

NORTHWEST FLORIDA STATE
COLLEGE

Department of Mathematics

AcroT_EX Bundle
Testing Form Creation
The AeB_Pro Version

D. P. Story

1. Introduction

This file is really meant for people who use Acrobat 5.0 or later and create PDF documents using the Acrobat Distiller. For those who use pdftex or dvipdfm, see the companion document `eqforms_pd.tex`.

This document contains examples of constructing the various types of form fields: choice fields, (list box, combo box), button fields (push button, check box, radio button) and text fields. In the section on text fields, there is a comb field defined, this is a feature of Acrobat 6.0, if you are using Acrobat 5.0, the combs will not appear.

The documentation for form fields can be found in the document `eformman.pdf`.

2. Choice Fields

There are two types of choice fields, list box and combo box.

2.1. List Box

List Box:

List Box:

Here's the same two list boxes using the new user interface, key-value system:

List Box:

List Box:

2.2. Combo Box

Combo Box:

Now, the same combo box using the user interface:

Combo Box:

3. Button Fields

There are three types of button fields, push buttons, radio buttons, and check boxes.

3.1. Pushbuttons

The following button is of standard design, and resets all forms in the document.

Push Button with user interface:

The push button below uses `aeb_pro` techniques in the `docassembly` environment in the preamble of this document. The JavaScript imports the icons and embeds them in the document as named icons (so they can be used and reused in several push buttons. There is also some JavaScript that attaches these icons to this particular field.

Push Button

The eCalculus series illustrate another technique of creating custom appearance faces for a push button. These icons were created in the `createImage` environment, using PostScript commands. Very cool!

Push Button

Push Button

Push Button

Push Button

Here's the same four buttons using the new user interface:

Push Button

Push Button

Push Button

Push Button

Another variation is to embed an EPS file in the document (using `graphicxsp`) and using it for an appearance face of a push button.

There are a couple of techniques that can be used to rescale an button

with a custom image. The first method—used in the first two buttons below—use the `calc` package to do arithmetic on natural width and height of the image. The commands `\widthOf` and `\heightOf` are defined in `graphicxsp` can be used to make calculations on the width and height respectively.

Push button, scaled by 0.25:

Push button, scaled to a width of .5in:

Another approach, which is my personal favorite, it to rescale the button using `graphicx` commands, in this case we use `\resizebox` to scale to a width of .5in.

Push button:

You can also simply set the width and the height of the button to whatever you wish, here with proportional scaling,

Push button: and here with *anamorphic scaling*,
The latter scales the image to fit exactly into the button dimensions,

some distortion may be apparent.

3.2. Radio Buttons

Radio Button: Radio Button:

Here's the same radio buttons using the friendly user-interface:

Radio Button: Radio Button:

Custom Radio:

3.3. Checkboxes

Check Box:

Now, using the friendly user-interface

Check Box:

The check box below also has a custom appearance. The icons were created with the `createImage` environment.

Custom Appearance Check Box: the user-friendly version:

4. Text Fields

The user enters text into a text field. The field may accept a single line of text, as with Text Field 1, or may accept multiple lines of text, as in Text Field 2, which has been rotated.

Text Field 1:

Text Field 2

Comb Field:

Calculated Field: Enter numbers in the first two fields, the third is the sum of the first two.

Number 1:

Number 2:

Total:

Now the same text fields using the friendly interface:

Text Field 1:

Text Field 2

Comb Field:

Calculated Field: Enter numbers in the first two fields, the third is the sum of the first two.

Number 1:

Number 2:

Total: