Table of Contents

Acumen
Training

The Acrobat User

Acrobat Spell Checking
Acrobat has a seldom-used spelling checker that can examine the contents of text fields and annotations. It is easy to use and easy to integrate into a form's user interface.

Short topics this month

I'm in the final stages of turning out a new class and a book (tell you about the latter next month), so this issue's articles are shorter than average. Sorry, Gang!

PostScript Tech

The Miter Limit
This mildly obscure PostScript parameter does not sound like something to which you would ever need to pay attention. Surprisingly, under some circumstances it can solve an otherwise intractable visual problem.

Class Schedule

Apr–May–Jun

What's New?

Introducing a New Class: *Variable Data PostScript*
Learn how to create data-driven PostScript documents. First class: April 28.

Contacting Acumen

Telephone number, email address, postal address, all the ways of getting to Acumen.

Journal feedback: suggestions for articles, questions, etc.

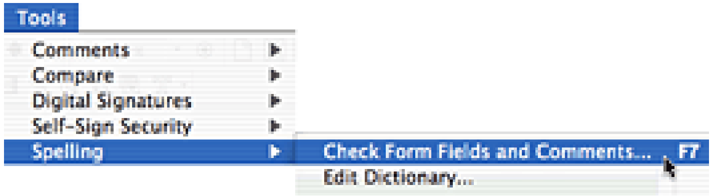
Acrobat Spell Checking

One feature of Acrobat 5 that escaped my attention for a surprisingly long time was its built-in spelling checker. Acrobat can check the spelling of the contents of text fields and comments (annotations); the process is very much like checking the spelling in any word processor or page layout program.

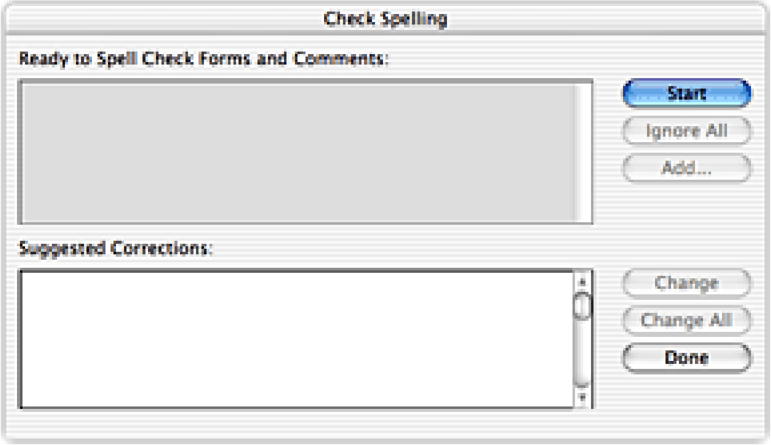
This month, we shall look at how to use this the spelling checker and how to add a "Check Spelling" button to your forms.

Selecting Spell Checking

You initiate spell checking in Acrobat by selecting *Tools>Spelling>Check Form Fields and Comments…*



Acrobat will present you with the *Check Spelling* dialog box; this is the "control panel" you use while Acrobat looks for misspellings in your form fields and annotations.

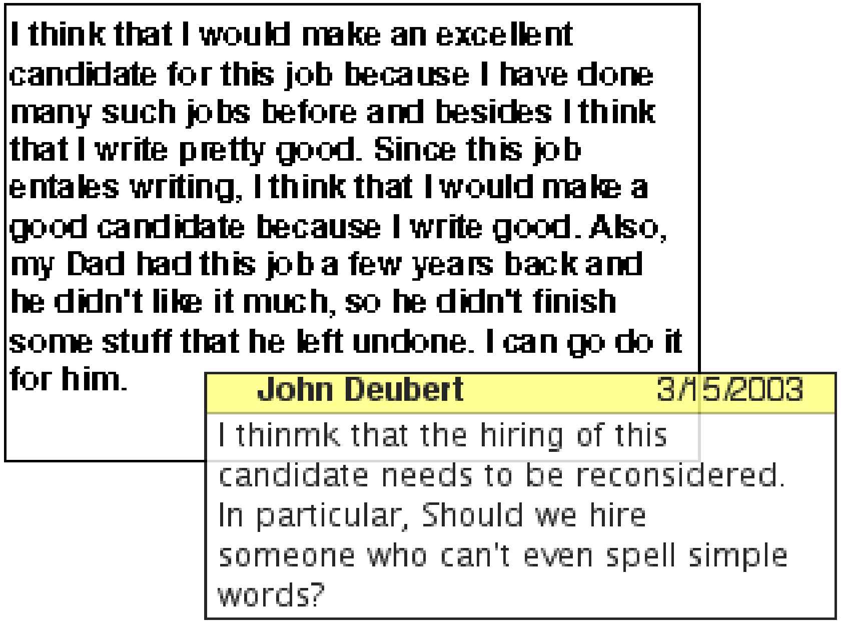


Click on the *Start* button and Acrobat will start scanning through the comments and text fields in your file.

Next Page ->

For Example

Acrobat scans through all of the PDF file’s text fields and comments, stopping at each misspelled word it discovers. If your Acrobat file contained the text field and annotation at right, then the *Check Spelling* dialog box report a misspelling as illustrated below at right.



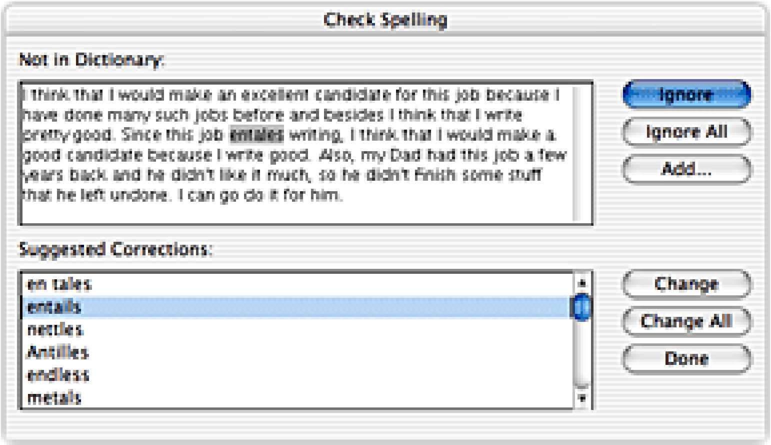
I think that I would make an excellent candidate for this job because I have done many such jobs before and besides I think that I write pretty good. Since this job entales writing, I think that I would make a good candidate because I write good. Also, my Dad had this job a few years back and he didn't like it much, so he didn't finish some stuff that he left undone. I can go do it for him.

John Deubert 3/15/2003

I thinmk that the hiring of this candidate needs to be reconsidered. In particular, Should we hire someone who can't even spell simple words?

Note that the upper field in the *Check Spelling* dialog box displays the original form field or comment text with the misspelled word highlighted.

The lower field presents a list of possible replacements for the misspelled word.



Check Spelling

Not in Dictionary:

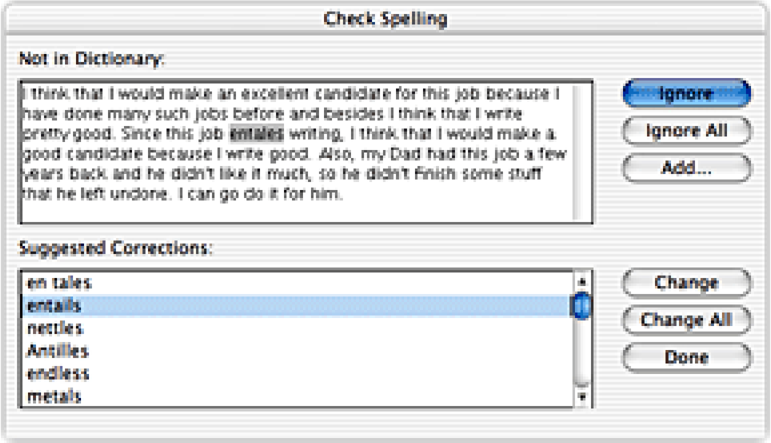
I think that I would make an excellent candidate for this job because I have done many such jobs before and besides I think that I write pretty good. Since this job entales writing, I think that I would make a good candidate because I write good. Also, my Dad had this job a few years back and he didn't like it much, so he didn't finish some stuff that he left undone. I can go do it for him.

Suggested Corrections:

- en tales
- entails
- nettles
- Antilles
- endless
- metals

Next Page ->

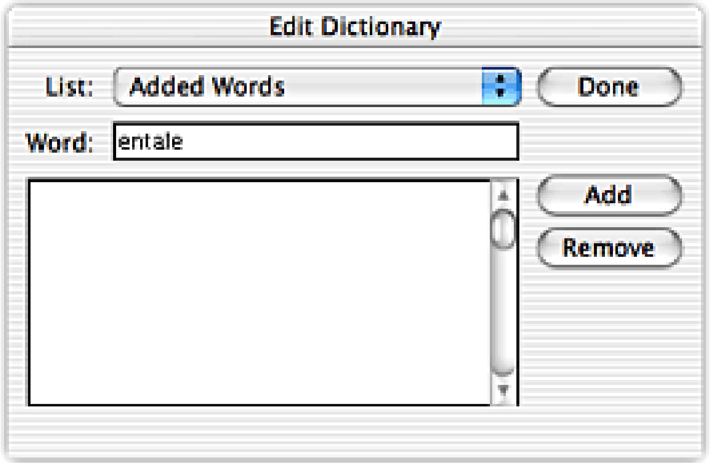
The buttons along the right side of the dialog box have meanings similar to the same buttons in any spelling checker:



I think that I would make an excellent candidate for this job because I have done many such jobs before and besides I think that I write pretty good. Since this job entales writing, I think that I would make a good candidate because I write good. Also, my Dad had this job a few years back and he didn't like it much, so he didn't finish some stuff that he left undone. I can go do it for him.

- en tales
- entails
- nettles
- Antilles
- endless
- metals

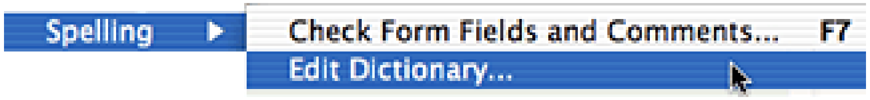
Ignore	Don’t correct that item.
Ignore all	Don’t correct any instances of that item.
Add	Add this item to your private dictionary. Acrobat will launch the *Edit Dictionary* dialog box (below) to let you tailor the new entry to your needs.
Change	Change the item to the replacement you select in the *Suggested Corrections* list.
Change All	Change all instances of the misspelled item to the replacement you select in the *Suggested Corrections* list.
Done	Quit the spell checker.



List: Added Words

Word: entale

By the way, you can edit your private dictionary at any time by selecting *Tools>Spelling>Edit Dictionary.*



Next Page ->

Not in *Acrobat Reader*

Unfortunately, the Spell Checking feature doesn’t exist in *Acrobat Reader.* You must have either *Acrobat Approval* or the full *Adobe Acrobat.*

Try it out yourself

At right is the same text field as is illustrated on the previous page. If you are reading this *Journal* with the full *Acrobat* or *Acrobat Approval,* type some text into this text field (or just leave the field’s default text in place) and select *Tools>Spelling>Check...* You can watch Acrobat check the spelling in this text field (and every other text field in this copy of the Journal).

I think that I would make an excellent candidate for this job because I have done many such jobs before and besides I think that I write pretty good. Since this job entales writing, I think that I would make a good candidate because I write good. Also, my Dad had this job a few years back and he didn't like it much, so he didn't finish some stuff that he left undone. I can go do it for him.

Next Page ->

Making a "Check Spelling" Button

At right we have the same text field as on the previous page, but this time, we have placed a button beneath it, labeled "Check Spelling." Click on this button (go ahead) and Acrobat will display the *Check Spelling* dialog box and otherwise carry out its usual spell-checking routine (assuming you are using something other than *Acrobat Reader* to look at this *Journal).*

Providing a *Check Spelling* button is a useful thing to do in forms that require the user to write some non-trivial piece of text (for example, explanatory notes in an insurance claim form).

I think that I would make an excellent candidate for this job because I have done many such jobs before and besides I think that I write pretty good. Since this job entales writing, I think that I would make a good candidate because I write good. Also, my Dad had this job a few years back and he didn't like it much, so he didn't finish some stuff that he left undone. I can go do it for him.

Check Spelling

To make such a button, we must:

- Create the button using the Form tool in *Acrobat.*
- Attach an *Execute Menu Item* action to the button's *Mouse Up* event; the menu item in this case is, of course, *Tools>Spelling>Check...*

I am going to assume you know generally how to make a button in Acrobat. However, let's look in detail at the steps involved in attaching the Spell Checking action to a button.

Next Page ->

Attaching the Action

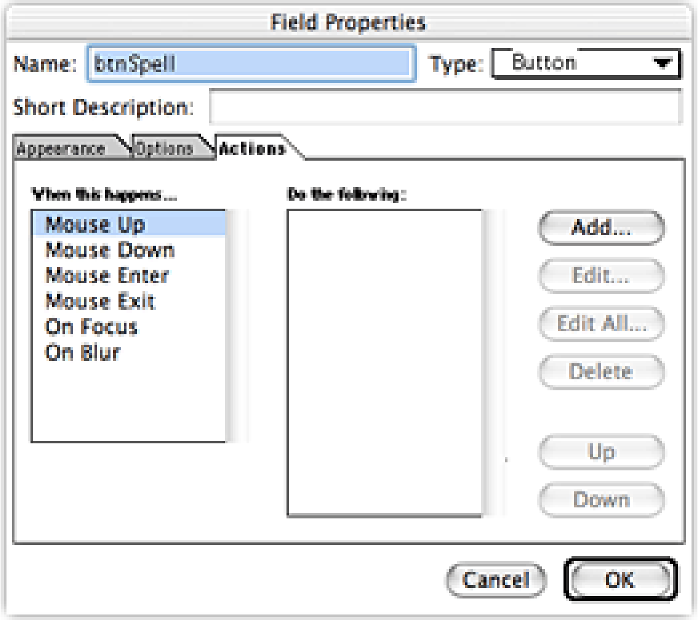
Step by Step

Start with the form open to the page that contains your button and the Form tool selected in the *Acrobat* toolbar.

1. Double-click on the button with the Form tool; Acrobat will present you with the Field Properties dialog box (at right). Click on the *Actions* panel.

2. In the *Actions* panel, select the *Mouse Up* event in the “When this happens” list and click on the *Add* button.

   Acrobat will present you with the *Add an Action* dialog box (next page).



Name: btnSpell

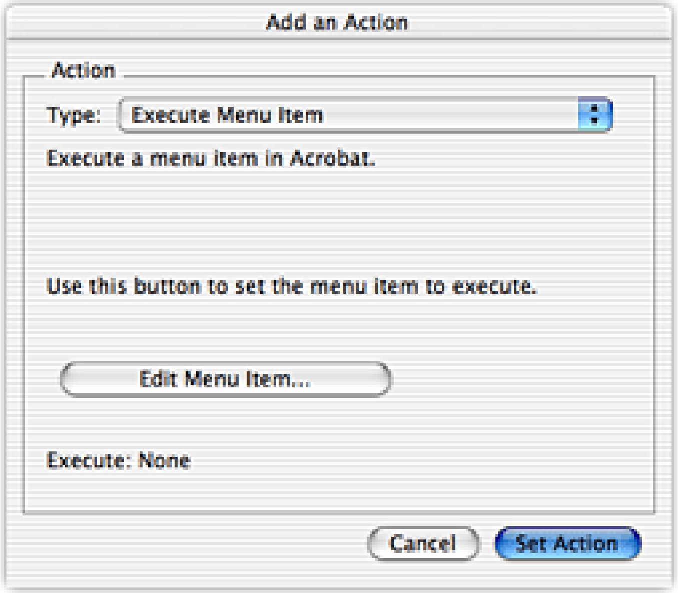
Type: Button

Next Page ->

3. In the *Add an Action dialog box,* select *Execute Menu Item* from the pop-up menu.

4. Click on the *Edit Menu Item* button.

   One of two things will happen, depending on whether you are on a Mac or a Windows machine.



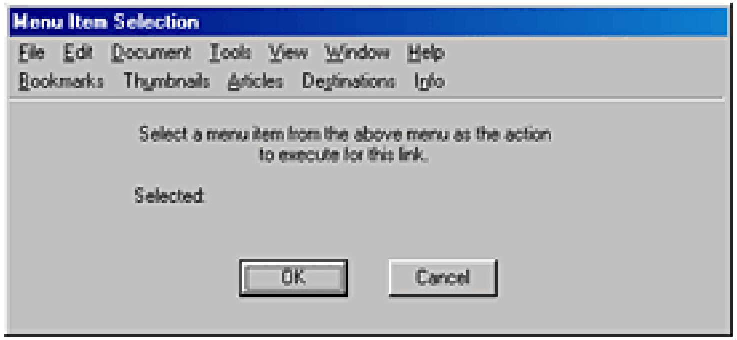
Type: Execute Menu Item

Execute a menu item in Acrobat.

Use this button to set the menu item to execute.

Execute: None

   - On the Mac, *Acrobat* will ask you to select a item from the *Acrobat* menu bar at the top of the screen. All of the Acrobat menu items will be selectable, although they won't carry out their usual activities; Acrobat will simply note which item you selected.
   - In Windows, *Acrobat* will present you with a new dialog box containing all of the normal *Acrobat* menus, as at right.



Select a menu item from the above menu as the action to execute for this link.

Selected:

5. In either case, select *Tools>Spelling>Check…*

6. Exit out of all dialog boxes until you are back at the Acrobat page.

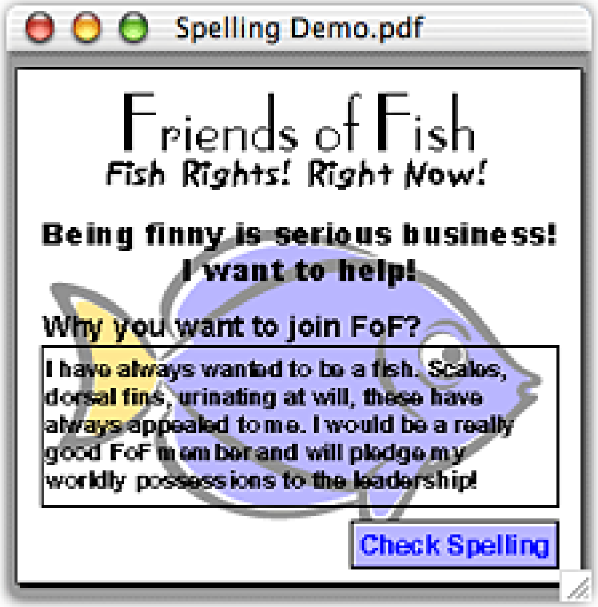
Next Page ->

You Try It

On the Acumen Training *resources* page, you will find a sample *Acrobat* file named *Spelling Demo.pdf* that contains a text field and a button, as at right. The button has no actions associated with it; you can turn it into a *Check Spelling* button by following the numbered steps, above.

Try it!

Next Page ->



Friends of Fish

Fish Rights! Right Now!

Being finny is serious business! I want to help!

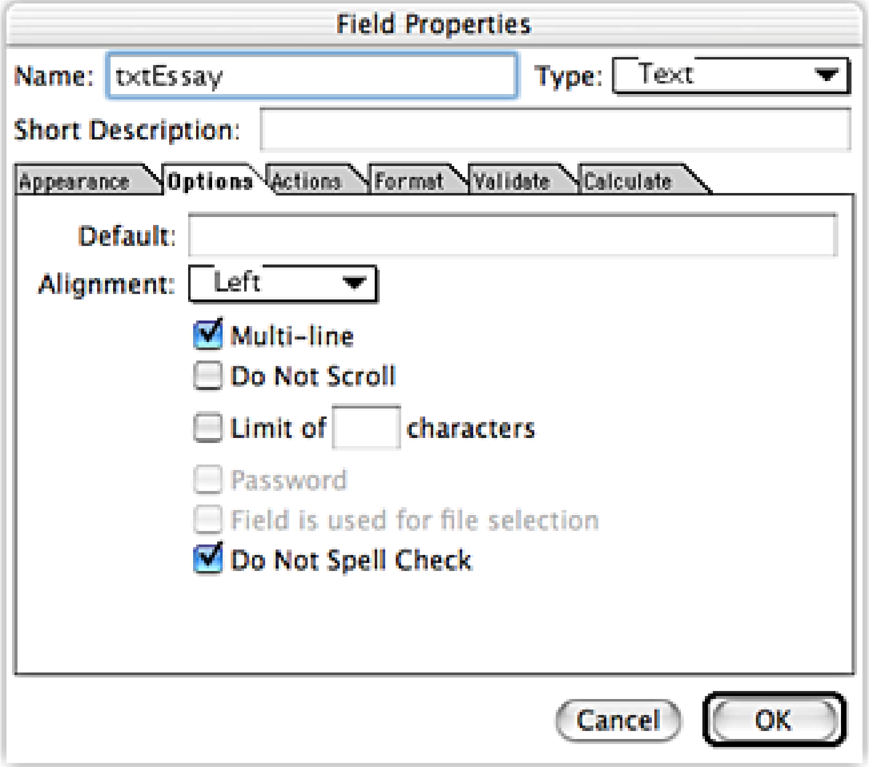
Why you want to join FoF?

I have always wanted to be a fish. Scales, dorsal fins, urinating at will, these have always appealed to me. I would be a really good FoF member and will pledge my worldly possessions to the leadership!

Preventing Spell Checking

Finally, you can tell *Acrobat* that a particular text field should *not* be checked for spelling; this is among the controls in the field’s *Field Properties* dialog box, as at right.

A Text field’s *Options* panel contains a check box labeled *Do Not Spell Check* (toward the bottom of the panel). If this check box is selected, the *Acrobat* spelling checker will ignore that text field. This is appropriate for a great many cases where the text in the text field is expected to contain words that a checker couldn’t know: addresses, numeric fields, passwords, etc.

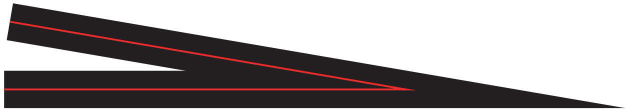


Name: txtEssay
Type: Text
Short Description:
Default:
Alignment: Left
☑ Multi-line
☐ Do Not Scroll
☐ Limit of characters
☐ Password
☐ Field is used for file selection
☑ Do Not Spell Check

Spelling checkers are a ubiquitous part of the manipulation and managing of text in the computer world. Any *Acrobat* form that asks the user for blocks of text really should provide access to the Acrobat spell-checking feature.

Return to Main Menu

The Miter Limit



The miter limit is a PostScript parameter that controls one of the details of how stroked lines are to be rendered. Its purpose is to limit the length of the spike that can result from a mitered line join at a corner in a stroked line. All in all, the miter limit is a fairly exotic parameter, not something you would think you'd ever need to (or want to) pay attention to when writing a PostScript program.

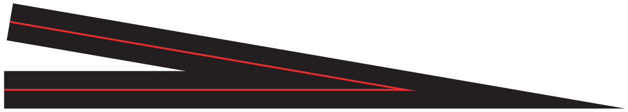
Indeed, most of the time in your PostScript programming life, you *will* ignore the miter limit. Occasionally, however, you will stumble upon an instance where a mitered join is being a problem and you will need to modify this parameter.

How, you may ask, can a mitered join be a problem?

This month, we shall discuss the what, exactly, the miter limit is and see how modifying it solved a mysterious PostScript problem in my own early PostScript experience.

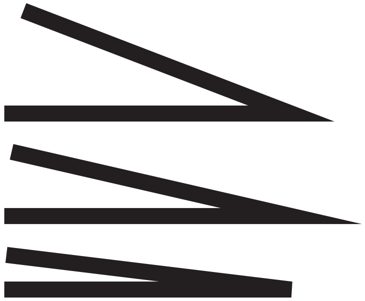
Next Page ->

Mitered Joins



By default, the PostScript *stroke* operator paints corners in the current path with a mitered line join; this is the standard, "pointy" line join we all know and love.

However, as the angle of a corner becomes increasingly acute, the mitered line join begins to look increasingly like a spike that extends a very long distance beyond the actual corner point in the path.

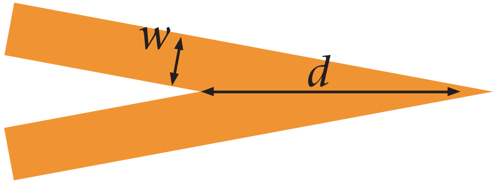


To fix this problem, PostScript automatically switches from mitering to beveling (a flat-looking line join) when the angle at the corner point becomes sufficiently acute. You can see this in the illustration at right: in the top two paths, the angle is obtuse enough that we get a mitered join; in the bottom path, the angle has become acute to a degree that PostScript has decided to bevel that line join.

Next Page ->

The Miter Limit

The angle at which PostScript automatically switches from mitering to beveling a line join is determined by a value called the *miter limit.* The miter limit is the maximum allowed value of ratio of the diagonal across a mitered join to the line width. That is, in the illustration at right, the miter limit is the maximum allowed value for *d/w.*



The default value of the miter limit is *10,* so PostScript switches to beveling when the diagonal across the mitered join would have been more than 10 times the line width.

setmiterlimit

The *setmiterlimit* operator lets you change the miter limit to whatever you wish. It takes a number from the stack and makes that number the current miter limit:

```
5 setmiterlimit
```

The above line of PostScript will make future stroked lines switch to beveling at a more obtuse angle than the default; you will get more beveling, fewer spikes.

Setting the miter limit to 1 effectively turns off mitering (since *d/w* goes to *1* as the angle at the corner goes to 180°).

The miter limit is part of the graphics state, so anything you do with *setmiterlimit* will be subject to *gsave* and *grestore.*

Next Page ->

The Story of "Super-Bold"

Now for an example of why *setmiterlimit* is ever useful.

Back in the mid-80's, after leaving Adobe Systems, I became one of the partners (the "PostScript Guy") in a start-up company that was selling a wysiwyg design system built around a "Fat Mac" (with a whole megabyte of memory!) and a PostScript printer. This system was aimed at sign makers, who at the time were still using rub-on letters for their typesetting.

One significant problem we had in creating a design system for sign makers is that standard Bold fonts are too light for signage; unfortunately, at the time, there were no Heavy or Black fonts available for PostScript. So, we created a style called "Super-Bold."

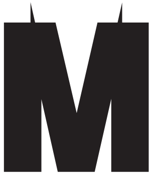
Super-Bold text was both printed with *show* and the charpath stroked with a line of some appropriate thickness. A *supershow* procedure would look something like this:

Helvetica-Bold

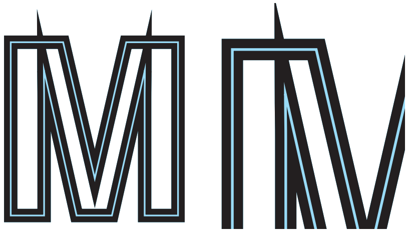
Helvetica-SuperBold

```
/supershow      % string => ---
{ dup gsave false charpath stroke grestore show } bind def
```

Pretty easy and quite a popular feature, but for one mystery: the upper case "M" in Helvetica-SuperBold grew horns! As you can see at right, these horns made the character quite unusable and were alarming to some of our more superstitious customers. (A few exorcisms were reported.)

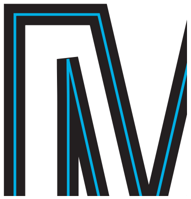


Next Page ->



You can no doubt see what was happening, but it took us an embarrassingly long time to figure it out. The capital “M” in Helvetica has unusually high “armpits” on either side; when we stroked the outline of the character, the mitered spike actually poked above the M’s shoulder, giving us the mysterious horn.

The fix was simple, once we figured out the problem: set the miter limit to some number smaller than the default (we chose *5*) so that the outlined character would have beveled corners at the armpits.



Our final definition of *supershow* looked like this:

```
/supershow     % (str) => ---
{    dup
     gsave
          5 setmiterlimit
          false charpath stroke
     grestore
     show
} bind def
```

This gave us a beveled armpit, as in the bottom illustration above; no more horns!

You probably won’t use the *setmiterlimit* operator very much, but in the cases where you need it, it may well be the only solution to your problem.

Return to Main Menu

Schedule of Classes, Apr – Jun 2003

Following are the dates and locations of Acumen Training's upcoming PostScript and Acrobat classes. Clicking on a class name below will take you to the description of that class on the Acumen training website. The Acrobat class schedule is on the next page.

The PostScript classes are taught in Orange County, California and on corporate sites world-wide. See the Acumen Training web site for more information.

PostScript Classes

PostScript Foundations	Mar 24–28	May 19–23	
New! **Variable Data PostScript**	Apr 28–May 2		Jun 16–20
Advanced PostScript		May 26–30	
PostScript for Support Engineers	Apr 14–18		
Jaws Development		*On-site only*	

PostScript Course Fees PostScript classes cost $2,000 per student.

On-Site Classes These classes may also be taught on your organization's site. Go to www.acumentraining.com/on-site.html for more information.

Registration Info →
Acrobat Classes →

Acrobat Class Schedule

These classes are taught quarterly in Costa Mesa, California, and on corporate sites. Clicking on a course name below will take you to the class description on the Acumen Training web site.

Acrobat Essentials	*No Acrobat classes scheduled for this quarter.* See the Acumen Training website regarding setting up an on-site class.
Interactive Acrobat	
Creating Acrobat Forms	
Troubleshooting with Enfocus' PitStop	
Acrobat Class Fees	*Acrobat Essentials* and *Creating Acrobat Forms* ($\frac{1}{2}$-day each) cost $180.00 or $340.00 for both classes. *Troubleshooting With PitStop* (full day) is $340.00. In all cases, there is a 10% discount if three or more people from the same organization sign up for the same class.

Registration ->

Return to Main Menu

Contacting John Deubert at Acumen Training

For more information

For class descriptions, on-site arrangements or any other information about Acumen’s classes:

Web site: http://www.acumentraining.com **email:** john@acumentraining.com

telephone: 949-248-1241

mail: 25142 Danalaurel, Dana Point, CA 92629

Registering for Classes

To register for an Acumen Training class, contact John any of the following ways:

Register On-line: http://www.acumentraining.com/registration.html

email: registration@acumentraining.com

telephone: 949-248-1241

mail: 25142 Danalaurel, Dana Point, CA 92629

Back issues

Back issues of the Acumen Journal are available at the Acumen Training website: www.acumenjournal.com/AcumenJournal.html

Return to First Page

What’s New at Acumen Training?

Introducing Variable Data PostScript

April 28 will see the introduction of a new course at Acumen Training: **Variable Data PostScript,** a five-day, hands-on class in producing variable data documents using the PostScript language. This class discusses everything from reading and parsing a data stream, to drawing charts and setting text, to maximizing printing speed using forms and compressed PostScript.

This is a hands-on class, with student exercises giving practice in all the major topics throughout the week. The student comes away with a wealth of information regarding advanced PostScript capabilities and specific techniques for generating documents from a data stream.

The class consists of two parts, which may be taken separately:

Advanced PostScript Features, teaching the student about PostScript level 2 and Level 3 features that are important to variable data printing. This part may be skipped by students who have taken the one-week *Advanced PostScript* course.

Variable Data Programming and Techniques, which applies the students' PostScript knowledge to producing documents from a data stream. Students who have taken *Advanced Postscript* may take this part of the course by itself.

For a complete course description and topic list, visit the Acumen Training website.

Return to First Page

Journal Feedback

If you have any comments regarding the *Acumen Journal,* please let me know. In particular, I am looking for three types of information:

Comments on usefulness. Does the Journal provide you with worthwhile information? Was it well written and understandable? Do you like it, hate it? Does it somehow make you want to brush your teeth with an electric sander?

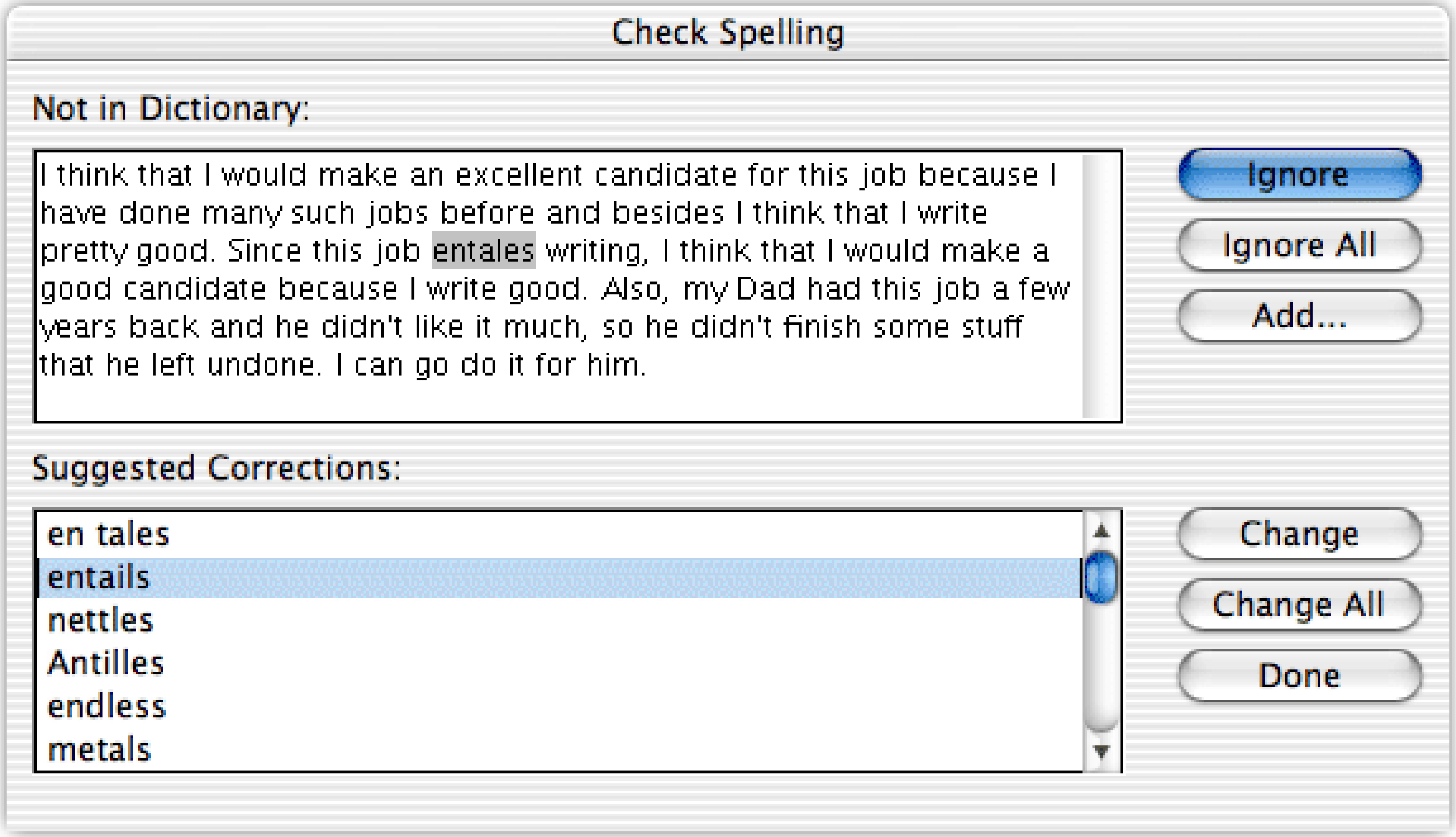
Suggestions for articles. Each Journal issue contains one article each on PostScript and Acrobat. What topics would you like me to write about?

Questions and Answers. Do you have any questions about Acrobat, PDF or PostScript? Feel free to email me about. I'll answer your question if I can. (If enough people ask the same question, I can turn it into a Journal article.)

Please send any comments, questions, or problems to:

journal@acumentraining.com

Return to Menu



Check Spelling

Not in Dictionary:

I think that I would make an excellent candidate for this job because I have done many such jobs before and besides I think that I write pretty good. Since this job entales writing, I think that I would make a good candidate because I write good. Also, my Dad had this job a few years back and he didn't like it much, so he didn't finish some stuff that he left undone. I can go do it for him.

Suggested Corrections:

- en tales
- entails
- nettles
- Antilles
- endless
- metals