i verktygspanelen, då ett nytt fönster med ögonblicksbilden visas.

Addera kurvor

Flera kurvor kan adderas till trendfönstret genom att välja ut ett DsTrendCurve objekt i runtime-navigatören och aktivera '+'-knappen i verktygspanelen.

Rutnät

När flera kurvor visas kan rutnätet anpassas till skalan för en specifik kurva genom att klicka på den färgade skalan för kurvan.

Exportera till text-fil

En trend kan exporteras till text-fil från Arkiv/Exportera i menyn. I export dialogen kan ett specifikt attribut väljas för exportering, eller alla attribut. För att undvika att skriva över tidigare exporterade filer kan filnamnet innehålla strängen '\$date' som kommer att ersättas med aktuell tid i det verkliga filnamnet.

6.2 Trend ögonblicksbild

Trend ögonblicksbilden är ett snapshot av trendkurvan tagen vid en specifik tillfälle.

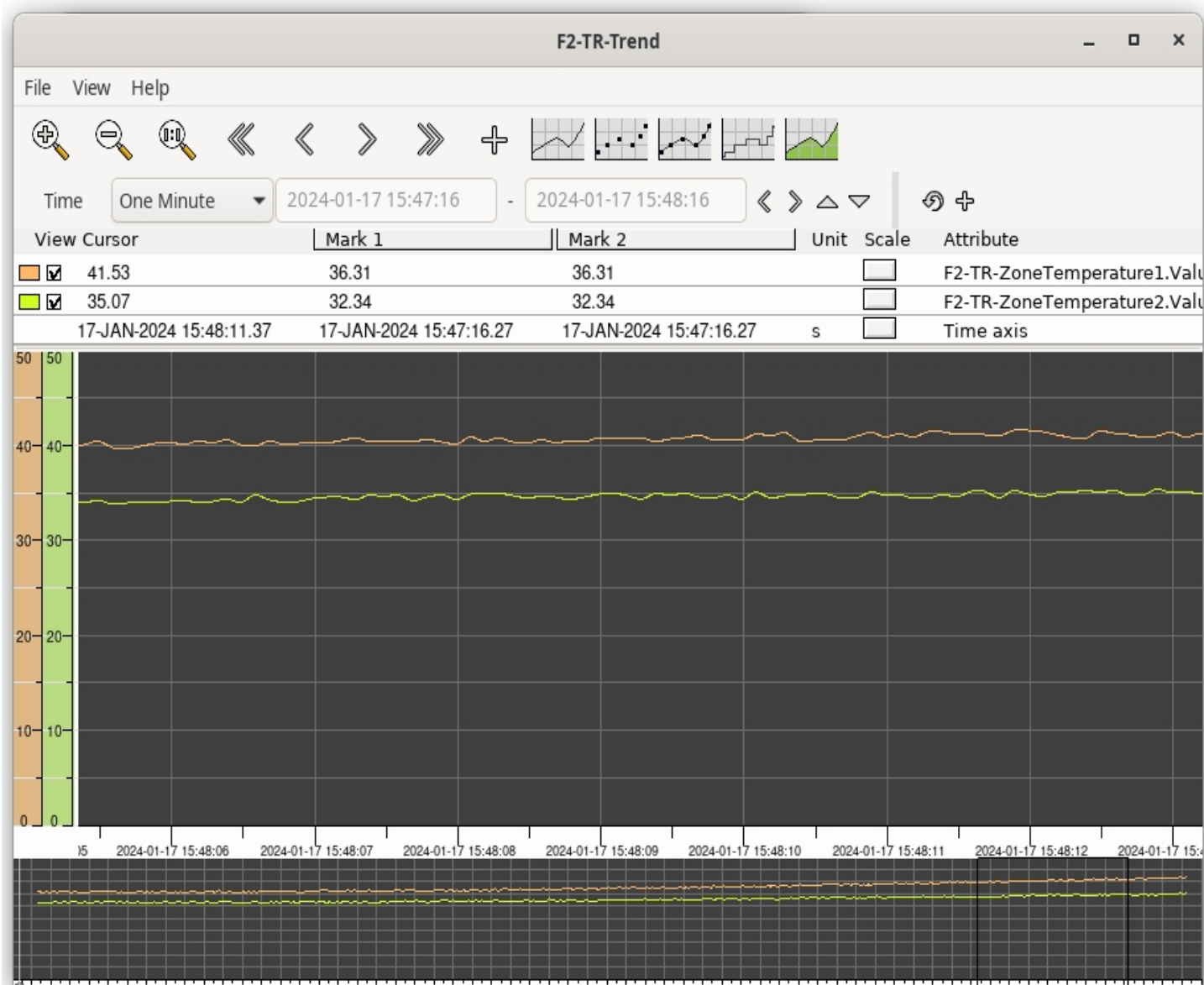


Fig Trend snapshot

Ett tidsintervall kan väljas i verktygspanelen för tider. När ett intervall har valts trycks på knappen 'Uppdatera kurva' för att rita om kurvan.

Spara kurva

En snapshot-kurva kan lagras genom att aktivera Arkiv/Spara i menyn.

Öppna sparad kurva

En sparad snapshot kurva öppnas från Arkiv/Öppna i menyn. För att öppna i ett nytt fönster aktiveras Arkiv/Ny för att öppna ett tomt snapshot fönster, och sedan Arkiv/Öppna i detta fönster.

6.3 Snabbkurvor

Snabbkurvor öppnas med 'Fast' metoden i t ex popumenyn för ett objekt. De kan även öppnas med Xtt kommandot 'open fast'. Snabbkurvor konfigureras med DsFastCurve objekt, och om flera kurvor ska visas i samma kurvfönster, med PlotGroup objekt.

Snabbkurvor används för att visa snabba förlopp under en kortare tidsperiod. På en triggssignal startas lagringen av signalens värde och lagringen pågår under en tid. När förloppet är avslutat visas kurvan upp i kurvfönstret, och ligger kvar där tills triggssignalen sätts och aktiverar en ny inspelning av förloppet.

Snabbkurvorna visas i ett kurvfönster. Ett kurvfönster kan visa upp till 20 olika kurvor.

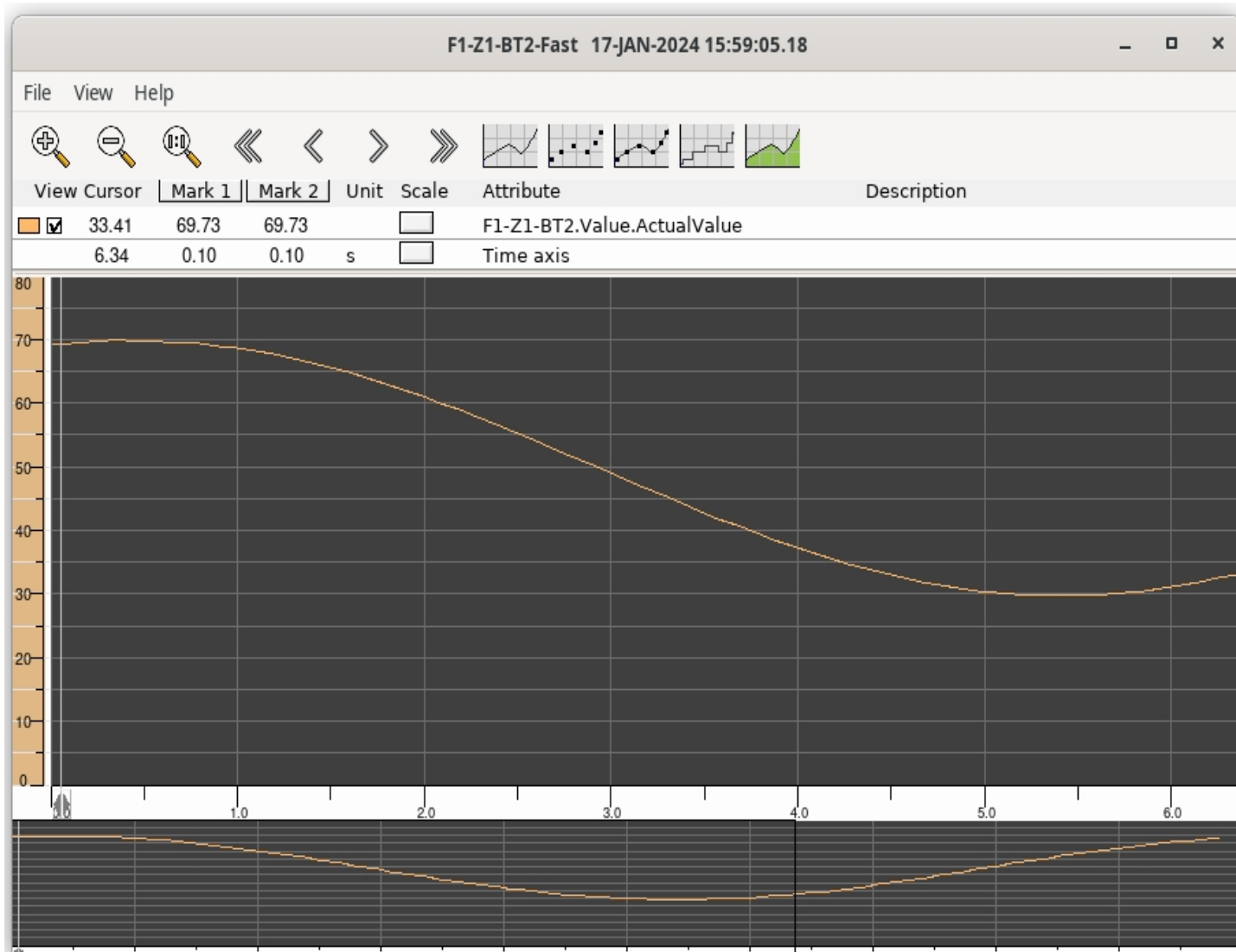


Fig Snabbkurva

6.4 Process historik

Process historik öppnas med 'History' metoden i t ex popumenyn för ett objekt. De kan även öppnas med Xtt kommandot 'open history'. Historiken konfigureras med SevHist objekt.

Process historik innebär att värdet på en signal, t ex av typen temperatur, tryck eller

flöde, vars värde lagras i en databas med en viss frekvens och under en viss tid. Historiken kan sträcka sig över flera år och antal mätvärden för en signal kan uppgå till flera miljoner.

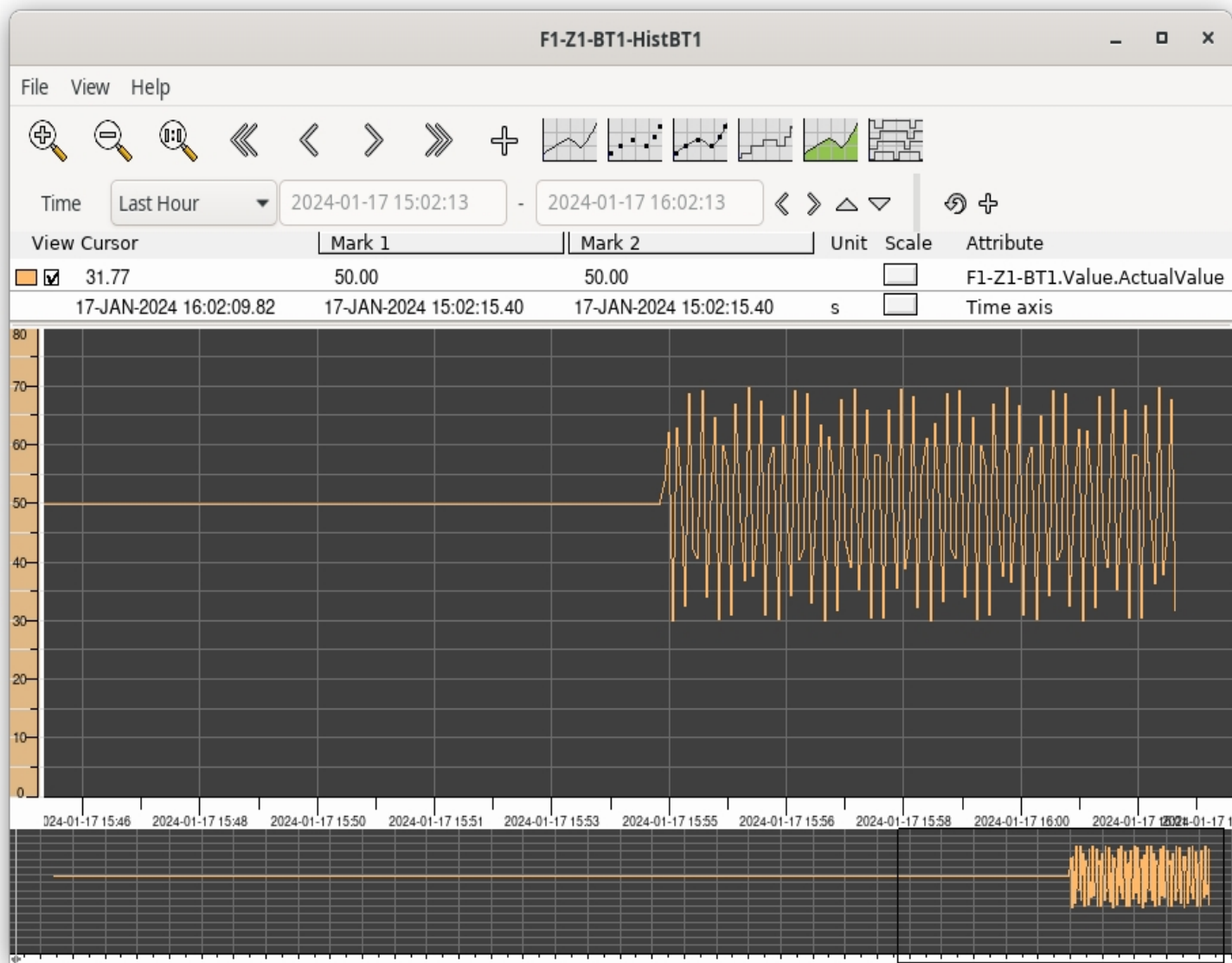


Fig Process historik

I kurvfönstrets övre del visas en tabell över de kurvor som visas. Första kolumnen är en färgmarkering för att identifiera den korresponderande kurvan.

Beskrivning av kolumner i tabellen

View	En checkbox med vilken man kan välja om en kurva ska visas eller inte.
Cursor	Visar värdet för kurvan vid den tidpunkt där cursorn är placerad.
Mark	Visar värdet för kurvan vid den tidpunkt där markören är placerad.
Unit	Visar enhet för kurvan.
Scale	Från 'Scale' kan man ställa in skal-området i y-led (Scale för raden Time axis ställer om i x-led).
Attribute	Anger det attribut som historik-kurvan visar.

Navigering

Verktyspanelen innehåller knappar för att zooma in och ut, och för att flytta visningen

åt vänster eller höger.

Man kan även navigera med navigationsrutan i fönstrets undre del. Den del av kurvan som visas markeras med en svart fyrkant. Genom att dra med MB1 kan man flytta den i x-led, och genom att dra med mittenknappen, kan man zooma in eller ut.

Navigering med tangentbordet sker med pil tangenterna. Pil upp och ner ändrar zoomningsgraden och pil vänster och höger flyttar visningen åt höger resp vänster.

Upplösning

Eftersom antalet mätvärden för ett attribut kan uppgå till flera miljoner, hämtar man bara upp ett urval av värden. När man först öppnar kurvan, visas hela tidsintervallet med några hundra punkter. För att få högre upplösning använder man knapparna med pil upp och pil ner i verktygspanelen. Pil upp zoomar in och ökar upplösningen i det tidsintervall som visas, dvs man läser in nya punkter i ett snävare tidsintervall med högre upplösning. Pil ner zoomar ut med minskad upplösning.

7 Hjälpfönstret

Att få hjälp

Hjälpfunktionen kan öppnas på följande sätt

Operatörsfönstret

I operatörsfönstrets mitt-del finns en hjälpknapp som visar hjälptexter för projektet.

Navigatorns meny

Menyentryt Help/Overview ger hjälp om ProviewR och operatörsmiljön. Help/Project ger hjälp om projektet.

Kommando

En hjälptext öppnas med xtt-kommandot 'help'. Till help skickar man det ämne man är intresserad av. Kommandot help help visar t ex hur man använder help.

Objektsmetoder

För objekt finns det två metoder som ger hjälp,
- Help, visar hjälp för objektet eller anläggningsdelen.
- Help Class, visar information om objektets klass.

Hjälp och infoknappar i processbilder

Tryckknappar i bilder kan kopplas till hjälptexter. Dessa är ofta markerade med ett frågetecken eller ett 'i'.



Fig Hjälpknappar

Navigera i hjälptexten

Rader med länkar markeras med en pil i radens vänsterkant. Först raden är en länk till föregående ämne.

Navigera från tangentbordet

PageUp och PageDown används för att scrolla upp och ner i texten. Man följer länkar genom att välja ut raden med länken med Pil-upp och Pil-ner, och sedan trycka på Pil-höger. Med Pil-vänster återgår man till föregående sida.

Med Ctrl+N går man till nästa avsnitt, och med Ctrl+P till föregående avsnitt.

Ctrl+A återgår till startsidan.

Navigera med musen

För att följa en länk klickar man på pilen, eller dubbelklickar på raden.

Genom att klicka på pilen på första raden återgår man till föregående sida.

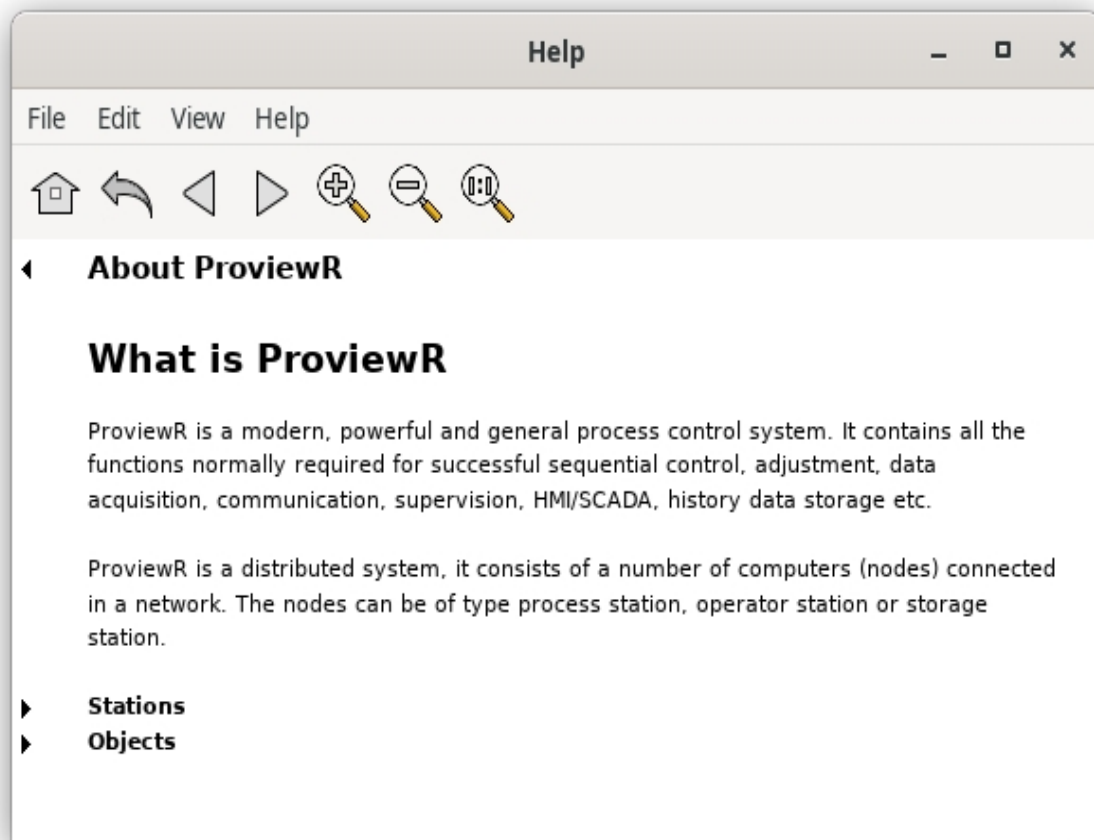


Fig Hjälpfönstret

8 Navigatör

Navigatören, även kallad Xtt, startas från knappen "Navigator" i operatörsfönstret. Den kan även startas som ett separat program, oberoende av operatörmiljön. Det görs ofta vid felsökning eller underhåll av systemet. Man startar då från ett terminalfönster.

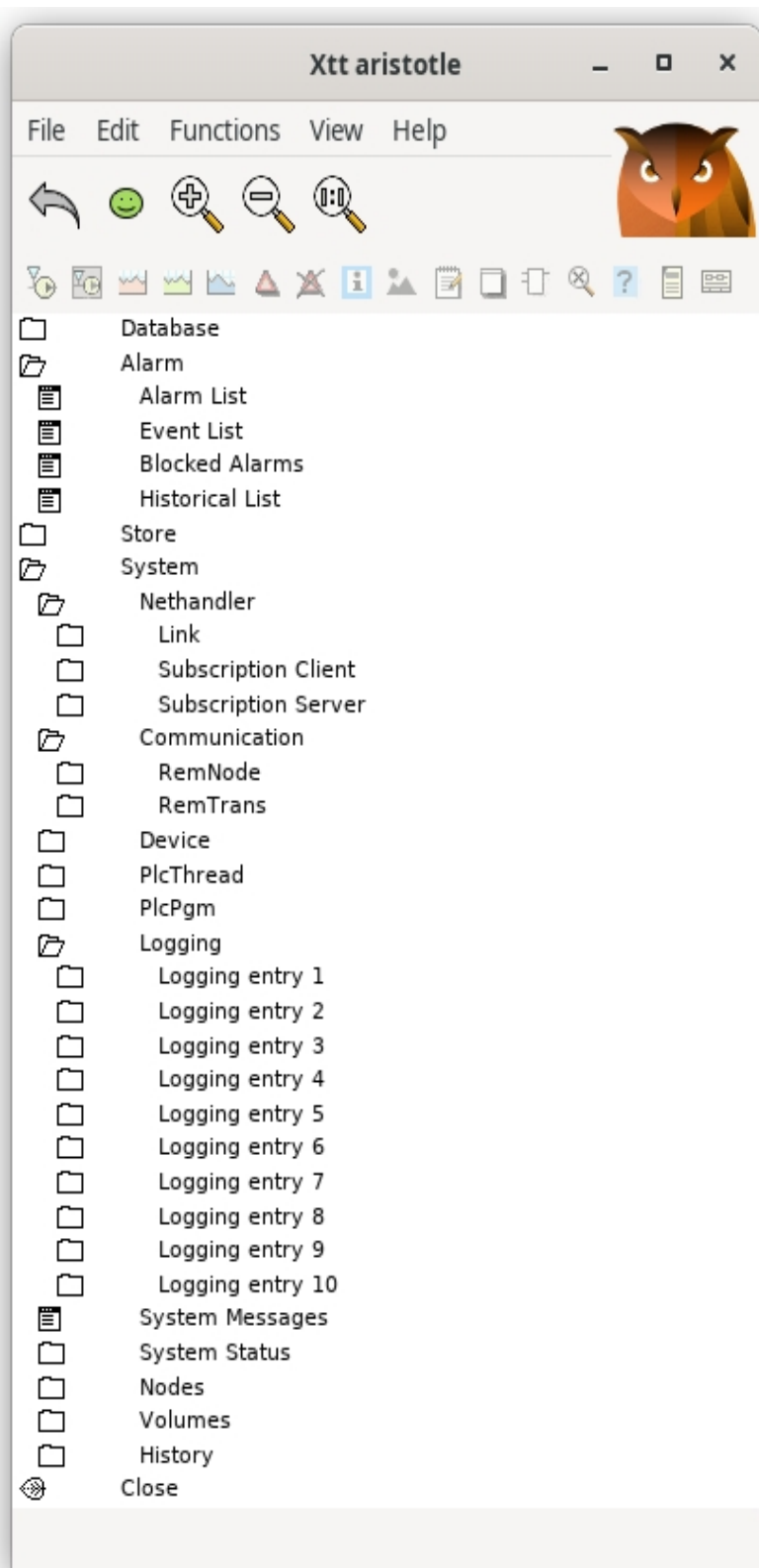


Fig Menyn i navigatorn

Figuren ovan visar menyn i navigatorn. Den översta mappen, "This Project" är konfigurerbar och kan innehålla en mapstruktur som är specifik för projektet (ofta saknas den helt). Övriga mappar är standard,

Database

Visar objektsträdet och objekten i databasen

Alarm / Alarm List	Öppnar larmlistan.
Alarm / Event List	Öppnar händleselistan.
Alarm / Blocked Alarms	Visar listan över blockerade larm.
Alarm / Historical List	Visar händelseloggen.
Store	Visar sparade samlingsbilder.
System / Nethandler / Link	Visar länkar till andra ProviewR system.
System / Nethandler / Subscription Client	Visar prenumerationer (client).
System / Nethandler / Subscription Server	Visar prenumerationer (server).
System / Communication / RemNode	Visar remote kommunikation (noder)
System / Communication / RemTrans	Visar remote kommunikation (transar)
System / Device	Visar I/O enheter
System / PlcThread	Visar Plc trådar
System / PlcPgm	Visar PlcPgm.
System / Logging	Dialog för att starta logging av data.
System / System Messages	Visar systemmeddelanden.
System / System Status	Visar systemstatus för noden.
System / Nodes	Visar statusbild för noder.
System / Volumes	Visar laddade volymer.
Close	Stänger navigatören.

Navigera

Navigera från tangerbordet

Det mest ergonomiska sättet att navigera i mapstrukturen är att använda piltangenterna. Med Pil-Upp och Pil-Ned väljer man ut en map eller ett löv. Med Pil-Höger öppnar man en mapp och med Pil-Vänster stänger man.

Ett objekt i databasen öppnar man med Shift Pil-Höger, dvs man trycker först ner Shift-tangenten och håller den nedtryckt medan man trycker på Pil-Höger. Om objektet inte har några barn räcker det att trycka på Pil-Höger.

Har man privilegiet RtWrite kan man ändra värden i attribut i databasen. Genom att öppna ett objekt, välja ut attributet och trycka på Pil-Höger öppnas ett inmatningsfält där man kan mata in ett nytt värde.

Navigera med musen

Man naturligtvis även navigera med musen. En mapp öppnas genom att man klickar på mappen (eller dubbelklickar på texten). Sedan stänger man genom att återigen klicka på mappen.

Ett objekt i databasen öppnas genom att trycka in Shift och klicka på mappen/lövet för objektet.

Kommandon

Med Functions/Command, Ctrl+B, öppnar man en kommando-prompt på nedersta raden i navigatören. Härifrån kan man mata in kommandon och exekvera skript. Tillgängliga kommandon finns beskrivna i kapitlet Kommandon och skript i kapitlet Script

Samlingsbild

Men Functions/Collect/Show, Ctrl+N, tittar man på insamlade objekt och attribut. Vid felsökning vill man ofta titta på ett antal olika attribut från olika objekt samtidigt. Genom att gå runt i objekts-trädet och samla in attribut med Functions/Collect/Insert, Ctrl+V, och sedan aktivera Functions/Collect/Show visas dessa attribut gemensamt på en sida.

Om man vill spara samlingsbilden till en senare session, görs det med 'save' kommandot. För att spara en samlingsbild under namnet 'mycollection' skriver man kommandot

```
> store/collect mycollection
```

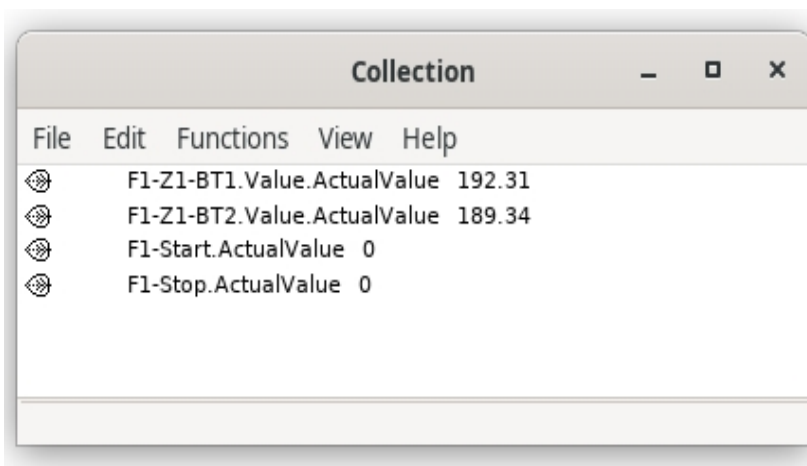


Fig Separat samlingsbild

Bilden kommer att återfinnas under 'Store' mappen, och kan även återskapas med kommandot

```
> @mycollection
```

Samlingsfönster

En samlingsbild kan kopieras till ett separat fönster med /Functions/Collect/Copy To Window. Även ett tomt fönster kan skapas från Functions/Collect/New Window. Nya attribut kan adderas till förstret genom att välja ut attributet i runtime navigatören, och aktivera Edit/Insert (Ctrl+V) i samlingsfönstret. Med Shift+PilUpp och Shift+PilNer kan attribut flyttas upp resp. ner i fönstret.

Ett samlingsfönster sparas från Edit/Save (Ctrl+S) med ett specificerat namn. Det kan sen öppnas från Functions/Collect/Open i runtime navigatören, eller från File/Open i ett samlingsfönster. Det önskade fönstret öppnas från en lista med alla sparade samlingsfönster.

8.1 Database

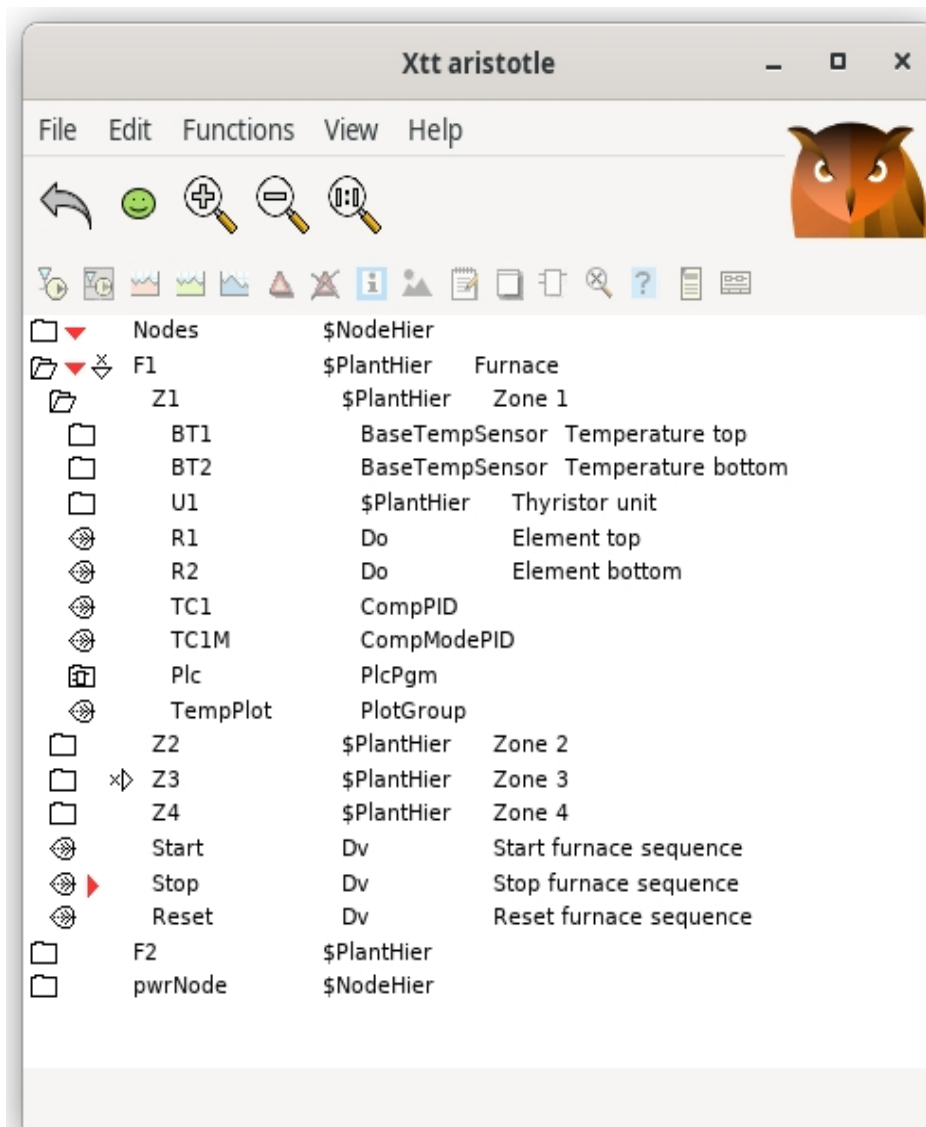


Fig Objektträdet i navigatören

Under mappen 'Database' visas objektträdet. Här kan man navigera i objektsträdet, öppna objekt och titta på innehållet i olika attribut i objektet. Om man har privilegier kan man även ändra på attributvärden.

Högerklickar man på ett objekt öppnas en popupmeny med objektets metoder. Knappar för att aktivera objektets metoder finns även på den undre verktygspanelen i navigatören. Endast metoder som är relevanta för ett objekt är tryckkänsliga.

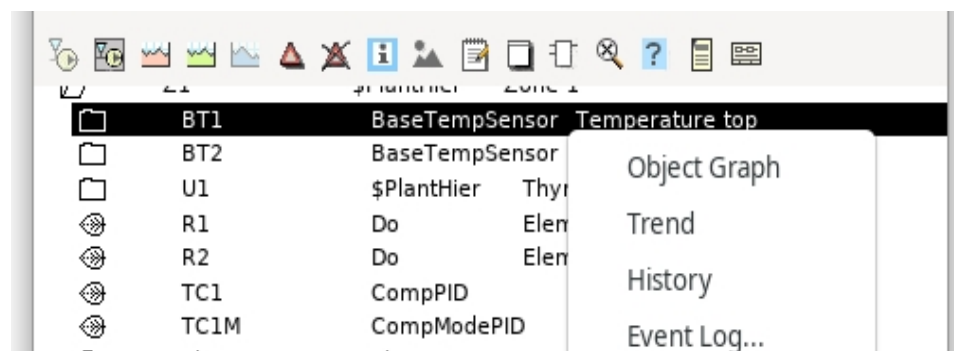
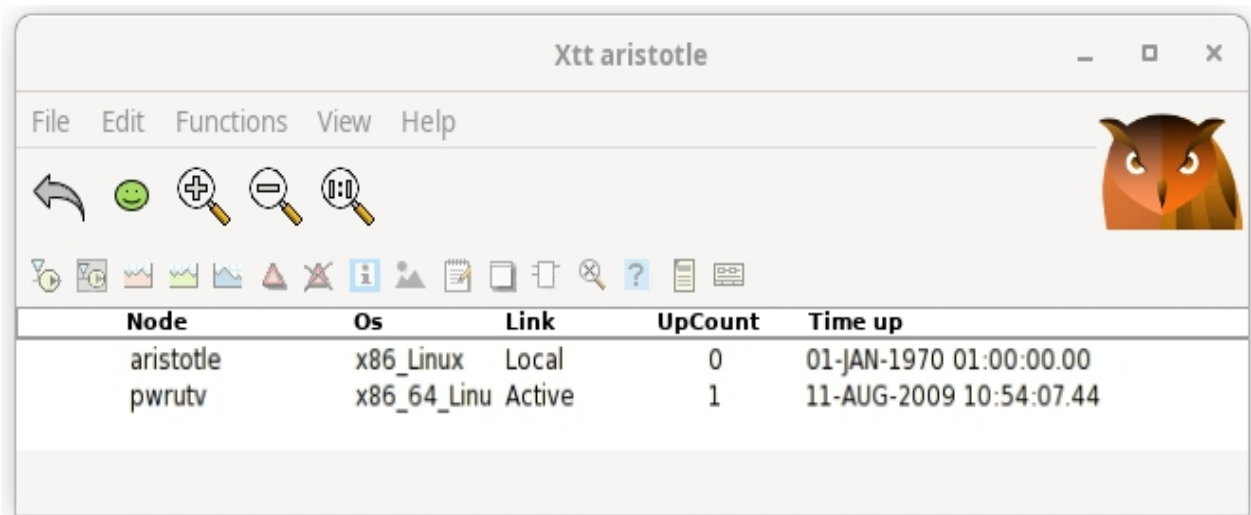


Fig Metoderna kan aktiveras både från verktygspanelen och popupmenyn

8.2 System / Nethandler / Link

Mappen 'System / Nethandler / Link' visar länkar till andra ProviewR-stationer. Även den egna noden finns med i listan.

I figuren nedan kan man se att den lokala noden aristotle, har kontakt med noden pwrutv.



The screenshot shows a window titled 'Xtt aristotle' with a menu bar (File, Edit, Functions, View, Help) and a toolbar with various icons. Below the toolbar is a table with the following data:

Node	Os	Link	UpCount	Time up
aristotle	x86_Linux	Local	0	01-JAN-1970 01:00:00.00
pwrutv	x86_64_Linu	Active	1	11-AUG-2009 10:54:07.44

Fig Länkar till andra noder

Beskrivning av kolumnerna i bilden

Node	Nodnamn.
Os	Operativsystem och hårdvara för noden.
Link	Status för länken. Kan vara Up, Active, Connected, Down eller Local.
UpCount	Antal gånger länken till noden har etablerats.
TimeUp	Tid för senaste etablering av länken till noden.

8.3 System / Nethandler / Subscription Client

Bilden visar prenumerationer som den aktuella stationen har lagt upp på andra stationer.

När en operatörstation visar data från processtationer i en processbild, läggs det upp en prenumeration på varje data som visas. Det innebär att processtationerna med jämna mellanrum skickar nya data till operatörsstationen, som då uppdaterar processbilden med dessa data.

Om man har ett fält i en processbild som inte uppdateras, kan man gå in Subscription Client tabellen och se om prenumerationen finns med och om den uppdateras. Varje gång nya data tas emot för en prenumeration räknas 'Count' upp och i 'Time' läggs tidpunkten in.

På första raden, i 'Unknown', kan man se antalet prenumerationer där attributet är okänt. Det kan vara prenumerationer på en station som ännu inte är i drift, men det kan också vara prenumerationer som är felstavadade, eller på objekt som inte längre existerar.

Xtt aristotle					
File Edit Functions View Help					
 					
					
Local:		0	Remote:	0	Unknown: 1
Subid	Time	Count	Node	Size	Attribute
_S128.1.99.72:2	11:17:46.42	149	aristotle	4	Test-V1-Plc-Logic-Reset.ActualValu
_S128.1.99.72:3	11:17:46.42	149	aristotle	4	Test-V1-Plc-Logic-Error.ActualValue
_S128.1.99.72:4	11:17:46.42	148	aristotle	4	Test-V1-Plc-Logic-Success.ActualV
_S128.1.99.72:5	11:17:46.42	148	aristotle	4	Test-V1-Plc-Logic-Active.ActualValu
_S128.1.99.72:6	11:17:46.42	147	aristotle	4	Test-V1-Plc-Logic-Dv1.ActualValue
_S128.1.99.72:7	11:17:46.42	147	aristotle	4	Test-V1-Plc-Logic-Dv2.ActualValue
_S128.1.99.72:8	11:17:46.42	146	aristotle	4	Test-V1-Plc-Logic-Dv3.ActualValue
_S128.1.99.72:9	11:17:46.42	144	aristotle	4	Test-V1-Plc-Logic-ResAnd.ActualVa
_S128.1.99.72:10	11:17:46.42	143	aristotle	4	Test-V1-Plc-Logic-ResOr.ActualValu

Fig Klient prenumerationer

Beskrivning av kolumnerna

Subid	Identitet på prenumerationen.
Time	Senaste ankomsttid för prenumurationsdata.
Count	Antal gånger data har mottagits.
Node	Nod som skickar data för prenumerationen.
Size	Storlek i bytes på prenumurations-data.
Attribute	Det attribut i databasen som prenumerationen gäller.

8.4 System / Nethandler / Subscription Server

Bilden visar prenumerationer som andra stationer har lagt upp på den aktuella noden, dvs prenumertioner som skickas cykliskt från den aktuella noden till andra noder, vanligtvis från processtationer till opertörsstationer.

Xtt aristotle					
File Edit Functions View Help					
 					
					
Subid	Count	Node	Size	Offset	Attribute
_S128.1.99.72:2	76	pwrutv	4	88	Test-V1-Plc-Logic-Reset.ActualVal
_S128.1.99.72:3	76	pwrutv	4	88	Test-V1-Plc-Logic-Error.ActualValu
_S128.1.99.72:4	75	pwrutv	4	88	Test-V1-Plc-Logic-Success.Actual\
_S128.1.99.72:5	75	pwrutv	4	88	Test-V1-Plc-Logic-Active.ActualVal
_S128.1.99.72:6	74	pwrutv	4	88	Test-V1-Plc-Logic-Dv1.ActualValue
_S128.1.99.72:7	74	pwrutv	4	88	Test-V1-Plc-Logic-Dv2.ActualValue
_S128.1.99.72:8	73	pwrutv	4	88	Test-V1-Plc-Logic-Dv3.ActualValue
_S128.1.99.72:9	71	pwrutv	4	88	Test-V1-Plc-Logic-ResAnd.ActualV
_S128.1.99.72:10	70	pwrutv	4	88	Test-V1-Plc-Logic-ResOr.ActualVal

Fig Server prenumerationer

Beskriving av kolumnerna

Subid	Identitet på prenumerationen.
Count	Antal gånger data har skickats.
Node	Nod till vilken data för prenumerationen skickas.
Size	Storlek i bytes på prenumerations-data.
Offset	Offset i det objekt som data hämtas ifrån.
Attribute	Det attribut i databasen som prenumerationen gäller.

8.5 System / Communication / RemNode

Visar en lista noder som den aktuella noden kommunicerar med mha Remote.

Remote är en funktion för att skicka data mellan noder. Noder som man vill kommunicera med konfigureras med RemNode objekt, och olika transaktioner som ska utbytas mellan noderna konfigureras med RemTrans objekt. Ett antal olika protokoll för datautbytet finns inlagt, Modbus, MQ, Serial, TCP, UDP, 3964R och ALCM.

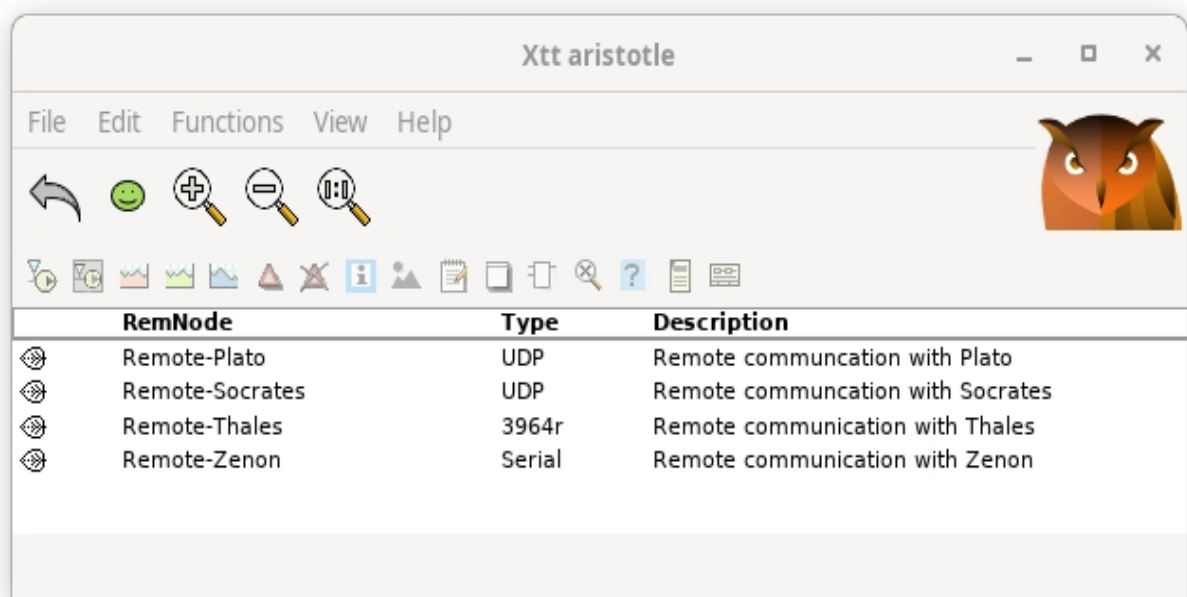


Fig Lista på RemNoder

Beskrivning på kolumnerna

RemNode	Namn på RemNode objektet, endast de två sista namnleden visas.
Type	Typ av kommunikation, Modbus, MQ, Serial, TCP, UDP, 3964R eller ALCM.
Description	Beskrivning.

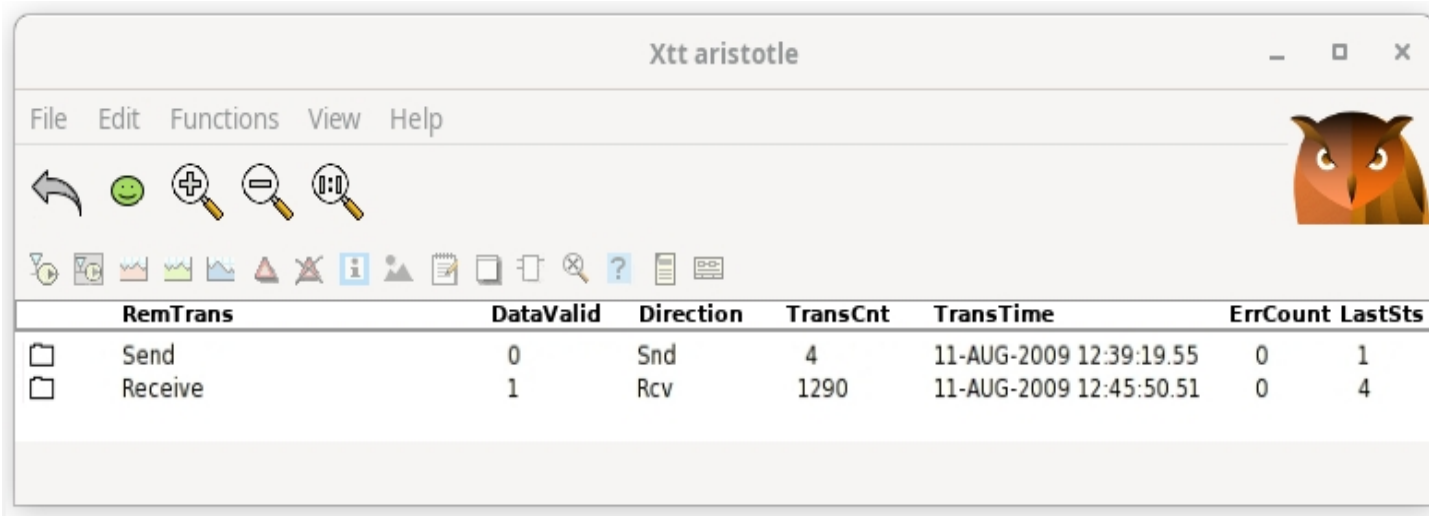
Genom att välja ut en nod och trycka Pil-Höger visas RemTrans-objekten för noden (det går även att klicka på mappen).

Med Shift/Pil-Höger öppnas RemNode objektet.

8.6 System / Communication / RemTrans

Visar en lista på samtliga RemTrans objekt.

Ett RemTrans objekt specificerar en transaktion som skickas mellan två noder mha Remote funktionen. Transaktionen innebär att man skickar eller tar emot data.



The screenshot shows the 'Xtt aristotle' application window. It has a menu bar with 'File', 'Edit', 'Functions', 'View', and 'Help'. Below the menu is a toolbar with various icons including a back arrow, a green smiley face, a magnifying glass with a plus sign, a magnifying glass with a minus sign, and a magnifying glass with a pause symbol. There is also a small owl icon in the top right corner. The main area contains a table with the following data:

	RemTrans	DataValid	Direction	TransCnt	TransTime	ErrCount	LastSts
☐	Send	0	Snd	4	11-AUG-2009 12:39:19.55	0	1
☐	Receive	1	Rcv	1290	11-AUG-2009 12:45:50.51	0	4

Fig Lista på RemTrans-objekt

Beskrivning på kolumnerna

RemTrans	Namn på RemTrans objektet, endast sista namnledet visas.
DataValid	Anger att data finns som ännu ej har skickas (vid sändning) eller tagits hand om (vid mottagning).
Direction	Riktning på transaktionen, Snd (sändning) eller Rcv (mottagning).
TransCnt	Antal sända eller mottagna transaktioner.
TransTime	Tidpunkt för senaste transaktionen.
ErrCount	Antal misslyckade transaktioner.
LastSts	Status för senaste transaktionen.

Genom att välja ut en transaktion och trycka Pil-Höger visas innehållet i transaktionen (det går även att klicka på mappen). Om attributet StructFile är ifyllt med den include fil som innehåller datastrukturen för transaktionen, visas datainnehållet i klartext. Med Shift/Pil-Höger öppnas RemTrans-objektet.

8.7 System / Device

Visar lista med I/O enheter.

I/O systemet i ProviewR är indelat i fyra nivåer, agent, rack, kort och kanal. I Device listan visas alla kort-objekt.

Xtt aristotle							
File Edit Functions View Help							
 							
							
	Device	ErrorCount	Process	PlcThread	Address	Class	Description
	R1-Ai1	0	Plc	100ms	0	Ssab_AI32uP	
	R1-Di1	0	Plc	100ms	0	Ssab_DI32D	
	R1-Do1	0	Plc	100ms	0	Ssab_DO32DKS	
	UC11-M1	-	Plc	10msPn	-	ABB_ACS880_PnModule	
	UC12-M1	-	Plc	10msPn	-	ABB_ACS880_PnModule	
	KF11-M1	-	Plc	10msPn	-	PnModule	6ES7131-6BF01-0BA0
	KF11-M2	-	Plc	10msPn	-	PnModule	6ES7131-6BF01-0BA0
	KF11-M3	-	Plc	10msPn	-	PnModule	6ES7131-6BF01-0BA0
	KF11-M4	-	Plc	10msPn	-	PnModule	6ES7131-6BF01-0BA0
	KF11-M5	-	Plc	10msPn	-	PnModule	6ES7131-6BF01-0BA0
	KF11-M6	-	Plc	10msPn	-	PnModule	6ES7132-6BF01-0BA0
	KF11-M7	-	Plc	10msPn	-	PnModule	6ES7132-6BF01-0BA0
	KF11-M8	-	Plc	10msPn	-	PnModule	6ES7134-6JD00-0CA1
	KF11-M9	-	Plc	10msPn	-	PnModule	6ES7134-6JD00-0CA1

Fig Lista över I/O enheter

Bekrivning av kolumnerna

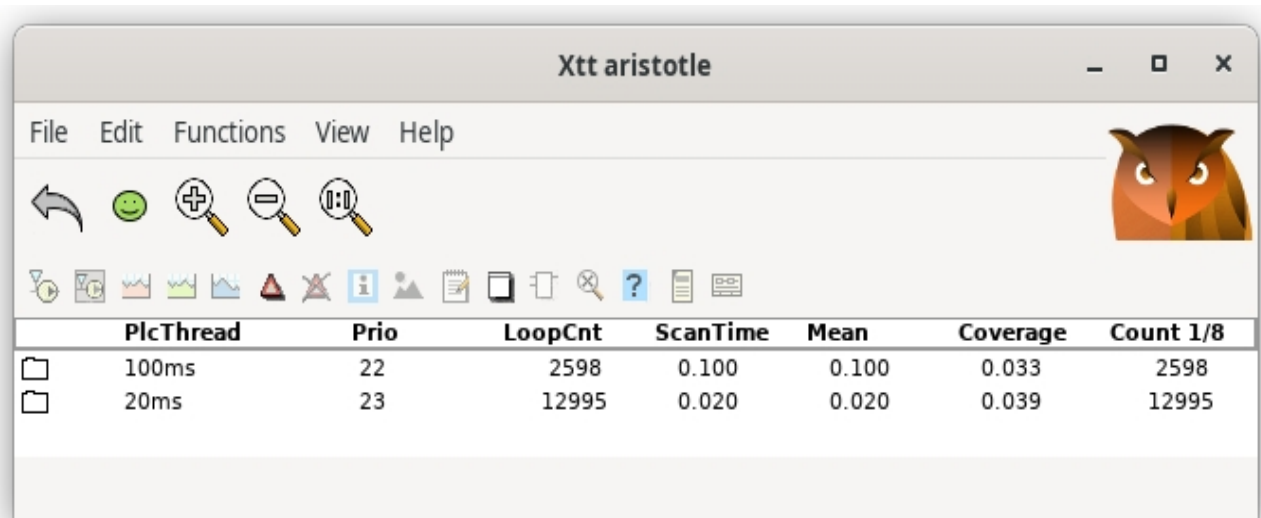
Device	Namn på kort-objektet. De två sista namnleden visas.
Class	Kortobjektets klass.
ErrorCount	Felräknare.
Process	Process som hanterar kortet. 1: plc, 2 io_comm.
PlcThread	Plctråd som hanterar kortet.
Address	Adress som är konfigurerad för kortet.

Genom att välja ut ett enhet och trycka Pil-Höger visas kanalerna för enheten (det går även att klicka på lövet).

8.8 System / PlcThread

Visar lista på trådar i plc-programmet.

Olika Plc program går med olika prioritet och med olika cykeltid. De är kopplade till PlcThread objekt som bestämmer prioritet och cykeltid för exekveringen.



The screenshot shows a window titled 'Xtt aristotle' with a menu bar (File, Edit, Functions, View, Help) and a toolbar with various icons. An owl icon is in the top right corner. Below the toolbar is a table with the following data:

	PlcThread	Prio	LoopCnt	ScanTime	Mean	Coverage	Count 1/8
📁	100ms	22	2598	0.100	0.100	0.033	2598
📁	20ms	23	12995	0.020	0.020	0.039	12995

Fig Lista på plc trådar

Beskrivning av kolumner

PlcThread	Namn på trådobjektet.
Prio	Trådens prioritet.
LoopCnt	Räknare för antal cykler sedan start av ProviewR.
ScanTime	Konfigurerad cykeltid.
Mean	Medelvärde på uppmätt cykeltid.
Coverage	Exekveringstiden för tråden i förhållande till cykeltiden i procent.
Count 1/8	Antal cykler där exekveringstiden är mindre än 1/8 av cykeltiden.
Count 1/4	Antal cykler där exekveringstiden är mindre än 1/4 av cykeltiden och större än 1/8.
Count 1/2	Antal cykler där exekveringstiden är mindre än 1/2 av cykeltiden och större än 1/4.
Count 1/1	Antal cykler där exekveringstiden är mindre än cykeltiden och större än 1/2 cykeltiden.

8.9 System / PlcPgm

Visar en lista på alla PlcPgm objekt.

Ett PlcPgm objekt omfattar plc-kod som styr en anläggningsdel, eller utför andra typer av beräkningar. PlcPgm objektet är kopplade till ett tråd objekt som bestämmer prioritet och cykeltid.

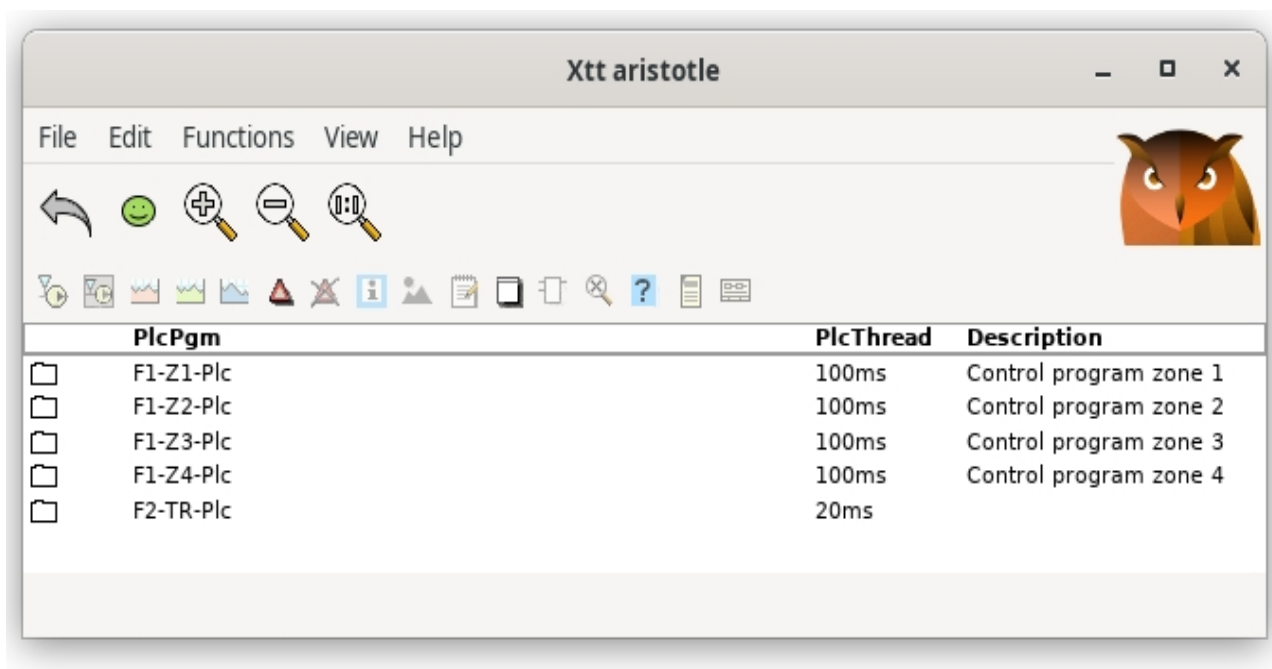


Fig Lista på PlcPgm objekt

Beskrivning av kolumner

PlcPgm	Namn på PlcPgm objektet.
PlcThread	Plc tråd som objektet är kopplat till.
Description	Beskrivning.

PlcTrace för ett PlcPgm öppnas genom att välja ut PlcPgm objektet och trycka på Ctrl+L.
Med Pil-Höger visas all fönster i Plc-programmet.

8.10 System / Logging

Med loggnings-funktionen kan man logga signaler och värden i databasen på en text-fil. Innehållet i textfilen kan visas i ett kurvfönster, eller läsas in i ett kalkyl-program, t ex Excel or Calc.

Upp till 100 attribut kan loggas i en loggfil, och det finns 10 st entry som var och en hanterar en loggfil.

Att starta en loggning går till på följande sätt.

- Samla ihop de attribut som ska loggas i samlingsbilden, t ex genom att leta upp attributen i databasen och trycka på Ctrl+V.
- Kontrollera med Ctrl+N att rätt attribut finns i samlingsbilden.

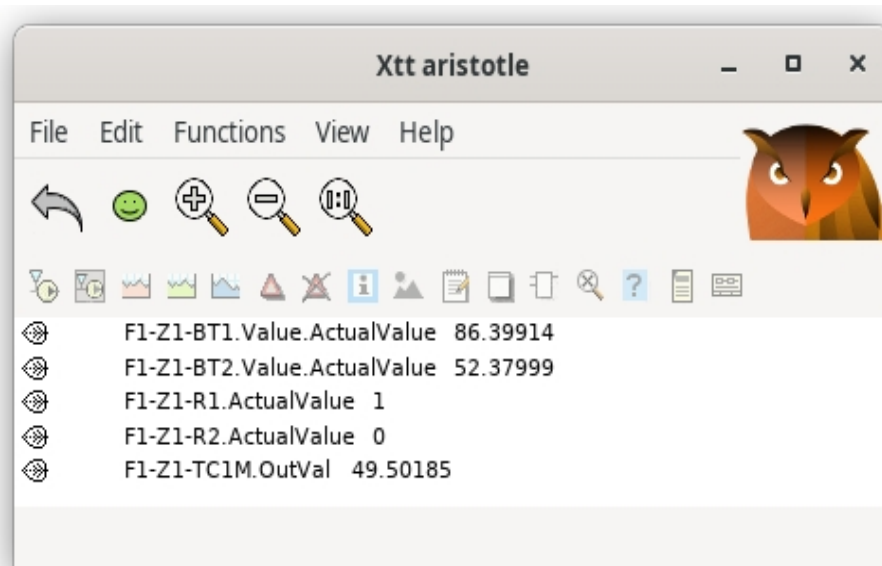


Fig Attribut insamlade i samlingsbilden

- Öppna Logging entry 1 (System / Logging / Logging Entry 1).
- Klicka på 'Insert' knappen. Attributen överförs nu från samlingsbilden till listan på attribut i loggentry bilden.
- Ange loggtid i ms, och ev annat filnamn.
- Starta loggningen genom att klicka på 'Start' knappen.

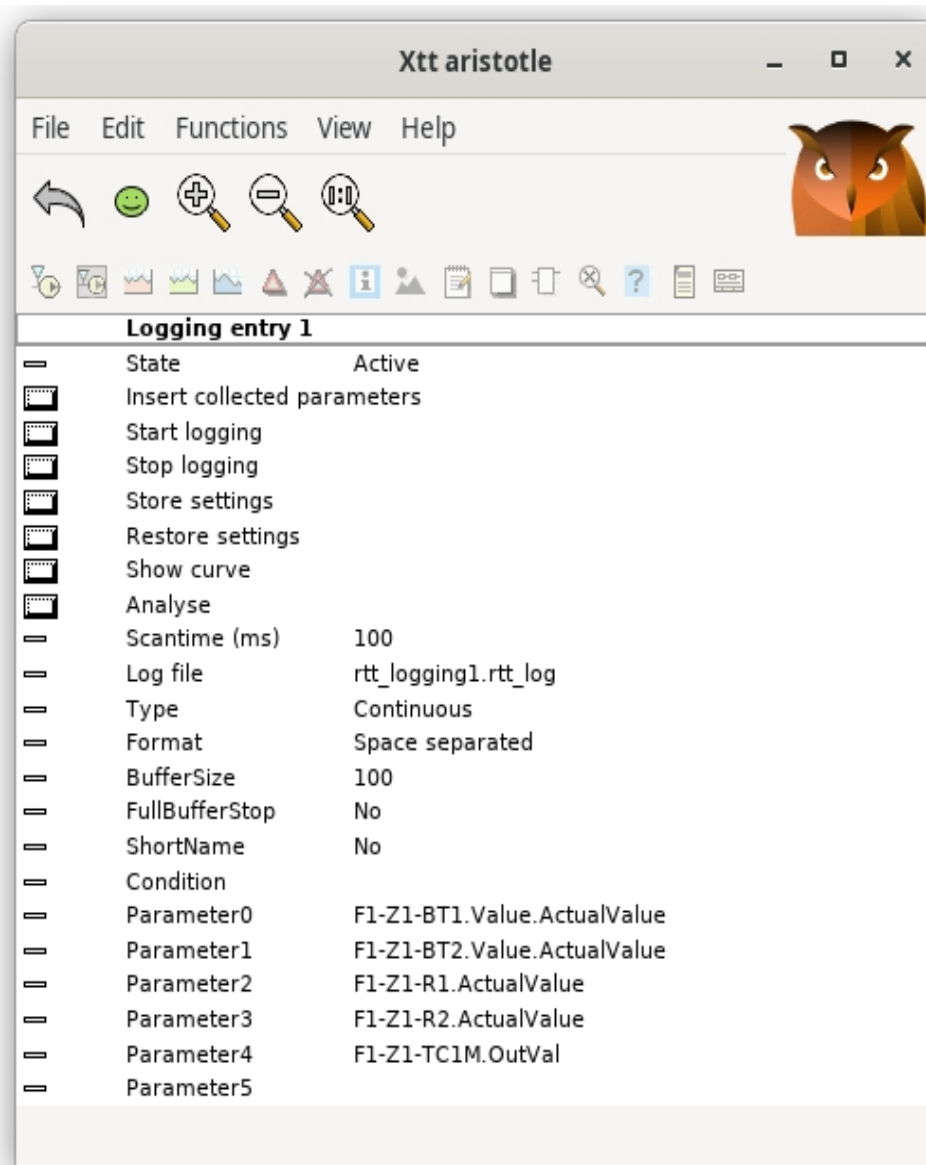


Fig Loggentry med loggningen started

- När loggningen är klar klickar man på 'Stop' knappen. Observera att man inte får terminera navigatoren under tid loggningen pågår. Däremot kan man lämna loggentry-sidan.
- Titta på loggresultatet i kurvform genom att aktivera 'ShowFile'.

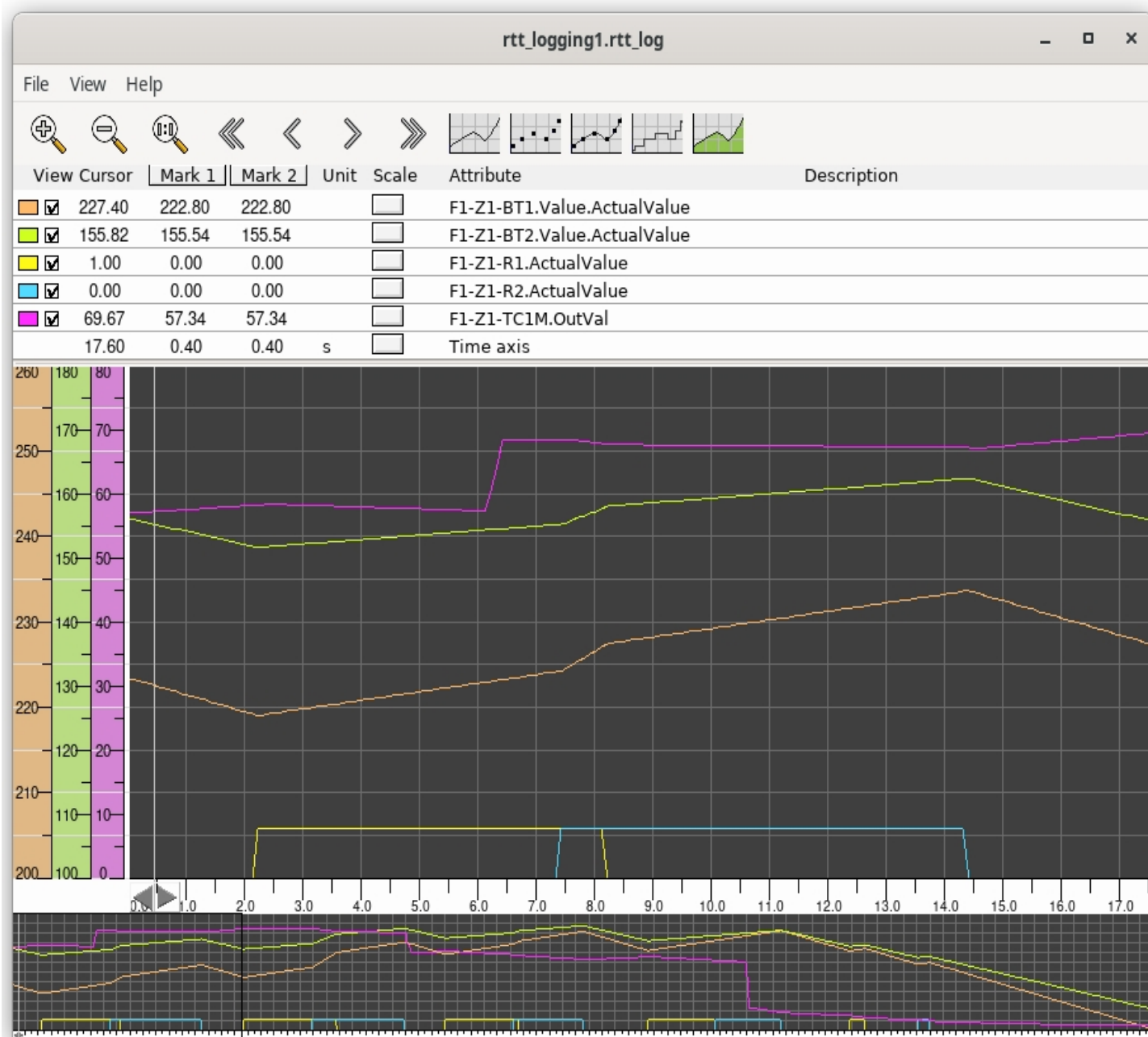


Fig Loggningen i kurvform

Loggningen kan vara av två typer, antingen loggas värdet av attributen kontinuerligt med en viss frekvens (kontinuerlig loggning), eller loggas ett attribut varje gång dess värde ändras (händelsestyrd loggning).

Kontinuerlig loggning

Värdet av attributen i entryt loggas vid varje cykel. På filen skrivs dessutom tiden sedan loggningen startades. Filen kan visas i Xtt's kurvfönster med 'ShowFile' eller läsas in i ett kalkylprogram.

Händelsestyrd logging

Ett attribut i entryt loggas om värdet av attributet har ändrats. I loggfilen skrivs tiden för ändringen och det nya värdet. Denna typ av loggning kan inte

visas med 'ShowFile'.

Buffer

Loggningen sker först internt i en buffer. När bufferten är full töms den på fil. Vid snabba tidskritiska förlopp kan det vara lämpligt att öka bufferstorleken för att undvika att störningar som kan uppstå när bufferten skrivs på fil.

Villkorlig loggning

I 'Condition kan man ange ett digitalt attribut som styr loggningen. Loggning sker enbart om Condition attributet är sant.

Man kan även ange ett uttryck i 'Condition. Uttrycket startar med 'expr' följt av villkoret omgivet av parenteser. I villkoret kan man hämta upp attribut-värden med GetA funktionen för analoga attribut, GetD funktionen för digitala attribut och GetI funktionen för heltals attribut.

Exempel

```
expr( GetD("H1-H2-Dv1.ActualValue") && GetA("H1-H2-Av2.ActualValue") > 2.5)
```

Loggningen utförs när Dv'n 'H1-H2-Dv1' är 1 och Av'n 'H1-H2-Av2' är större en 2.5.

Lagra

Logg konfigurationen kan sparas med 'Store' knappen, för att återskapas vid ett senare tillfälle med 'Restore' knappen.

8.11 System / System Messages

Proivew's systemlogg innehåller informations-, varnings- och felmeddelande som skrivs ut av systemets olika serverprocesser. Även applikationer kan skriva på systemloggen genom att använda errh gränssnittet.

Meddelandena i loggfilen har olika grad av allvar, som makeras med en bokstav och färg:

F	rött	Fatal, är den allvarligaste formen av felmeddelande.
E	rött	Error, är ett felmeddelande.
W	gult	Warning, är ett varningsmeddelande.
I	grönt	Information, är ett informations meddelande.
S	grönt	Success, är ett meddelande om en framgångsrikt utförd aktivitet.

Dessutom markeras uppstart av ProivewR med en gulfärgad rad.

Framför allt ska man vara uppmärksam på röda meddelanden som innebär att någonting är fel i systemet.



System status visar en lista på noder i systemet, och genom att öppna en nod visas statusbilden för noden.

Det mest intressanta in bilden är SystemStatus. Detta är en indikation på hur systemet mår.

Om SystemStatus är grön, mår systemet bra, om den är gul eller röd, är någonting fel i systemet.

Färgmarkeringarna har samma betydelse som i Systemloggen ovan, dvs

F	blinkande rött	Fatal, är den allvarligaste formen av felstatus.
E	rött	Error, markerar felstatus.
W	gult	Warning, markerar varnings-status.
I	grönt	Information, är ett informations status.
S	grönt	Success, innebär all allt är ok.
	grått	Null status, funktionen är inte aktiverad.

Status för systemet är en summastatus av status för system- och applikations-processer. Under systemstatus finns en lista på dessa processer, och här visas även status för dem. Dessutom visas processernas senaste eller allvarligaste meddelande i systemloggen.

Om systemstatus t ex är gul, innebär det att någon system eller applikations process har gul status. Förhoppningsvis kan man genom att läsa status texten och titta på systemloggen lista ut vad felet är.

rt_tmon	Timer monitorn	Hanterar sändning av prenumerationsdata.
rt_emon	Event monitorn	Hanterar larm och händelser.
rt_bck	Backup process	Hantera backup av attribut och objekt.
rt_linksup	Link supervision	Övervakning av länkar till andra noder.
rt_trend	Trend process	Hanterar lagring av trendkurvor.
rt_fast	Fast process	Hanterar lagring av snabbkurvor.
rt_elog	Event logg	Hanterar logging av händelser.
rt_webmon	Web monitorn	Förser webgränssnittet med databasinformation.
rt_webmonmh	Web mh monitor	Förser webgränssnittet med larm och händelser.
rt_sysmon	System monitor	Övervakar systemet.
plc	Plc processen	Hanterar I/O och exekverar plc-koden.
rs_remote	Remote processen	Hanterar remote kommunikation.
opc_server	Opc server process.	
rt_statusrv	Status server	Webservice för runtime monitorn och övervakningscentralen.
rt_post	Post server	Sänder larm via email eller sms.
rt_report	Report server	Genererar rapporter.
rt_sevhistmon	Storage monitor	Samlar och sänder historisk data till lagringsserver.
rt_powerlink	Ethernet powerlink serve	Hanterar Powerlink kommunikation.

8.13 System / Volumes

Visar laddade volymer.

För varje volym visas namn, klass, version och en bokstavskombination LCM där L står för loaded, C för cached och M för mounted.

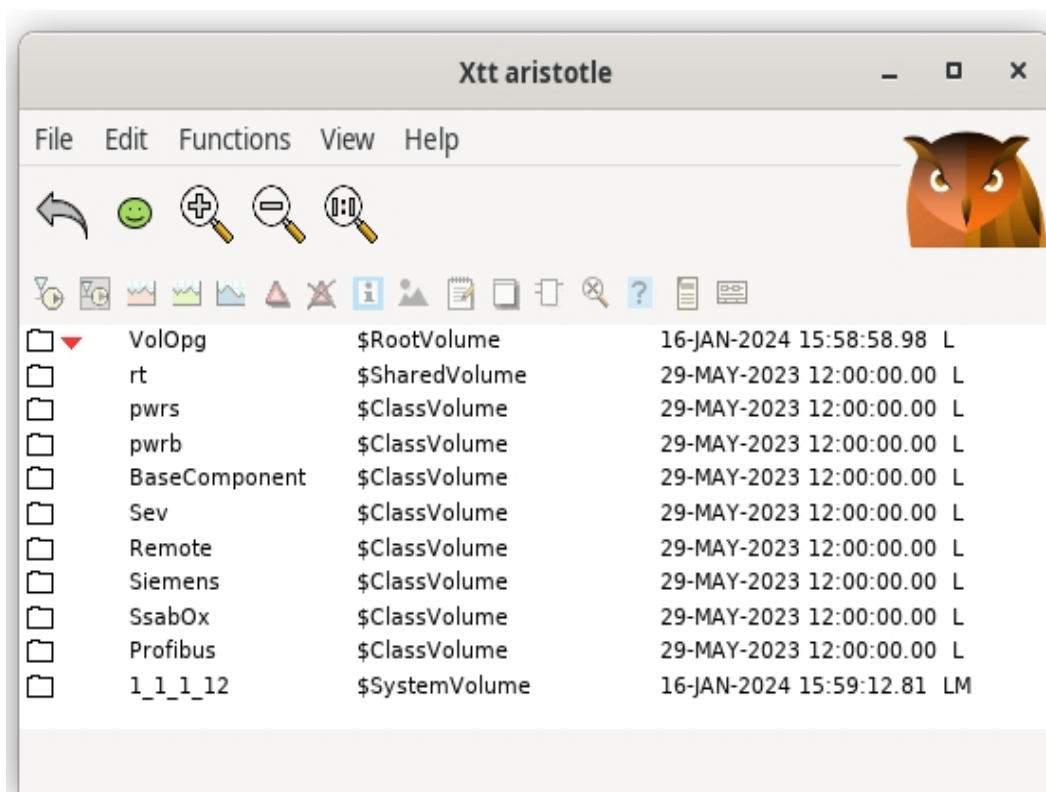
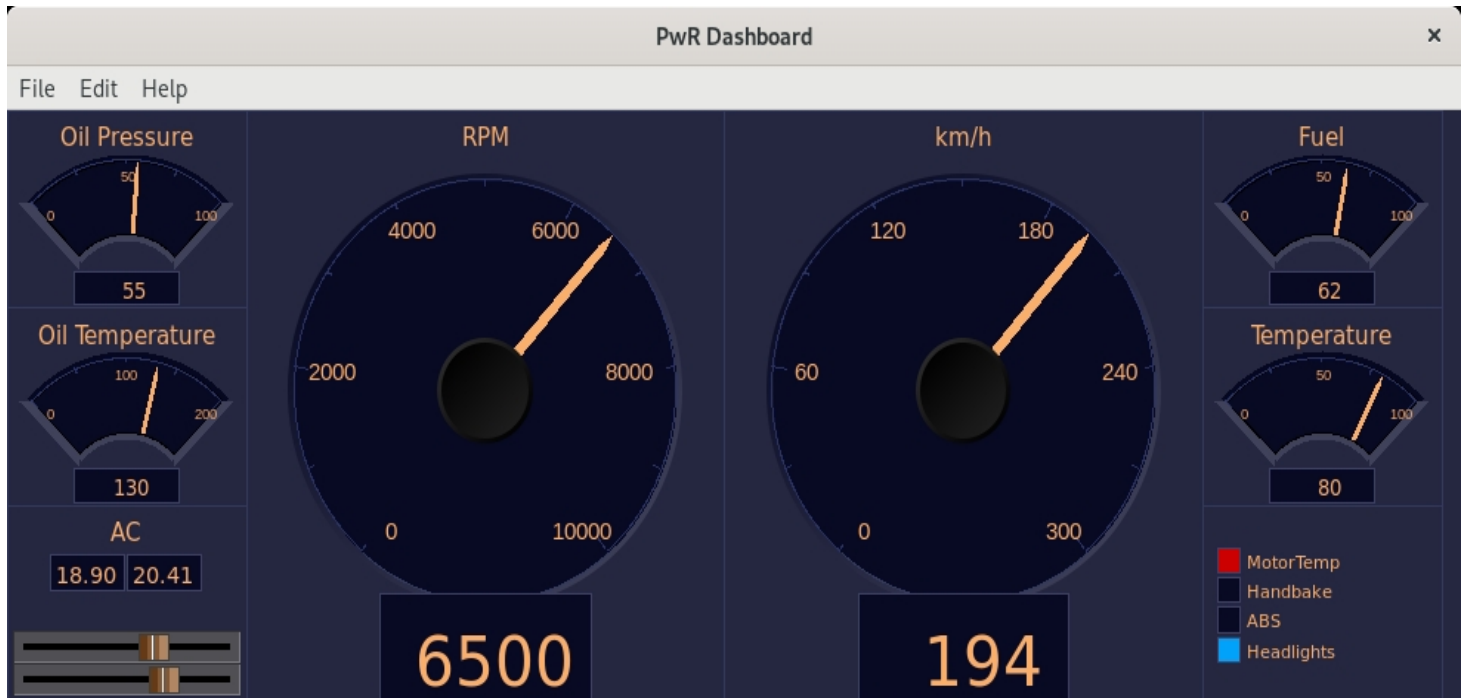


Fig Laddade volymer

9 Dashboard

Ett dashboard (instrumentbräda) är en förenklad graf med ett mycket begränsat antal byggstenar. Byggstenarna kallas för celler och kan visa värden i form av indikatorer, staplar, trendkurvor, givare etc.



Ett dashboard kan skapas in Ge editorn eller i operatörmiljön.

Dashboardet är indelat i rader och kolumner där cellerna placeras. En cell har initialt höjden av en rad och bredden av en kolumn, men storleken kan utökas till flera rader och kolumner.

En cell kan innehålla fler element. En stapel cell till exempel, kan innehålla tre element. Varje element kopplas till ett analog attribut som visas med en stapel. En trendcell kan innehålla två element och visa två kurvor.

Det finns tre huvudtyper av celler, analoga, digital och objekt.

Analoga celler

Bar

En stapelcell visar ett analog värde med en stapel och ett värdefält. Det kan innehålla tre element.

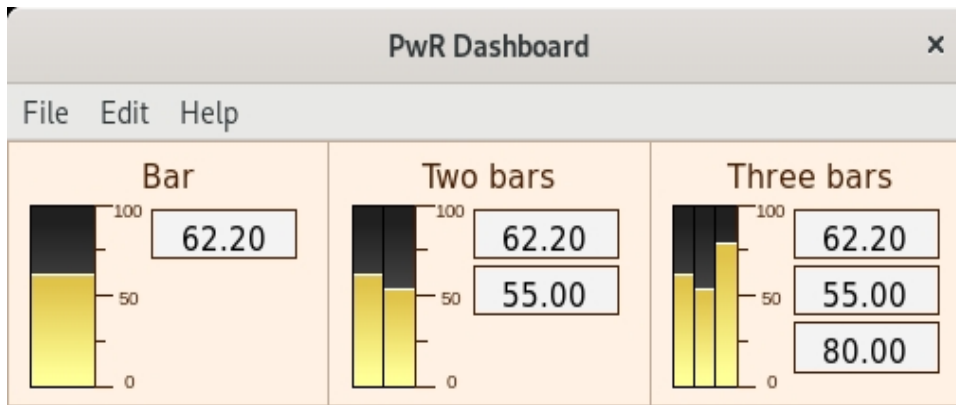
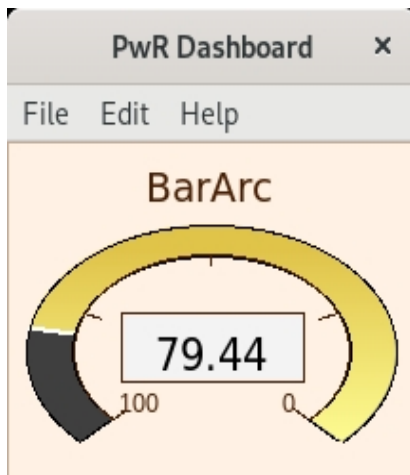
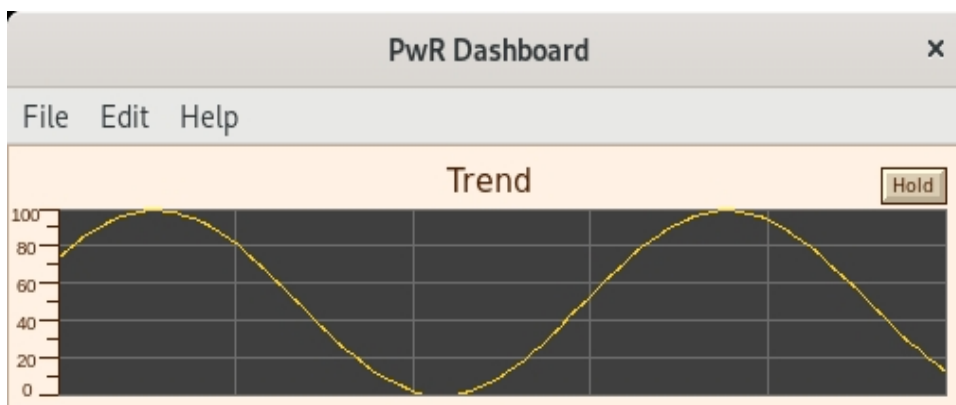


Fig Stapelceller med ett, två och tre element

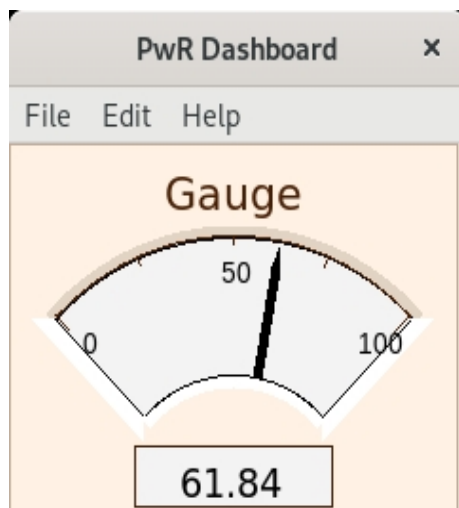
BarArc



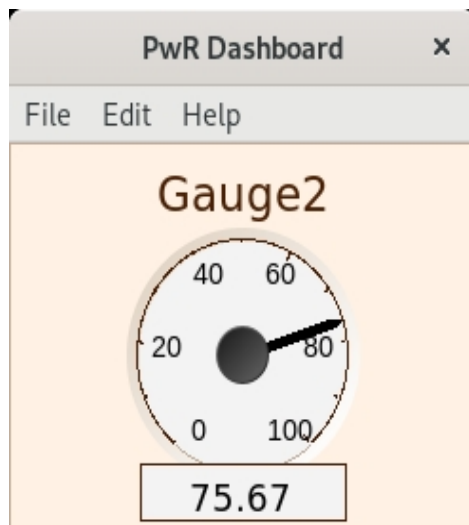
Trend



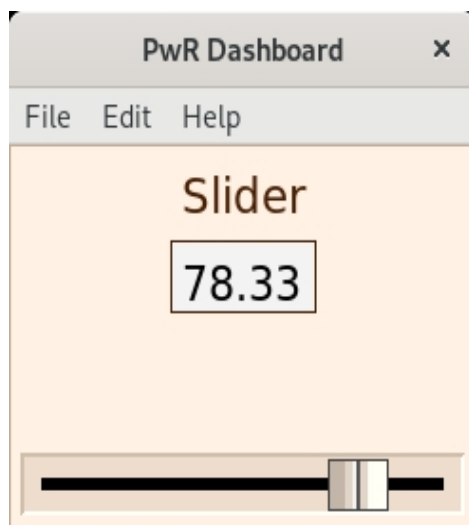
Gauge



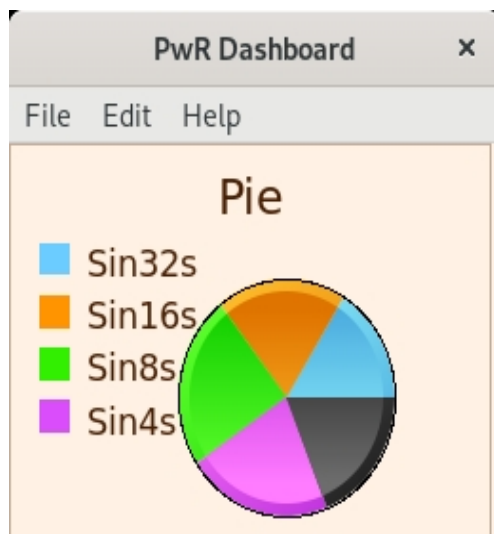
Gauge2



Slider

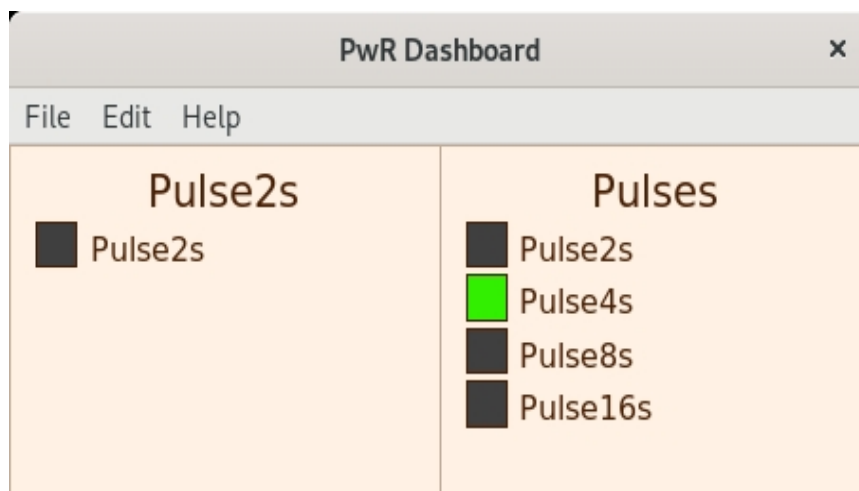


Pie

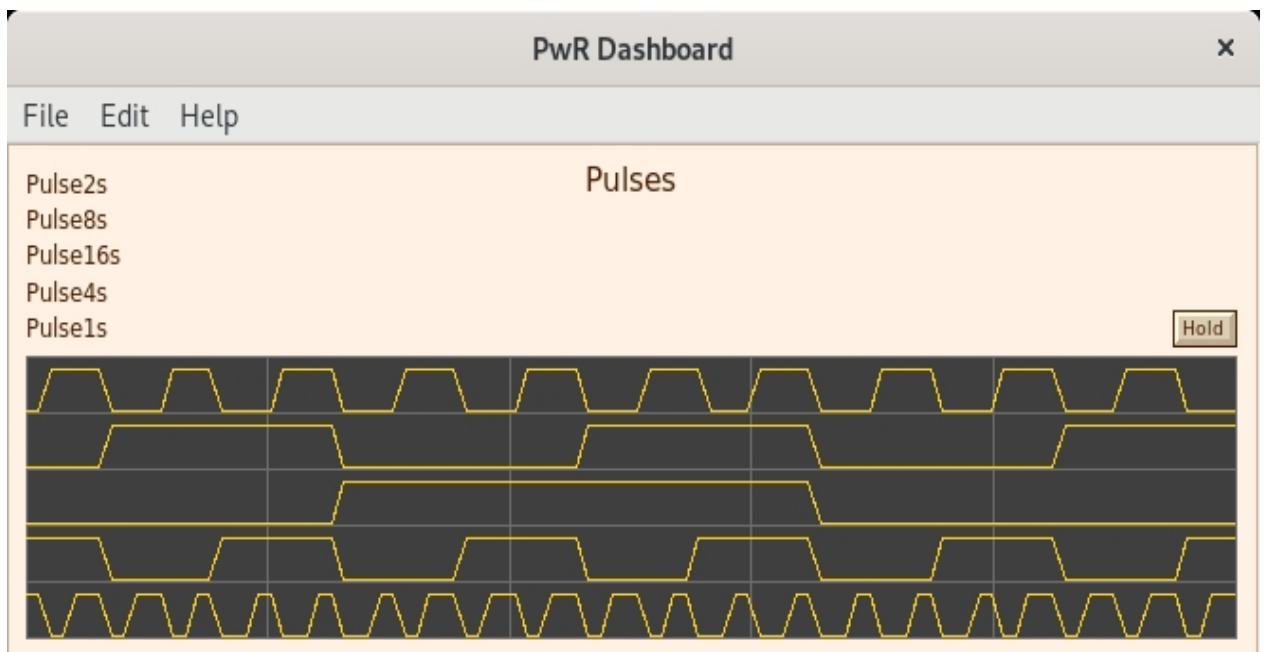


Digitala celler

Indicator



DigitalTrend



Objektceller

Objektceller finns för några objekt, t ex DsTrendCurve.

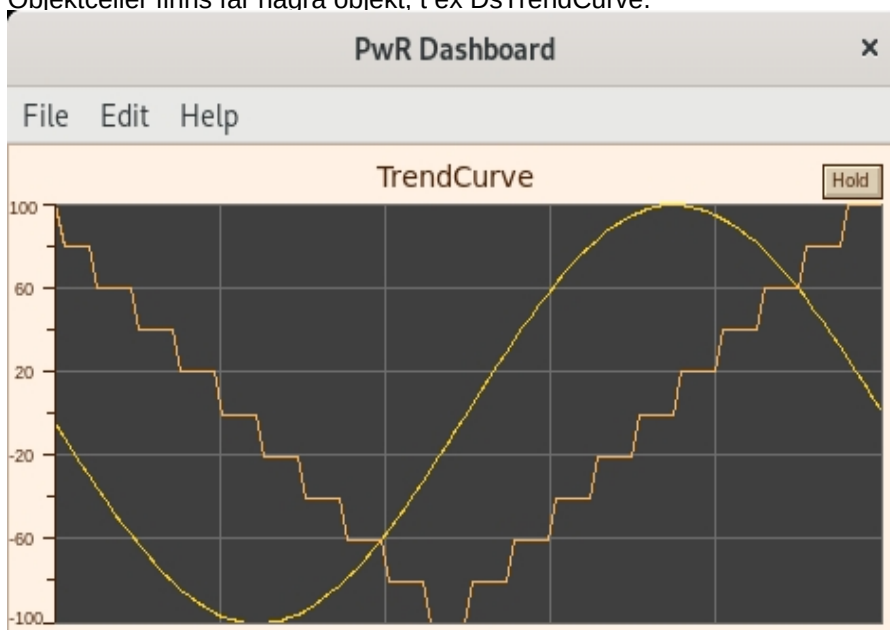


Fig Objektcell för DsTrendCurve

Skapa ett dashboard i rt_xtt

Genom att välja ut ett attribut in i navigatorn, och aktivera Funktioner/Dashboard/Lägg till (Ctrl+D) i meny, adderas attributet till ett dashboard. Om något dashboard befinner sig i editering-mod lägg det in där, annars läggs det in i "PwR Dashboard". Man kan addera celler tills alla rader och kolumner är fyllda.

För ett analog attribute skapas en stapelcell. Cell-typen kan ändras genom att gå över i

editiering-mod från File/Edit (Ctrl+E) in menyn. Öppna cellens attribute (dubbelklick) och välj en annan typ i Dash.Type. Lämna editerings-mod med Ctrl+E.

I editerings-mod, öppnas Ge editorn in begränsad mod. Några funktioner i editerings-mod är

- Addera. Skapa en tom cell.
- Ta bort. Ta bort en cell.
- Kopiera. Kopiera celler.
- Klistra in. Klistra in kopierade celler.
- Koppla. Koppla utvalt attribut i navigatorn.
- Slå ihop. Slå ihop celler. Utvalda celler slås ihop i den cell som var först utvalt.
- Från GraphAttribut kan cykeltiden och dashboardets storlek ändras.

10 Plc trace

Plc trace används för att studera och felsöka plc-koden.

Trace öppnas med metoden 'Open Plc', eller via en korsreferens i plc-koden.

Man kan även öppna genom att välja ut ett PlcPgm objekt in navigatorn och aktivera 'Functions/OpenProgram' (Ctrl+L) i menyn.

Till höger finns ett navigationsfönster, där det visade området är markerat med en fyrkant. Genom att flytta fyrkanten genom att dra med MB1 flyttar man det visade området. Man kan även zooma genom att dra med mittenknappen.

Digitala objekt med hög status markeras med rött i koden.

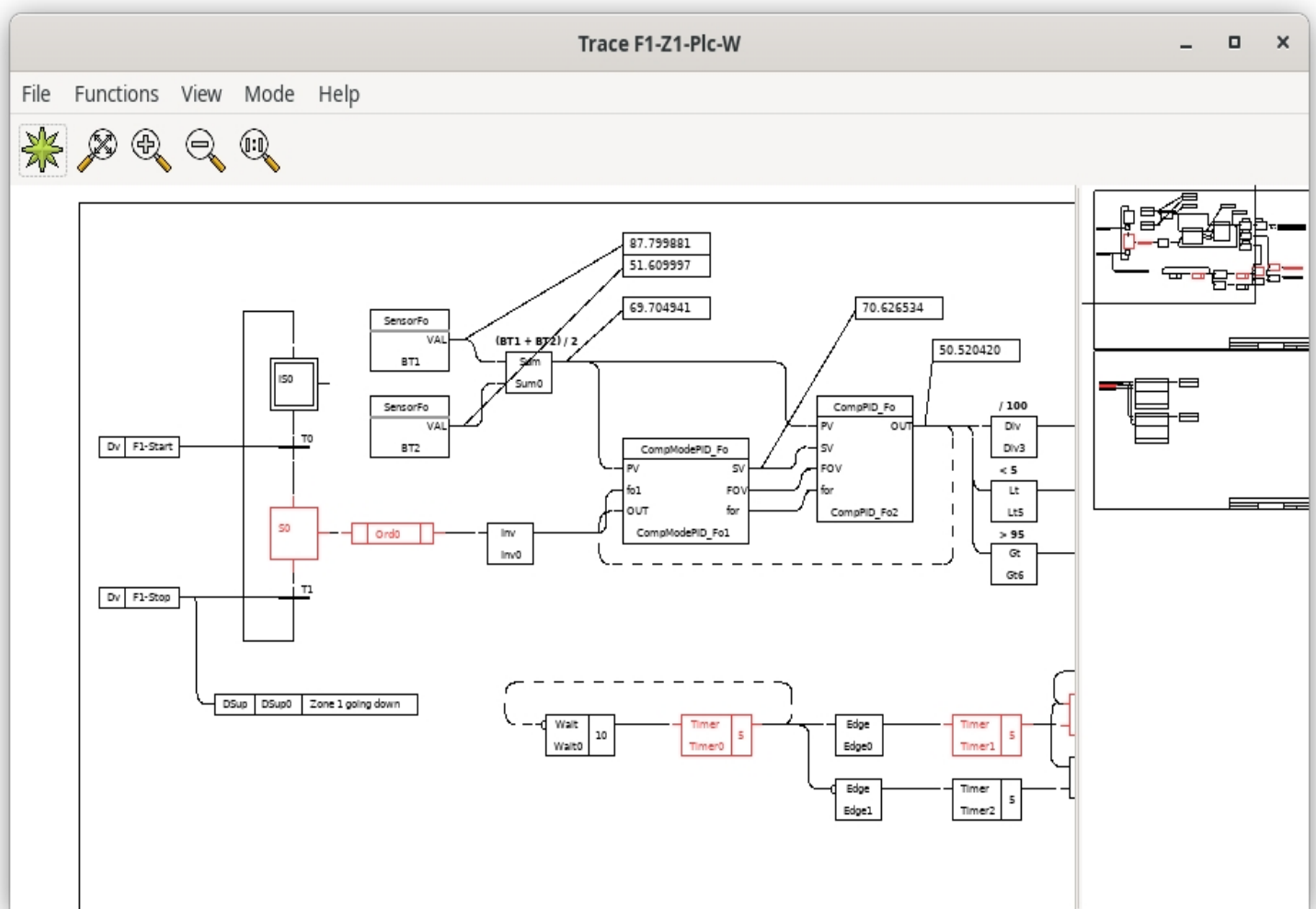


Fig Plc Trace

Man kan titta på värdet på analoga och digitala utgångar genom att skapa sk analys-noder som visar värdet på utgången. Dessa skapas genom att dra med mittenknappen från utgången.

En uppsättning analysnoder kan sparas och återskapas med 'File/Save Trace' och 'File/Restore Trace' i menyn.

Genom att högerklicka på ett objekt visas metoderna för objektet. En mycket användbar metod i det här sammanhanget är korsreferenser. Korsreferenserna visas även om man dubbelklickar på ett objekt.

Korsreferenser

Korsreferenserna är en lista på var ett objekt eller attribut förekommer i koden och i processbilder. Om referensen innebär en läsning av värdet, markeras detta med en bok, om det innebär en skrivning av värdet, markeras det med en penna. Läsning kan vara att värdet visas i en bild eller hämtas upp i plc-koden med ett Get objekt i koden. Skrivning kan ske med en tryckknapp i en bild eller med ett Set eller Sto objekt i koden.

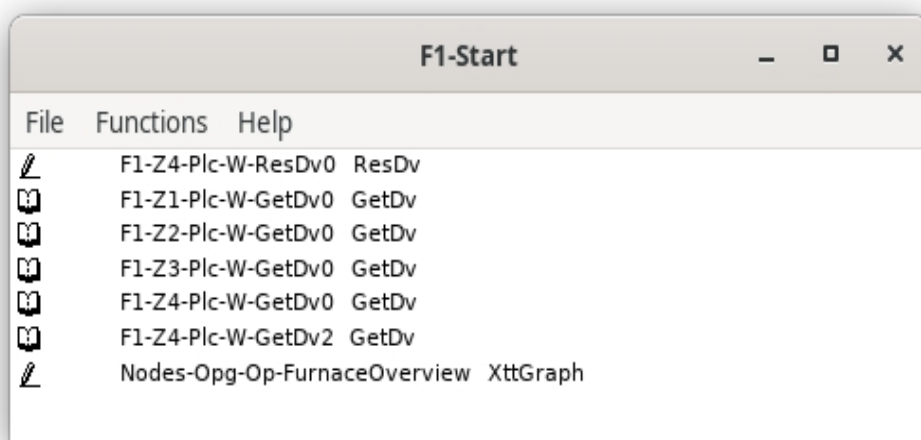


Fig Korsreferenser

Genom att högerklicka på korsreferensen kan man öppna plc't eller process-grafen som referensen pekar på. På plc referenser kan man även dubbelklicka. När Plc Trace öppnas är referensen markerad och centrerad.

Felsökning med Plc Trace och Korsreferenser

Plc trace och korsreferenser är ett mycket effektivt sätt att felsöka på. En felsökning utgår ofta från ett larm eller en felindikering i en process bild. Genom att aktivera metoderna 'Crossreferences' eller 'Open Plc' letar man upp det eller de ställen i koden där objektet skrivs. Genom att analysera koden sluter man sig till vilken signal som saknas, tar upp korsreferenserna för denna och öppnar Plc Trace där denna skrivs. På detta sätt kan man nästla sig bakåt i koden för att slutligen hitta orsaken.

Simulering

Vid simulering, dvs när man testkör ett system utan I/O system, kan man sätta över Plc Trace i simuleringsmod, med Mode/Simulate i menyn. I simuleringsmod kan man togglar digitala signaler genom att klicka med Ctrl/Shift MB1 på dem i Plc Trace fönstret.

11 Setup-skript

En setup-fil kan skapas på hemma-katalogen med namnet

```
xtt_setup.rtt_com
```

Den kommer att exekveras när operatörmiljön (eller rt_xtt) startas, och kan innehålla script satser och xtt kommandon.

Här är några exempel på användbara kommandon.

Öppna en graf

```
open graph my_graph
```

Skapa en underhålls meny i navigatören

```
create item/text="Maintenance"/menu/destination="DataBase"/before  
create item/text="My graph"/command="open graph my_graph"/pixmap=graph  
/dest=Maintenance/first
```

Ta bort menyalternativ ur navigatörens standardmeny

```
delete item /name=exit  
delete item /name=system-nethandler
```

Kortkommando för hieraki i databasen

```
define rb9 "show children /name=hql-rb9"
```

Kortkommando för en graf

```
define my_graph "open graph my_graph"
```

11.1 Symbol

En xtt symbol kan användas som kortkommando eller som variabel i ett kommando. Om symbolen används som variabel i ett kommando ska den omgärdas av apostrofer.

Symboler skapas med define kommandot.

Exempel på en symbol använd som ett kortkommando

```
xtt> define p1 "show child/name=hql-hvk-pumpar-pump1"  
xtt> p1
```

Exempel på en symbol använd som en variabel

```
xtt> define p1 hql-hvk-pumpar-StartPump1  
xtt> open trace 'p1'
```

12 Användare och privilegier

ProviewR innehåller ett system med användare som tilldelas privilegier. För att utföra vissa saker krävs privilegier, och endast om användaren har dessa privilegier tillåts han att utföra dem.

Följande privilegier används i operatörmiljön:

RtRead	Läs rättigheter i runtime.
RtWrite	Skriv rättigheter i runtime. Gör att man kan ändra värden på attribut från navigatören.
RtEvents	Privilegium för att hantera larm och händelser.
System	System. Ger behörighet till det mesta.
Maintenance	Privilegium för underhållspersonal.
Process	Privilegium för process tekniker.
Instrument	Privilegium för instrument tekniker.
Operator1	Privilegium för operatör.
Operator2	Privilegium för operatör.
Operator3	Privilegium för operatör.
Operator4	Privilegium för operatör.
Operator5	Privilegium för operatör.
Operator6	Privilegium för operatör.
Operator7	Privilegium för operatör.
Operator8	Privilegium för operatör.
Operator9	Privilegium för operatör.
Operator10	Privilegium för operatör.

Användare

När operatörmiljön eller Xtt startas, loggar man antingen in som en användare, eller så tilldelas man default privilegier som definieras i Security objektet.

Om uppstarten av operatörmiljön sker med ett operatörsplats-objekt hämtas användare från User objektets UserName attribut. Om användare saknas eller är ogiltig öppnas inloggings-fönstret och man måste logga in med användarnamn och passerord.

Om Xtt startas utan OpPlace objekt, hämtas privileger från Security-objektets DefaultXttPriv attribut. Om där inte finns några privilegier som ger läs rättigheter, öppnas inloggings-fönstret.

Navigatören

För att göra förändringar i databasen från navigatören krävs RtWrite eller System. Detta gäller även kommandot 'set parameter'.

Plc Trace

För att kunna göra ändringar från Plc Trace i simuleringsmod krävs RtWrite eller System.

Objektsbilder

Generellt för objektsbilder krävs det RtWrite eller System för att göra ändringar i databasen från dem. Det finns några undantag. I ChanAi och ChanAo har även Instrument behörighet att ändra signalområden, och att sätta en Ao i test-mod. I PID och Mode-bilderna

har Process behörighet att ändra förutom RtWrite och System. I Mode-bilden kan man dessutom styra behörigheten genom att öppna bilden med kommandot 'open graph' och använda /access funktionen.

Processbilder

I Processbilder väljer konstruktören för varje inmatningsfält och tryckknapp vilka privilegier som krävs för att göra en ändring.

13 Funktionstangenter

Man kan använda funktionstangenterna för att snabbt ta upp bilder eller påverka objekt i databasen.

Funktionsanrop

Följande funktioner kan anropas när en funktionstangen trycks ner.

SetDig()

Sätter ett attribut av typen Boolean. Namnet på attributet ges som argument.

ResDig()

Återställer ett attribut av typen Boolean. Namnet på attributet ges som argument.

ToggleDig()

Togglar värdet på attribut av typen Boolean. Namnet på attributet ges som argument.

Command()

Exekverar ett xtt-kommando. Kommandot ges som argument.

Setup-fil

Kopplingen mellan tangenter och funktionerna läggs i filen Rt_xtt på inloggningskatalogen.

I filen skrivs en rad för varje tangent. På raden skrivs

- eventuella modifierare (Shift, Control eller Alt).
- <Key> följt av namnet på tangenten, t ex <Key>F7 eller <Key>m.
- ett kolon följt av funktionen som ska anropas med argument.

Ett '#' tecken först på raden anger en kommentar.

Exempel

```
#  
# Configuration of Global function keys  
  <Key>F5: Command(event ack /prio=A)  
  <Key>F6: Command(event ack /prio=NOA)  
  <Key>F7: Command(show alarm)  
  <Key>F8: Command(show event)  
  <Key>F9: Command(close all/except=navigator)  
Shift Control <Key>m: SetDig(F1-Start.ActualValue)  
Shift Control <Key>n: SetDig(F1-Stop.ActualValue)  
Shift Alt <Key>n: SetDig(F1-Reset.ActualValue)
```

14 Statusmonitorn

Statusmonitorn är ett verktyg för att övervaka och hantera process- och operatörs-stationer. Man adderar noder som man vill övervaka och statusmonitorn visar status för noden i form av en färgmarkering och status-texten.

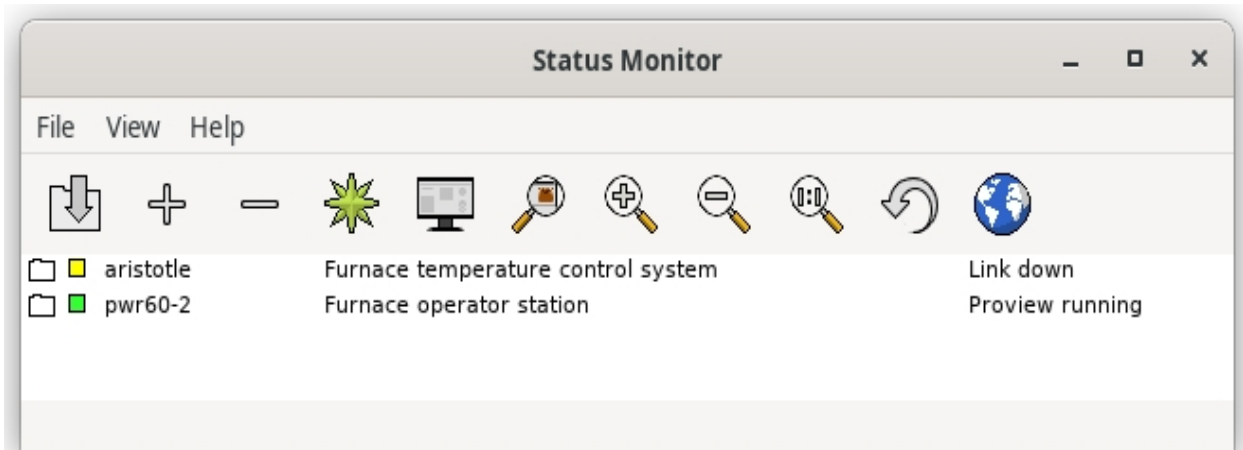


Fig Status monitorn

Kommandot för att starta statusmonitor är

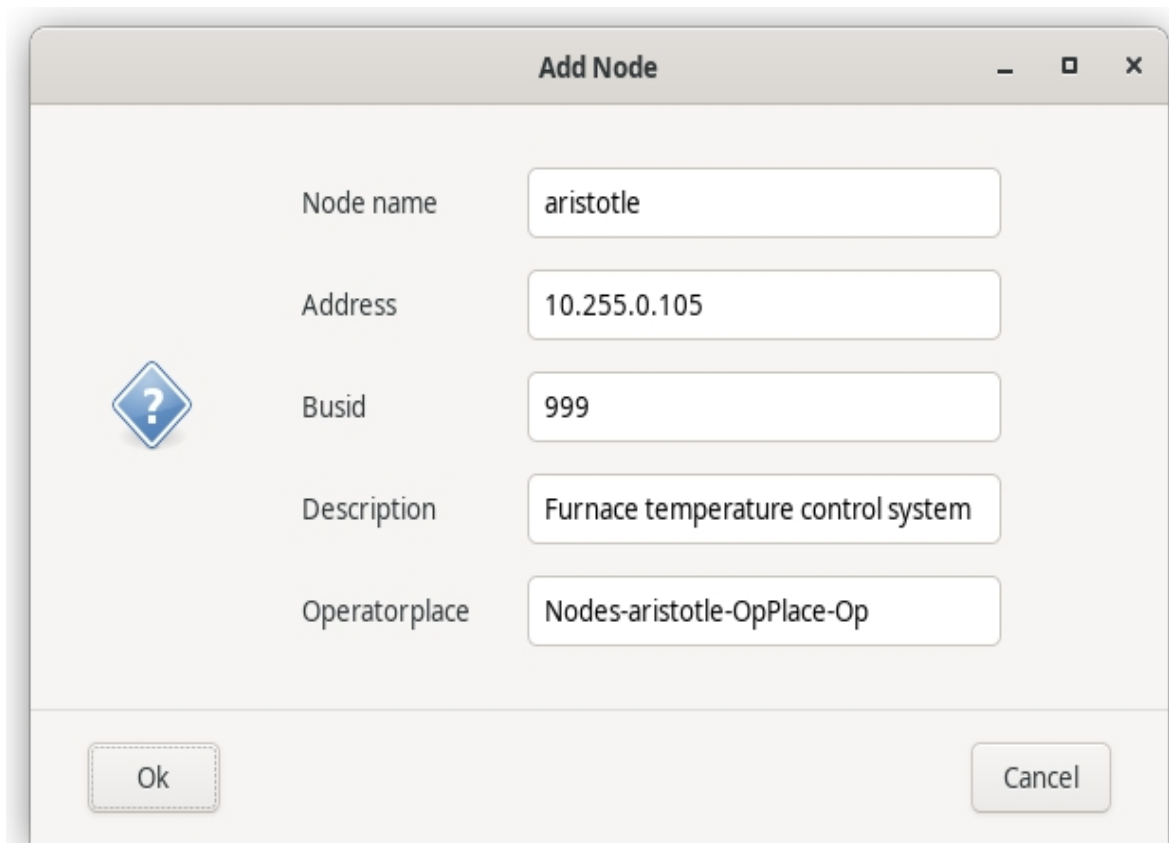
```
> rt_statusmon [-l 'language'][-c 'config-file'][-m 'status-index'][-e][-t][-o][-g]
```

- l Språk, en_us, sv_se, de_de eller fr_fr.
- m Visa en användardefinierad status istället för systemstatus.
- c Konfigurationsfil. Default \$HOME/rt_statusmon.dat.
- e Visa nod-beskrivning från konfigurationsfilen.
- t Dölj knappar för att starta runtime-monitorn i verktygspanelen och menyn.
- o Öppna kartbild vid start.
- g Koppla inte upp mot ProviewR runtime.

Addera och ta bort noder

En nod adderas till statusmonitorn från File/Add Node i meny, eller från plus-tecknet i verktygspanelen. En dialog öppnas där nodens namn anges. Man kan även ange en beskrivning om man vill ha en annan beskrivning än den i nodens node-objekt. Det är även möjligt att ange ett operatörsplats-objekt på noden, man kan då öppna denna operatörsplats från menyn eller verktygspanelen. OBS! Ange inte en operatörsplats som redan är i bruk på en operatörsplats. OpPlace objektet ska vara av typen 'Underhåll' eller dedicerad för statusmonitorn.

En nod tas bort från statusmonitorn genom att noden väljs ut, och 'File/Remove Node' aktiveras i menyn.



Node name	aristotle
Address	10.255.0.105
Busid	999
Description	Furnace temperature control system
Operatorplace	Nodes-aristotle-OpPlace-Op

Ok Cancel

Fig Addera en nod till monitorn

Status

Default är det systemstatus som övervakas. Systemstatus är en summastatus för för processerna i systemet, och berättar hur noden mår ur systemets synvinkel. I den här moden fungerar statusmonitorn som en övervakning för den systemansvarige.

Man kan även visa applikations-status, dvs en status som sätts av en applikation i systemet, och som t ex visar status för processen eller för en anläggningsdel. På det här sättet kan statusmonitorn fungera som en slags överordnad operatörsplats, där man med hjälp av felstatus indikerar att noden behöver operatörens uppmärksamhet, och operatören sedan kan gå vidare och öppna operatörsplatsen på noden, om situation kräver detta.

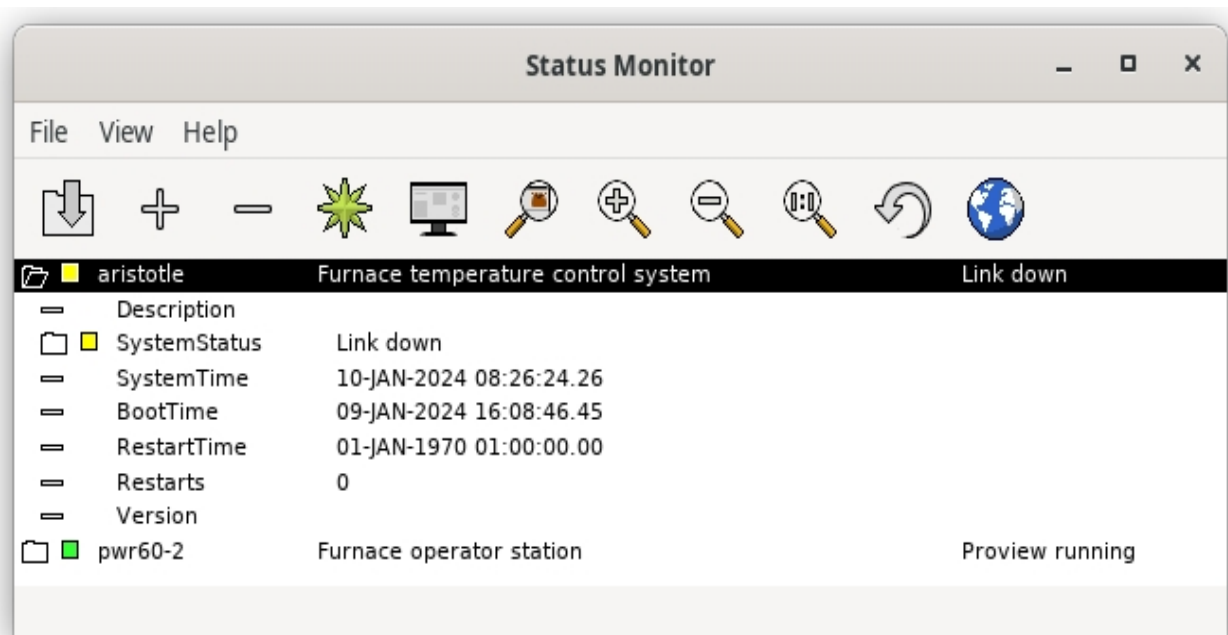


Fig Info om en nod

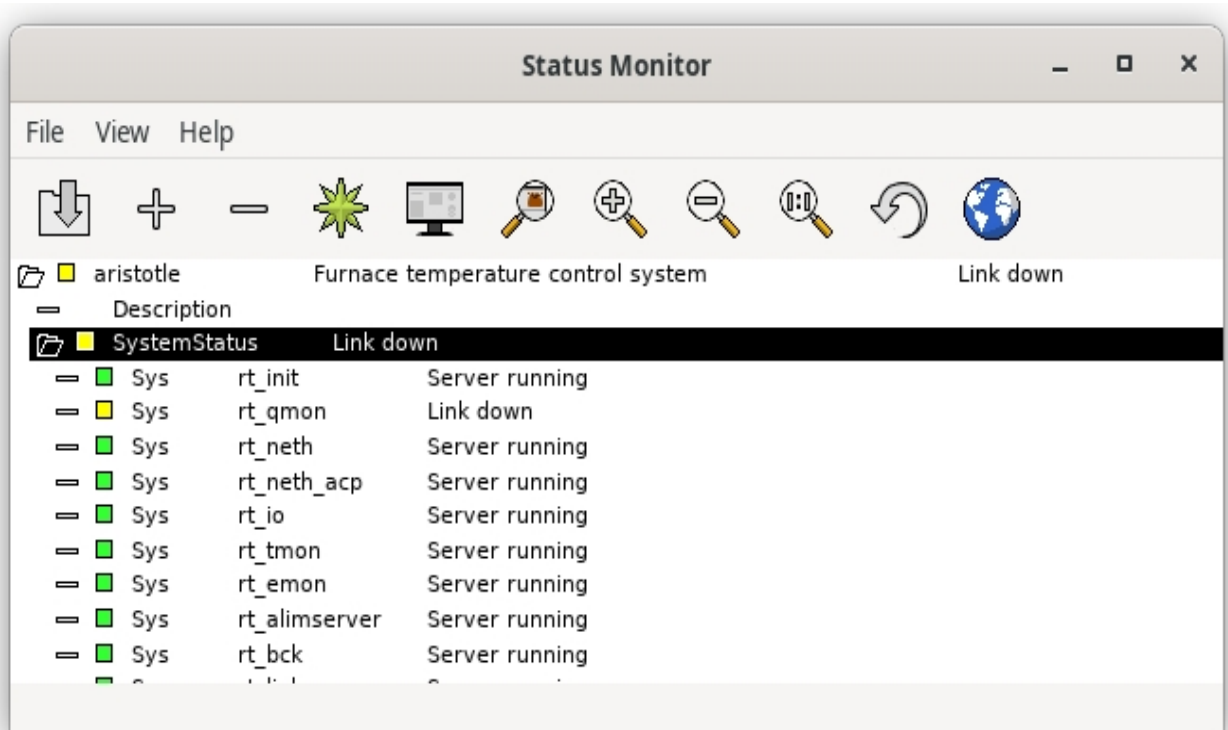


Fig Status for processer på noden

Om en kartbild är ritad över anläggningen, kan denna öppnas från jordgloben i vertygspanelen. Indikatorer kommer att visa status på varje nod, och en popup-menü kan konfigureras till att öppna operatörsplats och runtime-navigator.

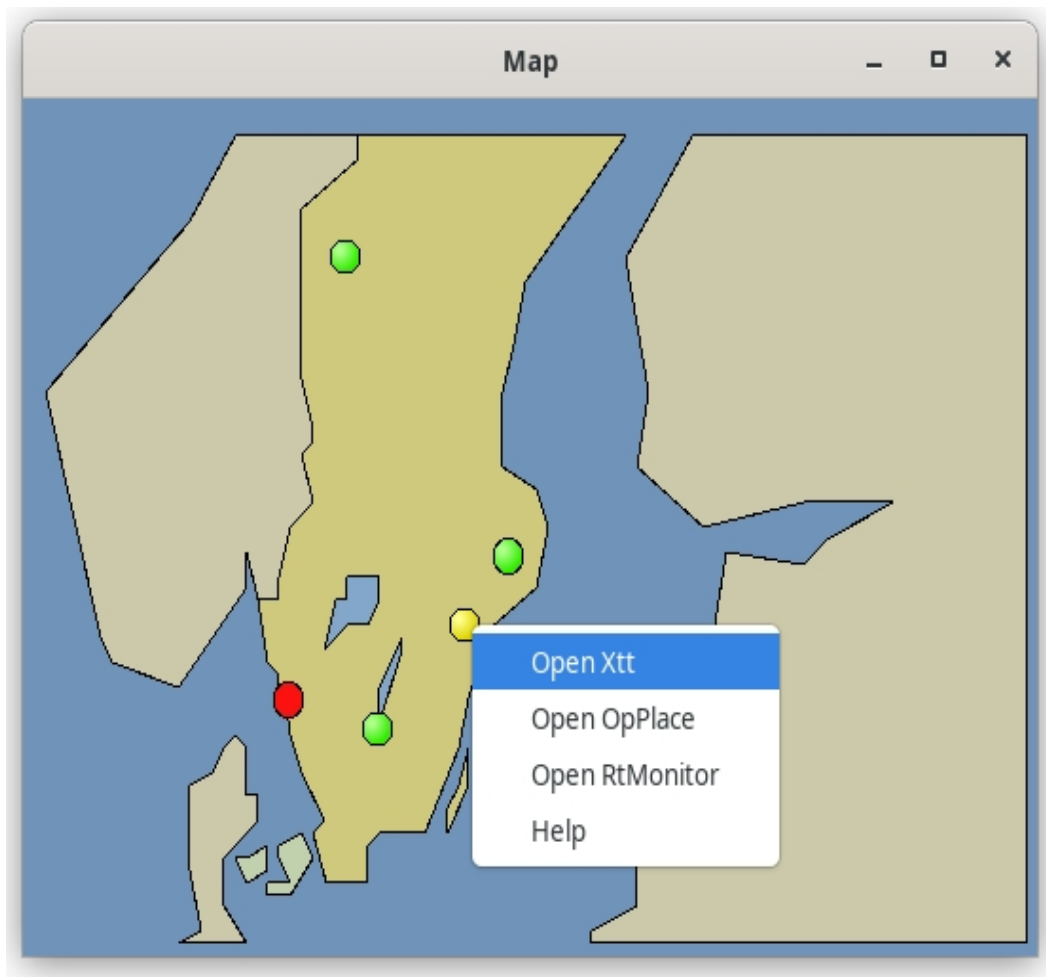


Fig Kartbild

Kommandon

Statusmonitorn har en kommando-rad som öppnas med Ctrl+B. Kommandona kan framför allt användas i popup-menyer i kart-grafen.

Följande kommandon är definierade

help

Visa hjälptexten för ett ämne.

```
help 'topic'
```

Se Xtt's hjälp-kommand för mer information.

select node

Välj ut en nod i nodlistan.

```
select node /node=
```

```
/node          Nodnamn.
```

open graph

Öppna en Ge graf.

```
open graph 'graph'
```

open opplace

Öppna operatörsplats för specificerad nod.

```
open opplace /node=
```

```
/node          Nodnamn .
```

open xtt

Öppna runtimenavigator för specificerad nod.

```
open xtt /node=
```

```
/node          Nodnamn .
```

open rtmon

Öppna runtime montor för specificerad nod.

```
open rtmon /node=
```

```
/node          Nodnamn .
```

15 Xtt kommandon

add parameter	Addera ett attribut till en bild
add menu	Addera ett menyalternativ till en xtt meny
collect	Addera ett attribut till samlingsbilden
call method	Anropa en xtt metod för ett objekt
check method	Kontrollera om en xtt metod är relevant för ett objekt
collect show	Visa samlingsbilden
collect clear	Töm samlingsbilden
close alarmlist	Stäng larmlistan
close eventlist	Stäng händleslistan
close graph	Stäng en graf
close navigator	Stäng navigatorn
create opmenuitem	Skapa ett menyalternativ i operatörsfönstret
create item	Skapa ett xtt menyalternativ
crossreference	Visa korsreferenser
define	Definiera en symbol
delete opmenuitem	Ta bort ett xtt menyalternativ
delete item	Ta bort ett menyalternativ i operatörsfönstret
emit signal	Sänd en Ge signal.
eventlist	Hantera händelselistan
exit	Stäng xtt
help	Visa hjälp
login	Inloggning av användare
logout	Utloggning av användare
logging	Logga attribut till fil
open graph	Öppna en graf
open jgraph	Öppna en java graf
open file	Öppna en fil eller URL
open fileview	Öppna en fileview
open navigator	Öppna navigatorn
open trace	Öppna trace för ett plc-fönster
open trend	Öppna en trendkurva
open operatorwindow	Öppna operatörsfönstret
read object	Läs innehållet av ett objekt från fil
search	Leta efter objekt eller sträng
set advanceduser	Sätt avancerad användare
set display	Sätt format for visning av utvalt attribut
set folder	Aktivera en flik i ett flikfönster.
set parameter	Sätt värdet på ett attribut
set signal	Sätt invert, conversion och test på signaler.
set subwindow	Visa in graf i ett Ge window objekt.
set graph	Hantering av graf.
setup	Xtt konfigurering
show version	Visa xtt version
show symbol	Visa en symbol
show plcpgm	Visa PlcPgm objekt
show plcthread	Visa PlcThread objekt
show links	Visa länkar
show logfiles	Visa xtt loggfiler
show subsrv	Visa prenumerations servern
show subcli	Visa prenumerations klienten
show device	Visa I/O enheter

show remnode	Visa RemNode objekt
show remtrans	Visa RemTrans objekt
show database	Visa runtime databasen
show file	Visa filer
show graph	Visa graf-filer
show time	Visa aktuell tid
show default	Visa default katalog
show children	Visa ett objekts barn
show message	Visa ett meddelande
show objectlist	Visa alla objects av en specifik klass
show objid	Visa ett objekts identitet
show parameter	Visa ett objektsattribut
show logging	Vissa ett loggingentry
show eventlist	Visa händselistan
show alarmlist	Visa larmlistan
show user	Visa nuvarande användare
sound	Spela upp ett ljud definerat av ett ljudobjekt
store	Lagra en lista på attribut in en fil
write object	Skriv innehållet i ett objekt till en fil

15.1 Kommando call method

Anropa en xtt metod för ett objekt.

Xtt metoder aktiveras normalt från objektets popupmeny. Det här är ett alternativt sätt att anropa en metod, t ex från en kommando tryckknapp i en Ge graf.

xtt> call method /method= /object=

/method= Namnet på metoden, dvs attributet ButtonName
i definitionen av metoden.

/object= Namn på objektet.

15.2 Kommando check method

Anropa en xtt-metods filterfunktion för ett objekt.

Filterfunktionen används för att avgöra om en metod knapp ska visas eller ej. Detta kommando kan användas för att göra en metod knapp okänslig i en Ge graf. Det returnerar filter-värdet till anroparen.

Kommandot kan endast användas i en Ge graf.

xtt> check method /method= /object=

/method=	Namnet på metoden, dvs attributet ButtonName i definitionen av metoden.
/object=	Namn på objektet.

15.3 Kommando add parameter

Addera ett attribut till listan av attribut, och visa attributets värde.

En lista på attribut skapas med kommandot 'show parameter', och fler attribut kan adderas till listan med 'add parameter'.

Objekt som matchar specificerad klass, namn och hierarki visas i listan.

x tt> add parameter /parameter= /name= /class= /hierarchy=

/parameter=	Namn på attributete som ska visas.
/name=	Namn på objektet, Wildcard är tillåtet.
/class=	Visar objekt av angiven klass.
/hierarchy=	Visar objekt under angivet objekt i hierarkin.

Se även

show parameter

15.4 Kommando show parameter

Skapa en ny lista, addera attribut till listan, och visa värdet på attributet.

Fler attribut kan adderas till listan med 'add parameter'.

Objekt som matchar klass, namn och hierarki beskrivningens visas på skärmen.

xrt> show parameter /parameter= /name= /class= /hierarchy=

/parameter=	Namn på attributet som ska visas.
/name=	Namn på objektet, Wildcard är tillåtet.
/class=	Visar objekt av angiven klass.
/hierarchy=	Visar objekt under angivet objekt i hierarkin.

Se även

add parameter

15.5 Kommando set parameter

Sätt värdet på ett attribut.

x tt> set parameter /name= /value=

/name= Namn på attributet.

/value= Värde som ska sättas på attributet.

/bypass= Gå förbi behörighetsrestriktioner.

15.6 Kommando set folder

Aktivera en flik i ett flikfönster objekt.

x tt> set folder 'graphname' /name= /index=

/name= Namn på filkfönster objektet.

/index= Index på filken som ska aktiveras. Indexeringen startar från 0.

15.7 Command set signal

Sätt invert, conversion, test och testvalue för en signal.

Conversion kan sättas på Di, li och Ai,

Invert kan sättas på Di och Do.

Test kan sättas på Do, Ao och Io.

Testvalue kan sättas på Do.

x tt> set signal conversion /on [/name=]

x tt> set signal conversion /off [/name=]

x tt> set signal invert /on [/name=]

x tt> set signal invert /off [/name=]

x tt> set signal test /on [/name=]

x tt> set signal test /off [/name=]

x tt> set signal testvalue /on [/name=]

x tt> set signal testvalue /off [/name=]

/name	Signalnamn. Om signalnamnet ej är angivet används den utvalda signalen.
-------	---

/on	Sätt entitet till.
-----	--------------------

/off	Set entitet från.
------	-------------------

15.8 Command set subwindow

Visar en graf in ett Ge window objekt. Kommandot används i kommando knappar för att byta innehållet i ett window objekt. /source anger namnet på den graf som ska visas. Även objekt-bilder kan visas genom att aktuellt objekt med /object.

x tt> set subwindow 'graphname' /name= /source= [/object=]

'graphname'	den graf där window objektet ligger.
/name	namn på window objektet. '\$current' anger nuvarande graf.
/source	namn på den Ge graf som ska visas i window objektet.
/object	anger aktuellt objekt när grafen är en objekts-bild.
/x0	Övre vänstra x-koordinaten i Ge för grafens gräns. Om x0, y0, x1, y1 anges kan en annan del av grafen än den som har specificerats i Graph attribute visas. Implementerad för grafer i multiview celler, inte för window-objekt.
/y0	Övre vänstra y-koordinaten i Ge för grafen.
/x1	Nedre högra x-koordinaten i Ge för grafen.
/y1	Nedre högra y-koordinaten i Ge för grafen.
/continue	Normalt exekveras inte några fler actions i kommando-knappen, eftersom den nuvarande grafen kan ha bytt ut sig själv. För att också exekvera efterföljande actions för kappen, adderas /continue.

Om 'set subwindow' kommandot används i ett script, och byter ut den nuvarande grafen, ska scriptet avslutas med ett anrop till exit(GLOW__SUBTERMINATED).

```
main()  
  set subwindow motor1 /name=subw1 /source=motor1.pwg  
  ...  
  exit( GLOW__SUBTERMINATED);  
endmain
```

15.9 Kommando add menu

Addera ett menyalternativ till xtt's meny-hierarki.

Menyn placeras sist i menyn. Menyalternativet kan användas för att exekvera ett xtt kommando eller visa attributen för ett objekt.

xtt> add menu /text= /command=

xtt> add menu /text= /object=

/text	Text för menyalternativet.
/command	Xtt som exekveras när menyalternativet aktiveras.
/object	Objekt som visas när menyalternativet aktiveras.

Se även

create item

delete item

15.10 Kommando collect

Addera ett attribut till samlingsbilden.
Om namnet inte anges adderas det utvalda objektet.

x tt> collect

x tt> collect /name=

/name Namn på attributet.

Exempel

x tt> collect /name=hql-hvk-Start.ActualValue

Se även

collect show

collect clear

15.11 Kommando collect show

Visa samlingsbilden.

```
xtt> collect show
```

Se även

collect

collect clear

15.12 Kommando collect clear

Töm samlingsbilden.

```
xtt> collect clear
```

Se även

collect

collect show

15.13 Kommando create opmenuitem

Skapa ett menyalternativ i operatörsfönstrets meny.

Menyalternativet kan användas för att exekvera ett xtt kommando.

xtt> create opmenuitem /name= /command= /pixmap=

/name	Namn på menyalternativet inklusive namn på ovanliggande menyer. Namnet bör inledas med 'Functions-' eller med dess översättning. För övriga namnled skapas nya undermenyer om de inte redan finns. Ex Functions-MyMenu-MyGraph
/command	Xtt kommando som exekveras när menyalternativet aktiveras.
/before	Menyalternativet läggs först i menyn.
/after	Menyalternativet läggs sist i menyn.
/pixmap	Pixmap för menyalternativet: 'graph' eller 'curve'.

Exempel

xtt> create opmenuitem /name="Functions-MyMenu-MyGraph"/after/pixmap=graph

Se även

delete opmenuitem

15.14 Kommando create item

Skapa ett menyalternativ i xtt's meny hierarki.
Menyalternativet kan användas för att exekvera ett
xtt kommando, eller för att öppna en meny.

xtt> create item /text= /command= /pixmap= /destination= /after /before /firstchild /lastchild
xtt> create item /text= /menu /destination= /after /before /firstchild /lastchild

/text	Text på menyalternativet.
/command	Xtt kommando som exekveras när menyalternativet aktiveras.
/menu	När menyalternativet aktiveras öppnas en meny.
/destination	Destinations meny. Kan vara förälder eller syskon. Om destination är en null-sträng skapas menyalternativet på topnivån.
/after	Placerar menyalternativet som syskon efter destinationen.
/before	Placerar menyalternativet som syskon före destinationen.
/firstchild	Placerar menyalternativet som första barn till destinationen.
/lastchild	Placerar menyalternativet som sista barn till destinationen.
/pixmap	Pixmap för menyalternativet: 'map', 'leaf', 'graph', 'list' eller 'script'.

Exempel

```
xtt> create item /text="Motor1"/dest=Maintenance-Motors/command="open graph motor1"/first
```

Se även

add menu
delete item

15.15 Kommando crossreference

Visar korsreferenser.

Korsreferenser kan visas för

- Signaler.
- Objektreferenser med GetData.
- Referenser i Ge grafer.
- c-funktioner eller strängar som förekommer koden i CArithm eller DataArithm objekt.

Om det inte anges några kvalifierare visas korsreferenser för det utvalda objektet.

x tt> crossreference

x tt> crossreference /name=

x tt> crossreference /function= [/brief]

x tt> crossreference /string= [/brief]

/name Namn på ett objekt.

/function Namn på en c-funktion anropad i en CArithm eller DataArithm.

/string Sträng som letas efter i koden för CArithm och DataArithm objekt.

Exempel

x tt> cross /name=hql-hvk-Start

x tt> cross /function="CreateHvkObject"

15.16 Command `emit signal`

Sänd en Ge signal.

Signalen är sänd till en specificerad graf, eller till alla grafer och multiview om graf inte är specificerad.

xtt> emit signal /signalname= [/graph=] [/instance=]

/signal_name

Signalnamn.

/graph

Graf eller multiview till vilken signalen är riktad.

/instance

Instansobjekt om grafen är en objektsbild.

15.17 Kommando exit

Stäng xtt.

```
xtt> exit
```

15.18 Kommando define

Definiera en symbol.

```
xrt> define 'symbolname' 'text'
```

Se även

symbol
show symbol
symbolfile

15.19 Kommando delete opmenuitem

Ta bort ett menyalternativ från operatörsfönstrets meny.
Menyalternativet kan vara skapat av användaren, eller tillhöra standardmeny.

x tt> delete opmenuitem /name=

/name Namn på menyalternativet.

Exempel

x tt> delete opmenuitem /name=Functions-Curves

Se även

create opmenuitem

15.20 Kommando delete item

Ta bort ett menyalternativ ur Xtt-menyn.

Menyalternativet kan vara skapat av användaren, eller tillhöra xtt's standardmeny.

x tt> delete item /name=

/name Namn på menyalternativet.

Exempel

x tt> delete item /name=exit

Se även

add menu
create item

15.21 Kommando eventlist

Hantera händelselistan

x tt> eventlist load

Ladda händelselistan.

x tt> eventlist unload

Ta bort händelselistan.

x tt> eventlist acknowledge /priority=

Kvittera det sista larmet med angiven priritet. Prioriteten kan vara A, B, C, D, I eller NOA. NOA kvitterar det sista larmet som inte är ett A larm.

x tt> eventlist acknowledge /autoacknowledge=

Kvittera alla larm automatiskt. Kvitteringen kommer att ske cykliskt med cykeltiden specificerad av /autoacknowledge. Den här funktionen är gjord for test och kräver System privilegium.

x tt> eventlist acknowledge /all

Kvittera alla larm.

Se även

show alarmlist

show eventlist

15.22 Kommando help

Visar hjälp-information för ett ämne.

Hjälp-informationen letas upp i en hjälp fil. Filen kan vara bas-hjälpfilen, projekt-hjälpfilen eller någon anna hjälpfil.

Om ingen hjälpfil anges letas efter ämnet i bas- och projekt-hjälpfilerna.

**xtt> help 'subject' [/popnavigator] [/bookmark=] [/helpfile=]
[/returncommand=][/width=][/height=]**

/popnavigator	Hjälpfönstret läggs ovanpå övriga fönster och ges input focus. Detta kan användas när kommandot ges i en Ge graf.
/bookmark	Nammet på ett bokmärke inom ett ämne. Visningen kommer att placeras på bokmärket.
/helpfile	En hjälpfil som innehåller information om angivet ämne.
/returncommand	Ett kommando som exekveras vid retur från hjälpsamnet.
/width	Önskad bredd på hjälpfönstret.
/height	Önskad höjd på hjälpfönstret.

Se även

helpfile

15.23 Kommando login

Inloggning med användare och passerord. Användarens privilegier kommer att hämtas från användardatabasen. Dessa avgör användarens behörighet.

```
xtd> login 'username' 'password'
```

Se även

logout
show user

15.24 Kommando `logout`

Loggar ut en användare, och återgår till den ursprungliga användaren.

xtt> logout

Se även

login

show user

15.25 Kommando open graph

Öppna en Ge graf. Grafen kan anges men en pwg-fil eller med ett XttGraph objekt.

```
x tt> open graph 'filename' /width= /height= /scrollbar /menu /navigator  
/instance= /focus= /inputempty  
x tt> open graph /object= /focus= /inputempty
```

/width	Bredd i pixel.
/height	Höjd i pixel.
/scrollbar	Fönstret skapas med skrollister.
/menu	Fönstret skapas med en meny.
/navigator	En navigationsfönster öppnas.
/object	Ett XttGraph-objekt. Om namnsträngen börjar med '*', ersätts * med nodenamnet.
/instance	Öppnar objektsbilden för det objekt som anges i instance.
/focus	Namn på ett inmatnings objekt. Objektet kommer att ges input focus när grafen öppnas.
/inputempty	Texten i inmatningsfältet töms.
/pwindow	Filnamn för ett annat fönster som grafen ska placera ovanför. Strängen \$current anger nuvarande fönster när grafen öppnas från en tryckknapp i detta fönster.
/pinstance	Instance namn för pwindow grafen, om pwindow är en objektsbild.
/fullscreen	Öppna fönstret med full skärm utan ram.
/maximize	Öppna fönstret maximerat med ram.
/iconify	Öppna fönstret ikoniserat.
/hide	Visa inte fönstret på skärmen.
/x0	Övre vänstra x koordinat i Ge för grafens gränslinje. Om x0, y0, x1, y1 anges visas en annan del av grafen än den som är specificerad Graph attributes x0, y0, x1, y1.
/y0	Övre vänstra y koordinat i Ge för grafens gränslinje.
/x1	Undre högra x koordinat i Ge för grafen gränslinje.
/x1	Undre högra y koordinat i Ge för grafen gränslinje.

15.26 Kommando open jgraph

Öppna en java frame. Java frame'en kan vara exporterad från Ge eller editerad i t ex JBuilder.

xjt> open jgraph 'name'

15.27 Command open navigator

Öppna runtime navigatorn. Om ett object har angivits kommer detta att visas och väljas ut i objektstrådet.

x tt> open navigator [/object=]

/object

Namn på ett objekt eller attribut som visas och väljs ut.

15.28 Command open file

Öppnar en fil eller URL.

Om filnamnet innehåller ett snedstreck ska det omges av citationstecken.

x tt> open file 'filename'

Exempel

x tt> open file "\$pwrp_doc/manual.pdf"

15.29 Kommando open fileview

Öppnar en fileview som visar filer som matchar det specificerade mönstret. Fileview'en kan startas i 'Open File' eller 'Save File' mod, dvs för att öppna eller spara filer.

I öppnings mod, väljs en fil ut och det utvalda filnamnet skrivs till ett sträng attribut i databasen (target). Samtidigt sätts ett digitalt attribut (trigger) för att indikera att öppningen ska utföras. Själva öppningen av filen kan utföras av ett DataFRead objekt i plc programmet, men måste programmeras av användaren.

I spara mod. kan en fil väljas ut, men man kan även mata in ett filnamn. Filnamnet skrivs till ett strängattribut och en digital signal sätts. Själva spara funktionen kan utföras med ett DataFWrite objekt från plc programmet.

xrt> open fileview /file= /target= /trigger= /type=save [/ftype=]

xrt> open fileview /file= /target= /trigger= /type=open

/file	Mönster för filnamn, innehållande wildcard '*', för filer som ska visas i fileview'en.
/ftype	Default filtyp. Om ett angivet filnamn (is spara mod) inte innehåller filtyp, adderas den defaulta filtypen till filnamnet.
/target	Strängattribut där det utvalda eller inmatade filnamnet ska lagras.
/trigger	Boolean attribut som sätt för att indikera att öppna eller spara ska utföras.
/type	'save' eller 'open'

Exempel

```
open fileview /file="$pwrp_load/*.txt"/ftype="txt"/target=P1-Sv.ActualValue/trigger=P1-Dv.ActualValue/type=save
```

15.30 Kommando open trace

Öppna Plc Trace för ett plc-fönster. Om namn inte anges öppnas trace för det utvalda PlcPgm dller PlcWindow objektet.

x tt> open trace [/name=] [/center=]

/name	PlcPgm eller PlcWindow.
/center	Objekt i plc-fönstret som ska centeras och makeras. Ange sista segmentet av namnet.

Exempel

x tt> open trace /name=hql-hvk-flow /center=Pid0

15.31 Kommando open trend

Öppnar kurvfönstret för ett DsTrend eller PlotGroup objekt.

xtt> open trend [/name=] [/title=]

/name DsTrend eller PlotGroup. Kan vara en lista av DsTrend-objekt separerade med komma-tecket.

/title Titel på kurvfönstret.

Exempel

xtt> open trend /name=hql-hvk-flow-Trend,hql-hvk-temperature-Trend

15.32 Kommando open operatorwindow

Öppna operatörsfönstret.

```
xtt> open operatorwindow 'opplace-object'
```

15.33 Kommando close graph

Stäng en Ge graf. Grafen kan anges som en pwg-fil eller med ett XttGraph objekt.

xtt> close graph 'filename' [/instance=] [/iconify]

xtt> close graph /object=

/object	Ett XttGraph-objekt. Om namnsträngen börjar med '*', ersätts * med nodeobjektet.
/instance	Stänger objektsbilden för angiven instans.
/iconify	Grafen blir ikoniserad istället för stängd.

15.34 Kommando `close alarmlist`

Stänger larmlistan.

```
xtt> close alarmlist
```

15.35 Kommando `close eventlist`

Stänger händleselistan.

`xtr> close eventlist`

15.36 Kommando close navigator

Stänger navigatören.

```
xtt> close navigator
```


15.37 Kommando search

Letar efter ett objektsnamn eller en sträng.

```
xtt> search 'object'
```

```
xtt> search /regularexpression 'expression'
```

```
xtt> search /next
```

15.38 Kommando **plcscan**

Sätt plc scan till eller från för alla plcprogram eller för utvalt eller specificerat program.

System privileger krävs.

xrt> plcscan [/on] [/off] [/all] [/name=]

/on	Sätt scan till.
/off	Sätt scan från.
/all	Sätt scan till eller från för alla plcprogram.
/name	Sätt scan till eller från för specificerat plcprogram eller fönster.

15.39 Kommando read object

Läs innehållet i ett objekt från ej fil.

Läser innehållet i en fil, vanligtvis skapad med kommandot 'write object', och lägger in data i det specificerade objektet.

xtt> read object /object= /file=

/object

Namn på objektet.

/file

Namn på fil att läsa från.

15.40 Kommando set advanceduser

Sätt eller återställ advanced user.

```
xtt> set advanceduser  
xtt> set noadvanceduser
```

Se även

advanced user

15.41 Command `set display`

Sätt format for visning av utvalt attribut. Ett attribut kan visas som decimalt, hexadecimalt, oktalt, binärt, eller som heltal eller flyttal. Objektidentiteter och attribut-referenser kan visas som identiteter.

```
xtt> set display default
xtt> set display hexadecimal
xtt> set display decimal
xtt> set display octal
xtt> set display binary
xtt> set display integer
xtt> set display float
xtt> set display identity
```

15.42 Command `set graph`

Utför lagring av öppna inmatningsfält med egenskapen `EscapeStore` satt, som ännu inte är lagrade i databasen.

xtt> set graph 'filename' /escapestore [/instance=]

`/escapestore` Utför lagring av öppna inmatningsfält med egenskapen `EscapeStore` satt, som ännu inte är lagrade i databasen.

`/instance` Instansobject specificerat för grafen.

15.43 Xtt setup

Unställning av xtt egenskaper.

ConfigureObject
DefaultDirectory
Scantime
AlarmMessage
AlarmBeep
AlarmReturn
AlarmAck
SymbolFilename
Verify
AdvancedUser

RttConfigure objekt.
Default filkatalog för kommandofiler.
Scantid för att updatera värden.
Visa senaste larm på Xtt's meddelanderad.
Ge ljudsignal om det finns okvitterade larm.
Visa retur händelser i händelselistan.
Visa kvittens händelser i händleslistan.
Symbolfil.
Sätt verifiering vid exekvering av kommandofiler.
Användaren är avancerad.

15.44 Kommando show version

Visa xtt version.

```
xtt> show version
```


15.45 Kommando show symbol

Visa en symbol, eller alla symboler.

xtt> show symbol 'symbol'

xtt> show symbol

Visa symbol 'symbol'

Visa alla symboler

Se även

define

symbol

15.46 Kommando show plcpgm

Lista alla PlcPgm objekt is systemet.

```
xtt> show plcpgm
```

15.47 Kommando `show plcthread`s

Visa alla PlcThread objekt.

PlcThread objekt innehåller information och statistic för plc-trådarna.

xrt> show plcthreads

15.48 Kommando show links

Visa länkar till andra ProviewR system.

```
xtt> show links
```

15.49 Kommando `show logfiles`

Visa xtt logg-filer på arbetskatalogen.

En logg-fil skapas med logging funktionen i rtt och xtt och har filtypen `.rtt_log`.

En loggfil öppnas med dubbel-klick, eller genom att välja ut filen och trycka på return.

xtt> show logfiles

15.50 Kommando show alarmlist

Öppna larmlistan.

```
xtt> show alarmlist
```

15.51 Kommando show eventlist

Öppna händleslistan.

```
xtt> show eventlist
```

15.52 Command `show message`

Öppna ett fönster med ett meddelande.

xtt> show message /text= [/title=]

/text	Meddelande.
/title	Titel på fönstret.

15.53 Command `show objectlist`

Öppna ett fönster med en lista på all objekt av en specific klass eller att antal specifika klasser. När ett objekt i listan aktiveras öppnas objektbilden för objektet.

xrt> show objectlist /class= [/title=] [/sort]

/class	Namn på en klass, eller flera klasser avgränsade med komma-tecken.
/title	Titel på fönstret.
/sort	Objekten sorteras i alfabetisk ordning.

15.54 Kommando `show objid`

Visar objektsidentitet för ett objekt.

Om namn inte är angivet, visas identiteten för utvalt objekt.

xtt> show objid [/name=] [/hexadecimal]

/name Namn på objektet.

/hexadecimal Visar objektindex i hexadecimal form.

15.55 Kommando show user

Visa nuvarande användare med privilegier.

```
xtt> show user
```

15.56 Kommando `sound`

Spela upp ett ljud definerat med ett `Sound` eller `SoundSequence` objekt.

```
xtt> sound 'object'
```

15.57 Kommando store

Lagra en attributlista till en skriptfil, eller lagra symboltabellen till en script-fil.

Den genererade script-filen kan exekveras från kommando prompten med '@'filnamn'. Den återfinns även bland script-filer i menyn under 'Store'.

Om /collect används, återskapas attributlistan som en samlingsbild.

xtt> store 'filename' [/collect]

xtt> store 'filename' /symbols

/collect

Listan återskapas som en samlingsbild.

/symbols

Den nuvarande symboltabellen lagras.

15.58 Kommando show logging

Visar ett logging entry.

```
xtt> show logging /entry=
```

15.59 Kommando logging

Ett antal kommandon för att hantera logging funktionen i Xtt.

logging start
logging stop
logging set
logging show
logging store

15.60 Kommando write object

Skriv innehållet i ett objekt till en fil.

x tt> write object /object= /file=

/object

Namn på objektet.

/file

Namn på fil att skriva till.

16 Xtt script

Exekvera ett script

Datatyper och declarationer

Datatyper
Datatyps konvertering
Variabel deklarationer
Operatorer

Uttryck

main-endmain
function-endfunction
if-else-endif
while-endwhile
for-endfor
break
continue
goto
include

In och utmatnings funktioner

ask()
printf()
say()
scanf()

Filhanterings funktioner

fclose()
felement()
fgets()
file_search()
fopen()
fprintf()
fscanf()
translate_filename()

Sträng funktioner

edit()
element()
extract()
sprintf()
strchr()
strrchr()
strlen()
strstr()
toupper()

tolower()

Databas funktioner

GetAttribute()
SetAttribute()
GetChild()
GetParent()
GetNextSibling()
GetClassList()
GetClassListAttrRef()
GetNextObject()
GetNextAttrRef()
GetObjectClass()
GetNodeObject()
GetRootList()

System funktioner

exit()
get_namespace()
set_namespace()
system()
time()
tzset()
verify()

Vektor funktioner

arrayclear()
arraypush()
arraysize()
sort()

Matematiska funktioner

cos()
EVEN()
MAX()
MIN()
ODD()
random()
sin()

Diverse funktioner

CutObjectName()
ConfirmDialog()
TextDialog()
MessageError()
MessageInfo()
GetCurrentText()
GetCurrentObject()
get_pwr_config()
get_node_name()
getmsg()
EVEN()
ODD()

```
get_language()
GetUser()
GetPrivileges()
GetGraphInstance()
GetGraphInstanceNext()
SetSubwindow()
Quit()
```

xtt-commands

xtt-commands

16.1 Exekvera ett script

En script-fil exekveras från kommando-raden med kommandot

```
xtt> '@filename'
```

16.2 Datatyper

Datatyperna är float, int och string.

int	heltals värde.
float	32-bitars flyttals värde.
string	textsträng, 80 tecken lång (null terminerad).

Det finns tre olika tabeller som en variabel kan deklareras i: local, global och extern. En lokal variabel är känd inom en funktion, en global är känd i alla funktioner inom en fil, en extern är känd i alla filer som exekveras i en session.

16.3 Datatyps konvertering

Om ett uttryck består av variabler och funktioner med olika datatyper kommer variablerna att konverteras med företräde string, float, int. Om två operander har typerna float och string, eller int och string, kommer resultatet att bli string. Vid en tilldelning, kommer värdet av ett uttryck att konverteras till typen för den tilldelade variabeln. Detta gäller även om resultatet är en sträng och variablerna är av type float och int.

Exempel

```
string str;
int i = 35;
str = "Luthor" + i;
The value in str will be "Luthor35".

float f;
```

```
string str = "3.14";
int i = 159;
f = str + i;
The value in f will be 3.14159.
```

16.4 Variabel deklARATIONER

En variabel måste deklareras innan den används.

En deklaration består av

- tabellen (global eller extern, om local ska inte tabellen anges)
- datatypen (int, float eller string)
- variabelnamn (känsligt för stora och små bokstäver)
- antal element, om variabeln är en vektor inom hakparenteser, eller om en vektor med längd 0, bara hakparenteser.
- likameds tecken följt av initierings värde, om det utelämnas är initialvärdet 0 eller en null-sträng
- semikolon.

En extern variabel bör tas bort (med delete) när den inte längre används.

Även globala variabler kan tas bort med deletegb.

Exempel

```
int i;
float flow = 33.4;
string str = "Hello";
float width[5] = (1.20, 2.44, 4.81, 7.77, 9.20);
extern int jakob[20];
string simon[];
global float ferdinand = 1234;
...
delete jakob[];
deletegb ferdinand;
```

Namnrymd

Externa variabler kan deklareras i olika namnrymd genom att sätta namnrymd med `set_namespace()`. Det kan användas för att exekvera uppsättningar av script samtidigt i samma process. En uppsättning script kan hantera gemensamma externa variabler, och genom att använda olika namnrymder kan samma uppsättning exekveras i flera versioner utan sammanblandning av de externa variablerna.

16.5 Operatorer

Operatorerna har samma funktion som i c, med vissa begränsningar. Alla operatorer är inte implementerade. Vissa operatorer (+,=,==) kan även operera på string variabler. Prioriteten för operatorer är densamma som i c.

Operator	Beskrivning	Datatyper
+	plus	int, float, string
-	minus	int, float
*	multiplikation	int, float

/	division	int, float
++	inkrement, postfix only.	int, float
--	dekrement, postfix only	int, float
>>	bitar höger-skift	int
<<	bitar vänster-skift	int
<	mindre än	int, float
>	större än	int, float
<=	mindre eller lika med	int, float
>=	större eller lika med	int, float
==	lika med	int, float, string
!=	ej lika med	int, float, string
&	bitvis och	int
	bitvis eller	int
&&	logisk och	int
	logisk eller	int
!	logisk not	int
=	tilldelning	int, float, string
+=	addera och tilldela	int, float
-=	minus och tilldela	int, float
&=	logisk och och tilldelning	int
=	logisk eller och tilldelning	int

16.6 Script uttryck

main-endmain	Main funktion.
function-endfunction	Funktions deklaration.
if-else-endif	Villkorlig exekvering.
while-endwhile	While loop.
for-endfor	For loop.
break	Avsluta while eller for loop.
continue	Fortsätt while eller for loop.
goto	Hoppa till label.
include	Inkludera script fil.

16.6.1 main-endmain

main och endmain satserna kontrollerar var exekveringen börjar och slutar. Om det inte finns några main och endmain satser, startar exekveringen i början på filen, och slutar i slutet på filen.

Exempel

```
main()  
    int a;  
  
    a = p1 + 5;  
    printf( "a = %d", a);  
endmain
```

16.6.2 function-endfunction

En funktions deklARATION består av

- datatypen för funktionens retur värde
- namnet på funktionen
- en argumentlista separerade med kommatecken och omgiven av parenteser. Argumentlistan innehåller typdeklARATION och namn på varje argument.

Argumenten som skickas med vid anropet kommer att konverteras till den typ som är deklarerad i argumentlistan. Om ett argument ändrar värde inne i funktionen, kommer det nya värdet att överföras till anroparen. På detta sätt blir det möjligt att returnera andra värden än retur värdet för funktionen. En funktion kan innehålla en eller flera return satser. Return satsen kommer att flytta över exekveringen till anroparen och returnera det angivna värdet.

Exempel

```
function float calculate_flow(float a, float b)
    float c;
    c = a + b;
    return c;
endfunction
```

```
...
flow = korr * calculate_flow( v, 35.2);
```


16.6.3 if-else-endif

Raderna mellan en if-endif stats exekveras om uttrycket i if-statsen är sant. Uttrycket ska omges av parenteser. Om en else sats hittas mellan if och endif, kommer raderna mellan else och endif att exekveras när uttrycket är falskt.

Exempel

```
if ( i < 10 && i > 5)
    a = b + c;
endif
```

```
if ( i < 10)
    a = b + c;
else
    a = b - c;
endif
```

16.6.4 while-endwhile

Raderna mellan en while-endwhile sats exekveras så länge uttrycket i while-satsen är sant. Uttrycket ska omges av parentser.

Exempel

```
while ( i < 10)
  i++;
endwhile
```

16.6.5 for-endfor

Raderna mellan en for-endfor sats exekveras så länge mitten-uttrycket i for-satsen är sant. for-satsen består av tre uttryck, avgränsade med semicolon och omgivna av parenteser. Det första uttrycket exekveras före den första loopen, det tredje exekveras efter varje loop, och det i mitten exekveras före varje loop, och om det är sant, gör ytterligare ett varv, annars lämnas loopen.

Exempel

```
for ( i = 0; i < 10; i++)  
    a += b;  
endfor
```

16.6.6 break

En break sats kommer att söka efter nästa endwhile eller endfor sats och försätta exekveringen på raden efter denna.

Exempel

```
for ( i = 0; i < 10; i++)  
    a += b;  
    if ( a > 100)  
        break;  
endfor
```

16.6.7 continue

En continue sats kommer att söka efter närmast föregående while eller for sats och försätta att utföra loopen.

Exempel

```
for ( i = 0; i < 10; i++)  
    b = my_function(i);  
    if ( b > 100)  
        continue;  
    a += b;  
endfor
```

16.6.8 goto

En goto sats orsakar ett hopp i exekveringen till en rad som är definierad med en label. Label raden avslutas med ett kolon.

Exempel

```
b = attribute("MOTOR-ON.ActualValue", sts);
if (!sts)
    goto some_error;
...
some_error:
    say("Something went wrong!");
```

16.6.9 include

En script include-fil som innehåller funktioner kan inkluderas med `#include` satsen. Default filtyp för filen är `'.rtt_com'`.

Exempel

```
#include <my_functions>
```

16.7 In och utmatnings funktioner

Funktion	Beskrivning
ask	Skriv en fråga och läs ett svar.
printf	Formaterad utskrift.
say	Skriv en text.
scanf	Formaterad läsning.

16.7.1 ask()

int ask(string question, (arbitrary type) reply)

Beskrivning

Promptar för inmatning med den angivna strängen.
Returnerar antal lästa element, 1 or 0.

Arguments

string	question	Prompt.
godtycklig typ	reply	Inmatat svar. Kan vara int, float eller string.

Exempel

```
string reply;  
  
ask( "Do you want to continue? [y/n] ", reply);  
if ( reply != "y")  
    exit();  
endif
```

16.7.2 printf()

int printf(string format [, (arbitrary type) arg1, (arbitrary type) arg2])

Beskrivning

Formaterad utskrift. C-syntax. Format argument och inget, ett eller två värde argument.

Returnerar antal utskrivna tecken.

Argument

string	format	Format.
godtycklig typ	arg1	Värde argument. Valfritt, Kan vara int, float eller string.
godtycklig typ	arg2	Värde argument. Valfritt. Kan vara int, float eller string.

Exempel

```
printf( "Watch out!");  
printf( "a = %d", a);  
printf( "a = %d och str = %s", a, str);
```

16.7.3 say()

```
int say( string text)
```

Beskrivning

Skriver ut en sträng.

Argument

string	text	Text att skriva ut.
--------	------	---------------------

Exempel

```
say( "Three quarks for Muster Mark!");
```

16.7.4 scanf()

```
int scanf( string format , (godtycklig typ) arg1)
```

Beskrivning

Formaterad inmatning. C-syntax.
Returnerar antal inlästa tecken.

Argument

string	format	Format.
godtycklig typ	arg1	Värde argument. Returnerat. Kan vara int, float eller string.

Exempel

```
scanf( "%d", i);
```

16.8 Filhanterings funktioner

Funktion	Beskrivning
fclose	Stäng en fil.
felement	Extrahera ett element från senaste lästa rad.
fgets	Läs en rad från en fil.
file_search	Sök efter filer.
fopen	Öppna en fil.
fprintf	Formaterad utskrift till fil.
fscanf	Formaterad läsning från fil.
translate_filename	Översätt omgivnings-variabler i ett filnamn.

16.8.1 fclose()

```
int fclose( int file)
```

Beskrivning

Stänger en öppnad fil.

Argument

int	file	fil-id returnerad av fopen.
-----	------	-----------------------------

Exempel

```
int infile;  
infile = fopen("some_file.txt", "r");  
...  
fclose( infile);
```

16.8.2 felement()

```
string felement( int file int number, string delimiter, string str)
```

Beskrivning

Extraherar ett element från en sträng av element läst från en file med fgets() funktionen. felement() kan användas istället för element() när den lästa strängen är längre än sträng-storleken 256. felement() kan hantera rader upp till 1023 tecken.

Argument

int	number	elementets nummer.
string	delimiter	avgränsnings tecken.

Exempel

```
string elem1;
int file;
string line;

file = fopen( "my_file.txt", "r");
while( fgets( line, file))
    elem1 = felement( 1, " ");
endwhile
```

16.8.3 fgets()

```
int fgets( string str, int file)
```

Beskrivning

Läser en rad från en angiven fil.
Returnerar noll vid filslut.

Argument

string	str	Läst rad. Returnerad.
int	file	Fil id returnerad av fopen.

Exempel

```
file = fopen("some_file.txt","r");  
while( fgets( str, file))  
    say( str);  
endwhile  
fclose( file);
```


16.8.4 file_search()

```
int file_search( string pattern, string found_file, int pass)
```

Beskrivning

Söker efter filer.

Ett sökmönster med wildcard kan anges för att söka på flera filer.

Söksekvensen är indelad i passen init, next och end. Vid första anropet anges pass init (1). Vid sökning av flera filer med samma mönster anges pass next (0). Sökningen avslutas med pass end (2).

Returnerar udda status om en fil har hittats, annars jämn status.

Argument

string	pattern	Namn på fil som ska sökas efter, kan innehålla wildcard (*).
string	found_file	Hittad fil.
int	pass	Pass. Init (1), next (0) eller end (2).

Exempel

```
string pattern = "*.txt";
string found_file;
int sts;

sts = file_search(pattern, found_file, 1);
while (sts & 1)
    printf("Processing %s\n", found_file);
    ...
    sts = file_search(pattern, found_file, 0);
endwhile
file_search(pattern, found_file, 2);
```

16.8.5 fopen()

```
int fopen( string filespec, string mode)
```

Beskrivning

Öppnar en fil för att läsa eller skriva.

Returnerar en fil identitet, Om filen inte kunde öppnas, returneras noll.

Argument

string	filespec	Namn på filen.
string	mode	Access mod

Exempel

```
int infile;  
int outfile;  
  
infile = fopen("some_file.txt","r");  
outfile = fopen("another_file.txt","w");  
...  
fclose( infile);  
fclose( outfile);
```

16.8.6 fprintf()

```
int fprintf( int file, string format [, (godtycklig typ) arg1,  
(godtycklig typ) arg2])
```

Beskrivning

Formaterad utskrift på fil. C-syntax. Format argument och inget, ett eller två värde argument.
Returnerar antal utskrivna tecken.

Argument

int	file	Fil id returnerat av fopen.
string	format	Format.
godtycklig typ	arg1	Värde argument. Valfritt. Kan vara int, float eller string.
godtycklig typ	arg2	Värde argument. Valfritt. Kan vara int, float eller string.

Exempel

```
int outfile;  
outfile = fopen( "my_file.txt", "w");  
if (!outfile)  
    exit();  
fprintf( outfile, "Some text");  
fprintf( outfile, "a = %d", a);  
fclose( outfile);
```

16.8.7 fscanf()

`int fscanf( int file, string format, (godtycklig typ) arg1)`

Beskrivning

Formaterad läsning från fil. C-syntax.
Returnerar antal inlästa tecken.

Argument

<code>int</code>	<code>file</code>	Fil id.
<code>string</code>	<code>format</code>	Format.
<code>godtycklig typ</code>	<code>arg1</code>	Värde argument. Returnerat. Kan vara int, float eller string.

Exempel

```
int file;  
int i;  
  
file = fopen( "my_file.txt", "r");  
if (file)  
    fscanf( file, "%d", i);  
    fclose( file);  
endif
```

16.8.8 translate_filename()

```
string translate_filename( string fname)
```

Beskrivning

Byter ut omgivnings variabler i ett filnamn.

Argument

string	fname	Ett filnamn.
--------	-------	--------------

Returns

string	Sträng med utbytta omgivningsvariabler.
--------	---

Exempel

```
string fname1 = "$pwrp_db/a.wb_load";  
string fname2;  
fname2 = translate_filename( fname1);
```

16.9 Sträng funktioner

Funktion	Beskrivning
edit	Ta bort överfödiga mellanslag och tab.
element	Extrahera ett element från en sträng.
extract	Extrahera en delsträng från en sträng.
sprintf	Formaterad utskrift till en sträng.
strchr	Returnera första förekomsten av ett tecken i en sträng.
strrchr	Returnera sista förekomsten av ett tecken i en sträng.
strlen	Beräkna längden på en sträng.
strstr	Returnera första förekomsten av en delsträng i en sträng.
tolower	Konvertera till små bokstäver.
toupper	Konvertera till stora bokstäver.

16.9.1 edit()

```
string edit( string str)
```

Beskrivning

Tar bort inledande och avslutande blanktecken och tabbar, och ersätter flera tabbar och blanktecken med ett blanktecken.
Returnerar den editerade strängen.

Argument

string	str	sträng som ska editeras.
--------	-----	--------------------------

Exempel

```
collapsed_str = edit(str);
```

16.9.2 element()

```
string element( int number, string delimiter, string str)
```

Beskrivning

Extraherar ett element från en sträng av element.
Returnerar det extraherade elementet.

Argument

int	number	elementets nummer.
string	delimiter	avgränsnings tecken.
string	str	sträng med element.

Exempel

```
string str = "mary, lisa, anna, john";  
string elem1;  
elem1 = element( 1, ",", str);
```


16.9.3 extract()

```
string extract( int start, int length, string str)
```

Beskrivning

Extraherar de angivna tecknen från angiven sträng.
Returnerar de extraherade tecknen som en sträng.

Argument

int	start	start positionen för första tecknet. Första tecknet har position 1.
int	length	antalet tecken som ska extraheras.
string	str	sträng som tecknen ska extraheras från.

Exempel

```
extracted_str = extract( 5, 7, str);
```

16.9.4 sprintf()

```
int sprintf( string str, string format [, (arbitrary type) arg1, (arbitrary type) arg2])
```

Beskrivning

Formaterad utskrift. C-syntax. Format argument och inget, ett eller två värde argument.

Returnerar antal utskrivna tecken.

Argument

string	str	Sträng att skriva till.
string	format	Format.
godtycklig typ	arg1	Värde argument. Valfritt, Kan vara int, float eller string.
godtycklig typ	arg2	Värde argument. Valfritt. Kan vara int, float eller string.

Exempel

```
string str;  
int items;  
  
sprintf( str, "Number of items: %d", items);
```

16.9.5 strchr()

```
int strchr( string str, string c)
```

Beskrivning

Returnerar första förekomsten av ett tecken in en sträng.

Argument

string	str	Sträng att söka i.
string	c	Tecken att söka efter.

Returns

int	Index för första förekomsten av ett tecken. Första tecknet har index 1. Returnerar noll om tecknet inte hittas.
-----	---

Exempel

```
string str = "index.html";  
int idx;  
  
idx = strchr( str, ".");
```

16.9.6 strrchr()

```
int strrchr( string str, string c)
```

Beskrivning

Returnerar sista förekomsten av ett tecken in en sträng.

Argument

string	str	Sträng att söka i.
string	c	Tecken att söka efter.

Returns

int	Index för den sista förekomsten av tecknet. Första tecknet har index 1. Returnerar noll om tecknet inte hittas.
-----	---

Exempel

```
string str = "/usr/local/pwrrt";  
int idx;  
  
idx = strrchr( str, "/");
```

16.9.7 strlen()

```
int strlen( string str, string c)
```

Beskrivning

Beräknar längden av en sträng.

Argument

string	str	Sträng att beräkna längden för.
--------	-----	---------------------------------

Returns

int	Strängens längd.
-----	------------------

Exempel

```
string str = "/usr/local/pwrrt";  
int len;  
  
len = strlen( str);
```

16.9.8 strstr()

```
int strstr( string str, string substr)
```

Beskrivning

Returnar första förekomsten av en delsträng i en sträng.

Arguments

string	str	Sträng att söka i.
string	substr	Delsträng att söka efter.

Returns

int	Index för första förekomsten av delsträngen. Första tecknet har index 1. Returnerar noll om delsträngen inte hittas.
-----	--

Exempel

```
string str = "index.html";  
int idx;  
  
idx = strstr( str, ".html");
```

16.9.9 tolower()

```
string tolower( string str)
```

Beskrivning

Konverterar en sträng till små bokstäver.

Argument

string	str	sträng som ska konverteras.
--------	-----	-----------------------------

Returns

string	sträng med små bokstäver.
--------	---------------------------

Exempel

```
string str1 = "Buster Wilson";  
string str2;  
str2 = tolower( str);
```

16.9.10 toupper()

```
string toupper( string str)
```

Beskrivning

Konverterar en sträng till stora bokstäver.

Argument

string	str	sträng som ska konverteras.
--------	-----	-----------------------------

Returns

string	sträng med stora bokstäver.
--------	-----------------------------

Exempel

```
string str1 = "Buster Wilson";  
string str2;  
str2 = toupper( str);
```


16.10 System funktioner

Funktion	Beskrivning
exit	Avsluta scriptet.
get_namespace	Hämta aktuell namespace.
set_namespace	Sätt namespace för externa variabler.
system	Exekvera ett shell kommando.
terminate	Avsluta processen.
time	Hämta systemtiden.
tzset	Sätt tidszon.
verify	Skriv ut exekverade rader.

16.10.1 exit()

int exit()

Beskrivning

Avslutar exekveringen av en fil.

Exempel

```
exit();
```

16.10.2 get_namespace()

```
string get_namespace()
```

Beskrivning

Returnerar aktuell namespace.

Exempel

```
string current_namespace;  
current_namespace = get_namespace();
```

16.10.3 set_namespace()

```
set_namespace( string namespace)
```

Beskrivning

Sätt namespace för externa variabler.

Maxstorleken på namespace är 31 tecken. Om längden på argumentet överstiger maxlängden används de 31 sista tecknen i strängen.

Argument

string	namespace	Ny namespace.
--------	-----------	---------------

Exempel

```
set_namespace(p1);
```

16.10.4 system()

```
int system( string cmd)
```

Beskrivning

Exekvera ett shell kommando.

Argument

string	cmd	Shell kommando att exekvera.
--------	-----	------------------------------

Returns

int	Returvärdet är -1 vid fel eller annars returstatus för kommandot.
-----	---

Exempel

```
string cmd;  
  
cmd = "firefox http://www.proview.se";  
system( cmd);
```

16.10.5 terminate()

int terminate()

Beskrivning

Avsluta processen.

16.10.6 time()

```
string time()
```

Beskrivning

Returnerar nuvarande tid i strängformat.

Exempel

```
string t;  
t = time();
```

16.10.7 tzset()

```
string tzset( string timezone)
```

Beskrivning

Sätt tidszon.

Exempel

```
tzset("Europe/Stockholm");
```


16.10.8 verify()

```
int verify( [int mode])
```

Beskrivning

Sätter eller visar verifikationens mod. Om verifiering är till, visas alla exekverade rader på skärmen.

Returnerar nuvarande verifikations mod.

Argument

int	mode	verifikation till (1) eller från (0). Valfri.
-----	------	---

Exempel

```
verify(1);
```

16.11 Vektor funktioner

Funktion

arrayclear

arraypush

arraysize

sort

Beskrivning

Töm en dynamisk vektor.

Addera till en dynamisk vektor.

Hämta storleken på en vektor.

Sortera en vektor.

16.11.1 arrayclear()

```
int arrayclear((arbitrary array type) array)
```

Beskrivning

Ta bort alla element i en vektor och sätt storleken till 0.
Vektor-argumentet ska specificeras med hakparenteser.

Returnerar operationens status.

Argument

(arbitrary array type)	array	Namn på vektorn. Ska anges med hakparenteser.
------------------------	-------	---

Exempel

```
float temp[];  
...  
arrayclear(temp[]);
```

16.11.2 int arraypush((godtycklig vektortyp) array

Beskrivning

Addera ett element sist i vektorn och lägg in angivet värde i elementet.
Vektor-argumentet ska specificeras med hakparenteser.

Returnerar operationens status.

Arguments

(godtycklig vektortyp)	array	Namn på vektorn. Ska anges med hakparenteser.
(godtycklig typ)	value	Värde add addera till vektorn.

Exempel

```
float x[];  
  
arraypush(x[], 29.2);
```

16.11.3 int arraysize((godtycklig vektortyp) array)

Beskrivning

Hämta storleken på en vektor, dvs antalet element i vektorn.
Vektor-argumentet ska specificeras med hakparenteser.

Returnerar storleken på vektorn.

Argument

(godtycklig vektortyp) array

Namn på vektorn. Ska anges med hakparenteser.

Exempel

```
float x[];  
int size;  
...  
size = arraysize(x[]);
```

16.11.4 int sort((godtyclig vektortyp) array1 [(godtyclig vektortyp) array2 - array8])

Beskrivning

Sortera en sträng-vektor i alfabetisk ordning, eller en int- eller float-vektor i numerisk ordning.

Upp till sju ytterligare vektorer kan anges som kommer att sorteras på samma sätt som den första vektorn. Dessa vektorer ska ha samma storlek, eller vara större än den första vektorn.

Vektor-argumenten ska specificeras med hakparenteser.

Returnerar operationens status.

Arguments

(godtycklig vektortyp)	array1	Namn på vektorn som ska sorteras. Ska anges med hakparenteser.
(godtycklig vektortyp)	array2 - array8	Ytterligare valfria vektorer som kommer att sorteras på samma sätt som den första vektorn. Ska anges med hakparenteser.

Exempel

```
string name[];  
string description[];  
int sts;  
...  
sts = sort(name[], description[]);
```

16.11.5 Matematiska funktioner

Funktion	Beskrivning
cos	Cosinus funktion.
EVEN	Testa om värdet är jämnt.
MAX	Returnera största värdet.
MIN	Returnera minsta värdet.
ODD	Test om värdet är udda.
random	Returnera ett slumptal.
sin	Sinus funktion.

16.11.5.1 cos()

float cos(float angle)

Beskrivning

Returnerar cosinus-värdet för en vinkel i grader.

Argument

float	angle	Vinkel i grader.
-------	-------	------------------

Exempel

```
float x;  
float angle = 45;  
  
x = cos(angle);
```


16.11.5.2 EVEN()

int EVEN(int val)

Beskrivning

Testa om ett värde är jämnt.

Returnerar 1 om värdet är jämnt, annars 0.

Argument

int	val	Värde.
-----	-----	--------

16.11.5.3 MAX()

float MAX(float v1, float v2)

Beskrivning

Returnerar det största av två värden.

Argument

float	v1	Värde.
float	v2	Värde.

Exempel

```
float f1;  
float f2;  
float max;
```

```
max = MAX(f1, f2);
```

16.11.5.4 MIN()

float MIN(float v1, float v2)

Beskrivning

Returnerar det minsta av två värden.

Argument

float	v1	Värde.
float	v2	Värde.

Exempel

```
float f1;  
float f2;  
float min;  
  
min = MIN(f1, f2);
```

16.11.5.5 ODD()

int ODD(int val)

Beskrivning

Testa om ett värde är udda.

Returnerar 1 om värdet är udda, annars 0.

Argument

int	val	Värde.
-----	-----	--------

16.11.5.6 random()

```
float random(float min_value, float max_value)
```

Beskrivning

Returnerar ett slumpmässigt värde i det angivna intervallet.

Argument

float	min_value	Undre gräns för slumptalet.
float	max_value	Övre gräns för slumptalet.

Exempel

```
float val;
```

```
val = random(0.0, 100.0);
```

16.11.5.7 sin()

float sin(float angle)

Beskrivning

Returnerar sinus-värdet för en vinkel i grader.

Argument

float	angle	Vinkel i grader.
-------	-------	------------------

Exempel

```
float x;  
float angle = 45;
```

```
x = sin(angle);
```

16.11.6 Databas funktioner

Function	Description
GetAttribute()	Hämta värde för ett attribut.
SetAttribute()	Sätt värde på ett attribut.
GetChild()	Hämta barn till ett objekt.
GetParent()	Hämta föräder till ett objekt.
GetNextSibling()	Hämta syskon för ett objekt.
GetClassList()	Hämta första instansen av en klass.
GetClassListAttrRef()	Hämta första attributinstansen av en klass.
GetNextObject()	Hämta nästa instans av en klass.
GetNextAttrRef()	Hämta nästa attributinstansen av en klass.
GetObjectClass()	Hämta klassen för ett objekt.
GetNodeObject()	Hämta nod-objektet.
GetRootList()	Hämta första objektet i rot-listan.

16.11.6.1 GetAttribute()

(variable type) GetAttribute(string name [, int status])

Beskrivning

Hämta värdet för angivet attribut. Typen av det returnerade värdet beror på typen av attributet. Attributet kommer att konverteras till int, float eller string.

Argument

string	name	namn på attributet som ska hämtas.
int	status	status för operationen. Returnerad. Om noll kunde inte attributet hämtas. Valfritt.

Exempel

```
int alarm;  
int sts;  
  
alarm = GetAttribute("Roller-Motor-Alarm.ActualValue");  
on = GetAttribute("Roller-Motor-On.ActualValue", sts);  
if ( !sts)  
    say("Could not find motor on attribute!");
```


16.11.6.2 SetAttribute()

```
int SetAttribute( string name, (arbitrary type)value [, int publicwrite])
```

Beskrivning

Sätt värdet på angivet attribut.

För att sätta värdet på ett vanligt attribut krävs RtWrite privilegiet. Det är dock möjligt skriva till attribut som är definierade som PublicWrite även utan detta privilegie, om publicwrite argumentet anges. PublicWrite attribut can bl a hittas i klasserna PublicAv, PublicIv och PublicDv.

Returnerar status av operationen.

Argument

string	name	namn på attributet som ska sättas.
<godtycklig typ>	value	värde som ska sättas.
int	publicwrite	om 1, skrivning till ett attribut definierad som PublicWrite är tillåtet även utan RtWrite privilegiet.

Exempel

```
float value = 22.2;
int sts;

sts = SetAttribute("Roller-Motor-Reference.ActualValue", value);
if ( !(sts & 1))
    printf( "SetAttribute error %d\n", sts);
endif
```

16.11.6.3 GetChild()

```
string GetChild( string name)
```

Beskrivning

Hämta första barnet till ett objekt. Nästföljande barn kan hämtas med `GetNextSibling()`.

Returnerar namnet på barnet. Om det inte finns något barn, returneras en null-sträng.

Argument

string	name	objektets namn.
--------	------	-----------------

Exempel

```
string child;  
  
child = GetChild("Roller-Motor");
```

16.11.6.4 GetParent()

```
string GetParent( string name)
```

Beskrivning

Hämta föräldern till ett objekt.

Returnerar förälderns namn. Om det inte finns någon förälder returneras en null-sträng.

Argument

string	name	objektets namn.
--------	------	-----------------

Exempel

```
string parent;  
  
parent = GetChild("Roller-Motor");
```

16.11.6.5 GetNextSibling()

string GetNextSibling(string name)

Beskrivning

Hämtar nästa syskon till ett objekt.

Returnerar namnet på syskonet. Om det inte finns något nästa syskon returneras en null-sträng.

Argument

string	name	objektets namn.
--------	------	-----------------

Exempel

```
string name;
int not_first;

name = GetChild("Rt");
not_first = 0;
while ( name != "")
    if ( !not_first)
        create menu/title="The Rt objects"/text="'name'"/object="'name'"
    else
        add menu/text="'name'"/object="'name'"
    endif
    not_first = 1;
    name = GetNextSibling(nname);
endwhile
if ( !not_first )
    MessageError("No objects found");
```

16.11.6.6 GetClassList()

```
string GetClassList( string class)
```

Beskrivning

Hämta första objektet av angiven klass. Nästa objekt av klassen kan hämtas med `GetNextObject()`. Returnerar namnet på första objektet. Om det inte finns några instanser av klassen returneras en null-sträng.

Argument

string	name	namn på klassen.
--------	------	------------------

Exempel

```
string name;  
  
name = GetClassList("Dv");
```

16.11.6.7 GetClassListAttrRef()

```
string GetClassListAttrRef( string class)
```

Beskrivning

Hämta första objekt eller attribut-objekt av en viss klass. Nästa attribut-objekt av klassen kan hämtas med GetNextAttrRef().

Returnerar namnet på det första objektet eller attribute-objektet. Om det inte finns någon instans, returneras en null-sträng.

Argument

string	name	namn på klassen.
--------	------	------------------

Exempel

```
string name;  
  
name = GetClassListAttrRef("Dv");
```

16.11.6.8 GetNextObject()

```
string GetNextObject( string name)
```

Beskrivning

Hämta nästa objekt i klasslistan.
Returnerar namnet på objektet. Om det inte finns något nästa objekt returneras en null-sträng.

Argument

string	name	objektets namn.
--------	------	-----------------

Exempel

```
string name;  
  
name = GetClassList("Di");  
while ( name != "")  
    printf("Di object found: %s", name);  
    name = GetNextObject(name);  
endwhile
```

16.11.6.9 GetNextAttrRef()

```
string GetNextAttrRef( string name)
```

Beskrivning

Hämta nästa objekt eller attribut-objekt i klasslistan.
Returnerar namnet på objektet eller attribute-objektet. Om det inte finns någon nästa objekt, returneras en null-sträng.

Argument

string	classname	namn på klassen.
string	name	namn på attribut-objekt.

Exempel

```
string name;  
  
name = GetClassListAttrRef("Di");  
while ( name != "")  
    printf("Di object found: %s", name);  
    name = GetNextAttrRef("Di", name);  
endwhile
```


16.11.6.10 GetObjectClass()

```
string GetObjectClass( string name)
```

Beskrivning

Hämta klassen för ett objekt.
Returnerar klassens namn.

Argument

string	name	objektets namn.
--------	------	-----------------

Exempel

```
string class;  
  
class = GetObjectClass("Motor-Enable");
```

16.11.6.11 GetNodeObject()

```
string GetNodeObject()
```

Beskrivning

Hämta nod objektet.

Returnerar namnet på nod objektet.

Exempel

```
string node;  
node = GetNodeObject();
```

16.11.6.12 GetRootList()

```
string GetRootList()
```

Beskrivning

Hämta första objektet i rot listan.

Returnerar namnet på rotojektet. Nästa objekt i rotlistan kan hämtas med `GetNextSibling()`.

Exempel

```
string name;

name = GetRootList();
while( name != "")
    printf( "Root object found: %s", name);
    name = GetNextSibling(name);
endwhile
```

16.11.7 Diverse funktioner

CutObjectName()	Klipp av ett objektsnamn.
ConfirmDialog()	Öppna en konfirmerings-dialog
TextDialog()	Öppna ett text info fönster.
MessageError()	Skriv ett felmeddelande.
MessageInfo()	Skriv ett informationsmeddelande.
GetCurrentText()	Hämta text för utvalt alternativ.
GetCurrentObject()	Hämta utvalt objekt.
get_pwr_config()	Hämta konfigurations-värden
get_node_name()	Hämta nodnamn
getmsg()	Hämta text för status.
EVEN()	Testa om ett värde är jämnt.
ODD()	Testa om ett värde är udda.
get_language()	Hämta nuvarande språk
GetUser()	Hämta nuvarande användare.
GetPrivileges()	Hämta privilegier för nuvarande användare.
GetGraphInstance()	Hämta instansobjekt för en objektgraf.
GetGraphInstanceNext()	Hämta instansobjekt för nästa graf.
SetSubwindow()	Set graf för ett Ge Window-objekt.
Quit()	Avsluta operatörmiljön.

16.11.7.1 CutObjectName()

```
string CutObjectName( string name, int segments)
```

Beskrivning

Kapa av de första segmenten av ett objektsnamn.
Returnerar de sista segmenten av ett objektsnamn. Antalet segment som returneras specificeras i det andra argumentet.

Argument

string	name	Objektsnamn med path.
int	segments	Antalet namnsegment som ska returneras.

Exempel

```
string path_name;  
string object_name;  
  
path_name = GetChild("Rt-Motor");  
object_name = CutObjectName( path_name, 1);
```

16.11.7.2 ConfirmDialog()

```
int ConfirmDialog( string title, string text)
```

Beskrivning

Öppna ett konfirmerings-fönster.
Returnerar 1 om Ja-kappen har aktiverats, annars 0.

Argument

string	title	Titel.
string	text	Konfirmerings text.

Exempel

```
if ( ! ConfirmDialog( "Confirm", "Do you really want to..."))  
    printf( "Yes is pressed\n");  
else  
    printf( "No is pressed\n");  
endif
```

16.11.7.3 TextDialog()

```
int TextDialog( string title, string text, [int image])
```

Beskrivning

Visa ett fönster med en text och en ikon som markerar fel, varning, info eller fråga.

Arguments

string	title	Titel.
string	text	Text.
int	image	Typ av ikon, 1 fel, 2 varning, 3 info, 4 fråga.

Exempel

```
TextDialog( "Error", "Motor is already started", 1);
```

16.11.7.4 `MessageError()`

```
string MessageError( string message)
```

Beskrivning

Skriv ett xtt felmeddelande på skärmen.

Exempel

```
MessageError("Something went wrong");
```


16.11.7.5 MessageInfo()

```
string MessageInfo( string message)
```

Beskrivning

Skriv ett xtt info meddelande på skärmen.

Exempel

```
MessageInfo("Everything is all right so far");
```

16.11.7.6 GetCurrentText()

```
string GetCurrentText()
```

Beskrivning

Hämta texten på det utvalda menyalternativet.

Exempel

```
string text;  
  
text = GetCurrentText();
```

16.11.7.7 GetCurrentObject()

```
string GetCurrentObject()
```

Beskrivning

Hämta utvalt objekt. Om inte något objekt är utval returneras en null-sträng.

Exempel

```
string object;  
  
object = GetCurrentObject();
```

16.11.7.8 get_pwr_config()

```
string get_pwr_config( string name)
```

Beskrivning

Hämta värdet för en ProviewR konfigureringsvariabel.
Konfigureringsvariabler sätts i /etc/proview.cnf.
Returnerar värdet på konfigureringsvariabeln.

Exempel

```
string id;  
  
id = get_pwr_config( "qcomBusId");
```

16.11.7.9 get_node_name()

```
string get_node_name()
```

Beskrivning

Hämta nodnamnet för aktuell nod.
Returnerar nodnamnet.

Exempel

```
name = get_node_name();
```

16.11.7.10 getmsg()

string getmsg(int status)

Beskrivning

Hämta texten för ett status värde.
Returnerar texten.

Exempel

```
msg = getmsg(sts);
```

16.11.7.11 EVEN()

```
int EVEN( int sts)
```

Beskrivning

Testa om ett heltal är jämnt.

Returnerar 1 om talet är jämnt, 0 om talet är udda.

Exempel

```
sts = SetAttribute( "Pump-V1-Switch.Description", "Valve switch open");  
if ( EVEN(sts))  
    printf("Couldn't set attribute\n");  
endif
```

16.11.7.12 ODD()

int ODD(int sts)

Description

Testa om ett heltal är udda.

Returnerar 1 om heltalet är udda, 0 om det är jämnt.

Exempel

```
sts = SetAttribute( "Pump-V1-Switch.Description", "Valve switch open");  
if ( ODD(sts))  
    printf("Set operation successful\n");  
endif
```


16.11.7.13 get_language()

string get_language()

Beskrivning

Hämta nuvarande språk.

Returns

string

Nuvarande språk, t.ex. en_us, sv_se, de_de.

Exempel

```
string lng;

lng = get_language();
if ( lng == "sv_se")
    create opmenu/name="Funktioner-Bilder-Översikt"/command="open graph overview"
endif
if ( lng == "en_us")
    create opmenu/name="Functions-ProcessGraphs-Overview"/command="open graph overview"
endif
```

16.11.7.14 GetUser()

```
string GetUser()
```

Beskrivning

Hämta nuvarande användare.

Exempel

```
string user;  
  
user = GetUser();
```

16.11.7.15 GetPrivileges()

```
int GetPrivileges()
```

Beskrivning

Hämta privilegier för nuvarande användare.

Exempel

```
int priv;

priv = GetPrivileges();
if ( priv & 4)
    # Has system privilege
    ...
endif
```

16.11.7.16 GetGraphInstance()

```
string GetGraphInstance( string graph)
```

Beskrivning

Hämta instansobjektet för en öppnad objektsbild.
Returnerar instansobjektet, eller en tom sträng om grafen inte är öppnad.

Argument

string	graph	Filnamn för graf.
--------	-------	-------------------

Exempel

```
string instance;  
  
instance = GetGraphInstance( "$pwr_exe/pwr_c_dv.pwg" );
```

16.11.7.17 GetGraphInstanceNext()

string GetGraphInstanceNext(string graph, string previous)

Beskrivning

Hämta nästa instansobjekt för den angivna objektsbilden.
Används när flera versioner av samma objektsbild är öppen för olika objekt.
Returnerar instansobjektet, eller en tom sträng om det inte finns någon mer instans.

Argument

string	graph	Filname för graf.
string	previous	Föregående instans.

Exempel

```
string instance;  
  
instance = GetGraphInstance( "$pwr_exe/pwr_c_dv.pwg");  
while ( instance != "")  
    printf( "Instance %s\n", instance);  
    instance = GetGraphInstanceNext( "$pwr_exe/pwr_c_dv.pwg", instance);  
endwhile
```

16.11.7.18 SetSubwindow()

```
string SetSubwindow(string graph, string window, string source, [string object, int self])
```

Beskrivning

Visar en graf in ett Ge window objekt. Kommandot används i kommando knappar för att byta innehållet i ett window objekt. source anger namnet på den graf som ska visas. Även objekt-bilder kan visas genom att aktuellt objekt med object argumentet.

Om funktionen exekveras i fönstret som byts ut, ska self argumentet vara 1.

Argument

string	graph	Namn på graf.
string	window	Namn på Ge windowobjekt.
string	source	pwg-fil or ge scrip som ska visas i window objektet.
string	object	Instansobjekt om grafen är en objektsbild.
int	self	Ska vara 1 om funktionen exekveras i det fönster vars innehåll byts ut.

Exempel pwg-fil

```
SetSubwindow("$current", "W1", "motor1.pwg", 0, 1);
```

Exempel script-fil

```
SetSubwindow("$current", "W1", "@m1script", 0, 1);
```

16.11.7.19 Quit()

Quit()

Beskrivning

Avsluta operatörsmiljön.

16.11.8 Xtt kommandon

Alla xtt kommandon finns tillgängliga i script koden. En xtt-kommando rad ska INTE avslutas med semikolon. Variabler kan substitueras i kommandot genom add omges av apostrofer.

Exempel

```
string name = "PUMP-VALVE-Open.ActualValue";  
float value = 2.2;  
set parameter/name='name'/value='value'
```

Exempel

```
string name;  
string parname;  
int j;  
int i;  
for ( i = 0; i < 3; i++)  
    parname = "vkv-test-obj" + (i+1);  
    create obj/name='parname'  
    for ( j = 0; j < 3; j++)  
        name = parname + "-obj" + (j+1);  
        create obj/name='name'  
    endfor  
endfor
```