

Yak

COLLABORATORS

	<i>TITLE :</i> Yak		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 27, 2024	

REVISION HISTORY

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

Contents

1	Yak	1
1.1	Yak v1.60 Dokumentation	1
1.2	Copyright	1
1.3	Eksisterende brugere: OBS!	2
1.4	Introduktion	2
1.5	Vittigheder	3
1.6	Begrænsninger	4
1.7	Start af Yak	4
1.8	Ændring af Yak's opsætning	5
1.9	Muligheder	6
1.10	Aktivering under mus	6
1.11	Flyt vindue frem	6
1.12	RMB-aktivering	6
1.13	MMB-aktivering	7
1.14	Tasteaktivering	7
1.15	Klik vindue frem	7
1.16	Klik vindue tilbage	7
1.17	Bladre skærme	8
1.18	Skærmaktivering	8
1.19	'*' i mønster	8
1.20	Ingen drev-klik	8
1.21	Inkluderingsmønstre	9
1.22	Aktivering af skærme	9
1.23	Klikskærme	9
1.24	Standardtitel på en skærm	9
1.25	Flytte vinduer frem	9
1.26	Klikke vinduer frem	10
1.27	Diverse opsætninger	10
1.28	Klikvolumer	10
1.29	Aktiveringsforsinkelse	10

1.30	MMB Shift	10
1.31	Sort kant	10
1.32	Blanking vindue	11
1.33	Museblankingsmetode	11
1.34	Blank mus ved tastetryk	11
1.35	Ventetider	11
1.36	Skærmbanking	11
1.37	Musebanking	11
1.38	Definering af Yak Genveje	12
1.39	Beskrivelse af genveje	12
1.40	Automatic Definition	13
1.41	Manuel definition	13
1.42	Genvejskommandoer	13
1.43	Dos-kommando	14
1.44	Luk vindue	14
1.45	Zip vindue	15
1.46	Minimér vindue	15
1.47	Maksimér vindue	15
1.48	Bladre vinduer	15
1.49	Bladre vinduer tilbage	15
1.50	Ændre palette	15
1.51	Skærm frem	16
1.52	Skærm tilbage	16
1.53	Aktivér Workbench	16
1.54	Centrér skærm	16
1.55	Skærmbanking	17
1.56	Indsæt tekst	17
1.57	Indsæt dato	18
1.58	Vis Yak brugerflade	18
1.59	Sæt standard public skærm	18
1.60	Menu genvej	18
1.61	AmigaDos mønstre	19
1.62	Dato format strenge	20
1.63	Problemer	21
1.64	Program historie	21
1.65	Kompilering af Yak	23
1.66	Yak's udviklingshold	23
1.67	Tak til	24
1.68	Kontakt til forfatteren	25
1.69	Mere om genveje	25
1.70	Lokalisering	29

Chapter 1

Yak

1.1 Yak v1.60 Dokumentation

Yak Version 1.60
af Gaël Marziou og Philippe Bastiani
udgivet d. 25. september 1994.

INDHOLDSFORTEGNELSE

Copyright og Distribution
Eksisterende brugere: OBS!
Introduktion
Begrænsninger
Start af Yak
Ændring af opsætning
 Muligheder Mønstre Ventetider Diverse Genveje
GenvejsKommandoer
Mere om Genveje
AmigaDos Mønstre
Dato format strenge
Problemer
 Lokalisering
Program Historie
Kompilering af Yak
Tak til
 Yak's udviklingshold
Kontakt til forfatteren

1.2 Copyright

Yak (selve programmet, source kode og dokumentation) er Copyright © 1992, 1993 Martin W. Scott & Gaël Marziou. Alle Rettigheder forbeholdes.

Yak er gratis at distribuere. Source koden er inkluderet, og du må ændre det til eget brug, men INGEN ændringer må distribueres. Hvis du mener at dine ændringer kan have andres interesse, så send dem til mig og så inkluderer jeg dem i fremtidige versioner.

Eftersom Yak er gratis, er der INGEN GARANTIER. Forfatteren er ikke ansvarlig for tab eller skade som følge af brug af Yak; alt foregår på brugerens eget ansvar.

Der må ikke tages penge for Yak, udover en symbolsk kopieringsafgift. Yak må ikke distribueres sammen med kommercielle produkter uden forfatterens tilladelse. Yak skal distribueres med al dokumentation intakt og uændret, og helst også med source koden. Fred Fish har tilladelse til at distribuere Yak på sine fine samling disketter.

Selvom Yak er FreeWare, er alle DONATIONER MEGET VELKOMNE, enten penge eller noget du selv har skrevet. Se Kontakt til forfatteren Contact}.

1.3 Eksisterende brugere: OBS!

Formatet på Yak's indstillingsfil er ændret igen, hvormed indstillingerne er delt i to filer: "S:Yak.prefs" til indstillinger og "S:Yak.hotkeys" til Genveje. Der er et konverteringsprogram medleveret, der kan konvertere 1.3/1.4 indstillingsfiler til 1.5 formatet. Konverteringen er ikke fuldstændig: Du skal selv indtaste PopCommand og DateFormat strengene ved at bruge det nye Genvej Definitionsvindue (se Genveje om hvordan det gøres). Derudover oprettes en Genvej for "Vis Indstillinger", hvilket sandsynligvis er overflødig, udfra oplysningerne om CX_POPKEY herunder.

Der er nogle vigtige ændringer fra v1.4 til v1.5 af Yak. Vigtigst af alt er kravet til stack forhøjet til 4500 bytes, så husk at ændre det i den ikon, du bruger til at starte Yak.

For det andet: AppIcon er som standard OFF, og skal aktiveres ved at bruge Tool typen APPICON=TRUE. Denne Tool type erstatter Tool typen NOICON.

For det tredje: Tool typen CX_POPKEY bliver brugt til at bestemme hvilken Genvej, der åbner opsætningsvinduet, og denne Genvej kan ikke ændres fra Genvejsvinduet (forskelligt fra tidligere versioner). Du kan dog definere andre Genveje til at åbne Yak's vindue.

Selvfølgelig er der mange andre ændringer og du bør kigge dette dokument igennem for at lære dem at kende.

1.4 Introduktion

Yak står for "Yet Another Kommodity" (Jeg har aldrig været god til at stave - se Vittigheder), og det er et muse/vindue manipulationsprogram i stil med DMouse, MightyMouse etc.

Hvorfor skrive endnu et? Ingen af de andre (og jeg har afprøvet næsten alle sammen) var ikke det helt rigtige. Yak har følgende muligheder:

- o Automatisk aktivering af vinduet under musen, der kun aktiverer når musen stopper. Du kan også specificere hvilke skærme det skal virke på.

(Kompatibelt med popup-menu type programmer)

- o Vinduet kan blive flyttet frem efter automatisk aktivering.
- o TasteAktivering af vinduer
- o Aktivering af vinduer når menu-knappen presses
- o Klik vinduer frem eller tilbage; hvilke skærme det skal virke på, kan specificeres
- o Bladre skærme med musen
- o Muse og Skærm blanking (Genvej for blanking også)
- o Omfattende genvejssystem (som FKey's) med mulighed for:
 - Udføre Dos Kommando
 - Indsætte tekst (indeholdende Genveje)
 - Indsætte dato (i valgfrit format hvis locale installeret)
 - Luk/Skift/Minimér/Maksimér vinduer
 - Bladre Skærme og Vinduer
 - Aktivér WorkBench
 - Centrering af forreste skærm
 - Blanke Skærm
 - Vise en palette på forreste skærm (reqtools nødvendig)
- o TasteKlik med regulérbar volume
- o Ingen Klik (for drev)
- o Wildstar (som StarBurst). Brug '*' som mønster.
- o Valgfri AppIcon til at vise opsætningsvindue
- o Pænt gadtools opsætningsvindue

Har du set det før? Det er en kombination af AutoPoint, ClickToFront, Blanker og IHelp/Fkey standard commodities (på Extras disken), med en anelse KCommodity og DMouse.

Og det er ret lille: Det fylder ca. 29k på disk og ca. det samme når det kører. Denne effektive udnyttelse af hukommelse skyldes brug af overlays; opsætningsvinduet hentes kun ind når det skal bruges. Det betyder at Yak bruger mere hukommelse når du åbner opsætningsvinduet (ca. 45k).

1.5 Vittigheder

Disse er ikke oversat, da de er best på engelsk. Det kan godt være at de er gamle, men de er da gode :)

Q. What's the difference between a goldfish and a goat?

A. A goldfish mucks around in fountains!

Og hvis du forstod den, forstår du helt sikkert også denne...

Q. What's the difference between a magic wand and a truncheon (politistav).
 A. A magic wand is used for cunning stunts.

(Jeg kan **aldrig** sige det sidste svar korrekt)

1.6 Begrænsninger

Yak har kun en almindelig Skærm-blanker. Siden OS 2.0 blev introduceret er der kommet en sand strøm af fancy Skærm-blankere, såsom Spliner, ASwarm (i forskellige udgaver), FracBlank og mange flere. Jeg tror de fleste har deres egen favorit, så derfor er der ikke noget specielt i Yak.

Der er ingen muse-acceleration. Den, der er indbygget i systemet (Input Prefs) synes jeg er tilstrækkelig. Hvis jeg får nok forespørgsler, vil jeg tilføje hurtigere acceleration (indtil nu har jeg ikke fået nogen).

1.7 Start af Yak

Det er meningen at Yak skal startes fra sys:WBStartup skuffen, men det kan køres fra Shell (hvilket dog ikke er anbefalelsesværdigt, da det vil bruge væsentlig mere hukommelse). Alle Yak indstillinger gemmes i to filer, en opsætningsfil "S:Yak.prefs", og en Genvejsfil, "S:Yak.hotkeys". Det er best at benytte installationen, og så første gang du bruger Yak, sæt det op som du kan lide det, og gem opsætningen til senere brug.

NB: Yak skal bruge 4500 bytes i stack. Sæt denne værdi i Yak's ikon, eller hvis du starter fra Shell:

```
stack 4500
run >nil: yak
```

De eneste Tool typer Yak bruger er WorkBench standarderne (såsom DONOTWAIT), standarderne for Commodities (såsom CX_POPUP), dem der bruges til AppIcon og sprog (LANGUAGE). Tool typerne er:

Tool	Type	Kategori	Beskrivelse	Default
CX_POPKEY	HOTKEY		Tast for opsætningsvindue	RCommand Help
CX_PRIORITY	HELTAL		Prioritet for dette commodity	0
CX_POPUP	BOOLEAN		Vis opsætning ve opstart	NO
APPICON	BOOLEAN		Hvis TRUE, vises AppIcon	FALSE
ICONNAME	STRENG		Navn på AppIcon	"Yak!"
ICONXPOS	HELTAL		x-koordinate for AppIcon	flydende
ICONYPOS	HELTAL		y-koordinate for AppIcon	flydende
LANGUAGE	STRENG		Navn på sprog	Ikke specificeret

Du bør altid bruge Tool typen "DONOTWAIT", hvis du starter Yak fra SYS:WBStartup.

AppIcon faciliteten er valgfri og er som standard slået fra. Skriv

APPICON=TRUE i Yak's ikon, hvis du vil benytte den. Når AppIcon benyttes putter Yak en AppIcon på Workbench skærmen, og når du dobbeltklikker på den, kommer opsætningsvindues frem. Selve ikonen er den samme som den, der blev brugt til at starte Yak, hvilket giver dig mulighed for at ændre AppIconens udseende efter ønske.

CX_PRIORITY kan være nødvendig for at Yak kan virke korrekt sammen med andre af dine Commodities. Se for eksempel omtalen af RMB/MMB ← Aktivéring.

LANGUAGE Tool typen skal du bruge for at få Yak til at "tale" et andet sprog end det indbyggede engelsk. For at få Yak til at "tale" dansk skal du skrive:

```
LANGUAGE=dansk
```

1.8 Ændring af Yak's opsætning

Alle Yak's indstillinger kan ændres ved at bruge opsætningsvinduet. Når Yak er startet, kan dette vindue kaldes frem på en af følgende måder:

Tryk RCommand Help (dvs. højre Amiga taste og Help tasten.)
Denne kombination kan ændres vha. Tool typen CX_POPKEY.

Dobbelt klik på Yak's AppIcon (hvis denne Tool type er valgt)

Starte Yak igen (fra ikonen på Workbench)

Via Commodities Exchange programmet (findes på Extras disken).

Vinduet indeholder mange knapper, der er grupperet i klasser. Klasserne er:

- Muligheder
- Mønstre
- Ventetider
- Diverse
- Genveje

Husk at trykke RETURN, ENTER eller TAB efter at have ændret i en tekstramme. Ellers bliver ændringen ikke registreret. Hvis du bare klikker uden for boksen, sker der ingen ændring. (TAB aktiverer den næste streng, så den kan redigeres).

Det er heller ikke muligt at kopiere tekst mellem to tekstrammer, hvilket ellers kan være behageligt, specielt i Vinduet med definitioner af Genveje. Hvis du vil gøre dette muligt, må du anskaffe "NewEdit" af Uwe Roehm. Dette program muliggør kopiering til og fra Clipboard. Det kan findes på en Fred Fish disk og FTP.

Desuden udfører de to knapper "Gem" og "Afslut" standard operationerne for Commodities ved af hhv. gemme opsætningsvinduet og afslutte Yak. Ved at trykke på vinduets lukke knap (øverste venstre hjørne) gemmer kun opsætningsvinduet og afslutter ikke Yak.

Opsætningsvinduet har også "Gem" og "Afslut" som menupunkter, der virker på

samme måde som knapperne med samme navn. Derudover er der menupunkterne "Gem opsætning" og "Hent opsætning". Alle ændringer i Yak's opsætning vil gå tabt med mindre du vælger "Gem opsætning". Hvis du vil tilbage til din sidst gemte opsætning, kan du benytte "Hent opsætning".

1.9 Muligheder

Du har følgende muligheder med Yak:

- Aktivering under mus
- Flyt vindue frem
- Tasteaktivering
- Klik vindue frem
- Klik vindue tilbage
- MMB-aktivering
- RMB-aktivering
- Bladre skærme
- Skærmaktivering
- MMB Shift

1.10 Aktivering under mus

Automatisk aktivering af vindue under mus. Dette opfører sig nærmest ligesom Commodore's AutoPoint Commodity, idet det kun aktiverer vinduet når musen stopper. Det er kompatibelt med popup-menu programmer som det glimrende "MagicMenu".

Bemærk at "Aktivering af vindue" og "Flyt vindue frem" kun finder sted, hvis der IKKE presses en valg-tast (mus eller tastatur). Derved undgås konflikter med andre programmer, og det er muligt at undgå aktivering/fremflytning hvis det er uønsket.

Se også: Flyt vindue frem Aktivering af skærme

1.11 Flyt vindue frem

Virker kun, hvis der er valgt "Aktivering under mus". Udover at aktivere vinduet, bliver det også flyttet frem, undtagen når vinduet under musen har et forespørgsels-vindue åbent.

Se også: Aktivering under mus Aktivering af skærme Flytte vinduer frem

1.12 RMB-aktivering

Når dette vælges, bliver vinduet under musen aktiveret når der trykkes på højre muse knap, uanset om der valgt "Aktivering under mus" eller ej. Dette kan være nyttigt til at få den menu, du ønsker uden at vente på

"Aktivering under mus" eller selv klikke i vinduet for at gøre det aktivt.

For eksempel, hvis du bruger dybdeknappen til at komme til en anden skærm, bliver den nye skærm ikke aktiveret. Hvis du har valgt "RMB-aktivering" kan du trykke på den højre muse tast, og de korrekte menuer dukker op.

Bemærkning 1: Hvis der ikke er et vindue under musen. bliver det første vindue på skærmen aktiveret.

Bemærkning 2: For at virke korrekt med popup-menu programmer, kan det være nødvendigt at sætte Yak's Tool type CX_PRIORITY højere end popup-menu programmets CX_PRIORITY.

1.13 MMB-aktivering

Når dette er valgt, vil vinduet under musen blive valgt, når der trykkes på den midterste museknap, uanset om der valgt "Aktivering under mus". Dette kan være nyttigt til at få den menu, du ønsker uden at vente på "Aktivering under mus" eller selv klikke i vinduet for at gøre det aktivt.

Bemærkning 1: Hvis der ikke er et vindue under musen. bliver det første vindue på skærmen aktiveret.

1.14 Tasteaktivering

Aktiverer vinduet under musen når der trykkes på en tast. Det er kun nødvendigt med enten "Aktivering under mus" eller "Tasteaktivering".

1.15 Klik vindue frem

Gør dig i stand til at få et vindue foran de andre ved at dobbelt klikke i det.

Som med "Aktivering under mus"/"Flyt vindue frem" er det muligt a undgå dette ved at trykke på en valg-tast.

Når Bladre skærme er aktiv, virker dette også for skærme.

Se også: Klikskærme Klikke vinduer frem

1.16 Klik vindue tilbage

Gør dig i stand til at skubbe et vindue bag de andre ved at trykke og holde den venstre museknap og derefter trykke på den højre museknap.

Som med "Aktivering under mus"/"Flyt vindue frem" er det muligt a undgå dette ved at trykke på en valg-tast.

Når Bladre skærme er aktiv, virker dette også for skærme.

Se også: Klikskærme

1.17 Bladre skærme

Ved at bruge den samme sekvens med musen som "Klik vindue frem/tilbage", kan en skærm flyttes frem/tilbage i forhold til de andre, hvis:

enten vinduet under musen er et Baggrundsvindue
(som Workbench vinduet)
eller der kun er ét vindue på skærmen.

Se også: Klik vindue tilbage Klik vindue frem

1.18 Skærmaktivering

Når denne mulighed er valgt, aktiverer Yak de skærme der bliver bladret frem til ved at benytte genvejen for "Bladre skærme" eller venstre-Amiga-m. Det er en valgmulighed for at undgå konflikter med visse programmer. Funktionen virker som (men ikke identisk med) Steve Tibbet's WindX.

Ved skærmaktivering mener jeg, at der aktiveres et vindue på skærmen. Dette vindue kan være:

- det vindue, der var aktivt, sidst denne skærm blev besøgt efter bladrning vha. Yaks genveje.
- det vindue, der er under musepilen bliver aktiveret, hvis skærmen aldrig har været besøgt.
- det første vindue på skærmen, hvis begge førnævnte betingelser fejler.

1.19 '*' i mønster

Når dette vælges, er det muligt at benytte * som AmigaDos mønster. Dette bruges i MSDOS og UNIX, hvor der normalt på Amigaen benyttes #?. Det virker ligesom programmet StarBurst.

1.20 Ingen drev-klik

Når dette vælges stopper dine disketteredrev med at "klikke" når der ikke er en diskette i.

1.21 Inkluderingsmønstre

Mønster tekstrammerne gør dig i stand til at specificere hvilke skærme og vinduer, der skal være følsomme for Yak's forskellige muligheder. Alle mønstrene er inkluderingsmønstre, dvs. at titlen på vinduet eller skærmen skal PASSE MED mønstret for at den specielle funktion virker. Alle mønstre er standard AmigaDos mønstre .

Følgende mønstre er tilgængelige:

```
Aktivering af skærme
Klikskærme
Flytte vinduer frem
Klikke vinduer frem
```

1.22 Aktivering af skærme

Aktivering under mus virker på skærme, hvis titel passer med dette mønster.

```
Standardtitel på en skærm
```

1.23 Klikskærme

Klik vindue frem og Klik vindue tilbage virker på skærme, hvis standardtitel passer med dette mønster.

```
Standardtitel på en skærm
```

1.24 Standardtitel på en skærm

Standardtitlen på en skærm varierer ikke. Vinduer fra diverse programmer kan ændre skærmens titel, men ikke standardtitlen på skærmen. For eksempel bliver titlen på Workbenchskærmen tit ændret af programmer, hvis et af programmets vinduer er aktivt, men standardtitlen forbliver den samme. Så når der skal vælges et skærmønster, skal du bare benytte standardtitlen på skærmen. Det er den titel, der ses når ingen vinduer er aktive. Det vil i de fleste tilfælde være meget nemt at finde standardtitlen på en skærm, men en gang i mellem kan det være nødvendigt at bruge et program som ARTM.

Se også AmigaDos mønstre .

1.25 Flytte vinduer frem

Flyt vindue frem virker på skærme, hvis titel passer med dette mønster.

1.26 Klikke vinduer frem

Klik vindue frem virker på skærme, hvis titel passer med dette mønster.

1.27 Diverse opsætninger

Dette vindue indeholder et par enkelte funktioner, der kontrolleres via disse knapper og tekstrammer. De er:

```
Klikvolume
Aktiveringsforsinkelse
'*' i mønster           Sort kant
Ingen drev-klik
```

1.28 Klikvolume

Kontrollerer lydstyrken af TasteKlik, der er den lyd der frembringes når du taster en tast. Værdien nul betyder 'ingen klik' (det lyder logisk, men der bliver ikke engang åbnet for lyden). Maksimum er 64.

1.29 Aktiveringsforsinkelse

Kontrollerer hvor lang tid Yak skal vente efter musen er stoppet over et vindue, før det bliver aktiveret. Denne værdi skal ligge mellem 0 og 5, hvor hvert trin svarer til 10ms.

En forsinkelse på nul betyder ingen forsinkelse. Logisk, eller hvad?

1.30 MMB Shift

Når dette vælges, vil Yak fortolke den midterste muse knap (på en mus med 3 knapper) som den venstre shift-taste.

Faktisk sker det kun når du trykker den venstre muse knap ned mens du holder den midterste muse knap nede. Dette gør det lettere at vælge flere ikoner på workbench eller flere filer i en fil-vælger uden at røre tastaturet.

Der sker ikke noget med eventuelle andre kombinationer for genveje, som den midterste muse knap indgår i.

1.31 Sort kant

Når dette er valgt vil Yak lægge en sort kant rundt om alle dine skærme. Dette er kun muligt under Workbench 3.0 eller nyere, da det benytter et nyt flag i graphics.library.

1.32 Blanking vindue

Museblanking metode

Blank mus ved tastetryk

Ventetider

1.33 Museblankingsmetode

Denne knap bestemmer hvilken metode, der benyttes til at blanke musepilen. 'Ingen' slår museblanking helt fra, 'Sprites' blanker musen ved at slukke for (alle) sprites, og 'Copper' blanker musen ved at ændre copper-listen. Den sidste mulighed slukker kun for sprite 0 (musepilen), så terminal programmer, der benytter en sprite vil virke upåklageligt, men metoden er lidt mindre robust, så det kan ske at musen kommer frem igen et øjeblik.

Se også: Problemer

1.34 Blank mus ved tastetryk

Hvis dette er valgt, bliver musepilen blank lige så snart du trykker på en tast. Hvis det ikke er valg, bliver musen kun blanket efter det angivne interval.

1.35 Ventetider

I disse tekstrammer angives, hvor lang tid, der skal gå før skærm hhv. mus skal blankes.

Skærm ventetid

Muse ventetid

1.36 Skærmbanking

Hvis der ikke kommer noget input fra brugeren via mus eller tastatur i denne periode, som angives i sekunder, bliver skærmen sort. Sættes værdien til 0, bliver skærmbanking slået helt fra. Blankingen sker ved at åbne en skærm med 2 farver i samme opløsning som den forreste skærm.

1.37 Museblanking

Hvis musen ikke bliver flyttet i den specificerede periode, forsvinder musepilen. Dette virker ikke hvis Museblankingsmetode er sat til 'Ingen'.

1.38 Definerings af Yak Genveje

Ved at klikke på "Redigér genveje" åbnes et nyt vindue, der lader dig definere, redigere og slette genveje. En genvej er en tastekombination, der udløser en eller anden form for reaktion. Der er mange reaktioner at vælge iblandt i Yak. Genveje defineres ved at benytte en beskrivelse af den genvejsstreng, man ønsker. Denne beskrivelse er en meget fleksibel måde at beskrive input på.

Du kan have lige så mange genveje, som du ønsker, og hver reaktion kan fremkaldes af mere end én genvej.

Der er to lister i dette vindue. Den til venstre indeholder de mulige kommandoer, og den til højre viser de genveje, der i øjeblikket er defineret for denne kommando.

For at tilføje en ny genvej, må du først vælge den kommando, du ønsker den skal udføre (ved at klikke på den i den venstre liste). Derefter klikker du på Tilføj knappen under genvejslisten.

For at redigere en eksisterende genvej, dobbeltklikker du på genvejen i den højre liste.

For at slette en genvej fra Yak's liste, skal du bruge Slet knappen.

For visse kommandoer, vil der være ekstra muligheder. Knappen under Genvejs-listen bestemmer hvad der skal ske når den aktuelle genvej aktiveres. For øjeblikket er det muligt at der ikke sker nogen ændring, at Workbenchskærmen kommer forrest eller at Standard Public Skærm kommer forrest. Dette vil typisk blive brugt ved Dos-kommando genveje til at vise hvilken skærm det åbnede vindue bliver åbnet på.

Argument rammen er en streng ramme, der bliver aktiv for visse Kommandoer, og det er via denne ramme muligt definere en streng, der tilknyttes genvejen.

Når du er færdig med at redigere genvejene, klikker du på "Tilbage..." for at komme til opsætningsvinduet, eller klik på lukke-knappen for at 'gemme' Yak. Husk at gemme dine ændringer ved at bruge "Gem Opsætning" i menuen i opsætningsvinduet.

Se Genvejskommandoer
Beskrivelse af genveje

1.39 Beskrivelse af genveje

Her kan du definere eller ændre de genveje, der bruges af Yak.

Beskrivelser af de eksisterende genveje er vist i tekstrammen. Der er to måder at definere en genvej på:

Automatisk definition

Manuel definition

Efter at have ændret en genvej, skal du klikke på Ok-knappen for at registrere den nye beskrivelse.

Klik på Afbryd for at annullere.

Se også: Problemer

1.40 Automatic Definition

Alt du behøver er at vælge genvejs-klassen i den venstre liste, og derefter trykke de taster og/eller museknapper, du vil bruge som genvej. Yak konverterer dem automatisk om til en beskrivelse.

Hvis du trykker en tast to gange, bliver den fjernet fra beskrivelsen.

Med Tast-listen til højre kan du vælge, hvordan genvejen skal aktiveres.

Se også: Mere om genveje

1.41 Manuel definition

Klik i tekstrammen og skriv selv beskrivelsen af genvejen. Yak oversætter strengen og opdaterer knapper osv.

OBS: 'NewPrefs', 'PointerPos', 'Timer' og 'Event' klasserne er endnu ikke understøttet!

ADVARSEL: commodities.library har nogle fejl:

- her er en forkert beskrivelse:
`'NumericPad a'`

- her er nogle ubrugelige beskrivelser:
`'NumericPad [' and 'NumericPad]'` (med et fransk tastatur)

Se også: Mere om genveje

1.42 Genvejskommandoer

De mange kommandoer, der er til rådighed er:

- Dos-kommando
- Luk vindue
- Zip vindue
- Minimér vindue
- Maksimér vindue
- Bladre vinduer
- Ændre palette
- Skærm frem

Skærm tilbage
Aktivér Workbench
Centrer skærm
Skærmlanking
Indsæt tekst
Indsæt dato
Vis Yak grænseflade
Bladre vinduer tilbage
 Sæt standard public skærm
Menu genvej

1.43 Dos-kommando

Argument: Den kommando, der skal udføres

Udfører den Dos-kommando, der er specificeret i Argument tekstrammen. Bemærk at du kan benytte 'Muligheder' til at flytte Workbench eller standard public skærm forrest. Det er nyttigt, hvis kommandoen åbner et vindue.

Kommandoen køres asynkront, så det er ikke nødvendigt at starte den med 'run'. Hvis kommandoen genererer output (eller kræver input) åbnes et konsol vindue. Du kan selvfølgelig benytte omdirigeringspile som i Shell'en.

Eksempler:

En genvej, der åbner et shell-vindue:

Dette knyttes pr. tradition til genvejen 'lcommand esc', og min er sat op til at køre kommandoen

```
"NewShell CON:79/177/582/78/AmigaShell/CLOSE/ALT2/58/660/197"
```

Bemærk brugen af ALT-flaget i konsol-specifikationen. Det er dårligt dokumenteret (læs: ikke nævnt). Faktisk bruger jeg to genveje, en til at starte en normal shell og en til at starte en CShell.

En genvej til at frigive ubrugt hukommelse:

SAS/C bruger shared libraries, der ofte kan optage værdifuld chip-ram. Jeg har en genvej sat op med kommandoen 'avail >nil: flush', der frigiver denne hukommelse.

En genvej, der viser indholdet af alle disketter, der indsættes:

Sæt beskrivelsen til "DiskIndsat", og skriv "Dir df0:" i argument-tekstrammen.

Se også: Problemer Mere om genveje

1.44 Luk vindue

Argument: INGEN

Lukker det aktive vindue. Det svarer til at klikke på vinduets luk-knap.

1.45 Zip vindue

Argument: INGEN

Zip det aktive vindue. Det svarer til at klikke på vinduets 'Skift størrelse' knap.

1.46 Minimér vindue

Argument: INGEN

Gør det aktive vindue så lille som muligt.

1.47 Maksimér vindue

Argument: INGEN

Gør det aktive vindue så stort som muligt.

1.48 Bladre vinduer

Argument: INGEN

Kald det nederste vindue frem. Nyttigt til at få fat i vinduer, der er godt begravet.

1.49 Bladre vinduer tilbage

Argument: INGEN

Virker modsat 'Bladre vinduer': lægger det forreste vindue nederst.

1.50 Ændre palette

Argument: INGEN

Åbner en palette på den forreste skærm. Paletten kører asynkront og du kan åbne lige så mange som du vil (begrænset af hukommelse). Yak kan dog ikke afsluttes mens der er åbne paletter. Det er muligt at benytte 'Muligheder' til at åbne paletten på en bestemt skærm, hvis du ønsker det.

BEMÆRK: Du skal have `retools.library` installeret for at kunne åbne paletter.

ADVARSEL: Luk altid paletten før du lukker skærmen, ellers ender du i bedste fald med en åben skærm eller i værste fald går computeren ned.

1.51 Skærm frem

Argument: INGEN

Kalder den bageste skærm frem.

Se også: Skærmaktivering

1.52 Skærm tilbage

Argument: INGEN

Put den forreste skærm bag alle de andre.

Se også: Skærmaktivering

1.53 Aktivér Workbench

Argument: INGEN

Aktiverer et vindue på Workbench, og kalder Workbench skærmen forrest, hvis det er nødvendigt. Dette gør dig i stand til at benytte Workbench menuerne uden at du behøver at aktivere et Workbench vindue. Nyttigt hvis du for eksempel har en shell på størrelse med skærmen åben.

1.54 Centrér skærm

Argument: INGEN

Vandret centrering af den forreste skærm.

1.55 Skærmbanking

Argument: INGEN

Blanker skærmen med det samme.

Se også: Ventetider .

1.56 Indsæt tekst

Argument: tekst der skal indsættes

Indsætter den tekst, der er specificeret i Argument tekstrammen ind i læsestrømmen.

Strengen behandles således:

```
\n   konverteres til linieskift
\r   konverteres til linieskift
\t   konverteres til tabulator
\ \  konverteres til backslash \
<genvejsbesk.> konverteres til specificerede genvej
\<    konverteres til <
```

Ved hjælp af denne konvertering kan indtætte-strengene udføre mange nyttige ting. For eksempel har jeg en genvej, der indsætter mit navn og datoen. Dette gøres ved at skrive følgende i Argument tekstrammen (uden "):

```
"Martin W Scott, <lcommand d>"
```

Her er 'lcommand d' en anden genvej jeg har sat op til at indsætte datoen. ved at bruge mere komplicerede strenge, kan du lave simple makroer for andre programmer.

PAS PÅ: Når du bruger at kalde en anden genvej, skal du være opmærksom på at du ikke bruger rekursive defintioner som f.eks.

```
f1 indsæt tekst "<f2>"
f2 indsæt tekst "<f1>"
```

Hvis der trykkes f1 eller f2 opstår der en uendelig løkke. Hvis du er dum nok til at gøre dette, kan du starte Commodities Exchange og gøre Yak Inaktiv. Vælg derefter Vise brugerflade og slet eller omdefinér de forkerte genveje.

Du skal også være opmærksom på at strenge, der kalder andre genveje (som eksemplet ovenfor) ikke nødvendigvis virker som du forventer. Hvis du for eksempel skrev '<lcommand d>\n' i Argument tekstrammen, forventer du at der kommer en dato efterfulgt af et linieskift. Faktisk sker det i omvendt rækkefølge, fordi på det tidspunkt Yak får 'lcommand d' genvejen har linieskiftet været igennem input-handleren og er sendt til det aktive vindue.

1.57 Indsæt dato

Argument: dato format streng

Indsætter datoen ind i læse strømmen, og dermed ind i det aktive vindue. Hvis du bruger AmigaDos 2.1 eller nyere kan du specificere formatet helt efter eget ønske. Formatet skrives i Argument tekstrammen. Se Dato formatstreng

Hvis du er uheldig nok til at bruge AmigaDos 2.0, bliver datoen i DD-MM-ÅÅ formatet.

Eksempel: formatet "%e %B %Y" genererer datoer af formen "1 Maj 1993"

1.58 Vis Yak brugerflade

Argument: INGEN

Viser Yak's opsætningsvindue. Vinduet åbner på den aktive skærm, hvis det er en public skærm, ellers kalder den standard public skærm forrest før den åbner vinduet på den. Det er samme funktion som udføres af CX_POPKEY.

1.59 Sæt standard public skærm

Argument: INGEN

Sætter den nuværende skærm som standard. Det virker kun hvis den aktive skærm er public.

1.60 Menu genvej

Argument: menu, emne og eventuelt underemne

Sender et input event til menuen i det aktive vindue, og simulerer derved et valg i menuen (hvis der er én). Dette kan være nyttigt til at give genveje til programmer der ikke har nogen eller ikke har nok.

Argument strengen specificerer menuens nummer, emne nummeret og eventuelt underemne nummeret. Alle disse numre adskilles med mellemrum og starter ved 0 ifølge Intuitions regler (dvs. at den første menu er nummer 0 og ikke 1).

Bemærk at en adskiller i menuen tæller som et emne, så hvis der er flere adskillere før det emne du ønsker, må du tælle dem med for at få det korrekte emne nummer.

Vær dog ikke bange for at lave en fejl i specifikationen af argument strengen, da Yak udfører alle nødvendige check før der sendes et input event for ikke at forstyrre programmet.

Her er et par eksempler på argument strenge:

Du synes, der ikke er nok genveje i Workbenchen:

For at få en genvej til "Sidste melding" i "Workbench" menuen: "0 4"

For at få en genvej til "Kun ikoner" underemnet i "Vise" undermenuen til "Vinduer" menuen: "1 7 0"

1.61 AmigaDos mønstre

AmigaDos mønstre bruges til at inkludere/ekskludere en liste af navngivne skærme/vinduer for en bestemt funktion. Mønster specifikationerne er en hjælp til kompatibilitet med andre programmer, du bruger.

Mønstrene tager højde for små og store bogstaver, dvs. at "Amiga" ikke er det samme som "AMIGA".

Standard AmigaDos mønstre er:

```
? Mønster for en enkelt karakter
# Gentager det efterfølgende udtryk 0 eller flere gange
(ab|cd) Mønster for ét af de udtryk, der er adskilt med '|'
~ Negerer det efterfølgende udtryk. Det passer på alle
  strenge der ikke passer til udtrykket. Dvs. ~(bøh) passer
  på alle strenge, der ikke er præcis "bøh"
[abc] Karakter klasse: passer på hvilken som helst af karaktererne
  i klassen
a-z Karakter område (kun i karakter klasser)
% Passer altid på 0 karakterer (nyttig i "(foo|bar|%)").
* Synonym for "#?", ikke tilgængelig som standard.
  Tilgængelig hvis '*' i mønster er valgt.
```

Hvis du ikke er vant til at bruge mønstre, kan dette synes ganske overvældende. Læs i din computers manual for yderligere detaljer.

Der er umiddelbart to ting du kan bruge det til: Enten at lave en endelig liste af navne, hvor en bestemt funktion skal virke, eller en endelig liste af navne, hvor en bestemt funktion ikke skal virke. For at få en funktion til at VIRKE for alle objekter (det være skærme eller vinduer, som det nu passer) kan du benytte mønstret "#?", der passer på alt. For at en funktion skal virke på N objekter med navnene "navn1" til "navnN" skal du bruge

```
(navn1|navn2| ... |navnN)
```

og for at den IKKE skal virke for disse navne, sætter du en tilde ~ foran som

```
~(navn1|navn2| ... |navnN)
```

Et eksempel: Jeg ønsker ikke at "Flytte vinduer frem" skal flytte Workbench vinduet eller nogen vinduer fra ProText (tekstbehandling fra Arnor) frem, så jeg ekskluderer dem med mønstret

```
~(Workbench|#?Arnor#?)
```

Bemærk at det andet 'navn' egentlig er et mønster, der passer på alle titler

med teksten "Arnor" i.

Endnu et eksempel: Jeg ønsker ikke at aktivere Directory Opus's skærm, når jeg bladrer om til den. Dette program viser ikke sin titel i titelbjælken (afhænging af hvilken version det er), så jeg bliver nødt til at bruge ARTM eller Xoper for at finde skærmens navn, og finder ud af at det er "DOPUS.1". Jeg finder ud af at "1" sikkert bliver til "2", hvis jeg startede det endnu en gang, og jeg bestemmer mig for at ekskludere alle DOpus skærme ved at bruge

```
~(DOPUS#?)
```

Endnu endnu et eksempel: Jeg ønsker ikke "Aktivering af skærme" på BrowserII's skærm. Standardtitlen (når der ikke er nogle aktive vinduer) er BrowserII efterfulgt af forfatterens navne og copyright årene. Disse år vil ændres når der kommer en ny version, så derfor ekskluderer jeg skærmen ved at skrive:

```
~(BrowserII#?)
```

Bemærkning: Skærme eller vinduer med udefinerede titler (dvs. sat til NULL) passer altid til mønstrene.

1.62 Dato format strenge

I dato indsætningsgenvejene, skal du specificere en dato format streng i locale-format (hvis du bruger AmigaDos 2.1 eller nyere). De følgende muligheder er tilgængelige i locale.library:

```
%a - forkortet navn på ugedag
%A - navn på ugedag
%b - forkortet månedsnavn
%B - månedsnavn
%c - samme som "%a %b %d %H:%M:%S %Y"
%C - samme som "%a %b %e %T %Z %Y"
%d - dagens nummer med 0 foran
%D - samme som "%m/%d/%y"
%e - dagens nummer mellemrum foran
%h - forkortet månedsnavn
%H - time med brug af 24 timers ur med 0 foran
%I - time med brug af 12 timers ur med 0 foran
%j - juliansk dato
%m - måneds nummer med 0 foran
%M - minutter med 0 foran
%n - indsæt linieskift
%p - AM eller PM strenge
%q - time med brug af 24 timers ur
%Q - time med brug af 12 timers ur
%r - samme som "%I:%M:%S %p"
%R - samme som "%H:%M"
%S - sekunder med 0 foran
%t - indsæt tabulator
%T - samme som "%H:%M:%S"
%U - uge nummer, hvor søndag betragtes som den første ugedag
%w - dagens nummer
```

%W - uge nummer, hvor mandag betragtes som den første ugedag
%x - samme som "%m/%d/%y"
%X - samme som "%H:%M:%S"
%y - årstal med to cifre med 0 foran
%Y - årstal med fire cifre med 0 foran

Det er en temmelig udtømmende liste, der skulle tage højde for de flestes behov. Du kan indsætte tekst efter ønske i format strengen. Et par eksempler:

```
"Klokken er %X"   giver (f.eks.) "Klokken er 20:44:16"  
"Hav en god %A!" giver (f.eks.) "Hav en god mandag!"
```

Hvis du vil have flere detaljer, må du ty til AutoDocs for locale.library, hvis du har dem.

1.63 Problemer

Der er et par problemer, som jeg i øjeblikket er klar over. Først og fremmest vil en shell fremkaldt af en Dos-kommando genvej ikke have stack eller være i den skuffe der blev bestemt ved opstart (i startup-sequence). Den opsatte sti er der dog stadig. Tilføj stack og CD i din Shell-Startup fil. Som standard har processer, der startet på denne måde opstarts disken (SYS:) som nuværende skuffe.

Hvis du ikke er tilfreds med Yak's museblanking, kan du prøve Commodores MouseBlank commodity fra WB3.0. Det skulle blanke musen korrekt på alle skærme. På AGA maskiner vil Copper-blanking give problemer, hvis du benytter en højopløsnings-pil. Brug Sprites som blanking metode istedet for.

BEMÆRKNING TIL OneKeyII brugere: Gør OneKeyII inaktiv når du ændrer en genvejs beskrivelse!

BEMÆRKNING TIL AMOS BRUGERE: Jeg kan ikke lide AMOS (nok om min mening), dels fordi det er så system uvenligt. Det overtager fuldstændig input strømmen, så muse blankere (i programmer som Yak) træder i kraft, fordi de tror der ikke har været noget input, og musen bliver ikke synlig igen fordi der ikke er nogen bevægelse med musen til at gøre det. Eftersom Yak benytter en rimelig dårlig blanking metode, kan der opstå problemer med at musen forsvinder og ikke kommer frem igen. Det er der to løsninger på:

- 1) Benyt 'Copper' blanking
- 2) Sæt Ventetiden for museblanking til nul. Musen blankes stadig ved tastetryk, men den blankes ikke efter et stykke tid.

På denne måde skulle problemet med den forsvindende mus i AMOS være løst.

1.64 Program historie

(* = nyhed)

v1.60

- * 'MMB Shift' mulighed tilføjet.
- Fejl rettet i installer script vedrørende sprog valg.
- Muse blanking virker nu bedre med et Picasso II grafik kort, men der er stadig et problem. Vil nogen være så venlig at give mig sådan et kort, så jeg kan rette problemet, det vil være en dejlig gave :-)
- Fejl ved initialisering af Ventetider-rammen i blanking-vinduet rettet.
- Skærmbanking bliver nu foretaget i Yaks hovedrutine for at undgå at input.device hænger når skærm-promotions programmer forsøger at promotere Yaks sorte skærm.

v1.59

- Enforcer hit ved tryk på 'Diverse' fjernet.
- Enforcer hit ved åbning af en palette fjernet.
- Fejl rettet: Klik vindue frem knappen havde ingen virkning.
- Fejl rettet: Skærmaktiveringsknappen havde ingen virkning.
- Fejl rettet: Ingen drev-klik og '*' i mønster knapperne var inverterede.
- Fejl rettet: Museblankingsmetodeknapp havde ingen virkning.
- Fejl rettet: Automatisk definition af genveje brugte V38 nøgleord og kunne derfor ikke bruges med V37.
- Fejl rettet: Klik vindue frem og Klik vindue tilbage fik computeren til at gå ned.
- * Yak husker nu det vindue, der var sidst aktiveret sidste gang en skærm var fremme vha. bladring af skærmene.
- * Ny genvej "Menu genvej". Bruges til at vælge et emne fra menuen.

v1.58

- Fejl i 'Bladre vinduer tilbage' genvej fjernet.
 - Fejl i tekstrammerne for 'Klikvolum' og 'Aktiveringsforsinkelse' fjernet. Værdier udenfor de gyldige grænser fik computeren til at gå ned.
 - Fejl ved museblanking når Yak blev gjort inaktiv vha. CLIXchange fjernet.
 - Fejl ved afslutning af Yak fjernet. Det var opstået igen i v1.57.
-

- Fejl i 'Klik vindue tilbage'.
- * Når alt-tasten trykkes blanker Yak ikke musen. Dette gør det nemmere at benytte menuerne fra tastaturet.
- * Nu muligt at kompilere Yak med både SAS/C 6.51 og DICE !
- * Sort kant for skærme (kun for 3.0+)
- * Nyt udseende for opsætningsvinduet designet af Reza Elghazi.
- * Palette titlen lokaliseret.
- * Skærm mønstre bruger nu skærmens standard titel i stedet for den nuværende titel. Det gør det lettere at lave mønstre for skærme som Workbench og BrowserII.
- * Ny genvej til at sætte standard public skærm.
- * Ny metode til automatisk definition af genveje ved at taste den. Programmeret af Philippe Bastiani.
- * Nu kan Yak kun åbne en palette pr. skærm.
- * Tilføjet dansk katalog, installations script og dokumentation.
- * Overlay kode fjernet.

Se den engelske dokumentation for historie før v1.58.

1.65 Kompilering af Yak

Yak er skrevet så det kan kompileres af SAS/C, og fra og med v1.4a er der brugt SAS/C 6.2. Koden er ren ANSI, så brugere af andre kompilere, der overholder ANSI-C, skulle ikke have problemer med at kompilere Yak - de eneste steder der skal ændres er de nøgleord der er specifikke for SAS/C (som `__saveds`).

Yak bruger overlays, men dette kan kompileres ud ved at fjerne `USE_OVERLAYS` definitionen i `SCOPTIONS`.

GTB 2.0 blev brugt til at generere brugerfladen. Den genererede kode (`popup.c`, `popup.h`) skal kun ændres en anelse, specielt til at bruge `NewLookMenus` under `WB3.0` og nyere. Ændringer er indiceret ved kommentarer, der begynder med fire stjerner, f.eks. `/**** ADDED */`.

1.66 Yak's udviklingshold

Udviklingen af Yak er teamwork.

Programmering

Gael Marziou
Philippe Bastiani
Reza Elghazi (GUI)

Oversættelser, forslag og beta test

Alex Galassi : italiensk dokumentation, installer script og katalog.
Ingolf Koch : tysk dokumentation, installer script og katalog.
Christian Høj : dansk dokumentation, installer script og katalog.
Patrick van Beem : hollandsk installer script og katalog.
Johan Billing : svensk katalog.
Peter Eriksson : svensk installer script.

1.67 Tak til

Yak er skrevet udelukkende i C, og kompileret med SAS/C 6.3. Tak til Reza Elghazi for hjælpen ved overgangen til 6.2.

Udseendet af opsætningsvinduet er kreeret ved hjælp af det glimrende (bortset fra den lille fejl i v2.0 :) GadToolsBox v2.0b fra Jaba Development.

Yak bruger reqtools.library, der er Copyright Nico François.
Også en tak til Steve Koren for SKsh, Matt Dillon for DMouse (som svarede mange af mine "hvordan gør jeg det"-spørgsmål) og Kai Iske for KCommodity, hvorfra lyden til TasteKlik er 'lånt'.
Yak bruger også WB2CLI, et meget nyttigt lille link-modul af Mike Sinz.

Tak til Heinz Wrobel for hans vidunderlige version af RCS til Amiga, til Pierre Carette og Sylvain Rougier for BrowserII, Martin Korndörfer for MagicMenu og til mine venner fra den franske Amiga post-liste, der hjalp mig med den franske version af Yak.

Genvejs definitions-dokumentationen er taget fra Toolmanager, med tilladelse fra Stefan Becker.

Tak til Stefan Sticht for hans Public Domain MouseBlanker commodity - herfra fandt jeg ud af 'Copper'-metoden til at blanke musen.

Og en stor tak til alle de folk der har skrevet til mig med forslag til Yak, og fejl-rapporter.

Tak også til Martin Huttenloher for hans vidunderlige ikonsamling: MagicWB.

Tak til Osma Ahvenlampi, der tegnede en flot ikon til Yak i MagicWB stilen.

Tak til Nicola Salmoria, der forklarede mig princippet bag 'Sort Kant' muligheden i WB 3.0.

Og sidst, men ikke mindst, tak til Martin Scott, der opfandt Yak.

1.68 Kontakt til forfatteren

Hvis du vil i kontakt med forfatteren for at give kommentarer, forslag, fejl-rapportering, lovprisninger, penge osv., kan det ske på følgende adresse:

Gaël Marziou
Cidex 103
38920 CROLLES
FRANCE

Eller via EMAIL (foretrækkes): Gael_Marziou@grenoble.hp.com

og:

Philippe Bastiani
7 Boulevard Ricard
13003 MARSEILLE
FRANCE

Vil du i kontakt med manden bag den danske oversættelse kan dette ske på følgende adresse:

Christian Høj
Damstræde 81
9220 Aalborg Ø
Danmark

Eller via e-mail: cbh@vision.auc.dk

1.69 Mere om genveje

Denne information er fra dokumentationen til ToolManager2, og er reproduceret med tilladelse fra det programs forfatter, Stefan Becker.

Hvordan en genvej defineres

=====

Dette kapitel beskriver hvordan en genvej beskrives ved en Input Beskrivelses Streng, der bliver fortolket af Commodities. Hver gang en genvej aktiveres, genererer Commodities et Event, der bliver brugt af et commodity i kæden. En Beskrivelses Streng har følgende syntax:

```
[<klasse>] {[-][<valg-tast>]} [-][upstroke] [<TastKode>]
```

Alle nøgleord er uafhængige af store og små bogstaver.

'klasse' beskriver InputEvent klassen. Denne parameter er valgfri, og hvis den mangler benyttes standarden 'rawkey'. Se InputEvent klasser.

Valg-taster er 'signaler' der sættes eller slettes ved genvejs-aktiveringen; ellers bliver der ikke genereret noget Event. For hver valg-

tast, der skal sættes, skal man give dens nøgleord. Alle andre valg-taster forventes at blive slettet som standard. Hvis du vil ignorere en valg-tast, sættes '-' foran nøgleordet. Se Valg-taster.

Normalt bliver der genereret et genvejs-Event, når der trykkes en tast. Hvis Event'et skal genereres når tasten slippes angiver man nøgleordet 'upstroke'. Når både tryk og slip af tasten skal generere et Event bruges '-upstroke'.

TastKode er afhængig af InputEvent klassen. Se TastKoder.

Bemærkning: Vælg dine genveje omhyggeligt, fordi Commodities har høj prioritet i InpuEvent kæden (dvs. eksisterende definitioner ikke virker).

InputEvent klasser

=====

Commodities understøtter de fleste InputEvent klasser der genereres af input.device. Denne del beskriver de klasser, der er nyttigst for genveje.

'rawkey'

Dette er standard klassen og dækker alle tastatur Events. For eksempel 'rawkey a' eller 'a' genererer et Event hver gang, der trykkes 'a'. Der skal specificeres en TastKode for denne klasse. Se rawkey TastKoder.

'rawmouse'

Denne klasse beskriver alle Events fra museknapperne. Der skal specificeres en TastKode for denne klasse. Se rawmouse TastKoder.

'diskinserted'

Events fra denne klasse genereres når der indsættes en diskette i et drev. Denne klasse har ingen TastKoder.

'diskremoved'

Events fra denne klasse genereres når der fjernes en diskette fra et drev. Denne klasse har ingen TastKoder.

Valg-Taster

=====

Nogle nøgleord synonymmer blev tilføjet Commodities V38. Disse er markeret med '*'.

'lshift', 'left_shift' *
Venstre shift tast.

'rshift', 'right_shift' *
Højre shift tast.

'shift'
En af shift tasterne.

'capslock', 'caps_lock' *
Caps lock tast.

``caps'`
Enten shift tast eller caps lock tast.

``control', `ctrl' *`
Control tast.

``lalt', `left_alt' *`
Venstre alt tast.

``ralt', `right_alt' *`
Højre alt tast.

``alt'`
Én af alt tasterne.

``lcommand', `lamiga' *, `left_amiga' *, `left_command' *`
Venstre Amiga/Command tast.

``rcommand', `ramiga' *, `right_amiga' *, `right_command' *`
Højre Amiga/Command tast.

``numericpad', `numpad' *, `num_pad' *, `numeric_pad' *`
Dette nøgleord skal bruges for hvilken som helst tast på det numeriske tastatur.

``leftbutton', `lbutton' *, `left_button' *`
Venstre muse knap. Se bemærkning herunder.

``midbutton', `mbutton' *, `middlebutton' *, `middle_button' *`
Midterste muse knap. Se bemærkning herunder.

``rbutton', `rightbutton' *, `right_button' *`
Højre muse knap. Se bemærkning herunder.

``repeat'`
Denne valg-tast sættes når tastatur-gentag er aktiv. Kun brugbar for InputEvent klassen `'rawkey'`.

Bemærkning: Der er en fejl i Commodities V37, der forhindrer brug af `'leftbutton'`, `'midbutton'` and `'rbutton'` som valg-taster. Denne fejl er rettet i V38.

TastKoder
=====

Hver InputEvent klasse har sine egne TastKoder:

TastKoder for InputEvent klasse `'rawkey'`

Nogle nøgleord synonymer blev tilføjet Commodities V38. Disse er markeret med `'*'`.

``a'-`z', `0'-`9', ...`
ASCII tegn.

`f1', `f2', ..., `f10', `f11' *, `f12' *
 Funtions tasterne.

`up', `cursor_up' *, `down', `cursor_down' *
 `left', `cursor_left' *, `right', `cursor_right' *
 Cursor tasterne.

`esc', `escape' *, `backspace', `del', `help'
 `tab', `comma', `return', `space', `spacebar' *
 Special taster.

`enter', `insert' *, `delete' *
 `page_up' *, `page_down' *, `home' *, `end' *
 Taster på det numeriske tastatur. Alle disse TastKoder skal benyttes
 sammen med nøgleordet `numericpad'!

TastKoder for InputEvent klassen `rawmouse'

Disse nøgleord blev tilføjet Commodities V38. De er ikke tilgængelige i
 V37.

`mouse_leftpress'
 Tryk venstre muse knap.

`mouse_middlepress'
 Tryk midterste muse knap.

`mouse_rightpress'
 Tryk højre muse knap.

Bemærkning: For at benytte en af disse TastKoder, skal det tilsvarende
 nøgleord angives, f.eks:

rawmouse leftbutton mouse_leftpress

Eksempler på genveje

=====

`ralt t'
 Hold højre Alt tast og tryk "t"

`ralt lalt t'
 Hold venstre *og* højre Alt tast og tryk "t"

`alt t'
 Hold en af Alt tasterne og tryk "t"

`rcommand f2'
 Hold højre Amiga tast og tryk funktions tast 2

`numericpad enter'
 Tryk Enter tasten på de numeriske tastatur

```
`rawmouse midbutton leftbutton mouse_leftpress'  
    Hold midterste muse knap og tryk venstre muse knap  
  
`diskinserted'  
    Indsæt en diskette i vilkårligt drev.
```

1.70 Lokalisering

Yak er nu lokaliseret. i øjeblikket understøttes engelsk i programmet og hollandsk, fransk, tysk, italiensk, svensk og dansk i de medfølgende kataloger. Hvis du har lyst til at oversætte Yak til dit eget sprog, er du velkommen. Det er ikke nødvendigt at være udvikler for at lave en oversættelse. Det eneste du behøver er at kende dit eget sprog godt :-)

Vær opmærksom på at der er adskillige ting, der kan oversættes:

Selve Yak ved at skrive et katalog.
Installer scriptet.
AmigaGuide dokumentationen.

Du kan oversætte dette helt eller delvist, som du ønsker.

I Catalogs skuffen finder du en fil der hedder 'yak.ct', der indeholder de strenge, der bruges i Yak's brugerflade, der er bygget fra 'yak.cd', der også findes i Catalogs skuffen.

Den er klar til at blive oversat. Her er et eksempel:

```
MISCELLANEOUS_STRING
```

```
; Miscellaneous;
```

Hvis du vil oversætte 'Miscellaneous' til fransk, skriver du oversættelsen på den tomme linie som dette:

```
MISCELLANEOUS_STRING
```

```
Divers  
; Miscellaneous;
```

Når du har oversat alle strengene i 'yak.ct' kan du enten bruge catcomp eller en PD erstatning som CatEdit, KitCat eller FlexCat, til at generere katalog-filen, eller send 'yak.ct' til mig, og så genererer jeg kataloget for dig.

Der er 5 linier til credits i Yak-menuen reserveret til oversætteren, så tøv endelig ikke med at skrive dit navn og din adresse her.