



Indhold

Indledning.....	2
Anvendelse af processer.....	3
Opret proces.....	4
Processkabelon.....	6
Udvidelse af processkabelon.....	13
HTML (Hyper Text Markup Language).....	13
Panther kode.....	14
Scopes.....	18
Konto detaljer.....	19
Medlems produkt.....	20
Rykker.....	21
Variable.....	27
Faste variable til udlæsning:.....	27
Medlemsdata.....	27
Midlertidige variable til ind og udlæsning.....	29
Indlæs data.....	29
Variabel typer.....	30
Udlæs data.....	32
Udlæsning fra midlertidige variable.....	32
Udlæsning af hjælpe object data.....	33
Typer (ikke skrevet færdig).....	35
Logik (ikke skrevet færdig).....	36
HTML til skabelon sider.....	37
HTML koder.....	38



Indledning

Vejledningen beskriver generel opsætning af processer i Panther Admin. Processer er små programmer der automatisere opgaver, eller dele af opgaver. Ved at anvende processer kan opgaver systematiseres og deles på tværs af medarbejdere, faggrupper, IT-systemer, eksterne partnere mm.



Anvendelse af processer

Processer anvendes dagligt af de mennesker der arbejder i og med Panther Admin. En proces vil typisk være tilgængelig til anvendelse via en knap, der så iværksætter en serie af opgaver. Et kort eksempel kan være at opgradere et Internet abonnement til en højere hastighed. Her kan processen sættes op til automatisk at udføre opgaver som provisionering af kundens/medlemmets udstyr, sende bekræftelse med e-mail, skrive note på kunden i notesystem, sende en regning osv.

Mobiltelefon	
Telefonnummer	
E-mail (Faktura)	
EAN (Nemhandel)	
Betalingservice	
Min. fakturabeløb	0,00
Max. fakturabeløb (fakturerings afbrydes hvis beløb er større)	100000,00
Betalingsfrist i dage	14
	<input type="button" value="Flyt konto"/>
Saldo	0,00
Forfalden saldo	0,00

Processer	
<input type="button" value="Installation af internet"/>	<input type="button" value="Spørgsmål til kundeservice"/>

Figur 1 Knapper der initiere processe.

Når en proces er startet op, afslutter den sig selv når alle opgaver den indeholder er udført.



Opret proces

Simpel oprettelse af proces, med information om hvordan processen gøres tilgængelig.

Procedure:

- 1) Klik i hovedmenuen på "Admin"
- 2) Klik i undermenuen på "Proces"
- 3) Under teksten "opret ny process" skriv da et navn: fx "Ring tilbage om Wifi"
- 4) Vælg scope "customer product". Hvilket betyder at processen kan startes ved at klikke ind på et medlem, og et af de produkter medlemmet har købt, eller er i gang med at købe.
- 5) Tryk "opret"
- 6) Herefter er processen oprettet, men indeholder ingen opgaver. Knappen er derfor en tom skald, der nu kan udfyldes med det den er tiltænkt. Klik på processen "Ring tilbage om Wifi" for at få lov at redigere/udfylde opgaver i proces skabelonen.

Proces Liste

Proces navn	Proces skridt	Aktive
Scope : account		
Installation af internet	6	2
Spørgsmål til kundeservice	3	4
Scope : customerproduct		
Ring tilbage om Wifi	0	

Opret ny process

Proces navn : Start scope :

Fil : Der er ikke valgt nogen fil

Figur 2 Processkabeloner



7) Processen er som udgangspunkt utilgængelig for brugere. I højre side under "Adgangsgrupper", vælg de grupper der skal kunne se/arbejde med processen og tryk "gem".

Figur 3 Adgang til proces

8) Herefter er processen tilgængelig. Dog stadig tom. Processen blev lagt under det scope der hedder "customer products" og er derfor tilgængelig på medlemmer under produkter.

Figur 4 Proces er tilgængelig

9) For at processen skal give mening skal der fyldes opgaver ind i proces skabelonen. Se næste afsnit om processkabelon.

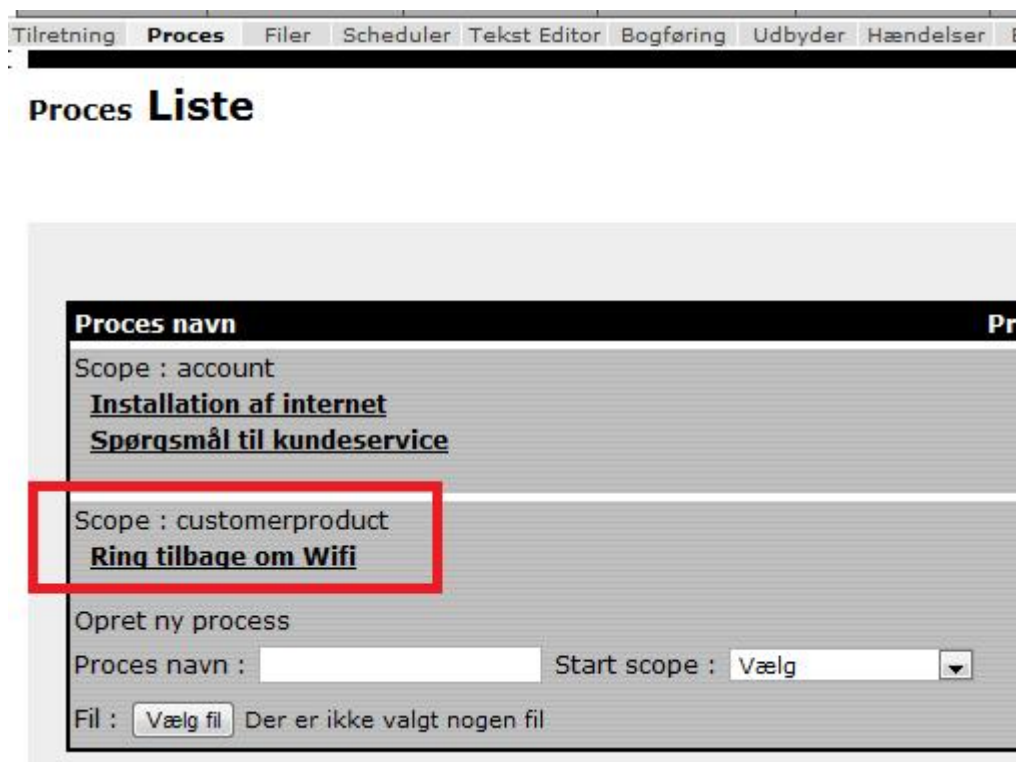


Processkabelon

Her vises hvordan en processkabelon konfigureres med 2 opgaver og resultatet heraf. Eksemplet er tænkt at skulle give en kortfattet gennemgang så man forstår selve flowet. At opbygge processer er en kreativ proces og her gælder det om at udforme og opsætte egne ideer. Det kræver typisk at man bruger lidt tid på det.

Processkabeloner defineres i serier af opgaver, hvor nogle venter på hinanden og andre ikke. De kan løbende oprettes og tilpasses/redigeres i forhold til behov.

- 1) Klik i hovedmenuen på "Admin"
- 2) Klik i undermenuen på "Proces"
- 3) Klik på processkabelonen "Ring tilbage om Wifi"

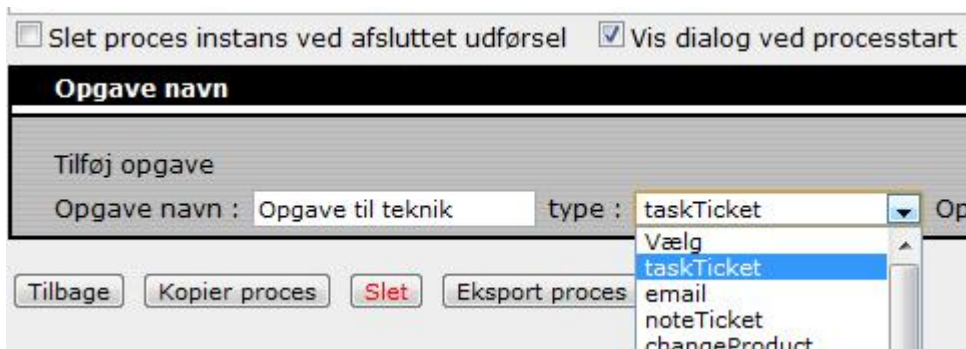


Figur 5 Processkabelon

4) I dette tilfælde åbnes en tom proces, dvs. der er kun mulighed for at indsætte nye proces-trin som kaldes opgaver. Her kan første trin oprettes ved i "opgave navn" at skrive "opgave til teknik", og herefter under

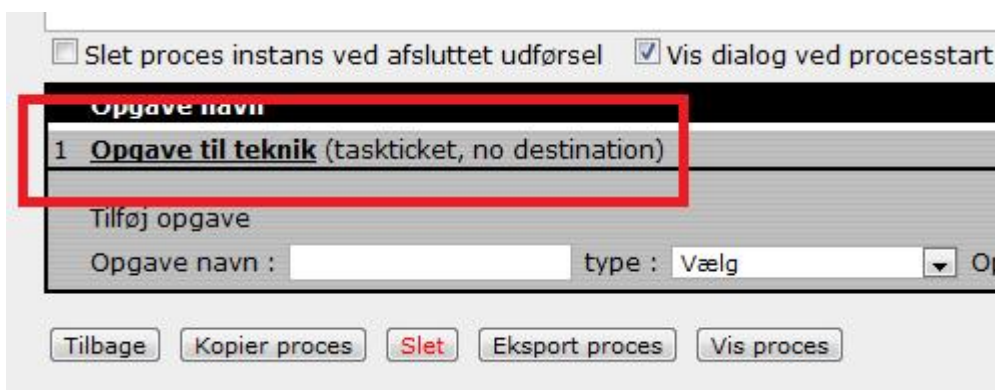


"type", vælg "task-ticket". Første opgave fra systemet bliver derfor at oprette en opgave til en medarbejder der kan hjælpe medlemmet.



Figur 6 Opgave oprettelse af arbejdsopgave

5) Tryk "opret" for at oprette processtrinnet.



Figur 7 Rediger opgave

6) Klik på opgaven for at redigere i den.

7) Udfyld herefter felterne. Bemærk at der ved udfyldning indsættes en række variabel. Variabelnavne bruges fordi der arbejdes i skabeloner.



Task ID : 16

Opgave navn :

Responsible user :

Responsible usergroup (Choose either user OR usergroup) :

Subject :

Content :

```
{.customer::getname.} vil gerne ringes o
kundenavn: {.customer::getname.}
kundernummer: {.customer::getid.}
telefonnummer: {.customer::getmobile.}
```

Figur 8 Variable i opgave

- 8) I opgave skabelonen indsættes i dette tilfælde en "responsible user". Responsible user er den bruger der skal modtage den konkrete opgave.
- 9) Subjekt udfyldes som i en e-mail.
- 10) I body indsættes den tekst der ønskes sendt. I dette tilfælde bruges **variablen** `{.customer::getname.}` til at hente kundens navn ud. Herefter der i Content feltet beskrives hvad der skal gøres.
- 11) Herefter skrives flere oplysninger om medlemmet, herunder navn, nummer og et telefonnummer der hentes ud.
- 12) Tryk "gem" for at afslutte og herefter "tilbage".
- 13) Herefter indsættes en opgave mere. Denne opgave skriver i medlemmets note at medlemmet har henvendt sig og det er aftalt at en tekniker ringer tilbage.
- 14) Indtast i Opgave navn: "Note" og under type vælg "Note ticket" og tryk opret.

Opgave navn

1 **Opgave til teknik** (taskTicket, user : **Casper Larsen** subject: Ring til medlem ang. Wifi pro

Tilføj opgave

Opgave navn : type : Opgave trin :

Vælg
taskTicket
email
noteTicket
changeProduct

Figur 9 Oprettelse af opgave noteticket

- 15) Klik på den nye opgave for at redigere.



Task ID : 14

Opgave navn : Note

Kundeobjekt (customer) :

Installationsadresseobjekt :

TicketID (blank for nyt) :

Afsender brugerID :

Subject : Problemer med Wifi

Content : Vi har dd. aftalt at vi ringer tilbage

Figur 10 redigering af noteticket

16) Indtast "Subjekt" - overskrift der beskriver det der er sket og noget tekst der også beskriver det, så det vil blive husket fremover.

17) Tryk "gem", og "tilbage".

18) Nu er første proces færdig. Den indeholder 2 opgaver, hvor første opgave sender en arbejdsopgave til en tekniker, og herefter notere information om denne på medlemmets noteblad. Opgaverne afvikles automatisk med det samme.

Slet proces instans ved afsluttet udførelse Vis dialog ved processtart

Opgave navn

1 **Opgave til teknik** (taskTicket, user : Casper Larsen subject: Ring til medlem ang. Wifi prob

2 **Note** (noteTicket)

Tilføj opgave

Opgave navn : type : Vælg Opgave trin : 3

Tilbage Kopier proces Slet Eksport proces Vis proces

Figur 11 Liste af opgaver i proces

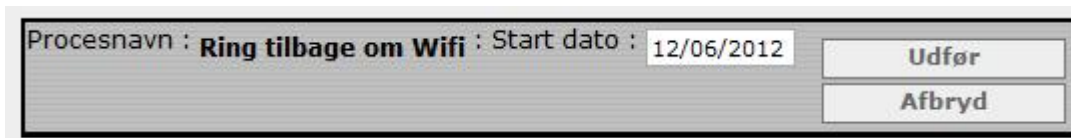
Resultatet kan ses i hhv. opgaver (Overfane Medlemmer, underfane Opgaver) og i Note på medlemmet.

19) For at starte processen, gå da til produkt på et medlem og klik på knappen "Ring tilbage om wifi".



Figur 12 Opstart af proces

Herefter popper en menu op:



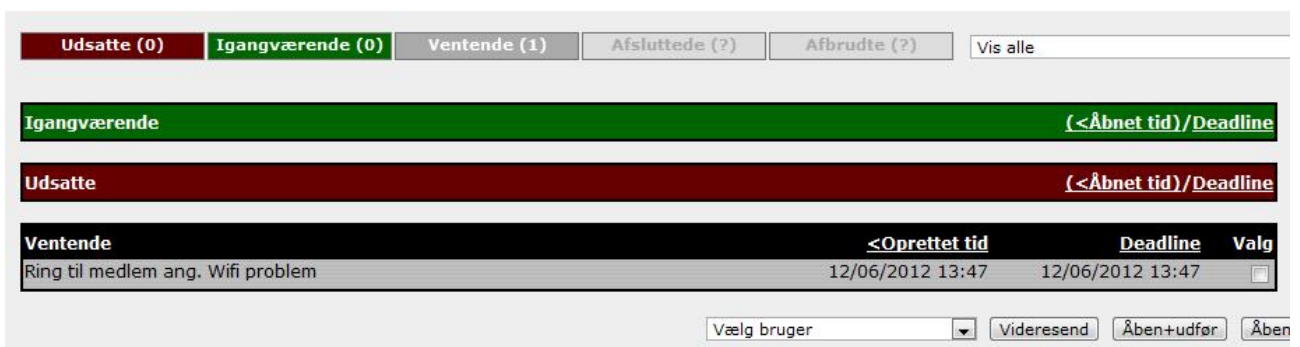
Figur 13 Opstart af proces på dato

20) Resultater: Opgave, en opgave til bruger "Casper"



Figur 14 Ventende opgaver

21) Ventende opgaver: Klik på overfanebladet Medlemmer og herefter Opgaver.



Figur 15 Åbning af opgave



22) Åben opgaven

Quicklinks Liste **Opgaver** Udtræk Konto admin Beskeder Adresser

Medlemmer Opgaver

Filter

Opgave åbnet

Udsatte (0) **Igangværende (1)** Ventende (0) Afsluttede (?) Afbrudte (?) Vis alle

Igangværende (<Åbnet tid)/Dea

Ring til medlem ang. Wifi problem

Eskild Keller vil gerne ringes op angående wifi problem. kundenavn: Eskild Keller kundenummer: 51764 telefonnummer: 23303400

12/06/2012 1
(12/06/2012 13

- Udfør
- Gem
- Afbryd**
- Udsæt
- Videresend
- Fortryd åbni

Udsatte (<Åbnet tid)/Dea

Ventende <Oprettet tid **Deadline**

Vælg bruger Videresend Åben+udfør

Figur 16 Arbejds på opgaven og luk

23) Udfør opgaven

24) Noter:



Tilføj ny note til medlem

12/06/2012 13:51, Casper Larsen (casper)

Der er ikke valgt nogen fil

ID / Dato	Emne
220 12/06/2012 13:51	Proces robot, Vi har dd. aftalt at vi ringer tilbage angående wifi problemet.

Figur 17 Tekst i notesystem



Udvidelse af processkabelon

Når processer har været anvendt i en periode, er der typisk opstået nye ideer til effektivisering. Her er der så mulighed for at åbne processer igen og ændre/tilføje opgaver.

Udvidelsen i dette eksempel går ud på at registrere informationen fra den medarbejder der først har talt med medlemmet. Her vil vi gerne have at medarbejderen registrerer hvorvidt kundens kabel forbindelse virker og om der er yderligere kommentarer der kan bruges inden den tekniske support tager over.

- 1) Klik i hovedmenuen på "Admin"
- 2) Klik i undermenuen på "Proces"
- 3) Klik på processkabelonen "Ring tilbage om Wifi"
- 5) Indtast i øverste felter der skal kunne anvendes ved indtastning:

Proces navn : Ring tilbage om Wifi Start scope : customerproduct

Scope indhold : customerproduct,productconf,customer,instaddress,account,customergroup,provider,portsetup,s

```

<b>INFO:</b><br>
<table>
<tr><td>Virker medlemmets kabelforbindelse</td><td>
[.kabel;select|ja:ja,nej:nej,ved ikke:ved ikke;;Mandatory;.]</td></tr>
<tr><td>Note:</td><td>[.notetekst;textarea;;;.]</td></tr>
</tr>
    
```

Slet proces instans ved afsluttet udførsel Vis dialog ved processtart

Opgave navn
1 Note (noteTicket)
2 Opgave til teknik (taskTicket, user : Casper Larsen subject: Ring til medlem ang. Wifi problem)

Figur 18 Indtastningsformular

Forklaring til skærbillede

HTML (Hyper Text Markup Language)

INFO 	skriver INFO med fed skrift

	linieskift
<table>	skemaform start
<tr>	indsætter første række
<td>	indsætter første kolonne i rækken
</td>	afslutter første kolonne i rækken
</tr>	afslutter første række
</table>	afslutter skemaform



Panther kode

```
[.kabel;select|ja:ja, nej:nej, ved ikke: ved ikke;; Mandatory;.]
```

[.kabel; opretter en variable der hedder kabel

select| opretter et rullegardin med valgmuligheder

ja:ja, nej:nej;; i dette tilfælde skal man kunne vælge ja og nej, og variabelen skal indeholde ja og nej.

Mandatory;.] betyder at dette felt SKAL udfyldes for at komme videre. Det er ikke valgfrit.

```
[.notetekst;textarea;;;.]
```

[.notetekst; opretter en variabel der hedder notetekst

textarea;;;.] opretter et indtastningsfelt.

(mere om variable i afsnit om variable)

7) Det er nu forventningen af personen der opretter opgaven indtaster de relevante informationer når opgaven initieres. Herefter vil vi gerne have systemet til at gemme de indtastede informationer i Notefeltet på medlemmet og i den opgavebeskrivelse der er sendt videre til teknik. Rediger herefter først "Note" opgaven ved at klikke på den.

Task ID : 17	
Opgave navn :	Note
Kundeobjekt (customer) :	
Installationsadresseobjekt :	
TicketID (blank for nyt) :	
Afsender brugerID :	
Notetype :	
Subject :	Problemer med wifi
Content :	Vi har dd. aftalt at vi ringer tilbage angående wifi problemet. Virker netværket med kabel: {.c_kabel.} Note: {.c_notetekst.}
Subject :	Problemer med wifi

Figur 19 Indtastning boks med Panther kode og HTML

Her indtastes følgende, som skal gemmes på kundens note fane.



{.c_kabel.} henter informationen der er tastet ind i variabelen "kabel".

{.c_notetekst.} Henter informationen der er tastet ind i variabelen notetekst.

8) Tryk gem og tilbage.

9) Sidste skridt er at sørge for at den information der indtastes også kommer helt ud til den tekniker der har fået opgaven. Herefter redigeres "Opgave til teknik" og variable indsættes så disse også vises på skærmen.

Teknikker billedet kan virke lidt forvirrende når alt står i en lang linie. Her anvendes koden
 til lineskift så præsentationen bliver mere overskuelig.

Task ID : 16	
Opgave navn :	Opgave til teknik
Responsible user :	casper
Responsible usergroup (Choose either user OR usergroup) :	None
Subject :	Ring til medlem ang. Wifi problem
Content :	<pre> {.customer::getname.} vil gerne ringes op angående wifi problem.

 kundenavn: {.customer::getname.}
 kudenummer: {.customer::getid.}
 telefonnummer: {.customer::getmobile.}

 Virker netværket med kabel: {.c_kabel.}

 Note: {.c_notetekst.} </pre>

Figur 20 Proces med Panther kode, variable og HTML

{.c_kabel.} henter informationen der er tastet ind i variabelen "kabel".

{.c_notetekst.} Henter informationen der er tastet ind i variabelen notetekst.



10) Tryk gem. I fremtiden vil processen vise følgende ved initiering:

Procesnavn : **Ring tilbage om Wifi** : Start dato : 13/06/2012

INFO:
 Virker medlemmets kabelforbindelse har brug for at få justeret

Note:

10.127.127.1:e448.c77d.ddda

Figur 21 Resultat af Panther kode og HTML

11) Noten vil vise følgende:

Vælg fil Der er ikke valgt nogen fil

ID / Dato	Emne
234 13/06/2012 11:15	Casper Larsen, Vi har dd. aftalt at vi ringer tilbage angående wifi problemet Virker netværket med kabel: ja Note: medlemmet har brug for at få justeret kanaler

Figur 22 Resultat af Panther kode i Notefelt

12) Arbejdsopgaven til teknikkeren vil se ud som følgende:



Udsatte (0) **Igangværende (1)** Ventende (0) Afsluttet

Igangværende

Ring til medlem ang. Wifi problem

Eskild Keller vil gerne ringes op angående wifi problem.

kundenavn: Eskild Keller
kundenummer: 51764
telefonnummer: 23303400

Virker netværket med kabel: ja

Note: medlemmet har brug for at få justeret kanaler

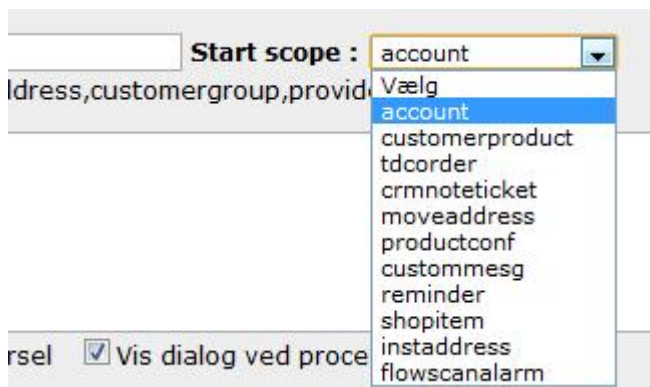
Udsatte

Figur 23 Arbejdsopgave til medarbejder, dannet af proces.



Scopes

Scopes definerer de steder i systemet hvor en proces kan initieres og hvorfra der kan hentes data. En proces har altid et udgangspunkt. Udgangspunktet er bestemmende for hvilke data der overføres til processen ved opstart. En proces der opstartes fra fanebladet medlemsdetaljer, vil kunne trække medlemmets detalje informationer med sig videre. Hvis processen startes fra et medlems produkt-specifikation, vil processen kunne trække medlemmets produktdetaljer med sig.



Figur 24 Scope vælger ved ny proces.

Processer kan startes fra:

- medlemsdetaljefanen - account
- kundeprodukt - customerproduct
- notefanen - crmnoteticket
- eksternt kald til SOAP API - custommesg
- rykkerprocedure

I det næste vil det der blive præsenteret eksempler på proces der startes inde fra Panther Admin systemet.



Konto detaljer

Processer der skal kunne startes direkte fra medlemsfanebladet skal oprettes i det scope der hedder "account". Processer af den slags er typisk processer der er relateret til noget generelt på medlemmet eller medlemmets adresse. Eksempler på dette kan være relateret til:

- Spørgsmål / Dialog
- Installationer
- Betalinger
- Flytning

Når processerne er oprettet vil de lægge sig under knapperne på medlemmets konto fane.

Betalingservice	
Min. fakturabeløb	0,00
Max. fakturabeløb (fakturering afbrydes hvis beløb er større)	100000,00
Betalingsfrist i dage	14
	<input type="button" value="Flyt konto"/>
Saldo	0,00
Forfalden saldo	0,00

Processer	
<input type="button" value="Installation af internet"/>	<input type="button" value="Spørgsmål til kundeservice"/>

Figur 25 Proces knapper på konto detaljer



Medlems produkt

Processer der skal startes på de produkter medlemmer allerede har, eller er i gang med at købe skal lægges ind i det scope der hedder "customerproduct". Processer der startes herfra er typisk meget produkt specifikke. Typiske processer under scope "customerproduct" er:

- spørgsmål
- opgradering / nedgradering / lukning
- fejlhåndtering specifikt relateret til et produkt
- bestilling af teknikker

The screenshot shows a web interface for 'ORUM - Eskild Keller, 484'. At the top, there are tabs for 'primary' and 'Alle konti', and a 'Produktgruppe' dropdown menu with a 'Tilføj produkt' button. Below this is a table titled 'Produkter' with columns for 'Produkt', 'Antal', and 'Næste faktura periode'. The table lists two products: 'Høj 100d 50u' and 'Høj mta off', both with an amount of 1 and status 'lukket'. Below the table are buttons for 'Kør afregning' and 'Rabat'. At the bottom, there is a 'Rediger Høj 100d 50u' section with a 'Processer' button highlighted by a red box. Other elements include 'Sidste ændring (root) 23/05/2012 10:09' and a 'Ring tilbage om Wifi' button.

Figur 26 Proces knap på medlems produkt blad

Læs mere om processer der startes under medlemsprodukt ved at læse afsnittet om processkabeloner i dette dokument.



Rykker

I Panther Admin sættes rykkerhåndtering op i processer. Rykker processer kan indeholde et eller flere rykkertrin, der hver især starter en proces. I det efterfølgende vises hvordan en simpel rykkerproces kan være sat op i Panther Admin.

- 1) Klik i hovedmenuen på "Admin"
- 2) Klik i undermenuen på "Rykkerprocedurer"



Figur 27 Rykker procedure opsætning

3) I listen ovenfor er der vist et system som kun har en rykkerprocedure, hvilket er helt normalt. Hvis en forening laver regnskab for andre foreninger kan der være forskellige rykkerprocedure. I den viste rykkerprocedure er der 3 trin.

4) Klik på Rykkerproceduren "Standard", for at få vist den rykker proces der normal anvendes i Panther admin.

Trin	Tekst	Vent	Betalingsfrist	Gebyr	produkt	Fakt.	Proces	Skabelon
1	Rykker 1	5	8		Ingen	<input type="checkbox"/>	Ingen	reminder_orum1.php [edit]
2	Rykker 2	5	8		Ingen	<input type="checkbox"/>	Ingen	reminder_orum1.php [edit]
3	Opgave ang. afbrydelse	5	8		Ingen	<input type="checkbox"/>	Afslut rykning Ingen Afslut rykning	Ingen [edit]

sidste Tilføj ny Gem

Figur 28 Opsætning af rykkertrin

5) Efter klik på "Standard" foldes rykker processen ud. I venstre side kan de ses at proceduren sættes op i 3 trin. Der kan tilføjes nye trin.

- Trin** Det trin i rækkefølgen den enkelte linie beskriver.
- Tekst** Den tekst der overføres til proces eller skabelon.
- Vent** Antal dage før rykkerproceduren hopper videre, efter betalingsfrist.
- Betalingsfrist** Antal dages betalingsfrist for en rykker.
- Fakt** Om rykkeren skal skrives ud på næstkommende faktura.



Proces Den proces der skal initieres hvis det ønskes at en proces afvikles på det givne rykkertrin.

Skabelon Den skabelon der udskriver rykkeren i PDF. Denne er skrevet i php og kan redigeres.

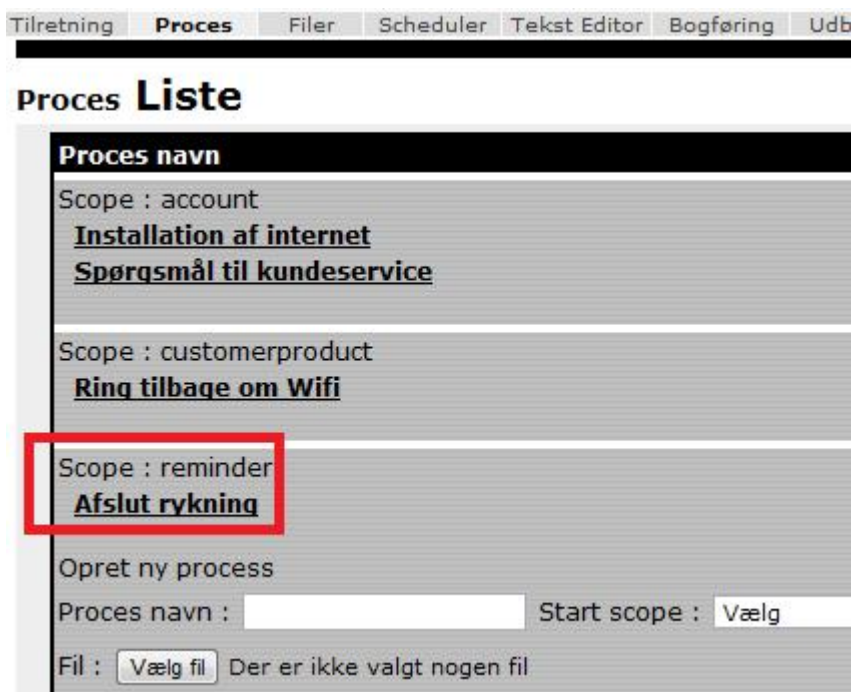
6) I ovenstående eksempel er der ikke indført automatik på trin 1 og 2. Trin 1 og 2 giver blot mulighed for at printe rykkere ud. Trin 3 starter her en proces.

Der er kun oprettet en proces under scope for rykkere og det er navngivet "Afslut rykning".

7) Klik i hovedmenuen på "Admin"

8) Klik i undermenuen på "Proces"

9) Under Scope: "Reminder" ligger processen "Afslut rykning".



Figur 29 Rykker scope

10) Klik på proces skabelonen "Afslut rykning".



Slet proces instans ved afsluttet udførelse Vis dialog ved proces afslutning

Opgave navn	
1	Rykning afsluttet (noteTicket)
2	Afslut kundeforhold (taskTicket, user : Eskild Keller subject: A)

Tilføj opgave
 Opgave navn : type :

Tilbage Kopier proces Slet Eksport proces Vis proces

Figur 30 Rykker proces, der aktives fra rykker trin.

11) Processen skriver en note i notesystemet om at medlemmet har modtaget 2 rykkere, der ikke er betalt og at medlemsforholdet derfor afsluttes.

Task ID : 20

Opgave navn :

Kundeobjekt (customer) :

Installationsadresseobjekt :

TicketID (blank for nyt) :

Afsender brugerID :

Notetype :

Subject :

Content :

Figur 31 Rykker opgave opsætning

12) Herefter sender processen en (tastticket) arbejdsopgave til en medarbejder, og beder om at slutte medlemsforholdet.



Task ID : 19	
Opgave navn :	Afslut kundeforhold
Responsible user :	Eskild
Responsible usergroup (Choose either user OR usergroup) :	None
Subject :	Afslut kundeforhold {,account::getnumber,}
Content :	Kunde har ikke betalt regninger link: {,account::getnumber,}

Figur 32 Rykker opgave opsætning

Responsible user	Den bruger der skal modtage arbejdsopgaven
Responsible usergroup	Den brugergruppe der skal modtage arbejdsopgaven
Subject	Arbejdsopgavens navn
Content	Indhold i arbejdsopgaven. I dette tilfælde er der i arbejdsopgaven et direkte link til medlemmets detalje blad.

13) Når et medlem når trin 3, bliver der i dette tilfælde oprettet en opgave til en medarbejder / medarbejdergruppe der kan tage kontakt til medlemmet, eller lukke medlemmet ned. Samtidigt vil der blive skrevet et notat om dette, så medlemmet og medarbejdere på et senere tidspunkt kan se hvad der er sket.

14) Har man flere forskellige rykker processer hvor nogle processer skal køres på nogle grupper medlemmer og andre på individuelle medlemmer, kan dette sættes op på hhv. grupper og medlemmers konti.

15) For individuel opsætning på et medlem: Medlemmets detaljeblad, og valg på konto.



primary ny konto Aktive

Konto indstillinger

Sidste ændring **(Eskild Keller) 18/01/2012 22:34**

Fakturakonto Nej

Kontonavn

Primær konto

Første fakturacyklus dag (?) 01/01/2012

Login

Kodeord

Adgang til shop

Tillad kort-betaling i shop

Moms % 25% VAT (DK)

Debitor gruppe (PBS) 0

Rykkeropsætning Standard

Næste rykkertrin

Figur 33 Individuel rykkeropsætning på medlem

16) Ved opsætning af rykker-proces for medlemsgruppe

Grupper Medlemmer Produkter Overvågnin

Grupper Gruppe adgang

Medlemmer Grupper

Gruppe	Konto	Rabat	Medlemmer
>Alle >	[rediger]	[tilføj]	0
ORUM >	[tilføj]	[tilføj]	527

Figur 34 Opsætning af rykker procedure på medlemsgruppe

17) På konto indstillinger for gruppen, tryk rediger. Under rykkeropsætning vælg den rykker opsætning der skal anvendes.



Konto indstillinger	
Moms%	25% VAT (DK) ▼
Adgang til shop	<input checked="" type="checkbox"/>
Debitor gruppe	<input checked="" type="checkbox"/>
Debitorgruppe	0
Første fakturacyklus dag (?)	<input type="radio"/> Medlem oprettelse <input checked="" type="radio"/> Dato defineret nedenfor 01/01/2012
Betalingsfrist (i dage)	14
Min. fakturabeløb	0,00
Max. fakturabeløb (fakturerings afbrydes hvis beløb er større)	100000,00
Rykker opsætning	Standard ▼

Figur 35 Valg af Rykkerprocedure på medlemsgruppe



Variable

Diverse variable til anvendelse til ind- og udlæsning af data i databasen.

Faste variable til udlæsning:

Panther Admin bygger på en database hvor i alt information lagres. Denne information hentes og gemmes i "objecter". Der er fx et medlemsobject der indeholder de medlemsdetaljer der kan ses på medlemmets detalje side. Der er ligeledes objecter der indholder information om produkter, og en lang række mindre interessante objecter.

For at trække data ud via en proces skal man kende de objecter der skal bruges og de datavariabelnavnene der er relevante for de enkelte udtræk. Hvis målet fx er at hente et mobiltelefonnummer ud, vil man altså pege på medlemsobjectet og heri datavariablen for mobiltelefonnummer. `{.object::datavariabel.}` i dette tilfælde `{.customer::getmobile.}`

```

Subject : Ring til medlem ang. Wifi problem
Content : {.customer::getname.} vil gerne ringes o
          kundenavn: {.customer::getname.}
          kundenummer: {.customer::getid.}
          telefonnummer: {.customer::getmobile.}
    
```

Figur 36 Variable der anvendes når data skal læses ud i processer

I det efterfølgende vil vi beskrive de mest almindelige objekter og datavariabeler.

Medlemsdata

Medlemsdetaljer er lagret i objectet "customer" og har følgende variable

getid	returnere systemets medlems id (bruges til opbygning af direkte web-link"
getcustomergroupname	returnere den fulde medlemsgruppe
getnumber	returnere foreningens medlemsnummer
getname	returnere medlemmets fulde navn
getname2	returnere medlemmets navn, i navne felt 2
getcompany	returnere hvis virksomhed, så firmanavn
getadrstreet	returnere gade/vejnavn
getadrnumber	returnere gade/vejnummer
getadrfloor	returnere sal
getadrside	returnere side (højre, venstre, midt for etc)
getaddress	returnere adressefelt 1 (komplet med vejnavn, nummer, sal, side)
getaddress2	returnere adressefelt 2 (komplet med vejnavn, nummer, sal, side)
getzipcity	returnere postnummer og by
getzip	returnere postnummer
getcity	returnere by
getcountry	returnere land
getemail	returnere e-mail adresse
getphone	returnere telefonnummer
getmobile	returnere mobiltelefonnummer
getfax	returnere fax nummer
getcontactperson	returnere kontaktperson
getmodifiedstr	returnere senest opdateret tidspunkt
getcreatedstr	returnere oprettet gade tidspunkt

Processer



getuserlogin returnere brugerloginnavn til selvbetjening
getuserpassword returnere kodeord til selvbetjening



Midlertidige variable til ind og udlæsning

Ved opstart af proces, er der muligheder for at indsamle / indtaste information, der ikke på forhånd er given. Den information kan være relateret til det arbejde der senere skal udføres, eller alternativt at gemme information omkring den handling der skal foretages.

En proces kan ved opstart kræve svar på en række spørgsmål, disse svar indlæses i de midlertidige variable og kan herefter overføres til de næste proces trin. Indhold af midlertidige variable kan være spørgsmål direkte til medlemmet som "årsag til skifte af kanalpakke", hvoraf valg kan være "skifte udbyder", "økonomi" og "andre årsager". Ved skifte af udbyder kan der så vælges hvilke udbyder medlemmet har valgt og alle disse information kan automatisk videreføres til medlemmets notefelt, inden processen fortsætter og nedgradere / lukke abonnementet eller hvad der nu må være relevant i den givne proces.

De variable der sættes op i en proces, vil kunne indeholde data så længe processen køres, og overføres vises, ændres mm. mellem de forskellige procestrin. Når processen afsluttes slettes variabel information sammen med processen.

Grupper	Medlemmer	Produkter	Overvågning	Mediering	Provisionering	Billing	Admin	Log		
Tilretning	Proces	Filer	Schedulere	Tekst Editor	Bogføring	Udbyder	Hændelser	Brugere	Rykkerprocedurer	ESG

Proces Rediger

Proces navn : Ring tilbage om Wifi Start scope : customerproduct

Scope indhold : customerproduct,productconf,customer,instaddress,account,customerproduct,provider,portsetup,switch

```

<b>INFO:</b><br>
<table>
<tr><td>Virker medlemmets kabelforbindelse</td><td>
[.kabel;select|ja:ja,nej:nej,ved ikke:ved ikke;;Mandatory;.]</td></tr>
<tr><td>Note:</td><td>[.notetekst;textarea;;;.]</td></tr>
</tr>
    
```

Adgangsgrupper:

- administrators
- antenne
- antenne/bestyrelse
- antenne/bogholderi
- antenne/installator
- antenne/kundeservice
- bogholderi
- kundeservice
- superusers
- teknik

Slet proces instans ved afsluttet udførsel Vis dialog ved processtart

Opgave navn	Ændre	Aktive
1 Note (noteTicket)	Ned fiern	
2 Opgave til teknik (taskTicket, user : Ole subject: Ring til medlem ang. Wifi problem)	Op fiern	

Figur 37 eksempel på ikke fast variabel, select boks

Indlæs data

Et input felt defineres ved at bruge firkant-parenteser [].

[.variablename;type;;mandatory;.]

variablename navnet på den variabel der input værdien skal gemmes i. Skal være et navn bestående af bogstaverne a-z og 0-9. Navnet må ikke indeholde mellemrum eller særlige tegn.

[.variablename;**type**;;mandatory;.]

type felt-typen der skal være input felt. (se variabeltyper nedenfor).



[.variablename;type;;**mandatory**;.]

mandatory mandatory betyder at feltet SKAL være udfyldt. Formularen vil brokke sig hvis feltet er mandatory og brugeren ikke har udfyldt det.

Variabel typer

De mest almindelige variabel input felter er af typerne

- text
- textarea
- select
- radio
- checkbox
- textdate

text

text opretter et tekst felt hvor i man kan skrive op til 256 tegn.



Figur 38 text boks

kode: [.notetekst;text;;;.]

textarea

textarea er til de situationer hvor der er behov for at skrive mere end en linie tekst. Her vil man så få præsenteret et tekstfelt med flere linier.

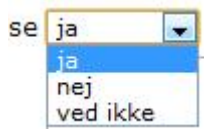


Figur 39 textarea med flere linier

kode: [.notetekst;textarea;;;.]

select

select er til de situationer hvor man på forhånd har defineret en række valg.



Figur 40 select boks med mulighed for valgene: ja, nej, ved ikke

kode: `[.kabel;select|ja:ja,nej:nej,ved ikke:ved ikke;;Mandatory;.]`

radio

Radio knapper er der hvor man skal vælge en ud af flere, på en simpel måde. Samme funktionalitet kan oprettes med en select boks.



Figur 41 select boks med valg af farve

kode: `[.farve;radio|gul:gul,sort:sort, hvid:hvid;;Mandatory;.]`

checkbox

checkboxboksen er når man har en række informationer man gerne vil klikke af:



Figur 42 Check bokse til valg af produkter

kode:

```
Medlemmet har:
Internet [.internet;checkbox|yes;;;.]
TV [.tv;checkbox|yes;;;.]
Telefoni [.telefoni;checkbox|yes;;;.]
```

Ved at tilføje "yes" som markeret med fed ovenfor, vil fluebenet ud fra den checkbox være sat på forhånd.

text-date

text-date fremkalder en kalender, hvori man med musen kan klikke en dato af, og den dato bliver så skrevet ned i variabelen til efterfølgende anvendelse.



Figur 43 text-date dato felt med kalender pop-up

kode Dato [.dato;text-date;;;.]

Andre variabel typer

Der er flere forskellige variabel typer, og der kommer hele tiden flere til, så denne vejledning er ikke fuldstændig. Af andre variabel typer kan pt. nævnes:

- fileupload
- object
- objecttext
- timeoffset

Udlæs data

Når en variabel er udfyldt, og skal anvendes i en eller flere af de efterfølgende opgaver i en proces, kan disse hentes/skrives ud som systemvariable, blot ved at tilføje et "c_" til det variabelnavn der er valgt i processkabelonen.

Udlæsning fra midlertidige variable

Skal vi hente datoen der blev sat i figur Figur 43 text-date dato felt med kalender pop-up så gøres dette ved at skrive

kode {c_dato.}

eller data ud fra Figur 42 Check bokse til valg af produkter

kode {c_internet.}, {c_tv.}, {c_telefoni.}



Udlæsning af hjælpe object data

Når en proces kører, kan det være nødvendigt at trække informationer ind fra diverse rækker i databasen, tidspunkter fra systemet mm. Til det er det muligt at bruge hjælpe objekter, der indeholder informationen.

Nedenfor er de mest almindelige hjælpe objekter beskrevet:

- * getDate (returns current date in timestamp)
- * getDatePlusDaysStr(stamp, days) (returns date in dd/mm/YYYY plus days given by "days")

example:

{.helpers::getDatePlusDaysStr(helpers::getDate, 8).} returns current date plus 8 days.

- * getCurrentDateStr (returns current date in dd/mm/YYYY)
- * getCurrentTimeStr (returns current datetime in dd/MM/YYYY HH:mm)
- * getDayofMonth(helpers::getDate)
- * getMonthofYear(helpers::getDate)
- * getDayofYear(helpers::getDate)

crmnoticket fields

- * getFileURL (return url of attached file, blank if none)
- * getsubject (returns subject of notefield)
- * getcontent (returns body of notefield)
- * getcontact (returns contact field of notefield)
- * getnotetype (returns type of notefield)
- * getuserlogin (returns userlogin of user who added the ticket)

customerproduct fields

- * phonenumber (returns phonenumber, if phone product)
- * connidvalue (returns phonenumber or switch:MAC address, if internet product)
- * connid (returns full connection id)
- * startdatestr (Start time for product)
- * expiresdatestr (Expires time for product)
- * tempclosed (Returns true if product is temporary closed)
- * nextbillstr (Returns nextbill time)
- * quantity (Returns quantity)
- * active (returns true if product is active, else false)
- * Options (returns options for product)

productconf fields

- * getname (returns name for the product)
- * getident (returns identifier for the product)
- * getparameter (returns parameters for the product)

account fields

- * getstartdatestr (Cyclus day)
- * getenableshop (returns 1 if shop is enabled)
- * getpayvisa (returns 1 if payment by cards is allowed)
- * getnumber (returns account number)
- * getcompany (returns company)
- * getcontactperson (returns account holder name)
- * getcontactperson2 (returns additional account holder name)
- * getaddress (returns address on account holder)
- * getzipcity (returns zip and city on account holder)
- * getpbsdeal (returns PBS number if available)
- * getcreditperiod (returns credit period in days)
- * getdebt (returns debt if available)
- * getdebt due (returns debt due if available)



customergroup

- * getname (returns group name)
- * getnote (returns group note)

provider

- * getname
- * getemail

portsetup

- * getubr (returnsubr number)
- * getmodemip (return modem ip)
- * getswitch (returns switch ip)
- * getswitchport (returns switch port (MAC address for cable modem))
- * getcablein (returns cable in (DSL))
- * getcableout (returns cable out (DSL))



Typer (ikke skrevet færdig)

Hvad kan de forskellige procestrin.

The screenshot shows a window titled "Tilføj opgave" (Add task). It has a text input field for "Opgave navn" (Task name) and a dropdown menu for "type" (type). The dropdown menu is open, showing a list of task types. Below the input fields are four buttons: "Tilbage" (Back), "Kopier proces" (Copy process), "Slet" (Delete), and "Eksport proces" (Export process). The list of task types includes:

- Vælg
- taskTicket
- email
- noteTicket
- changeProduct
- pdf
- addProduct
- upgradeProduct
- sms
- startProcesses
- startProcess
- jumpProcess
- consignorLabel
- emoslet
- TDC SPWeb
- Wait
- findProduct
- addCustomer
- changeCustomer
- changeAccount
- addCustomer
- changeCustomer
- changeAccount
- moveAddress
- TDC Columbine
- billing
- findAccount
- changeTDCOrder
- setDiscount
- Intrum Justitia
- changeInstaddress
- statustext
- reload
- set variables
- TDC Columbine new
- smssend
- smsreceive
- transactions
- messagecustomer
- flowscanAlarm

Beskrivelse af de vigtigste ...



Logik (ikke skrevet færdig)

Når der oprettes processer kan i der i mange tilfælde være behov for at indføre logik. Logik er at en automatiseret proces udfører opgaver afhængigt af andre. Hvis fx der ved indtastning er klikket ja ved at en bruger har Internet, så sker der noget andet end hvis der ikke er indtastet noget. Et andet eksempel er hvad der sker hvis en proces ikke når at bliver udført, inden en foruddefineret deadline.

Vent på at ...

Hvis X, så Y, ellers Z



HTML til skabelon sider

Når man arbejder med processer og informationer der manuelt skal ud og ind via processer, kan det hurtigt blive rodet at arbejde med rå-data. Det kan derfor være en hjælp at sætte sine brugergrænseflader op med HTML.

Indtastningsside **uden** HTML

Figur 44 Uden HTML kan man ikke styre hvordan tekstfelter placeres

kode:

INFO:

Virker medlemmets kabelforbindelse [.kabel;select|ja:ja,nej:nej,ved ikke:ved ikke;;Mandatory;.]
 Note:[.notetekst;textarea;;;.]

Indtastningsside **med** HTML

Figur 45 Med HTML kan man styre hvordan tekstfelter er placeret

kode:

```
<b>INFO:</b><br>
<table>
<tr>
    <td>Virker medlemmets kabelforbindelse</td>
    <td>[.kabel;select|ja:ja,nej:nej,ved ikke:ved ikke;;Mandatory;.]</td></tr>
<tr>
    <td>Note:</td><td>[.notetekst;textarea;;;.]</td></tr>
</table>
```



HTML koder

Der findes en lang række HTML koder. I Panther admin behøver man kun at anvende nogle få koder, for at opnå hvad der er behov for i relation til layout. Traditionelt set skal man skrive en masse HTML for at komme igang, herunder koder som <HTML><HEAD><BODY> osv. men det skal man ikke i Panther admin. Her skrives kun de koder man normalt vil skrive i sin <BODY> kode.

Vigtige koder

<code> tekst </code>	den tekst der skrives mellem en start <code></code> og slut <code></code> bliver skrevet med bold - fed
<code>
</code>	linie skift
<code><table> </table></code>	det der sættes ind mellem en tabel start <code><table></code> og tabel slut <code></table></code> vil blive sat op i en tabel skabelon
<code><tr> </tr></code>	definere række start <code><tr></code> og række slut <code></tr></code>
<code><td> </td></code>	definere kolonne start og kolonne slut

Forklaring af eksempel fra figur *Figur 45 Med HTML kan man styre hvordan tekstfelter er placeret*

<code>INFO:
</code>	skriver INFO med fed, og laver et lineskift
<code><table></code>	starter en tabel
<code><tr></code>	opretter en række
<code><td>Virker medlemmets kabelforbindelse</td></code>	opretter og afslutter en kolonne
<code><td>[.kabel;select ja:ja,nej:nej,ved ikke:ved ikke;;Mandatory;.]</td></code>	
<code></tr></code>	afslutter senest oprettede række
<code><tr></code>	opretter en række
<code><td>Note:</td></code>	opretter og afslutter en kolonne
<code><td>[.notetekst;textarea;;;.]</td></code>	opretter og afslutter en kolonne
<code></tr></code>	afslutter senest oprettede række
<code></table></code>	afslutter tabel formen