```
/****
 *
 * Implementation of change-card-dialog.h.
 *
 */

#include "change-card-dialog.h"
#include "label.h"
#include "rolodex-menu-ui.h"
#include <InterViews/box.h>
#include <InterViews/button.h>
#include <InterViews/frame.h>
#include <InterViews/glue.h>
#include <InterViews/streditor.h>
#include <stdlib.h>


ChangeCardDialog::ChangeCardDialog(Rolodex* r, RolodexMenuUI* rmui) :
        View(NULL, r) {
    okcb = new OKChangeButton(r, this);
    ccb = new CancelChangeButton(rmui);
}

ChangeCardDialog::~ChangeCardDialog() {
}

void ChangeCardDialog::Compose() {
    /*
     * Make the window of this a VBox.
     */
    w = new VBox();
    ((VBox*) w)->Align(Center);

    /*
     * Stick some stretchable glue spacing above the prompting message.
     */
    ((VBox*) w)->Insert(new VGlue(round(0.20 * inch), 0, round(0.50 * inch)));

    /*
     * Insert the prompting message into the top of the vbox.  The prompting
     * message itself is in an hbox with some glue on each side.  This glue
     * will enable horizontal stretching of the message, which stretchability
     * will transfer out to the entire scene that the message is in.
     */
    ((VBox*) w)->Insert(new Label(
        "Enter Id of Card to Change:",
            "times", "bold", "r", 18));

    /*
     * Stick some stretchable glue spacing between the message and the field
     * editors below.
     */
    ((VBox*) w)->Insert(new VGlue(round(0.20 * inch), 0, round(0.50 * inch)));

    /*
     * Insert the string editor to enter the card id.  The subsequent call to
     * Message("") clears the blank chars out after the editor is built and
     * sized.  In this way, the editor properly sized and initially empty when
     * user frist types in it.
     */
    ((VBox*) w)->Insert(new ShadowFrame(seid =
        new StringEditor(bs, "                         ")));
    seid->Message("");

    /*
     * Compose the OK and Cancel dialog buttons and insert into the window.
     */
    ComposeButtons();
}

void ChangeCardDialog::ComposeButtons() {
    /*
     * Make an HBox to hold the buttons
     */
    hb = new HBox();

    /*
     * Enter the buttons into the box, delimited with glue.
     */
    hb->Insert(new HGlue(round(0.2 * inch), hfil, hfil));
    hb->Insert(okcb);
    hb->Insert(new HGlue(round(0.2 * inch), hfil, hfil));
    hb->Insert(ccb);
    hb->Insert(new HGlue(round(0.2 * inch), hfil, hfil));

    /*
     * Put the hbox into the window, with some glue.
     */
    ((VBox*)w)->Insert(new VGlue(round(0.2 * inch), 0, vfil));
    ((VBox*)w)->Insert(hb);
    ((VBox*)w)->Insert(new VGlue(round(0.2 * inch), 0, vfil));
}

String* ChangeCardDialog::GetId() {
    return new String(seid->Text());
}

OKChangeButton::OKChangeButton(Rolodex* r, ChangeCardDialog* ccd) :
        PushButton("OK", new ButtonState(), true) {
                // ^^^   ^^^^^^^^^^^^^^^^^  ^^^^ Default starting state; since
                // ^^^   ^^^^^^^^^^^^^^^^^  ^^^^ we will not use ButtonStates,
                // ^^^   ^^^^^^^^^^^^^^^^^  ^^^^ our starting state is always
                // ^^^   ^^^^^^^^^^^^^^^^^  ^^^^ true
                // ^^^   ^^^^^^^^^^^^^^^^^
                // ^^^   ^^^^^^^^^^^^^^^^^ Dummy value for the ButtonState of
                // ^^^   ^^^^^^^^^^^^^^^^^ this button.  We will not normally
                // ^^^   ^^^^^^^^^^^^^^^^^ use ButtonState objects
                // ^^^
                // ^^^ Text label inside button

    this->r = r;
    this->ccd = ccd;
```

```
}

void OKChangeButton::Press() {
    r->Change(atoi(ccd->GetId()->ConstConvert()), NULL);
}

CancelChangeButton::CancelChangeButton(RolodexMenuUI* rmui) :
        PushButton("Cancel", new ButtonState(), true) {
    this->rmui = rmui;
}

void CancelChangeButton::Press() {
    rmui->ShowEmptyDialog();
}
```