



ProviewR
OPEN SOURCE PROCESS CONTROL

rt_xtt

Användarhandledning

Revision:	02 08 13
Version:	V4.0-1

Inledning.....	4
Operatörs-fönster.....	5
Okvitterade larm.....	5
Knappar för att öppna bas-funktioner.....	5
Funktions-knappar.....	5
Processbilder.....	6
Öppna en processbild.....	6
Stänga en processbild.....	6
Ändra i databasen.....	6
Layout.....	6
Distribution.....	6
Larmlista.....	7
Öppna larmlistan.....	7
Kvittering av larm.....	7
Larm raden.....	7
Meny.....	7
Konfigurering.....	7
Händelselista.....	8
Öppna händelselistan.....	8
Händelse raden.....	8
Menu.....	8
Runtime navigatören.....	9
Navigera med tangentbordet.....	9
Navigera med musen.....	9
Trace.....	10
Öppna trace.....	10
Trace-fönstret.....	10
View.....	10
Trace.....	10
Simulate.....	10
Menu.....	10
Systembilder.....	11
System/Nethandler/Link.....	11
System/Nethandler/Subscription Client.....	11
System/Nethandler/Subscription server.....	11
System/Communication/RemNode.....	11
System/Communication/RemTrans.....	11
System/Device.....	11
System/PlcThread.....	12
System/PlcPgm.....	12
System/Logging.....	12
Klassbilder.....	13
Användarklasser.....	13
Hierarkibilder.....	13
Trendkurvor.....	14
Kurv-fönstret.....	14
Logging.....	15
Starta loggingen.....	15
Kontinuerlig logging.....	15
Händelsestyrd logging.....	15
Buffer.....	15
Lagra.....	15
Kommandon.....	15
Villkorlig logging.....	15
Attribut.....	15

Javabilder.....	17
Öppna en javabild.....	17
Stänga en javabild.....	17
Konfigurering.....	17
Behörighetskontroll.....	18
Runtimenavigatören.....	18
Trace.....	18
Klassbilder.....	18
Ge och java-bilder.....	18
Funktionstangenter.....	19
Funktionsanrop.....	19
SetDig().....	19
ResetDig().....	19
ToggleDig().....	19
Command().....	19
Resursfil.....	19
Hjälp.....	20
Få hjälp.....	20
Operatörsfönstret.....	20
Meny.....	20
Kommando.....	20
Tryckknappar i bilder.....	20
Navigera i hjälptexten.....	20
Hjälpfiler.....	20
Syntax.....	20
Exempel.....	22
Generera hjälpfil från wb_load-filer.....	22
Generera html-filer från hjälpfil.....	23
Popup meny.....	24
Konfigurering.....	24
Konfigurering.....	26
Operatörsmod.....	26
Underhållsmod.....	26
Symbolfil.....	26
Kommandon.....	27
Script.....	28
Appendix A.....	29
Exempel på loggning.....	29

Inledning

rt_xtt är ett verktyg för operatörs-kommunikation, underhåll och felsökning. Xtt innehåller bl a följande komponenter:

- Operatörs-fönster. Ett fönster med de senaste larmen och snabbknappar för t ex bilder.
- Processgrafik. Processgrafik ritas i Ge-editorn och öppnas i xtt från operatörsfönstret eller från kommando-raden.
- Larmlista och händelselista.
- Navigatör. Med runtime-navigatören kan man titta i databasen och navigera i menyer och hjälp-filer.
- Trace. Felsökning i plc-programmet.
- Kommando-tolk. I xtt finns en kommandorad från vilken man kan exekvera en mängd kommandon.
- Script-hanterare. Man kan skriva script i ett c-liknande script-språk.
- System-bilder. Det finns ett antal bilder som visar information om näthanterare, plc-trådar mm.
- Objektsbilder. För vissa klasser finns objektsbilder (t ex PID, Av). För användar-klasser kan objekts-bilder ritas i Ge-editorn.
- Trend-kurvor. Visning av innehållet i trend-objekt.
- Logging. Attribut i databasen kan loggas på fil och visas i kurvform.
- Java-bilder. Man kan göra bilder i java (i t ex JBulider) och öppna dessa från xtt.
- Behörighetskontroll.
- Funktionstangentbord. Funktions-tangenter kan programmeras för att öppna bilder och exekvera kommandon.
- Hjälp. Xtt kan visa dokumentation om xtt-funktioner och proview-klasser. Det går även att skriva hjälp-texter om projektet, process-bilder, plc-program och larm.

Xtt kan startas i operatörs-mod eller i underhålls-mod. I operatörs-mod visas ett operatörsfönster med bl a larmtexter och snabb-knappar. I underhålls-mod visas runtime-navigatören.

Operatörs-fönster

Operatörsfönstret är det fönster som operatören använder för att öppna process-bilder, larm-fönster mm.

Okvitterade larm

Till vänster i operatörsfönstret visas senaste okvitterade larm. I övre delen visas A-larm med röd-makering. Genom att ändra på fönster-storleken kan man se mellan 2 och 5 A-larm. Här finns även en knapp för att kvittera det senaste A-larmet och en siffra som visar antalet rådande larm. I den undre delen visas B- C- D-larm samt info-meddelanden.

B-larm är gul-markerade, C-larm blå, D-larm violett och info-meddelanden gröna. Larmen visas i prioritets-ordning snarare än tids-ordning, dvs om det finns B-larm visas det senaste okvitterade B-larmet. Endast om det inte finns några okvitterade B-larm visas eventuella C-larm, osv. Det finns även en knapp för att kvittera det visade larmet.

Knappar för att öppna bas-funktioner

I mitten av operatörsfönstret finns ett antal knappar för att öppna vissa bas-funktioner i xtt. En del knappar syns ej vid minimal fönsterstorlek.

- Alarmlist. Öppnar larm-fönstret som visar rådande och/eller okvitterade larm.
- Eventlist. Öppnar händelselistan.
- Navigator. Öppnar runtime-navigatorn.
- Help. Visar hjälp-texter för projektet.

Funktions-knappar

Till höger finns det plats för upp till 15 knappar. Dessa konfigureras med XttGraph-objekt (se nedan) och kan användas för att öppna process-bilder eller sätta signaler i databasen.

Processbilder

Processbilder visar tillståndet i en anläggning, dvs hämtar värden från realtids-databasen och presenterar dessa i en bild som staplar, kurvor eller siffer-värden, eller genom att ändra färg och form på objekt i bilden. Operatören ges även möjlighet att påverka processen genom att klicka på tryckknappar och mata in värden.

Öppna en processbild

En process bild kan öppnas på följande sätt:

- med en funktions-knapp i operatörsfönstret.
- med en funktions-tangent på tangentbordet.
- med xtt-kommandot *'open graph'*.
- med xtt-kommandot *'show graph'* som listar alla .pwg-filer. Genom att dubbel-klicka på en fil i listan kan man öppna bilden.
- med en tryckknapp i en bild av typen *'Command'* som innehåller kommandot *'open graph'*.
- med ett meny-entry i xtt av typen *'Command'* som innehåller kommandot *'open graph'*.

Stänga en processbild

En processbild stängs genom att man klicka med MB3 någonstans i bilden, eller med *'Close'* i fönstrets meny (Alt+F4). Man kan även stänga en bild med *'close graph'* kommandot.

Ändra i databasen

En processbild är uppbyggd av objekt i form av fyrkanter, cirklar, ventiler, ramar osv. På objekten kan man sätt olika typer av dynamik så att de ändrar färg eller form, beroende på värden i databasen. Från vissa objekt kan man ändra värden i databasen genom att klicka eller dra på objekten. De objekt som man kan klicka eller dra på, *'hot'*-markeras när markören är placerade på dem. Det innebär att markören ändras till ett hårkors, samt att grafiken hos objektet förändras något. Vanligtvis ritas det med något tjockare linjebredd. Innan ändringen i databasen verkställs, kontrolleras att användaren har privilegier att ändra på det här objektet.

Layout

När en bild öppnas mha kommandot *'open graph'* kan man påverka hur mycket av bilden som visas, och hur bilden påverkas när man ändrar storlek på den.

Normalt anger man vid editering av bilden hur stor del av bildarean som ska visas, genom att ange övre vänstra och under högra hörnet för bilden. Har man gjort detta anpassas bilden skalning så att den alltid visar detta område. Dessutom låses förhållandet mellan fönstrets höjd och bredd så att det motsvarar utsnittet.

Om man inte anger ett bild-utsnitt kommer inte bilden att skalas om vid ändring av fönsterstorlek. Man kan då öppna bilden med

- scroll-lister.
- ett navigations-fönster med vilket man kan scrolla och zooma.
- en meny med zoom-funktioner.

Se kommandot *'open graph'* och kvalifierarna */scrollbar*, */navigator* och */menu*.

Distribution

Bilderna ritas i Ge-editorn. Denna genererar en .pwg som ska kopieras till målnodens pwrp_exe. Eventuella externa submodeller måste också kopieras.

Larmlista

Larmlistan visar rådande och okvitterade larm.

Öppna larmlistan

Larmlistan kan öppnas

- Från knappen 'Alarmlist' i operatörsfönstrets mittsektion.
- Från meny-entryt 'Alarm/Alarm list' i runtimenavigatörens standardmeny.
- Med kommandot 'show alarmlist'.

Kvittering av larm

Från larmbilden kan senaste okvitterade larm kvitteras från menyn 'Functions-Acknowledge' eller *Ctrl+K*.

Kvittering kan även ske från operatörsfönstrets kvittens-knappar, och med kommandot 'alarm acknowledge'. Kommandot kan t ex läggas ut på funktions-tangenter.

Larm raden

På en rad i larmfönstret visas ett rådande och/eller okvitterat larm.

I början på raden finns en färgmarkering och en bokstav som markerar larmets prioritet. A-larm markeras med rött, B-larm med gult. C-larm och D-larm markeras enbart med bokstav och Info-meddelanden med grönt utan bokstav. Därefter följer två symboler, en klocka som markerar att larmet är okvitterat, samt ett utropstecknet i en varnings-triangel som markerar att larmet är rådande. Efter detta följer tiden, larmtexten samt larm-namnet. Larm-namnet är vanligtvis objektet som ger upphov till larmet. Man kan välja i menyn om man vill dölja larmtexten eller larm-namnet.

Meny

File/Close	Stänger larmfönstret.
Functions/Acknowledge	Kvitterar senaste okvitterade larm.
Functions/Open Plc	Öppnar trace för larm-objektet. Detta kräver att larm-objektet är ett plc-objekt.
Functions/Display objekt in navigator	Visar larm-objektet i runtime-navigatörn.
View/Zoom in	Zoomar in.
View/Zoom out	Zoomar ut.
View/Zoom reset	Återställer till ursprunglig zoom-faktor.
View/Display hundredth	Visar hundradels sekund i larm-tiden.
View/Hide event name	Döljer larm-namnet.
View/Hide event text	Döljer larm-texten.
Help/Alarmlist	Visar hjälptext om larmlist-fönstret.
Help/Help on selected event	Visar hjälptext för utvalt larm. Detta kräver att det Dsup eller Asup-objektet för larmet finns som ett ämne (topic) i projektets xtt-hjälpfil.

Konfigurering

I user-objektet konfigurerar man en del saker som hör till larm-listan: MaxNoOfAlarms och SelectList. Vilket user-objekt som xtt läser beror på i vilken mod xtt startas. I operatörs-mod kopplas OpPlace-objektet till ett User-objekt via OpNumber-attributet. I underhålls-mod används det User-objekt som finns angivet i RttConfig-objektet.

Händelselista

Händelselistan visar larm och händelser. Händelser kan vara när ett larm blir rådande, när det kvitteras och när det upphör att vara rådande. Via Dsup och Asup-objekt kan man även skicka händelser som enbart visas i händelselistan och inte i larmlistan.

Öppna händelselistan

Händelselistan kan öppnas

- Från knappen *'Eventlist'* i operatörsfönstrets mittsektion.
- Från meny-entryt *'Alarm/Event list'* i runtimenavigatörens standardmeny.
- Med kommandot *'show eventlist'*.

Händelse raden

Händelseraden är ganska lik larm-raden i larmlistan. Markering av prioritet, tid, text och namn är densamma. Skillnaden ligger i att det finns tre olika typer av händelser, när ett larm går till, när det kvitteras och när larmtillståndet försvinner. Detta markeras med olika symboler, ett utropstecken när larmet går till, en bock för kvittens och ett korsat utropstecken när larmtillståndet upphör.

Menu

Funktionen hos menyn är densamma som för larmlistan.

Runtime navigatören

Med runtime navigatören navigerar man i en trädstruktur bestående av mappar och löv. Man kan navigera med musen eller med tangentbordet. Ur ergonomisk synvinkel är tangentbordet att föredra.

Navigera med tangentbordet

Här använder man framför allt piltangenterna för att navigera.

Med 'Upp' och 'Ner' väljer man ut en map eller ett löv. Mappen öppnas med 'Pil-höger' och stängs med 'Pil-vänster'. För att stänga en mapp kan man antingen välja ut den öppna mappen, eller välja ut ett stängt barn, under mappen.

Ett objekt i realtidsdatabasen öppnar man med 'Shift Pil-höger', dvs man trycker först på Shift-tangenten och håller den nedtryckt medan man trycker på 'Pil-höger'. Om man har satt 'advanced user' i setup'en räcker det med att trycka på 'Pil-höger' för att öppna objektet, om att objektet inte har några barn.

Om man har satt 'advanced user' kan man även ändra värdet på attribut med 'Pil-höger'. Detta har då samma funktion som 'Change value' i menyn.

Navigera med musen

Välj ut en hieraki genom att klicka på texten.

Öppna/stäng en map genom att klicka på mappen eller dubbelklicka på texten till höger om mappen.

Öppna ett objekt genom att klicka på mappen/lövet med Shift MB1, eller dubbelklicka på texten med Shift MB1.

Trace

Trace används för att felsöka i plc-program.

Med trace kan man titta på koden i ett plc-program och se status för olika plc-objekt.

Öppna trace

Trace öppnas genom att välja ut ett PlcPgm objekt i runtime-navigatören och aktivera 'Open Plc' i menyn. Ett visst PlcPgm-objekt hittar man genom att navigera i databasen, i menyentry't

System//PlcPgm i xtt standard-meny som listar alla plc-program, eller med kommandot 'show plcpgm'.

Man kan även öppna trace med kommandot 'open trace'.

Trace-fönstret

När man har öppnat trace för ett PlcPgm-objekt får man upp ett huvud-fönster med plc-koden, och ett navigations-fönster. Plc-koden kan vara ganska omfattande, vilket gör att man inte kan titta på all kod på en gång i huvudfönstret. Med hjälp av navigations-fönstret kan man zooma in den del av koden man är intresserad av. Den del av koden som visas i huvudfönstret visas som en fyrkant i navigations-fönstret, och denna kan man flytta på genom att dra med MB1, eller ändra storlek med MB2.

Det finns tre olika moder i trace-fönstret: view, trace och simulate.

View

I view-mode visas plc-koden. Vissa plc-objekt innehåller under-fönster med plc-kod. Dessa öppnas med Shift/DubbelKlick/MB1.

Trace

I trace-mode kopplas logiska plc-objekt till aktuella värden i databasen. Om värdet är 1 röd-markeras objektet. För analoga objekt och lite mer komplexa logiska objekt finns det inte något självklart attribut som kan kopplas till röd-markeringen. Dem kan man koppla sk analysobjekt till. Genom att dra med MB2 från ett utgång på ett objekt, sedan släppa MB2 på något fritt utrymme i fönstret, skapas ett analys-objekt som visar värdet på utgången. En uppsättning analys-objekt kan sparas genom att aktivera File/SaveTrace i menyn och återskapas vid ett senare tillfälle med File/RestoreTrace.

Simulate

Simulate-mod kan användas vid simulering av ett PlcPgm. Förutom de funktioner som finns i trace-mod kan man även sätta värden på logiska plc-objekt med Shift/Ctrl/DubbelKlick MB1. Simulate-mod kräver att användaren har privilegierna System eller RtWrite.

Menu

File/Close	Stänger fönstret.
File/SaveTrace	Sparar samtliga analys-objekt i fönstret.
File/RestoreTrace	Återskapar tidigare sparade analys-objekt.
Functions/Display object in navigator	Visar utvalt plc-objekt i runtime-navigatören.
Functions/Collect insert	Lägger in objektet i runtime-navigatörens samlingsbild (det attribut som objektet kopplats till i databasen).
View/Zoom in	Zoomar in.
View/Zoom out	Zoomar ut.
View/Zoom reset	Återställer till ursprunglig zoom-faktor.
Mode/View	Visar plc-koden utan koppling till databasen.
Mode/Trace	Trace-mod.
Mode/Simulate	Simulerings-mod.

Systembilder

Det finns ett antal systembilder som sammanställer information ur databasen och pooler. Systembilderna kan öppnas från standard-menyn i xtt eller med 'show' kommandot.

System/Nethandler/Link

Bilden visar de noder aktuell nod har QCOM kontakt med.

Kolumn	Beskrivning
Node	Nodnamn
Os	Operativsystem och hårdvara för noden.
Link	Status för länken. Kan vara Up, Active, Connected eller Down.
UpCount	Antal gånger länken till noden har gått upp.
TimeUp	Senast länken gick upp.

System/Nethandler/Subscription Client

Visar prenumerationer som aktuell node har begärt från andra noder.

På första raden visas antalet prenumerationer på den egna noden, på andra noder och på okända objekt. Observera att antalet okända prenumerationer, om denna är större än noll när samtliga noder är uppe, indikerar att det finns prenumerationer på objekt som inte existerar.

Kolumn	Beskrivning
Subid	Identitet på prenumerationen.
Time	Senaste ankomst för prenumerations-data.
Count	Antal gånger data har mottagits.
Node	Node som skickar data för prenumerationen.
Size	Storlek i bytes på prenumerations-data.
Attribute	Attribut för prenumerationen.

System/Nethandler/Subscription server

Visar prenumerationer som andra noder har begärt från aktuell nod.

Kolumn	Beskrivning
Subid	Prenumerations identitet.
Count	Antal gånger data för prenumerationen har skickats.
Node	Nod som data skickas till.
Size	Storlek på data byte.
Offset	Offset för prenumerations-attributet i objektet.
Attribute	Attribut som prenumereras på.

System/Communication/RemNode

Visar data för all RemNode-objekt på noden.

System/Communication/RemTrans

Visar data för alla RemTrans-objekt på noden.

När man öppnar en remtrans kan man se transbuffrarna i form av en c-struct. Detta kräver att struct-namnet och h-filen där structen finns definierad är inlagd in i remtrans-objektets StructName- resp StructFile-attribut. Se kommandot `show object /type` för de begränsningar som gäller vid tolkning av c-structar.

System/Device

Visar data för alla kort-objekt på noden.

System/PlcThread

Visar data för alla plc-trådar på noden.

System/PlcPgm

Listar alla PlcPgm på noden.

System/Logging

Med loggningsbilden kan man logga attribut i databasen på en fil. Loggning på maximalt tio filer kan göras samtidigt.

Här följer en lathund:

- Samla ihop de attribut som ska loggas i xtt's samling-bild.
- Gå in i loggnings-bilden för ett ledigt entry.
- Lägg in värden på tid, loggfil och typ, om default-värdena inte duger. Typen kan vara Cont(1) eller Event(2). Cont innebär att värden loggas kontinuerligt med angiven periodtid, Event innebär att värdena loggas när de ändrar värde.
- Klicka på *Insert* för att lägga in de attribut som valts ut i samlingsbilden.
- Klicka på *Start* för att starta loggningen. När loggningen är startad sätts 'Active' till 1.
- Stoppa loggningen genom att klicka på *Stop*.
- Titta på loggningen genom att klicka på *ShowFile*.

För övriga parametrar se kapitlet Logging och kommandot 'logging'.

Klassbilder

En klassbild presenterar innehållet i ett objekt av en viss klass. Alla objekt som tillhör klassen kan visas med klassbilden. En klassbild öppnas genom att man väljer ut objektet i navigatören och aktiverar *Functions/Open object graph* (Ctrl/G), eller med kommandot *'open graph /instance'*.

Klassbilder finns bl a för klasserna PID, Mode, DsTrend, PlotGroup samt för signaler och kanaler.

Behörigheten för att ändra i en klass-bild varierar mellan olika klasser. I avsnittet behörighetskontroll finns beskrivet vilka privilegier som krävs.

Användarklasser

Man kan göra klassbilder till användarklasser på följande sätt. Skapa en Ge-bild med samma namn som klassen, men med små bokstäver. I kopplingar till olika attribut i bilden skriver man '\$object' där man normalt skriver ett objekts-namn, t ex `$object.Length##Float32`. Kopiera .pwg-filen till \$pwrp_exe och öppna bilden från menyn med *Functions/Open object graph* (Ctrl/G) eller med kommandot *'open graph /instance'*.

Hierarkibilder

Om det finns flera hierarkier som liknar varandra kan man göra en gemensam bild för dem. Bilden för en hierarki öppnas med kommandot *'open graph/instance'* genom att skicka med hieraki-namnet som instans.

Trendkurvor

Det finns olika typer av trendkurvor i xtt. Här beskrivs de som konfigureras med DsTrend- och PlotGroup-objekt.

En trendkurva visar värdet av en signal under en den senaste tiden i form av en kurva. Tidsperioden kan variera beroende på hur ofta nya värden lagras. Värden för en signal lagras i ett DsTrend-objekt som har plats för 478 värden.

Man kan visa flera trender i samma diagram genom att kombinera ett antal DsTrend-objekt i en PlotGroup. I ett PlotGroup-objekt finns det plats för 20 DsTrend-objekt och här kan man även ange skalområden för koordinat-axlar.

En trendkurva öppnas genom att välja ut ett DsTrend-objekt eller ett PlotGroup-objekt och välja *'Functions/Open Object Graph'* i menyn, eller med kommandot `'open trend'`.

Kurvfönstret

Kurvfönstret visar trendkurvorna i diagramform, med tiden på x-axeln. Nutid (0) ligger längst till höger i diagrammet och kurvorna förskjuts åt vänster när nya mätvärden registreras.

Till vänster visas axlar för skalområden för de olika trenderna. Bakgrundsfärgen för en axel visar vilken trendkurva den motsvarar i diagrammet. I Diagrammet finns horisontella linjer, och genom att klicka på en koordinataxel kan man få de horisontella linerna att anpassa sig efter just denna axel. Med dubbelklick på en axel kan man ändra min och max-värde för denna.

Med *View/Show names* (Ctrl/W) i menyn visas signalnamnet för respektive kurva. Genom att klicka i den färgade rutan framför namnet kan man påverka om kurvan ska döljas eller ritas ut. I diagrammet kan maximalt 10 kurvor ritas ut åt gången, dvs max 10 kurvor kan vara markerade för att ritas samtidigt.

Under diagrammet finns ett navigations-fönster. I detta visas trendkurvorna i hela sin utsträckning i x- och y-led. Den del som visas i huvud-diagrammet markeras med en svart rektangel. Genom att flytta på rektangeln med MB1 och ändra storlek med MB2 kan man påverka vilket område som visas i huvud-diagrammet.

Man kan även zooma från menyn med *View/Zoom in* (Ctrl/i) resp *View/Zoom out* (Ctrl/o). Från menyn finns också funktioner för att visa kurvorna som fyllda, och för att få ljus bakgrundsfärg i diagrammet.

Logging

Loggingsfunktionen i xtt för det möjligt att logga värden i realtids-databasen på en fil. Detta kan användas t ex vid felsökning av snabba förlopp där man inte hinner se samspelet mellan olika signaler på annat sätt.

Starta loggingen

Xtt kan hantera upp till tio loggfiler samtidigt och i varje loggfil kan hundra parametrar loggas. En logging hanteras från loggingsbilden under System/Logging. Här finns tio olika entryn att välja mellan, där varje entry hanterar en loggfil.

Man startar en logging genom att samla ihop de parameterar som ska loggas i xtt's samling-bild. Därefter går man in i ett ledigt entry och klickar på insert-kappen. De insamlade parametrarna kopieras då in i loggnings-entryt. Lägg in cykeltid och klicka på start-kappen. Loggningen är nu igång och detta markeras med en etta i Active.

Loggningen stoppas genom att klicka på stopp-kappen. Man kan nu titta på resultatet genom att klicka på *ShowFile*. ShowFile öppnar ett kurvfönster som visar loggingen i ett diagram.

Loggingen kan vara av två typer, antingen loggas värdet av parametrarna kontinuerligt med en viss frekvens (kontinuerlig logging), eller loggas en parameter varje gång dess värde ändras (händelsestyrd logging).

Kontinuerlig logging

Värdet av parametrarna i entryt loggas vid varje cykel. På filen skrivs dessutom tiden sedan loggingen startades. Filen kan visas i xtt's kurv-fönster med 'ShowFile' eller läsas in av tex Excel för vidare bearbetning.

Händelsestyrd logging

En parameter i entryt loggas om värdet av parameteren har ändrats. På loggfilen skrivs tiden för ändringen och det nya värdet. Denna typ av logging kan inte visas med 'ShowFile'.

Buffer

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

Lagra

Med *Store/Restore* funktionen kan man lagra och återskapa uppsättningen i ett loggings-entry.

Kommandon

Loggingsfunktionen kan även styras med kommandot '*logging*' och därmed initieras med ett script. Observera att *Store*-funktionen lagrar entryt i form av ett script som kan användas utgångspunkt för ev modifieringar.

Villkorlig logging

I *ConditionParameter* kan man ange en digital parameter som styr loggingen. Logging sker enbart om *ConditionParameter*'n är true.

Attribut

Attribut	Beskrivning
Active	Visar om loggingen är startad.
Insert	Kopierar parameterar från samlingsbilden till loggings-entryt.

Start	Startar loggningen.
Stop	Stoppa loggningen.
Store	Lagrar uppsättningen och parameterar i loggings-entryt.
Restore	Läser in senaste lagrade uppsättning.
ShowFile	Visar loggings-resultatet i kurv-diagram. Endast kontinuerlig logging kan visas.
Time	Cykeltid i ms.
File	Namn på loggfil.
Type	Typ av logging. 1: kontinuerlig, 2: händelsestyrd.
BufferSize	Storlek på buffer i 512 bytes block.
FillBufferStop	Stoppa loggningen när buffern är full.
ShortName	Dölj hieraki-leden i namnet på loggning-parameterarna.
ParameterX	Parameterar som ska loggas.
ConditionParameter	Villkors parameter. Om en parameter har angetts här sker enbart loggningen om parametern har värdet 1.

Javabilder

Bilder gjorda i JBuilder eller någon annan java-IDE kan visas i xtt.

Öppna en javabild

Javabilder öppnas med kommandot 'open jgraph' där man skickar med namnet på den java-klass som ska startas.

För kunna öppna java-klasser startar xtt java i en subprocess. Vidare öppnas en QCOM-länk mellan xtt och java-processen, och över denna skickas information i form av xtt-kommandon. Java processen exekverar kommandot 'open jgraph' medan xtt exekverar övriga kommandon. Om ett kommando dyker upp i java-processen, t ex från en tryckknapp av kommando-typ, slussas detta till xtt. Detta gör att man t ex kan öppna en ge-bild från en java-bild.

Stänga en javabild

En javabild stängs med *Close* i menyn.

Konfigurering

Java processen startas genom att i OpPlace-objektets OpWinProgram lägga in "jop". Dessutom måste man se till att klasser som ska öppnas finns i CLASSPATH.

Behörighetskontroll

Varje gång xtt utför en operation som förändrar något värde i databasen, sker det först en behörighetskontroll. Om användaren har behörighet att utföra en ändring, genomförs ändringen, annars refuseras den. Behörigheten beror den akutella användarens privilegier, som hämta ifrån proview's användardatabas. De privilegier som är aktuella i runtime-miljön är RtRead, RtWrite, System, Maintenance, Process, Instrument, Operator1, Operator2, ..., Operator10.

Runtimenavigatören

För att göra förändringar från runtimenavigatören krävs *RtWrite* eller *System*. Detta gäller även kommandot 'set parameter'.

Trace

För att göra ändringar från trace i simulerings-mod krävs *RtWrite* eller *System*.

Klassbilder

Generellt för klassbilder krävs det *RtWrite* eller *System* för att göra ändringar från dem. Det finns några undantag. I ChanAi och ChanAo har även *Instrument* behörighet att ändra signal-områden, 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 den med kommandot 'open graph' och använda /access funktionen.

Ge och java-bilder

I Ge och java-bilder kan konstruktören i varje inmatingsfält eller trycknapp bestämma vilka vilka privilegier som krävs för att göra en ändring.

Funktionstangenter

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

Funktionsanrop

Följande funktioner kan anropas när en funktionstangent trycks ner.

SetDig()

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

ResetDig()

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

ToggleDig()

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

Command()

Exekverar ett xtt-kommando. Kommandot ges som argument.

De flesta xtt-kommandon kan användas här, t ex kommandon för att öppna bilder, öppna larm och händelse-listor, kvittera larm.

Resursfil

Kopplingen till funktionerna läggs i filen Rt_xtt (OBS! case sensitive) på inloggnings-katalogen. Detta är en X-windows resurs-fil och för närmare beskrivning av syntax se *X Windows System User's Guide*. I resursen hotkeys lägger man kopplingen mellan tangenter och funktions-anrop.

Exempel

```
! Configuration of Global function keys
*hotkeys: \
  <Key>F1: Command(event ack /prio=A) \n\
  <Key>F2: Command(event ack /prio=NOA) \n\
  <Key>F3: Command(show alarm) \n\
  <Key>F4: Command(show event) \n\
  <Key>F5: Command(close all/except=nu4_översikt) \n\
  <Key>F6: Command(open graph /object=*-OpNu4-nu4_översikt) \n\
  <Key>F7: Command(open graph /object=*-OpNu4-nu4_status_trp) \n\
  <Key>F8: Command(open graph /object=*-OpNu4-nu4_alla_plåtar) \n\
  <Key>F9: Command(open graph /object=*-OpNu4-nu4_vägval) \n\
  <Key>F10: SetDig(vhx-nu4-pf-InTillRikt.ActualValue) \n\
  <Key>F11: SetDig(vhx-nu4-pf-EjRikt.ActualValue) \n\
  <Key>F12: SetDig(vhx-nu4-pf-InMellanHög.ActualValue) \n\
  Shift<Key>F7: Command(open graph /object=*-OpNu4-nu4_sts)
```

Hjälp

Xtt kan användas för att visa upp hjälp-texter, och navigera mellan olika ämnen i hjälptexterna. Här beskrivs hur man hittar ett ämne, hur man navigerar, och hur man skriver en hjälp-fil.

Få hjälp

Hjälp kan öppnas på följande sätt

Operatörsfönstret

I operatörsfönstrets mitt-del finns en hjälp-knapp som visar hjälp-texter för projektet.

Meny

Help/Overview ger hjälp om Proview och operatörs-miljön. *Help/Projekt* ger hjälp om projektet.

Kommando

En hjälp-text öppnas med rtt-kommandot `help`. Till `help` skickar man med ämnet man är intresserad av. Kommandot `help help` visar t ex hur man använder kommandot `help`.

Tryckknappar i bilder

Tryckknappar i bilder kan kopplas till hjälp-texter. Det sker mha en tryckknapp av Command-typ och `help`-kommandot.

Navigera i hjälptexten

Rader som innehåller länkar till andra ämnen markeras med en pil. Genom att klicka på pilen eller dubbelklicka på raden öppnar man texten som länken pekar på.

Den översta radan är markerad med en vänster-pil. Genom att klicka på den, eller dubbelklicka på raden, kommer man tillbaka till föregående sida.

Man kan även navigera med tangentbordet genom att välja ut en länk med pil-upp och pil-ner. Öppna länken med pil-höger och återgå med pil-vänster.

Hjälpfiler

Xtt läser hjälptexter från hjälpfiler. För skriva hjälptexter för ett projekt skapar man filen `$pwrp_exe/xtt_help.dat`.

Syntax

Hjälpfilen byggs upp med taggar, som påverkar hur hjälptexten visas. Det finns även taggar för att skapa länkar till ämnen i hjälpfilen.

<topic>

`<topic>` inleder ett ämne och ska skrivas på radens första position. Topic-taggen ska följas av det sökord som hjälpfunktionen ska söka efter. Alla följande rader fram till `</topic>` taggen kommer att visas som text för ämnet.

`<topic> 'key'`

</topic>

Avslutar ett ämne. `</topic>` ska skrivas på radens första position.

Exempel

```
<topic> start engine
The engine will be started by...
</topic>
```

Följande kommando visar texten för detta ämne

```
x tt> help start engine
```

<bookmark>

Ett bokmärke är en rad inom ett ämne som kan visas med link-taggen eller med /bookmark kvalifieraren i help-kommandot. Bookmark-taggen ska placeras i slutet på en rad och ska följas av ett namn.

```
'some text' <bookmark> 'name'
```

Exempel

```
This is a bookmark <bookmark> first_engine
```

Följande kommando kommer att visa texten för ämnet och skrolla till bokmärket

```
x tt> help start engine /bookmark=first_engine
```

<link>

<link> taggen är en länk till ett annat hjälp-ämne. <link> taggen ska placeras i slutet på en rad. När raden med länken aktiveras med dubbel-klick, pil-höger eller return, kommer texten för ämnet att visas. Link-taggen ska följas av ämnet, och kan även följas av ett bokmärke och hjälpfilen som innehåller ämnet, separerade med komma-tecken. En rad som innehåller en länk kommer att markeras med en pil.

```
'some text' <link> 'topic'[, 'bookmark'][, 'helpfile']
```

I <link> kan man även ange en URL till en .html eller .pdf fil. I så fall startas en webbläsare med den angivna URL'en. URL'en får inte innehålla några av de URL-symboler som kan konfigureras i WebBrowserConfig objektet.

Exempel

```
Link to first engine <link> show engine, first_engine
Link to datasheet <link> http://platon:8080/pwr/fotocell.html
```

<index>

<index> taggen är en speciell länk som visar innehållet i hjälpfilen, dvs en lista på alla ämnen i alfabetisk ordning.

```
'some text' <index>
```

<h1>

<h1> taggen visar en rad som en rubrik med större text-storlek. Taggen ska placeras i början på raden. En rubrik-rad kan inte innehålla några länkar.

Exempel

```
<h1>This is a h1 header
```

<h2>

<h2> taggen visar en rad som en rubrik med fet text omgiven av gråa linjer. Taggen ska placeras i början på raden. En rubrik-rad kan inte innehålla några länkar.

 taggen visar en rad med fet text. Taggen ska placeras i början på raden.

Exempel

```
<b>This is a bold line
```

<t>

<t> taggen gör det möjligt att skriva kolumner. Endas tre kolumner (två <t> taggar) tillåts.

Exempel

```
Col1 <t> Col2 <t> Col3
```

<include>

Inkluderar en annan hjälp-fil. <include> taggen ska inte placeras i ett ämne.

```
<include> 'filename'
```

<i>

<i> skriver ut en rad utan att ta hänsyn till eventuella taggar i raden. Ska placeras i början på raden.

Exempel

```
<topic> helpfile_example  
Start and stop of engines.
```

```
Engine 1 <link> helpfile_example, bm_engine_1  
Engine 2 <link> helpfile_example, bm_engine_2  
Characteristics <link> helpfile_example, bm_char
```

```
<h1>Engine 1 <bookmark> bm_engine_1  
Start engine one by pressing the start button.  
Stop engine one by pressing the stop button.
```

```
<h1>Engine 2 <bookmark> bm_engine_2  
Start engine two by pressing the start button.  
Stop engine two by pressing the stop button.
```

```
<h2>Characteristics <bookmark> bm_char
```

```
<b><t>Engine1 <t>Engine2  
Max speed <t> 3200 <t> 5400  
Max current <t> 130 <t> 120  
</topic>
```

Generera hjälpfil från wb_load-filer

Med programmet co_convert kan man generera en xtt hjälpfil från en wb_load-fil

Hjälpfiler för xtt genereras med kommandot

```
co_convert -xv -d $pwrp_exe/ $pwrp_db/userclasses.wb_load
```

Kommandot genererar en hjälpfil \$pwrp_exe/'volymsnamn'_xtthelp.dat som man lämpligt-vis lägger en länk till i projektets xtt-hjälpfil \$pwrp_exe/xtt_help.dat:

Exempel för klassvolyman vhx2r:

```
<topic> index
...
<b>Använder klasser<link>cvolvhx2r,"",$pwrp_exe/cvolvhx2r_xtthelp.dat
</topic>
...
<include> $pwrp_exe/cvolvhx2r_xtthelp.dat
```

Generera html-filer från hjälpfil

co_convert kan även konvertera en xtt-hjälpfil till html. Detta gör att knappar i en process-bild som använder help-kommandot även fungerar när bilden tas upp på webben. Man kan även lägga en länk i menyn till html-filerna mha ett WebLink objekt.

Kommandot

```
co_convert -d $pwrp_web -t $pwrp_exe/xtt_help.dat
```

konverterar filen \$pwrp_exe/xtt_help.dat och alla filer som den inkluderar. Html-filerna läggs på \$pwrp_web. Kommandot exekveras av *Generate web* i Ge-editorn.

Popup meny

Om man klickar med MB3 på ett objekt öppnas en popup-menyn. Meny-alternativen varierar beroende på objekts-typ och konfigurerings. Här visas exempel på de vanligaste meny-alternativen.

Meny-alternativ	Beskrivning
Graph	Visar process-bild för objektet
Trend	Visar trendkurva för objektet
Help	Visar hjälptext för objektet
Open Object	Öppnar attribut-editorn som visar värdet på de olika attributen i objektet.
Open Plc	Öppnar plc-fönstret där objektet förekommer.
Help Class	Visar hjälptext för klassen som objektet tillhör.
Crossreferens	Visar var objektet refereras i plc-koden.
DataSheet	Visar datablad för objektet
Collect	Lägger in objektet i xtt's samlings-bild.

Konfigurerings

Vissa meny-entryn i popup-menyn kräver en viss konfigurerings. För varje entry finns dels en filter-funktion som avgör om ett entry ska visas eller ej, samt en metod som anropas när ett entry har aktiverats. Här beskrivs de metoder som är generella för alla klasser. Det finns även metoder som är dedicerade till objekt av en viss klass. Metoderna kan fungera även på användar-klasser om de innehåller de attribut som de olika metoderna använder (DefGraph, DefTrend, HelpTopic och DataSheet).

Graph

Öppnar en Ge-graf. Metoden letar efter attributet 'DefGraph' i objektet eller i något objekt som ligger ovanför objektet i hierarkin. DefGraph attributet ska innehålla ett XttGraph objekt som öppnar Ge-grafen.

DefGraph-attributet ska vara av typen pwr_tObjid, och finns bl a i \$PlantHier och signal-objekt.

Help

Visar en xtt hjälp-text. Metoden letar efter attributet 'HelpTopic' i objektet och visar hjälptexten för detta ämne.

HelpTopic-attributet ska vara av typen pwr_tString40 och finns bl a i \$Planthier och signal-objekt.

Trend

Öppnar en trendkurva. Metoden letar efter attributet 'DefTrend' i objektet. DefTrend attributet ska innehålla ett DsTrend eller PlotGroup objekt.

DefTrend-attributet ska vara av typen pwr_tObjid och finns bl a i \$PlantHier och signal-objekt.

DataSheet

Startar en web-browser som visar att datablad för objektet. Metoden letar efter attributet 'DataSheet' och öppnar den URL som attributet innehåller. URL'en kan innehålla symboler definierade i ett WebBrowserConfig-objekt.

DataSheet-attributet ska vara av typen pwr_tURL och finns bl a i \$PlantHier och signal-objekt.

Help Class

Metoden fungera automatiskt för bas och systemklasser. Vill man att de ska fungera även på användarklasser ska man genererar xtt hjälp-filer från wb_load-filen och göra denna tillgänglig för help-funktionen i xtt. Se avsnittet *Hjälp*.

Open Plc

Open plc kräver att flw-filer är distribuerade (flw-filer skapas på \$pwrp_load när ett plc-fönster sparas).

Crossreferences

Korsreferenser kräver att rtt-filer är skapade och distribuerade (skapas med 'create rttfiles' i Utilities).

Konfigurering

Operatörsmod

En operatörsplats konfigureras med ett OpPlace-objekt och ett User-objekt. För att konfigurera processbilder används XtGraph-objekt.

Underhållsmod

I underhålls-mod konfigureras xtt med ett RttConfig-objekt, med namnet RttConfig. Objektet ska ligga under nod-objektet.

Symbolfil

I RttConfig-objektet kan man ange en symbol-fil. Denna exekveras när xtt startas, och man kan t ex modifiera menyn eller definiera symboler där.

Här följer några användbara kommandon.

Genväg till en databas-hieraki

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

Genväg till en ge-bild

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

Skapa meny-alternativ i xtt menyn

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

Ta bort meny-alternativ ur standard-menyn

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

Kommandon

Kommando-tolken i xtt fungerar som en knutpunkt i xtt, eftersom det är genom kommandon de flesta funktioner i xtt aktiveras.

Kommandon kan matas in interaktivt från kommandoraden som öppnas med *Functions/Command* (Ctrl/B) i meny. Det finns en recall-funktion så att men med PilUpp och PilNer kan välja mellan tidigare inmatade kommandon.

Kommandon kan även exekveras från tryckknappar i Ge-bilder, de konfigurerbara knapparna i operatörs-fönstret och från funktionstangenter.

Kommandona finns beskrivna i *Handbok för operatörer*.

Script

Rtt-script är ett sätt att programmera rtt-kommandon. Scripthanteraren ger dessutom möjlighet till att göra beräkningar, utföra villkors-satser, loop-satser, deklarerara variabler och funktioner. Mha rtt-script kan man bl a navigera i rtt, bygga menyer, leta efter objekt och hämta och tilldela attribut i realtidsdatabasen.

Ett script startas med '@' följt av scriptfils namnet och eventuella argument. Man kan även starta ett script från en menu-entry av typen command, eller från en tryckknapp av typen command.

Exempel

```
pwr_rtt> @my_script
```

Läs mer om script i *Handbok för operatörer*.

Appendix A

Exempel på loggning

Exempel på loggning med xtt (type=EVENT).

```
RTT LOGGING STARTED AT 23-JUL-1992 10:58:40.14
Parameter: VHX-IL-KRAN-DI_ORDERIN.ActualValue
Parameter: VHX-IL-KRAN-DI_SVARUT.ActualValue
Parameter: VHX-IL-KRAN-DO_ORDERUT.ActualValue
Parameter: VHX-IL-KRAN-DO_SVARIN.ActualValue
Parameter: VHX-IL-KRAN-SENAST_ORDER.ActualValue
23-JUL-1992 10:58:40.15 VHX-IL-KRAN-DI_ORDERIN.ActualValue 0
23-JUL-1992 10:58:40.15 VHX-IL-KRAN-DI_SVARUT.ActualValue 0
23-JUL-1992 10:58:40.15 VHX-IL-KRAN-DO_ORDERUT.ActualValue 0
23-JUL-1992 10:58:41.54 VHX-IL-KRAN-DO_ORDERUT.ActualValue 1
23-JUL-1992 10:58:41.54 VHX-IL-KRAN-DO_SVARIN.ActualValue 0
23-JUL-1992 10:58:41.94 VHX-IL-KRAN-DI_SVARUT.ActualValue 1
23-JUL-1992 10:58:44.04 VHX-IL-KRAN-DI_SVARUT.ActualValue 1
23-JUL-1992 10:58:44.34 VHX-IL-KRAN-DI_ORDERIN.ActualValue 1
23-JUL-1992 10:58:45.44 VHX-IL-KRAN-DO_ORDERUT.ActualValue 0
23-JUL-1992 10:58:45.44 VHX-IL-KRAN-DO_SVARIN.ActualValue 1
23-JUL-1992 10:58:45.94 VHX-IL-KRAN-DI_ORDERIN.ActualValue 0
23-JUL-1992 10:58:45.94 VHX-IL-KRAN-DI_SVARUT.ActualValue 0
23-JUL-1992 10:58:54.94 VHX-IL-KRAN-DO_ORDERUT.ActualValue 1
23-JUL-1992 10:58:54.94 VHX-IL-KRAN-DO_SVARIN.ActualValue 0
23-JUL-1992 10:58:55.24 VHX-IL-KRAN-DI_SVARUT.ActualValue 1
23-JUL-1992 10:58:55.74 VHX-IL-KRAN-DI_ORDERIN.ActualValue 1
23-JUL-1992 10:58:56.74 VHX-IL-KRAN-DO_ORDERUT.ActualValue 0
23-JUL-1992 10:58:56.74 VHX-IL-KRAN-DO_SVARIN.ActualValue 1
23-JUL-1992 10:58:57.34 VHX-IL-KRAN-DI_ORDERIN.ActualValue 0
23-JUL-1992 10:58:57.84 VHX-IL-KRAN-DI_SVARUT.ActualValue 1
```