

BESKRIVELSE AF NYHEDER AUTOMATION VER. 22



Dette dokument beskriver nye funktioner i Automation version 22.

Der er i dokumentet vist en * for nyheder, som er kommet efter brugermøderne i oktober 2018, dvs funktioner, som også er i version 21.

Nyheder mærket med ** er kommet til efter brugermøderne i 2019.

Sidst rettet: Juli 2020

INDHOLD

| | | |
|-------|-------------------------------------------------------------------------------------|----|
| 1 | Nye logoer | 7 |
| 2 | Forberedelser til ny komponentdatabase | 8 |
| 3 | Ændringer på symboler og tilsl.punkter | 9 |
| 3.1 | Symboltypen kan ikke ændres i projektet | 9 |
| 3.1.1 | Slutte/brydekors kan fravælges for Relæ-symboler..... | 9 |
| 3.2 | Man kan ikke ændre tilstand på komponentsymboler | 10 |
| 3.3 | Symboltype 2 udgår | 10 |
| 3.4 | En enklere definition af klemmer | 10 |
| 3.5 | Klemmer er ALTID gennemgangsklemmer..... | 10 |
| 3.5.1 | Konvertering af tilslutningspunkter | 11 |
| 3.5.2 | Syntaksændring | 11 |
| 3.6 | Oversigt over ændringer af tilslutningspunkter | 12 |
| 3.7 | Visning af parkede streger | 15 |
| 3.8 | Symboleditor tjekker for overholdelse af regler | 15 |
| 3.9 | Udvidelse på designcheck | 15 |
| 4 | Få din database i topform | 16 |
| 4.1.1 | Sådan kan du gøre | 16 |
| 4.2 | Komponentguiden..... | 17 |
| 4.2.1 | Advarsel for manglende felter i databaseopsætning..... | 17 |
| 4.2.2 | Kodebogstaver fra 81346-2 er også i Komponentguiden..... | 17 |
| 4.2.3 | Valg af diagramsymboler vises med alternativ | 17 |
| 4.2.4 | Ændringer på tilslutningspunkter | 18 |
| 4.2.5 | Mekaniske symboler..... | 18 |
| 4.2.6 | Man kan flytte komponenter fra en varegruppe til en anden | 18 |
| 4.2.7 | Justerbar hjælpekolonne | 18 |
| 4.2.8 | Automec og priser er ikke længere mappet ved nye installationer | 18 |
| 5 | Anvendelse af undernavne - præcisering..... | 19 |
| 5.1 | Navngivning af kort i en sammensat plc..... | 19 |
| 6 | Tilbehør – hvordan arbejder jeg med det? | 21 |
| 6.1 | Formål med at lave om på arbejdsgang og -metode | 21 |
| 6.1.1 | Bedre overblik | 21 |
| 6.1.2 | Elektrisk tilbehør | 21 |
| 6.1.3 | Mekanisk tilbehør | 21 |
| 6.1.4 | Fast tilbehør | 22 |
| 6.1.5 | Kopi af komponent med mekanisk tilbehør | 22 |
| 6.1.6 | Kopi og referencebetegnelser | 22 |
| 6.2 | Man kan konfigurere Tilbehørsfanen | 23 |
| 6.3 | Evt billeder af tilbehør vises på fanen | 23 |
| 6.4 | Supportsymboler og tilbehør | 24 |
| 6.4.1 | Supportsymboler i eksisterende projekter | 24 |
| 6.4.2 | Supportsymboler bliver ignoreret i Komponentmenuen fra ver. 21 | 24 |
| 7 | Klemmer i Komponentgruppe 0 og tilbehør * | 25 |
| 7.1.1 | Lidt historie | 25 |
| 7.1.2 | Hvis du har klemmer med tilbehør og de ikke er placeret mekanisk | 25 |
| 7.1.3 | Hvis du har klemmer med tilbehør, og de er placeret mekanisk | 26 |
| 7.1.4 | Hvordan kan jeg rette op, hvis der er fejl i styklisten pga komponentgruppe 0 | 26 |

| | | |
|-------|------------------------------------------------------------------------------|----|
| 8 | Lidt smårettelser til Komponentmenuen | 27 |
| 8.1.1 | Grene lukkes efterhånden..... | 27 |
| 8.1.2 | Vinduet forbliver lukket | 27 |
| 8.1.3 | Popup på Muligt tilbehør | 27 |
| 8.1.4 | Indstilling af kolonner i symboldelen | 27 |
| 8.2 | Placering af mekaniske symboler | 28 |
| 8.3 | Ændring af genvej | 28 |
| 9 | Tilknyttede datablade kan zippes | 29 |
| 10 | Noget om lister – import og eksport..... | 30 |
| 10.1 | Indlæs styk- og komponentlister vha formatfil | 30 |
| 10.2 | Tildel data fra en indlæst komponentliste til projektet | 30 |
| 10.3 | Listeeksport til Excel kan skrive til en eksisterende skabelon..... | 31 |
| 10.4 | Indholdsfortegnelsen kan vise sider efter en valgt dato | 32 |
| 10.5 | Man kan hoppe til specifikke streger fra en styk- eller komponentliste | 32 |
| 10.6 | Antal gentagelser og antal erstatninger i Listeopsætning er udvidet..... | 33 |
| 10.7 | Nyt datafelt på Styk- og komponentlister | 33 |
| 10.8 | PLC-liste skal inkludere plc'ens undernavn i sin sortering * | 34 |
| 10.9 | Eksport til Cablemanager | 34 |
| 10.10 | PDF-eksport via kommandolinje | 35 |
| 10.11 | Navnelisten i detailtegninger er med hyperlink..... | 35 |
| 11 | Monteringsassistenten | 36 |
| 11.1 | To brugere kan montere samme tavle | 36 |
| 11.2 | Forbindelsen kan vælges fra listen eller fra diagrammet | 36 |
| 11.3 | Alle oprettede sagsnumre på projektet vises | 36 |
| 11.4 | Oversigtsvinduet har fået en Monteret-knap | 36 |
| 11.5 | Ledningsnumre i ledningslisten | 37 |
| 11.6 | Eksport til Excel..... | 37 |
| 11.7 | Mulighed for at ændre status på alle på en gang..... | 37 |
| 12 | Udvidelser til Objektlisteren (F7) | 38 |
| 12.1 | Objektlister – ny standardopsætning * | 38 |
| 12.2 | Objektlisteren kan nu have filter i alle kolonner | 38 |
| 12.3 | Objektlister - alle symboltyper kan vises på symbolfanen * | 38 |
| 12.4 | Objektlisteren kan også vise tilbehør | 39 |
| 12.5 | Objektlister kan vise stregart | 39 |
| 13 | Udvidelser på symboldatafelter | 40 |
| 13.1 | Nyt fast symboldatafelt | 40 |
| 13.2 | Symboldatafelter oprettes kun i projektet | 40 |
| 13.3 | Symboldatafelter med formular | 40 |
| 13.4 | Ny formel editor for streg og symboldatafelter..... | 41 |
| 13.5 | Værktøjet Rediger symbolers datafelter | 43 |
| 14 | Småtingsafdelingen | 45 |
| 14.1 | Valg af referencebetegnelser med Som-siden og Slet-knap* | 45 |
| 14.2 | Indstillinger ved Indsæt ref.ramme huskes * | 45 |
| 14.3 | Eksport til DWG og DXF | 45 |
| 14.4 | Kodebogstaver fra 81346-2 findes som lister i programmet | 46 |
| 14.5 | Forbedret kontaktspejl | 46 |
| 14.6 | Udvidelse af funktionen Læg på linje * | 47 |
| 14.7 | Design af symbol med Symbolgenerator | 47 |



| | | |
|---------|--------------------------------------------------------------------|----|
| 14.8 | Rotér et objekt med 10° * | 48 |
| 14.9 | Ikonet Ny og Filer Ny har nu samme funktion * | 48 |
| 14.10 | Antal senest åbnede filer kan indstilles * | 48 |
| 14.11 | Standardkabelmængden kan nu indstilles * | 49 |
| 14.12 | Streger med varedata | 49 |
| 14.13 | Udskift symbol 'Ignorerer symbolsti' | 49 |
| 14.14 | Hurtig adgang til farveopsætning i programmet | 50 |
| 14.15 | Højdekote kan overføres vha 'Bjælken' | 50 |
| 14.16 | Tekstegenskaber | 50 |
| 14.17 | Henvisninger – nu med eller uden pil | 51 |
| 14.18 | Brugerfladen er nu også på fransk | 51 |
| 14.18.1 | Tegningshoveder findes også på kroatisk | 51 |
| 14.19 | Direkte adgang til varedata i højreklik | 52 |
| 14.20 | Dobbeltklik i tegningshovedet åbner side/projektdata direkte | 52 |
| 14.21 | Oprydning i menupunkter og funktioner | 52 |
| 15 | Konvertering af tekster – fra gamle tekstjusteringer til nye | 53 |
| 15.1 | Indlæsning af listesider | 53 |
| 16 | Formula editor for line and symbol data fields | 54 |
| 16.1 | General | 54 |
| 16.2 | Operators | 55 |
| 16.2.1 | Arithmetic operators: | 55 |
| 16.2.2 | Boolean operators: | 56 |
| 16.2.3 | Comparison operators: | 56 |
| 16.2.4 | Equality operators: | 56 |
| 16.2.5 | String operator: | 57 |
| 16.2.6 | Variable: | 57 |
| 16.3 | Functions: | 57 |
| 16.3.1 | Function: DATAFIELD | 57 |
| 16.3.2 | Function: VAL | 57 |
| 16.3.3 | Function: ISVAL | 58 |
| 16.3.4 | Function: FORMAT | 58 |
| 2. | An optional width specifier, [width]. | 59 |
| 3. | An optional precision specifier, [".." prec]. | 59 |
| 4. | The conversion type character, type | 59 |
| 16.3.5 | Function: EXP | 60 |
| 16.3.6 | Function: POW | 60 |
| 16.3.7 | Function: SQRT | 60 |
| 16.3.8 | Function: SIN | 61 |
| 16.3.9 | Function: COS | 61 |
| 16.3.10 | Function: TAN | 61 |
| 16.3.11 | Function: ASIN | 62 |
| 16.3.12 | Function: ACOS | 62 |
| 16.3.13 | Function: ATAN | 63 |
| 16.3.14 | Function: ABS | 63 |
| 16.3.15 | Function: LN | 63 |
| 16.3.16 | Function: LOG | 63 |
| 16.3.17 | Function: TRUNC | 64 |
| 16.3.18 | Function: ROUND | 64 |
| 16.3.19 | Function: IF | 64 |

1 NYE LOGOER

Svært at skjule, men programmerne har fået nye logoer:

Automation



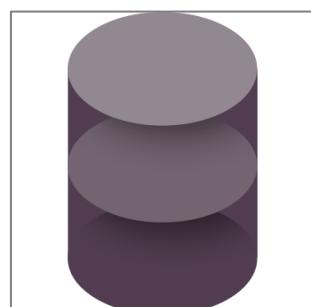
Automation Service



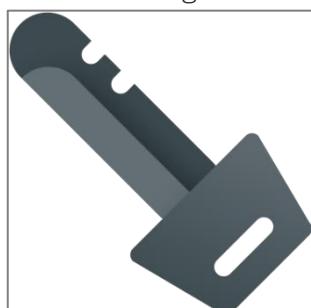
Automation Viewer



Database



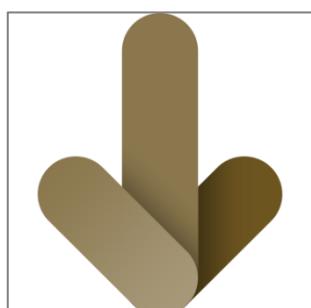
License Manager



License Manager Config



Installer



Vi er i gang med at implementere logoerne i programmer, på hjemmeside, i annoncer mm. Vore andre programmer skifter også logo, og selve firmalogo bliver også ændret en lille smule. Læs og se meget mere på www.pcschematic.dk.

2 FORBEREDELSE TIL NY KOMPONENTDATABASE

Vi bekendtgjorde på brugermøderne i oktober og november 2019, at der kommer en ny struktur på vores database. Og i den forbindelse rydder vi op i mange ting.

Generelt, så kommer man til at arbejde meget mere komponentorienteret, når der er valgt dele fra databasen. Og dette medfører i første omgang ændringer i symboler og projekter, som beskrives nedenfor.

Ændringerne har i denne version følgende overskrifter, som uddybes i afsnittene nedenfor:

- Der indføres begrænsning på, hvad der kan ændres på en komponent i projektet
- Symboltype 2 udgår
Der ryddes op i funktioner og typer af tilslutningspunkter

i

Når du læser et eksisterende projekt ind i ver. 22 og/eller henter en komponent i databasen, konverteres tilslutningspunkterne i projektet efter reglerne, som beskrives i dette kapitel.

Den nemmeste måde at se ændringerne på er ved at lave en ny installation – i stedet for en opdatering. Så kan du senere lave en opdatering.

Komponentportalen er ikke klar til frigivelse, endnu, men når dette sker, kommer der flere ændringer, som vil blive beskrevet på det tidspunkt.



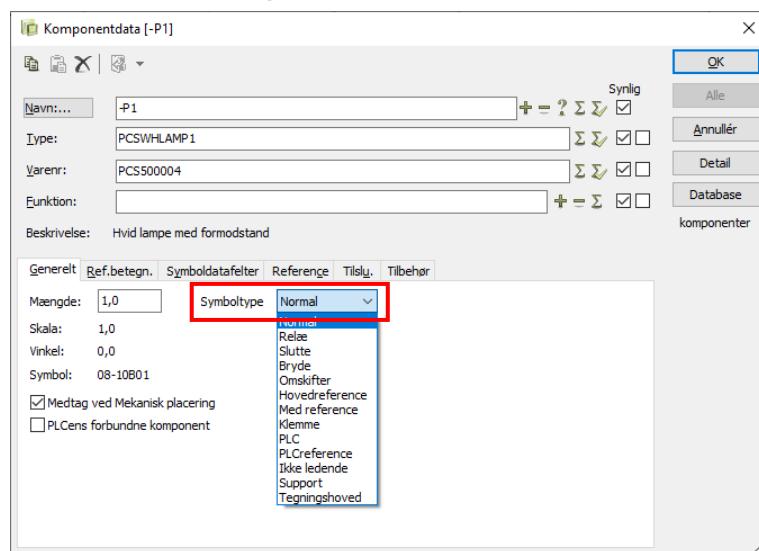
3 ÆNDRINGER PÅ SYMBOLER OG TILSL.PUNKTER

3.1 Symboltypen kan ikke ændres i projektet

Når man placerer et symbol, har man altid kunne ændre symboltype.

Dette kan man ikke fra ver 22, da det ikke giver mening.

Det har primært været nødvendigt, hvis man ønskede forskellige komponenters tilslutninger på klemrækken: Så ville man (tidligere) have lavet symbolet om til en klemme, og valgt en passende symboltype 2 og derefter skulle man lave om på symbolets tilslutningspunkter. Efter lidt diffuse regler.



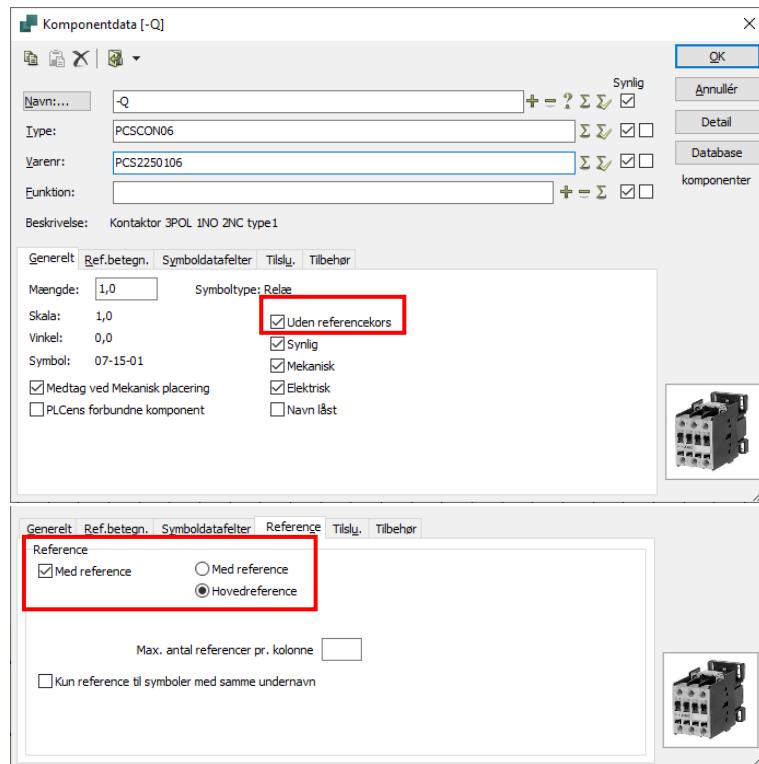
Dette laver vi om på nu!

I det følgende kan du læse, hvad dette betyder, når det medfører ændringer for en symboltype.

3.1.1 Slutte/brydekors kan fravælges for Relæ-symboler

I forbindelse med, at det ikke længere er muligt at skifte symboltype i projekterne, kan man nu på et Relæsymbol vælge at vise det uden referencekors.

Hvis du vælger dette, kommer fanen Reference på symbolet frem. På den måde kan du vælge, hvordan referencerne skal se ud uden referencekorset.ⁱⁱ



3.2 Man kan ikke ændre tilstand på komponentsymboler

Når en komponent er oprettet i databasen, må man formode, at den er oprettet korrekt. Det betyder, at har man fx valgt en komponent, hvor dets diagramsymbol er lavet med tilstande, kan man ikke ændre symbolets tilstand i diagrammet.

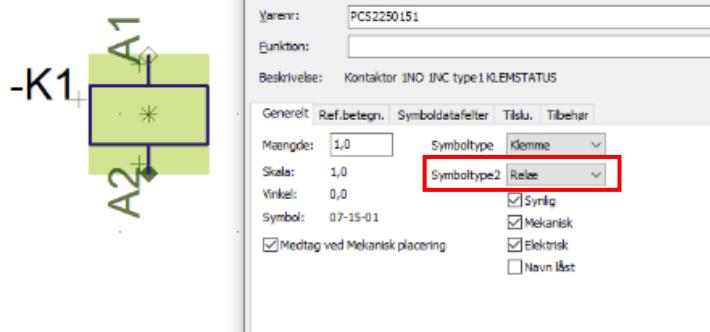


Dog, hvis symbolet er i tilstand 0, kan man godt. Oftest er symbolet valgt i tilstand 0, fordi et evt tilbehør angiver det egentlige funktion, og derfor skal man kunne skifte.

3.3 Symboltype 2 udgår

I forbindelse med oprydning og optimering af symboltyper, udgår symboltype 2.ⁱⁱⁱ

Symboltype 2 har alene eksisteret for at gøre det muligt at få tilslutninger på specielle komponenter i klemlisten. Dette løses ved, at alle tilslutningspunkter kan komme på klemlisten, se næste afsnit.



3.4 En enklere definition af klemmer

Indtil nu, har vi haft meget forskellige definitioner af klemmer, og vi har brugt symboltype 2 til dette. Det har handlet om, at man har ønsket – forståeligt – at få forbindelser på en liste, sådan at man kan bruge listen til at montere ud fra.

3.5 Klemmer er ALTID gennemgangsklemmer

Fremover gælder følgende:

- Komponenter med symboltypen Klemme, som vi herefter kalder Klemmer, er ALTID gennemgangsklemmer, dvs at der er det samme potentielle på begge sider af komponenten!
- Alle tilslutninger har samme navn.
- Der er minimum én tilslutning på hhv side 1 og side 2 af komponenten.
- Alle tilslutninger er standard med på klemlisten

At alle tilslutninger på en klemme har samme navn, betyder også at har du klemmer, hvor tilslutningerne allerede er navngivet, så behandler vi dem i programmet som normale komponenter og ikke som klemmer!

Det kommer også til at betyde, at de gamle typer som fx klem-plc, klem-relæ mm udgår, og bare bliver erstattet af normale plc- og relæ-symboler. Se mere senere om, hvordan du får deres forbindelser på klemlisten (det bliver nemmere 😊)

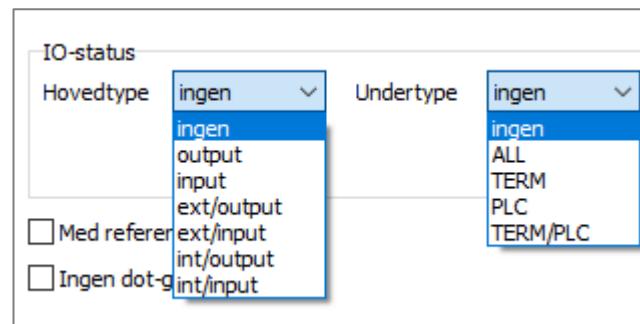


3.5.1 Konvertering af tilslutningspunkter

Som en konsekvens af, at der ikke længere findes Symboltype 2, er mange kombinationer af tilslutningspunkters hoved- og undertype irrelevante.

Nedenfor kan du se, hvad de forskellige kombinationer konverteres til.^{iv v}

De gamle muligheder – mange muligheder, hvoraf mange enten ikke blev brugt eller bare ikke blev forstået.



3.5.2 Syntaksændring

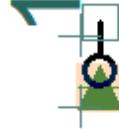
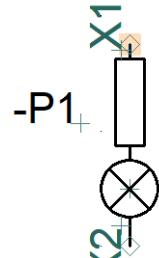
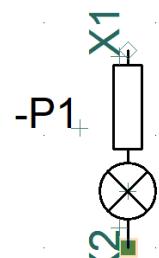
Symboltyperne kan styres vha databasen, og der ændres i syntaksen på følgende måde:

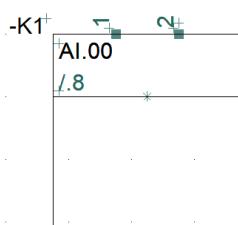
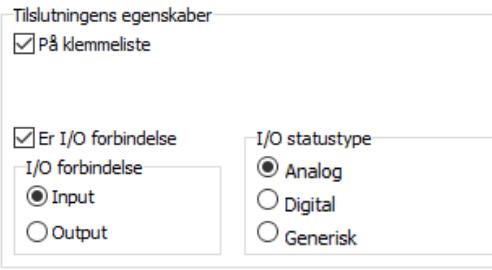
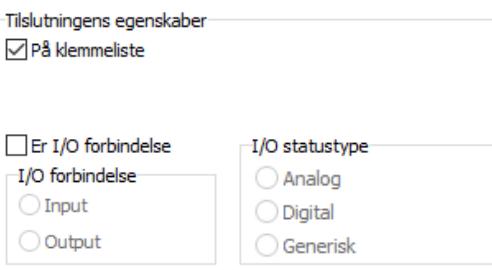
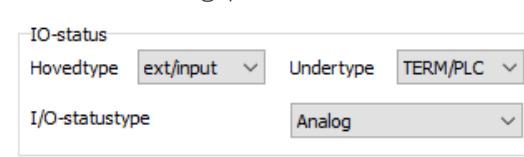
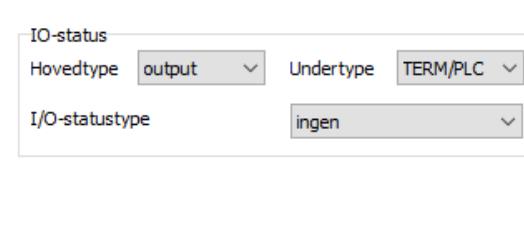
| Gammel syntaks | Ny syntaks |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Syntax=T,O (Terminal, Open) | Symboltype1 = Open |
| Syntax=T,C (Terminal, Closed) | Symboltype1 = Closed |
| Syntax=T,S (Terminal, Switch) | Symboltype1 = Switch |
| Syntax=T,P (Terminal, PLC) | Symboltype1 = PLC |
| Syntax=T,N (Terminal, Normal) | Symboltype1 = Normal |
| Syntax=T,R (Terminal, Relay) | Symboltype2 = Relay |
| Syntax *M (Symboltype = Hovedreference) | Syntaks ændres ikke, men fortolkes som Symboltype1 = Normal, med Reference sat til Hovedreference |
| Syntax *W (Symboltype = MedReference) ^{vi} | Syntaks ændres ikke, men fortolkes som Symboltype1 = Normal, med Reference sat til Med reference |
| Findes ej | Syntax=/I:IJ – tilslutningspunkt for laske. På side 1, dvs intern side. |
| Findes ej | Syntax=/I:EJ – tilslutningspunkt for laske. På side 2, dvs ekstern side. |

3.6 Oversigt over ændringer af tilslutningspunkter

| Før (til og med ver. 21) | Nu (fra ver. 22) |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>IO-status</p> <p>Hovedtype: input Undertype: TERM</p> <p>I klemlisten er der datafelt for T.pkt. indgangsside</p> | <p>Den interne side af klemmen hedder nu Side 1, og den kommer med på klemlisten, hvis der er flueben i 'På klemmeliste'.</p> <p>Sidevalget indikeres som tidligere.</p> <p>Klemliste indikeres med firkant.</p>  <p>Tilslutningens egenskaber</p> <p><input checked="" type="checkbox"/> På klemliste <input type="checkbox"/> Er laske</p> <p>Klemmeside <input checked="" type="radio"/> Side 1 (int) <input type="radio"/> Side 2 (ext)</p> |
| <p>IO-status</p> <p>Hovedtype: output Undertype: TERM</p> <p>I klemlisten er der datafelt for T.pkt. udgangsside</p> | <p>Den eksterne side af klemmen hedder nu Side 2, og den kommer med på klemlisten, hvis der er flueben i 'På klemmeliste'.</p> <p>Sidevalget indikeres som tidligere.</p> <p>Klemliste indikeres med firkant.</p>  <p>Tilslutningens egenskaber</p> <p><input checked="" type="checkbox"/> På klemliste <input type="checkbox"/> Er laske</p> <p>Klemmeside <input type="radio"/> Side 1 (int) <input checked="" type="radio"/> Side 2 (ext)</p> |
| <p>Her skulle man tidligere skifte symboltype til Normal, eller ændre tilslutningpunktet til 'Ingen/Ingen'.</p> <p>Der blev intet vist i diagrammet.</p> | <p>Hvis man IKKE ønsker tilslutningen på klemlisten:</p> <p>Side 1, som ikke kommer på klemlisten er vist med den almindelige t.pkt. ruder.</p> <p>Sidevalget beholdes.</p>  <p>Tilslutningens egenskaber</p> <p><input type="checkbox"/> På klemliste <input type="checkbox"/> Er laske</p> <p>Klemmeside <input checked="" type="radio"/> Side 1 (int) <input type="radio"/> Side 2 (ext)</p> |
| <p>Her skulle man tidligere skifte symboltype til Normal, eller ændre tilslutningpunktet til 'Ingen/Ingen'.</p> <p>Der blev intet vist i diagrammet.</p> | <p>Hvis man IKKE ønsker tilslutningen på klemlisten:</p> <p>Side 2, som ikke kommer på klemlisten er vist med den almindelige t.pkt. ruder.</p> <p>Sidevalget beholdes.</p>  <p>Tilslutningens egenskaber</p> <p><input type="checkbox"/> På klemliste <input type="checkbox"/> Er laske</p> <p>Klemmeside <input type="radio"/> Side 1 (int) <input checked="" type="radio"/> Side 2 (ext)</p> |



| Før (til og med ver. 21) | Nu (fra ver. 22) | | | | | | |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|------------|----------------------------------------------|-----------------------------------------------|--|-----------------------------------------------|
| Ej muligt. | <p>Hvis der er tale om en klemme, kan tilslutningspunktet have status Laske. Dette indikeres med en lille trekant.</p> <p>Sidevalget beholdes.</p> <p>Det er kun lovligt at forbinde en laskestreg til et laske-tilslutningspunkt.</p>  <p>Tilslutningens egenskaber</p> <table border="1"> <tr> <td><input type="checkbox"/> På klemmeliste</td> <td>Klemmeside</td> </tr> <tr> <td><input checked="" type="checkbox"/> Er laske</td> <td><input checked="" type="radio"/> Side 1 (int)</td> </tr> <tr> <td></td> <td><input type="radio"/> Side 2 (ext)</td> </tr> </table> | <input type="checkbox"/> På klemmeliste | Klemmeside | <input checked="" type="checkbox"/> Er laske | <input checked="" type="radio"/> Side 1 (int) | | <input type="radio"/> Side 2 (ext) |
| <input type="checkbox"/> På klemmeliste | Klemmeside | | | | | | |
| <input checked="" type="checkbox"/> Er laske | <input checked="" type="radio"/> Side 1 (int) | | | | | | |
| | <input type="radio"/> Side 2 (ext) | | | | | | |
| Ej muligt. | <p>Hvis der er tale om en klemme, kan tilslutningspunktet have status Laske. Dette indikeres med en lille trekant.</p> <p>Sidevalget beholdes.</p> <p>Det er kun lovligt at forbinde en laskestreg til et laske-tilslutningspunkt.</p>  <p>Tilslutningens egenskaber</p> <table border="1"> <tr> <td><input type="checkbox"/> På klemmeliste</td> <td>Klemmeside</td> </tr> <tr> <td><input checked="" type="checkbox"/> Er laske</td> <td><input type="radio"/> Side 1 (int)</td> </tr> <tr> <td></td> <td><input checked="" type="radio"/> Side 2 (ext)</td> </tr> </table> | <input type="checkbox"/> På klemmeliste | Klemmeside | <input checked="" type="checkbox"/> Er laske | <input type="radio"/> Side 1 (int) | | <input checked="" type="radio"/> Side 2 (ext) |
| <input type="checkbox"/> På klemmeliste | Klemmeside | | | | | | |
| <input checked="" type="checkbox"/> Er laske | <input type="radio"/> Side 1 (int) | | | | | | |
| | <input checked="" type="radio"/> Side 2 (ext) | | | | | | |
| Klemme og Symboltype 2 og indstilling på tilslutningspunkt | <p>For symboler af typen Normal, Relæ, Slutte, Bryde, Omskifter kan du nu vælge, hvorvidt deres tilslutninger skal på klemlisten.</p> <p>Ikke-valgte tilslutninger vises som tidligere med ruder.</p>  <p>Tilslutningens egenskaber</p> <table border="1"> <tr> <td><input type="checkbox"/> På klemmeliste</td> </tr> </table> | <input type="checkbox"/> På klemmeliste | | | | | |
| <input type="checkbox"/> På klemmeliste | | | | | | | |
| Klemme og Symboltype 2 og indstilling på tilslutningspunkt | <p>For symboler af typen Normal, Relæ, Slutte, Bryde, Omskifter kan du nu vælge, hvorvidt deres tilslutninger skal på klemlisten.</p> <p>Valgte tilslutninger vises med udfyldt firkanter, og de er pr definition på side 2, dvs den eksterne side af klemrækken.</p>  <p>Tilslutningens egenskaber</p> <table border="1"> <tr> <td><input checked="" type="checkbox"/> På klemmeliste</td> </tr> </table> | <input checked="" type="checkbox"/> På klemmeliste | | | | | |
| <input checked="" type="checkbox"/> På klemmeliste | | | | | | | |

| Før (til og med ver. 21) | Nu (fra ver. 22) |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Samme symbols tilslutningspunkter i ver. 21: | Her ses et PLC-input symbol. Hvis jeg ønsker dets tilslutninger på klemlisten, sætter jeg bare flueben i 'På klemliste'. |
| Tilslutningspunkt 1: |  |
| Hvis jeg ønsker at sætte tilslutningen på klemlisten skulle jeg gøre følgende: | Samtidigt man se og vælge, hvorvidt der er tale om en IO-forbindelse eller ej. Det er også nemt at vælge evt statustype. |
| Symboltypen skal ændres: | Tilslutning 1:  |
| Så skal tilslutningspunkt 1 indstilles: | Tilslutning 2:  |
| og derefter tilslutningspunkt 2: | vii |
|  | |
|  | |

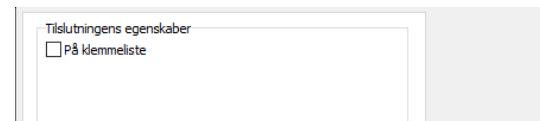
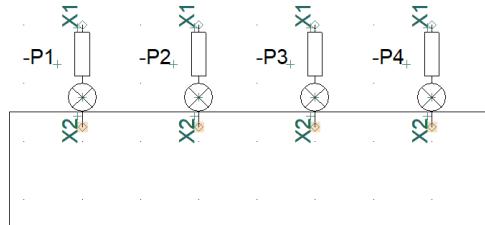


Tips:

Hvis du ønsker at markere flere tilslutninger på en gang for at sætte dem på klemmlisten:

Marker tilslutninger ved at trække musen hen over dem, og hold Ctrl-knappen nede, når du afslutter markeringen!

Og brug knappen Objektdata i Værktøjslinjen til at vælge egenskaber.

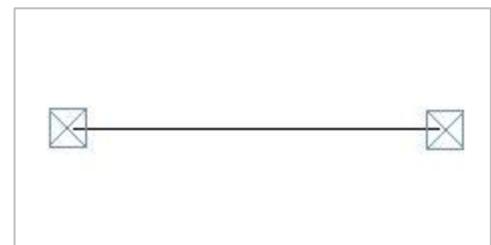


3.7 Visning af parkede streger

Vi har lavet en masse ændringer på, hvordan vi viser status på forbindelsespunkter.

Den 'gamle' måde at vise parkerede streger (ikke-forbundne, ledende streger) ligner rigtig meget en klemmes side 1.

Derfor har vi lavet en ny markering af parkerede streger. Husk, at den bliver mindre og mindre, jo mere du zoomer på siden.



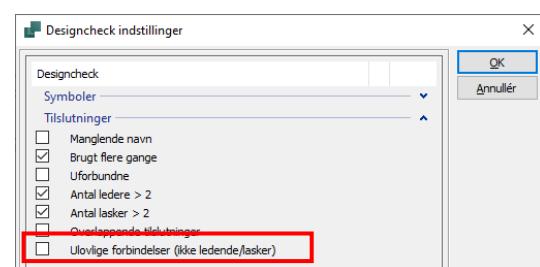
3.8 Symboleditor tjekker for overholdelse af regler

Fremover sørger symboleditor'en for, at man ikke kan oprette symboler med forkerte kombinationer af tilslutningspunkters funktionalitet.^{viii}

De gamle syntakser beholdes på dine eksisterende komponenter, men fortolkes fremover som vist i skemaet. Der er også indført to nye syntakser, og disse er også vist nedenfor.

3.9 Udvidelse på designcheck

Som følge af den nye mulighed for at forbinde lasker til dedikerede laske-tilslutninger, kan Designcheck også checke, om der er forbundet andet end lasker til et sådan tilslutningspunkt.



4 FÅ DIN DATABASE I TOPFORM

Husk, at sandheden om komponenter findes i databasen.

Det betyder også, at mange funktioner i programmet afhænger af, at din database fungerer, er indstillet korrekt, og også indeholder databasefejler for forskellige funktioner, fx Komponentguiden.

4.1.1 Sådan kan du gøre

Den database, vi leverer, indeholder alle nødvendige felter.

I selve database-programmet, kan du under Filer|Ny lave en tom database, som en kopi af vores standarddatabase.

Hvis du gør det, og giver den et fornuftigt navn, og importerer dine egne komponenter i denne, er du godt på vej.



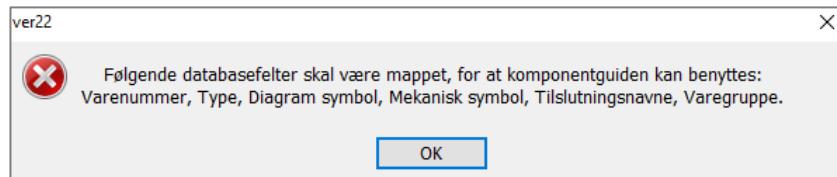
4.2 Komponentguiden

Vi forbedrer løbende på Komponentguiden og opfordrer til, at man bruger den til at oprette egne komponenter i databasen.

4.2.1 Advarsel for manglende felter i databaseopsætning

Komponentguiden kræver at databasen indeholder nogle bestemte felter. Hvis en af dem ikke er mappet (under databaseindstillinger), får du denne advarsel.

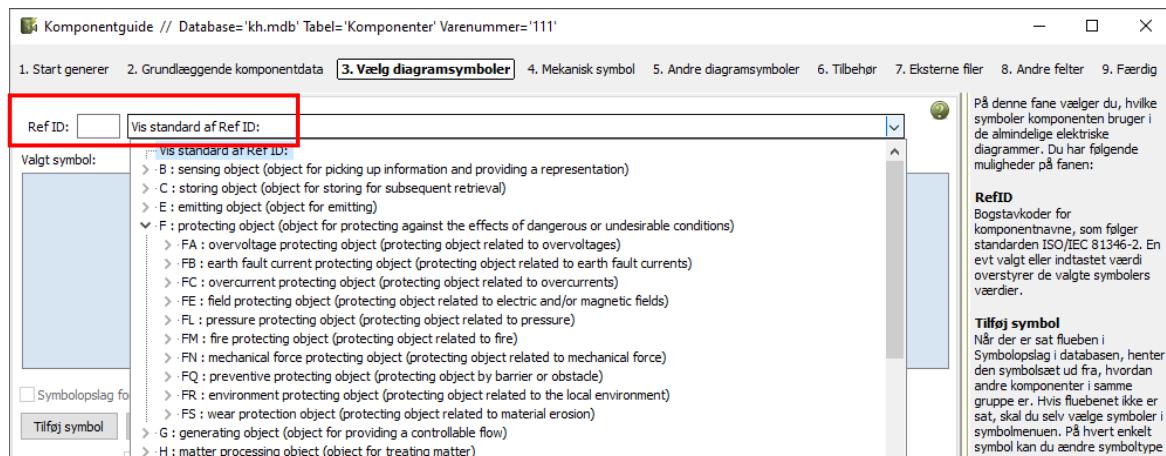
Udover disse felter,
skal du have feltet
CWCODE i databasen.



4.2.2 Kodebogstaver fra 81346-2 er også i Komponentguiden

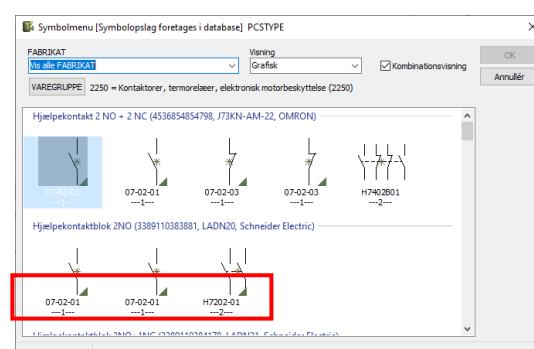
Når man opretter nye komponenter i Komponentguiden, er listen i RefID også udskiftet til den nye.

Så kan man selv vælge, om man vil have 1, 2 eller 3 kodebogstaver på sin komponent fra starten,



4.2.3 Valg af diagsymboler vises med alternativ

Der vises alternativer, når man leder efter symboler fra fanen Diagramsymboler.



4.2.4 Ændringer på tilslutningspunkter

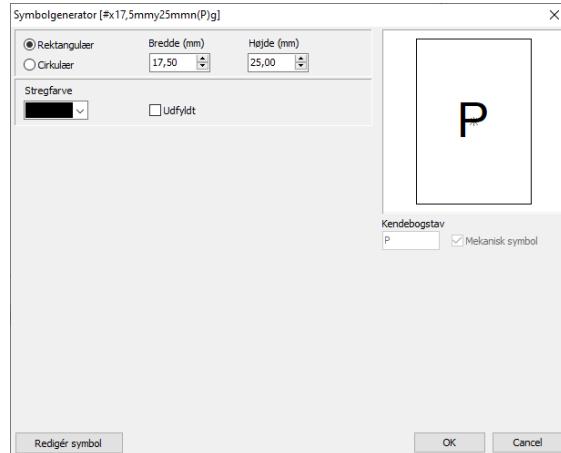
Alle ændringer på tilslutningspunkter, som er beskrevet i dette dokument, er også indført i Komponentguiden.

4.2.5 Mekaniske symboler

Når man vil vælge eller oprette mekaniske symboler til sin komponent, er Automec nu fjernet i en standardopsætning, dvs at hvis du bruger vores database, så er den væk.

Derudover er Symbolgeneratoren rettet til:

- man kan ikke længere vælge tilslutningspunkter her: men tidligere blev de bare fjernet ...
- komponentnavn hentes automatisk fra RefID
- der er allerede valgt funktionen Mekanisk symbol

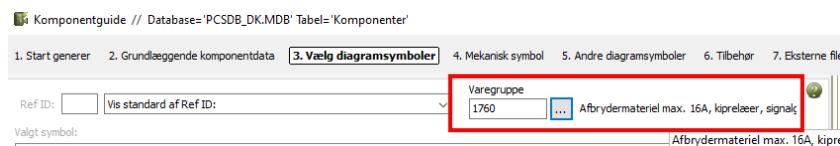


4.2.6 Man kan flytte komponenter fra en varegruppe til en anden

Hvis man har oprettet komponenter i én varegruppe (ofte blank), og ønsker at flytte dem, har man ikke tidligere kunnet gøre det vha Komponentguiden. Det kan man nu.^x

Hvis du redigerer én komponent, ændrer du varegruppen på første fane.

Hvis du redigerer flere komponenter, ændrer du varegruppen på fanen Vælg diagramsymboler, i feltet ved siden af RefID.



4.2.7 Justerbar hjælpekolonne

Hjælpekolonnen kan nu justeres i bredden. Så man kan have mere af teksten vist, når man har brug for den.

4.2.8 Automec og priser er ikke længere mappet ved nye installationer

Vi vil erstatte Automec med noget bedre i den kommende database. Vi har ikke indtryk af, at det har været meget brugt? Men det er ikke slettet, bare gjort mindre synlig.

Det samme gælder priser. Vi vedligeholder ingen priser i systemet, men det er åbent for brugere at gøre dette selv.

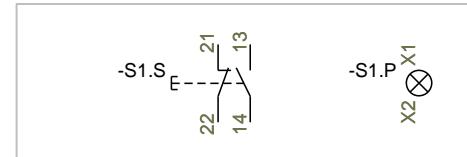
Til gengæld kan feltet bruges til andre ting, som skal 'summes sammen', fx kW eller kg. Det sidste kræver dog, at man har været konsekvent med enheder under oprettelsen af sine komponenter ...



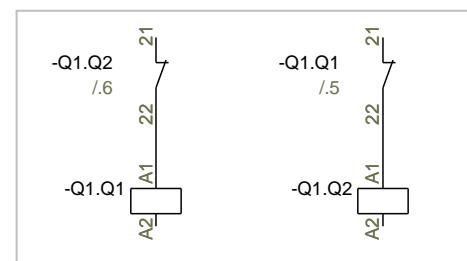
5 ANVENDELSE AF UNDERNAVNE - PRÆCISERING

Undernavne er et begreb i PCSchematic Automation, som har været der i mange programversioner.

Funktionen er lavet for at kunne vise underopdeling af en komponent, fx et lampetryk, som består – inde i selve komponenten – af en funktion for selve trykket og en funktion for lampen. Selve komponenten hedder S1!



En anden typisk anvendelse kan være en afbryder med to aktiveringsspoler, hvor de interne forbindelser og aktiveringar skal tydeliggøres.



Fælles for de to eksempler er, at undernavnet er en del af komponenten, dvs en del af en enkelt vare, og at undernavnet er fast pr vare og derfor også kan styres fra databasen. Selve komponenten hedder Q1!

Fælles gælder også, at man ikke skal underopdele en komponent, da man i diagrammer alene skal vise en komponents hovedfunktion, dvs i de to tilfælde ovenfor hhv en S og en Q.

Det betyder også, at det der vises i Komponentmenuen er *komponentens* navn og ikke *symbolets* navn.

Det betyder også, at tager man et relæ fra databasen, som er oprettet *uden* undernavne, og man giver den undernavn i f.m. placering i projektet – hvilket ifølge ovenstående jo også er forkert – så ser man nu relæets navn, uden undernavn, i listen. Og det er meningen.

Underopdelingen og undernavne er praktiske, hvis man skal vise den indre funktion – i kontakturen – eller hvis man skal erstatte én komponent med flere – i lampetrykket – da man jo kan se funktionen allerede. Meeeen, så kommer referencebetegnelser dvs produktaspektet ind.

5.1 Navngivning af kort i en sammensat plc

Vi har – desværre – fortalt vidt og bredt, at man også kan anvende undernavne til at indikere slot-nr på plc. Det bør man ikke gøre (mere).

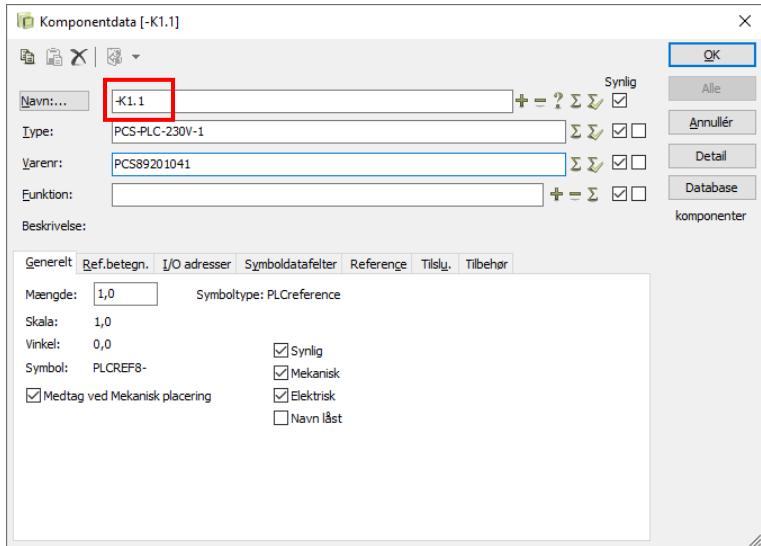
Sådan bør du navngive det enkelte kort i diagrammet:

Kort til Slot 1: -K1.1

Kort til Slot 2: -K1.2

Rent synsmæssigt er der ingen forskel, men det betyder, at programmet arbejder rigtigt og viser de rigtige IO'ere i

Komponentmenuen, og at import af IO-data rammer (mere) rigtigt.



Indlæsning af IO-lister fungerer fint, PLC-listen fungerer, Valg på IO-knappen fungerer, men du vil opleve, at Komponentmenuen ikke viser korrekte IO'er. Og dette skyldes, at undernavn er anvendt forkert. Desværre. Vi skal have rettet op på manualen, hæfter om hvordan man arbejder med PLC'er og YouTube-videoer.



6 TILBEHØR – HVORDAN ARBEJDER JEG MED DET?

Arbejdsgangen med tilbehør er anderledes fra version 20 og da der er lidt ændringer igen i version 21 og 22, så følger der nedenfor en gennemgang af, hvordan man arbejder med tilbehør i programmet.

6.1 Formål med at lave om på arbejdsgang og -metode

1. Som bruger ønsker jeg at have overblik over valgt tilbehør til en komponent.
2. Som bruger ønsker jeg, at det er nemt at vælge korrekt tilbehør.
3. Som bruger ønsker jeg at kunne kopiere en komponent med dets tilbehør

6.1.1 Bedre overblik

Når man åbner Komponentdata på en hovedkomponent, vises alt fast og valgt elektrisk og valgt (og muligt) mekanisk tilbehør. Hvis tilbehøret er oprettet med billeder, vises det, når man stiller sig på tilbehøret i fanen.

Når der er valgt tilbehør på en komponent, viser Komponentmenuen en træstruktur, hvor det valgte tilbehør vises som en gren. Evt fast tilbehør vises først i træstrukturen, når det er placeret.

Det optimale, når vi arbejder med tilbehør, er, at det mulige tilbehør er oprettet i databasen.

6.1.2 Elektrisk tilbehør

Vælges i Komponentmenuen på DIA-siden. Elektrisk tilbehør har sine egne diagramsymboler.

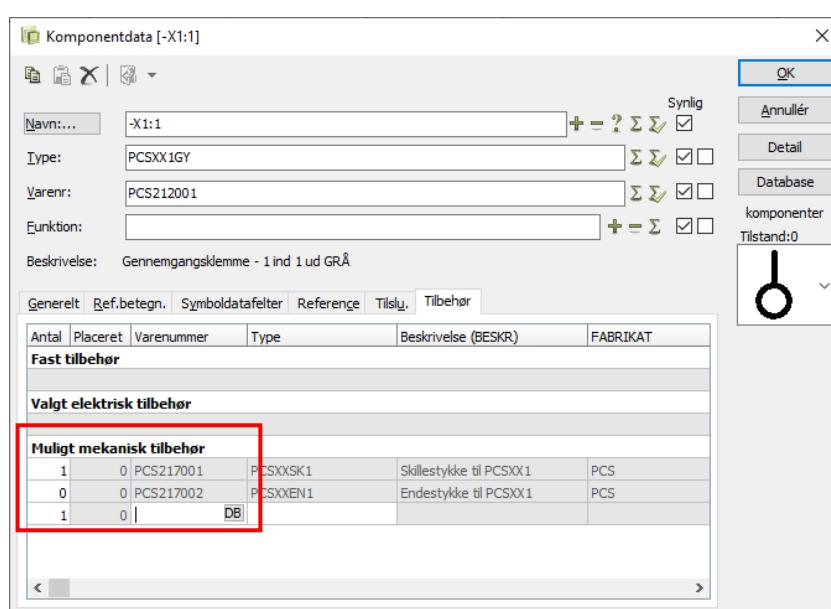
Hvis ikke tilbehøret vælges via Komponentmenuen, optræder det ikke i træstrukturen og opfattes ikke som tilbehør.

6.1.3 Mekanisk tilbehør

Den enkleste måde er at vælge det mekaniske tilbehør på fanen Tilbehør.

På den måde kommer alt det valgte tilbehør i styk- og komponentlisten, og hvis der er et mekanisk symbol, kan dette symbol efterfølgende placeres på GRP-siden via Vis ledige.

Man kan også vælge mekanisk tilbehør i Komponentmenuen på GRP-siden, dog kun det tilbehør, der har sit eget mekaniske symbol. Resten vælges på fanen Tilbehør.



6.1.4 Fast tilbehør

Defineret i databasen og kan kun ændres i databasen.

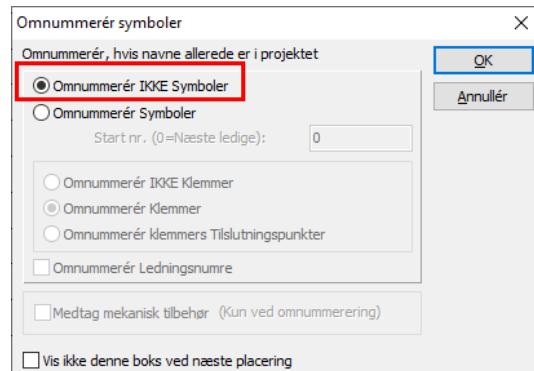
6.1.5 Kopi af komponent med mekanisk tilbehør

Tilbehør er knyttet til komponenten.

Når programmet er i tvivl om, hvilken komponent, det er knyttet til, kommer det ikke med i kopien.

Dette gælder fx, hvis man kopierer og ikke omdøber.

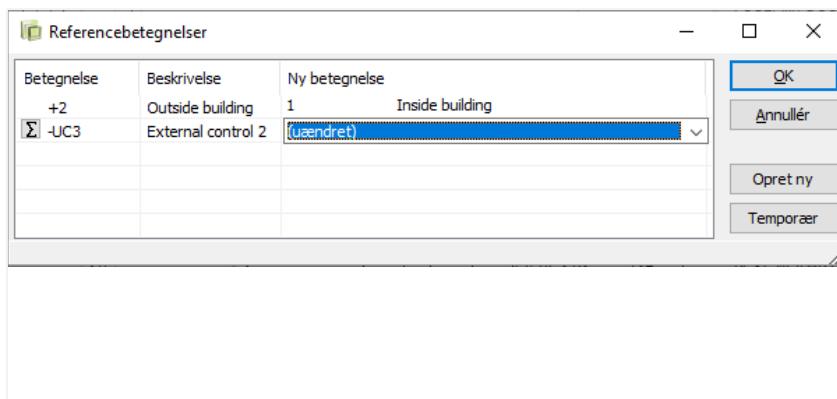
Vær opmærksom på, at hvis du arbejder med sidestrømsvejsnummerering, kommer tilbehøret ikke med! Uanset, om klemmer følger siden eller ej.



6.1.6 Kopi og referencebetegnelser

Når du arbejder med ref.betegnelser, hjælper programmet dig med at vælge en ny betegnelse for det kopierede.

Hvis du ikke ved, hvad det nye skal hedde, kan du vælge at lave en temporær / midlertidig betegnelse. Så kan du skifte på et senere tidspunkt.



Husk, at magen til ikke er det samme!

Samme varenummer medfører ikke samme komponentnavn.

Samme komponentnavn er ikke samme referencebetegnelse.

6.2 Man kan konfigurere Tilbehørsfanen

Man kan justere bredden af kolonner i fanen Tilbehør.

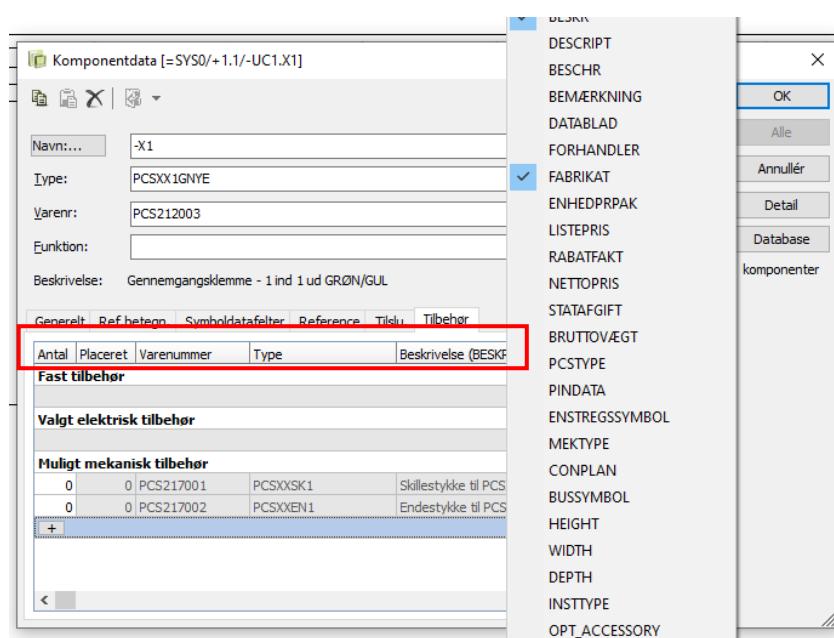
Hvis man højreklikker i den øverste linje på fanen, kan man vælge, hvilke databasefelter, der vises for tilbehøret.

Indstillingen bliver generel på alle komponenter.

De data, som hentes på denne måde, bliver også vist i popup-info på komponenten, så et godt råd er, at du skal begrænse antallet af felter.

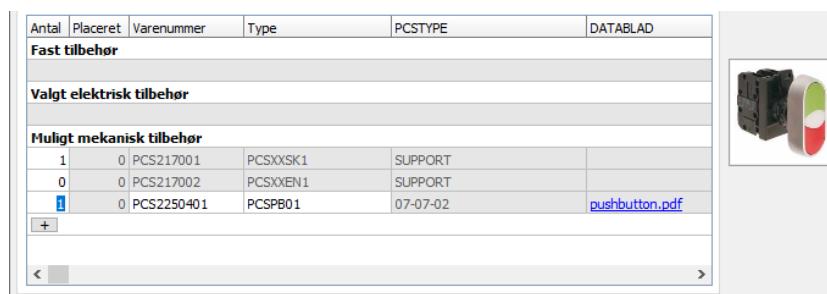
Man kan sætte fx links til fabrikantens hjemmeside i et databasefelt. Hvis man vælger dette databasefelt på tilbehørsfanen, kan man åbne det herfra.

Og links på hovedkomponentens databladet mm åbnes jo øverst i Komponentdatadialogen (mappe-ikonet) 😊



6.3 Evt billeder af tilbehør vises på fanen

Hvis et tilbehør – valgt eller muligt – har et billede i database, bliver det vist på tilbehørsfanen, når musen kører hen over det.^{xi}



6.4 Supportsymbolet og tilbehør

Supportsymbolet er den 'gamle' måde at angive tilbehør på. Hvad sker der med disse?

6.4.1 Supportsymbolet i eksisterende projekter

Supportsymbolet i eksisterende projekter bliver behandlet som tidligere. Dvs, at har du et eksisterende projekt, hvor dit mekaniske tilbehør er angivet på supportsymbolet, så er disse komponenter også en del af projektet i version 21 og 22.

Du skal dog være opmærksom på, at dette tilbehør ikke optræder som *tilbehør* i projektet, men som komponenter på samme niveau som hovedkomponenten, hvilket betyder, at dette tilbehør *ikke* bliver vist på fanen Tilbehør, og at det ikke ses i Komponentmenuen som tilbehør.

Det medfører også, at hvis du kopierer en komponent, som har tilbehør i form af supportsymbolet, så kommer dette tilbehør ikke med, for programmet ved ikke, at der er tale om tilbehør.

Men tilbehørskomponenten kommer (stadig) på stk- og komponentlister.

6.4.2 Supportsymbolet bliver ignoreret i Komponentmenuen fra ver. 21

Hvis en komponent alene har et supportsymbol angivet som elektrisk symbol i databasen, bliver dette symbol ignoreret i Komponentmenuen fra version 21.

Det betyder, at der ikke er 'ledige' supportsymbolet og dermed risiko for at sætte for mange tilbehørskomponenter ind på denne måde.

Det er kun, når supportsymbolet står som eneste diagramsymbol, at vi ignorerer det fra listen. Du skal selv være opmærksom på de komponenter, hvor (elektrisk) tilbehør er angivet med supportsymbolet i PCSTYPE i databasen.



7 KLEMMER I KOMPONENTGRUPPE 0 OG TILBEHØR *

Vi har lavet lidt om på reglerne for, hvordan vi håndterer tilbehør, generelt og især på klemmer.

Det betyder bl.a., at man ikke kan have tilbehør på klemmer i komponentgruppe 0. Hvis det ikke er noget, du har tænkt dybere over, så kan du roligt springe resten af dette afsnit over, for så har programmet sørget for at undgå gruppe 0 😊

7.1.1 Lidt historie

I ældre versioner, blev symboler, herunder også klemmer, sat ind i projektet, med komponentgruppe 0. Det betød bare, at man i projektet ikke havde taget stilling til, hvilke komponenter, der skulle bruges. Endnu.

Når man valgte varenummer, kunne man for alle andre komponenter end flerlagsklemmer, beholde denne komponentgruppe, og styklisten talte stadig rigtigt. Programmet kunne (næsten) altid gætte rigtigt.

For almindelige gennemgangsklemmer, betød det, at hvert klemmesymbol blev opfattet og behandlet som en komponent, dvs styklisten var ok, men når man satte det mekaniske symbol ind, kunne man ikke altid vide, hvilke symboler, der hang sammen. Dvs at man ikke kunne bruge højreklikfunktionen Gå til symbol på andre sider.

Programmet har ikke selv tildelt komponentgruppe 0 i flere versioner, men der er stadig mange komponenter 'derude', da man jo ofte genbruger sine projekter.

Det, som programmet har gjort i de seneste versioner, er, at komponenter får fortløbende komponentgruppenumre, sådan at dette kan bruges til at følge den enkelte klemme på alle sider i projektet; det giver nemlig et unikt id på en komponent. Og så kan man via højreklik gå til komponentens symboler på de forskellige sider.

Der er dog kommet flere og flere udfordringer i forbindelse med at 'gætte rigtigt'. Og især i forbindelse med klemmer, og næsten altid, hvis de har tilbehør. Og det er grunden til, at vi er i gang med at udfase komponentgruppe 0.

7.1.2 Hvis du har klemmer med tilbehør og de ikke er placeret mekanisk

Her er der ingen problemer. Styklisten og komponentliste passer, og du kan roligt arbejde videre.

Men, hvis du vil arbejde videre med klemrækken, enten ved at klikke på komponenten på diagramsiden for at åbne Komponentdata dialogen eller hvis du vil indsætte komponenterne på den mekaniske side, skal du bruge Ret op-funktionen, dvs tryk på knappen i Komponentvinduet. Du kan altså ikke redigere i klemrækken, hvis dens symboler er i komponentgruppe 0.

Når du trykker på Ret-op knappen, så sættes alt tilbehøret i klemrækken til at 'tilhøre' én af klemmerne i den valgte klemrække, og alle komponentgruppenumre sættes til 1, 2, 3 osv. Styklisten passer – stadig – og du kan frit placere klemmerne rigtigt fra Komponentmenuen på den mekaniske side.

I version 22 kan du rette alle klemrækkerne op på én gang.



7.1.3 Hvis du har klemmer med tilbehør, og de er placeret mekanisk

Når du har klemmer med tilbehør, der er placeret, kan der være rod i styklisten. Det afhænger af, hvordan du har valgt tilbehøret og hvordan du har placeret de mekaniske symboler.

7.1.3.1 Tilbehør valgt vha supportsymboler

Dette er den helt gamle måde at placere tilbehør på. Her kan der også være rod i styklisten.

7.1.3.2 Tilbehør valgt vha fanen Mekanisk tilbehør

Siden v18 har det været muligt at vælge tilbehør på fanen Mekanisk tilbehør. Hvis man bagefter har rettet komponentgruppen til 0, giver det rod i styklisten.

7.1.4 Hvordan kan jeg rette op, hvis der er fejl i styklisten pga komponentgruppe 0

Den letteste og enkleste metode lyder drastisk, for du skal

1. fjerne klemrækken på den mekaniske side!
2. gå til diagramsiden og ret op
3. gå til mekanisk side og placer klemrækken igen
 - a. højreklik og vælg komponentnavn, så har du hele rækken i trådkorset

Og så passer listerne og du kan gå fra mekanisk til diagrammet og tilbage igen på den valgte komponent!



8 LIDT SMÅRETTELSER TIL KOMPONENTMENUEN

Komponentmenuen kan nu huske størrelse mm.:

- Bredde på hele menuen
- Størrelser på de tre vinduer
- Symbolstørrelse
- Stakke eller ej?

8.1.1 Grene lukkes efterhånden

Når man har komponenter, der er oprettet med alternativt symbolvalg, lukkes grenene efterhånden, som er brugt.

8.1.2 Vinduet forbliver lukket

Hvis man har lukket vinduet, så forbliver det lukket i de fleste tilfælde:

Hvis man blot bladrer i sit projekt, eller hvis man tilføjer en komponent med kun ét symbol. Vinduet åbner, hvis man tilføjer en komponent med flere symboler – det er jo der, man vælger de næste symboler.^{xii}

8.1.3 Popup på Muligt tilbehør

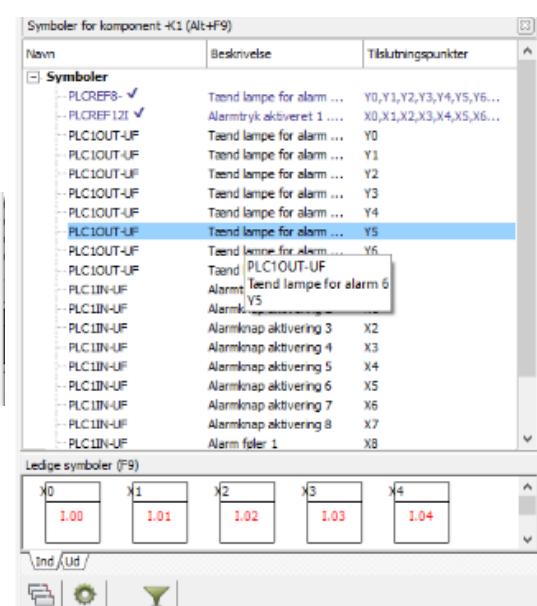
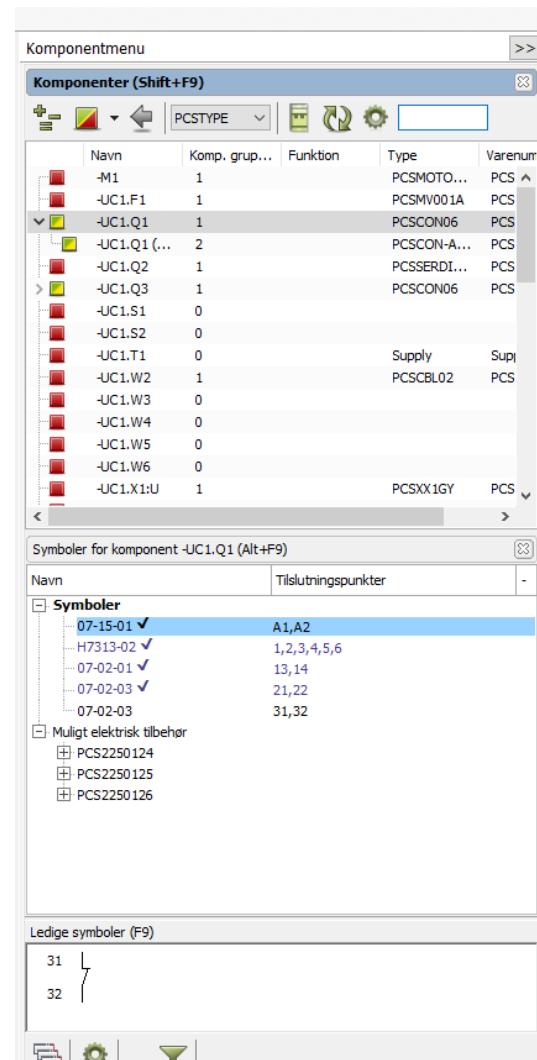
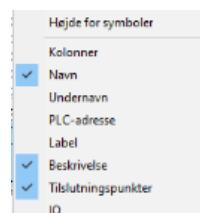
Muligt tilbehør vises nu med Type og Beskrivelse i en popup.

Og på selve Tilbehörsfanen på komponenten kan du se mere information, herunder evt. billede og evt. link til datablad.

8.1.4 Indstilling af kolonner i symboldelen

Man kan indstille hvilke kolonner, man ønsker at se i symbolvinduet, sådan at man vælge fx plc-symboler ud fra fx adresse og beskrivelse.

Når du trykker på tandhjulet nederst får du følgende valgmuligheder:^{xiii}



8.2 Placering af mekaniske symboler

Når man dobbeltklikker på et komponentnavn, hvor det mekaniske symbol ikke er placeret, får man det mekaniske symbol i trådkorset og sparer på den måde et flyt af musen til Vis ledige vinduet og et efterfølgende klik på symbolet.

8.3 Ændring af genvej

Nyheden ovenfor betyder, at den tidligere genvej 'Dobbeltklik vælger Komponentnavn' udgår.

Dobbeltklik på komponentnavn, ændres på følgende måde:

- På diagramside: der sker ingenting
- På mekanisk side: hvis det mekaniske symbol ikke er placeret, får man det i trådkorset.
- Vælg komponentnavn findes – stadig – i højreklik. Funktionen er især praktisk ifm placering af klemrækker.

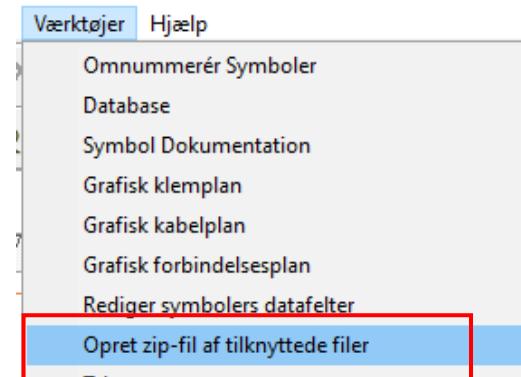


9 TILKNYTNDE DATABLADE KAN ZIPPES

Fra version 22 kan man oprette en zip-fil med tilknyttede datablade.

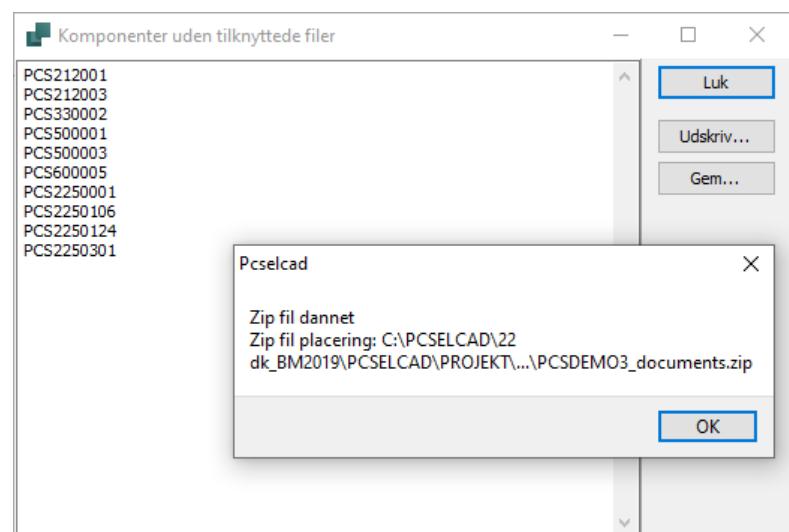
Funktionen er lavet som et værktøj, og ligger i Værktøjsmenuen.

Det fungerer på den måde, at den henter de filer, der er linket til i 'Foretrukket link-felt' og pakker dem i en zip-fil. Samtidigt laves en liste, over de komponenter, der ikke har et tilknyttet datablad.^{xiv}



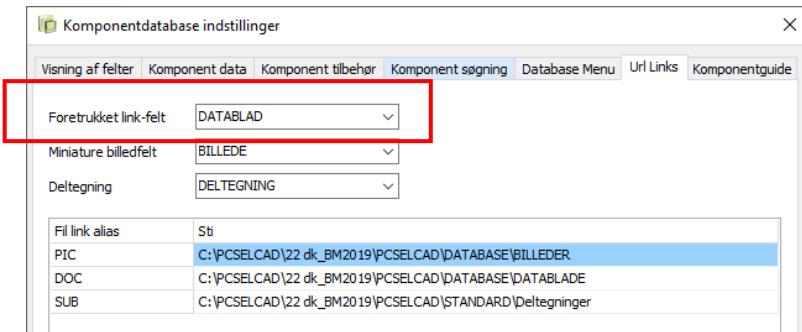
Zip-filen gemmes i samme mappe, som selve projektet og hedder det samme som pro-filen, fx
PCSDEMO3_documents.zip.

xv



Her er der linket til databladsfeltet DATABLADE, da modulet alene pakker filer, som findes i Alias-mappen og ikke de filer, som evt måtte findes på en www-adresse.

xvi



10 NOGET OM LISTER – IMPORT OG EKSPORT

Det at kunne importere og eksportere lister til og fra projekter er en væsentlig funktion. Derfor er der også mange ønsker til funktionalitet på dette område, og derfor er der også ofte nye funktioner her.

10.1 Indlæs styk- og komponentlister vha formatfil

Indlæs komponentliste spørger nu om den formatfil, som listen skal fortolkes med.

Dvs at når man indlæser, vælger man nu den korrekte formatfil, som indeholder information om, hvilke data, der er i hvilke kolonner. Tidligere havde komponentlisten et fast format, nu er det muligt at vælge forskellige formater.

Når man placerer en komponent fra listen, fungerer det som hidtil.

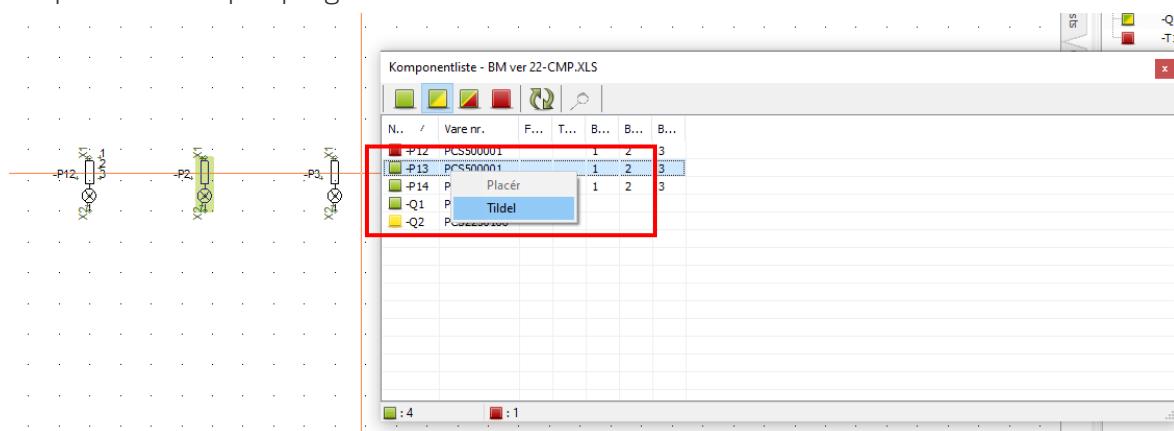
10.2 Tildel data fra en indlæst komponentliste til projektet

Man kan indlæse en komponentliste, som indeholder Komponentnavne og -data og tildele disse data til allerede placerede symboler i et eksisterende projekt.

Komponentlisten kan også indeholde data i symboldatafelter.

Det fungerer på følgende måde:

Start med at oprette en komponentliste i Excel. Den kan fx se ud som nedenfor. Listens format følger en formatfil, på samme måde, som vi definerer andre import- og eksportformater på i programmet.



10.3 Listeeksport til Excel kan skrive til en eksisterende skabelon

Når man eksporterer sine lister til Excel, er det ofte fordi skal genbruge data til noget andet, fx som baggrund for opmærkning.

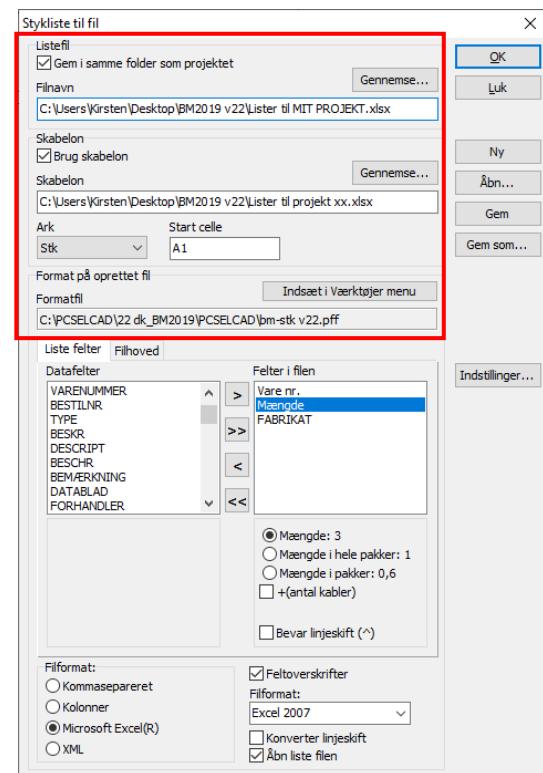
Listeeksport er udvidet, sådan at man i f.m. eksport til Excel kan vælge en eksisterende fil/skabelon, hvilket ark og hvilken celle man sender eksporten til.

På denne måde kan man have flere lister på hver sin fane i samme projekt, uden at man selv skal 'klistre dem sammen'.

Funktionen er kun tilgængelig ved Excel-eksport.^{xviii}

Sådan kan du bruge funktionen:

- Jeg har lavet en excel-fil med forskellige faner, en for hver type liste, som jeg bruger som skabelon til min første 'Liste-til-fil' fra et projekt.
- Du kan – som tidligere – gemme under eget valgt filnavn. Default-filnavne indikerer listetype; men når jeg bruger denne funktion, omdøber jeg til fx 'xx-lister'.
- Til de næste 'Liste-til-fil' fra dette projekt, vælger jeg 'xx-liste' som skabelon.
- For hver listetype har jeg lavet en fane, og man kan vælge startcelle.^{xix}
- Alle valg kan – som tidligere- gemmes i en formatfil.
- Alle 'Liste til fil' kan gemme i samme folder som selve projektet.^{xx}



10.4 Indholdsfortegnelsen kan vise sider efter en valgt dato

Listeopsætning under Indholdsfortegnelsen er udvidet med datovalg:

Dette giver mulighed for at lave en hurtig liste over ændrede sider.^{xxi}

| Titel | Revision | Sidste rettelistedat | Sidenr. | Titel |
|-------------------------|---------------------|----------------------|---------|-------|
| Front page | 06/02/2020 15.50.00 | | 2 | |
| Table of Contents | 06/02/2020 18.05.52 | | 4 | |
| Diagram | 06/02/2020 15.50.00 | | 6 | |
| Diagram | 06/02/2020 15.50.00 | | 7 | |
| Control circuit diagram | | | | |
| Control circuit diagram | | | | |
| Diagram | | | | |
| Control circuit diagram | | | | |
| Control circuit diagram | | | | |
| Control circuit diagram | | | | |
| Control circuit diagram | | | | |
| Control circuit diagram | | | | |
| Control circuit diagram | | | | |
| Arrangement - A4 - 1: | | | | |

Opsætning af indholdsfortegnelse

Kun kapitelfaneblade

OK

Kun sider i kapitlet

Annullér

Medtag alle sider i projektet

Opdater

Indrykning af sider i kapitler

Vis kun sidetitel ved kapitelfaneblade

Fra dato

06-02-2020

I spalter

10.5 Man kan hoppe til specifikke streger fra en styk- eller komponentliste

Hvis man inkluderer streger i sine styk- og komponentlister, kan man nu hoppe direkte til dem fra listen, på samme måde, som man kan fra en komponent.

^{xxii}

| Komponent | Varenr. | Type |
|-----------|---------|-------|
| 1 | asdf | streg |
| 2 | sdfg | streg |
| 3 | dfgh | streg |

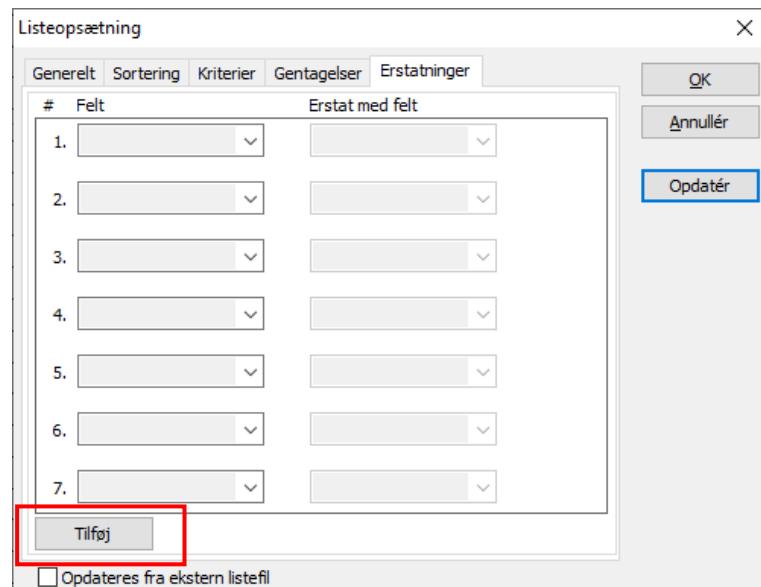
OBS: Hvis du indlæser et gammelt projekt, så SKAL du opdatere listen for at dette virker!



10.6 Antal gentagelser og antal erstatninger i Listeopsætning er udvidet

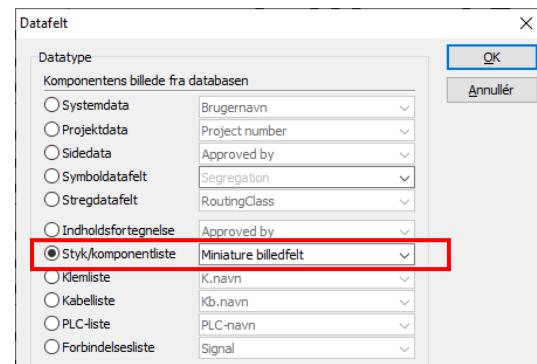
Man kan nu selv vælge hvor mange gentagelser, hhv erstatninger, der skal kunne laves i en liste.

xxiii



10.7 Nyt datafelt på Styk- og komponentlister

Der er et nyt datafelt, som kan indeholde thumbnails. Nedenfor er vist eksempel på en stykliste, som indeholder Varenr., Type, Fabrikat, Beskrivelse og det nye Miniature billedfelt.



Komponenter anvendt i projektet

PCS2250301

PCSSERDISCON01

PCS

PCS serviceafbryder type 1

PCS2250401

PCSPB01

PCS

Tryknap slutte



PCS2250411

PCSPB NC 01

PCS

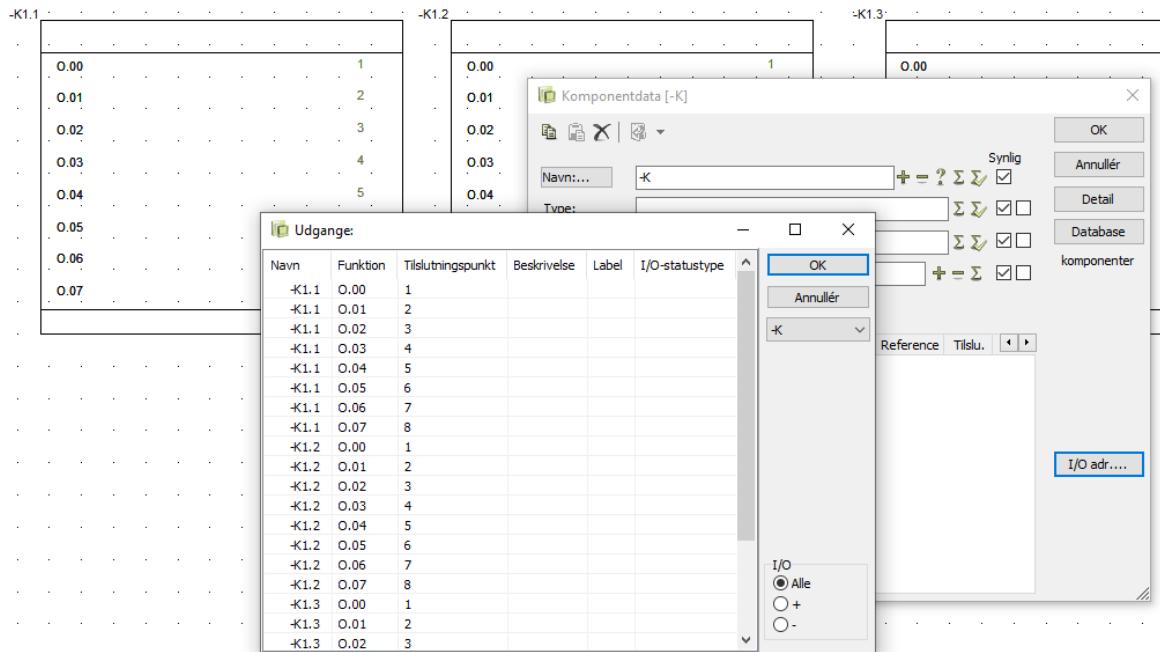
Tryknap bryde



10.8 PLC-liste skal inkludere plc'ens undernavn i sin sorterering *

Plc-listen inkluderer nu plc'en undernavn i sorteringer og andre visninger.

Dette gælder, når man leder efter en adresse vha I/O-knappen:



Og det gælder i plc-listen.

Inden du begynder på (eller fortsætter med) at bruge undernavne på plc'er, så bør du læse afsnit om undernavne på side 19.

xxiv

| Navn | I/O | Beskrivelse |
|---------|-----|----------------------|
| -K1.1:1 | .00 | Kort 1 - adresse .00 |
| -K1.1:2 | .01 | |
| -K1.1:3 | .02 | |
| -K1.1:4 | .03 | |
| -K1.1:5 | .04 | |
| -K1.1:6 | .05 | |
| -K1.1:7 | .06 | |
| -K1.1:8 | .07 | |
| -K1.2:1 | .00 | Kort 1 - adresse .00 |
| -K1.2:2 | .01 | |
| -K1.2:3 | .02 | |
| -K1.2:4 | .03 | |
| -K1.2:5 | .04 | |
| -K1.2:6 | .05 | |
| -K1.2:7 | .06 | |
| -K1.2:8 | .07 | |

10.9 Eksport til Cablemanager

Denne nye knap, gør det muligt at eksportere data direkte fra Automation til Cablemanager.



For at det virker skal man have et Symboldatafelt som hedder Segregation.

Ved ny installation er feltet oprettet, men ved opdatering er det ikke med.

Symboldatafeltet tilføjes på de kabler som skal eksporteres til Cablemanageren.

Symboldatafeltet SKAL være udfyldt for at kablet kommer med i eksporten.

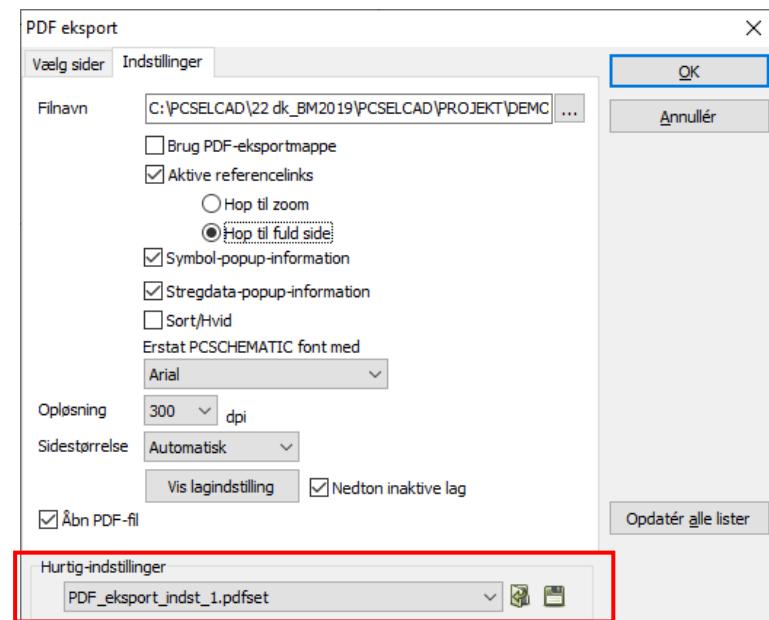


10.10 PDF-eksport via kommandolinje

Man kan starte PDF-eksport vha en kommandolinje, hvilket man har kunnet i flere versioner.

Det nye er, at kommandolinjen kan håndtere de indstillinger, der ligger i en 'hurtig indstillingsfil'.

I eksemplet her, har jeg blot trykket på Gem for at gemme de indstillinger, der er vist på skærmen. Default gemmes filen under xx\PCSELCAD.



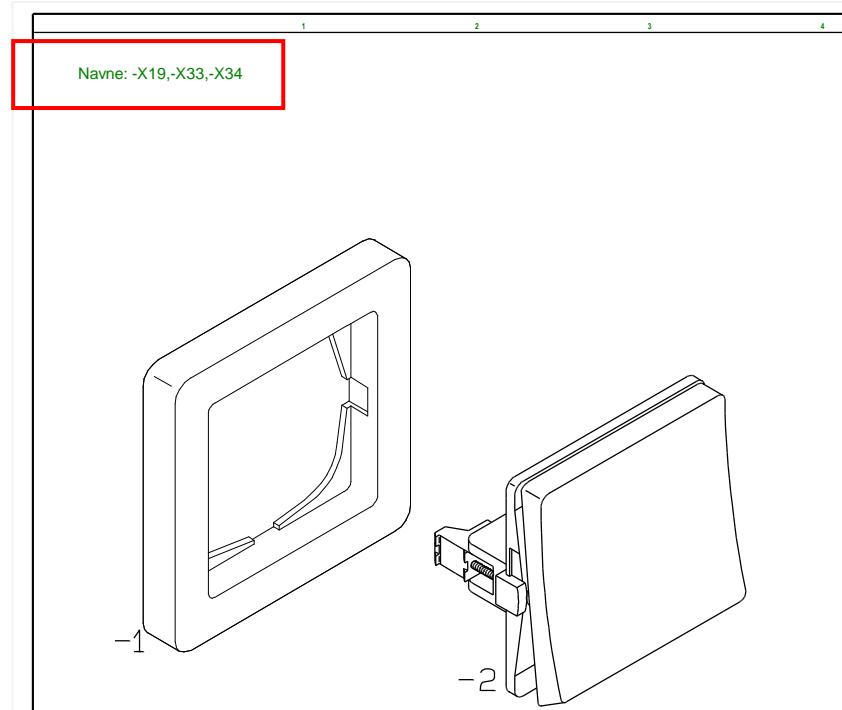
Formatet til kommandofilen er:

```
FILESAVEAS.PDF PDF_eksport_indst_1.pdfset c:\pcselcad\Projekter\MitProjekt.pro
```

10.11 Navnelisten i detailtegninger er med hyperlink

Når man klikker på navnelisten i Detailtegninger, er der nu også hyperlink funktion på disse, dvs at man hopper direkte ind i projektet på den valgte komponent.

xxv



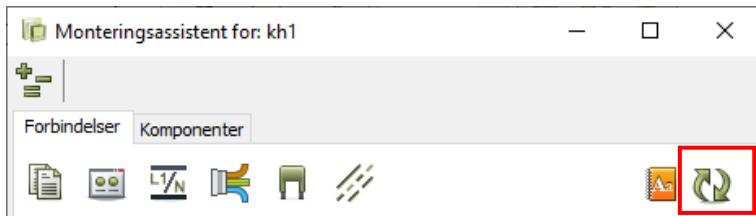
11 MONTERINGSASSISTENTEN

Monteringsassistenten er også blevet udvidet med et par ekstra funktioner.

11.1 To brugere kan montere samme tavle

Hvis man lægger sit projekt på et fælles drev, og opretter et sagsnr, så kan flere bruge åbne både projektet og sagen.

Der er altid en refresh-knap, og du får popup-besked, når antallet af brugere på et projekt ændrer sig.



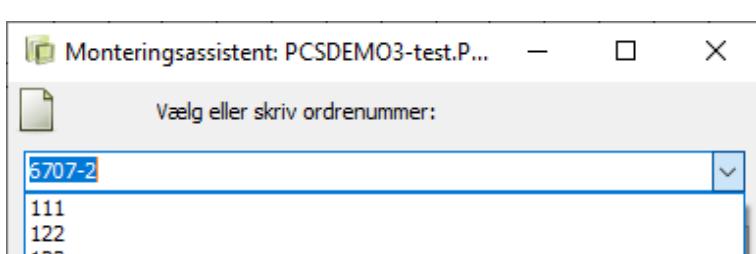
11.2 Forbindelsen kan vælges fra listen eller fra diagrammet

Monteringsassistenten bygger på forbindelseslisten fra projektet, og derfor kommer forbindelserne op i samme rækkefølge, som de findes i projektet.

Du kan dog også vælge en forbindelse direkte i diagrammet, hvorefter den vises i vinduet.

11.3 Alle oprettede sagsnumre på projektet vises

Alle oprettede sagsnumre på samme projekt ligger i en liste, når man åbner Monteringsassistenten.



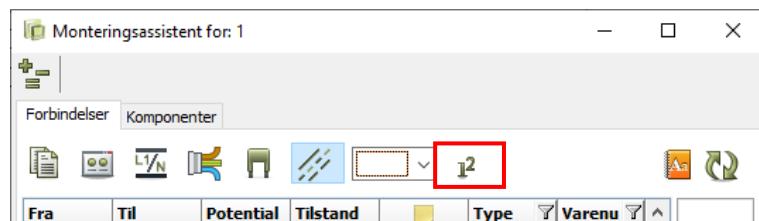
11.4 Oversigtsvinduet har fået en Monteret-knap

Hvis man bruger Monteringsassistenten på en tablet eller har en trykfølsom skærm, kan man nu kvittere for at en ledning er monteret på den store knap yderst til højre i Oversigtsvinduet.



11.5 Ledningsnumre i ledningslisten

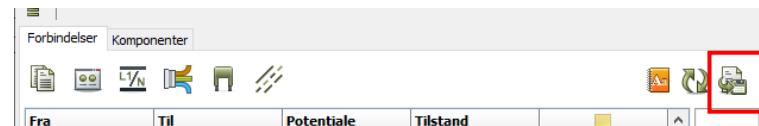
Ledningsnumre har nu sin egen kolonne i listen.



11.6 Eksport til Excel

Man kan nu eksportere til Excel direkte fra forbindelses- og komponentfanerne.

Det er den aktuelle liste, som eksporteres^{xxvi}

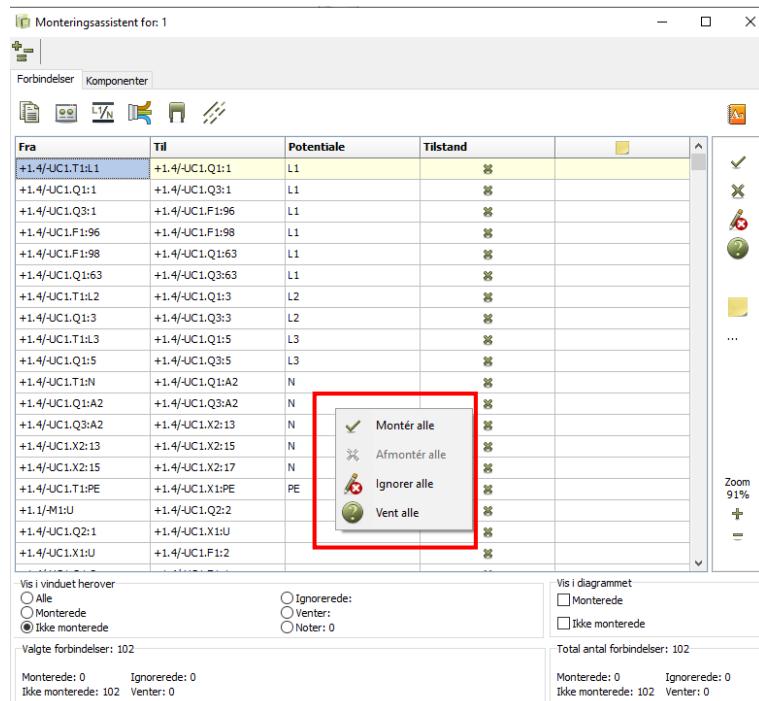


| Navn | Varenummer | Type | Funktion | Tilstand |
|-------------|------------|------------|----------|----------|
| +1.4/UC1.T1 | Supply | Supply | | |
| +1.1/M1 | PCS600005 | PCSMOTOR2A | | |

11.7 Mulighed for at ændre status på alle på en gang

I højreklik kan man ændre status på alle forbindelser på en gang.

Der findes en tilsvarende mulighed på Komponenter-fanen.



12 UDVIDELSER TIL OBJEKTLISTEREN (F7)

Objektlisten er et værktøj, som kan anvendes til mange formål i programmet. Dette betyder også, at der kommer ønsker til funktionaliteten, og disse ting er lavet nu.

12.1 Objektlister – ny standardopsætning *

Når man installerer Automation (ikke opdatering), har Objektlister fået en ny standardopsætning. Opsætningen inkluderer en Komponentfunktion – se også de to billeder her på siden.

Husk, du kan altid lave din egen opsætning om ved at højreklikke på rækken med kolonneoverskrifter.

12.2 Objektlisten kan nu have filter i alle kolonner

Man kan nu indstille
filtrering i alle kolonner i
Objektlisten. Det
fungerer på samme
måde som i fx Excel, så
du kan indtaste noget af
fx en type og derefter få
vist, de komponenter
som har dette i typebetegnelsen.

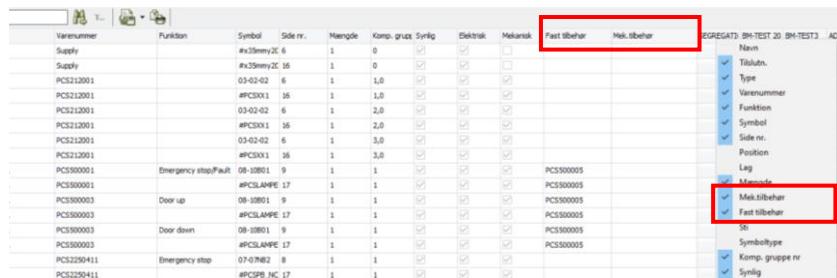
| Objektlist | | | | | | | | |
|------------------------|----------|---------------|--------------|---------|-----------|----------------|------------|---------------|
| Symboler | Signaler | Ledningsnumre | Kabler | Klemmer | PLC I/O | PLC I/O (ref.) | Tekster | Referencammer |
| Andre symt | | DB | Varenummer / | | Type | Funktion | Symbol | Side nr. |
| Komponent | Navn | Tilslutn. | | | | | | |
| =SYS0/+1.1/-UC1.F1 (0) | F1 | 5,6,3,4,1,2, | PCS2250001 | | PCSMV001A | | FBP-PCS225 | 47 |
| =SYS0/+1.1/-UC1.F1 (1) | F1 | 1,2,3,4,5,6, | PCS2250001 | | PCSMV001A | | #PCSMV1 | 15 |
| | F1 | 1,2,3,4,5,6 | PCS2250001 | | PCSMV001A | | H7315-21 | 6 |
| | F1 | 96,95 | PCS2250001 | | PCSMV001A | | 07-09KB3 | 7 |
| | F1 | 98,97 | PCS2250001 | | PCSMV001A | | 07-09KS3 | 8 |
| =SYS0/+1.1/-UC1.F2 (0) | F2 | 5,6,3,4,1,2, | PCS2250001 | | PCSMV001A | | FBP-PCS225 | 47 |
| =SYS0/+1.1/-UC1.F2 (1) | F2 | 1,2,3,4,5,6, | PCS2250001 | | PCSMV001A | | #PCSMV1 | 15 |

12.3 Objektlister - alle symboltyper kan vises på symbolfanen *

Man kan vælge at vise alle symboltyper på symbolfanen, hvilket betyder, at info om bl.a. klemmer og kabler kan vises sammen med de 'almindelige' symboler.

12.4 Objektlisteren kan også vise tilbehør

Man kan nu også vælge at vise tilbehør – både fast og mekanisk – i objektlisteren.

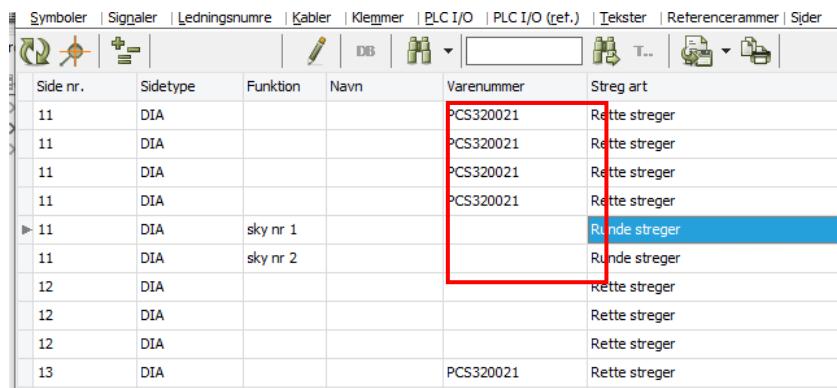


The screenshot shows a table of objects with columns for Varenummer, Funktion, Symbol, Side nr., Mængde, Komp. grupp., Syntg., Elektrisk, Mekanisk, Fast tilbehør, and Mek. tilbehør. A red box highlights the 'Fast tilbehør' and 'Mek. tilbehør' columns. To the right, a sidebar lists filter options with checkboxes, and two specific checkboxes for 'Mek. tilbehør' and 'Fast tilbehør' are checked, also highlighted with a red box.

12.5 Objektlisteren kan vise stregart

Objektlisteren kan vise stregart, sådan at man fx let kan finde sine skyer.

Skyer er afarten 'Runde streger'.



The screenshot shows a table of objects with columns for Side nr., Sidetype, Funktion, Navn, Varenummer, and Streg art. A red box highlights the 'Streg art' column. The 'Runde streger' entry for side number 11 is highlighted with a blue background, indicating it is selected or highlighted.

13 UDVIDELSER PÅ SYMBOLODATAFELTER

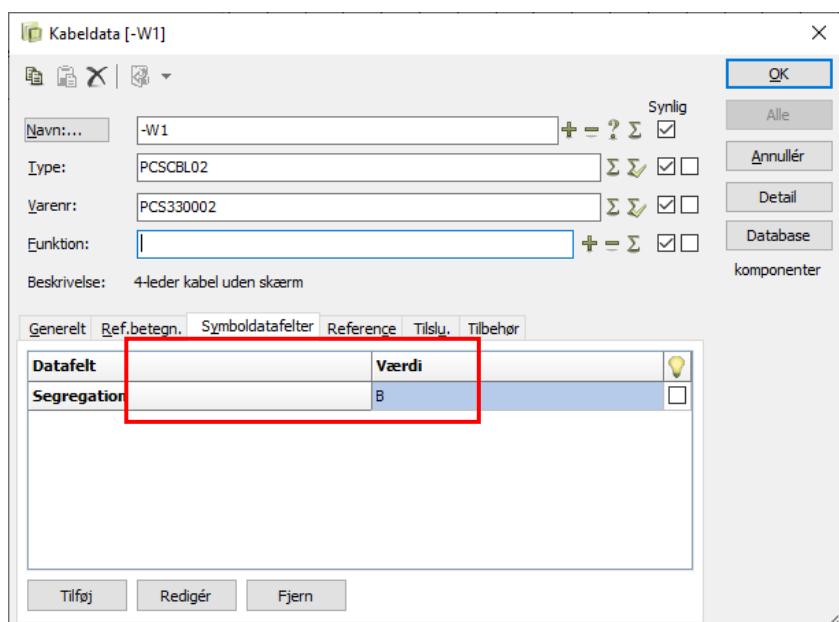
13.1 Nyt fast symboldatafelt

Datafeltet Segregation er et fast oprettet datafelt, som er i alle nye installationer. Feltet er beregnet til at sætte på kabler, som skal/kan eksporteres til Cablemanager.

Segregation = spor i føringsvejen.

Se også afsnittet om eksport til Cablemanager på side 34.

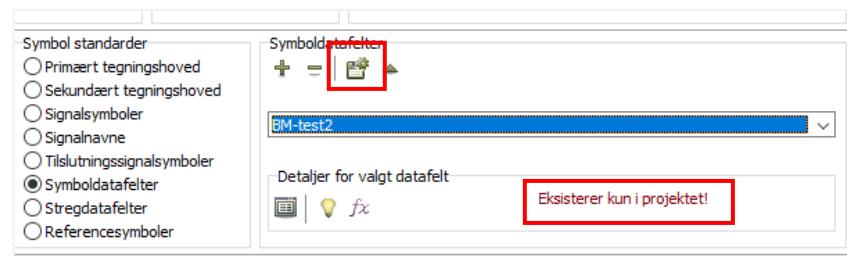
Du skal selv tilføje feltet på de ønskede kabler.



13.2 Symboldatafelter oprettes kun i projektet

Oprettelse af symboldatafelter er nu pr projekt og ikke som tidligere i hele programmet!

Hvis du ønsker at feltet skal være der altid, så skal du klikke på ikonet Gem som standard.



13.3 Symboldatafelter med formular

Symboldatafelter kan nu have formular på samme måde som stregdatafelter.

Formularer kan sammensættes af de faste datafelter (fx Navn og Type), ens egne datafelter og evt faste tekster.

Indholdet i datafeltet behandles som alle andre tekster, dvs du kan kontrollere font, farve, længde, ombrydning mm.

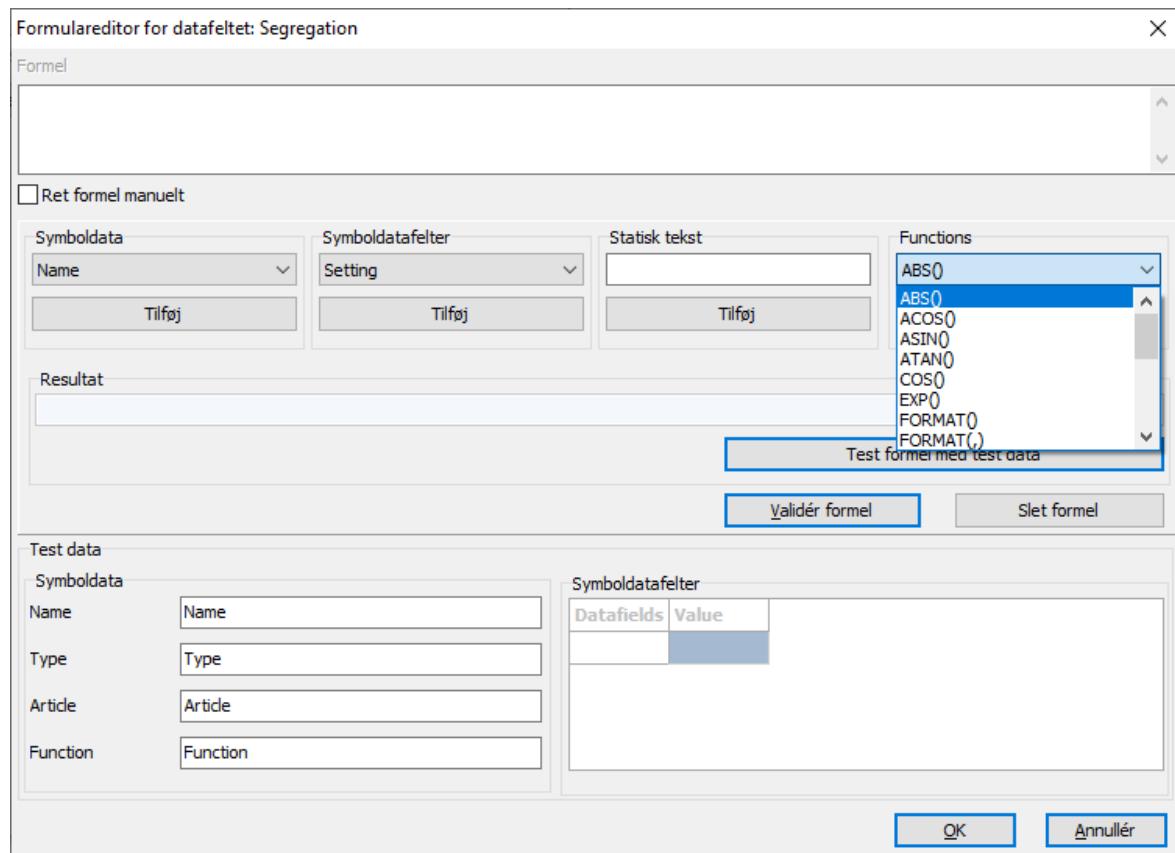


13.4 Ny formel editor for streg og symboldatafelter

Hvis du har brug for at have logiske, matematiske udtryk i dine streg- og symboldatafelter, har vi nu lavet en formeleditor, som du kan bruge,

Hvis du har brug for den fulde dokumentation til dette, så findes den i det sidste kapitel i dette dokument.

Dokumentationen findes kun på engelsk.

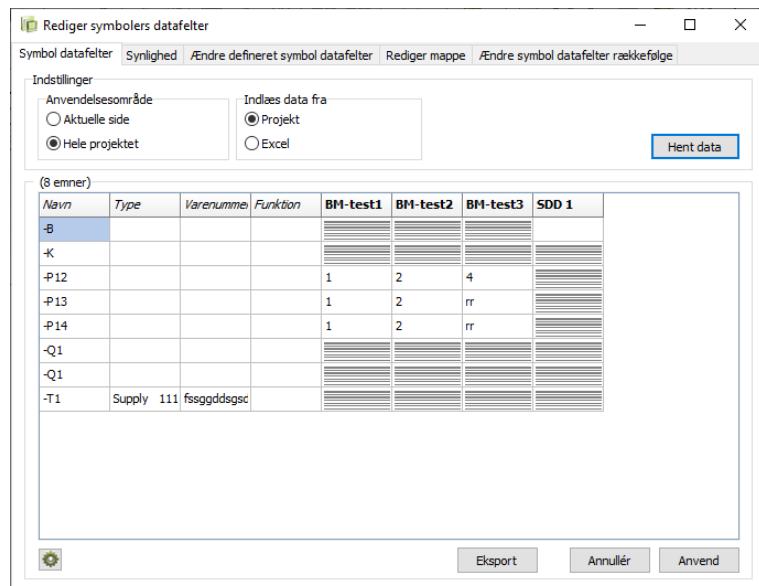


13.5 Værktøjet Rediger symbolers datafelter

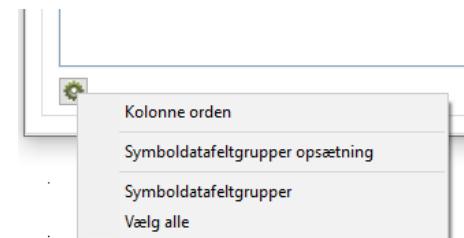
Vi har et værktøj, som kan redigere symbolers symboldatafelter på en nem og hurtig måde. Og hvis man har mange symboldatafelter oprettet til forskellige formål, giver værktøjet mulighed for at holde symbolerne vedlige uden at skulle åbne det enkelte symbol.

Man kan hente alle symboldata med indhold ind på denne fane. Og ændre dem i projektet ved at trykke Anvend.

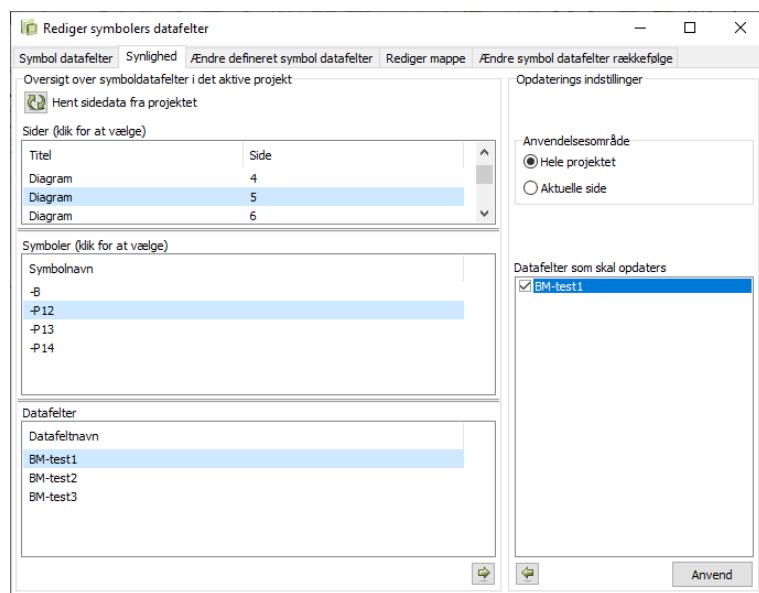
Der kan alternativt indlæses fra en Excel-fil og overføres til projektet.



Der er lavet en ny funktion for dem, som bruger MANGE symboldatafelter. Her er der mulighed for at kunne gruppere dem og ændre på grupperne

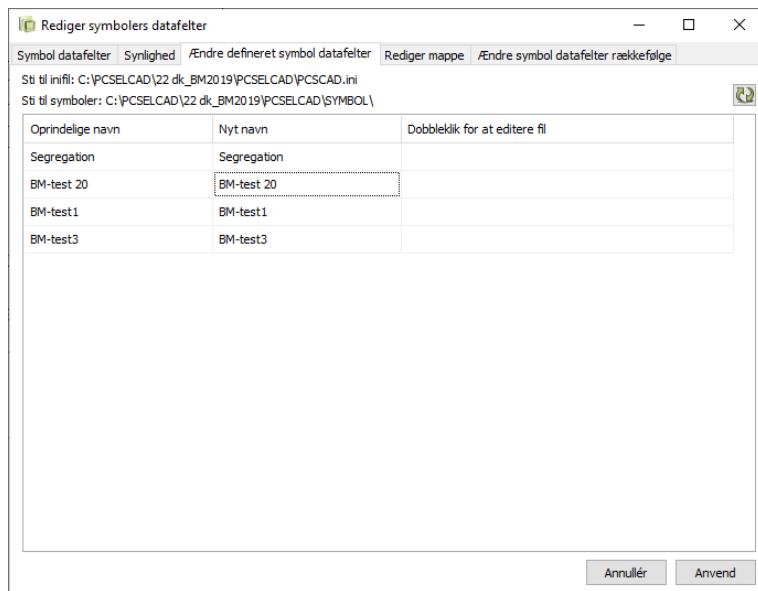


På denne fane kan man ændre synlighed på symboldatafelterne pr side eller i hele projektet.



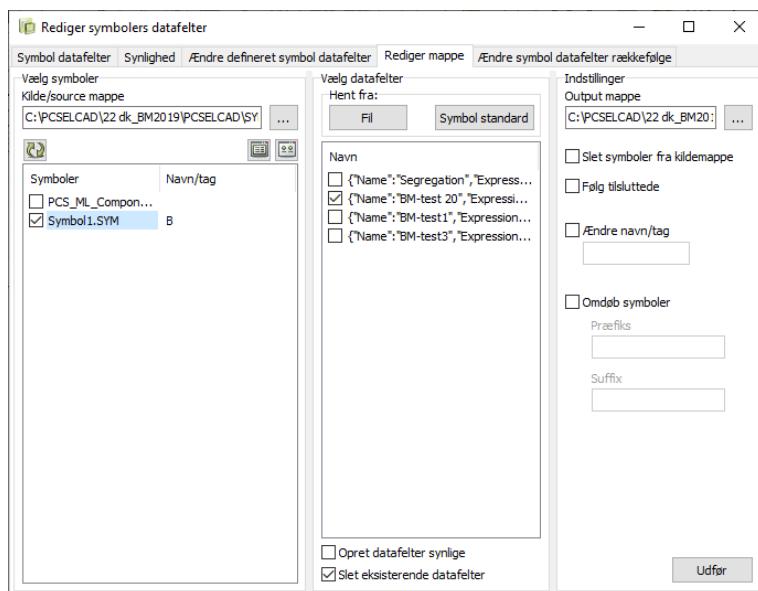
På denne fane kan man ændre symboldatafelters navne på eksisterende symboler.

Ændringen sker kun på symboler, der er oprettet med symboldatafelter og ikke de felter, som kun findes i projektet.

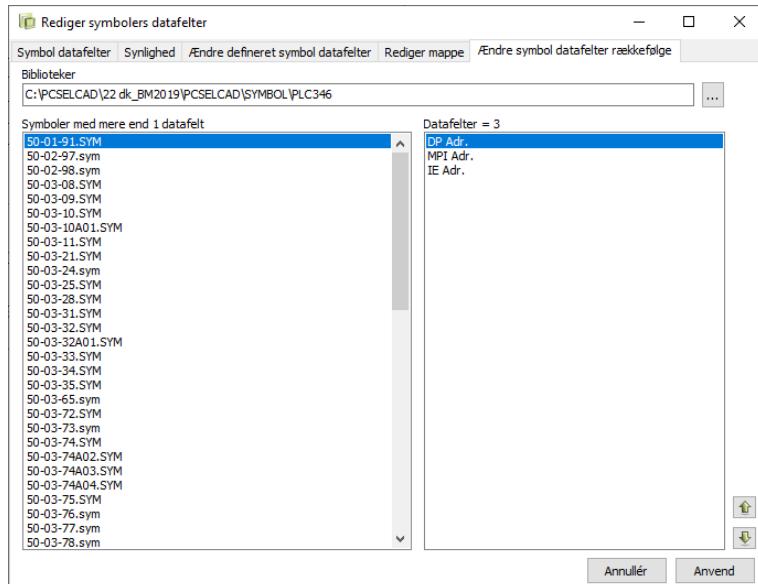


Her kan man tilføje symboldatafelter til symboldokumentation i en symbolmappe, eller slette eksisterende felter.

Der er også mulighed for at omdøbe mm..



På den sidste fane kan man lave om på rækkefølgen på faste symboldatafelter på symboler med mere end ét datafelt



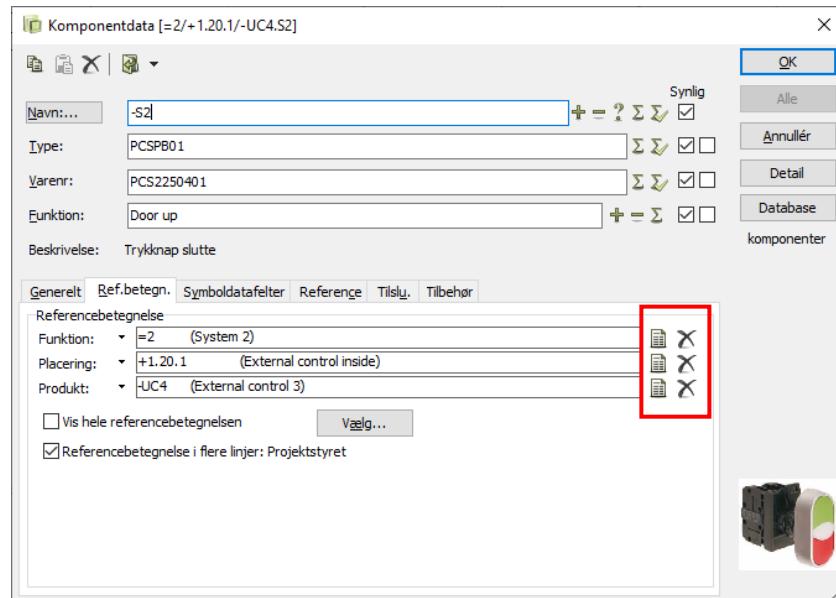
14 SMÅTINGSAFDELINGEN

Også i år er der en række mindre nyheder og ændringer i programmet.

14.1 Valg af referencebetegnelser med Som-siden og Slet-knap*

For at gøre det lettere at vælge ref.betegnelser på komponenter, er der kommet en 'Som siden'-knap og en 'Slet'-knap på fanen Ref.betegn.

xxix

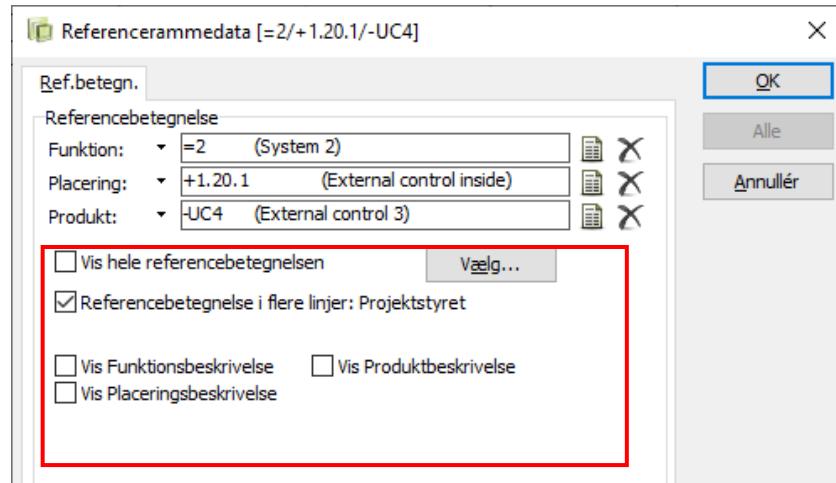


14.2 Indstillinger ved indsæt ref.ramme huskes *

Referencerammer har også en 'Som siden' og en 'Slet'-knap.

Derudover vil sidst valgte indstillinger blive husket, og anvendt ved næste placering af en ref.ramme

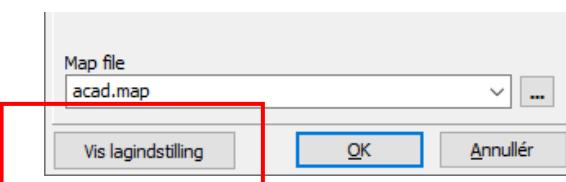
xxx



14.3 Eksport til DWG og DXF

Når man eksporterer til DWG eller DXF kan man nu kalde Lagindstillinger, sådan at man har mulighed for at ændre denne inden selve eksporten.

xxx

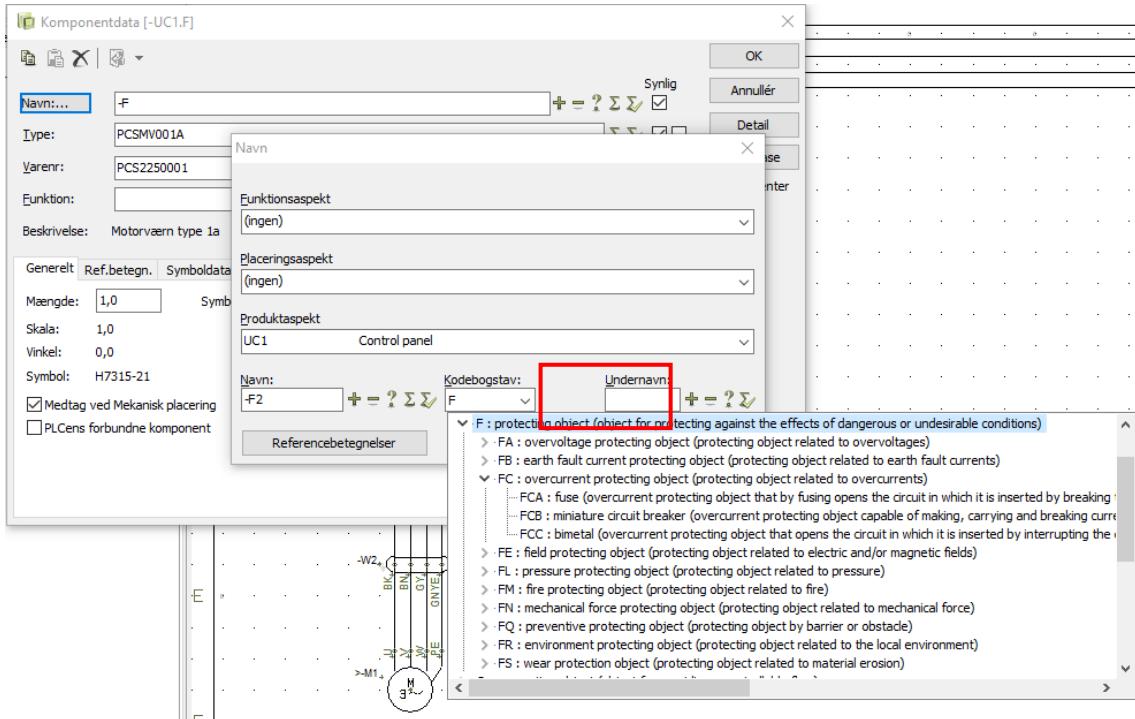


14.4 Kodebogstaver fra 81346-2 findes som lister i programmet

På fanen Navn, kan man nu finde en liste over alle kodebogstaver fra IEC ISO 81346-2, som anvendes til komponentbetegnelser i projekterne.

Den nye udgave af standarden indeholder både 1, 2 og 3 bogstavskoder, og hvis man ønsker at anvende dem, så findes de i listen. Pt er beskrivelserne kun på engelsk, men når den danske oversættelse foreligger, kommer den også i programmet.

Hvis du sætter et symbol ind, hvor fx kodebogstavet er 'F', så åbner listen sig fra 'F'. Når du vælger kodebogstaverne FCC, finder programmet selv næste ledige nr.

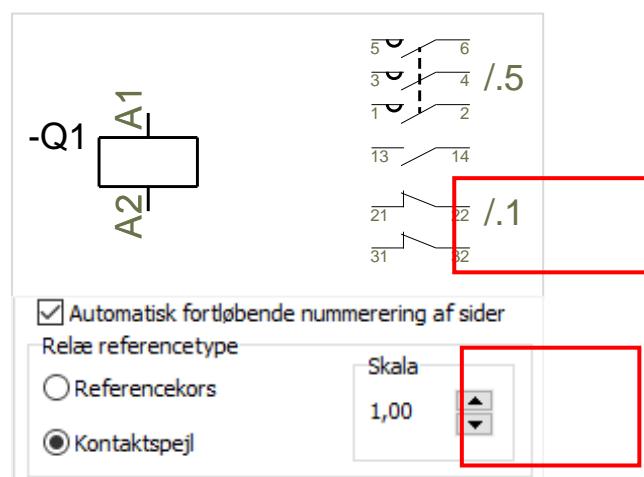


14.5 Forbedret kontaktspejl

Kontaktspejlet – en afart af slutte/bryde-korset – bliver nu vist med samme font og farve som andre tilslutninger og krydsreferencer.

På billedet har jeg flyttet referencesymbolet (højreklik på selve spolen).

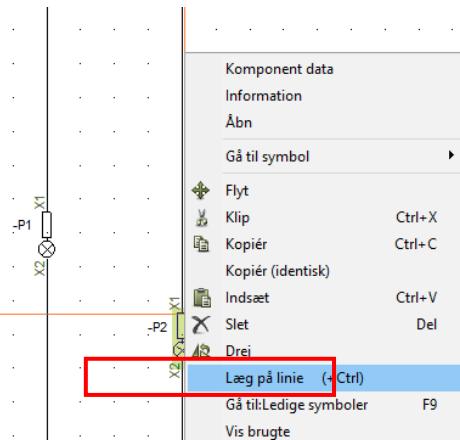
Derudover kan man justere størrelsen for kontaktspejlet under Projektindstillinger.



14.6 Udvidelse af funktionen Læg på linje *

Læg på linje er udvidet, så den nu kan lægge symboler på linje på samme side og på tværs af sider:

1. Marker det symbol, som ligger på den rigtige placering, dvs den der 'bestemmer'.
2. Klik nu på de øvrige symboler, som skal ligge på samme linje. Symbolerne behøver ikke at være på samme side, programmet husker positionen. Hvilket kan ses på 'peg-ud'-stregen, som kommer frem på siden.
3. Symbolerne lægger sig på linjen med det først markerede symbol.



Læg på linje kan også foregå omvendt:

1. Marker det symbol, der skal lægges på linje.
2. Ctrl+klik nu på det symbol, der ligger på den rigtige linje, dvs den der bestemmer.
3. Det først markerede symbol flytter sig nu hen på linje med det sidste symbol.

Ctrl+klik fungerer kun på samme side og ikke på tværs af sider. ^{xxxiii}

Husk: Læg på linje finder altid den letteste linje - uanset, om den er vandret eller lodret.

14.7 Design af symbol med Symbolgenerator

Når man indtaster Bredde og Højde, bliver preview opdateret dynamisk

Når man sætter tilslutninger på, skal vinduet – stadig – forlades, inden tilslutningerne vises.

^{xxxiv}

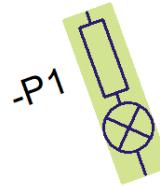
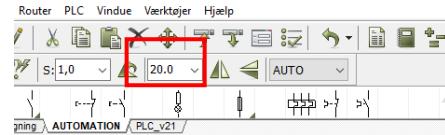


14.8 Rotér et objekt med 10° *

Når man markerer et objekt, og trykker Mellemrumstasten, roterer objektet 90°.

Hvis man samtidigt holder Ctrl nede, roterer emnet med 10°.

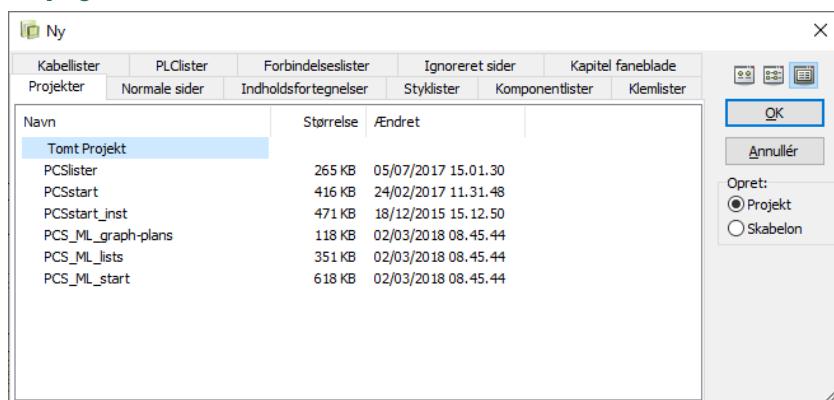
xxxv



14.9 Ikonet Ny og Filer|Ny har nu samme funktion *

Når du trykker på ikonet, åbner du nu dialogen med alle skabeloner.

xxxvi



14.10 Antal senest åbnede filer kan indstilles *

Man kan selv vælge, hvor mange filer, der skal være i fillisten.

For at ændre antallet af filer i listen, skal du tilføje følgende i sektionen [SystemData]:

Skriv fx MaxPickFiles=20 for at få de sidste 20 filer i listen.

Hvis du vælger et stort tal, så er listen måske længere end din skærm kan vise ... xxxvii

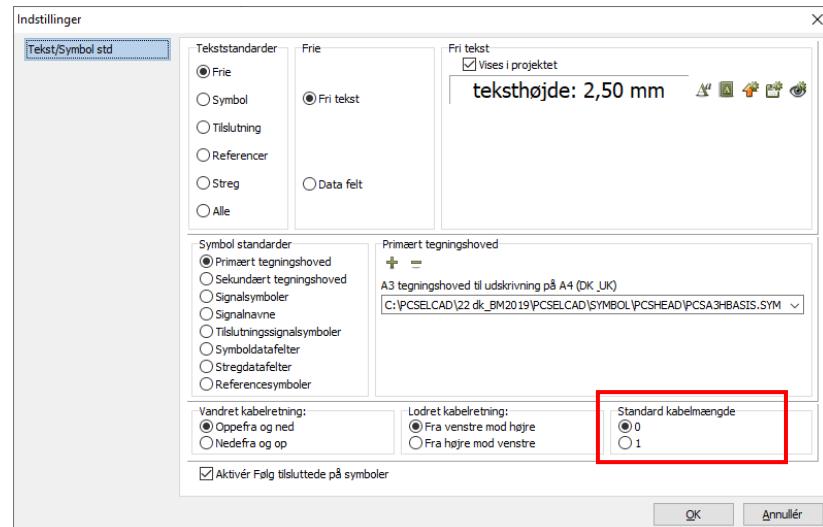
| Moduler... |
|------------------------------------------------------------------------|
| Afslut |
| 1 C:\PCSELCAD\21_dk_trunk\PCSELCAD\PROJEKT\DEMO\All texts in Arial.PRO |
| 2 C:\PCSELCAD\21_dk_trunk\PCSELCAD\PROJEKT\DEMO\PLC i database - 1.pro |
| 3 C:\PCSELCAD\21_dk_trunk\PCSELCAD\PROJEKT\DEMO\PCSDEMO4.pro |
| 4 C:\PCSELCAD\21_dk_trunk\PCSELCAD\PROJEKT\DEMO\PCSDEMO3.pro |
| 5 C:\PCSELCAD\21_dk_trunk\PCSELCAD\PROJEKT\DEMO\PCSDEMO2.pro |
| 6 C:\PCSELCAD\21_dk_trunk\PCSELCAD\PROJEKT\DEMO\PCSDEMO1.pro |

14.11 Standardkabelmængden kan nu indstilles *

Der er lavet en mulighed for at vælge standardkabelmængde.

Indstillingen er en systemindstilling, dvs det følger programmet, og nye kabler, som indsættes, får den valgte standardmængde.

Der ændres intet i eksisterende projekter.



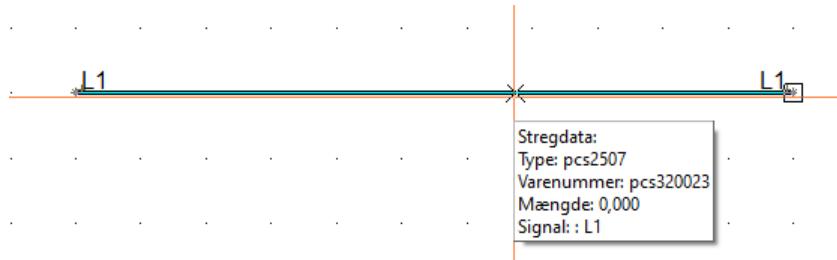
xxxviii

14.12 Streger med varedata

Man kan tegne en streg med varedata (hvis man tager stregen direkte fra pickmenuen).

Hvis man senere forlænger denne streg med en streg uden varedata, så 'limes' de to streger nu sammen og fortsætter med første stregs varedata. Tidligere havde der været to streger, hvilket kunne give problemer ifm fx Panelrouteren.

Det kræver, at man forlænger en 'rød streg' med en 'rød streg', dvs at tegneegenskaberne skal være ens: samme stregetype, farve, ledende.



14.13 Udskift symbol 'Ignorerer symbolsti'

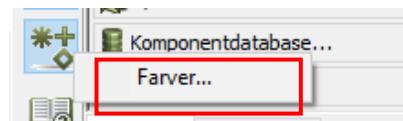
Når man vælger Alle i projektet eller Alle på siden, får du mulighed for 'Ignorerer symbolsti'.

Dvs at funktionen udskifter alle symboler med samme sidste del af symbolnavn, uafhængigt af, hvor det oprindelig blev hentet fra.

Denne funktion er også en del af 'Udskift alle symboler i projektet'.

14.14 Hurtig adgang til farveopsætning i programmet

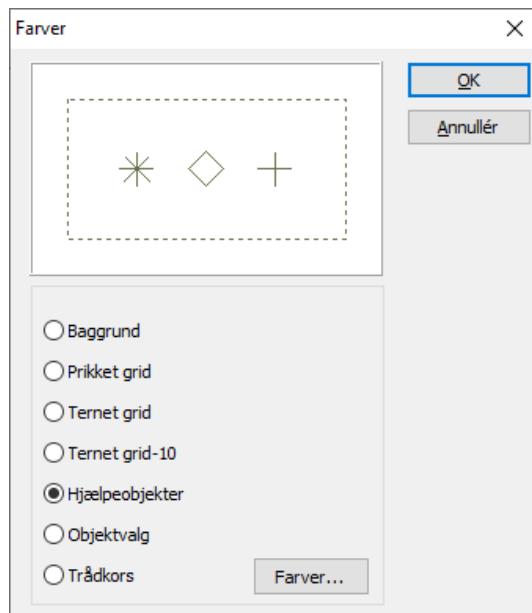
Når man højreklikker på ikonet for Referencesymboler, kan man vælge Farver



Menupunktet åbner grundindstillingen Farver, sådan at man nemt kan skifte farver på fx referencesymbolerne i sit projekt.

Funktionen er også tilgængelig i Automation Service.

xxxix xl



14.15 Højdekote kan overføres vha 'Bjælken'

Hvis man har placeret en komponent i en bestemt højde på en mekanisk tegning, kan man nu ændre den vha 'Bjælken'.

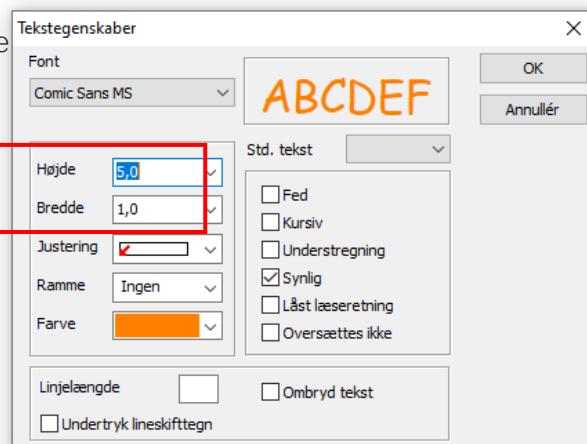


Og ikke vha Flyt, som det var tidligere.

14.16 Tekstegenskaber

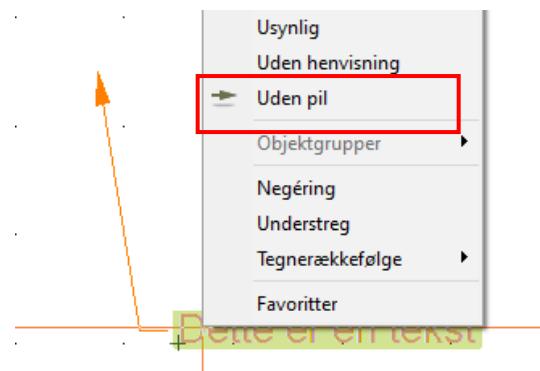
Windows-fonte kan også ændre både højde og bredde.

hej med dig



14.17 Henvisninger – nu med eller uden pil

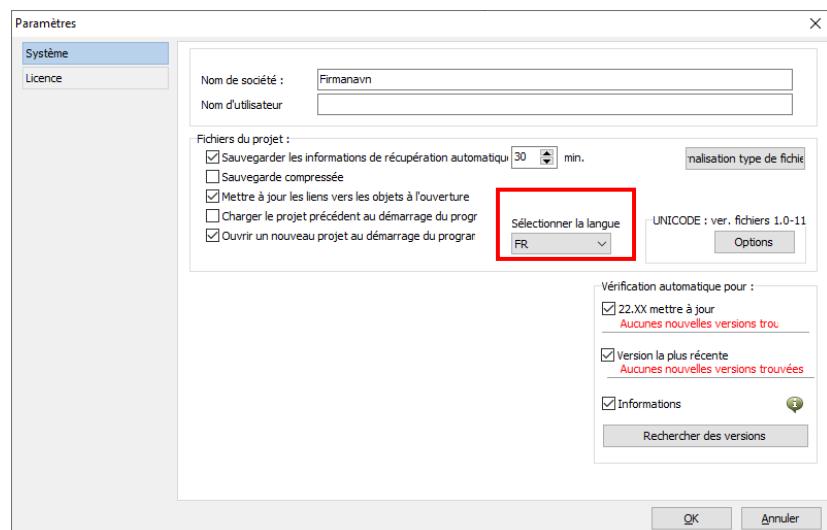
Man kan fravælge pilen på henvisningen.



14.18 Brugerfladen er nu også på fransk

Brugerfladen understøttes nu også på fransk.

'Vores' tegningshoved er endnu ikke oversat til fransk. Men hvis nogen kan hjælpe ...



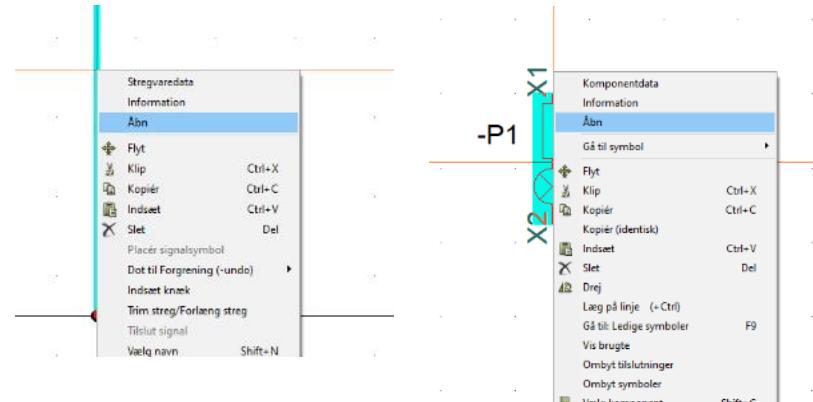
14.18.1 Tegningshoveder findes også på kroatisk

Stor tak til en af vore kunder for hjælpen 😊

14.19 Direkte adgang til varedata i højreklik

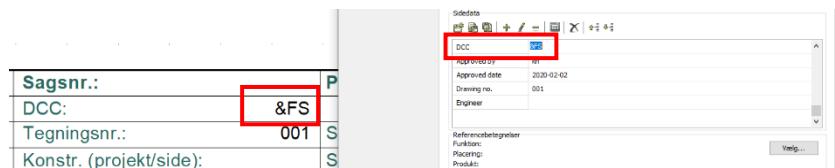
Højreklik på en streg med varedata åbner nu også enten en detailtegning eller record i databasen. Funktionen er på samme måde, som den 'gamle' funktion på symboler.

Der søges først efter en detailtegning (i projektet), derefter i databasen.^{xli}



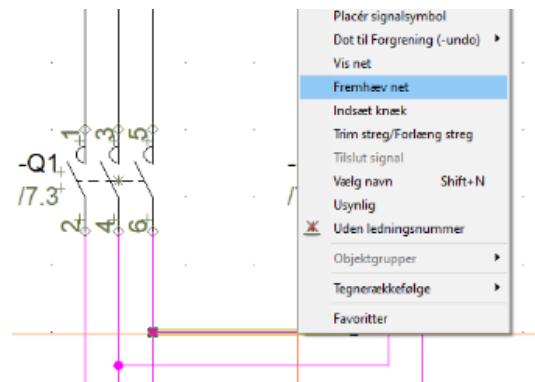
14.20 Dobbeltklik i tegningshovedet åbner side/projektdata direkte

Hvis du dobbeltklikker på et link i tegningshovedet, bliver side- eller projektdata åbnet med fokus på det valgte felt.^{xlii}



14.21 Oprydning i menupunkter og funktioner

Menupunkterne Vis net og Fremhæv net er fjernet fra højreklikmenuen. Funktionen findes i stedet i Netnavigator, hvorfra du også kan udskrive nettet, hvis det ønskes.^{xliii}



15 KONVERTERING AF TEKSTER – FRA GAMLE TEKSTJUSTERINGER TIL NYE

Vi har konverteret tekstjusteringer på følgende måde:

Vi har flyttet alle +'er op i top, og har placeret selve teksten på samme position som før. Tilslutningsnavne er dog stadig justeret i bunden, da de skal justere sig efter stregen, de står på.

Hvis man i sine lister har forskellige teksthøjder, kan det betyde, at man bliver nødt til at ændre listen, hvis datafelterne ikke længere står på linje.

Vi har lavet forskellige efterjusteringer, sådan at man kun (meget) sjældent vil have brug for at ændre selve liste-symbolet.

15.1 Indlæsning af listesider

Når man designer en liste (tegningshoved), er der nogle regler, som skal følges:

- alle datafelter for et datasæt SKAL ligge på linje (i eksemplerne nedenfor er et datasæt forskellige data for samme navn)
- datafelterne ligger på linje, når +'erne ligger på linje
- første datafelt skal have egenskaben Aktiver næste
- datafelter kan være fra projektet eller fra databasen

OG ... man vil jo gerne fortsat have listen til at fungere som tidligere.

Læg mærke til +'erne i billedet:
Dette er den 'gamle' liste, som umiddelbart fungerer fint. (+ er på feltets nederste del).

Listen er designet til at hver række kun fylder en linje, så vi laver intet om. Hvis indholdet mod forventning fylder mere, så vokser teksten opad.

Dette er en anden liste. Her er hver række designet til at fyde to linjer. Hvis indholdet fylder mere end én linje, så vokser teksten den forkerte vej – opad – derfor laver vi en liste af denne type om, sådan at +'erne er ovenover selve feltet. For så vil næste linje være rette sted.

| | | | |
|--------------|--------|------|----------|
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |

| | | | |
|--------------|--------|------|----------|
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |
| Navn. | Varenr | Type | FABRIKAT |

16 FORMULA EDITOR FOR LINE AND SYMBOL DATA FIELDS

16.1 General

An expression may consist of constants (texts, numbers or Boolean constants), variables and functions.

Supported are, arithmetic operators, boolean operators, comparison operators, equality operators, and string operator.

All function names, variable names, constant names and operators are case insensitive.

Please note that the expression must return a text value to be valid.

In order to do expressions with arithmetic sub expression, where data fields are included, one must convert the content of the data field to a **numeric value**, before it could be used in a sub arithmetic expression.

E.g. if the data field named: '**Diameter**' contains the text: '4', this text value must be converted to a number before using it in a sub arithmetic expression.

So, a sub arithmetic expression like:

`3.141 * VAL(DATAFIELD('Diameter'))`

is valid, but:

`3.141 * DATAFIELD('Diameter')`

is invalid since one cannot multiple 3.141 with the text '3'.

Since the result of a data field expression must be a text, all results of sub arithmetic expressions **must be converted from numbers to texts**, if they are going to be part of the result.

So, a data field expression like:

`'Area: ' + FORMAT(POW(VAL(DATAFIELD('Diameter')) / 2, 2) * 3.141) + ' mm2'`

is valid, but:

`'Area: ' + POW(VAL(DATAFIELD('Diameter')) / 2, 2) * 3.141 + ' mm2'`

is invalid, since only element of same type could be added together.

Predefined Boolean constants:

TRUE : Boolean true value

FALSE : Boolean false value.

Character strings must be encapsulated by the character: ' if a ' is to use in a text it must be done by specifying two ' in sequence.

So, a result test like: **Don't do this** should be specified as: '**Don't do this**'

Numbers are implemented as **floating point value**. The decimal separator must be . (a period).

The range support is: **-1.79e-308** to **-1.79e+308**, with 15 to 16 significant digits.



16.2 Operators

Operators behave like predefined functions that are part of the data field expression language.

For example, the expression: **1 + 2**, is build from two number: **1** and **2** and the **+** operator. This expression would return the number **3**.

All operators except **+**, **-**, and **not** are demanding two operands, the **+** and **-** and **not** may be used as unary operator in front of expressions or numbers.

So: **-(7 + 1)** would return the value **--8** and **not (1 < 2)** would return **false**.

To prioritize certain section of an equation one may use **(..)** to encapsulate sections which needs higher priority.

So, an expression like **1 + 2 * 3** would result in the number **7**, where as **(1 + 2) * 3** would result in the number **9**.

So, for expressions with arithmetic operators: ***** and **/** are always solved before: **+** and **-** operators.

Boolean operator **and** is solved before **xor** which is solved before **or**.

The operator groups are solved in the following sequence: **arithmetic operators**, **comparison operators**, **equality operators** and last **Boolean operators**.

16.2.1 Arithmetic operators:

| Operator | Operation | Operand type | Example |
|----------|----------------|--------------|--------------------|
| + | Addition | Number | 1.2 + 5 |
| - | Subtraction | Number | 3.5 - 2.1 |
| * | Multiplication | Number | 1.21E3 * 12 |
| / | Division | Number | 355 / 113 |

Example for addition (+):

FORMAT(VAL(DATAFIELD('Custom 1')) + VAL(DATAFIELD('Custom 2')))

The calculation result will be displayed as text in the data field where this formula is created for.

Example for subtraction (-):

FORMAT(VAL(DATAFIELD('Custom 1')) - VAL(DATAFIELD('Custom 2')))

The calculation result will be displayed as text in the data field where this formula is created for.

Example multiplication (*):

FORMAT(VAL(DATAFIELD('Custom 1')) * VAL(DATAFIELD('Custom 2')))

The calculation result will be displayed as text in the data field where this formula is created for.

Example division (/):

FORMAT(VAL(DATAFIELD('Custom 1')) / VAL(DATAFIELD('Custom 2')))

The calculation result will be displayed as text in the data field where this formula is created for.

16.2.2 Boolean operators:

| Operator | Operation | Operand type | Example |
|----------|-----------------------|--------------|---------|
| not | Negation | Boolean | Not A |
| and | Conjunction | Boolean | A and B |
| or | Disjunction | Boolean | A or B |
| Xor | Exclusive disjunction | Boolean | A xor B |

Expressions with Boolean operators **and** and **or** are always short circuited.

So, an **AND** expression like: $(7 < 2)$ and $(3 < 4)$ would only solve the first part $(1 < 2)$ and since this is false it would skip the remaining part.

Likewise, an **OR** expression like: $(7 > 3)$ OR $(1 < 2)$ would only solve the first part $(7 > 3)$ and since this is true it would skip the remaining part.

16.2.3 Comparison operators:

| Operator | Operation | Operand type | Result type | Example |
|----------|--------------------------|------------------|-------------|------------------|
| > | Greater than | String or Number | Boolean | 233 > 122 |
| >= | Greater than or equal to | String or Number | Boolean | 'ABC' >= 'ABCD' |
| < | Less than | String or Number | Boolean | Name < 'HELLO' |
| <= | Less than or equal to | String or Number | Boolean | VAL('1212') <= 2 |

Both side of a comparison operator must be of same type.

16.2.4 Equality operators:

| Operator | Operation | Operand type | Result type | Example |
|----------|------------|------------------|-------------|-----------------|
| = | Equality | String or Number | Boolean | 233 = 122 |
| <> | Inequality | String or Number | Boolean | 'ABC' <> 'ABCD' |

Both side of an equality operator must be of same type.



16.2.5 String operator:

| Operator | Operation | Operand type | Example |
|----------|---------------|--------------|-------------------|
| + | Concatenation | String | 'Hello' + 'world' |

16.2.6 Variable:

The number of variables is fixed to 4 predefined variables.

For a symbol expression it is the default symbol fields (Name, ...).

For a line expression it is the default line fields (Name, ...).

In the formula editor one can select the valid ones from the combo box in the group **Symbol data** for symbol expression or **Line data** for line expression.

16.3 Functions:

The expression editor supports the following functions:

16.3.1 Function: DATAFIELD

Returns a text value of a used defined data field.

Syntax: **DATAFIELD('name')**, where name is a text constant.

Example: If a data field named **Diameter** contains the value text **123**, the **DATAFIELD('Diameter')** would return the text value **123**.

The name of the data field is case insensitive.

Exceptions: If the name is not a valid data field name an **Undefine data field error** will be returned.

16.3.2 Function: VAL

Returns the number of a text value if possible.

The **VAL** function accepts either comma or point as decimal separators, but not any thousand separators. The exponential notation is supported as well.

Syntax: **VAL(value)**, where value is a string.

Examples:

- **VAL('-123,3')** or **VAL('-123.3')** would both return the number **-123.3**
- **VAL(DATAFIELD('Pressure'))** would, if the value of data field '**Pressure**' is the text **16** return the number **16**.

- `VAL(DATAFIELD('Pressure'))` would, if the value of data field '**Pressure**' is the text **16bar** raise a **Not a valid number exception**.

Exceptions: If the specified text value is not a valid number or is out of supported number range.

16.3.3 Function: ISVAL

Returns a boolean true value if the value could be converted to a number and a Boolean false value otherwise.

The **ISVAL** function accepts either comma or point as decimal separators, but not any thousand separators. The exponential notation is support as well.

Please notice that the result of this function can only be used in Boolean sub equations. Boolean values cannot be converted to texts.

Syntax: **ISVAL(value)**, where value is a string.

Examples:

- `ISVAL('-123,3')` would return a Boolean **true** value, but `ISVAL('HELLO')` would return Boolean **false** value.
- `ISVAL(DATAFIELD('Pressure'))` would, if the value of data field '**Pressure**' is the text **16**, return a logical **true** value.
- `ISVAL(DATAFIELD('Pressure'))` would, if the value of data field '**Pressure**' is the text **16bar**, return a logical **false** value.
- `IF(ISVAL(DATAFIELD('Pressure')), 'A number', 'Not a number')`, would, if the value of data field '**Pressure**' is the text **16**, return '**A number**'.
If the value of data field '**Pressure**' is the text **16bar**, the result would be '**Not a number**'.

Exceptions: None

16.3.4 Function: FORMAT

Returns a string with the specified format.

Syntax: **FORMAT(number)** or **FORMAT(format, number)**, where number is the number to format and format is a string with the desired format.

If the format is not included, the format is **%g**

The allowed format is:

`"%" ["-"] [width] ["."] prec] type`

A format specifier begins with a **%** character. After the percent sign come the following elements, in this order:

1. An optional left justification indicator, `["-"]`



2. An optional width specifier, [width].
3. An optional precision specifier, [". " prec].
4. The conversion type character, type.

The following types are possible:

| Value | Meaning |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| e | Scientific The argument must be a floating-point value. The value is converted to a string of the form " -d.ddd...E+ddd ". The resulting string starts with a minus sign if the number is negative. One digit always precedes the decimal point. The total number of digits in the resulting string (including the one before the decimal point) is given by the precision specifier in the format string; a default precision of 15 is assumed if no precision specifier is present. The "E" exponent character in the resulting string is always followed by a plus or minus sign and at least three digits. |
| f | Fixed The argument must be a floating-point value. The value is converted to a string of the form " -ddd.ddd... ". The resulting string starts with a minus sign if the number is negative. The number of digits after the decimal point is given by the precision specifier in the format string—a default of 2 decimal digits is assumed if no precision specifier is present. |
| g | General The argument must be a floating-point value. The value is converted to the shortest possible decimal string using fixed or scientific format. The number of significant digits in the resulting string is given by the precision specifier in the format string; a default precision of 15 is assumed if no precision specifier is present. Trailing zeros are removed from the resulting string, and a decimal point appears only if necessary. The resulting string uses the fixed-point format if the number of digits to the left of the decimal point in the value is less than or equal to the specified precision, and if the value is greater than or equal to 0.00001. Otherwise the resulting string uses scientific format. |

Examples:

- **FORMAT(123.3)** would return the string **123,3** or **123.3** depending on the region settings in windows.
- **FORMAT('%.4f', 123.3)** would return the string **123,3000** or **123.3000** depending on the region settings in windows.
- **FORMAT('%.0f', 123.5)** would return the string **124**.
- **FORMAT('%e', 122.5)** would return the string **1,2250000000000E+002** or **1.2250000000000E+002** depending on the region settings in windows.

Exceptions: None

16.3.5 Function: EXP

Returns the exponent of the value where the base is e.

Syntax: **EXP(value)**, where value is a number.

Example:

- **EXP(1)** would return the number **2.71828182845905**
If the result is too big, an infinite number would be returned.
- **FORMAT(EXP(VAL(DATAFIELD('Diameter'))))**
The value of data field 'Diameter' will be used as base of e.
The calculation result will be displayed as text in the data field where this formula is created for.

Exceptions: None

16.3.6 Function: POW

Returns base raised to the exponent.

Syntax: **POW(base, exponent)**, where base and exponent are both numbers.

Example:

- **POW(10, 2)** would return the number: 100
- **FORMAT(POW(VAL(DATAFIELD('Diameter')), VAL(DATAFIELD('Custom 1'))))**
For base the value of data field 'Diameter', and for exponent the value of data field 'Custom 1' will be used. The calculation result will be displayed as text in the data field where this formula is created for.

Exceptions: If the result is too big, a positive or negative infinite number would be returned, based on the sign of the base.

16.3.7 Function: SQRT

Returns the square root of the value.

Syntax: **SQRT(value)**, where value is a number.

Example:

- **SQRT(81)** would return the number: 9
- **FORMAT(SQRT(VAL(DATAFIELD('Diameter'))))**
Here, the value from data field 'Diameter' will be used for calculation of square root.



- **FORMAT(SQRT(VAL(DATAFIELD('Diameter')))) * VAL(DATAFIELD('Custom 3'))**
Here, the result of SQRT from data field 'Diameter' is directly multiplied with value from data field 'Custom 3', and the calculation result is displayed as text in the data field where this formula is created for.

Exceptions: if value is negative a NaN (Not a Number) number would be returned.

16.3.8 Function: SIN

Returns the sine of the value in degrees.

Syntax: **SIN(value)**, where value is a number.

Example:

- **SIN(45)** would return the number **0.707106781186547**
- **FORMAT(SIN(VAL(DATAFIELD('Angle'))))**
The value from data field 'Angle' will be used for sine calculation, and for display of calculation result as text in the data field where this formula is created for.

Exceptions: None

16.3.9 Function: COS

Returns the cosine of the value in degrees.

Syntax: **COS(value)**, where value is a number.

Example:

- **COS(45)** would return the number **0.707106781186547**
- **FORMAT(COS(VAL(DATAFIELD('Angle'))))**
The value from data field 'Angle' will be used for cosine calculation, and for display of calculation result as text in the data field where this formula is created for.

Exceptions: None

16.3.10 Function: TAN

Returns the tangent of the value in degrees.

Syntax: **TAN(value)**, where value is a number.

Example:

- **TAN(45)** would return the number **1**

- **FORMAT(TAN(VAL(DATAFIELD('Angle'))))**

The value from data field 'Angle' will be used for tangent calculation, and for display of calculation result as text in the data field where the formula is created for.

Exceptions: None

16.3.11 Function: ASIN

Returns the principal angle in degrees of the inverse sine of the value in the range -1 to 1, both included.

Syntax: **ASIN(value)**, where value is a number.

Example:

- **ASIN(0.707106781186547)** would return the number **45**

- **FORMAT(ASIN(VAL(DATAFIELD('Angle'))))**

The value from data field 'Angle' will be used for inverse sine calculation, and for display of calculation result as text in the data field where the formula is created for.

Exceptions: If value is outside the range -1 to 1 a NaN (Not a Number) number would be returned.

16.3.12 Function: ACOS

Returns the principal angle in degrees of the inverse cosine of the value in the range -1 to 1, both included.

Syntax: **ACOS(value)**, where value is a number.

Example:

- **ACOS(0.707106781186547)** would return the number **45**

- **FORMAT(ACOS(VAL(DATAFIELD('Angle'))))**

The value from data field 'Angle' will be used for inverse cosine calculation, and for display of calculation result as text in the data field where the formula is created for.

Exceptions: If value is outside the range -1 to 1 a NaN (Not a Number) number would be returned.



16.3.13 Function: ATAN

Returns the principal angle in degrees of the inverse tangent of the value.

Syntax: ATAN(value), where value is a number.

Example:

- ATAN(1E100) would return the number 90

Exceptions: None.

16.3.14 Function: ABS

Returns the absolute value of the value.

Syntax: ABS(value), where value is a number.

Example:

- ABS(-10) would return the number 10

Exceptions: None.

16.3.15 Function: LN

Returns the natural logarithm of the value.

Syntax: LN(value), where value is a number.

Example:

- LN(2.71828182845905) would return the number 1

Exceptions: if value is negative a NaN (Not a Number) number would be returned, if value is zero an infinite negative number would be returned.

16.3.16 Function: LOG

Returns the logarithm with base 10 of the value.

Syntax: LOG(value), where value is a number.

Example:

- LOG(100) would return the number 2

Exceptions: if value is negative a NaN (Not a Number) number would be returned, if value is zero an infinite negative number would be returned.

16.3.17 Function: TRUNC

Returns the truncated number.

Syntax: **TRUNC(value)**, where value is a number.

Example:

- **TRUNC(100.999)** would return the number **100**
- **FORMAT(TRUNC(VAL(DATAFIELD('Custom 5'))))**
The value from data field 'Custom 5' will be used for truncation, and for display of result as text in the data field where the formula is created for.

Exceptions: None.

16.3.18 Function: ROUND

Returns the rounded value.

Syntax: **ROUND(value)**, where value is a number.

Example:

- **ROUND(100.5)** would return the number **101**
- **FORMAT(ROUND(VAL(DATAFIELD('Custom 5'))))**
The value from data field 'Custom 5' will be used for rounding, and for display of result as text in the data field where the formula is created for.

Exceptions: None.

16.3.19 Function: IF

Return either the true or false result based on the condition

Syntax: **IF(condition, true_expression, false_expression)**, where condition is a Boolean expression and **true_value** and **false_value** is either text or number values.

Examples:

- **IF(ISVAL(DATAFIELD('Test1')) AND VAL(DATAFIELD('Test1')) > 100, 'Yes', 'No')**
would return the text value 'Yes', if the data field contains the text value **101**, but if the data field contained the text value **100**, the result would be the text value 'No'. Likewise the result would be 'No', if the data field contained a text value which could not be converted to a valid number, like e.g. 'hello'



If the order of the elements in the condition part is switched like:

- **IF(VAL(DATAFIELD('Test1')) > 100 AND ISVAL(DATAFIELD('Test1')), 'Yes', 'No')**
the expression is only valid, if the data field '**Test1**' contained a text value which could be converted to a valid number, like e.g. **123**

So if the data field contained the value '**ABC**', the result would be **!Invalid!**, since the **VAL(DATAFIELD('Test1'))** function would throw a **Not a valid number expression.**

This is not the case in the first example, since a logical **AND** operator is short circuited, when the first Boolean **false** result is met, and therefore the **VAL(DATAFIELD('Test1'))** part is never executed.

- This short circuit also applies for the **IF** function.

So, if the condition is **true**, only the **true_expression** would be executed and if the condition is **false**, only the **false_expression** would be executed.

So, if one wants to ensure that the value of a given data field is only used in an arithmetic sub expression, if it contains a valid text, which could be converted to a number, one could specify a data field expression like:

IF(ISVAL(DATAFIELD('Test1')), FORMAT(VAL(DATAFIELD('Test1')) * 2), 'Invalid number in data field: Test1')

So if the value of the data field named '**Test1**' contains the text value **4**, the result would be **4**, and if the data field contained a text value which could not be converted to a number, the result would be '**Invalid number in data field: Test1**'.

Exceptions: None.