

Apr 08, 02 10:21

comments_add.tcl

Page 1/1

```
# Set the form variables 'name', 'email', 'text', and 'url'.

# The procedure 'set_the_usual_form_variables' works as the procedure
# 'set_form_variables' except that it also provides versions of the
# variables where all occations of "'" in the values are exchanged
# with "''"; these versions of the variables are named with QQ
# preceding the variable-name and are useful because Oracle needs to
# be fed "''" when we really mean "'". Thus if a form variable myvar
# contains "O'Brian", the variable QQmyvar contains "O''Brian".

set_the_usual_form_variables

# Test the content of the form variables
# ...

# Get a database handle
set db [ns_db gethandle]

# The query
ns_db dml $db "insert into comments (id, url, name, email,
                                text, insertdate)
              values (comment_sequence.nextval, '$url', '$QQname',
                    '$QQemail', '$QQtext', sysdate)"

# Redirect to the Show-comments page
ns_returnredirect "comments.tcl?url=$url"
```

Apr 08, 02 10:20

comments.sql

Page 1/1

```
drop table comments;

create table comments (
  page varchar(100) not null,
  name varchar(100) not null,
  email varchar(100),
  text varchar(2000) not null,
  insertdate date
);

insert into comments (page, name, email, text, insertdate)
values ('dollarkurs.tcl', 'Martin Elsman', 'mael@itu.dk', 'Great service!', sysdate);
```

Apr 08, 02 10:21

comments.tcl

Page 1/2

```

# Set the form variable 'url' - the url for the page for
# which we want the comments
set_form_variables

# Check the validity of the url-address
if { ![regexp {^http://.+}$ $url] } {
    ns_return 200 text/html "Error: expecting URL!"
    return
}

# Get a database handle
set db [ns_db gethandle]

# The select-query
set query "select id, url, name, email, text, insertdate
          from comments
          where url = '$url'"

# Perform the query
set selection [ns_db select $db $query]

# Loop through the rows
set comments ""
while { [ns_db getrow $db $selection] } {
    # make it possible to refer to the column names
    # as tcl-variables
    set_variables_after_query

    # Append a comment to the tcl-variable 'comments'
    append comments "
        <li> $text <br><a href=\"mailto:$email\">$name</a>
        - $insertdate
        - <a href=\"delete_form.tcl?url=$url&id=$id\">admin</a><p>"
}

# Return a page with the comments to the user - the perfect
# solution returns a message 'no reader comments' if there are
# no comments associated with the page

ns_return 200 text/html "
<html>
<body bgcolor=white>
<h2>Comments to <a href=\"$url\">$url</a></h2>
<ul>$comments</ul>
<h2>Add a Comment?</h2>
<form action=comments_add.tcl>
<input type=hidden name=url value=\"$url\">
<table align=center>
  <tr><td><b>Your email:</b><br>
    <input type=text name=email size=30>
  </td></tr>
  <tr><td><b>Your name:</b><br>
    <input type=text name=name size=30>
  </td></tr>
  <tr><td><b>Your comment:</b><br>
    <textarea wrap=virtual name=text cols=70 rows=5></textarea>
  </td></tr>
  <tr><td align=center><input type=submit value=\"Add Comment\">
  </td></tr>
</table>
</form>

```

Apr 08, 02 10:21

comments.tcl

Page 2/2

```

<hr>
<address>
<a href=\"mailto:mael@it.edu\">mael@it.edu</a>
</address>
</body></html>"

```

Apr 08, 02 10:22

delete_form.tcl

Page 1/1

```
# set the form variables 'url' og 'id'
set_form_variables

# Testing form-variables
if { ![regexp {^http://.+}$ $url] } {
    ns_return 200 text/html "Error: expecting URL!"
    return
}

if { ![regexp {^[0-9]+$} $id] } {
    ns_return 200 text/html "Error: security violation!"
    return
}

# Returning a page to the user
ns_return 200 text/html "
<html>
<head>
<title>Comment Service Password Request
</title>
</head>
<body bgcolor=white>
<h2>Comment Service Password Request</h2>
To delete the comment, enter the administrator password:
<form action=delete.tcl>
<input type=password name=password>
<input type=hidden name=url value=$url>
<input type=hidden name=id value=$id><p>
<input type=submit value=Delete>
</form>
<hr>
<address>
<a href=\"mailto:mael@it.edu\">mael@it.edu</a>
</address>
</body>
</html>"
```

Apr 08, 02 10:23

delete.tcl

Page 1/1

```
# set the form variables 'id', 'url', and 'password'
set_form_variables

# Check the validity of the url-address
if { ![regexp {^http://.+}$ $url] } {
    ns_return 200 text/html "Error: expecting URL!"
    return
}

# Check that $id is indeed an integer
if { ![regexp {^[0-9]+$} $id] } {
    ns_return 200 text/html "Error: security violation!"
    return
}

# Check the password
if { [string compare $password "yes!!!"] != 0 } {
    ns_return 200 text/html "The password you entered
is incorrect..."
    return
}

# The password is correct - get a database handle
# and do the deletion...
set db [ns_db gethandle]
ns_db dml $db "delete from comments where id = $id"

# Redirect the user to the comments page
ns_returnredirect "comments.tcl?url=$url"
```

Apr 08, 02 10:31

project_new_form.html

Page 1/1

```

<html>
<body bgcolor=white>
<center>
<h1> Projektbørs - opret nyt projekt</h1>
<form action=project_new.tcl>
<table border=0>
<tr><th align=left>Projektittel<td align=right><input type=text name=title size=
30>
<tr><th align=left>Dit navn<td align=right><input type=text name=name size=30>
<tr><th align=left>Din emailadresse<td align=right><input type=text name=email s
ize=30>
<tr><th align=left>Vælg et password<td align=right><input type=password name=pas
sword size=30>
<tr><td colspan=2><textarea name=text cols=50 rows=10></textarea>
</table>
<input type=submit value="Opret nyt projekt">
</form>
</center>
<hr>
<address>
<a href="mailto:mael@itu.dk">mael@itu.dk</a>
</address>
</body>
</html>

```

Apr 08, 02 10:31

project_new.tcl

Page 1/1

```

# Set the form variables 'title', 'name', 'email', and 'password'
# and the QQ versions for insertion in the database.

set_the_usual_form_variables

# check form variables

if { ![regexp {[a-zA-ZæøåÆØÅ \-]+} $title] } {
    ns_return 200 text/html "Titlen må kun indeholde almindelige
        bogstaver!"
    return
}

if { ![regexp {^.+@.+\.+.$} $email] } {
    ns_return 200 text/html "Du skal skrive en rigtig emailadresse!"
    return
}

if { $text == "" } {
    ns_return 200 text/html "Du mangler en projektbeskrivelse!"
    return
}

if { $name == "" } {
    ns_return 200 text/html "Du mangler at skrive dit navn!"
    return
}

set db [ns_db gethandle]

set insert_sql "insert into project (title, text, adm_name, adm_email, adm_passw
ord)
                values ('$QQtitle', '$QQtext', '$QQname', '$QQemail', '$QQpasswo
rd')"

if [ catch { ns_db dml $db $insert_sql } errmsg ] {
    ns_return 200 text/html "Oprettelse af projekt fejlet - prøv
        at gå tilbage og skriv en anden titel..."
    return
}

ns_returnredirect "project.tcl"

```

```

Apr 08, 02 10:31                project.tcl                Page 1/1

# return HTTP headers
ReturnHeaders

# immediately, write something for the user to look at
ns_write "<html>
<body bgcolor=white>

<center>
<h1>Projektbørs</h1>
\[\&nbsp; <a href=\"project_new_form.html\">Opret nyt projekt</a>&nbsp; \]
</center><hr>"

# Get a database handle from the web-server
set db [ns_db gethandle]

set query "select adm_name, adm_email, text, title
          from project order by title"

# Perform the query
set selection [ns_db select $db $query]

# Loop through the rows
while { [ns_db getrow $db $selection] } {
    set_variables_after_query

    # Write project to the client

    ns_write "<table width=95% align=center cellpadding=5 bgcolor=lightblue>
              <tr><th align=left>$title
              <td align=center> vedligeholdes af [mailto $adm_email $adm_name]
              <td align=right> <a href=\"project_update_form.tcl?title=[ns_urlen
code $title]\">opdater</a>
              <tr><td colspan=3> $text
              </table><p>"
}

# Finally, write the end of the page to the client
ns_write "<hr><address>[mailto mael@it.edu mael@it.edu]</address>
</body></html>
"

```

```

Apr 08, 02 10:31                project_update_form.tcl                Page 1/1

# set the form variable title
set_the_usual_form_variables

set db [ns_db gethandle]
set selection [ns_db lrow $db "select adm_name, adm_email, text from project whe
re title = '$QQtitle'"]
set_variables_after_query

return_page_with_title "Projektbørs - Opdatering" "
Velkommen [mailto $adm_email $adm_name]. <p>

Indtast dit password og dine ændringer til projekttteksten for
projektet ``$title``:
<center>
<form action=project_update.tcl>
<input type=hidden name=title value=\"$title\">
<table border=0><tr><td align=center colspan=2>
<textarea cols=50 rows=10 name=text>$text</textarea><p>
<tr><td><b>Password</b>&nbsp; <input type=password name=password><td align=right
>
<input type=submit value=\"Opdater projektttekst\">
</table>
</form>
</center>"

```

Apr 08, 02 10:31	project_update.tcl	Page 1/1
<pre># set the form variables title, password, text set_the_usual_form_variables set db [ns_db gethandle] set selection [ns_db lrow \$db "select adm_name, adm_email, text, adm_password from project where title = '\$QQtitle'"] set_variables_after_query if { \$password != \$adm_password } { ns_return 200 text/html "Du har indtastet et forkert password! Brug din browser til at gå tilbage..." return } if { \$QQtext == "" } { set update_sql "delete from project where title = '\$QQtitle'" } else { set update_sql "update project set text = '\$QQtext' where title = '\$QQtitle'" } ns_db dml \$db \$update_sql ns_returnredirect "project.tcl"</pre>		

Apr 08, 02 10:25	projekt.sql	Page 1/1
<pre>drop table project_member; drop table project; create table project (title varchar(100) primary key, text varchar(1000) not null, adm_email varchar(100) not null, adm_navn varchar(100) not null, adm_password varchar(20) not null); create table project_member (prj references project, email varchar(100) not null, name varchar(100) not null, unique(prj, email)); insert into project (title, text, adm_email, adm_name, adm_password) values ('Man on the moon', 'Vi vil have en dansker paa maanen - en astronaut samt to programmoerer soeges til at deltage i projektet!', 'mael@itu.dk', 'Martin Elsmann', 'astro'); insert into projekt_member (prj, email, name) values ('Man on the moon', 'kenneth@itu.dk', 'Kenneth Riis');</pre>		

Apr 08, 02 10:32	model.sql	Page 1/1
<pre> drop table restaurant_rating; drop table restaurant; drop sequence restaurant_id_sequence; create sequence restaurant_id_sequence start with 1; create table restaurant (restaurant_id integer primary key, name varchar(100) not null, address varchar(1000)); create table restaurant_rating (restaurant_id references restaurant, text varchar(1000), fullname varchar(100), email varchar(100), rating integer, check(rating >= 0 and rating <= 10)); </pre>		

Apr 08, 02 10:32	restaurant_add0.tcl	Page 1/1
<pre> set_the_usual_form_variables if { (![regexp {^0 ([1-9][0-9]*)\$} \$rating]) (\$rating > 10) (\$text == "") (\$fullname == "") (![regexp {^.+@.+\.+\$} \$email)] } { ns_returnredirect "restaurant_add.tcl?restaurant_id=\$restaurant_id" return } set db [ns_db gethandle] set insert_sql "insert into restaurant_rating (restaurant_id, text, fullname, email, rating) values (\$restaurant_id, '\$QQtext', '\$QQfullname', '\$QQemail', \$rating)" ns_db dml \$db \$insert_sql ns_returnredirect "restaurant.tcl" </pre>		

Apr 08, 02 10:32

restaurant_add.tcl

Page 1/1

```
# source some utility procedures
source "/web/mael/www/rating/util.tcl"

set_the_usual_form_variables

# Get a database handle
set db [ns_db gethandle]

# We are provided with either a restaurant_id or a restaurant name
if { [info exists restaurant_id] } {

    check_id $restaurant_id

    set query "select name from restaurant
              where restaurant_id = $restaurant_id"
    set name [database_to_tcl_string $db $query]
} else {

    check_name $name

    # Lets try to find it!
    set query "select restaurant_id from restaurant
              where name = '$QQname'"
    set selection [ns_db 0or1row $db $query]

    if { $selection == "" } {

        # Go to Oracle for a fresh restaurant_id
        set query2 "select restaurant_id_sequence.nextval from dual"
        set restaurant_id [database_to_tcl_string $db $query2]
        ns_db dml $db "insert into restaurant (restaurant_id, name)
                    values ($restaurant_id, '$QQname')"
    } else {
        set_variables_after_query
    }
}

# Now, both restaurant_id and name are set; return forms to the user

return_page_with_title "Your comments about the restaurant $name" "
<form action=restaurant_add0.tcl>
<input type=hidden name=restaurant_id value=$restaurant_id>
<textarea rows=5 cols=40 name=text></textarea><br>
<b>Email:</b>&nbsp;
<input type=text name=email size=30><br>
<b>Name:</b>&nbsp;
<input type=text name=fullname size=30><br>
<b>Rate (between 0 and 10):</b>&nbsp;
<input type=text name=rating size=2>&nbsp;
<input type=submit value=\"Rate it\">
</form>"
```

Apr 08, 02 10:32

restaurant_show.tcl

Page 1/1

```
# source some utility procedures
source "/web/mael/www/rating/util.tcl"

# Set the form variable 'restaurant_id'
set_form_variables

check_id $restaurant_id

set db [ns_db gethandle]

set query "select text, fullname, email, rating
          from restaurant_rating where restaurant_id = $restaurant_id"

set selection [ns_db select $db $query]

set body "<table width=95% bgcolor=lightblue cellpadding=5>
        <tr><th>Rating<th>Comment<th>Rater"

# loop through the comments
while { [ns_db getrow $db $selection] } {
    set_variables_after_query
    append body "<tr><th> [genstars $rating]
                <td> $text
                <td> [mailto $email $fullname]"
}

append body "</table>"

set name [database_to_tcl_string $db "select name from restaurant
                                where restaurant_id = $restaurant_id"]

return_page_with_title "Ratings - $name" $body
```

```

Apr 08, 02 10:32      restaurant.tcl      Page 1/1
# source some utility procedures
source "/web/mael/www/rating/util.tcl"

# start building up the body of the page
set body "<table width=95% bgcolor=lightblue border=0 cellpadding=5>
<tr><th>Restaurant<th>Average Score<th>Ratings<th>&nbsp;</th>"

# get a database handle
set db [ns_db gethandle]

# the complex query that calculates the scores
set query "select restaurant.restaurant_id, name,
          avg(rating) as average, count(*) as ratings
          from restaurant, restaurant_rating
          where restaurant.restaurant_id =
                restaurant_rating.restaurant_id
          group by restaurant.restaurant_id, name
          order by average desc"

set selection [ns_db select $db $query]

# loop through the restaurants
while { [ns_db getrow $db $selection] } {
    set_variables_after_query

    # convert floating point average to an integer value by floor'ing
    set average_int [format "%.0f" $average]

    # generate html-code for stars
    set stars [genstars $average_int]

    append body "<tr><td>
    <a href=\"restaurant_show.tcl?restaurant_id=$restaurant_id\">$name</a>
    <td>&nbsp;<td align=center>$ratings
    <td><a href=\"restaurant_add.tcl?restaurant_id=$restaurant_id\">rate it</a>
    >"
}

append body "</table>"

return_page_with_title "Restaurant Rating" "
$body
<form action=restaurant_add.tcl>
<h2>Rate new restaurant - type its name</h2>
<input type=text name=name size=30>&nbsp;<input type=submit value=\"Rate it...\">
</form>"

```

```

Apr 08, 02 10:32      util.tcl      Page 1/1
proc return_page { title body } {
    ns_return 200 text/html "<html>
    <title>$title</title><body bgcolor=white>
    $body
    <hr>
    <address>
    <a href=\"mailto:mael@it.edu\">mael@it.edu</a>
    </address>
    </body>
    </html>
    "
}

proc return_page_with_title { title body } {
    return_page $title "<h2>$title</h2> $body"
}

# A procedure for generating stars
proc genstars { n } {
    if { $n == 0 } {
        return ""
    } else {
        return "<img src=http://www.it.edu/courses/W2/star.gif>
        [genstars [expr $n - 1]]"
    }
}

proc check_id { id } {
    if { ![regexp {[0-9]+$] $id} } {
        ns_return 200 text/html "ERROR: check_id!"
        exit
    }
    return
}

proc check_name { name } {
    if { ![regexp {[a-zA-Z\_-\.\ '0-9&#x20;E&#x21;A]+$] $name} } {
        ns_return 200 text/html "Please provide a
        real name! Use the back-button on your browser to
        go back and fix the problem."
        exit
    }
    return
}

```

```

Apr 08, 02 10:36          pk_plade_ny.tcl          Page 1/1
# set the form variables 'titel', 'kunstner', 'aar'
set_the_usual_form_variables

if { ![regexp {[A-Za-z\-\_]' 0-9/]+$} $kunstner ] } {
    ns_return 200 text/html "Brug back-knappen i din browser og indtast et orden
tligt kunstnernavn"
    return
}

if { ![regexp {[A-Za-z\-\_]' 0-9/]+$} $titel ] } {
    ns_return 200 text/html "Brug back-knappen i din browser og indtast en orden
tlig titel"
    return
}

if { ![regexp {[0-9][0-9][0-9][0-9]$} $aar ] } {
    ns_return 200 text/html "Brug back-knappen i din browser og indtast et orden
tligt årstal"
    return
}

set insert_sql "insert into pk_plade (plade_id, kunstner, titel, aar)
                values (plade_id_sequence.nextval, '$QQkunstner', '$QQtitel', '$
aar'"

set db [ns_db gethandle]

# if the ns_db call fails, there is probably already a record entry
# with this (titel, kunstner, aar) triple...
catch { [ns_db dml $db $insert_sql] } errmsg

ns_returnredirect "pk_plader_vis.tcl"

```

```

Apr 08, 02 10:36          pk_plader_kunstner_vis.tcl          Page 1/1
# set the form variable 'QQkunstner'
set_the_usual_form_variables

if { ![regexp {[A-Za-z\-\_]' 0-9/]+$} $kunstner ] } {
    ns_return 200 text/html "Brug back-knappen i din
                        browser og indtast et
                        ordentligt kunstnernavn"
    return
}

set db [ns_db gethandle]

set query "select kunstner, titel, plade_id, aar
           from pk_plade where kunstner = '$QQkunstner'"

set selection [ns_db select $db $query]

set table "<table border bgcolor=silver>
           <tr><th>Kunstner<th>Titel<th>År"
while { [ns_db getrow $db $selection] } {
    set_variables_after_query
    append table "<tr><td>$kunstner
                <td><a href=\"pk_sange_vis.tcl?plade_id=$plade_id\">$titel</a>
                <td>$aar\n"
}
append table "</table>"

ns_return 200 text/html "<html>
<body bgcolor=white><center>
<h2>Pladekartotek - plader for $kunstner</h2>
$table</center>
<hr>
<address>
<a href=\"mailto:mael@it.edu\">mael@it.edu</a>
</address>
</body>
</html>"

```


Apr 08, 02 10:36	pk_sange_vis.tcl	Page 1/1
<pre> # set the form variable 'plade_id' set_form_variables # check the form variable if { ![regexp {[0-9]+\$} \$plade_id] } { ns_return 200 text/html "Error: plade_id is not a number - are you trying to fool the system?" return } # get a handle to the database set db [ns_db gethandle] # get the plade title set plade_titel [database_to_tcl_string \$db "select titel from pk_plade where plade_id = \$plade_id"] # get the songs set query "select titel from pk_sang where plade_id = \$plade_id" set selection [ns_db select \$db \$query] # construct an unordered list of songs set liste "" while { [ns_db getrow \$db \$selection] } { set_variables_after_query append liste "\$titel -- slet\n" } append liste "" ns_return 200 text/html "<html> <body bgcolor=white><center> <h2>Plade: \$plade_titel</h2> \[Tilbage til hovedside Vis plader \] \$liste <p> <form action=pk_sang_ny.tcl> <input type=hidden name=plade_id value=\$plade_id> Opret sang: <input type=text name=titel size=30> </form></center> <hr> <address> mael@it.edu </address> </body> </html>" </pre>		

Apr 08, 02 10:36	pk_sang_ny.tcl	Page 1/1
<pre> #set the form variables 'QQtitel', 'plade_id' set_the_usual_form_variables #Test the form variables if { ![regexp {[0-9]+\$} \$plade_id] } { ns_return 200 text/html "Error: plade_id is not a number - are you trying to fool the system?" return } if { ![regexp {[A-Za-z\-\._' 0-9/]+\$} \$titel] } { ns_return 200 text/html "Brug back-knappen i din browser og indtast en ordentlig titel" return } set insert_sql "insert into pk_sang (plade_id, titel) values (\$plade_id, '\$QQtitel') set db [ns_db gethandle] # catch failures due to the same title being # inserted more than once! catch { ns_db dml \$db \$insert_sql } errmsg ns_returnredirect "pk_sange_vis.tcl?plade_id=\$plade_id" </pre>		

Apr 08, 02 10:36

pk_sang_slet.tcl

Page 1/1

```
# set the form variables 'QQtitel', 'plade_id'
set_the_usual_form_variables

# check the form variables
if { ![regexp {[A-Za-z\-\.\_ ' 0-9/]+}$} $titel ] } {
    ns_return 200 text/html "Error: weird titel - are you
        trying to fool the system"
    return
}

if { ![regexp {[0-9]+$} $plade_id] } {
    ns_return 200 text/html "Error: plade_id is not a number -
        are you trying to fool the system?"
    return
}

set delete_sql "delete from pk_sang
    where plade_id=$plade_id
    and titel='$QQtitel'"

set db [ns_db gethandle]
ns_db dml $db $delete_sql
ns_returnredirect "pk_sange_vis.tcl?plade_id=$plade_id"
```

Apr 08, 02 10:36

pladekartotek.html

Page 1/1

```
<html>
<body bgcolor=white>
<h2>Pladekartotek</h2>
[<a href="pk_plader_vis.tcl">Vis plader</a>] <p>
<form action="pk_plader_kunstner_vis.tcl">
<b>Indtast kunstner</b><br>
<input type=text name=kunstner size=30>
</form>
<hr>
<address>
<a href="\mailto:mael@itu.dk\">mael@itu.dk</a>
</address>
</body>
</html>
```

Apr 08, 02 10:36	pladekartotek.sql	Page 1/1
<pre> drop table pk_sang; drop table pk_plade; drop sequence plade_id_sequence; create table pk_plade (plade_id integer primary key, titel varchar(200), kunstner varchar(200), aar char(4), unique(titel, kunstner)); create sequence plade_id_sequence start with 1; create table pk_sang (plade_id references pk_plade, titel varchar(200), unique(titel, plade_id)); insert into pk_plade (plade_id, titel, kunstner, aar) values (plade_id_sequence.nextval, 'Eric Clapton Unplugged', 'Eric Clapton', '1992'); insert into pk_sang (plade_id, titel) values (1, 'Layla'); insert into pk_sang (plade_id, titel) values (1, 'Tears in heaven'); </pre>		

Jun 09, 02 1:00	Table of Content	Page 1/1
Table of Contents		
1	<i>comments_add.tcl</i> ... sheets 1 to 1 (1) pages 1- 1 28 lines	
2	<i>comments.sql</i> sheets 1 to 1 (1) pages 2- 2 14 lines	
3	<i>comments.tcl</i> sheets 2 to 2 (1) pages 3- 4 68 lines	
4	<i>delete_form.tcl</i> sheets 3 to 3 (1) pages 5- 5 38 lines	
5	<i>delete.tcl</i> sheets 3 to 3 (1) pages 6- 6 30 lines	
6	<i>project_new_form.html</i> sheets 4 to 4 (1) pages 7- 7 22 lines	
7	<i>project_new.tcl</i> sheets 4 to 4 (1) pages 8- 8 41 lines	
8	<i>project.tcl</i> sheets 5 to 5 (1) pages 9- 9 44 lines	
9	<i>project_update_form.tcl</i> sheets 5 to 5 (1) pages 10- 10 23 lines	
10	<i>project_update.tcl</i> .. sheets 6 to 6 (1) pages 11- 11 25 lines	
11	<i>projekt.sql</i> sheets 6 to 6 (1) pages 12- 12 28 lines	
12	<i>model.sql</i> sheets 7 to 7 (1) pages 13- 13 23 lines	
13	<i>restaurant_add0.tcl</i> . sheets 7 to 7 (1) pages 14- 14 19 lines	
14	<i>restaurant_add.tcl</i> .. sheets 8 to 8 (1) pages 15- 15 53 lines	
15	<i>restaurant_show.tcl</i> . sheets 8 to 8 (1) pages 16- 16 33 lines	
16	<i>restaurant.tcl</i> sheets 9 to 9 (1) pages 17- 17 48 lines	
17	<i>util.tcl</i> sheets 9 to 9 (1) pages 18- 18 45 lines	
18	<i>pk_plade_ny.tcl</i> sheets 10 to 10 (1) pages 19- 19 30 lines	
19	<i>pk_plader_kunstner_vis.tcl</i> sheets 10 to 10 (1) pages 20- 20 40 lines	
20	<i>pk_plader_vis.tcl</i> ... sheets 11 to 11 (1) pages 21- 21 44 lines	
21	<i>pk_plade_slet.tcl</i> ... sheets 11 to 11 (1) pages 22- 22 25 lines	
22	<i>pk_sange_vis.tcl</i> sheets 12 to 12 (1) pages 23- 23 50 lines	
23	<i>pk_sang_ny.tcl</i> sheets 12 to 12 (1) pages 24- 24 28 lines	
24	<i>pk_sang_slet.tcl</i> sheets 13 to 13 (1) pages 25- 25 27 lines	
25	<i>pladekartotek.html</i> .. sheets 13 to 13 (1) pages 26- 26 15 lines	
26	<i>pladekartotek.sql</i> ... sheets 14 to 14 (1) pages 27- 27 29 lines	