```
ad_form -name form_name \
    -mode display | edit \                          ;# Defaults to edit.
    -action "URL to act upon" \
    -extend \
    -export "list of vars to export as hidden" \
    -actions {{" Label " key}...} \                 ;# Creates buttons. Check the
                                                    ;# clicked on action with [form get_action  name]
    -html "string" \                                ;# HTML to be included in the form declaration.
    -has_edit 0 | 1 \                               ;#  Suppress the Edit button added by the form builder.
    -has_submit 0 | 1 \                             ;#  Suppress the Submit  button added by form builder.
    -select_query_name \                            ;# Name of query to populate form.
    -confirm_template "string" \                    ;# See notes.
    -cancel_url "string" \                          ;# URL to redirect to if user hits cancel button.
    -cancel_label "string" \
    -show_required_p \                              ;# Show red stars by required elements (default true)
    -form {
    element_name:key(sequence_name)
    { element_name:datatype(widget) , optional, multiple, to_sql(linear_date),
                                      from_sql(linear_date_no_time), to_html(sql_date)
          {label "string"}
          {value "value to be assigned to element"}
          {values "list of values"}                 ;# Defaults for multiple values (select, checkboxes)
          {options {{label  value} {...}}}          ;# Values for select and radio buttons.
          {before_html "string"}                    ;# Displayed immediately before the rendered element.
          {after_html "string"}                     ;# Displayed immediately after the rendered element.
          {html {name value name value ...}}        ;# Name-value attribute pairs for the HTML tag.
          {mode display | edit | <empty>}           ;# If empty, use the form's -mode value.
          {display_value "string"}                  ;# Value used when the element is in display mode.
          {section "string"}                        ;# Displays a bar above the element with this value.
          {maxlength integer}                       ;# The maximum allowable length in bytes.
          {sign}                                    ;# Sign hidden widdget (to prevent tampering).
          {help}                                    ;# Display helpful hints (date widget only?)
          {help_text "Show this as help text under element." }
          {format "MONTH DD YYYY HH12:MI AM"}
    }
    {...}
} -validate {
          { element_name
               { tcl code that returns 0 or 1 ($element_name is set) }
               "message to be shown in case of error"
          }
          {...}
} -on_request {
    A code block which sets the values for each element of the form  meant to be modifiable by
    the user when the built-in key management feature is being used or to define options for select
    lists etc.
    This block is executed every time, except when the form is being submitted.
} -on_submit {
    When the form is submitted, this code block will be executed before any new_data or
    edit_data code block.
    * Use this if your form doesn't interact with the database or if the database type involved
      includes a Tcl API that works for  both new and existing data.
    * The values of the form's elements are available as local variables.
} -after_submit {
    This code block will be executed after the three blocks on_submit, new_data or edit_data have
     been executed. It is useful for putting in stuff like ad_returnredirect that is the same for new
    and edit.
} -on_refresh {
    Executed when the form comes back from being refreshed using  javascript with the
    __refreshing_p flag set.
} -select_query {
     A query that returns a single row containing values for each element of the form meant to be
     modifiable by the user. Can only be used if an element of type key has been declared.
} -new_data {
    * Executed when a form for a new database row is submitted.
    * Should insert the data into the database or create a new database object or content repository
      item containing the data.
} -edit_data {
    * Executed when a form for an existing database row is submitted.
    * Should update the database or create a new content revision for the exisiting item if the
      data's stored in the content repository.
} -new_request {
    * Sets the values for each element meant to be modifiable by the user.
    * Use when a single query to grab database values is insufficient.
    * Needs an element of type key.
    * This block complements the -edit_request block.
    * Set the values as local variables in the code block, and they'll get fetched and used as
      element values for you.
} -edit_request {
    * Sets the values for each element meant to be modifiable by the user.
    * Use when a single query to grab database values is insufficient.
    * Needs an element of type key.
    * Only executed if the page is called with a valid key, i.e. a self-submit form to add or edit an
      item called to edit the data.
    * Set the values as local variables in the code block, and they'll get fetched and used as
      element values for you.
}
```

Required
Optional
Value

# ad_form Quick Reference Sheet
by Roberto Mello
last updated: Aug 1st 2003

## Datatypes
(defined in template::data::validate)

| | | | |
|---|---|---|---|
| boolean | filename | search | url |
| date | integer | string | |
| email | keyword | text | |

## Widgets
(defined in template::widget)

| | | |
|---|---|---|
| AmpmFragment | hidden | richtext |
| button | inform | radio |
| checkbox | input | search |
| comment | menu | select |
| currency | monthFragment | submit |
| date | multiselect | text |
| dateFragment | numericRange | textarea |
| file | password | |

## Notes:
- confirm_template:Display the data then include /packages/acs-templating/resources/forms/confirm-button
- Key element: defaults to acs_object_id_seq

## Examples
- Multiple select with five choices, in a 4-line select.

```
{my_key:text(multiselect),multiple
    {label "Select some values"}
    {options {first second third fourth fifth}}
    {html {size 4}}}
```

- Date widget, with help hints

```
{expiration_date:date,to_sql(linear_date),
             from_sql(linear_date_no_time)
    {label "When does listing expire?"}
    {format "MONTH DD YYYY"} {help} }
```

- Multiple checkboxes

```
{days:text(checkbox),multiple
   {label "Some Days"}
   {options {{"Sun" 0} {"Mon" 1} {"Tue" 2}}}
   {values $days}}
```

- Multiple elements being put in array

```
{foo.1:datatype(widget) ...}
{foo.2:datatype(widget) ...}
```

Then in ad_page_contract:

```
{foo:array,optional}
```

- Your Example Here

## URLs of Interest:
http://openacs.org/api-doc/proc-view?proc=ad_form
http://www.jongriffin.com/static/openacs/ad_form/using-ad-form
http://www.rubick.com:8002/openacs/ad_form