

Enhancing Documents by Customizing Annotations

Gary Staas

PDF Dream

www.pdfdream.com

Biography

For over 20 years, Gary Staas has developed software for major corporations, such as Hewlett-Packard and Tandem Computers. He has led and managed software teams to complete and bring products to market.

Most recently he worked at Adobe Systems for 4 years, helping software developers use the Acrobat APIs to write plug-ins and integrate their applications with Acrobat. In addition to direct support, Gary wrote much of the Acrobat Software Development Kit documentation for Acrobat 3.0 and 4.0. He also developed and taught training classes in using the Acrobat API to developers at Adobe's corporate offices as well as other cities in the US and Europe.

He has published articles in a variety of technical publications.

Gary started PDF Dream to provide in depth support for those wishing to learn the ways of the Acrobat API so they can fully utilize Acrobat and PDF.

Outline

What are annotations?

Why are they useful?

Customizing—

- Existing annotations
- New annotation types

What's useful to know about annotations?

What can you do with annotations?

What are annotations?

Items that add information to a document

Have an *appearance*

Part of page—but distinct from *page contents*

Not part of page contents streams

Add capability to a document

Field annotations in Acrobat Forms, for instance

Typically added with a tool

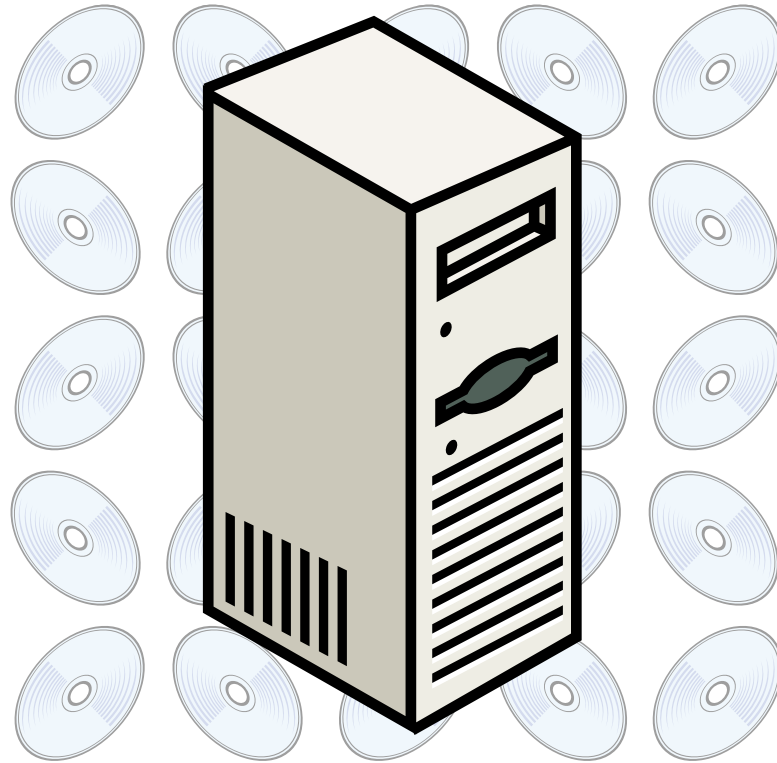
- Note tool, markup tools...

Standard annotations built-in to Acrobat

- Pencil, note, mark up, sound, movie...



Data structure with a face



Annotation appearance

Can be anything!

Represented by PDF drawing operators, similar to PostScript

Changeable

- Select appropriate appearance for circumstances

Visible or invisible

Printable or not printable

- Printed page can look *different* than Acrobat viewer appearance
 - Print stamp over page
 - Prevent printed document being useful to unauthorized person

Annotation appearance

Appearance can be stored in PDF file

- Don't need creating plug-in to view or print custom annotations

Acrobat form fields

Appearance can be exported to FDF file

Why are annotations useful?

Substantial Acrobat viewer support

Beefed up for Acrobat forms

First class PDF citizens

Defined in *PDF Reference Manual*

Flexible

- Contain any data you want
- Look any way you want

Dynamic—can change while viewing

User interaction rich

- Acrobat forms
- Change appearance appropriately
 - Acrobat form field shows data entered

Why are annotations useful?

Customizable

- Enhance existing annotations
- Custom annotation types

Security—can keep user from adding or changing annotations

Make documents come alive

Acrobat viewer support

Annotations well integrated—really part of Acrobat

Viewing annotations

- Appearance drawn automatically by Acrobat viewer

Manipulating annotations

- Add and delete
- Move by dragging
- Custom interaction

Managing annotations

Handling annotations

Annotation handler in Acrobat API

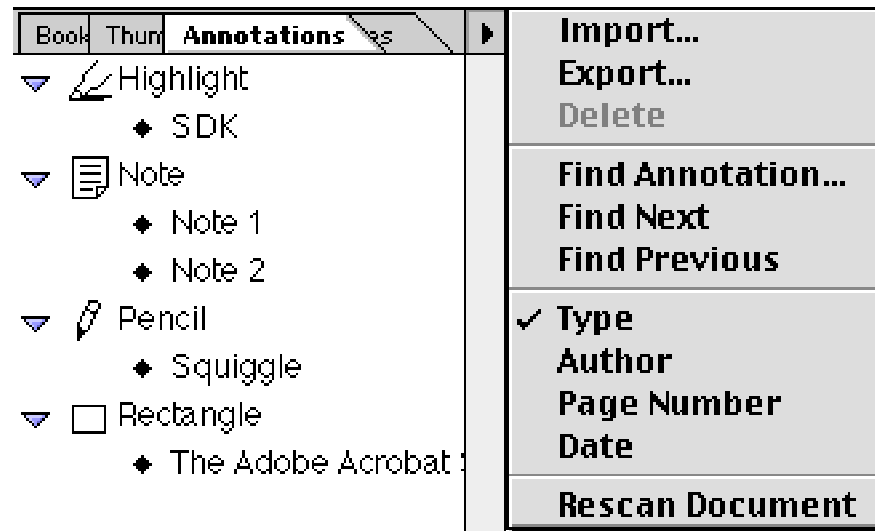
Managing annotations

Annotations pane

- List and sort annotations
 - type, author, page #, date
- Search for text in annotations

Export and import annotations in FDF

- All data, including appearances



Can manage custom annotations in future

Add a few functions to annotation handler

Handling annotations

Acrobat has *annotation handler* for every annotation type

Default annotation handler for standard annotations

Create new annotation type by defining new annotation handler in plug-in

What exactly are annotations?

Entries in annotation array for a page

[annot₁
annot₂
annot₃
...
annot_n]

Dictionary with an appearance

Annotation dictionary

Dictionary is set of key-value pairs

- Key is a *name*
 - /name
- Value is any PDF object

Objects can be names, strings, arrays, other dictionaries!
- Inherently extensible

Example: text annotation

```
<<  
/Type /Annot  
/Subtype /Text  
/Rect [266 116 430 204]  
/Contents (text for two)  
>>
```

Annotation appearance

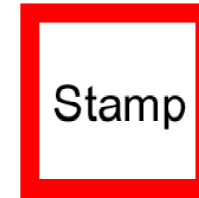
Appearances for standard types built in

Text, link, attached file, markup...

Custom types

- Must define appearance

Stamper annotation sample



Appearance keys in annotation dictionary

Added for Acrobat forms

Appearance is *Form XObject*

- Encapsulates set of drawing operations
- Uses standard PDF page drawing operators
 - Any combination of text, graphics, images
 - Could make it look like a PDF page

Form field button appearance

Annotation appearance

Can build appearance on the fly with
Acrobat API

- Create list of drawing operators
- Use *PDFEdit* API

Treats appearance as list of objects with attributes. Could draw a colored box with text of a chosen font inside—or arbitrarily complex construct.

Enhancing existing annotations

Plug-ins can manipulate standard annotations

Acrobat API functions

- Change text and link annotations
PDTextAnnotSetOpen
- Modify annotations with get/set attribute functions
 - Change annotation's dictionary
PDAnnotGetColor, PDAnnotSetColor

Modifying existing annotation behavior

Default annotation handler for standard annotations

Can modify these handlers

AVAppGetAnnotHandlerByName function gets annot handler for an annotation type

Example: Link annotation handler

- Change to “snap link” behavior
- Goes to link destination as long as mouse button pressed—then snaps back to previous view

Annotation dictionaries

PDF Reference Manual

Section 6.6, "Annotations"

Predefined key-value pairs

- Define an annotation's attributes
- Understood and interpreted by viewer
- Often apparent
 - Color
 - Border

Required keys

Type

- *Always Annot*

Subtype

- Each type has unique name
 - Text
 - Link
 - Square, Circle
 - StrikeOut, Highlight, Underline

Rect

- Position on page

User space coordinates—view invariant

Optional keys

T (Title)

M (ModDate)

AP (Appearance)

F (Flags)

Apply to all annotations

- Hidden
- Print
- Read-only

Some keys used only for certain
annotation types

- *Contents*—Text annotations

...and others

Custom Keys

Define your own keys

- For both standard and custom annotations
- Use appropriate prefix for key to avoid collisions with other annotations
- Value can be any object
 - Dictionary
 - Stream (images)
 - ...

If Acrobat doesn't know about a key, it ignores it

- Freedom to add new keys
- Can't redefine existing keys

Custom Keys

Can represent arbitrary data

Any data structure can be used for value of key

Could represent a C struct or C++ class in a dictionary with the keys being the instance variable field names in struct or class

Annotation handler

Defines annotation behavior

AVAnnotHandler struct

List of functions

Functions called automatically in response to events—

- mouse click, key press...

Functions determine what happens when—

- you click on annotation
- move cursor
- type keyboard data

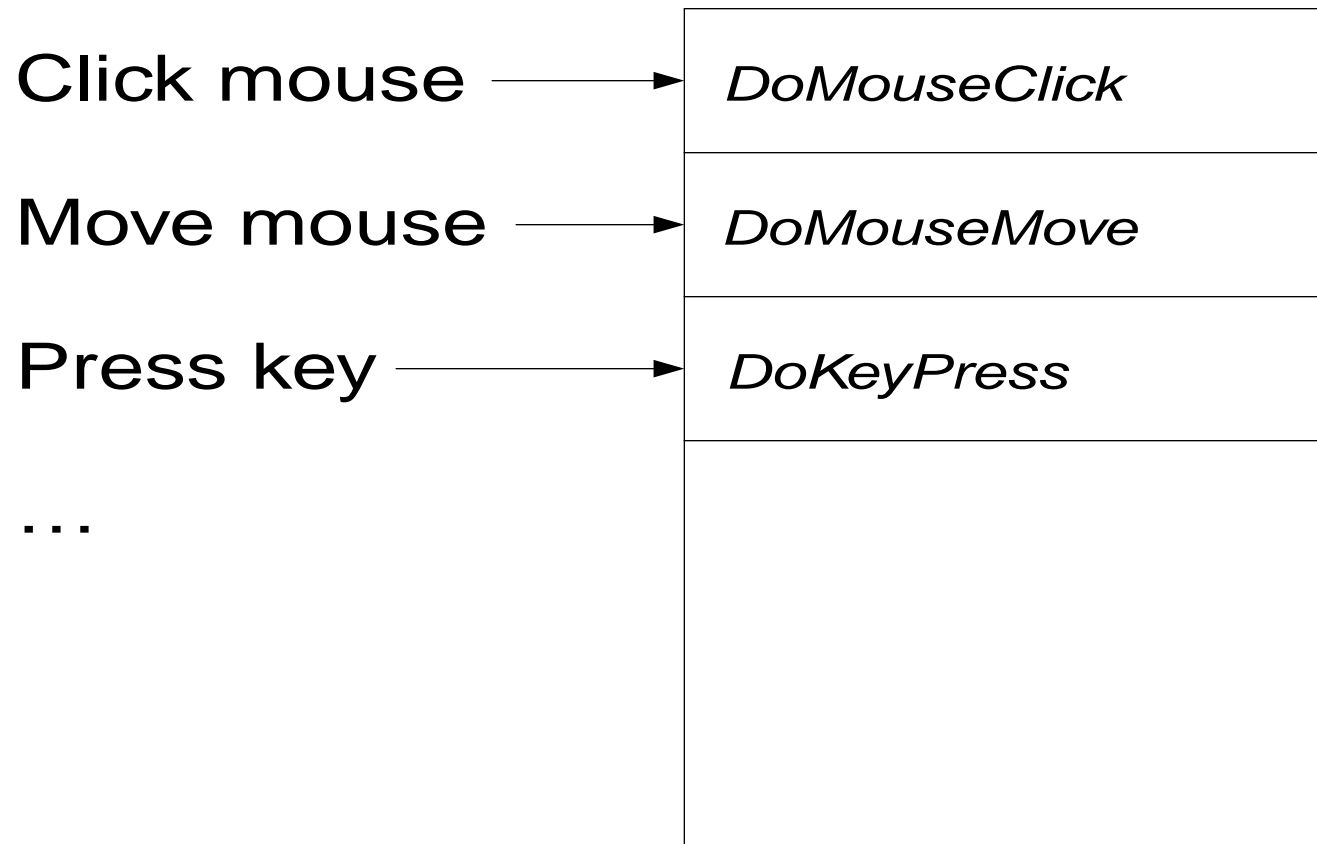
Defines other characteristics

- “Range of influence”

Annotation handler

Event

Function table



Plug-in

Plug-in needed to add or change annotation handler

Plug-in is a *program* that adds capability to Acrobat by using Acrobat API

Annotation's behavior determined by the program

- Design code to make annotations do what you want
- Great flexibility

Can repurpose existing software

Annotations as Database

Set of annotations is a database

- Key-value pairs

Individual annotation is database record

Record could also be set of annotations on a page, each with a key, as in Acrobat Forms

Could search database

Visual representation of database

- Plotting data
- Interact with annotations
 - move points
 - change colors or other attributes
 - change attributes with pop up menu

Integrate with existing database software
in annotation handler

Forms annotations

Example of complex annotation to show what can be done

Forms plug-in defines *Widget* annotation type

User interface for creating annotations and setting annotation attributes

Sophisticated interaction with user

- Data entry
- Type checking
- Appearance definition
- Javascripts to customize field action
 - Adds Javascript interpreter—existing software

Summary

Add information and value to documents

Annotation can contain any *data* desired

Annotation can have any *appearance* desired

Customize existing annotations

Create *new* annotations

Behavior only limited to what you can do with a software program

Sample plug-ins

Available from www.pdfdream.com

Further information

PDF Reference Manual

- **Section 6.6, “Annotations”**
Defines keys in annotation dictionary
- **Section 6.14, “Acrobat Forms”**
Defines form field dictionaries

Acrobat SDK

- *Acrobat Core API Overview*
 - “Handlers”
See annotation handler section
- *Acrobat Core API Reference*
 - *AVAppRegisterAnnotHandler* function
Detailed programming information

Enhancing Documents by Customizing Annotations

Copyright 2000 Gary Staas

gstaas@pdfdream.com

www.pdfdream.com

