

Innhold

1 HouseModeller3D - Symbolspråket	1
1.1 Innledning	1
1.2 Generelt om Symbolfiler, Grupper, og Symboler	1
1.3 DrawType	2
1.4 Parametre	3
1.5 Instruksjoner	3
1.6 Instruksjoner som utgår	13
1.7 Uttrykk i symboler	14
1.8 Vinduer og Dører	18
1.9 Tilleggssymboler rundt åpninger	20
1.10 Bånd v.h.j.a. symboler	21
1.11 Oppriss åpningsmarkering	22
1.12 Takkantsymboler	22
1.13 Terrengsymboler på fasade	23

1 HouseModeller3D - Symbolspråket

1.1 Innledning

Dette skrevet er en kort beskrivelse av symbolspråket i HouseModeller3D programmet. Dette er en referansemanual. For å forstå hvordan språket er tenkt brukt kan det være til hjelp å se på eksempler. Eksempler på de fleste muligheter i symbolspråket finnes blant de mange symboler definert på leksikon i standard leveransen. Du kan bruke Symboldefinisjoner kommandoen i Vinduer menyen for å studere dem. Alternativt kan du bruke Liste kommandoen i Utfør menyen for å lage en tekstfil med listing av alle symboldefinisjoner.

Symbolreferanser legges til på huset og på etasjene for uttegning på planer og fasader/oversikter. I tillegg er det et eget opplegg for annotering av fasader, snitt, og oppriss tegninger. Det finnes også spesialiserte symboler i egne grupper, f.eks. vinduer, takkanter, åpningsbånd.

De fleste symboler har en såkalt DrawType som brukes til å styre tegnemetode på de forskjellige tegningstyper. Andre, spesialiserte, DrawTypes brukes f.eks. til henting av variabler fra symbolet.

For å beskrive syntaksen av instruksjoner og uttrykk, brukes en enkel notasjon. Navn som skal skrives inn direkte slik de står vises i skrifttypen *courier*. Navn som beskriver det som skal skrives inn vises i skrifttypen *courier kursiv*. Termer som kan utelates omslutes av hakeparentes []. Termer som kan repeteres mange ganger etterfølges av tre prikker (...). F.eks. [*uttrykk*]... betyr at *uttrykk* kan forkomme 0 eller flere ganger.

1.2 Generelt om Symbolfiler, Grupper, og Symboler

Symboler refereres ved navn som består av to deler: gruppenavn og symbolnavn. Samme symbolnavn kan forkomme med forskjellige gruppenavn. Symboldefinisjonene lagres på leksikon. Enkelte symboldefinisjoner kan overstyres på husfilene om nødvendig.

Når et symbol defineres angis gruppenavnet. Settet av definerte grupper oppstår som kombinasjonen av alle grupper i alle symboldefinisjoner, dvs. gruppene ikke er predefinerte. En ny gruppe opprettes enkelt ved å definere et symbol med det ønskede gruppenavn.

Man kan definere at et symbol er usynlig. Dersom alle symboler i en gruppe er usynlige, vil heller ikke gruppen komme med i menyene. Dette brukes for å definere delsymboler (funksjoner) som kan kalles fra andre symboler.

De fleste tegn kan brukes i et gruppe- eller symbolnavn, men følgende tegn er pr. i dag ikke tillatt: kolon (:), parenteser (), semikolon (;), prosent (%), understrek (_). Det er lurt å holde seg til alfanumeriske tegn da settet av ikke-tillatte tegn kan bli revidert i framtiden.

For å kunne ha flere språk, har hvert symbol to gruppenavn og to symbolnavn. Den ene er id-navn, som brukes for å identifisere symbolet. Disse navn må brukes i "call" instruksjoner. De andre navn er de som vises i brukergrensesnittet. Slike

brukernavn blir forskjellige for de forskjellige språk. Når HouseModeller3D leser inn leksikon, setter den opp disse brukernavn i riktig språk. Hvis den er tom, blir den det samme som id-navn. Brukernavnene bør være tomme på leksikon, slik at navnene forekommer på språkfilene.

Innholdet i hvert symbol består av tre hoveddeler: DrawType definisjon, parameterdefinisjoner og instruksjonene. Instruksjonene utføres når symbolet tegnes (når symbolet blir aktivisert eller “kalt” fra HouseModeller3D).

Prosenttegnet brukes for kommentarer. Hele linjen etter prosenttegnet ignoreres.

1.3 DrawType

DrawType *type* [, *type*]... DrawType deklarasjon

DrawType må være første linje i symbolinnholdet. En liste av tall angir i hvilke tilfeller symbolet skal tas hensyn til. Hver situasjon i HouseModeller3D hvor et symbol kan kalles har sin egen kode, f.eks. har hver tegningstype sin egen kode. “DrawType 1,24,25” sier at symbolet tegnes på plantegninger, fasade og oversikt.

Ved å referere til (den globale) variabelen DrawType kan symbolet selv finne ut hva slags tegning symbolet ble kalt fra.

DrawType-linjen tas ikke hensyn til i delsymboler (funksjoner). Det er vanlig å utelate DrawType-linjen på symboler som bare brukes som funksjoner, f.eks. `g_rekt`.

DrawType-koder for nye symboler		
1	Plantegning	
6	Oppriss (3D symboler)	Ny 3D type
7	2D annoteringer på fasade og snitt	
15	Oversikt	Erstatter 10
24	Fasade	Erstatter 4
25	Snitt	Erstatter 5
10000	Hent variabler (for dører og vinduer)	

Følgende DrawType-kodene brukes bare i “storebroren” HamboHus, men forekommer i symboldefinisjonene:

DrawType-koder bare i HamboHus		
30	Tegningsark forgrunn/bakgrunn	
200	Tittelfelt	
201	Vindu-/Dørtabell	
202	Takstoltablett	Ikke i bruk
203	Bjelkelagstabelt	
204	Opprissstabelt	
220	Bygges	

Pga moderniseringer i 3D tegnemåte, finnes det noen gamle DrawTypes som tegner i to faser. Disse er fortsatt i bruk for gamle symboler, men bør ikke brukes i nye symboler:

Gammeldags DrawType-koder		
4	Fasadetegning	fasade i to faser
5	Snitt	snitt i to faser
10	Oversiktstegning, Utvendig	oversikt i to faser

Utgåtte DrawType koder tolkes fortsatt, men de er det samme som en annen DrawType:

Utgåtte DrawType-koder		
2	Bjelkelagsplan	samme som 1
3	Takplan	samme som 1
11	Oversiktstegning, Innvendig	samme som 10
12	Oversiktstegning, Plan	samme som 10
13	Oversiktstegning, Bindingsverk	samme som 10
20	Generell tegning, Bindingsverk (Fasade = Nei)	samme som 10
21	Generell tegning, Fasade (Fasade = Ja)	samme som 10
31	Tegningsark og grafikk bakgrunn	samme som 30 og 1

Opprinnelig ble DrawType brukt til å skille mellom mange forskjellige 3D tegningstyper. Senere ble lag introdusert for å dekke en del av det samme. F.eks. kan man skru av og på bindingsverk på oversikt v.h.j.a. lag. Dermed ble DrawType 11,12,13,20, og 21 overflødige og de blir nå tolket som 10. HouseModeller3D bruker DrawType 10 internt for alle disse.

Likeledes brukes lag til å bestemme hva som kommer med på plan og bjelkelagstegninger. DrawType 2 brukes ikke lenger, og HouseModeller3D tolker den som 1.

Før ble 3D tegninger gjort i to faser, en “shield”-fase og en “tegne”-fase. I dagens HouseModeller3D er tegnemåten endret slik at uttegningskoden hele tiden har en komplett “modell” av det som skal tegnes. Dermed finnes det ikke lenger to faser og heller ikke shield begrepet.

For å beholde kompatibilitet er de gamle DrawTypes 4, 5 og 10 beholdt med to faser, dvs. symboldefinisjoner skrevet med to faser vil fortsette å fungere. HouseModeller3D tegner da symbolet to ganger etter hverandre, med fase = 1 og så med fase = 2. Gamle “shields” blir polygoner. Når symboler skrives om eller når det lages nye symboler, bør man gå over til den nye tegnemåten uten faser. Da brukes de nye DrawTypene 15, 24, og 25. M.a.o. skal innholdet for DrawType 4, 5, eller 10 være den gamle to-fase tegnemåte, mens DrawType 15, 24, og 25 skal inneholde de nye polygon instruksjonene.

“Shields” skal ikke lenger forekomme på plantegninger (de vil ikke bli tatt hensyn til) - den nye tegnemåten er med fylte symboler. Det er gjort spesielle grep (ForcePolygonFill) for at gamle symboler som hadde shields skal kunne fungere som før. Dette gjelder dette for følgende symboler: Ventil i vegg, Miniavløp, Sikringsskap, Ettløps Leca, og Dobbeltløpet Leca.

Evt. oppriss-symboler i 3D (DrawType 6) må være den nye typen uten faser (gamle to-fase symboler kan ikke brukes).

DrawType 7 brukes for 2D annoteringer på 3D tegninger, f.eks. mållinjer på snitt. Plassering av symbolet er som regel i 3D, mens tegning foregår i 2D “opp-på” 3D tegningen. Selve symboldefinisjonene skal bare inneholde instruksjoner i 2D.

DrawType 10000 brukes av HouseModeller3D for å hente data fra et symbol. Dette brukes i HouseModeller3D 1.2 for vinduer og dører - symbolene oppgir selv dimensjoner på utsparinger, o.l.

Symboler som tegnes under eksport til DXF har som regel variabelen modelExport definert og satt til 1. Dette gjør det mulig å ha forskjellige oppførsel for eksport og for vanlig uttegning. Funksjonaliteten brukes bl.a. til å la være å tegne “pynt”, f.eks. mursteinsmønster på pipe, ved eksport.

1.4 Parametre

```
par navn [ tekst ] [ real ] [= standardverdi ]
par navn [ tekst ] int [= standardverdi ]
par navn [ tekst ] enum tekstliste [= standardverdi ]
par navn [ tekst ] mat materialkodeliste [= standardverdi ]

tekstliste = ( tekst [= verdi ] [, tekst [= verdi ] ] )
materialkodeliste = ( materialkode [ : materialkode ] [, materialkode [ : materialkode ] ] )
```

Med parameterdefinisjonene, som kommer før instruksjonene i symbolinnholdet, deklarerer hvilke parametre som angis når symbolet kalles (fra spesifikasjoner på husfilene eller fra andre symboler ved “call”). *Navn* er variabelnavnet som brukes for å referere til parameteren i instruksjonene, mens *tekst* er teksten som kommer fram på skjermen når man spesifiserer et symbol på husfilen. Alle parametrene er reelle tall, men de kan skrives ut og/eller velges på forskjellige måter. Parametertypene er reelt tall, heltall, liste av navngitte verdier, og materialkoder.

I HouseModeller3D 1.2 kan et symbol ha opp til 20 parametre.

1.5 Instruksjoner

En instruksjon begynner med et navn. Store og små bokstaver oppfattes i HouseModeller3D 1.2 som like, men dette kan bli endret i framtiden. Argumentene adskilles med komma. Kommandoer adskilles fra neste kommando med semikolon (;). Semikolon er nødvendig på slutten av en linje.

Symbolspråket har et begrep om nåværende posisjon, kalt pennposisjon. Mange instruksjoner, f.eks. linjetegning og kall av funksjoner, starter fra nåværende posisjon. Når man arbeider med symboldefinisjoner, må man hele tiden vite hvor “pennen” er. Ellers er det god stil å ikke satse på pennposisjonen unødig, f.eks. bør det gjøres en flytt instruksjon før man begynner å tegne.

Mange av instruksjonene kan forekomme både i 2- og 3-dimensjonal form. Dersom z-koordinaten utelates betyr det at pennen skal holde seg i samme plan som sist spesifiserte z-koordinat (i motsetning til å flytte til $z=0$ planet). Tidligere var det veldig viktig å ikke bruke den 3-dimensjonale form for 2D-symboler. Dette er ikke like viktig nå (fordi HouseModeller3D er endret slik at symboler i 2D ikke kan ødelegge for hverandre), men det er likevel god form at man aldri bruker z-koordinat i instruksjonene for DrawType 1 og 7.

Hvis ikke annet er angitt er koordinater og lengder i meter, og vinkler i grader.

<code>set <i>variable</i> = verdi</code>	sett lokal variabel
<code>cset <i>variable</i> = verdi</code>	sett global variabel

Set og cset tilordner verdien til variabelen. Lokale variabler, som settes med `set`, eksisterer bare i det symbolet der de er definert. Globale variable, som settes med `cset`, eksisterer også inne i delsymboler (funksjoner). Forskjellen mellom `set` og `cset` kommer bare fram når nye variable opprettes. En eksisterende variabel kan settes både med `set` og `cset`, men vil ikke forandre status fra lokal til global, eller omvendt, av den grunn. (En del globale variable defineres av HouseModeller3D programmet før den tegner et symbol. Disse beskrives nedenfor.)

I nyere versjoner av HouseModeller3D kan variabler inneholde både tall og nye datatyper, f.eks. tekster, objekter og arrayer.

<code>ma <i>x</i>, <i>y</i> [, <i>z</i>]</code>	flytt absolutt (move absolute)
---	--------------------------------

Flytter pennen til angitt koordinat. Hvis *z* utelates endres ikke z-koordinaten (flytting i x-y planet).

<code>mr <i>dx</i>, <i>dy</i> [, <i>dz</i>]</code>	flytt relativt (move relative)
--	--------------------------------

Flytter pennen et stykke fra der den befinner seg. Hvis *dz* utelates endres ikke z-koordinaten (flytting i x-y planet).

<code>la <i>x</i>, <i>y</i> [, <i>z</i>]</code>	linje absolutt (line absolute)
---	--------------------------------

Tegner en linje fra der pennen befinner seg til angitt koordinat. Hvis *z* utelates tegnes linjen i x-y planet dvs. z-koordinat endres ikke.

Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

<code>lr <i>dx</i>, <i>dy</i> [, <i>dz</i>]</code>	linje relativt (line relative)
--	--------------------------------

Tegner en linje et stykke fra der pennen befinner seg. Hvis *dz* utelates tegnes linjen i x-y planet dvs. z-koordinat endres ikke.

Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

<code>circle <i>radius</i></code>	sirkel
-----------------------------------	--------

Tegner en sirkel med angitt radius. Sirkelens senter er der pennen befinner seg. Pennposisjonen etter at sirkelen er tegnet er også sirkelens sentrum, dvs. pennposisjonen påvirkes ikke av denne instruksjonen. Sirkelen tegnes i x-y planet. HouseModeller3D tegner sirkler v.h.j.a. Bézierkurver slik at de fungerer på 3D tegninger når koordinatsystemet er transformert til sirkelens plan.

Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

<code>arc <i>radius</i>, <i>startVinkel</i>, <i>sluttVinkel</i></code>	bue
--	-----

Tegner en bue med *radius* fra *startVinkel* til *sluttVinkel*. Retningen avhenger av fortegnet av (*sluttVinkel* - *startVinkel*): positiv betyr mot klokka, negativ betyr med klokka. Buens senter er der pennen befinner seg. Pennposisjonen etter at buen er tegnet er sluttpunktet av buen. HouseModeller3D tegner ark v.h.j.a. Bézierkurver slik at de fungerer på 3D tegninger når koordinatsystemet er transformert til arkets plan (i motsetning til tidligere versjoner der de fungerte bare på DrawType 1, 2, og 7.).

Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

```

polygon [ nøkkelord ["=" uttrykk]...koordinat...endpolygon          polygon
nøkkelord: "stroke" "fill" "nofill" "nostroke" "thick" "linetol" "dosnitt" "nosshield"
pt x, y [, z]                                                    et punkt på polygonen

```

Polygon instruksjonen brukes å tegne polygonflater og kanter, samt "tykke" (extruded) polygoner. Instruksjonen er den moderne måten å tegne på (uten faser) og erstatter tidligere shield instruksjoner med tilhørende linjer.

Mellom polygon og endpolygon listes punktene på polygonen med pt instruksjoner. Retning spiller ingen rolle for tegning, men det er en konvensjon at polygoner defineres mot sola. Det er lurt å følge konvensjonen da retningen kan tenkes brukt i framtiden til å bestemme forside og bakside av flaten.

Nøkkelordene brukes til å gi ønsket virkning. Evt. uttrykk angir sant (true eller 1) eller usant (false eller 0), unntatt tykk og linetol som har tall parameter. Hvis uttrykk etter nøkkelord mangler, blir det tolket som sant.

Polygonen kan være fylt (fill) eller tegnet med linjer (stroke) eller begge deler. Den fylles før eventuelle kanter tegnes.

Polygonen kan tegnes som et volum med tykkelse (extruded). Bruk thick= tykkelsen i meter. Dersom en toleranse spesifiseres med (linetol), utelates linjene i tykkelsen dersom vinklene mellom linjene er slik at avvik til en rett linje er mindre enn toleransen. Toleransen oppgis i radianer. Dette brukes f.eks. på buede vinduer for å unngå linjer i tykkelse der buen deles opp. Nøkkelordene thick og linetol virker på samme måte som tykk og toleranse i gamle shield-instruksjoner.

Når polygoner med tykkelse brukes i 3D, kan de også tegnes på snitt. Dersom dosnitt er sant, blir volumet snittet langs snittlinjen. Dette brukes f.eks. ved snitt av vindskier. Dersom dosnitt ikke er sant, tegnes hele 3D volumet.

Nøkkelordet nosshield brukes for å få en gjennomsiktig polygon i 3D. Dette brukes på polygoner importert under IFC-import.

Eksempel, innmaten av g_rektmedfyll:

```

par l
par b
polygon fill stroke; pt 0,0; pt 1,0; pt 1,-b; pt 0,-b; endpolygon
ma 0,0

```

Eksempel fra innmaten av g_RekkverkstenderNy:

```

polygon stroke fill thick=-dyb
  pt 0,0,0
  pt tyk,0,delta
  pt tyk,0,h
  pt 0,0,h-delta
endpolygon

```

Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

```

bezier [ nøkkelord ["=" uttrykk]...koordinat...endbezier          kurve
nøkkelord: "stroke" "fill" "nofill" "nostroke" "debug"
pt x, y [, z]                                                    et punkt i kurvedefinisjonene

```

Bezier instruksjonen brukes å tegne kurver og flater med kurvede kanter. Instruksjonen erstatter den gamle draw instruksjonen.

Mellom bezier og endbezier listes punktene på kurven v.hj.a. pt instruksjoner.

Kurven kan være fylt (fill) eller tegnet med linjer (stroke) eller begge deler. Kurven fylles før kantene tegnes.

Nøkkelordene brukes til å gi ønsket virkning. Evt. uttrykk angir sant (true eller 1) eller usant (false eller 0). Hvis uttrykket mangler, blir det tolket som sant.

Nøkkelordet debug brukes for å få tegnet et kryss ved hvert punkt.

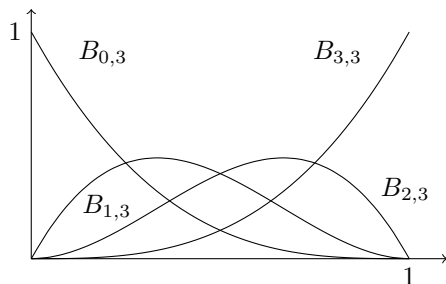
Fire punkter definerer et Bézier kurvesegment. De to endepunktene er punkter på selve kurven, mens de to andre punkter er kontrollpunkter. Disse punktene regulerer kurvens form slik at tangenten til kurvens endepunkt går gjennom kontrollpunktet. Når flere sammenhengende segmenter lages, brukes endepunktet til et segment som startpunktet for neste. På

denne måten blir det slik at hver tredje punkt tolkes som et punkt på kurven, mens de to mellomliggende punktene tolkes som kontrollpunkter. En velformet kurve består av $4 + 3 * n$ punkter ($n \geq 0$).

Matematisk er en Bézier kurve en tilnærming v.h.j.a. Bernstein polynomer. Bernstein n . grads polynomet over intervallet $[0, 1]$ defineres som:

$$B_{k,n} = \binom{n}{k} t^k (1-t)^{n-k} \quad \text{for } 0 \leq t \leq 1$$

Kubiske Bernstein polynomer uttrykkes som:



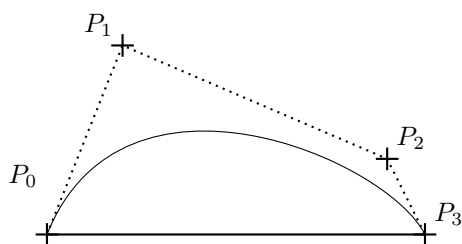
$$B_{0,3} = 1 - 3t + 3t^2 - t^3$$

$$B_{1,3} = 3t - 6t^2 + 3t^3$$

$$B_{2,3} = 3t^2 - 3t^3$$

$$B_{3,3} = t^3$$

Den kubiske Bézierkurven, $P(t)$, fra P_0 til P_3 med kontrollpunktene P_1 og P_2 , er konstruert fra kubiske Bernstein polynomer på følgende måte:



$$x(t) = \sum_{k=0}^3 x_k B_{k,3}(t)$$

$$y(t) = \sum_{k=0}^3 y_k B_{k,3}(t) \quad \text{for } 0 \leq t \leq 1$$

$$z(t) = \sum_{k=0}^3 z_k B_{k,3}(t)$$

hvor x_k , y_k , og z_k er koordinatene for de fire punktene, P_k .

Utvikles uttrykket for x-komponenten får vi:

$$\begin{aligned} x(t) &= x_0 B_{0,3} + x_1 B_{1,3} + x_2 B_{2,3} + x_3 B_{3,3} \\ &= x_0 + (-3x_0 + 3x_1)t + (3x_0 - 6x_1 + 3x_2)t^2 + (-x_0 + 3x_1 - 3x_2 + x_3)t^3 \end{aligned}$$

og tilsvarende uttrykk for y og z-komponentene.

Noen grunnleggende egenskaper ved disse kurvene er:

- En Bézierkurve går gjennom punktene P_0 og P_3 .
- Retningen på vektorene P_0P_1 og P_3P_2 er lik retningen for tangentene i henholdsvis P_0 og P_3 .
- En Bézierkurve befinner seg helt og holdent innenfor den såkalte konvekse rammen som bestemmes av sluttpunktene og kontrollpunktene. I to dimensjoner vil den konvekse rammen være polygonen som formes ved å knytte sammen punktene P_0, \dots, P_3 med en "gummistrikk".

Følgende eksempel tegner en fylt sirkel i 4 deler (innmat av `g_sirkelmedfyll`):

```
par r
set magic = (sqr(2.0) - 1.0) * 4.0 / 3.0
bezier fill stroke
  pt r, 0.0
  pt r, r * magic
  pt r * magic, r
  pt 0, r
  pt -r * magic, r
  pt -r, r * magic
  pt -r, 0
  pt -r, -r * magic
```

```

    pt -r * magic, -r
  pt 0, -r
    pt r * magic, -r
    pt r, -r * magic
  pt r, 0
endbezier
ma 0,0

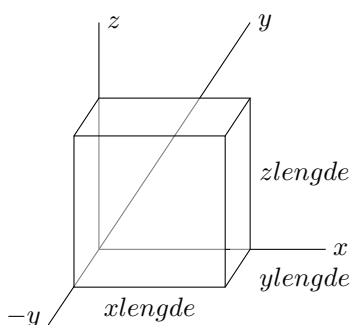
```

Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

box *xlengde, ylengde, zlengde*

boks

Spesialtilfelle for enkle bokser. Boksen plasseres med origo i pennposisjon og aksene som vist nedenfor. Kan lages av andre instruksjoner, men forekommer så ofte at det er laget en egen instruksjon for dette enkle tilfellet. For DrawType 15, 24, og 25 (de nye 3D typer) blir de 6 flatene laget først, så tegnes de 12 linjene. For de gamle 3D DrawTypes, lages shields i fase 1 og linjer i fase 2.



Utseende av linjer og fyll styres vanligvis fra laget symbolet er i, men egenskapene kan overstyres i symboldefinisjonen i spesielle tilfeller. Se egne instruksjoner for dette.

layerapp *lagNummer*

lag utseende (layer appearance)

Setter opp hele utseende – linjer, fyll, og tekst – iht. til det angitte lagnummer. Dette betyr at påfølgende instruksjoner vil fungere som om symbolet var i et annet lag. Hvis lagNummer er -1 blir utseende satt tilbake til laget som symbolet tilhører.

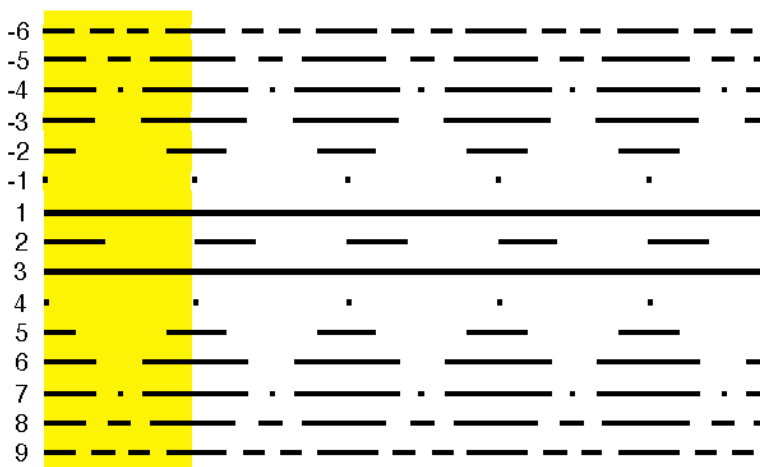
Instruksjonen brukes for symboler som bruker flere utseender, f.eks. sofaer og stoler på anmeldelsestegninger. På denne måten kan de forskjellige utseender styres på vanlig "lagmåten".

Det er avsatt et eget nummerserie for lag referert direkte fra symboler, 8000-8999. Lagdefinisjonene blir lettere å vedlikeholde dersom nye lag legges til i dette området.

linjetype *linjeType, repetisjonsLengde*

linjetype, dvs. stipling

LinjeType tilsvarende linjetyperne på HP plottere med noen små variasjoner. Mulige linjetyper:



Det markerte feltet tilsvarer en periode med lengde repetisjonsLengde. Repetisjonslengden spesifiseres i meter på arket. Tidligere var typene -1 til -6 tilpassede, mens typene 4 til 9 var tilsvarende uten tilpasning. I nyere HouseModeller3D versjoner tilpasses alle linjer, dermed spiller det ingen rolle om negative eller positive tall brukes. (Positive tall foretrekkes.) Tilpasning gjøres slik at linjen består av et helt antall mønstre. Dersom flere linjer er sammenhengende på den måten at neste linje starter der forrige slutter, blir tilpasning regnet ut bare én gang over alle de sammenhengende linjer som helhet. Dette gir god virkning for kurver, men kan gi uønskede effekter (“hull”) i skarpe hjørner, f.eks. hjørner av et rektangel. De uønskede effektene kan unngås ved å tegne linjene i en ikke-sammenhengende rekkefølge.

Dersom en ulovlig *linjeType* spesifiseres, brukes type 1. Den tidligere muligheten for å tegne en linje som en prikk i hver ende finnes ikke i nyere HouseModeller3D versjoner.

Når stipling styres av en stipling spesifikasjon på en husfil eller leksikon, brukes type 1 for heltrukken (stiples nei), og type 2 for stippet (stiples ja). Ellers for symbolreferanser kan man velge blant stiplingstyper i en meny (gir linjetyper 1, 2, 4-9). Stipling spesifisert i symbolreferansen brukes alltid, dvs. definisjonen i laget overstyres.

<i>linecolor rgbFarge</i>	linjefarge (line color)
---------------------------	-------------------------

Angi linjefargen som en RGB-farge. Hver komponent (rød, grønn, blå) angis med et tall mellom 0 og 65535 (hex FFFF), Eksempler: `rgb(65535, 0, 0)` er helt rød, `rgb(65535/2, 65535/2, 65535/2)` er middels-grå.

Som regel bør utseende styres av laget symbolet er i. Setting av linjefarge i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet. Når symbolene tegnes med “fargekoding” blir overstyringer som denne ikke tatt hensyn til.

<i>linewidth mmTykk</i>	linjebredde (line width)
-------------------------	--------------------------

Angi linjetykkelse i millimeter på papiret. Dersom utskriften gjøres med en minimumstykkelser (Arkiv-> Utskriftssettinger) vil minimumstykkelser kunne overstyre den angitte tykkelsen.

I HouseModeller3D 1.2 blir linjetykkelsen tatt hensyn til kun ved utskrift, dvs. ikke på skjermen. (Dette pga problemer med håndtering av tynne linjer med zoom.)

Som regel bør utseende styres av laget symbolet er i. Setting av linjetykkelse i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet.

<i>filltype fyllType</i>	filltype (fill type)
--------------------------	----------------------

Filltype kan være: 0 for ikke-fyll, 1 for vanlig fyll, og 2 for fyll med fargeblanding. Fyll med fargeblanding gir et gjennomsiktig utseende da fargen blandes med fargen som ligger under på uttegningsstidspunktet. Fargeblanding fungerer greit på skjermen, men ikke på alle skrivere.

Som regel bør utseende styres av laget symbolet er i. Setting av filltype i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet.

<i>fillcolor rgbFarge</i>	fillfarge (fill color)
---------------------------	------------------------

Angi fillfargen som en RGB-farge. Hver komponent (rød, grønn, blå) angis med et tall mellom 0 og 65535 (hex FFFF), Eksempler: `rgb(65535, 0, 0)` er helt rød, `rgb(65535/2, 65535/2, 65535/2)` er middelsgrå.

Som regel bør utseende styres av laget symbolet er i. Setting av fillfarge i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet. Når symbolene tegnes med “fargekoding” blir overstyringer som denne ikke tatt hensyn til.

<i>fillalpha alfa</i>	fillalfa (fill alpha)
-----------------------	-----------------------

Angi fillalfa (gjennomsiktighet). Et tall mellom 0 og 1. 0 er helt gjennomsiktig, 1 er helt ugjennomsiktig.

Som regel bør utseende styres av laget symbolet er i. Setting av fillalfa i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet. Når symbolene tegnes med “fargekoding” blir overstyringer som denne ikke tatt hensyn til.

<code>< tekst</code>	venstrejustert tekst
<code>- tekst</code>	sentrert tekst
<code>> tekst</code>	høyrejustert tekst

Skriver ut tekst. Teksten kan være resultat av et uttrykk som beskrevet i uttrykksskrivet, bl.a. kan tekster settes sammen med “+”. Deler av en tekst kan skrives ut v.h.j.a. *substr*. Tegnene `<`, `-`, og `>` betyr venstre-, senter-, og høyrejustert.

Eksempler:

```
ma b / 2, -d / 2; -'S'          bokstav 'S' på skap, sentrert
ma vx1 - 0.032, 0.0080; -(pn + "+ ndrawing)  skala i tittelfelt, sentrert
<nout('3',dm1*1000) + ' x ' + nout('3', dm2 * 1000)  bjelke i tabell, venstre
```

Bruk av disse instruksjoner gir vertikal plassering ved tekstlinje av 1. linjen (slik det har vært før). Instruksjonene justerer hver linje av en multilinje tekst. Se `txlinejust` og `txboxjust` for flere måter å justere tekst.

`text` *tekst* tekst

Skriver ut tekst iht til gjeldende tekst-settninger. Brukes når man trenger mer funksjonalitet enn de enkle instruksjonene `<`, `-`, og `>` gir.

Det er vanlig å angi tekstlinjustering (`txlinejust`) og tekstboksjustering (`txboxjust`) før tekst instruksjonen gis. Man kan også spesifisere tekststørrelse, retning, og om tekstboksen skal fylles med egne instruksjoner.

`charup` *xopp, yopp* tekst opp-retning (character up vector)

Setter tekstretningen. Retningen spesifiseres som en vektor som angir "opp retningen". Lengden av vektoren har ingen betydning. Noen vanlige retninger:

<code>charup 0, 1</code>	mot høyre
<code>charup -1, 0</code>	oppover
<code>charup 0, -1</code>	mot venstre
<code>charup 1, 0</code>	nedover
<code>charup 1, 1</code>	45° oppover mot høyre

`txfont` *skrifttype* skrifttype (text font)

Setter skrifttypen. Eksempler:

```
txfont "Courier"
txfont "Palatino Italic"
txfont "Helvetica Bold"
```

For å finne et lovlig skrifttypenavn, kan man dobbeltklikke på en tekst på en husfil. Velg skrifttypen du vil bruke i det nye vinduet, og lukk vinduet. Les av navnet fra spesifikasjonen `Tekst skrifttype`.

Som regel bør skrifttypen styres av laget symbolet er i. Setting av skrifttype i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet.

`txsize` *h* skriftstørrelse (text size)

Setter skriftstørrelsen i punkter (1/72 tomme) på papiret, dvs. uavhengig av tegningskala. Bruk `txsize` eller `txwsize` som erstatning for den gamle `chsize`.

Som regel bør skriftstørrelsen styres av laget symbolet er i. Setting av skriftstørrelsen i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet.

`txwsize` *wh* skriftstørrelse i huskoordinater (text world size)

Setter skriftstørrelsen i meter i husets koordinater, dvs. teksten skaleres iht skala. Bruk `txsize` eller `txwsize` som erstatning for den gamle `chsize`.

Som regel bør skriftstørrelsen styres av laget symbolet er i. Setting av skriftstørrelsen i symboldefinisjonen overstyrer dette normalutseende og bør brukes med varsomhet.

`txlinejust` *linjust* tekstlinjer justering

Setter linjustering for hver linje av tekst innenfor tekstboksen. Verdiene kan være:

Verdi	Forklaring
0	venstrejustert
1	sentrert
2	høyrejustert

txboxjust *hjust,vjust,margs*

tekstboks plassering iht punkt

Angir plassering av tekstboksen i forhold til spesifikasjonspunktet.

Fra tekstens innhold konstrueres en tekstboks. Boksens bredde og høyde regnes ut automatisk fra innholdet, dvs. linjeding spesifiseres manuelt med returtegn i teksten.

Valgene for horisontal plassering er:

Verdi	Horisontal plassering
0	venstrejustert
1	sentrert
2	høyrejustert

Valgene for vertikal plassering er:

Verdi	Vertikal plassering
0	mot overkant
1	første linje sentrert
2	første linje underkant
4	teksten i første linje sentrert
5	første linje skriftlinje
6	sentrert
7	mot underkant

De vanligste vertikale plasseringer er mot overkant, sentrert, mot underkant, og første linje skriftlinje. De to andre plasseringer med første linje (nr. 1 og 2) finnes pga kompatibilitet med eldre versjoner.

Plasseringen *teksten i første linje sentert* tar hensyn til omsluttende rektangel av den aktuelle teksten. Dette brukes for å sentrere tekster som mangler underlengder (dvs bokstaver som går nedenfor skriftlinjen). Et eksempel er identifikasjonstekster for vegger på plantegninger.

Med *margs* spesifiseres om margene skal være innenfor tekstboksen eller utenfor. Hvis *margs* er null, blir margene inkludert i tekstboksen, ellers er de utenfor.

Både horisontale og vertikal plassering har de samme muligheter som for tekster i modellen.

txfill *nei-ja*

fyll bak tekst

Angir om tekstboksen skal fylles. 0 betyr at boksen ikke skal fylles, forskjellig fra null at det skal fylles.

På tegninger er det mye brukt å ha delvis gjennomskiktig hvit fyll.

Fyllstilen kan komme fra laget symbolet er plassert i, men foreløpig er det mer vanlig å spesifisere det direkte, f.eks.

```
push
txfill true
layerapp xxx % skriv lagnummeret som har det riktige fyllet
<"teksten her"
txfill false
pop
```

I framtiden kan det tenkes at lagene får en tekstfyllstil uavhengig av lagets fyllstil, slik at instruksjonen ovenfor blir unødvendige.

txmarg *t,r,b,l*

tekstmarger

Margene rundt tekster spesifiseres i piksler (1/72 tomme). De er i rekkefølgen: topp, høyre, bunn, venstre.

Margene brukes når tekstboksen regnes ut og er spesielt viktig når boksen fylles. Om margene legges til før eller etter boksen plasseres styres av *margs* i *txboxjust* instruksjonen.

image *imageNavn*

bilde (image)

Tegner et bilde fra en bildefil iht. gjeldende størrelse (*imagesize*) og opp-retning (*imageup*). Bildet tegnes v.h.j.a Quartz, noe som betyr at de fleste bildeformater kan brukes – f.eks. jpeg, tiff, png, og pdf.

HouseModeller3D leter etter filen først etter predefinerte bilder på leksikon (*Bilder med pikselgrafikk*), så i mappene Logoer og Bilder sammen med programmet.

Instruksjonen brukes nå mye mindre enn tidligere, fordi det er lettere å bruke egne bildeobjekter.

`imagesize bredde, høyde` bildestørrelse (image size)

Angir størrelsen av et bilde i papirkoordinater (ved referansemålestokken for grafikkvinduer). Dersom både bredden og høyden er null, blir bildet normalstørrelse, dvs. størrelsen definert i bildet. Dersom bare én er null, blir størrelsen satt slik at forholdet mellom bredde og høyde holdes uendret. F.eks. dersom bildet er normalt 3×4 cm og det settes `imagesize 0, 0.08` vil bildet tegnes 6×8 cm.

`imageinrect imageNavn, p0x, p0y, p1x, p1y, p2x, p2y, p3x, p3y` bilde (image) i rektangel

Tegner et bilde fra en bildefil iht. i den angitte rektangel. Rektangelet oppgis som x og y koordinater av de fire punktene, fortrinnsvis rundt mot sola. På denne måten kan bildet skaleres og roteres, og tegnes med punkter oppgitt i huskoordinater.

Bildet tegnes v.h.j.a. Quartz, noe som betyr at de fleste bildeformater kan brukes – f.eks. jpeg, tiff, png, og pdf.

HouseModeller3D leter etter filen først etter predefinerte bilder på leksikon (Bilder med pikselgrafikk, så i mappene Logoer og Bilder sammen med programmet).

Instruksjonen brukes nå mye mindre enn tidligere, fordi det er lettere å bruke egne bildeobjekter.

`imageup xopp, yopp` bilde oppretning (image up vector)

Setter bilderetningen, dvs. rotering av bildet. Retningen spesifiseres som en vektor som angir “opp retningen”. Lengden av vektoren har ingen betydning. Noen vanlige retninger:

<code>imageup 0, 1</code>	urotert
<code>imageup -1, 0</code>	rotert 90° mot venstre
<code>imageup 0, -1</code>	rotert 90° mot høyre
<code>imageup 1, 0</code>	opp ned
<code>imageup 1, 1</code>	rotert 45° mot høyre

`translate dx, dy [, dz]` translasjon

Translasjon; 2-dim eller 3-dim versjon. Ganger gjeldende transformasjonsmatrisen med en translasjonsmatrise. Det angitte punktet blir det nye nullpunktet. Pennen flyttes ikke.

`scale sx, sy [, sz]` skalering

Skalering; 2-dim eller 3-dim versjon. Ganger gjeldende transformasjonsmatrise med en skaleringsmatrise. NB! Dersom man ikke vil skalere i en bestemt retning, må man sette skaleringen til 1.0. Hvis skaleringen settes til 0.0, vil alle koordinatene i den gitte retningen bli 0.0 (ganget med 0.0).

Eksempler:

`scale 0.5, 1.0` skaler til halvstørrelse i X retning, behold skalering i Y, på plantegning
`scale -1.0, 1.0, 1.0` speilvending rundt Y-aksen, på 3D tegning

`rotate ax [, ay, az]` rotering

Rotasjon; 2-dim eller 3-dim versjon. Ganger gjeldende transformasjonsmatrisen med en rotasjonsmatrise.

`push` push transformasjon

Sparer transformasjonen på toppen av transformasjonslisten.

For å jobbe i et lokalt koordinatsystem gjøres vanligvis `push` etterfulgt av `translate`, `rotate`, og/eller `scale`. Da blir nye instruksjoner tolket i de nye lokale koordinater. Ved `pop` fåes tilbake det lokale koordinatsystemet før siste `push`.

Push og pop flytter ikke pennen.

Når et symbol tegnes er det en implisitt `push` og `pop` rundt alle instruksjoner, dvs. `push` og `pop` rundt hele innholdet er unødvendig. Dette gjelder også ved `call` instruksjonen.

I HouseModeller3D 1.2 blir også utseende, f.eks. linjetyper og farger, spart og satt tilbake med `push` og `pop`. Dette kan bli endret i framtiden.

`pop` pop transformasjon

Setter tilbake sist sparte transformasjon fra toppen av transformasjonslisten, og fjerner den fra transformasjonslisten.

`call [gruppeNavn _] symbolNavn [(argument, argument...)]` kall

Kaller et annet symbol (funksjon). Dersom *gruppeNavn* utelates antas det at symbolet er i samme gruppe som det kalledes symbolet.

I HouseModeller3D 1.2 er maksimalt antall argumenter 20. Argumentlista må starte med "(" og avsluttes med ")". Komma brukes mellom argumentene. Overflødige argumenter ignoreres. Utelatte argumenter får normalverdien som er definert i symbolet som kalles.

Translasjon til nåværende pennposisjon er implisitt i funksjonskall, slik at pennposisjonen før kallet tilsvarer (0, 0, 0) inne i symbolet. Det gjøres også en implisitt push og pop ved kallet.

En tekstvariabel kan inneholde symbolnavnet som skal kalles, det signaliseres med kolon foran *symbolNavn*. Eksempel som tegner et rektangel 2 × 1 meter:

```
set aname = g_rekt"
call :aname(2.0, 1.0)
```

For å få tegnet et vindu eller en dør kan det skrives (...) istf. den vanlige parameterlisten. Da leses parametrene fra vindus- eller dørdefinisjonen.

F.eks. brukes følgende kall for å tegne små vinduer og dører i tabeller:

```
call :wdSymbol(wdmat) (...);
```

wdmat er den spesielle materialkoden laget av HouseModeller3D (vinduer 1000000-1999999, dører 2000000-2999999). Først brukes `wdSymbol(wdmat)` til å finne symbolnavnet for vinduet eller døra, og så kalles symbolet med parametere fra definisjonen.

`return [uttrykk]` returner

Returnerer fra funksjonskall. Dersom et uttrykk er angitt returneres verdien til kallende symbol, dvs. `call` brukes som en funksjon. På denne måten kan `call` brukes videre i andre uttrykk.

`if betingelse` hvis – ellers
 instruksjon...
`[elseif betingelse`
 instruksjon...]
`[else`
 instruksjon...
`endif`

Første *betingelse* evalueres, dersom verdien er $\neq 0$, utføres instruksjonene opp til neste `elseif`, `else`, eller `endif`, deretter ignoreres instruksjonene opp til `endif`. Hvis verdien er 0, evalueres betingelsene i `elseif` instruksjonene inntil en *betingelse* med verdi $\neq 0$, `else` eller `endif` påtreffes. Deretter utføres instruksjoner på samme måte som ved `if`. Instruksjonene etter `else` utføres dersom ingen betingelser har verdi $\neq 0$.

`do` gjør - mens
 [instruksjon...]
`[while betingelse`
 instruksjon...]
`enddo`

Instruksjonene mellom `do` og `enddo` repeteres inntil `while`-betingelsen evaluerer til en verdi som er usann ($= 0$).

`mpoint x, y [, z]` målsettingspunkt

Målsettingspunkt for automatisk målsetting. Tas bare hensyn til dersom symbolet er i et lag som målsettes.

`dprint tekstuttrykk` skriv debug info
`dprintln tekstuttrykk` skriv debug info med linjeskift

Skriver ut den spesifiserte teksten i tekstvinduet. "\n" i teksten fører til ny linje. `dprintln` har en implisitt "\n" på slutten, dvs. avslutter alltid med ny linje. Brukes som hjelpemiddel under skriving av symboldefinisjoner. Ferdige symboldefinisjoner skal ikke inneholde `dprint` eller `dprintln` instruksjoner.

1.6 Instruksjoner som utgår

Symbolspråket har gjennomgått store endringer i sin levetid, bl.a. endringer ang. pennbegrepet, tekstutseende, faser i 3D tegning, og fyll av flater. Det finnes noen instruksjoner som er på vei ut og som ikke bør brukes i nye symboler, men som fremdeles tillates. Disse listes her.

<code>draw [bezier] [shield] [debug] koordinat...enddraw</code>	tegn kurve cubic splines
<code>pt x, y</code>	et punkt på kurven
<code>pt $x, y (dx, dy)$</code>	et punkt på kurven med spesifisert retning
<code>pt (x, y)</code>	skjult punkt (hjelper til å bestemme retningen)

`draw` tegner kurver, enten cubic splines eller (mer moderne) bézierkurver. Tegning av cubic splines vil bli tatt vekk i en senere versjon av HouseModeller3D. Nye og reviderte symboler bør bruke `bezier` istf. `draw` instruksjonen.

En Bézierkurve lages dersom `bezier` forekommer. Ellers lages en cubic spline (dvs. den opprinnelige virkemåten). En `shield`-polygon (skyggepolygon) lages dersom nøkkelordet `shield` forekommer – skal bare brukes i fase 1. Hvis `debug` forekommer blir punktene tegnet med kryss. Nøkkelordene kan komme i hvilken som helst rekkefølge.

Følgende forklaringen gjelder cubic splines (se `bezier` instruksjonen for en forklaring om bézierkurver). Kurven angis med en liste av koordinater (i form av `pt` instruksjoner) mellom `draw` og `enddraw`. Dersom retningen ikke er angitt vil den bli samme som en sirkel gjennom foregående, dette, og neste punkt ville ha hatt. Retningen kan angis på to måter: med usynlige punkter, eller med retningsvektor. Et punkt i parentes er et usynlig punkt og brukes bare for å bestemme retningen. En vektor i parentes etter et vanlig punkt angir hvilken retning kurven skal ha gjennom dette punktet. Kurven som tegnes mellom to punkter er en kubisk spline med følgende matematiske form (i det komplekse planet):

$$z(t) = z_1 + (3t^2 - 2t^3)(z_2 - z_1) + rt(1-t)^2\delta_1 - st^2(1-t)\delta_2$$

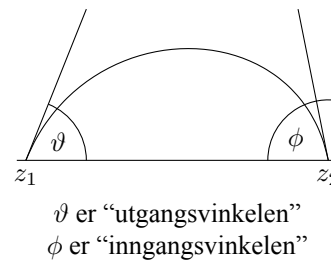
$$r = |2 \sin \vartheta / ((1 + |\cos \psi|) \sin \psi)|$$

$$s = |2 \sin \varphi / ((1 + |\cos \psi|) \sin \psi)|$$

$$\delta_1 = e^{i\vartheta}(z_2 - z_1)$$

$$\delta_2 = e^{-i\phi}(z_2 - z_1)$$

$$\psi = (\vartheta + \phi)/2$$



For mer info: Donald E. Knuth: *TEX and METAFONT, New Directions in Typesetting*. Tung Yun Mei: *LCCD, A Language for Chinese Character Design*, Software Practice & Experience, December 1981, Vol. 11, No. 12, p. 1278.

<code>epen W_x, W_y, ϑ</code>	elliptic pen
<code>epen</code>	elliptic pen off

Skal ikke brukes. Ble brukt til å spesifisere elliptisk penn for kubisk-spline kurver.

<code>bsa2 [tykk [toleranse]] koordinat...es</code>	begin-end shield absolute 2-dim
<code>bsr2 [tykk [toleranse]] koordinat...es</code>	begin-end shield relative 2-dim
<code>bsa3 [tykk [toleranse]] koordinat...es</code>	begin-end shield absolute 3-dim
<code>bsr3 [tykk [toleranse]] koordinat...es</code>	begin-end shield relative 3-dim
<code>co $x, y [z]$ koordinatpunkt</code>	

Lager en `shield`-polygon (en "skyggepolygon", dvs. en polygon som skjuler for andre objekter i tegningen). Polygonen spesifiseres ved å angi koordinatene av hjørnepunktene (i form av en liste av `co`-instruksjoner) mellom `bsxx` og `es` instruksjonene. Koordinatene angis som absolutte koordinater eller relativt til forrige punkt. I de 2-dimensjonale variantene utelates `z`. Det er en konvensjon at polygonene spesifiseres mot klokka, men begge veier fungerer.

For DrawType 1 og 7 blir det ikke lenger laget shields (pr HamboHus versjon 6.2). Shield instruksjoner ignoreres.

For DrawType 4, 5, og 10 kalles symbolet to ganger, en gang med `phase = 1`, og en gang med `phase = 2`. I fase 1 skal det lages shields, mens i fase 2 tegnes linjer. Når en `shield` instruksjon forekommer i fase 2 blir de omsluttende linjer tegnet.

Dersom et `shield` har tykkelse, blir polygonen ekstrudert i 3D med den angitte tykkelsen. I fase 2 blir omrisset av volumet tegnet, dvs. begge polygoner samt tykkelseslinjer ved punktene. Dersom en toleranse spesifiseres utelates linjene i tykkelsen dersom vinklene mellom linjene er slik at avvik til en rett linje er mindre enn toleransen i radianer. Dette brukes f.eks. på buede vinduer for å unngå linjer i tykkelse der buen deles opp.

For DrawType 4, 5, og 10 kan man teste på DrawType i symbolkoden, deretter teste på phase for å finne ut om man skal lage shields eller tegne linjer. For et symbol som skal tegnes ut på plantegning og oversiktstegning vil den typiske strukturen være slik:

```
DrawType 1,10
...parameterdefinisjoner...
if DrawType = 1
  ...kode for plantegning...
else
  if phase = 1
    ...lag shield-polygoner i 3-d...
  else
    ...tegn linjer 3-d linjer...
  endif
endif
```

Dersom shields lages i begge fasene, blir shield-polygonene laget i fase 1 mens omrissene tegnes i fase 2. Dette muliggjør følgende struktur når shield-polygonene skal både skygge og bli tegnet:

```
DrawType 1,10
...parameterdefinisjoner...
if DrawType = 1
  ...kode for plantegning...
else
  ...lag shield-polygoner i 3-d...
endif
```

Bruk av DrawTypes 4, 5, og 10 signaliserer til HouseModeller3D programmet at symbolet er skrevet på den gamle måten med shields og to faser. Symbolene bør etter hvert moderniseres til å bruke polygon og bezier instruksjoner, uten shields. At symbolet er skrevet om (og skal bare kalles én gang) signaliseres ved endring av DrawType til 15, 24, og/eller 25.

pen pennNummer	penn
----------------	------

Gjør ingenting. Ble tidligere brukt for å spesifisere penn nummer for plotter.

chsize bredde, høyde	tegnstørrelse
----------------------	---------------

Setter bokstavstørrelsen, men bare høyden tas hensyn til. Bredden ignoreres. Størrelsen spesifiseres i meter, f.eks. "chsize 0.002,0.003" for 3mm høye bokstaver. chsize setter også tekstretningen til høyre (0,1). Dersom en annen tekstretning enn høyre ønskes og må charup gjøres etter chsize.

Når symboldefinisjoner moderniseres bør chsize endres til txsize eller txwsize.

1.7 Uttrykk i symboler

Uttrykk som beskrevet i skrevet HouseModeller3D 1.2 Uttrykk kan brukes i symboler.

De vanligste uttrykk bruker variabler av type heltall, reelle tall, eller tekst. Alle tall må skrives med desimalpunktum (ikke komma) i symboldefinisjoner. Argumentene for sin, cos, tan og resultatene fra asin, acos, atan, atan2 er i radianer når disse kalles fra symboler. Resultatene fra de inverse trigonometriske funksjonene er normalisert slik:

$$-\pi/2 < \operatorname{atan} \beta < \pi/2 \quad -\pi/2 \leq \operatorname{asin} \beta \leq \pi/2 \quad 0 \leq \operatorname{acos} \beta \leq \pi$$

I forhold til tidligere programvareversjoner er tekst-typen betydelig utvidet og tatt inn i de vanlige uttrykkene. Tekstkonstanter skrives som teksten omsluttet av apostrofer (') eller anførselstegn ("). Dersom en apostrof (eller anførselstegn) skal være en del av teksten, må den skrives dobbelt eller på motsatt måte. Tekster som skrives etter hverandre med "+" i mellom settes sammen til en tekst. (I motsetning til tidligere må "+" tegnet alltid brukes.)

Funksjonen length gir lengden av teksten, trim brukes til å ta bort mellomrom på begynnelsen og slutten, og substr brukes til å plukke ut en del av teksten. Funksjonen nout brukes til å formatere et tall som tekst.

```
set n = tekst.length
set s = substr(tekst, indeks, lengde) % (som i språket c++)
set s = nout('3', dm1 * 1000) + ' x ' + nout('3', dm2 * 1000) + ' i '
```

I tillegg til de basis innebygde datatypene har symbolspråket to ekstra datatyper: array og objekt.

Arraykonstanter skrives slik: [*element* [, *element*]...]. Arrayer kan nestes. For å bruke et element i en array skrives: [*indeks*] der indeksering starter fra null. Man kan sette verdier til enkelt arrayelementer. Hvis man setter en verdi i en indeks forbi slutten, blir arrayen forstørret til minst indeksen. Antall elementer i en array gis av *array.length*.

```
set n = array.length
```

Objektkonstanter skrives slik: { *nøkkel* : *verdi* [, *nøkkel* : *verdi*]...}. Objekter kan nestes. Nøkler kan være navn regnet ut med uttrykk. For å bruke et element i en objekt skrives *objekt.nøkkel* eller *objekt[nøkkel]*. Sistnevnte formen må brukes for nøkler som er uttrykk.

Det finnes også funksjoner for å dele opp en tekst til en array, eller for å sette sammen en array til en tekst.

```
set n = splitString(tekst, separator, array)
set s = joinArray(array, separator)
```

Eksempler:

```
set s=[1,[7,8],{x:10,y:11}]
set s[2].x=s[1][0]

set alder={"ole":22,"per":19}
set alder["ole"]=23

set lines=["xx\tyy\tzz","aa\tbb\tcc","dd\tee\tff"];
set texts=[]
set i=0
do while i<lines.length
  set nfields=splitString(lines[i],fields,"\t")
  set texts[i]=fields % Extend array texts with each iteration
  set i=i+1
enddo
```

Vindu- og dørsymboler setter opp polygoner som arrayer av objekter slik (må skrives på en linje):

```
cset WDKarm = [{x: rapnx, y: rapny},
               {x: rapnx+rapnbr, y: rapny},
               {x: rapnx, y: rapny + lh}]
```

Arrayer, objekter, og resultater av *call* instruksjoner kan brukes i uttrykk for *set*, *cset*, *return*, eller som argumenter i *call* instruksjoner.

I tillegg til generelle funksjoner og tekstfunksjoner defineres en del variabler, avhengig av hvor i HouseModeller3D symbolet kalles fra. Noen symboler (som f.eks. tittelfelt, vindu-/dør) kan bare bli brukt i visse situasjoner, mens andre kan bli kalt fra alle typer tegninger. Variabelen *DrawType* angir hva slags tegning symbolet er kalt for.

Variabel/Funksjon	Forklaring
<i>DrawType</i>	Angir hvor symbolet er kalt fra.
p1 p2 p3	Tekstene under Prosjektidentifikasjon på husfilen - Tiltakshaver, Prosjekt, og Prosjektadresse. (For Blink er de samme feltene kustomisert til henholdsvis Byggherre, Byggadresse og Tegningsnummer.) Hvert felt kan bestå av flere linjer, men utforming av tittelfeltsymbolene setter ofte praktiske begrensninger til både lengde og antall linjer.
pn	Teksten i Prosjektkode under Prosjektidentifikasjon på husfilen.
phlogo	Filnavn for tiltakshaver logo fra Prosjektidentifikasjon på husfilen. Kan brukes som bildenavn (image) i image-instruksjoner.
pplogo	Filnavn for prosjekt logo fra Prosjektidentifikasjon på husfilen. Kan brukes som bildenavn (image) i image-instruksjoner.

Variabel/Funksjon	Forklaring
maal	En tekst som angir målestokk i format 1:n. Hvis flere målestokker forekommer på samme tegningsark, blir det laget en liste av de som forekommer. F.eks. 1:20, 1:50, 1:100.
date	Dato i formatet: dag.måned.år, f.eks. 23.9.07.
etasjeNavn (<i>etasjeNummer</i>)	Slår opp etasjenavnet på husfilen for den spesifiserte <i>etasjeNummer</i> . Hvis Etasjenavn ikke er tomt, returneres dette. Ellers returneres teksten "N. etasje", hvor N er <i>etasjeNummer</i> .
wdSymbol (<i>kode</i>)	Returnerer en tekst som er navnet av symbolet for det angitte vindu eller dør. (<i>Kode</i> er den spesielle koden laget av HouseModeller3D. Vinduer er 1000000-1999999, dører 2000000-2999999. Brukes til å tegne vindu eller dør ved å kalle vindu-/dørsymbolet fra tabellsymboler.
report (<i>nummer</i>)	Eksperimentell. Utfører rapport med det angitte nummer: Resultatet er hele rapporten som én tekst.
splitString (<i>tekst, tegn, array</i>)	Deler en tekst ved det angitte tegnet. Resultatet er en array av de enkelte delene.
joinArray (<i>array, tegn</i>)	Setter sammen delene av en array til én tekst. Mellom delene settes det angitte tegnet.
ndrawing	Tegningsnummeret, dvs. serienummer i listen av tegningsark.
zfloor	Z-koordinat for overkant gulv på pennens X-Y. Etasjen er den symbolet er i. Gir fornuftige resultater når symbolet er plassert i en etasje (ikke når symbolet er plassert på husnivå).
zceiling	Z-koordinat for underkant himling på pennens X-Y. Etasjen er den symbolet er i. Gir fornuftige resultater når symbolet er plassert i en etasje (ikke på husnivå).
zroof (<i>høyeste, overkant</i>)	Z-koordinat for tak ved pennens X-Y. Første argument er 0 for laveste tak, 1 for høyeste. Andre argumentet er 0 for underkant tak, 1 for overkant tak, og 2 for overkant tekking. Funksjonen tar ikke hensyn til etasjene, finner bare laveste eller høyeste tak av alle tak-etasjeskillere.
scale	Målestokken for den tegningsdelen symbolet forekommer på. F.eks. 1:50 gir <i>scale</i> = 50.
matdim1 (<i>materialkode</i>)	Dimensjon 1 for et material. Resultatet er vanligvis tykkelsen, men det blir lengden hvis tykkelsen er null.
matdim2 (<i>materialkode</i>)	Dimensjon 2 for et material. Resultatet er bredden.
okcode (<i>materialkode</i>)	1 hvis <i>materialkode</i> er en lovlig materialkode, 0 ellers. Tester ikke om materialet finnes på databasen, bare om koden er lovlig. Vanlige materialkoder er i området 1-999899 og 4000000-2000000000. I tillegg kommer automatisk generert vinduskoder 1000001-1999999, dørkoder 2000001-2999999, og reserverte koder 3000001-3999999. (Koder som ikke er ok er 0 og 999900-999999.)
defined (<i>variabel</i>)	1 hvis <i>variabel</i> er definert i symboltabellene, 0 ellers.
node (<i>node, nodeUttrykk</i>)	Brukes til å signalisere at et nodeuttrykk skal evalueres (se uttrykksskrivet). <i>node</i> er vanligvis 0 eller . (begge gir noden som er satt opp av HouseModeller3D). Eksempel: <code>set husz = zTerrengNaa(node(., son233), node(., son234))</code>

Variabel/Funksjon	Forklaring																
zTerrengNaa [(x, y)]	Beregner z for eksisterende terreng på et X-Y punkt. Dersom intet punkt oppgis beregnes terrengets Z på pennens posisjon. Ellers oppgis X og Y i globale huskoordinater. Dersom det ikke finnes terreng på det angitte X-Y blir resultatet en NaN som kan sjekkes for med numberOK.																
zTerrengFram [(x, y)]	Samme som zTerrengNaa, men for framtidig terreng.																
zKoteNaa [(z)]	Beregner eksisterende terreng z for koteangivelse fra en z-verdi i husets koordinater, dvs. z blir justert iht. spesifikasjoner i terrenget og for den aktuelle tegningen (fasade, snitt). Dersom det ikke er noe justering eller tegningstypen er noe annet, blir resultatet husets z koordinat.																
zKoteFram (z)	Samme som zKoteNaa, men for framtidig terreng.																
numberOK (uttrykk)	1 hvis uttrykk gir et gyldig nummer, 0 hvis det er NaN eller uendelig (f.eks. etter divisjon med null)																
layerVisible (lagnummer)	1 hvis laget er synlig, 0 ellers.																
aspectRatio (bildenavn)	Reelt tall som angir forholdet bredde / høyde for et bilde. Bildet må finnes på leksikon eller i mappene Logoer og Bilder sammen med programmet.																
evalstr (tekst)	Regner ut text uttrykk som er i teksten. Brukes til å regne ut uttrykk i tekst parametre og variabler. Et tekstuttrykk har formen $\#{uttrykk}$.																
fillColor	fyllfargen som er satt opp nå.																
lineColor	linjefargen som er satt opp nå.																
textheight (tekst)	Regner ut høyden av teksten i piksler. Brukes f.eks. til å formatere tabeller.																
textwidth (tekst)	Regner ut bredden av teksten i piksler. Brukes f.eks. til å formatere tabeller.																
onscreen	For symboler og vinduer/dører tegnet fra grafikken, er variabelen onscreen definert og satt til 1. Ellers kan variabelen være definert og satt til 0, eller være udefinert. Eneste bruk i øyeblikket er for å tegne målpunkter på grafikken, men ikke på tegninger.																
forcePolygonFill	Et symbol kan sette denne variabelen for å tvinge fram en tilnærming av gamle "shields" på plantegning. Når variabelen er satt og ikke 0, blir alle polygonene fylt med hvit (fargen fra laget brukes ikke). Variabelen brukes i følgende symboler: Ventil i vegg, Miniavløp, Sikringsskap, Ettløps Leca, og Dobbeltløpet Leca. Variabelen bør ikke brukes i nye symboler.																
modelExport	Definert og satt til 1 når symboler eksporteres til DXF, 0 ellers. Brukes bl.a. for å unngå eksport av en del pyntesymboler, f.eks. takstein ved raft.																
nout (format, verdi)	Konverterer tallverdi til tekst. <i>Format</i> kan inneholde følgende formatspesifikasjoner: <table border="1" data-bbox="694 1742 1284 2011"> <tbody> <tr> <td>n</td> <td>heltallformat med feltlengde n</td> </tr> <tr> <td>en</td> <td>e-format med n siffer etter komma</td> </tr> <tr> <td>fn</td> <td>f-format med n siffer etter komma</td> </tr> <tr> <td>xn</td> <td>kortest mulig format med n siffer etter komma</td> </tr> <tr> <td>l</td> <td>venstrejustert</td> </tr> <tr> <td>z</td> <td>yll med nuller (bare for høyrejustert)</td> </tr> <tr> <td>u</td> <td>uten fortegn</td> </tr> <tr> <td>rn</td> <td>grunntallspesifikasjon ($2 \leq n \leq 36$)</td> </tr> </tbody> </table>	n	heltallformat med feltlengde n	en	e-format med n siffer etter komma	fn	f-format med n siffer etter komma	xn	kortest mulig format med n siffer etter komma	l	venstrejustert	z	yll med nuller (bare for høyrejustert)	u	uten fortegn	rn	grunntallspesifikasjon ($2 \leq n \leq 36$)
n	heltallformat med feltlengde n																
en	e-format med n siffer etter komma																
fn	f-format med n siffer etter komma																
xn	kortest mulig format med n siffer etter komma																
l	venstrejustert																
z	yll med nuller (bare for høyrejustert)																
u	uten fortegn																
rn	grunntallspesifikasjon ($2 \leq n \leq 36$)																

Eksempler på *nout*:

<code>nout(' ', n)</code>	heltall, kortest mulig (feltlengde avhengig av tallverdien)
<code>nout('5', n)</code>	heltall, høyrejustert i 5 karakters felt
<code>nout('uz6', n)</code>	materialkode (feltlengde 6, ikke fortegn, fyll med nuller)

1.8 Vinduer og Dører

Vindu- og dørsymboler kalles for å tegne vinduer og dører på plan, fasader, og oversiktstegninger. Symbolene brukes i mange andre sammenhenger, bl.a. under eksport.

Symbolene kalles på fasader og oversikter for å tegne bånd rundt åpninger. I tidligere HouseModeller3D versjoner ble vindu/dør tegnet samtidig med båndene, nå tegnes de separat. Variabelen `attachedParts` er definert og satt til 1 når bånd skal tegnes. `attachedParts` brukes bare i de tilfellene der vinduet/døra har overtatt ansvaret for tegning av et eller flere bånd. (Ellers tegnes båndene basert på kledningsspesifikasjonene, en metode som foretrekkes. Metoden kan brukes med de aller fleste vinduer og dører.)

Setting av parametere fra symbolet (DrawType 10000)

Vindus- og dørsymboler kan sette opp en del parametere til HouseModeller3D. Dette betyr at mye av ansvaret for utseende er flyttet ut til symbolene. Bl.a. kan de oppgi polygoner for hull i bindingsverket og i kledningen, noe som brukes for ikke-rektangulære vinduer.

Følgende variabler som brukes når vinduer og dører skal settes opp initialt (DrawType 10000).

Variabel	Forklaring (DrawType = 10000)
<code>slag</code>	Slagretning som definert i åpningen (0 = inn, 1 = ut)
<code>apnx</code>	Start x for åpning i lokale veggkoordinater, som spesifisert i åpning
<code>apny</code>	Start y (underkant) for åpning i lokale veggkoordinater
<code>apnbr</code>	Bredde av åpning – brutto bredde, som spesifisert i åpning
<code>apnhd</code>	Høyde av åpning – brutto høyde, som spesifisert i åpning
<code>rapnx</code>	Start x for ramme i lokale veggkoordinater, initialt <code>apnx + adyttLeft</code>
<code>rapny</code>	Start y (underkant) for ramme i lokale veggkoordinatdytningsmonner, initialt <code>apny + adyttBottom</code>
<code>rapnbr</code>	Bredde av rammen (initialt åpningsbredden minus dytningsmonn), initialt <code>apnbr - adyttLeft - adyttRight</code>
<code>rapnhd</code>	Høyde av rammen (initialt åpningshøyden minus dytningsmonn), initialt <code>apnhd - adyttTop - adyttBottom</code>
<code>hasAttachedParts</code>	1 hvis vindu/dør overtar ansvar for minst ett bånd (dvs. skal kalles for tegning av bånd rundt åpninger)
<code>hasOverBand</code> <code>hasUnderBand</code> <code>hasLeftBand</code> <code>hasRightBand</code>	Settes for de aktuelle bånd symbolet ønsker å overta ansvaret for. Fører til at kledningsspesifikasjoner for de aktuelle båndene ikke tas hensyn til.
<code>adyttTop</code>	Dytningsmonn på toppen.
<code>adyttBottom</code>	Dytningsmonn på bunnen.
<code>adyttLeft</code>	Dytningsmonn på venstre siden.
<code>adyttRight</code>	Dytningsmonn på høyre siden.
<code>akarmdekkTop</code>	Kledning dekker over karm på toppen.
<code>akarmdekkBottom</code>	Kledning dekker over karm på bunnen.
<code>akarmdekkLeft</code>	Kledning dekker over karm på venstre siden.

Variabel	Forklaring (DrawType = 10000)
akarmdekkRight	Kledning dekker over karm på høyre siden.
adytt	UTGÅTT pr 6.0.3. Dyttingsmonn på venstre side.
akarmdekk	UTGÅTT pr 6.0.3. Kledning dekker over karm på venstre side.
kleDekk	Sant hvis det ikke skal lages hull verken i veggen eller huset. Ble brukt før på noen spesielle ikke-rektangulære vinduer.
drawAsHole	Sant hvis det skal tegnes som et "hull" uten markering. Spesielt nyttig for avslutninger i betongvegger.
bindPoly	Bindingsverkpolygon, settes av symbolet hvis vindu/dør ønsker noe annet enn normalen.
kledPoly	Kledningspolygon, settes av symbolet hvis vindu/dør ønsker noe annet enn normalen.
FSPoly	Fasadepolygon, settes av symbolet hvis vindu/dør ønsker noe annet enn normalen.
WDKarm	Karmpolygon, settes av symbolet hvis vindu/dør ønsker noe annet enn normalen.
WDRamme	Utgått, samme som WDKarm.

Alle variablene unntatt polygonene settes opp med "normalverdier" før kall til symbolet. Variablene `slag`, `apnx`, `apny`, `apnbr`, `apnhd` og rammestørrelsen kan brukes av symbolet til beregninger, men de leses ikke tilbake av HouseModeller3D (dvs. symbolet kan ikke overstyre verdiene). Verdiene kommer fra spesifikasjonene i åpningen. Høyde og bredde er brutto høyde og bredde som angitt i åpningen, mens rammeplassing og størrelse regnes ut ved å trekke fra dytningsmonn på hver side.

Vinduet eller døra kan overstyre både dytningsmonn og kledning-over-karm verdiene. De overstyrte verdiene brukes under automatisk utregning av polygonene. Overstyring brukes for ikke-rektangulære vinduer, og kan av og til virke merkelig f.eks. for trekantvinduet setter `adyttTop` = `adyttLeft` for polygon utregninger. Merkverdighetene skyldes måten bruttoåpning beregnes fra rammestørrelsen. Se brukermanualen for mer om dette.

Dersom en polygon ikke settes fra symbolet, brukes et rektangel. Dette er det vanlige tilfellet. Ikke-rektangulære vinduer og dører overstyrer polygonene ved å sette dem direkte. Det er mulig å sette bindingsverkpolygonen (dvs. "hullet" i bindingsverk) til en ikke-rektangulærpolygon, men det bør ikke gjøres da konstruksjonsresultatet kan bli merkelig. Andre polygoner kan settes til ønsket fasong.

Fasadepolygon er åpningen i veggtykkelsen utenom utvendig kledning. Forskjellen mellom fasadepolygonen og kledningspolygon fylles med egne flater i åpningene og på innsiden.

Dersom et vindu eller dør ønsker å overta ansvaret for bånd rundt åpning skal variabelen `hasAttachedParts` settes til true eller 1. Dette gjør at vindus- eller dørsymbolet vil bli kalt for tegning av båndene (med variabelen `attachedParts` definert). Variabelen for hvert bånd symbolet ønsker å overta skal også settes, f.eks. `hasOverBand`. Setting av disse fører til at HouseModeller3D ikke tar hensyn til andre kledningsspesifikasjoner for det aktuelle båndet. Eksempel:

```
if tykk > 0.0; cset hasAttachedParts = true; cset hasOverBand = true; endif
```

Her overtar trekantvindussymbolet ansvar for bånd over åpningen når parameteren "Bord over dimensjon" er større enn null. Bånd spesifisert i kledningsspesifikasjoner på sidene og under åpningen vil bli brukt, men bånd over åpningen tas ikke hensyn til (bånd over åpning kommer alltid fra vindussymbolet).

Vinduet eller døra kan overstyre de spesielle tegningsmåter, `kleDekk` og `drawAsHole`. Den første tegner vinduet eller døra utenpå kledningen, uten å lage hull, mens den andre tegner bare et hull uten merking.

Variabler tilgjengelig under uttegning

Følgende variabler er tilgjengelig når et vindu eller dørsymbol kalles for uttegning.

Variabel	Forklaring
phase	Er 1 eller 2 når man tegner med gamle symboler. Se forklaring om phase i DrawType avsnittet. Bruk av phase er på vei ut.
modelExport	1 når symbolet kalles under eksport til DXF, 0 ellers

Variabel	Forklaring
makeGlass	1 når symbolet kalles under eksport til DXF og polygoner skal lages for vindusglass, 0 ellers. (makeGlass sammen med modelExport er nå også brukt under vanlig 3D tegning.)
vindu	0 = dør, 1 = vindu
attachedParts	definert og satt til 1 dersom bånd rundt åpningen skal tegnes, ellers udefinert
slag	slagretning som definert i åpningen (0 = inn, 1 = ut)
apnbr	Bredde av åpning – bruttobredde
apnhd	Høyde av åpning – bruttohøyde
rapnbr	Bredde av rammen
rapnhd	Høyde av rammen
kapnbr	Bredde av kledningshull
kapnhd	Høyde av kledningshull
txl	På plan, bredden av åpningen. Ellers nettobredden av åpningen langs bunnen.
tyl	På plan, tykkelsen av åpningen (veggen). Ellers nettohøyde av åpningen langs venstre kant.
adyttTop	Dyttingsmonn på toppen.
adyttBottom	Dyttingsmonn på bunnen.
adyttLeft	Dyttingsmonn på venstre siden.
adyttRight	Dyttingsmonn på høyre siden.
akarmdekkTop	Kledning dekker over karm på toppen.
akarmdekkBottom	Kledning dekker over karm på bunnen.
akarmdekkLeft	Kledning dekker over karm på venstre siden.
akarmdekkRight	Kledning dekker over karm på høyre siden.
dorVink	Vinkelen for tegning av dører, fra Dør vinkel på plan i Tegningsparametre.
hengPlan	0 eller 1, fra Vis vindushengsling på plan i Tegningsparametre.
apnPlanTynn	0 eller 1, fra Åpninger tegnes tynne på plan i Tegningsparametre.
apnPlanYK	Ytterkant åpning på plan (tar hensyn til om åpninger skal tegnes tynne).
apnPlanIK	Innerkant åpning på plan (tar hensyn til om åpninger skal tegnes tynne).
bandTykk	Tykkelse av åpningsbånd, alltid 0,02 i HouseModeller3D.
dxl	UTGÅTT pr 5.1.2. Alltid 0.
dxr	UTGÅTT pr 5.1.2. Bredden av vindu/dør.
adytt	UTGÅTT pr 6.0.3. Dyttingsmonn på venstre side.
akarmdekk	UTGÅTT pr 6.0.3. Kledning dekker over karm på venstre side.

1.9 Tilleggssymboler rundt åpninger

Tilleggssymboler i gruppen “Symboler rundt åpninger” kan legges til i “Avvik i utvendig kledning” for en åpning og i “Løsning rundt åpning” i avvikende utvendig kledning for en vegg, og i “Løsning rundt åpning” i kledningsnormalverdier for etasjer og for huset. Tilleggssymbolene tegnes på fasader og oversiktstegninger.

Når tilleggssymbolene tegnes, er alle variablene for vinder og dører satt opp som beskrevet ovenfor. Følgende tilleggssymboler er også definert – disse gjør det mulig for tilleggssymbolet å plassere seg utenfor andre bånd som allerede finnes. Koordinatsystemet er lokale åpningskoordinater.

Variabel	Forklaring
DrawType	Alltid 4 for tilleggsymboler
phase	Er 1 eller 2 når man tegner med gamle symboler. Se forklaring om phase i DrawType avsnittet. Bruk av phase er på vei ut.)
modelExport	1 når symbolet kalles under eksport til DXF, 0 ellers
harBåndOver	1 når et bånd over finnes, 0 ellers
x1BåndOver xrBåndOver ybBåndOver ytBåndOver	Koordinater av båndets omfang (x på venstre side, x på høyre side, y på bunnen, y på toppen)
harBåndUnder	1 hvis et bånd under finnes, 0 ellers
x1BåndUnder xrBåndUnder ybBåndUnder ytBåndUnder	Koordinater av båndets omfang (x på venstre side, x på høyre side, y på bunnen, y på toppen)
harBåndVenstre	1 hvis et bånd på venstre side finnes, 0 ellers
x1BåndVenstre xrBåndVenstre ybBåndVenstre ytBåndVenstre	Koordinater av båndets omfang (x på venstre side, x på høyre side, y på bunnen, y på toppen)
harBåndHøyre	1 hvis et bånd på høyre side finnes, 0 ellers
x1BåndHøyre xrBåndHøyre ybBåndHøyre ytBåndHøyre	Koordinater av båndets omfang (x på venstre side, x på høyre side, y på bunnen, y på toppen)
bandTykk	Bånd tykkelse, alltid lik 0.02 i HouseModeller3D 1.2

1.10 Bånd v.hj.a. symboler

I HouseModeller3D 1.2 finnes det flere band som tegnes v.hj.a. egne symboler i gruppen "Kledningsbånd". Disse inkluderer "Vertikalt bord med trekant nederst", "Trekant med vannbrett" og "Plateskjøt".

Sortimentet kan utvides ved å lage et nytt symbol i gruppen "Kledningsbånd". I tillegg utvides menyen der man skal referere til det. Det finnes 4 menyer: 28 (horisontale bånd), 31 (Løsning rundt åpninger - sidene), 29 (Løsning over/under åpning), og 41 (Vertikale bånd).

Menytall under 100 gir innebygd oppførsel og skal ikke brukes. Tall 100 og over gir symbolsk uttegning, navnet i menyen brukes som navnet på symbolet. (Tallene 100 og over behøver ikke være samstemt i de forskjellige menyer, men det er sikkert lurt å gjøre det likevel. Tallene over 100 kan endres uten at det får konsekvenser, siden det er navnet som brukes.) Symboler med nummer under 1000 tegnes "opp-på", mens symboler over er innfelt (horisontalt).

Båndet blir klippet til veggen den er definert i, og dessverre finnes det kjente problemer med klipping for fasonger som ikke er rektangulære.

Koordinatsystemet er transformert slik at nullpunktet er nederste venstre hjørne av båndet.

Følgende variabler er tilgjengelige.

Variabel	Forklaring
drawType	Alltid 4
phase	Er 1 eller 2 når man tegner med gamle symboler. Se forklaring om phase i DrawType avsnittet. Bruk av phase er på vei ut.)

Variabel	Forklaring
modelExport	1 når symbolet kalles under eksport til DXF, 0 ellers
vindu	0 = dør, 1 = vindu
bandb	Båndbredde
bandh	Båndhøyde
bandTykk	Båndtykkelse, alltid lik 0.02 i HouseModeller3D 1.2

1.11 Oppriss åpningsmarkering

På oppriss finnes det to spesielle symboler, `g_Åpningsbredde` på oppriss og `g_Åpningshøyde` på oppriss, som brukes til å tegne åpningsbredde og åpningshøyde. Når symbolene tegnes er koordinatsystemet transformert til åpningsens nederste venstre hjørne.

Dersom en annen måte å vise åpningsbredde og -høyde ønskes, kan symboldefinisjonene overstyres på husfilen på vanlig måte.

Følgende variabler er definert:

Variabel	Forklaring
apnbr	bredden av åpningen.
apnhd	høyden av åpningen.

1.12 Takkantsymboler

Symboler i gruppen “Takkant symboler” tegnes ifm en takkant. Gruppen inkluderer mønepanner, mønekors, og takkstein ved raft.

Koordinatsystemet er globale huskoordinater og symbolet må selv transformere til lokale koordinater dersom det trengs. Følgende instruksjoner kan brukes til en slik transformasjon:

```

set dx = p2x - p1x; set dy = p2y - p1y; set dz = p2z - p1z
set l = sqrt(dx * dx + dy * dy + dz * dz)
set xx = dx / l; set xy = dy / l; set xz = dz / l
set yz = normx * xy - normy * xx
push
% Transformer: x langs linjen mellom de to punktene, z langs normal vektoren
% y er på planet, positiv for takflaten (n cross x)
if tekkykk > 0.001; set tt = tekkykk / normz; else; set tt = 0.0; endif
translate p1x, p1y, p1z + tt; % null langs raft
set ls = sqrt(xx * xx + xy * xy)
rotate atan2(yz, normz) * 180 / pi, atan2(-xz, ls) * 180 / pi, atan2(xy, xx) * 180 / pi
ma 0, 0, 0

```

Følgende variabler er definert:

Variabel	Forklaring
drawType	4 eller 24 for fasade og oversikt, 5 eller 25 for snitt
phase	Er 1 eller 2 når man tegner med gamle symboler. Se forklaring om phase i DrawType avsnittet. Bruk av phase er på vei ut.)
modelExport	1 hvis symbolet kalles under eksport til DXF, 0 ellers

Variabel	Forklaring
p1x p1y p1z	kantens første punkt
p2x p2y p2z	kantens andre punkt
normx normy normz	takflatens normalvektor
tykk	takflatens tykkelse
tekktykk	taktekkingens tykkelse
angp1	takpolygonens vinkel ved første punkt, i radianer (settes til $\pi/2$ dersom 0 eller 180°)
angp2	takpolygonens vinkel ved andre punkt, i radianer (settes til $\pi/2$ dersom 0 eller 180°)

Takkanter kan være definert til å bruke enkle tegnemuligheter v.h.j.a. spesifikasjoner i takkant-definisjonen: Vindski på tegning, Forkantbord tegnes, og Forkantbord dimensjon. Da blir vindskier og forkantbord tegnet automatisk v.h.j.a. de to symbolene AutoVindski og AutoForkantbord.

De automatisk symboler kaller videre til de vanlige takkantsymboler. Man kan endre oppførselen til den automatiske uttegningen ved å endre innhold i AutoVindski og AutoForkantbord.

Når de automatiske symbolene tegnes, blir verdiene av spesifikasjonene også tilgjengelige i variabler:

Variabel	Forklaring
drawType	24 for fasade og oversikt, 25 for snitt
vind	hvor mange vindskier: -1 (ingen), 1 (enkel), 2 (dobbel), 3 (trippel)
kapp	endekapping av vindski: 0 (rettkapp), 1 (skrå), 2 (vertikal)
vinddim	vindski dimensjon
vannb	vannbord tegnes: 0 (nei), 1 (ja)
vanndim	vannbord dimensjon
forkb	forkantbord: -1 (medtas ikke), 1 (vertikal)
forkdim	forkantbord dimensjon

1.13 Terrengsymboler på fasade

I symbolgruppen Terreng på fasade defineres symboler som skal tegnes ut for koter og for endemerking av linjene. Koordinatsystemet er transformert slik at origo er terrengpunktet som skal tegnes.

Nye symboler definert i gruppen Terreng på fasade blir umiddelbart tilgjengelig for bruk på fasadetegninger.

Variabel	Forklaring
koteHusz	kotens z koordinat i huskoordinater
kote	kotens z koordinat i justerte koordinater (juster iht. spesifikasjoner i terrenget)