NAME

forms – character based forms package

SYNOPSIS

#include <form.h>

MT-LEVEL

Unsafe

DESCRIPTION

The **form** library is built using the **curses** library, and any program using **forms** routines must call one of the **curses** initialization routines such as **initscr**. A program using these routines must be compiled with **–lform** and **–lcurses** on the **cc** command line.

The **forms** package gives the applications programmer a terminal-independent method of creating and customizing forms for user-interaction. The **forms** package includes: field routines, which are used to create and customize fields, link fields and assign field types; fieldtype routines, which are used to create new field types for validating fields; and form routines, which are used to create and customize forms, assign pre/post processing functions, and display and interact with forms.

Current Default Values for Field Attributes

The **forms** package establishes initial current default values for field attributes. During field initialization, each field attribute is assigned the current default value for that attribute. An application can change or retrieve a current default attribute value by calling the appropriate set or retrieve routine with a **NULL** field pointer. If an application changes a current default field attribute value, subsequent fields created using **new_field( )** will have the new default attribute value. (The attributes of previously created fields are not changed if a current default attribute value is changed.)

Routine Name Index

The following table lists each **forms** routine and the name of the manual page on which it is described.

forms Routine Name	Manual Page Name
current_field	form_page(3X)
data_ahead	form_data(3X)
data_behind	form_data(3X)
dup_field	form_field_new(3X)
dynamic_field_info	form_field_info(3X)
field_arg	form_field_validation(3X)
field_back	form_field_attributes(3X)
field_buffer	form_field_buffer(3X)
field_count	form_field(3X)
field_fore	form_field_attributes(3X)
field_index	form_page(3X)
field_info	form_field_info(3X)
field_init	form_hook(3X)
field_just	form_field_just(3X)
field_opts	form_field_opts(3X)
field_opts_off	form_field_opts(3X)

field_opts_on	**form_field_opts(3X)**
field_pad	**form_field_attributes(3X)**
field_status	**form_field_buffer(3X)**
field_term	**form_hook(3X)**
field_type	**form_field_validation(3X)**
field_userptr	**form_field_userptr(3X)**
form_driver	**form_driver(3X)**
form_fields	**form_field(3X)**
form_init	**form_hook(3X)**
form_opts	**form_opts(3X)**
form_opts_off	**form_opts(3X)**
form_opts_on	**form_opts(3X)**
form_page	**form_page(3X)**
form_sub	**form_win(3X)**
form_term	**form_hook(3X)**
form_userptr	**form_userptr(3X)**
form_win	**form_win(3X)**
free_field	**form_field_new(3X)**
free_fieldtype	**form_fieldtype(3X)**
free_form	**form_new(3X)**
link_field	**form_field_new(3X)**
link_fieldtype	**form_fieldtype(3X)**
move_field	**form_field(3X)**
new_field	**form_field_new(3X)**
new_fieldtype	**form_fieldtype(3X)**
new_form	**form_new(3X)**
new_page	**form_new_page(3X)**
pos_form_cursor	**form_cursor(3X)**
post_form	**form_post(3X)**
scale_form	**form_win(3X)**
set_current_field	**form_page(3X)**
set_field_back	**form_field_attributes(3X)**
set_field_buffer	**form_field_buffer(3X)**
set_field_fore	**form_field_attributes(3X)**
set_field_init	**form_hook(3X)**
set_field_just	**form_field_just(3X)**
set_field_opts	**form_field_opts(3X)**
set_field_pad	**form_field_attributes(3X)**
set_field_status	**form_field_buffer(3X)**
set_field_term	**form_hook(3X)**
set_field_type	**form_field_validation(3X)**
set_field_userptr	**form_field_userptr(3X)**
set_fieldtype_arg	**form_fieldtype(3X)**
set_fieldtype_choice	**form_fieldtype(3X)**
set_form_fields	**form_field(3X)**

set_form_init	**form_hook(3X)**
set_form_opts	**form_opts(3X)**
set_form_page	**form_page(3X)**
set_form_sub	**form_win(3X)**
set_form_term	**form_hook(3X)**
set_form_userptr	**form_userptr(3X)**
set_form_win	**form_win(3X)**
set_max_field	**form_field_buffer(3X)**
set_new_page	**form_new_page(3X)**
unpost_form	**form_post(3X)**

RETURN VALUES

Routines that return a pointer always return **NULL** on error. Routines that return an integer return one of the following:

E_OK	–	The function returned successfully.
E_CONNECTED	–	The field is already connected to a form.
E_SYSTEM_ERROR	–	System error.
E_BAD_ARGUMENT	–	An argument is incorrect.
E_CURRENT	–	The field is the current field.
E_POSTED	–	The form is posted.
E_NOT_POSTED	–	The form is not posted.
E_INVALID_FIELD	–	The field contents are invalid.
E_NOT_CONNECTED	–	The field is not connected to a form.
E_NO_ROOM	–	The form does not fit in the subwindow.
E_BAD_STATE	–	The routine was called from an initialization or termination function.
E_REQUEST_DENIED	–	The form driver request failed.
E_UNKNOWN_COMMAND	–	An unknown request was passed to the form driver.

SEE ALSO

curses(3X), and 3X pages whose names begin "form_" for detailed routine descriptions.

NOTES

The header **<form.h>** automatically includes the headers **<eti.h>** and **<curses.h>**.