Acrobat XI

The free guide to

Creating PDF Forms for Government

Learn to:

- Design forms for government
- Use Acrobat's form tools
- Create easy copy/paste JavaScripts
- Host forms on eGovernment websites
- Develop standards & best practices

VIDEO TUTORIALS

ted padova

Determine never to be idle. No person will have occasion to complain of the want of time who never loses any. It is wonderful how much may be done if we are always doing.

—Thomas Jefferson

This book is an effort to influence government offices worldwide on how to properly create forms for processing transactions, host forms on eGovernment websites, and develop some level of consciousness regarding how content in eGovernment should be developed.

This is a free publication and I encourage government offices to circulate and distribute this book in hopes that government workers can gain some knowledge for improving PDF forms hosted on eGovernment websites.

Comments, suggestions and feedback are welcomed.

—ted padova

ted@tedpadova.com

Creating PDF Forms for Government Using Acrobat XI

ted padova
ted@tedpadova.com

Chapter 1

IN THIS CHAPTER

Understanding eGovernment

The term eGovernment ranges considerably in definitions. The most common definition used today is *the use of technology to enhance access and delivery of government services that benefit citizens, partners and employees*. One of the most important uses of technology in this context is the Internet.

Although the use of the Internet, and more importantly the World Wide Web, has been available to the masses as a recent phenomenon since about 1993 when the first .gov websites were established, digital government has been around since the 1980s. Voice mail, computers, interactive cable television and other technologies, for example, were in use by federal and local governments in the USA in the early 1980s.

During the 80s, USA government was involved with experimentation related to broadcasting, transaction processing, public records accessing, computer conferencing and electronic surveillance. However, none of the early experiments amounted to a comprehensive approach that engaged citizens in government. Federal and local governments were developing digital methods and using a myriad of tools to work toward centralizing activities, but no early developments reached the masses.

It wasn't until 1993 during the Clinton Administration that eGovernment became a powerful public symbol when the White House announced the availability of direct mail to both President Bill Clinton and Vice President Al Gore. Both President Clinton and Vice President Gore played a key role in developing the concept of eGovernment that was later adopted by President George W. Bush and further embraced by President Barack Obama.

Throughout the 20 years of administrations of three presidents in the USA from 1992 through 2012, eGovernment services broadened to include everything from constituent advocacy, dissemination of information, transparency, to processing transactions. Efforts in centralizing services enjoyed much success and some malady.

Success was the development of services that engaged people in government in mass society much more than has been achieved in modern government. People could download congressional acts, examine financial records, and obtain information that was either unavailable to the public or much more difficult to obtain. Transparency in government became a hot campaign item for candidate elections in the USA and many legislator hopefuls promised more transparency in government when the votes were tabulated.

Many of these efforts however failed miserably. The failure in the USA Federal Government was the lack of preparation to take advantage of the new technologies. The Internet exploded in the mid-1990s and government departments scurried to upload information in many disjointed ways. There was no infrastructure to coordinate the way information was made available, no standards set forth to comply with, no centralized coordination of departments to monitor and approve consistency between departments. And campaign promises soon fizzled after elections.

USA Presidential Cabinet Offices

Many separate departments in both Federal and State government offices had individuals responsible for IT services, however at the top levels of cabinet secretary positions, many governments in the USA lacked a Chief Technology Officer, Chief Information Officer or department head responsible for IT services who reported to the top executive. The absence of this position resulted in a lack of overall planning and coordination of IT services and ultimately lack of eGovernment preparation for

global government offices such as the USA Federal Government and many individual State governments.

At the federal level out of 15 different cabinet secretary positions there exists no cabinet office in charge of technology through the Barack Obama administration. When President Barack Obama was elected he appointed the first Chief Technology Officer in the USA, however, the CTO for the USA Federal Government is not a cabinet level position.

Some US States have cabinet level authorities reporting directly to the chief executive. As shown in Table 1.1, a little less than half the US States have such a position.

Why is it important for a government to have a cabinet level authority for information technology? As you peruse various government websites you find much disparity in the appearance and content hosted on individual department websites. In many cases a government having separate websites for fifteen different cabinet posts visually looks like websites that could easily be mistaken for fifteen different countries. Without consistency in appearance and design —something that needs control by a high level cabinet authority, constituents have more difficulty navigating a site and locating relevant content.

Table 1.1

State	Cabinet Position	Link Destination
Arkansas	X	Communications Director
California	X	Chief Technology Officer
Colorado	X	Information Officer
Connecticut	X	Information Officer
Delaware	X	Information and Communications Technology
Kentucky	X	Office of Technology
Maryland	X	Information Technology
Michigan	X	Information Technology
Mississippi	X	Information Technology
Nebraska	X	Chief Information Officer
Nevada	X	Information Technology
New Jersey	X	Chief Technology Officer
New Mexico	X	Information Technology
New York	X	Office for Technology
North Dakota	X	Information Technology
Ohio	X	Information Technology
Oklahoma	X	Science and Technology
Pennsylvania	X	Office of Information Technology
South Dakota	X	Information and Telecommunications
Texas	X	Information Resources
Utah	X	Technology Services
Virginia	X	Secretary of Technology
Washington	X	Department of Information Systems
Total	23	
Percent	46%	

US States having cabinet level authority for information technology

Cabinet Technology Positions in the Third World

The annual USA Federal IT budget is over 80 billion dollars. That's more money than the total cost of government for over 75% of the countries in the world. Inasmuch as the USA spends more money on Information Technology than any country in the world, the organization of high-level technology positions in the USA is significantly inferior to many developing countries.

In South East Asia, for example, the Philippines has had one technology specialist in a cabinet post over 50 years. Another cabinet level post was added over 35 years ago —although this post was recently eliminated under the Aquino administration. A small country like the Philippines organizationally in terms of IT is more than 50 years ahead of the USA working with a fraction of the USA IT budget.

Ghana Africa has had a Minister of Information since 1957, Russia has had a Secretary of Science and Technology since Perestroika, the Peoples Republic of China has a Minister of Science and Technology, Botswana Africa has a Minister of Infrastructure, Science and Technology, Thailand has a Minister of Information and Communication Technology, and the list goes on for developing countries throughout the world.

In the USA it wasn't until President Obama was elected in 2008 that a Chief Technology Officer was appointed. Yet this position to date carries no cabinet level authority. The US CTO serves in the US Department of Science and Technology Office under the President's Science Advisor —a much lesser role than a cabinet secretary.

Inasmuch as America is the heart of technology in the world, it has yet to develop a well-organized body of government to handle information and technology other than adjunct duties within existing cabinet offices. Rather than coordinate eGovernment in the USA through a central authority, various branches of government offices determine their own path and host content within departments without central federal organization.

What are the Responsibilities of eGovernment?

In terms of *enhancing access and delivery of government services*, eGovernment uses the World Wide Web to host content that engages citizens in government. Web hosting for eGovernment provides several resources designed to help citizens interact with their governments. A non-inclusive list of resources can be described as:

- **Events Reporting.** Various laws, resolutions, achievements, and problems can be reported instantly online for constituents to review.
- **Disseminating Information.** Everything from the structure of government, how it operates, the history of government, access to libraries of archives, information related to

education, health, economy, and more are part of most eGovernment websites.

- **Transactions.** Performing transactions either via online forms processing or downloading forms for applications and processes are part of all eGovernment websites throughout the world. Transactions, in this regard, are any means of collecting information on a form (HTML or PDF).
- **Constituent Reporting.** eGovernment provides opportunities for constituents to communicate directly with government in free societies. People can submit ideas, remark on laws and resolutions, provide opinions, rank and rate government and elected officials, and more. This option engages citizens in a two-way communication with government.
- **Marketing.** Government can market the communities they serve to help increase revenue and business. Departments of tourism can promote visitations to areas of recreation and interest, business climate can be communicated to attract new business, and economic reporting can provide information related to the advantages of setting up business in a given region.
- **Health Services.** Reporting specifically on health related issues is a common item shared by many eGovernment sites.
- **Employment.** Also shared by many eGovernment websites is information related to a service area job market, employment services, and other job related issues.

The above items remain part of a non-inclusive list but suggest some of the more common interests shared by eGovernment websites. There are more specific types of information that eGovernment is concerned with and it varies depending on the government entity providing the information. As a matter of rule though, the mission of eGovernment is concerned with three main categories that include:

- **Disseminating information.** Whether it be the structure of government, the economy, education, reporting events, protecting citizens against terrorism and natural disaster, etc. these tasks fall under the umbrella of providing information to visitors of a government website.
- **Transactions**. Both online and downloadable forms for processing transactions is a critical part of all eGovernment websites.
- **Two-way communications.** In free societies, engaging citizens in government via opportunities to communicate directly with government officials is an important inclusion for eGovernment.

Chapter 2

Measuring the Quality of Content Hosted by eGovernment

The quality of the content displayed on eGovernment websites is critically important for communicating a positive image.

Chapter 2

IN THIS CHAPTER

Browse any government website and you find information readily available covering a huge array of different topics. If we were to rate the USA Federal and States eGovernment websites according to a single criterion for providing vast information, every state and the USA Federal Government would receive an A+ mark. These entities provide information related to everything from lost pets to pertinent information that helps constituents work with government agencies.

Measuring the Quality of Content Hosted by eGovernment

Peel away the top veneer of the Home page on any US State Government or the USA Federal Government websites and examine the quality of the content, the ease of accessing it, and the value to the constituent and you find many eGovernment websites receiving a failing mark when rating the value and accessibility of the content.

Information not only needs to be made available to constituents, they also need to easily locate documents. Spending hours poking around a website for pertinent information is definitely not the best way to engage citizens in government. It's nice to know the name of a Governor's dog appearing as the first visual on a website, but will this information serve constituents better than a link to a form for completing a transaction?

Hosting Content

Quality of content on eGovernment websites is rarely evaluated by publications and societies that perform evaluations and rate websites. The appearance of a website, navigation links, use of Web 2.0 technologies such as linking to Facebook, Twitter, YouTube, and other social media sites seems to be of most interest to evaluators. However, it's the content one obtains from an eGovernment website that is beneficial to the constituents. Content needs to be clear, well presented, and appealing for visitors.

Examining US State Organization Charts

As part of my research I sought to obtain an organization chart for each of the 50 USA states. The structure, reporting, and information about cabinet positions and commissions I consider a valuable asset for constituents searching information within given US State offices. Knowing who to contact, what areas of responsibilities are charged to a given office, how the reporting hierarchy is established, how to contact the department, and similar information can help constituents save time in contacting the right office for assistance. A well-designed organization chart is one of the best tools people can use to obtain this type of information.

To conduct the research I limited my searches to Internet websites. Since we're talking about eGovernment, I thought it best to stay within the framework of what's available online and how to find information using search engines both external and internal for each eGovernment website.

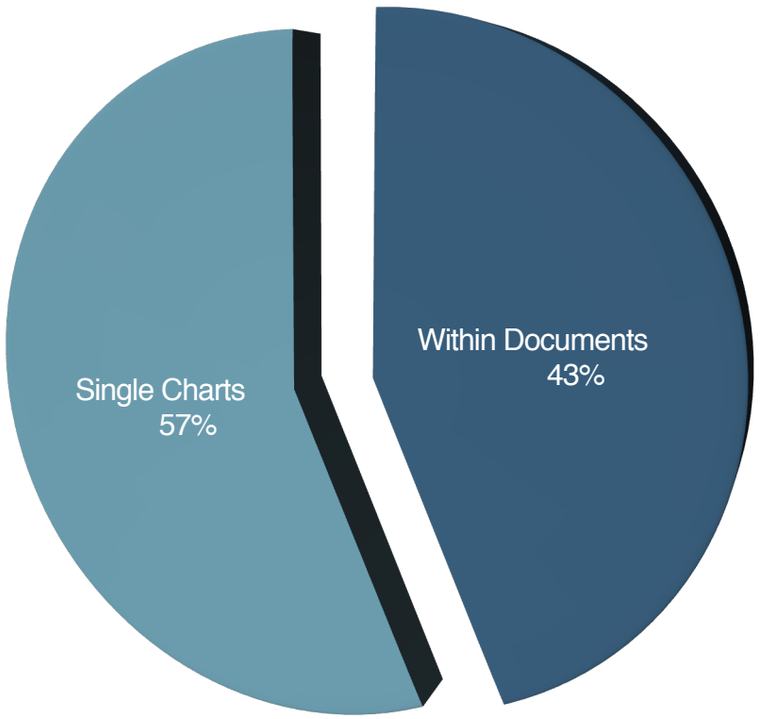
I used the same search criteria for locating organization charts among the 50 USA States. The first search was performed on the individual state's websites. The second search was performed in Google. I used keywords such as *org chart, organization chart, executive branch, state names*, and made several different

searches for each state using different combinations of the search criteria. When I found no results for a given organization chart, I expanded the search and sorted though various documents hosted on the states' websites. In some cases organization charts were buried in documents such as *annual reports, studies, and research documents*. Rather than skip a state, I did my best to find at least one organization chart for every state.

Rather than use information on 3rd party websites that collect government data and report information about the hierarchy within states' governments, I restricted the search to what can be found using search engines and navigating links for each state within a given government website. In other words, I didn't use 3rd party reporting to obtain the charts but was more interested in what the eGovernment sites offered their constituents. There actually exists websites that provide government organization charts but the downloads can cost as much as US $99.00. Imagine, websites are making money on information that should be made readily available by government to constituents.

The results of my study are shown in Table 2.1 for the multiple searches. The first and second columns indicate whether the organization chart was easily found as a single document or some navigation was needed to find an organization chart buried within another document. The last columns rate the quality of the organization charts that I'll address later.

The results in Table 2.1 show that an organization chart does exist for 46 of the 50 US States. After exhausting long periods of searching the Internet I couldn't find an organization chart for the states of Missouri, Oklahoma, Rhode Island, and Texas. This is not to say that organization charts for these four states don't exist. There may be a chart in some form on the websites. I just gave up after spending much time searching the states and finding no results. If organization charts do exist for these states, they certainly are not easily found by constituents.



Of those states providing an organization chart for state government, 57% hosted the chart as an individual document and they were easily found. The remaining charts (43%) were contained in a document that may or may not be the official

State	Single Chart	Within Doc	Rating Scale 1	2	3	4	5
Alabama		X				4	
Alaska	X				3		
Arizona		X				4	
Arkansas	X				3		
California	X				3		
Colorado	X		1				
Connecticut	X						5
Delaware		X				4	
Florida	X			2			
Georgia	X				3		
Hawaii		X					5
Idaho		X			3		
Illinois	X			2			
Indiana		X				4	
Iowa	X					4	
Kansas	X						5
Kentucky		X			3		
Louisiana						4	
Maine	X				3		
Maryland	X					4	
Massachusetts	X						5
Michigan	X					4	
Minnesota		X		2			
Mississippi		X				4	
Montana		X				4	
Nebraska	X					4	
Nevada	X				3		
New Hampshire		X				4	
New Jersey		X					5
New Mexico	X					4	
New York		X				4	
North Carolina	X			2			
North Dakota		X				4	
Ohio		X				4	
Oregon	X					4	
Pennsylvania	X					4	
South Carolina	X				3		
South Dakota		X				4	
Tennessee		X				4	
Utah		X					5
Vermont	X					4	
Virginia	X				3		
Washington	X				3		
West Virginia	X					4	
Wisconsin	X						5
Wyoming		X					5
Total	**26**	**20**	**1**	**4**	**11**	**22**	**8**
Percent	**57**	**43**	**➤ 1**	**1**	**24**	**49**	**17**

organization chart for the state. Some of the documents containing organization charts were prepared by librarians, office clerks, and different department personnel, therefore, some of these charts may have been created specifically for the documents I downloaded.

Quality can be measured in many ways. We can look at the viability of the content, whether it meets the needs of the constituent in terms of valuable data and related to the searches one performs to find it, and we can also include how well the information is presented in terms of design and appearance.

The Quality columns in Table 2.1 contain rating values 1 through 5. A 1 rating represents the highest mark for a well-designed chart, sufficient with clarity, and easy to locate.

I developed a personal rating scale so the evaluation results are subjective and clearly my own. Another evaluator might rate the quality of the charts differently however many would agree that those charts on the lower end of the scale belong in the lower end and those on the higher end of the scale would likely be agreed upon by different evaluators as worthy of the highest ratings. Take the State of Florida, for example. I rated this chart 2. The chart is clear and attractive however the state seal is not present on the chart. Some evaluators might give Florida a number 1 rating. I rated Florida 2 because I believe documents such as organization charts need the state seal in clear view.

On the other side of the scale, I evaluated the State of Maine with a 3. I consider a 3 rating not average but substandard. Maine has the state seal on the organization chart but the design is horrible and doesn't appear as a professional document. Again, some evaluators might rate Maine differently, however I would expect that no evaluator would place this chart at the top to suggest the design is exemplary.

A number 1 mark was the highest mark and *only one state out of the 46 states hosting charts received the this rating*. The organization chart from the State of Colorado is an appealing design, contains the state seal, and presents the various government offices in an attractive framework.

Colorado Organization Chart:

You can find the Colorado organization chart at:

http://www.colorado.gov/cs/Satellite?blobcol=urldata&blobheader=application%2Fpdf&blobheadername1=Content-Disposition&blobheadername2=MDT-Type&blobheadervalue1=inline%3B+filename%3D673%2F183%2FColorado+State+Org+Chart-2007-2008%2C0.pdf&blobheadervalue2=abinary%3B+charset%3DUTF-8&blobkey=id&blobtable=MungoBlobs&blobwhere=1167364306156&ssbinary=true

Most of the states receiving a number 2 mark were close in design to Colorado but they were not designed quite as well as the highest rated state. The state of Illinois has a very attractive organization chart and well conceived. However it's missing the state seal —a graphic element I believe is essential on all official

state documents. Both the states of Minnesota and North Carolina organization charts have the state seal placed on the document but the design is not quite as appealing as Colorado, Illinois, and Florida. Once again, some evaluators might suggest that the five top charts should all receive a 1 mark so this subjective evaluation could be argued. However, I suspect that most would agree that, of the 46 organization charts hosted by states, the states of Colorado, Florida, Illinois, North Carolina, and Minnesota are appropriately placed at the top.

A 3 rating was imposed for a poor design —no visual appeal and not the kind of clarity you find in the top 5 rated states. As you move to a 4 and 5 rating the charts get worse in terms of appearance with 5 representing designs that are clearly unprofessional and somewhat of an embarrassment.

Some of the charts rated at the low end were scanned documents —and poor scans at that. Some used graphic box elements like one might imagine coming from an elementary school student and some were incomplete. It should be pointed out however that with the exception of Massachusetts the number 5 rated charts were contained within various government publications. The charts evaluated may not be the official charts of the states but rather designs assembled by office workers. The Massachusetts organization chart was an HTML file so it differs quite a bit from the other PDF documents.

Examining USA Federal Government Organization Charts

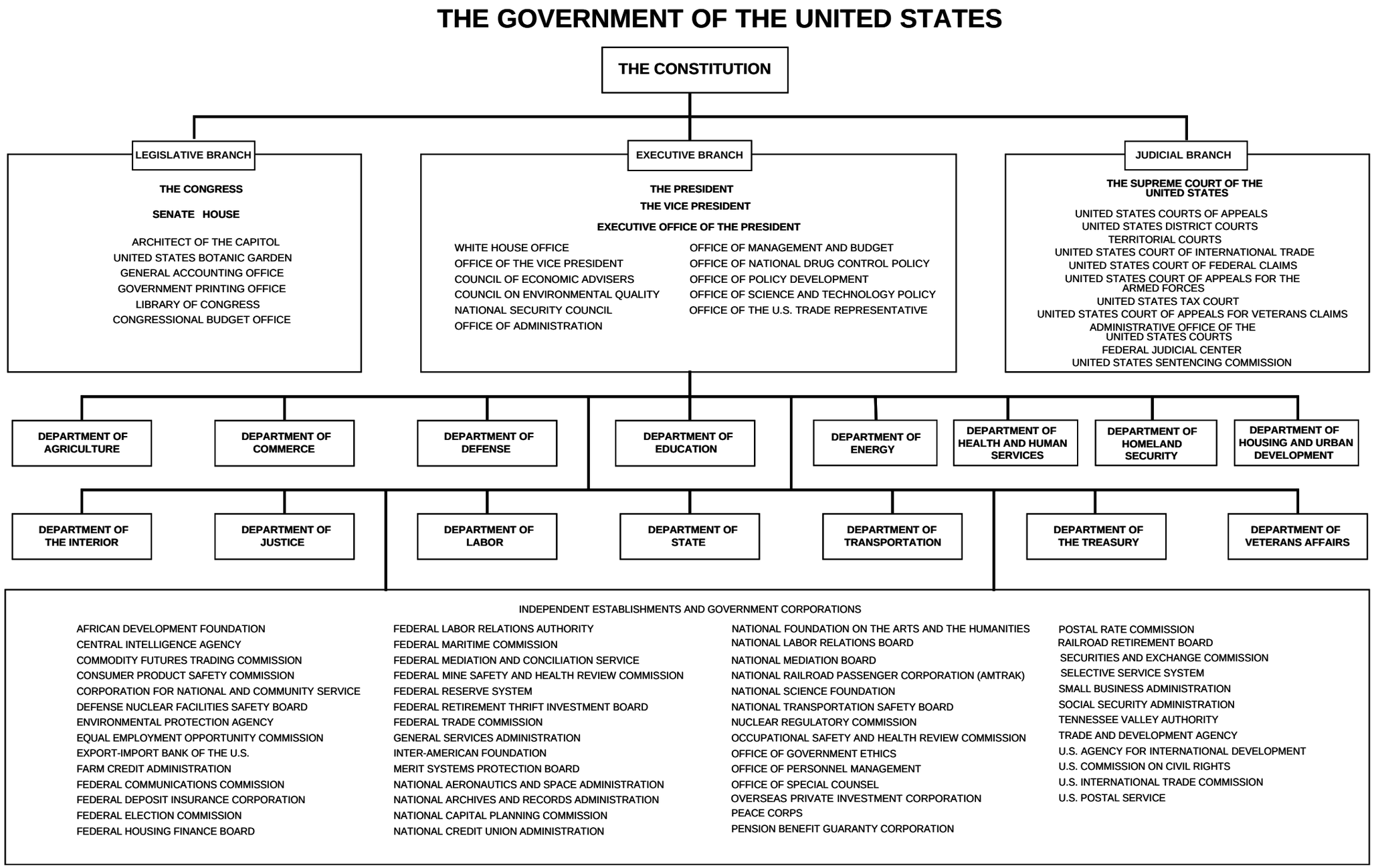
Organization charts for the USA Federal Government and each of the cabinet offices reporting to the President of the United States are equally as boring and unappealing as many of the US State organization charts that were rated 3 and below.

The charts do reflect the structure of the agencies, but the visual appeal is uninteresting and elementary *(see the gallery on the following pages)*.

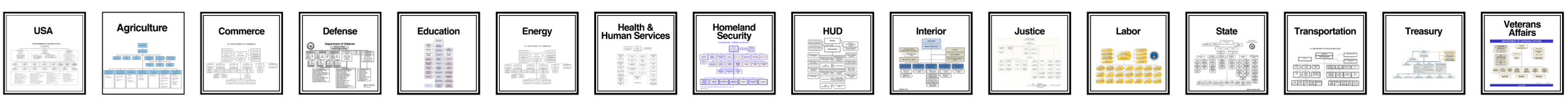
Of the 15 cabinet offices plus the USA government organization chart, 4 (or 25%) display the department seal. Some designs like the US Department of Transportation are simply embarrassing while many others look like someone tossed a copy of MS Visio, MS Project, or some other application that supports creating unattractive boxes on an office worker's desk. I found none of these 16 organization charts warranting even a number 2 mark in my rating scale.

If you look at documents distributed by private enterprise, organization charts are typically held in confidence and not made available to the public. A public document that communicates information about a given enterprise however is often found in annual reports. Many companies make available annual reports to shareholders and the public. Attention to visual presentation in these documents and most of the other documents hosted on company websites is paramount to the companies.

Gallery 2.1*: USA Organization Charts*



USA Federal Government Organization Chart



Imagine seeing an organization chart like some of the 5 rated state organization charts coming from a company like Nike, Exxon, Disney Studios, or Apple, Inc. You don't find private companies hosting information items on their websites, nor in printed form, without the fundamental design element of a logo appearing on the documents.

In years past one could argue that printing 4-color brochures, annual reports, and other documents was cost prohibitive in government and government couldn't keep up with private enterprise when it comes to the costs of printing and distribution of content. However, in an eGovernment society, there are no costs associated with printing and distribution —at least via electronic distribution. Government has not traditionally been concerned in an analog world with branding and identify. But times are changing, and now, government can take advantage of the marketing tools once left to private enterprise.

The appearance of documents hosted on websites is a visual introduction to an office. It communicates a message when a constituent first glances over a document. Is the message communicating a neat orderly flow of content that's easily read and where information can easily be extracted, or is the document a scanned fax layed out in courier type that looks like it came from a 3rd grade elementary class? The unfortunate reality that exists in eGovernment today is that content is often randomly thrown up on websites without careful attention to appearance and design that facilitate easy interpretation of the content.

Using Government Seals

To understand the importance for using government seals on documents, one needs to understand the meaning, sentiment, and history behind a seal and what the seal represents for a country's people.

Before the first Continental Congress adjourned on July 4, 1776 a resolution was passed that appointed Benjamin Franklin, John Adams, and Thomas Jefferson to form a committee to search out and bring back a seal for use by the newly established United States of America. These three men were among the five that drafted the United States Declaration of Independence.

The founding fathers of America wanted an emblem and coat of arms that demonstrated visible evidence of a nation of free people with great aspirations for the future. The task was not as haphazard as today's document developers slapping down a 5th generation scan on a form. It took six years and efforts of 14 people to agree on a design that represented the sentiment of the people at that time. The Great Seal of the United States was adopted on June 20, 1782.

The United States Great Seal went through some design changes as a result of recasting dies through the years until a new die cast

in 1904 modified an 1885 design. The die cast in 1904 resembles the seal currently used by the USA government.

This icon symbolizes so much about the original creation of a new country and the meaning it had for freedom lovers passionate enough to give their lives for their beliefs. Today the die and counter die, the press, and the cover are guarded by the Secretary of State and kept locked in the Exhibit Hall in the Office of the Secretary of State in a glass closure. The Great Seal is treated like a precious gem where government officers carry 2,000 to 3,000 documents a year to affix the seal related to matters of state.

In the modern era, the Great Seal is displayed on websites and graphics of much wider use than special matters of state. Unfortunately, government workers placing the seal image on documents is inconsistent and often crude in appearance. Not all forms display the Great Seal; and many that do display it, appear pixelated and poor representations of the original.

In the United States, or for that matter any country, the country seal represents history of the country and a belief system held by its citizens. It deserves respect and dignity. If countries place a seal on downloadable content such as forms, they should place the seal on all forms. The images should be crisp, clean, and communicate pride of a nation.

Finding Content on eGovernment Websites

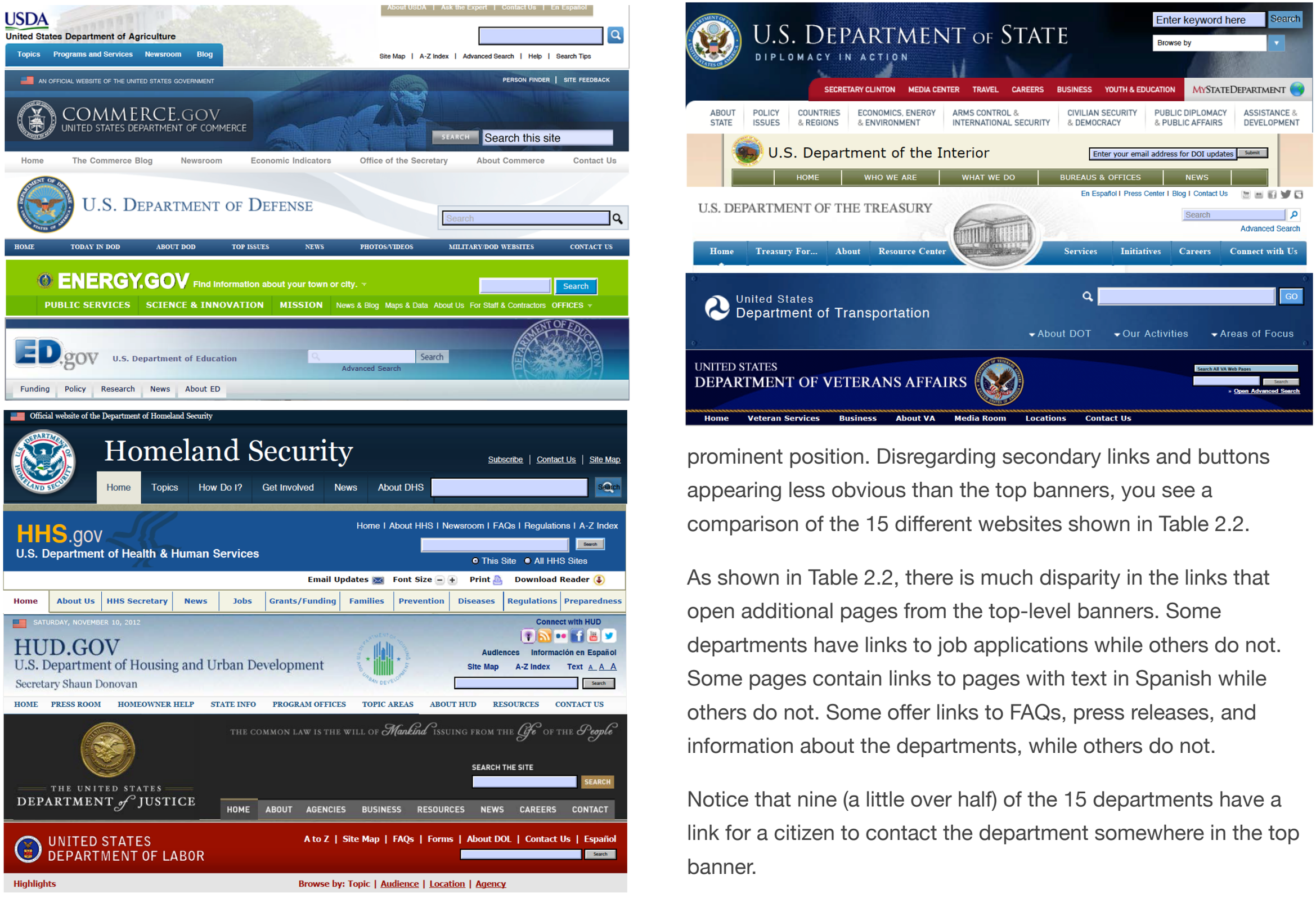
How information is organized and accessed on websites is critical to the functionality of eGovernment. Content that is well organized, flows logically, and intuitive, makes for easy navigation for constituents to find information. In a global view, eGovernment should adhere to standards and follow common rules for organizing content across all offices within a government structure.

The US President's Cabinet is comprised of 15 Secretaries with six other departments having cabinet level authority. When looking at the 15 individual cabinet secretary departments and viewing the websites for those departments you find no consistent standard for organizing information and hypertext links. Each department hosts different information, links to different sources, and uses different design models. If you remove the text where the U.S. appears at the top of each department's website, you might think that these 15 different websites, at a quick glance, come from 15 different countries.

The top banner on each website contains hypertext links. The banners are designed differently. Some banners contain a row of 3 or more buttons that link to other pages while some sites contain a row of buttons with a series of other links adjacent to the main button links. Some buttons contain drop-down menus where other pertinent links are available and some do not. The primary buttons appear in the top banner on each site in the most

Table 2.2 Federal Departments Top Banner Web Links

Department	Top Level Banner Links							
Agriculture	Topics	Programs & Services	Newsroom	Blog				
Commerce	Home	The Commerce Blog	Newsroom	Economic Indicators	Office of the Secretary	About Commerce	Contact Us	
Defense	Home	Today in DOD	About DOD	Top Issues	News	Photos/Videos	Military DOD Websites	Contact Us
Education	Funding	Policy	Research	News	About Ed			
Energy	Public Services	Science & Innovation	Mission					
Health & Human Services	Home	About Us	HHS Secretary	News	Jobs	Grants & Funding	Families	Prevention | Diseases | Regulations | Preparedness
Homeland Security	Home	Topics	Get Involved	News	About DHS			
Housing & Urban Development	Home	Who We Are	What We Do	Bureau & Offices	News	Topic Areas	About HUD	Resources | Contact Us
Interior	Home	Secretary Info	Index	Interior News	Contact Us	DOI Home		
Justice	Home	About	Agencies	Business	Resources	News	Careers	Contact
Labor	A to Z	Site Map	FAQs	Forms	About DOL	Contact Us	Español	
State	Secretary Clinton About State	Media Center Policy Issues	Travel	Careers	Business	Youth & Education	About State	Video
Transportation	Home	Citizen Services	Business Services	Government Services	Contact Us	My State Department		
Treasury	Home	Treasury For…	About	Resource Center	Services (Forms in menu)	Initiatives	Careers	Connect with Us
Veteran Affairs	Home	Veterans Services	Business	About VA	Media Room	Locations	Contact Us	



prominent position. Disregarding secondary links and buttons appearing less obvious than the top banners, you see a comparison of the 15 different websites shown in Table 2.2.

As shown in Table 2.2, there is much disparity in the links that open additional pages from the top-level banners. Some departments have links to job applications while others do not. Some pages contain links to pages with text in Spanish while others do not. Some offer links to FAQs, press releases, and information about the departments, while others do not.

Notice that nine (a little over half) of the 15 departments have a link for a citizen to contact the department somewhere in the top banner.

Of the 15 different departments only the US Department of Labor contains a link to forms in the top banner. The remaining 14 website banners do not include a link to a forms page.

The Department of Labor forms page is not well organized and displays links to a variety of different documents that are not all forms. However the forms link on the Home page takes the visitor to an area on the website where forms can be found.

Hosting eForms

Whereas government websites have traditionally focused on disseminating information to the citizenship, one of the most functional aspects of working within an eGovernment structure is performing a transaction via a government form.

I use the term transaction throughout this book to describe *processing a form for two-way communication between a constituent and a government office*.

eForms have been hosted on government websites around the world as more countries engage in eGovernment. Almost every government website from every country hosts forms to help engage people in a government process. The availability of the forms, or rather the obscurity of their locations, is one thing most governments share. As of this writing I have yet to see many government websites that host and organize eForms in a model capacity with forms that are properly constructed and made easily accessible to a country's citizens and its visitors.

The Nature of Downloadable Forms

An eForm is an electronic document that captures information. The form itself is a visual representation of some manner of computer processing that performs tasks for the user such as calculations, data entry formatting, validations, and so on. Some eForms can be filled in directly online within a Web browser and all eForms can be downloaded and filled in offline.

Forms populated with data can be submitted to host agencies by printing the form and hand carrying or faxing back to the host agency. eForms can also be emailed to host agencies or submitted electronically in XML format where servers can collect and route data to an agency's backend database system.

An eForm normally begins with an authoring application where the visual appearance of the document is crafted. After the design of the graphic elements, field objects are added that permit data entry on a computer. Computer programs can be used to both construct the graphic design and the field objects, or separate programs can be used to first create the design and a second application is used for creating the field objects.

In terms of the most popular eForm file formats that exist today on the Internet, you find forms created and finalized in Microsoft Word and Excel. These forms are both created and filled out in the same authoring application. The more popular file format used today by government for eForms is the Adobe Portable

Document Format (PDF). PDF forms begin with the design created in one of many different programs and then converted to the PDF format. Once in PDF format, Adobe Acrobat is used to add field objects. Constituents can then fill out the form using the free Adobe Reader software.

A forms authoring application gaining increasing popularity is Adobe LiveCycle Designer. Designer is both an authoring application and a forms creation tool that supports saving files in PDF format. You can design a form in LiveCycle Designer with text and graphic elements and add field objects, save the file as PDF, and deploy the forms. This application produces dynamic forms that can be used to spawn additional data fields and/or pages based on user input, expand fields to accommodate data, and be configured for data connections to connect the forms to databases. Like Adobe PDF files, users of the free Adobe Reader software can fill-in LiveCycle Designer PDF forms.

Finding Forms on Government websites

When looking at the 50 US States only the states of Massachusetts and Rhode Island post a link in the top banner of the Home page to forms. A total of less than 20% of the states post links on the Home page to a forms download page (10 out of 50) as shown in Table 2.3. All of the 10 states listed in Table 2.3 place links to secondary pages or contain drop down menus for downloading forms in conspicuous locations that make accessing forms easy for constituents.

Table 2.3 US States Posting Links to Forms Pages

State	Location on Home Page	Link Destination
Delaware	Footer	Separate page where forms links are organized according to category
Georgia	Sidebar	Separate page where forms links are organized according to category
Illinois	Body (Quick Links)	Categorical listings in drop down menu. Need to return to Home page to explore other categories.
Indiana	Footer	Links to pages where forms can be searched
Massachusetts	Top Banner	Links to page with categorical listing of forms.
Michigan	Sidebar	Button expands on Home page with links to several forms pages.
North Dakota	Sidebar	Links to forms downloads. Downloads are clearly defined in well-organized lists.
Ohio	Footer	Links to comprehensive list of all government forms. Sub pages support downloads.
Rhode Island	Top Banner	Links to pages where some forms can be downloaded. Not well organized.
South Dakota	Drop-Down Menu	Links to a page where forms are searched.

US States providing links on Home page to forms pages.

The remaining 40 US States rely on search engines for accessing forms. Some searches easily send a constituent to locations where forms can be downloaded and others are more obscure requiring one to spend some time navigating a website to find a form.

Some states have a handful of forms available for download while others have extensive forms to complete most government transactions.

If we assume that completing transactions is an important part of eGovernment, the ease for which constituents can download transaction documents or access pages for online processing is essential to the design of an eGovernment website.

PDF Forms and US Cities

Computer World Magazine and the Center for Digital Government (http://www.centerdigitalgov.com) conducted studies that examined over 11,000 US cities for eGovernment. These studies were not concerned with forms and content, but rather the design and structure of city web portals as well how well cities are investing in technology.

The studies reveal much different results in identifying the top ten cities with populations over 250,000. As you can see in Table 2.4 only the city of Seattle, Washington appeared in the top ten in both reports.

Regardless of where the cities are in terms of use of web technology, one thing all these cities as well as many other US cities have in common is hosting extensive PDF forms for processing transactions from police reports, city permits, building ordinances, recreational applications, income tax reporting to lost pets notices and much more.

The abundance of forms contained at the LGU level far exceeds the number of forms one can find on national and regional government websites.

Another thing these city websites have in common is that the organization of forms and links to forms requires some polish. No links appear on home pages for all the city websites and visitors are required to use internal search engines to locate forms. However executing a single search for *forms* shows results on many pages where links appear for downloading forms.

When browsing search results users are required to move around and look through a maze of webpages to locate a form. However, at the least, these city websites do take the user to an area that helps simplify the search.

Not all forms are designed well and many are not fillable as shown in Table 2.4. However, compared to national and regional offices forms are more easily found.

Table 2.4 US Cities over 250,000 and PDF Forms

Computer World		Center for Digital Government	
City	% Fillable	City	% Fillable
Fresno, California	40	Louisville, Kentucky	10
Tampa, Florida	30	Riverside, California	0
New Orleans, Louisiana	0	Boston, Massachusetts	10
Washington D.C.	20	Seattle, Washington	20
Houston, Texas	50	Aurora, Colorado	10
Virginia Beach, Virginia	60	Corpus Christi, Texas	40
Seattle, Washington	20	Henderson, Nevada	70
St. Paul, Minnesota	0	Charlotte, North Carolina	60
Kansas City, Kansas	10	Sacramento, California	40
Albuquerque, New Mexico	10	Fort Worth, Texas	0

Top 10 Cities over 250,000 eGovernment websites ranked by Computer World and the Center for Digital Government

The Importance of Having a Visible Link to Forms

Many eGovernment websites have embraced using Web 2.0 technologies. You find links to Facebook, YouTube, Twitter and other social media sites directly on eGovernment Home pages. Quite often you find embedded video displaying a new focus for a government office.

Browse around the US States websites and you can find the chief executive holding a puppy as a visual to support development of new animal shelters. It's cute, but is the information more important than processing a transaction?

As an example, if you drive down a highway and suffer from a severe automobile accident that renders you disabled, what is likely of more concern to you? Will it be viewing a video of the governor of your state playing with a dog on the mansion lawn or will it be quickly locating a form that helps you engage in an assistance program?

Processing transactions is a fundamental activity for government. Access to forms and effectively processing the data should be of paramount concern for eGovernment. Sadly though, government websites throughout the world devote more attention to social media connections, political advertisements, and current affairs.

Creating Solutions

When examining issues related to government offices, it's easy to identify problems but often not so easy to make suggestions for solutions. Everyone is a critic but few present concrete ideas for solving problems.

In this regard I'd like to share a view for what might be a simple solution for improving content on government websites. More specifically, let's look at the organization charts discussed earlier in this chapter.

Imagine for a moment that a national government office such as the USA Federal Government sponsors a competition among students in the country's best graphic design schools. Government informs the schools that there is a need to make the USA and each cabinet office organization charts more contemporary and with much more design appeal.

Faculty at selected schools submit final designs to the government office sponsoring the activity. Ultimately a given individual's artwork is selected for the organization chart makeovers. The reward is simply publicizing the student and school and publicly showcasing the artwork. Perhaps a trip to the White House and lunch with the President is the ultimate reward for the competition winner.

This is a relatively low cost solution for improving content on government websites. Albeit just a start, such a program can expand to other work in additional areas where government needs some help. What student in an art school wouldn't give anything they possess to be profiled by their government and add some accolades to their portfolio? This is a win-win situation for government and the selected individual. Government wins by creating a low cost solution to a problem and engages youth in participating in government activities. The individual wins by gaining significant notoriety for a job well accomplished.

With communication tools available to constituents in the 21st century, new methods can easily be explored. Government today has a huge advantage for exploring creative new ideas while developing low-cost solutions for a variety of problems.

Chapter 3

Understanding Forms in Government

The importance of forms in eGovernment

Chapter 3

Forms, and particularly forms in government, are a way of life in the modern era. From the time of birth we all encounter a form either directly or indirectly and it's highly unlikely that any individual of adult age in the USA and many other countries has not at one time or another filled-in a form.

By tradition, forms have been printed documents requiring completion either via handwriting or typing. In today's society forms are most often found on the Internet where constituents download a form saved in many different file formats or complete online HTML forms found on websites.

IN THIS CHAPTER

Understanding Forms in eGovernment

Most of the forms related to reporting and doing business with the USA Federal Government and all state governments are found on eGovernment websites. As a matter of fact, certain Web pages host listings of forms related to many different governmental departments. Browse www.forms.gov and you find a collection of forms related to many of the USA Federal Government departments, particularly those departments that fall within the President's Cabinet.

The Importance of Government Forms

A government form is most often the first introduction to a government office by a constituent or visitor. Before one interacts with a government official, one is likely to acquire a form. This is most obvious when downloading forms from government websites.

Even when walking into government offices, trays of forms are arranged on counters for consumers to complete before standing in line to submit a form. Visit a local post office, Department of State office, a local IRS office, or government office in almost any country and you find huge signs describing forms to be completed before moving to a window to interact with a government representative.

Since a form is the consumer's first introduction to a government office, the design, clarity, and functionality of the form needs to be well defined. It's a first impression and something that is supposed to ease the burden for following steps that result in some sort of action whether it be a permit, application, reporting document, or some similar type of action.

More importantly, those forms hosted on websites need much more clarity and ease of completion than forms found in local offices. No individual is accessible for fielding questions to clarify steps in completing a form when downloading documents from the Internet. The forms themselves need to be instructive and designed to facilitate simplicity in providing accurate data.

Web Hosted Forms

Forms are often saved in different file formats. You can find forms saved in Microsoft Office file formats such as Microsoft Excel and Microsoft Word. In fewer instances you can find forms saved as image files that were created from scanning analog forms. However, when examining forms hosted on government websites worldwide, the overwhelming majority of forms are saved in Adobe PDF format.

Adobe PDF is the de facto standard for most government forms for several reasons. First, Adobe PDF documents can capture the design and appearance of the native document including text, fonts, and graphics and display the form exactly the same on all computer systems. Secondly, the consumer is not required to purchase any special software to view and print the form. The

free Adobe Reader software is available for download from Adobe Systems' Web site.

A third reason for using the PDF format is that the forms can be made fillable where consumers can type information on a form and print the document with the field data displayed on the print output or electronically submit the form including the form data to the host agency.

Other types of forms found on some government websites are HTML forms that are completed within Web browsers. A consumer needs to have an Internet connection to complete the form and once completed can submit the data to the host agency. The data are then collected on a server and typically routed to a database program.

HTML forms are used much less frequently by government than PDF forms. HTML forms cannot represent the same document as printed forms found in local or federal agency offices as well as PDF forms. HTML deals primarily with data and therefore doesn't provide the host agencies with the same opportunities for filing and archiving the forms as do PDF documents.

A Form is a Form, is a Form

Inasmuch as many government offices host web pages listing forms on their websites, you typically find documents on these pages described as *Instructions*, *Reporting Notes*, *XYZ Descriptions*, *Use Policy*, *Brochure*, etc. These documents are NOT forms. An electronic form is a document that contains data fields. Period.

Scanned documents are NOT forms. They are image files. Descriptions are NOT forms. They are text files containing information about forms. Unfortunately many government websites confuse constituents by hosting an array of documents in a specific area where visitors expect to find forms.

Fillable Forms

A fillable form has live data fields. This means that the consumer can open a form in an Adobe Acrobat viewer —either the free Adobe Reader software or one of the commercial Adobe Acrobat products and type data on the form.

A non-fillable form is one without data fields. This type of form requires the consumer to print the form on a desktop printer and handwrite data on the form to complete it. In terms of non-fillable forms, the file format does not need to be a PDF document. A form can be saved as a JPEG file that can be printed from a JPEG viewer, a text format that can be printed from a text program, or another format requiring the consumer to use the program in which the form was originally created. For example, a Microsoft Word document requires the consumer to have Microsoft Word installed on a computer or a program such as OpenOfficeOrg Writer or IBM Lotus Symphony that can read Microsoft Word files.

When consumers use an Adobe PDF form, the burden of transposing data by office workers is lessened. Deciphering handwriting is not needed since the data appear clear and easily legible. If a host agency collects a form and the data, then transposing data from printed forms to computer systems is not needed thereby reducing labor costs significantly.

Static fillable forms

A static fillable form is one that is created in Adobe Acrobat and populated with data fields. The static nature of the document is related to the form fixed with a specific design that requires the form recipient to fill in the form using the allotted fields and field lengths constructed by the form designer. Figure 3-1 shows a static form from the US Department of State.

> ***Note***
>
> *Notice on the form in Figure 3.1 items 12-14. The form author determined the field length and the amount of text that can be typed into the field. If a form recipient gets this form and wants to add more than the space allotted for the information, the static characteristics of the form prevent the user from displaying more data than the prescribed field length.*

Figure 3.1 Sample Static Form

DOE Form 3250.1 (Test)
(10-78)

U.S. DEPARTMENT OF ENERGY

EXIT INTERVIEW

PRIVACY ACT STATEMENT
(Public Law 93-579)

The authority for requesting this information is derived from Public Law 95-91 (the Department of Energy Organization Act) which authorizes the Department of Energy to select and employ such officers and employees as are necessary to perform the functions vested in DOE. The information requested is intended for use by DOE personnel officials in evaluating the effectiveness of the personnel management program of DOE. Completion of this form is voluntary. However, if the employee fails to appear for the exit interview, his final paycheck may be delayed since necessary information will be obtained by mail.

1. EMPLOYEE'S NAME	2. TITLE	3. SCHEDULE, SERIES, GRADE

4. OFFICE	5. DIVISION	6. BRANCH	7. SEPARATION DATE

8. REASON FOR LEAVING

9. WERE YOUR SKILLS EFFECTIVELY UTILIZED? ☐ YES ☐ NO *(If not, please specify why below)*

10. WERE YOU TREATED FAIRLY?
☐ All of the time ☐ Most of the time ☐ Some of the time ☐ Never

11. RATE THE EFFECTIVENESS OF COMMUNICATION WITHIN YOUR ORGANIZATION *(check the most appropriate box)*
☐ Above average ☐ Average ☐ Below average ☐ Poor

12. WERE YOU ADEQUATELY TRAINED FOR YOUR JOB? ☐ YES ☐ NO IF NOT, WHAT ADDITIONAL TRAINING WAS REQUIRED?

13. PLEASE STATE BRIEFLY WHAT YOU LIKED MOST ABOUT YOUR JOB:

14. PLEASE STATE BRIEFLY WHAT YOU FOUND LEAST APPEALING IN YOUR JOB:

15. WOULD YOU BE WILLING TO RETURN TO WORK FOR DOE IN THE SAME OR ANOTHER CAPACITY? ☐ YES ☐ NO

16. WOULD YOU RECOMMEND EMPLOYMENT IN DOE TO A FRIEND? ☐ YES ☐ NO

17. GENERAL COMMENTS OR SUGGESTIONS:

18. SIGNATURE	19. DATE

20. PERSONNEL SPECIALIST'S NOTES:

Sample US Department of State static fillable form

Dynamic fillable forms

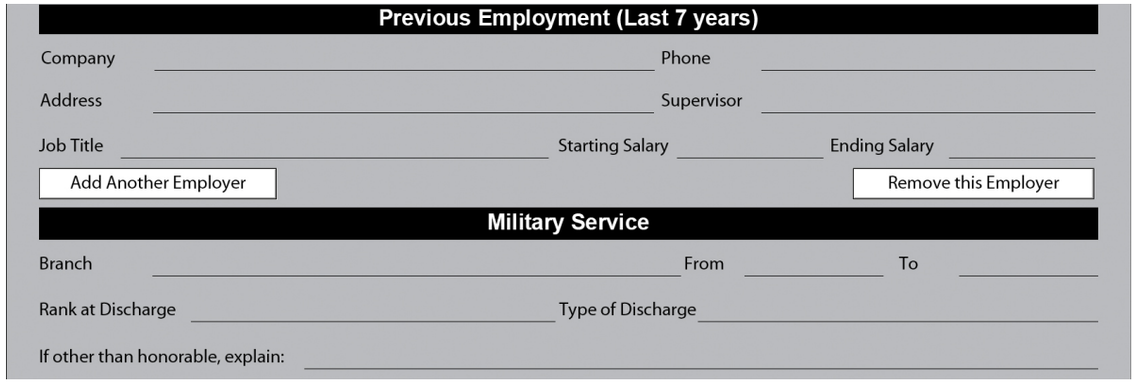
A dynamic form is one that accommodates either the user or the data needs. User needs are those when a user requires more space on a form to answer a question. Data needs are when you import data into a form from a data file that populates the form.

Dynamic forms can permit form recipients opportunities for expanding field lengths or populating additional fields and/or pages according to the amount of data the form recipient needs to adequately fill in a form.

The form in Figure 3.2 is a dynamic form. The first block of fields requesting the form recipient to complete employment history provides space for a single employer. When the form recipient clicks the Add Another Employer button, another block of fields dynamically appear in the form for accommodating information for a second employer. In Figure 3.3, the button was clicked twice to create two additional blocks of fields for employer information.

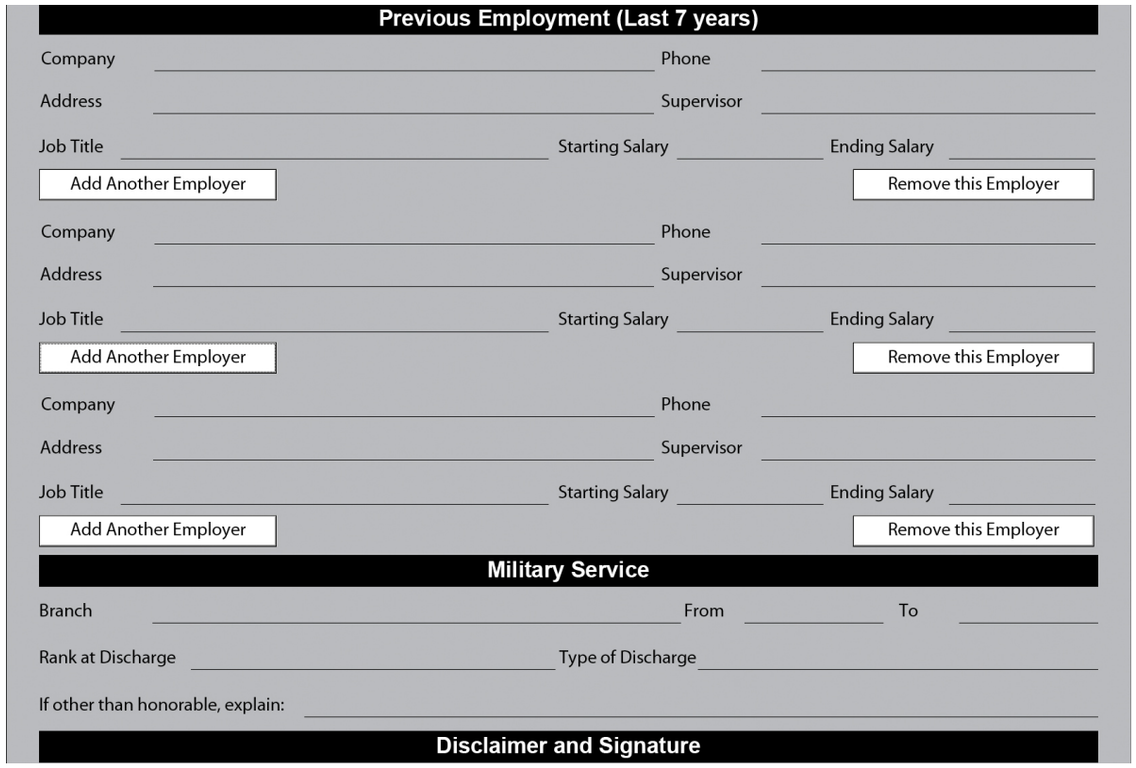
> ***Note***
>
> *The dynamic characteristics of a form, such as the one shown in Figures 3.2 and 3.3, can be completed by users of the free Adobe Reader software.*

Figure 3.2 DynamicForm



A dynamic form with a single block of fields for previous employer information

Figure 3.3 After Adding Two Blocks of Fields



The same form after clicking the Add Another Employer button twice

Fillable forms authoring applications

When creating PDF forms, the PDF form authors need one of the commercial Acrobat products. In Adobe Acrobat XI you can create PDF forms using Acrobat Standard (Windows only), or Acrobat Pro (Windows and Macintosh).

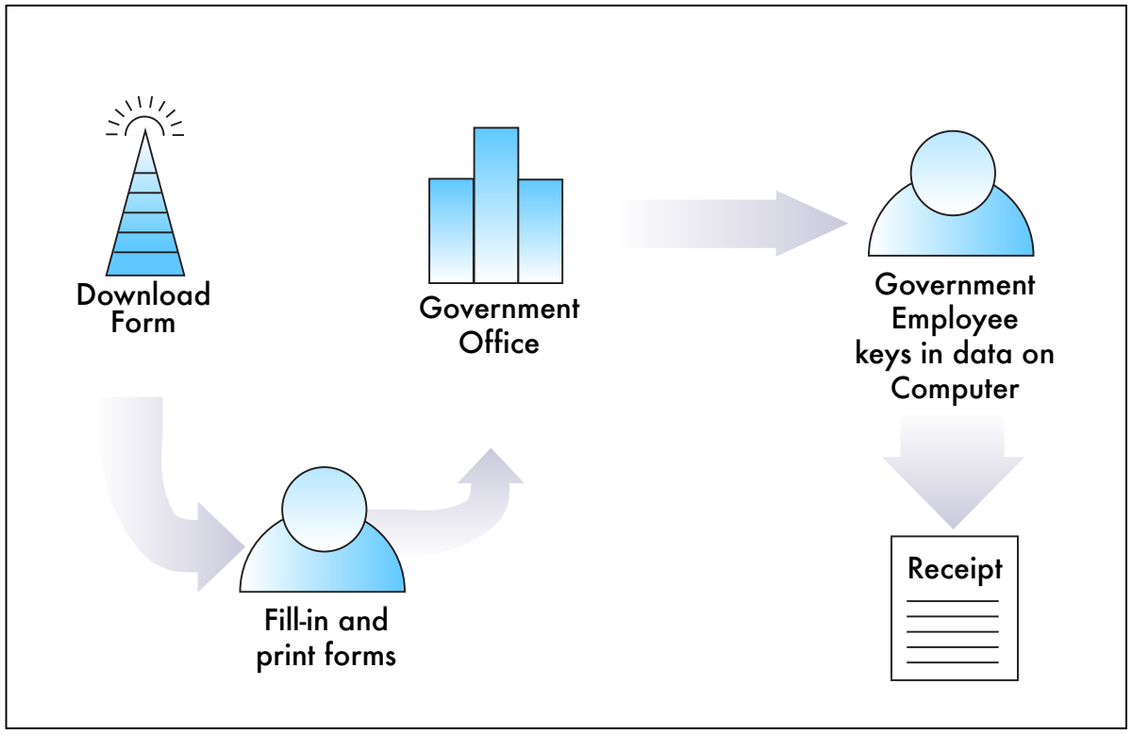
Acrobat supports creating static PDF forms. All the tools for adding field objects are available using any one of the commercial Acrobat viewers. To create dynamic forms you need another application. Prior to Acrobat XI, Adobe LiveCycle Designer shipped with Acrobat Pro and Pro Extended on Windows only. With Acrobat XI you need to purchase Adobe LiveCycle Designer separately. LiveCycle Designer is used to create forms similar to the form shown in Figures 3.2 and 3.3.

The tools used by the US Federal Government include Acrobat and Adobe LiveCycle Designer. Inasmuch as LiveCycle Designer is often used to create dynamic forms, all the forms researched for this book were static forms saved from Acrobat or Designer. There were no forms the author found available on the USA Federal Government websites that had dynamic characteristics similar to what you see in Figures 3.2 and 3.3.

Forms Processing Workflows

The mere hosting of electronic fillable forms and making those forms available to constituents is only one part of a true electronic workflow equation. By tradition, forms are downloaded from the Internet, filled in on a computer, printed, and the final print is either hand carried or faxed back to the host agency. As shown in Figure 3.4, data needs to be transposed from a printed form to a computer at some point in the forms processing steps.

Figure 3.4 A Semi Electronic Workflow

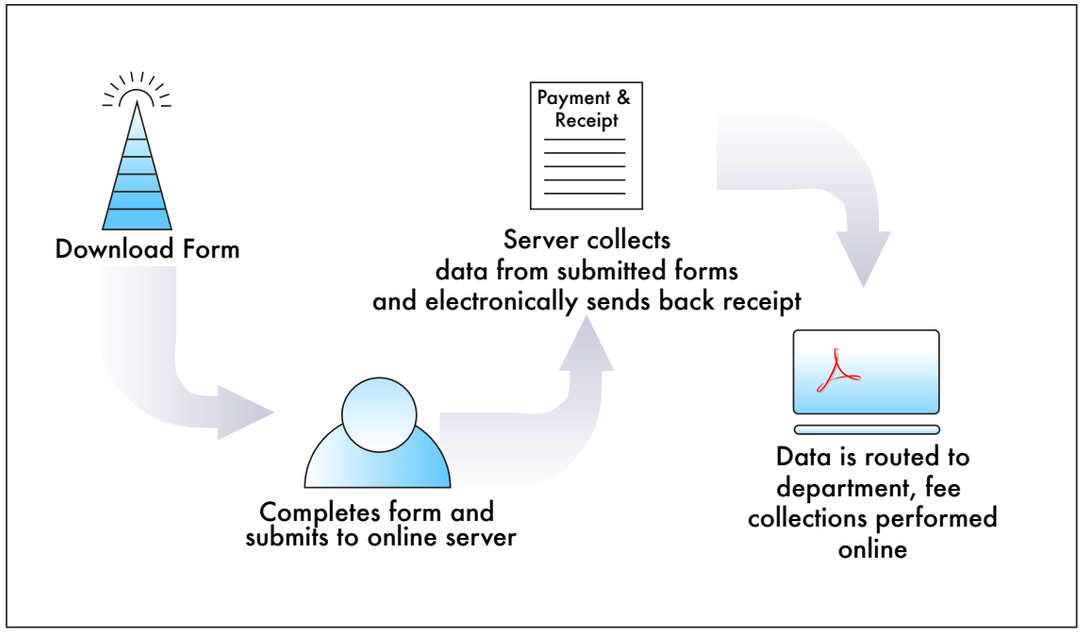


A semi-electronic workflow

The alternative and one that has not been implemented by almost all agencies in the USA Federal Government as well as all US States Government offices, as of this writing, is a true electronic workflow. In this type of workflow, a constituent downloads a form, completes the form, and submits the form back to a host

agency server along with an online payment. In Figure 3.5, the workflow is illustrated.

Figure 3.5 A Complete Electronic Workflow



Electronically processing data

The advantages for using a true electronic workflow are obvious. The government office does not need to spend time keying in data on computers and the use of paper is eliminated in the process. The burden on government is greatly lessened thereby releasing time from employees to divert their energies more to data analysis than in data handling. In some cases the workforce can be lessened to help save costs in government. Retraining employees to analyze data as opposed to key in data can often produce more revenue for government when employees devote attention to investigating falsified data and impose penalties on violators.

Although servers have been in existence for some time, government has not made use of modern technologies to create workflows such as the one shown in Figure 3.5. As of this writing I have found no government office in the world that one could look to as a model example for processing form data.

Forms Hosted on World eGovernment Sites

There are 206 sovereign states in the world. The one thing over 90% of the world governments share in common is hosting forms related to visa applications. Of the 206 world governments, 188 (91%) host various forms for visa and passport applications. Many governments do not have websites for the central government offices, however, there are various embassy websites and some other country websites that host visa application forms.

One hundred thirty-seven of the 188 websites hosting visa applications do not host any other type of form. Thirty-three countries have a limited set of forms related to visa applications and various other forms such as business, tourism, and other government operations. Of the 33 websites hosting a limited set of forms, approximately 36% (12 websites) are fillable forms. Three of the websites host MS Word files while the remaining forms are PDFs (Table 3.1).

Table 3.1 Countries hosting limited forms

Country	Fill-able	Not Fill-able	Word Files	Country	Fill-able	Not Fill-able	Word Files
Bahamas		X		Israel	X		
Bahrain		X		Japan		X	
Bangladesh		X		Mauritius		X	
Barbados		X		Mexico		X	
Bhutan		X		Micronesia	X		
Botswana		X		Namibia		X	
China			X	Netherlands	X		
Cyprus		X		Palau		X	
Eritrea		X		Papua, New Guinea		X	
Falkland Is.			X	Qatar	X		
Fiji		X		St. Lucia		X	
Finland	X			Saudi Arabia	X		
Germany	X			Slovenia			X
Grenada		X		Solomon Is.		X	
Iceland	X			Somalia		X	
Indonesia		X		Sweden	X		
Ireland		X					

Eighteen of the 188 websites hosting forms provide moderate to extensive PDF forms covering many government operations (Table 3.2). Only 4 of the 18 websites host fillable forms. Of all the world government websites only the country of Kenya has a link in the top banner to a forms page. Some websites have links in other areas on the home page but most governments do not provide links to organized forms pages. Visitors are required to search web sites to find forms.

Table 3.2 Countries hosting moderate to extensive PDF forms

Country	Fillable	Not FIllable
Australia	X	
Belize		X
Canada	X	
India		X
Kenya		X
Luxembourg	X	
Malaysia		X
Philippines		X
Singapore		X
Sudan		X
Taiwan		X
Tonga		X
Trinidad/Tobago		X
United Kingdom		X
United States	X	
Vanuatu		X

Kenya provides a link to Forms on the top banner of the Home page.

Chapter 4

Adobe PDF & Sustainability

Creating more sustainable workflows

Chapter 4

Government spends money and devotes time to supporting ideas for developing a green society. We find government websites posting information related to greenhouse gases and the carbon footprint. We're inundated with an explosion of information related to protecting the environment and what your government is doing about it.

IN THIS CHAPTER

When we look at paper usage in the USA, the average American consumes 700 pounds of paper annually —that's the world's highest per capita figure.

Printing a 500 page PDF document creates 15 pounds of CO2, consumes 50 gallons of water, and 5 pounds of paper. Hence the average American is tossing more than 2,000 pounds of CO2 in the atmosphere, consuming 7,000 gallons of water, and consuming 700 pounds of paper annually.

In terms of paper usage and costs for government, the USA Federal Government spends over 1.3 billion annually just for printing costs as reported in 2009 by Lexmark and O'Keeffe & Company.

What government attempts to do to reduce the carbon footprint is enact legislation that requires government offices to use recycled products. In the State of Washington, for example, legislation in 2009 required state agencies and state colleges having 25 or more employees to use recycled paper and further reduce printing by 30%.

Some of the measures proposed by governments help reduce the carbon footprint within government offices but they do nothing for constituents. Documents posted on government websites such as forms still typically require end users to print the forms and either mail or fax forms back to a host agency.

US State Leaders in Going Green

In 2007 Forbes conducted a study that ranked US States for measures of low carbon dioxide and greenhouse gases. The top three leaders that had the lowest levels of carbon dioxide emissions were the states of Vermont, Oregon, and Washington.

Policies in these three states were the strongest among all states surveyed. I have no reason to doubt that the three states are indeed leaders in taking initiatives to help improve air quality and reduce toxins. But, how well do these states do with government forms, creating fillable forms, and electronically routing forms? How well do they do beyond laws that limit paper use in government and extend measurable initiatives to the constituents?

I took a brief glance at forms hosted by the three states and examined forms from different departments. Only a small sample of forms was examined. My findings are not conclusive and do not represent the status of all forms hosted by each site. Rather, my analysis is more of a case-in-point view of some selected forms.

Vermont eForms

The Home page for the Vermont portal hosts a link for finding tax forms. No other links on the Home page take the visitor to an organized page for government forms. If you perform a search on

the site for forms, you arrive at a listing of forms that's not very well organized and not categorically defined.

Most of the forms contained on the www.vermont.gov website are not fillable. All forms I examined required the user to print and fax or hand carry forms to the host agency. In terms of sustainable measures for forms processing the State of Vermont is, in my opinion, substandard and not impressive.

Although not related to sustainability, I found most forms designed poorly in terms of appearances, not specific enough for how to process the form, and the forms are not secured. As a matter of fact some forms that contained calculations were not protected against user manipulation.

Forms that contain calculations in particular can take great advantage for using hierarchical field names. On one form where I found calculations included on the form, field names for a list of 18 items that need summing the data were named FillText1, FillText2, FillText3, etc. The JavaScript written to calculate the sum needed to specifically name each field requiring 18 lines of code. Had the PDF author used hierarchical names such as FillText.1, FillText.2, FillText.3 only 3 lines of code are needed to write the script for calculating a sum.

Oregon eForms

The www.oregon.gov website is similar to the Vermont website in terms of not posting a link to a well-organized forms page. When searching the portal for forms a disjointed list appears without any categorical organization of forms hosted by the state.

Oregon hosts forms in a number of different file formats including MS Word, MS Excel and PDF. Of the PDF forms they host, most are not fillable. One form I found interesting was a form for Air Quality Permit Holders. This form was fillable however the form filler needs to complete the form, print, and mail to the Oregon Department of Environmental Quality.

Washington eForms

Washington follows the lead by Vermont and Oregon in terms of the ease of finding government forms. No link exists on the Home page; and when searching for forms on the portal, a long non-categorized list appears. You need to poke around the Washington site to find the form you want to complete.

Forms on the www.washington.gov site contain many fillable forms while many are not fillable. Users are required to perform the same steps for processing forms such as printing, and mailing or faxing the forms.

The process for submitting forms is the same for these green states and no forms I found were setup for processing transactions online (other than some HTML forms).

Washington forms appearance-wise look much more professional and by and large were the best-designed forms of the three states.

The Best US State eForms

What do I consider the best state government office hosting and designing PDF forms? You might think that among the best is a state like California, New York, Pennsylvania or other state with more resources and larger budgets. However in my estimation these states don't compare with the State of North Dakota (www.nd.gov). This small state with less than 700,000 people has a direct link on the www.nd.gov website Home page for Forms that takes the visitor to a well-organized form contents page. Links are made to pages equally well organized for users to easily locate forms within departments.

North Dakota's forms designs are pristine with the state seal appearing on every form. Forms are layed out with design appearances that are attractive, consistent, and well organized making the forms logical and easy for constituents to fill out. All the forms are fillable forms and they are enabled for Reader users to save the form data.

Unfortunately North Dakota's forms are not perfect. Data routing via submit buttons are not found on the forms and there are design issues that need improvement. Apparently North Dakota uses the LiquidOffice Java Publishing Framework for creating the forms. I'm not familiar with the application, but after examining many forms from this state I find that all content appears as form fields even the background form design. This makes the forms bulky and overburdened with unnecessary data. Editing a form is difficult because a form editor cannot easily access the data fields from within a mess of content fields.

Nevertheless, compared to all other US States and the USA Federal Government, North Dakota excels in many areas related to well-designed fillable eForms.

Sustainability Conference Registrations

What we find in government is a paradoxical stand between eForms and sustainability. Government wants to convince constituents that it is taking great strides in measures toward environmental protection, yet just about every government worldwide has yet to adopt electronic workflows that actually result in significant reductions of paper and printing when it comes to processing transactions.

What's more astounding is that individual government departments charged with environmental protection laws and issues are yet to adopt simple measures that can reduce the use of paper by constituents.

When we move out of eForms for government and look at international conferences focused at delivering information and studies on sustainability we find the same type of paradox as found in government.

Sustainability conferences attract some of the best world authorities on environmental protection. The conferences are

designed to provide information for government and industry on how to reduce carbon footprints and achieve measures in sustainability.

For years I've examined conference websites and downloaded PDF forms for conference registrations. When a PDF form is hosted by an international conference on sustainability one would expect to download a form that can be electronically processed.

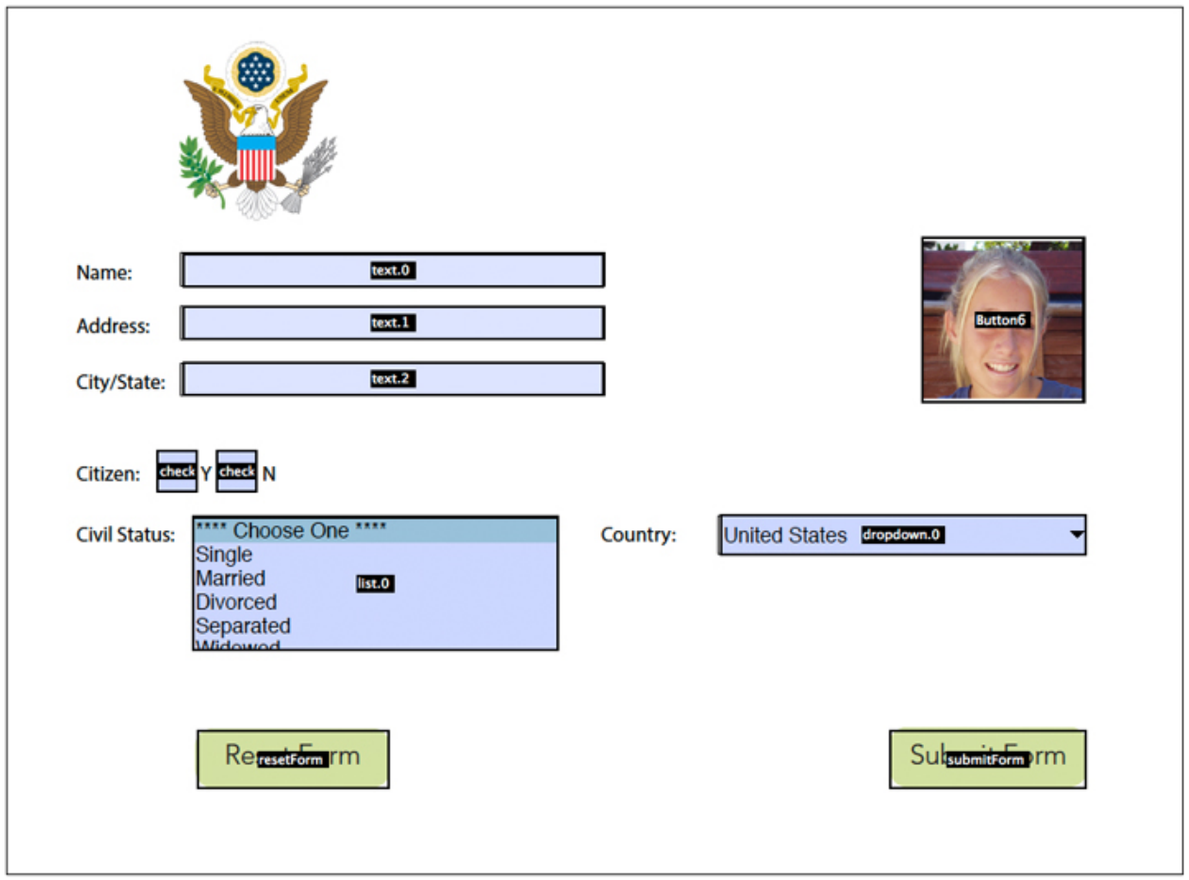
To date I haven't found a conference registration form for a sustainability conference that enables the user to fill out the form and submit the form with data back to the host. Almost all of the PDF forms for these conferences are not fillable. Virtually all PDF registration forms require the end user to print, handwrite the data, and mail or fax the form.

With the tools you have at your disposal from Adobe Systems, you can create fillable PDF forms, collect form data, and process data that is efficient, much easier for constituents, and helps you move to a more sustainable workflow.

Chapter 5

The Anatomy of an eForm

Properly constructing forms



Chapter 5

IN THIS CHAPTER

For the purposes of this discussion I restrict eForms to PDF forms. Government websites host a variety of form types such as MS Word files, WordPerfect files, MS Excel files, and HTML files.

As a matter of practice you should think of hosting only PDF forms and HTML forms. In some cases an HTML form may prove a better solution than a PDF form. All standardized government printed forms used for analog processing should be PDF forms.

The most important thing to consider when deploying forms is consistency. An eGovernment site that hosts PDF, MS Word, MS Excel and a variety of other file types makes it difficult for constituents to work with the agency forms. Because PDF forms can be designed with consistency and contain all pertinent form elements, PDF is the best format for downloadable forms. Moreover, PDF is the de facto standard among government offices worldwide.

Essential Form Elements

Every form hosted on eGovernment websites should include form elements consistently on every single form. These essential form elements include:

Fillable

Forms should be fillable meaning that they should have field objects where a constituent can type data. In addition to being easier for users to fill out forms, the data on forms can be captured and introduced into database management programs.

A few years ago I examined a random sample of forms hosted by the USA Federal Government. The research is a bit dated since forms are revised routinely. Therefore some improvements may have been made with regard to certain departments and the www.forms.gov website.

The USA hosts a special website (www.forms.gov) where many federal forms are stored. The USA Federal Government makes an effort to update and keep the website current. Not all forms are available; however, when searching for USA government forms, this is the best place to begin a search.

I randomly examined a sample of 20 forms from every cabinet department in the USA Federal Government and based conclusions on the sample data. Every effort was made to perform the same searches for forms and random selection of forms for analysis.

In my research through random sample testing I found 62% of the forms hosted by the USA Federal Government were fillable. Over one-third (38%) of the hosted forms were not fillable.

Document Metadata

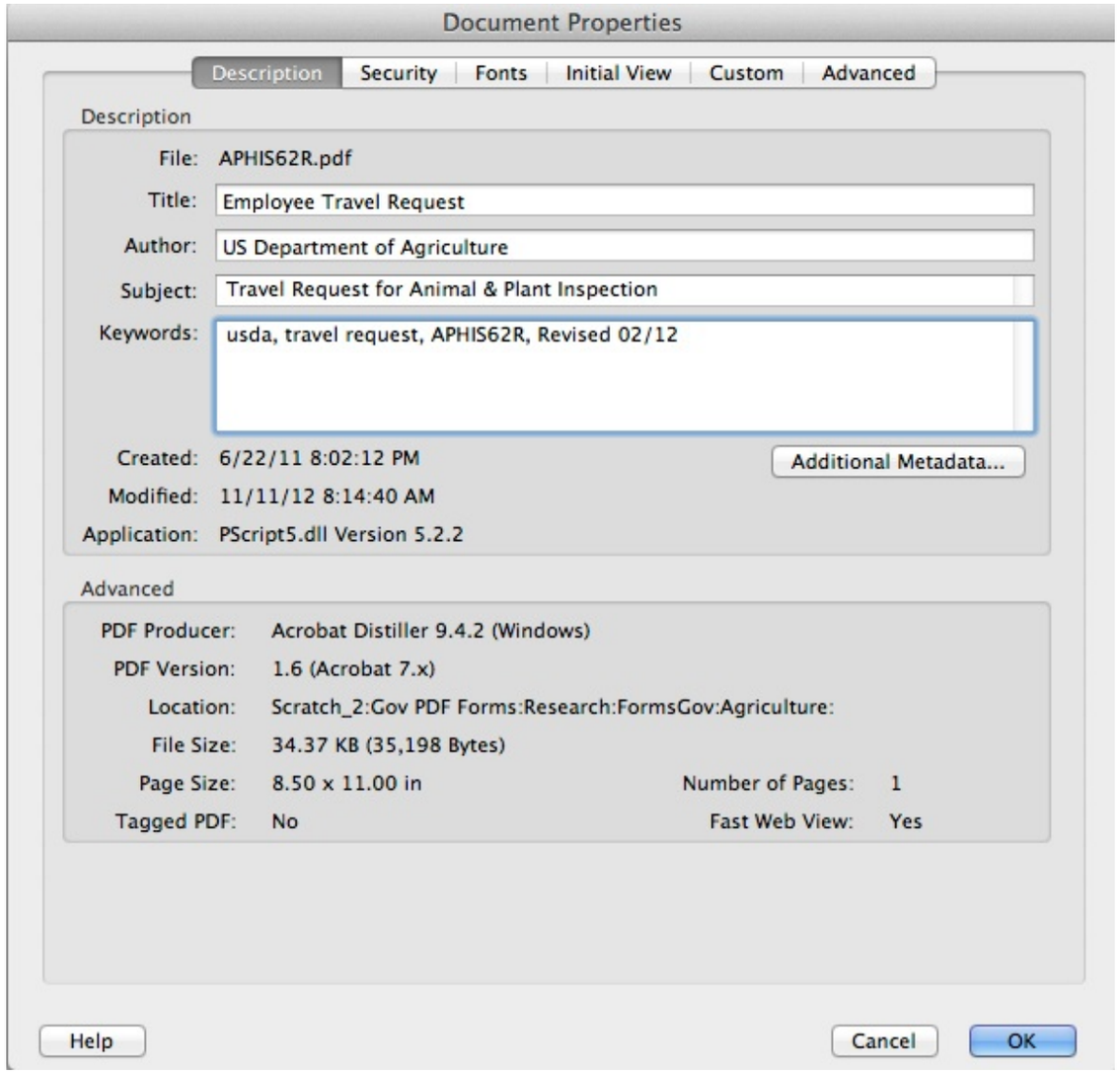
Certain data needs to be added to a form's document properties. In Acrobat you access document properties by pressing Ctrl (Windows) or Command (Macintosh) + D. You can also access Document Properties by clicking the File menu and choosing Properties.

> ***Note***
>
> *On the remaining pages of this book I use CTRL/Command to describe the Control key (Windows) and the Command key (Macintosh).*

In the Document Properties window click Description. Here you find editable text fields where you can supply information for the document Title, document Author, document Subject, and document Keywords.

The Title should contain the form title such as Application to Participate in an FCC Auction, Request for Hearing by an Administrative Judge, Employment Eligibility Application, or similar type of title that describes the form.

Figure 5.1 Document Properties Window



Document Properties Window (access using CTRL/Command + D)

The Author field is very important. Unfortunately, on most government forms hosted by the USA Federal Government this field is inappropriately filled in. The Author field should denote a department or division where a form is constructed and used. This field should not include the form author's name.

Government employees change frequently; however, department names change much less frequently, and it's much easier to run down a department than an individual if a form needs revision.

The Subject field might include a subset of the title field. You may have Job Application in the Title field and further describe the form in the Subject field by using a department name, such as the HR department responsible for processing job applications.

The Keywords field is used for a collection of descriptors that further demonstrate the form's use such as *effective 1/1/2013*, *supersedes form DOD101-A*, *Army Civil Service use only*, etc.

Why document metadata is important relates to managing forms on a website and within a forms repository. If you want to search for a form you might use Acrobat Search. In the Search window you find options for including Title, Author, Subject and Keywords fields as well as dates and Boolean expressions. Hence you may want to perform a search where *Title* contains Application, *Author* contains FCC, *Subject* contains Pilot, and *Keywords* contains Revision. When you perform a search, the only items appearing in the results window are those forms meeting the search criteria. You could search a repository of 100,000 or more forms and only those that meet your criteria are displayed in the search results window and the results are reported in a matter of seconds.

In my research I found less than 20% of USA Federal forms contained appropriate document metadata.

Document Security

EVERY form hosted on government websites should be secured with password security or certification to prevent tampering with the form.

Securing even the most basic forms is critical. Why this is important is due to any potential legal action a constituent may have against government. Attorneys preparing for litigation do not have time to read every word in every document used as exhibits in courtrooms. An attorney representing government may have seen a given form a thousand times while representing the client. Attorneys typically skim over documents they're familiar with and spend their time reading and studying new material essential for the case.

If you don't secure a document appropriately, users can alter text making it difficult for a government's legal department as well as having forms returned with erroneous information.

In my research of the USA Federal Government I found only 21.6% of the random sample forms secured. In the Homeland Security Department I found over 80% of the Homeland Security forms not secured.

Enabling

Prior to Adobe Reader XI forms authors needed to enable PDF files with special usage rights for Adobe Reader users. What this means is the PDF author needed to save the PDF from Acrobat with an enabling feature that permits the Reader user to save form data.

Adobe Reader XI no longer requires enablement. When PDFs are enabled with Adobe Reader usage rights, a licensing restriction is imposed by Adobe Systems. Enabled files restrict users to collecting and processing data on a maximum of 500 forms. Now with Adobe Reader XI and elimination for the need of enabling PDFs, there exists no license restriction on the number of forms you can retrieve and collect data.

If forms are filled-in by users working with Adobe Reader version X and below, the forms need to be enabled. If forms are completed in Adobe Reader XI, the forms do not need enablement.

Your best solution is to configure forms so that only Adobe Reader XI and above can view the PDFs. This is a matter of adding a simple JavaScript that prohibits filling out a form in earlier versions of Adobe Reader. (See Chapter 17 for steps on how to create JavaScripts that limit users to a given version of a PDF form).

Prior to the release of Acrobat XI, I found only 12.2% of the USA Federal forms enabled with Reader usage rights. Today this is a moot point for Reader XI users. Any legacy PDF containing form fields can be opened in Reader XI and the data can be saved without enablement.

Accessibility

Forms can be made accessible for screen readers via assistive devices for the vision and motion challenged form recipients. When a form is made accessible, the form can be read aloud using an assistive device so people with these challenges can complete the form without human assistance.

I found less than two-thirds of all forms on the USA Federal Government websites accessible.

Design Elements

The appearance and design of forms is critical. A form may be the first introduction a constituent or visitor has with a government office. It's a first impression. You want to be certain you give the best first impression you can to any visitor.

Analyzing visual design appearances is a subjective evaluation. If I were to grade the USA Federal forms for overall design appearances my personal opinion is that more than 40% of the forms I reviewed were substandard in terms of design and appearance.

Government Seal and Branding

Enterprises spend billions of dollars annually for branding their products and services. Visit websites, look over brochures, examine products, etc. and the one thing you find in common is the identity symbol, most often in the form of a logo, on every item.

Every government and every wing of government has a seal. This graphic element should be included on each and every form distributed by a government office. The seal is a branding item and immediately informs the user from what department the form originated.

Seals on government forms routinely are displayed as horrible pieces of artwork. Users drop in a gif image or a scan on a form that looks crude and unprofessional —hardly the same kind of graphic one would find on a Nike; Apple, Inc.; CNN; Colgate-Palmolive, etc. print or website. Government needs to devote the same attention for visual display as you find in enterprise.

Users who design forms should have access to crisp clean department seals that are positioned on forms in the same location with consistent design appearances.

In my research I found only a little over 25% of USA Federal Government forms contained a department seal.

Original Documents Converted to PDF

The no-no here is never, ever, use a scanned paper document as the background design for a form. If you don't have access to the original document, you need to recreate the file in MS Word, Adobe InDesign, or some other authoring program and convert the file to PDF. Original files converted to PDF not only look clean and attractive, they also contain searchable text.

Design Consistency

The overall design of forms should be consistent and appear similar. Inasmuch as the content changes you can maintain several standards such as whether to use keyline borders around elements, consistency with fonts on forms, consistency with color, checkbox and menu appearances, etc. These elements should be identical on ALL forms for all wings of government. Do not leave individual design licenses to separate departments. If you do, you end up with forms that look like they come from several different governments.

Form Handling Information

Every form should include a section for how the constituent proceeds to submit a form. Ideally, all your forms should contain Submit Form buttons that a user clicks to submit the form electronically. The user should be instructed to click the button when the form is completed and ready to submit.

If you are in the process of converting legacy forms, any forms not set up for electronic processing should include submission information such as printing, faxing or handing in at a local office.

If form instructions are contained in a separate document a document link to the instructions file should also be included in the form.

Field Construction Elements

The placement of field objects and how they are configured is important for all your form designs. Some of the more important considerations include:

Field naming conventions

Field names are transparent to the user but they are of great benefit to the forms authors and authors who need to revise forms.

Consistent naming conventions should be used on all forms within an entire government complex. Field names such as *Item 1*, *Item 2*, *Item 3* or *employment history one*, *employment history two*, *employment history three*, etc. are inappropriate and create many problems for authoring complex forms.

Field names should be constructed using hierarchical names such as *item.1*, *item.2*, *item.3*, etc. or *employmentHistory.1*, *employmentHistory.2*, *employmentHistory.3*, etc. Using field

names of a hierarchical order (no spaces and a dot + number) simplifies field construction, editing, calculations, and more. For examples on how hierarchical names are used in form designs, see Chapter 10.

Mutually exclusive fields

If you use check boxes or radio buttons for choices that require either/or responses, the fields should be mutually exclusive — meaning that only one choice can be made within a common field group. For example, if a processing fee is required with credit card information, you may have choices for a user to select AMEX, VISA, M/C, DISCOVER, etc. If fields are not mutually exclusive the user can check multiple items. Setting up mutually exclusive fields prohibits the user from clicking more than one choice.

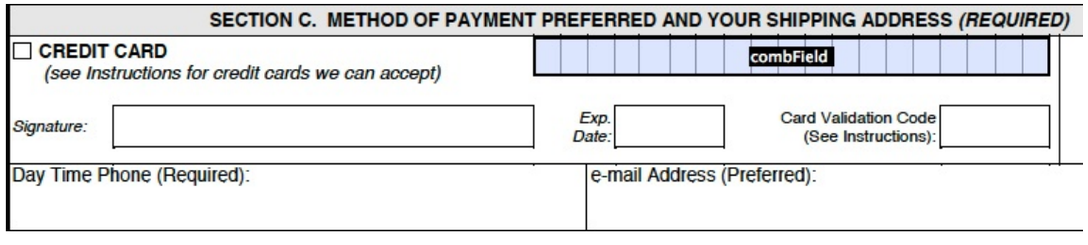
Checkboxes and radio buttons

Check boxes and radio buttons should appear to the left of text descriptions. Be certain all forms are designed consistently. Having some boxes to the left and others on the right can confuse users.

Comb fields

Comb fields are commonly found on government forms (see Figure 5.2). Don't rely on users adding spaces or tabs to fit data within hash marks. Acrobat has a comb field feature that proportionately spaces the data.

Figure 5.2 Comb Field



A form from the US NAtional Archives containing a comb field.

Reset buttons

Perhaps not all forms require a button to clear the data. Some forms though are best designed with reset buttons to help a user start over. Buttons should be unobtrusive but clearly visible.

If you design a complex form, be certain to add reset buttons for sections. Users can become frustrated if you have a reset button that clears all data including identifying information while they only want to clear a section.

Submit buttons

Submit buttons are used to send the form from the user's computer to the host agency. Submit buttons should appear at the same physical location on all forms such as top right, last page lower bottom, etc. Wherever your office decides to best locate submit buttons, be certain that the button placement is consistent across all forms.

Calculations

Calculations can be handled in several ways in Acrobat. When a calculation on a form is needed, do not rely on the user to perform the calculation. Use one of Acrobat's calculations options and lock the fields to prevent tampering with the results. (See Chapters 16 & 17 for creating calculation formulas).

Other Matters to Consider

Websites should contain information for users related to using Adobe Reader and the tools within Reader to complete forms. Websites should have a link where the Reader application can be downloaded free from Adobe Systems and help pages should be well formed and illustrated for using the Reader application.

In addition to hosting specific information about Adobe Reader there are two other items that deserve attention:

Signing Forms

Forms can be signed using a wet signature that the user applies directly in Adobe Reader. Adobe Reader supports several methods for applying signatures to forms as I explain in Chapter 15.

For constituents, you should post information on the government website regarding how a document needs signing.

Notarization

Some forms need to be notarized. There are Notaries now using certificates who can notarize PDF documents electronically (See Chapter 15 for more on eNotarization). A government website should include a description on how to acquire a notary and a list of approved notaries within a catchment area.

Developing Standards & Best Practices

There are other considerations you need to observe when designing forms. Standard line weights, white space and margins, header and footer designs, contact information, form numbering schemas, appropriate use of graphics and file formats, and more. A book of standards and best practices should be developed by a government office and adopted by all branches of government to maintain consistency.

When researching information on best practices for forms in government, try to browse different content hosted on the Internet and be certain to pick and choose items suggested by other governments and enterprises. To date, I haven't found a comprehensive solution from any organization. Some agencies make suggestions for form elements and how forms are designed. If you find agencies offering solutions for best practices and developing standards, look over sample forms the agencies host. Quite often you find some good suggestions about

designing forms but the forms coming from the agencies can be less than desirable.

An example of best practice solutions vs. actual form designs is found on the Nova Scotia, Canada website (https://www.gov.ns.ca/better-forms/better-forms-standards.asp). Nova Scotia hosts web pages containing excellent suggestions for developing forms standards and best practices. Much detail is defined for form elements and how forms should be constructed. I found much of the content helpful, but it wasn't all-inclusive for developing good forms construction standards.

In terms of forms hosting, the Home page on the Nova Scotia website contains no link to a well-organized forms page. The visitor needs to search the site to find forms. Search results report back 7,680 items found when searching for forms. The list is not well organized and requires the visitor to browse pages containing links to forms.

The PDF forms, linked to the forms pages, followed some of the suggestions made for developing standards, but not all standards were implemented. On the standards pages an item specified using Adobe Acrobat to create form fields, yet after downloading many forms, most contained no fields. The government seal was placed on all forms but the graphic was a poor scan. Form designs varied and no consistent appearance was evident among many forms.

The Australia government maintains best practices in some individual departments. Again, the information is helpful however forms hosted by the Australian Government (www.australia.gov) vary greatly in terms of being fillable, secured, and design consistency. Australia does include the government seal on all forms I downloaded and the site maintains well-organized lists for forms.

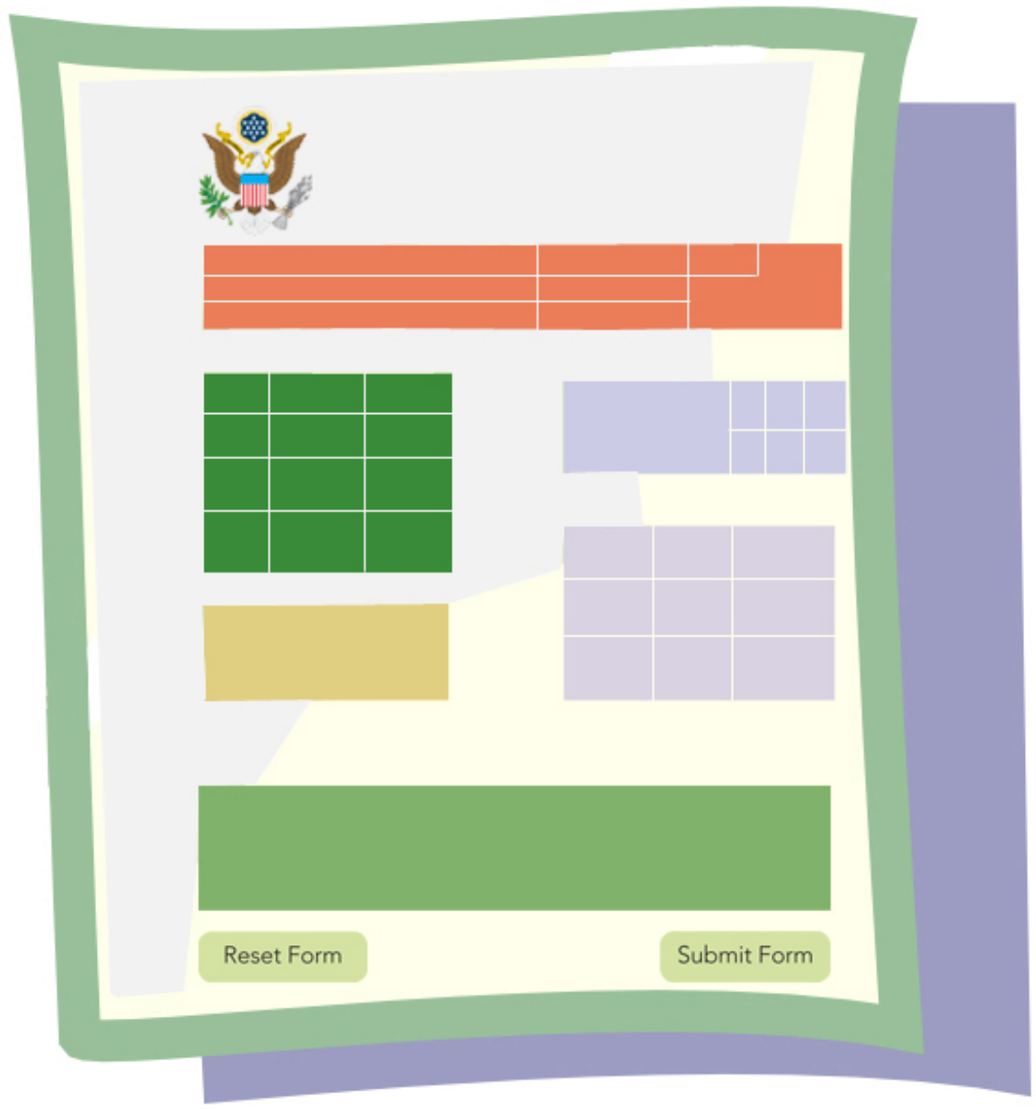
Another source that is helpful in understanding proper forms development is the Business Forms Management Association (BFMA). You find many worthwhile tips and descriptions on the BFMA website (www.bfma.org) and they provide a huge array of training programs for forms developers.

In short, I have found no model government office today that one can point to as exemplary and use for a basis to develop best practices and standards. One needs to research and take the best examples from several sources. If your office develops a model for government forms, your government could well become the leader for eForms development and management.

Chapter 6

Modifying Form Designs

Modifying the forms background design while preserving field objects



Chapter 6

IN THIS CHAPTER

Document Design Appearances

Modifying Designs

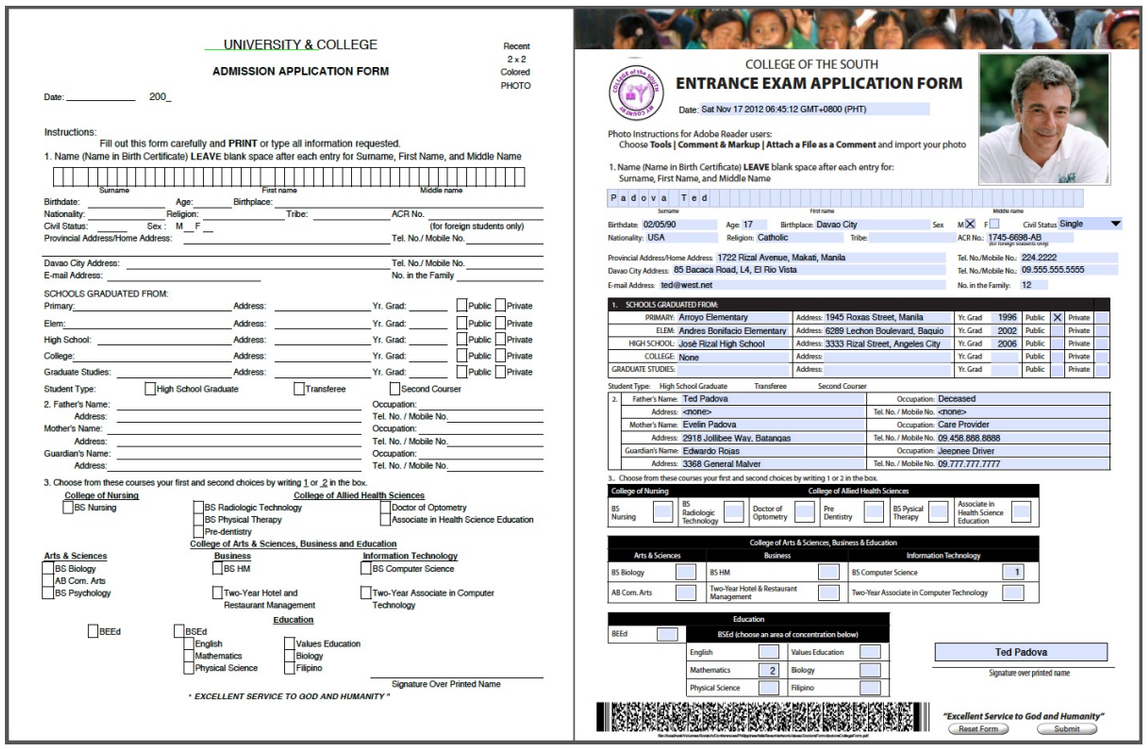
One of my greatest challenges with government workers is getting people to recognize the importance for deploying documents both attractive in appearance and well developed for easy interpretation by constituents. A form is just a document that people slap down responses on and submit back to the host. Who cares what the document looks like. If it does the job, it's good enough, right?

Wrong! Governments are competing internally and in a world economy for more business. Whether it be competitive business processing outsourcing, manufacturing, tourism, business development, exports, etc., government today is in fierce competition with its neighbors and within its own regional borders. With struggling deficits, high unemployment levels, and a global economic depression, government needs to harness all it can to attract more business.

The quality of the content you host in terms of design, appearance, usability and clarity is a statement about your office, how it's organized and how well it meets the challenge of its duties. A sloppy document, unappealing and unprofessional, communicates a statement you don't want communicated in a competitive regional or world economy.

Document Design Appearances

To illustrate a point, take a look at a form used for a public college admission. I created a makeover for the form as shown. If the forms came from two separate colleges and with all things being equal such as status of the college, courses available, prestige, geographic location, etc., what college would you choose for your child? A parent who views a college or university admission form makes a first impression from viewing the form. This impression is likely to occur before the individual interacts with personnel at a college admissions department.



It should be obvious that document design, ease of use, and ease of accessing documents are important factors when governments are competing for business.

Modifying Designs

Design revisions are commonplace when it comes to forms. If you submit forms to committees or supervisors, ultimately people want changes made on your designs. The health form I discuss in Chapter 24 went through 6 revisions before the form was finalized just for the design. Another 35 revisions were made for the field objects. Inevitably people find items that need to be added, deleted, or improved on a form. “Oh, we forgot one little thing,” is a phrase you hear often as a form designer. And, that one little thing just might take you days of recoding JavaScript on some forms.

When you create a form and populate the form with field objects, you won’t want to start over if the design background needs a change —especially if you have a complex form with many fields and JavaScript code. Fortunately Acrobat provides you easy methods for modifying form designs without interfering with field objects.

It’s important to understand that PDF forms have two areas where content are contained —the background design and the field objects. Although not specifically layers, think of the form design as a background layer and the field objects are contained on a separate foreground layer. You can edit the two layers independently.

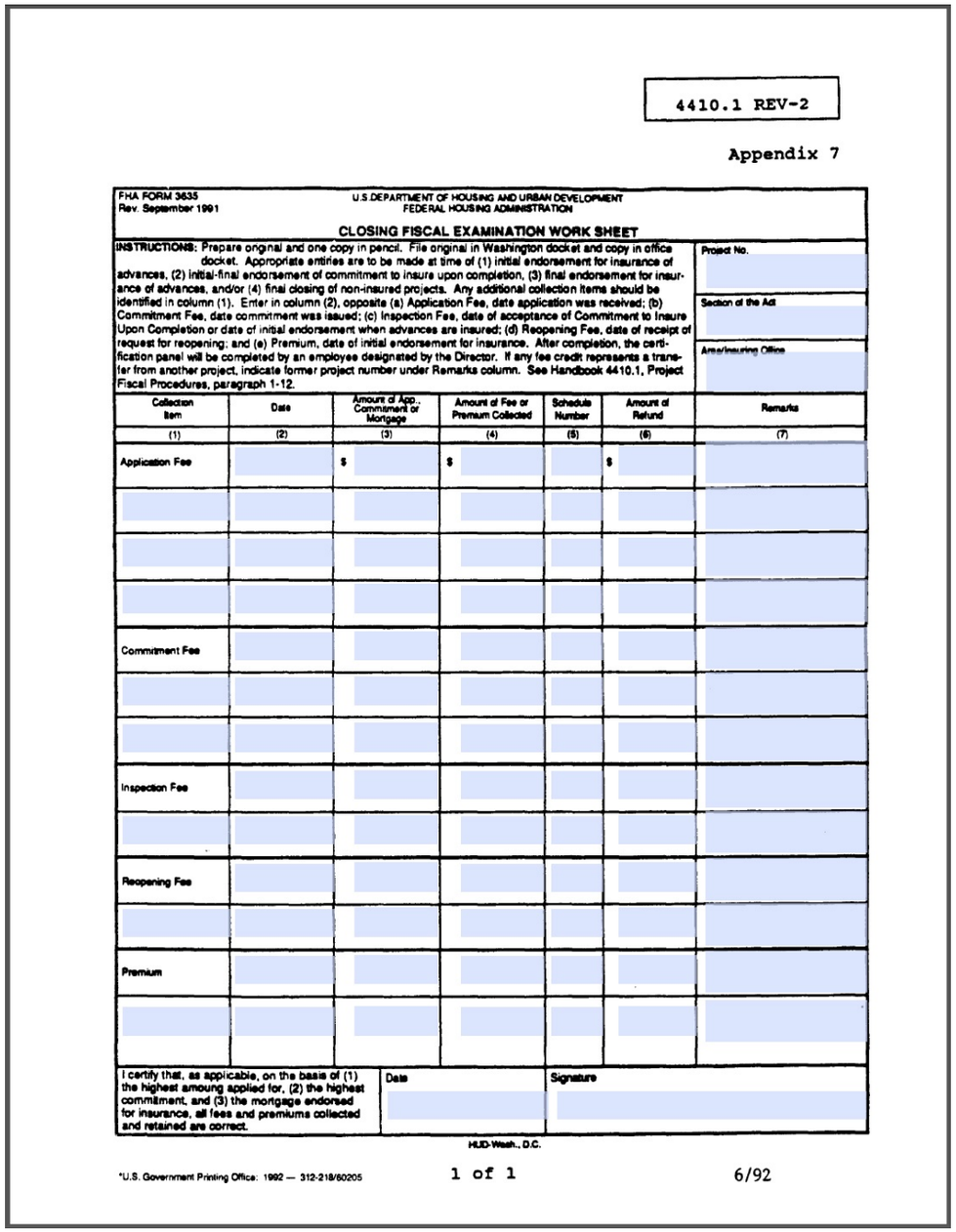
If you need to edit the form design, as a matter of practice, you should return to the original authoring application and make your edits. Do not rely on tools in Acrobat to edit text and modify images.

After you modify the form design in the original authoring application, export to PDF. In Acrobat use the Replace Pages command. Replace Pages should be an item you add to your custom tools set. If you haven't added Replace Pages to a custom tool set, as I explain in Chapter 7, open the Tools panel, click the Customize button in the top right corner of the Acrobat window, and choose Default to open the Default Tools. Open the Pages list and you find Replace Pages.

TIP

When you export a modified design to PDF, use a number preceding the document name. For example, suppose you have a form named PassportRenewal_SS341. When you make the first modification to the document save the revision as 0.PassportRenewal_SS341. The next modification is named 1.PassportRenewal_SS341, and so on. This helps you keep track of design modifications. When it comes time to replace a page you select the most recent revision to replace the current design. Since all forms named with a number at the beginning of the file name are designs only, you won't be confused about what form contains the field objects and what form is the modified design.

Figure 6.1 Original Form Hosted by the US Dept. of HUD



4410.1 REV-2

Appendix 7

FHA FORM 3635
Rev. September 1991

U.S. DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT
FEDERAL HOUSING ADMINISTRATION

CLOSING FISCAL EXAMINATION WORK SHEET

INSTRUCTIONS: Prepare original and one copy in pencil. File original in Washington docket and copy in office docket. Appropriate entries are to be made at time of (1) initial endorsement for insurance of advances, (2) initial-final endorsement of commitment to insure upon completion, (3) final endorsement for insurance of advances, and/or (4) final closing of non-insured projects. Any additional collection items should be identified in column (1). Enter in column (2), opposite (a) Application Fee, date application was received; (b) Commitment Fee, date commitment was issued; (c) Inspection Fee, date of acceptance of Commitment to Insure Upon Completion or date of initial endorsement when advances are insured; (d) Reopening Fee, date of receipt of request for reopening; and (e) Premium, date of initial endorsement for insurance. After completion, the certification panel will be completed by an employee designated by the Director. If any fee credit represents a transfer from another project, indicate former project number under Remarks column. See Handbook 4410.1, Project Fiscal Procedures, paragraph 1-12.

Project No.

Section of the Act

Area/Insuring Office

Collection Item	Date	Amount of App., Commitment or Mortgage	Amount of Fee or Premium Collected	Schedule Number	Amount of Refund	Remarks
(1)	(2)	(3)	(4)	(5)	(6)	(7)
Application Fee		$	$		$	
Commitment Fee						
Inspection Fee						
Reopening Fee						
Premium						

I certify that, as applicable, on the basis of (1) the highest amount applied for, (2) the highest commitment, and (3) the mortgage endorsed for insurance, all fees and premiums collected and retained are correct.

Date

Signature

HUD-Wash., D.C.

*U.S. Government Printing Office: 1992 — 312-218/60205

1 of 1

6/92

Original form hosted by U.S. Department of Housing and Urban Development

As an example for replacing pages in PDF documents I use a form from the U.S. Department of Housing and Urban Development. The original form was a scan from a paper form.

The appearance is horrible and not a document a government office would want to host on a website.

The form was fillable and, although the field names were not well described, the table does contain hierarchical field names. The original form does work properly, therefore, the task at hand is to simply rework the design and use the existing fields.

To rework this example I did the following:

1. **Open a design application.**
   I used Adobe InDesign CS6 for the makeover. A nice feature in Adobe InDesign is that you can import a PDF document, place the PDF on a working layer, and ghost the background to create a template. On a second layer, you create the new design. Before exporting to PDF you can trash the template layer.

2. **Rework the design.**
   In this example I created a new table. The lines are clean and structured well. I also broke the description at the top of the table into two columns. If you have large text blocks used for instructions or descriptions, breaking text into columns is easier for the user to read.

   The department seal appears at the top of the form along with text identifying the department. Both the seal and text description at the top of a form helps users know immediately where the form originated.

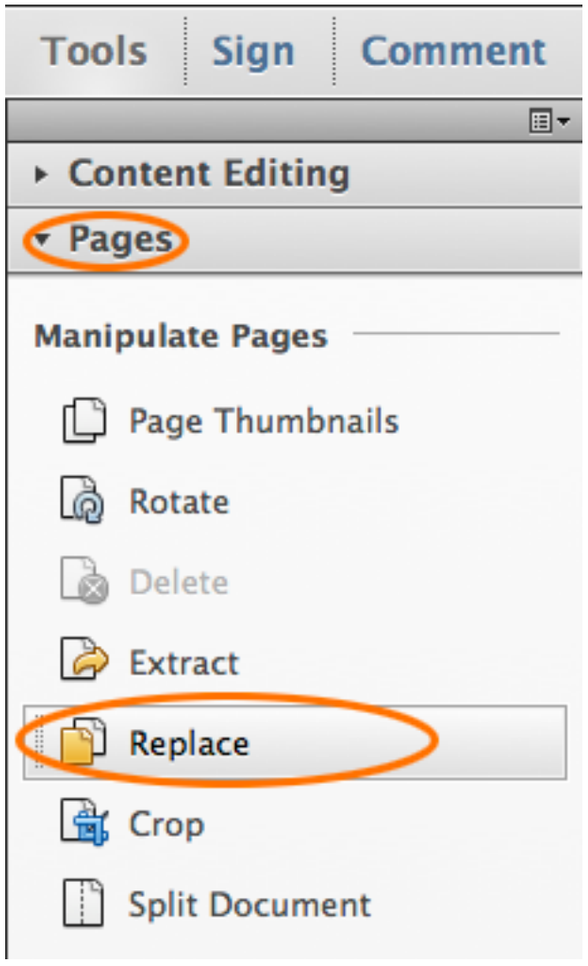
   At the bottom of the form are two buttons. The Reset Form button is used for clearing data and the Submit Form button is used for returning the form electronically to the host. These buttons were not present on the original form, however, when designing forms, these items should appear on most of the forms you deploy on your agency's website.

3. **Export to PDF.**
   When the form is completed, export to PDF. Adobe InDesign has a menu command for exporting directly to PDF.

4. **Open the original form in Acrobat.**
   Open the old form —the one containing the form fields, in Acrobat. You want this form in view in the Document Pane.

5. **Replace the Page.**
   Open the Pages panel and click the Replace Pages tool. A dialog box opens prompting you to choose the file for replacing pages. Browse your hard drive and locate the file you edited for the new design. Select the file name and click the Select button.

Figure 6.2



Open the Pages panel and click the Replace Pages tool.

Another dialog box opens prompting you for the page range to replace. The example used here is a single page form so the default text boxes for the Original and Replacement text boxes are 1. Click OK and the new design falls in behind the fields.

6.**Secure the file.**
Before deploying the form, be certain to add security to prevent a user from modifying the form.

7.**Save As to a new file.**
Choose File ➤ *Save As* and save the file as a new file name. Rewriting the file compacts the file and quite often reduces file size.

As you can see in this example, designing a simple form is not a monumental task. You don't need to be a professional graphic designer but you do need to acquire some design sense. Hosting consistent visually appealing form designs doesn't guarantee generating more business for a government office.

Figure 6.3 Design Makeover

U.S. DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT
FEDERAL HOUSING ADMINISTRATION

4410.1 REV-2
Appendix 7

FHA FORM 3635
REV. September 1991

CLOSING FISCAL EXAMINATION WORKSHEET

INSTRUCTIONS:
Prepare onginal and one copy in pencil. File original in Washington docket and copy in office docket. Appropriate entries are to be made at lime of (1) initial endorsement for insurance of advances, (2) initial-final endorsement of commitment to insure upon completion, (3) final endorsement for insurance of advances, and/or (4) final closing of non-insured projects. Any additional collection items should be identified in column (1). Enter in column (2), opposite (a) Application Fee, date application was received; (b) Commitment Fee, data commitment was issued;

(c) inspection fee , date of acceptance of Commitment to insure Upon Completion or date of initial endorsement when advances are insured; (d) Reopening Fee, date of receipt of request for reopening; and (e) Premium, date of initial endorsemant for insurance. After completion, the certification panel will be completed by an employee designated by the Director. If any fee credit represents a transfer from another project, indicate former project number under Remarks column. See Handbook 4410.1, Project Fiscal Procedures, Paragraph 1-12.

Project No.

Section of the Act

Area/Insuring Office

Collection Item	Date	Amount on App., Committment or Mortgage	Amount of Fee or Premium Collected	Schedule Number	Amount of Refund	Remarks
(1)	(2)	(3)	(4)	(5)	(6)	(7)
Application Fee						
Commitment Fee						
Inspection Fee						
Reopening Fee						
Premium						

I certify that, as applicable, on the basis of (1) the highest amoung applied for, (2) the highest commitment, and (3) the mortgage endorsed for insurance, all fees and premiums collected and retained are correct.

Date

Signature

HUD - Wash., D.C.

RESET FORM SUBMIT FORM

1 of 1, 6/92

Form makeover after replacing pages

Rather, it is one small component in the overall way you approach communicating professionalism and excellence in your department.

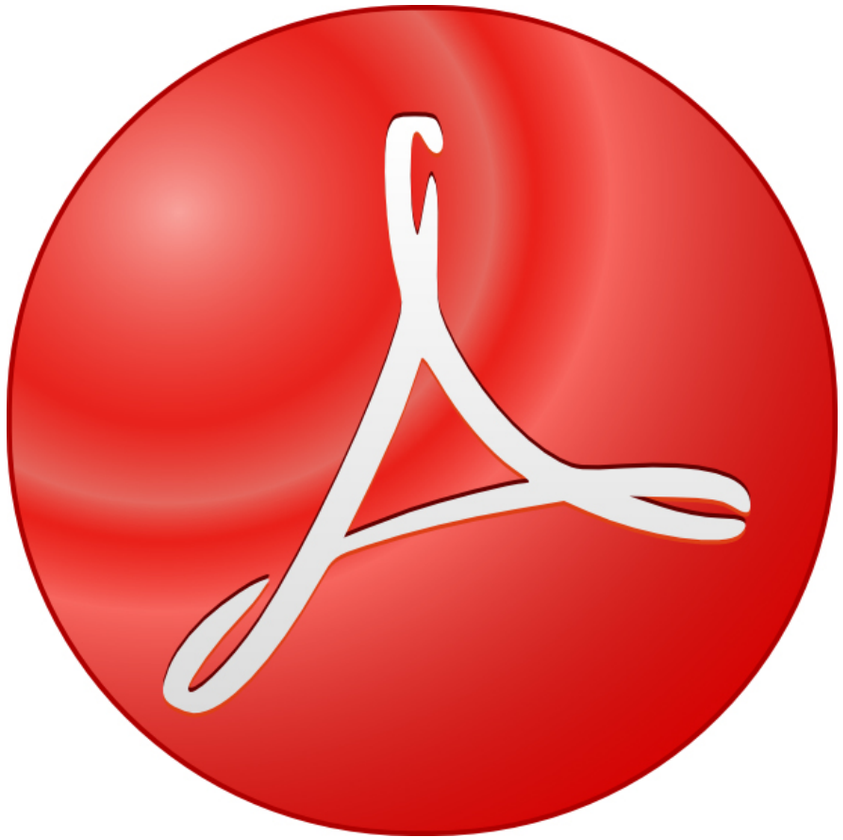
A few decades ago, Apple CEO, Steve Jobs was asked in an interview to comment about Microsoft. Jobs told the interviewer the problem he had with Microsoft is “they have no taste.” Jobs was relentless in delivering products with exceptional design and minute detail for appearance, font usage, and style. Today, Apple, Inc. has the largest market cap of any company in the world.

Apple, Inc. is on a path to become the world’s first trillion-dollar company. Go to a movie today and it’s hard to see a new film that doesn’t display an Apple product in at least one scene. Movie producers aren’t interested in performance of consumer electronic devices. What they want are objects in film for aesthetic purposes only. When it comes to aesthetics, Apple, Inc. is the world’s leader and something that has spearheaded the company in becoming the wealthiest company in the world.

Chapter 7

Understanding Acrobat Viewers

Working with Acrobat viewers in version XI



Understanding Acrobat Viewers

Chapter 7

IN THIS CHAPTER

Adobe provides two types of PDF viewers. The commercial product Adobe Acrobat XI Pro and a lighter commercial product Adobe Acrobat XI Standard (Windows only) are software products you need to purchase from Adobe Systems.

Adobe also provides a free software product, Adobe Reader, that permits users to view PDFs and fill out forms. If you are a forms author you need to purchase Adobe Acrobat XI Pro. Both Adobe Acrobat and Adobe Reader are cross-platform and almost equally supported on both Macintosh and Windows computers.

For the purposes of this book, I refer to the commercial product simply as Acrobat and the free application simply as Reader.

PDF stands for Portable Document Format. Files you view and work on in Acrobat and files you view and add data to forms and save the forms in Reader are PDF files. Note that Acrobat also supports filling in forms and saving forms with form data.

Authoring Forms in Acrobat

Adobe Acrobat was introduced originally as Acrobat Exchange in 1991. The program has experienced many changes through the years and currently, as of this writing, we are now in Acrobat XI (version 11).

For any kind of form authoring you should use only Acrobat XI. If you haven't upgraded to Acrobat XI, do your best to convince purchasing authorities that you need the most recent upgrade. Acrobat XI not only offers more features than previous versions of Acrobat, it has many fewer bugs and problems that are most noticeable for forms designers.

When designing forms, you typically work in an authoring program like MS Word, Adobe InDesign, or some other application that supports placement of text, objects, and creating rules (lines). When you finish creating a form in an authoring program you convert the native file to a PDF document. Acrobat itself provides you tools for performing such conversions that you can use if your authoring program does not support export to PDF. However, most of the popular programs such as Adobe InDesign, Adobe Illustrator, Microsoft Word, MS Excel, IBM Lotus Symphony, OpenOfficeOrg Writer and many other authoring programs provide you tools for exporting direct to PDF.

For design creations I find Adobe InDesign the best authoring application for forms. However, if you are most familiar with MS Word, you can create your original designs in Word and save the file as a PDF. For government offices, it's best to standardize on a single authoring program. This helps with technical support services and makes training new employees much easier.

Once you open a PDF in Acrobat you add field objects to a form. Fields are best added and managed in Acrobat, however programs such as Word and Adobe InDesign CS6 do provide options for adding fields. As a matter of practice though, you should plan on adding all fields in Acrobat.

For optimum workflow operations in larger offices it may be best to have personnel who are charged with designing forms and another department creating forms with fields and writing JavaScripts using Acrobat.

A good many problems in government and enterprise is that office workers charged with responsibilities for creating memos, letters, reports, briefs, etc. have been assigned duties for creating forms. These office workers often do not have design backgrounds and are typically not knowledgeable about how forms need to be constructed. Designing forms is a specialty and requires people with some background and experience with handling forms.

Using Adobe Reader

For a free application, Adobe Reader is remarkable. You can distribute PDF forms in an office and the entire office personnel can download Adobe Reader free for each computer. Once PDFs are opened in Reader, forms can be populated with data and the files can be saved with the data intact.

Reader supports searching PDFs; searching through catalogs of PDF files; creating and using digital signatures; and in Reader XI, a full compliment of comment and markup tools are available without enablement.

For government eForms, constituents don't need to purchase any additional software. They can fill out your forms and submit forms with data back to your host agency.

PDF Viewers and Handheld Devices

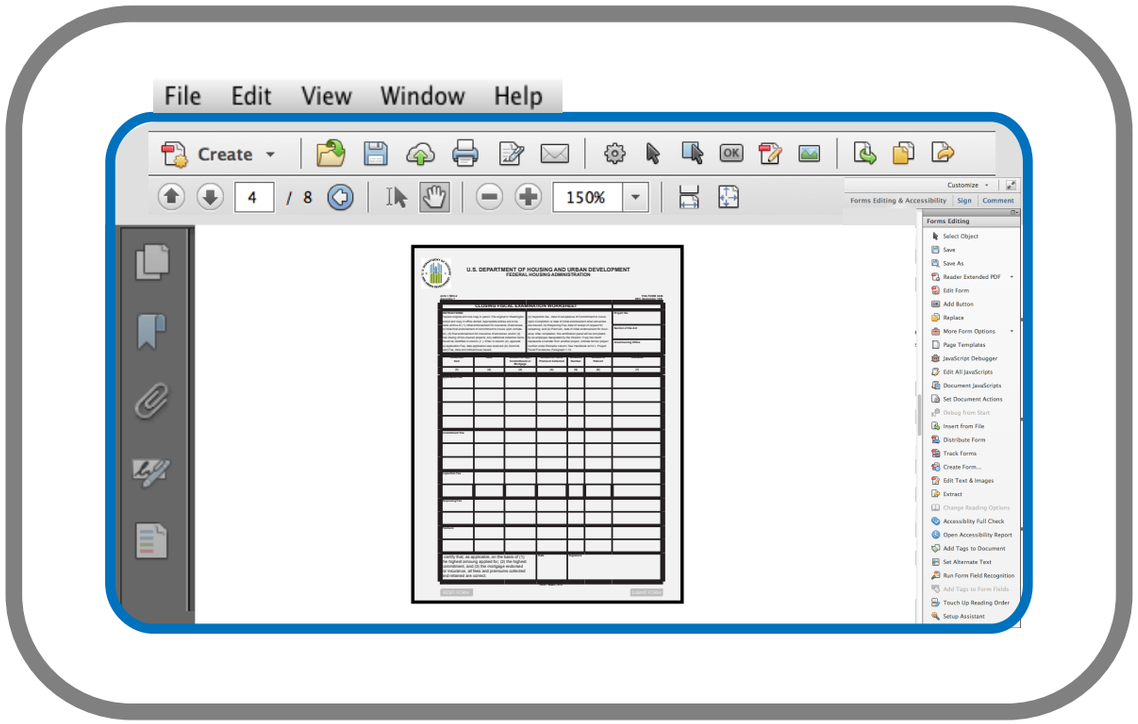
Each year we move closer to form fill-in on mobile and handheld devices. There are some PDF viewers that are available for Android and iOS devices that provide options for form fill-in and signing. However, JavaScripts you may use in a form are not yet supported for handhelds.

You may wish to continually browse the Web searching for new products and updates as they occur. Sooner or later we should see a PDF viewer that supports all the fill-in options you have with Adobe Reader.

Chapter 8

Exploring the Acrobat Interface

Getting around the Acrobat tools and menus



Exploring the Acrobat Interface

IN THIS CHAPTER

Chapter 8

Acrobat is a complex application that serves the needs of several industries. You find tools for comment and review, professional printing, legal services such as redaction, scanning and OCR recognition, conversion methods for documents and web pages, editing PDFs, working with interactive elements, encryption tools, searching, spell checking, signing documents, comparing documents, exporting content and more.

Fortunately for forms designers you don’t need to learn all the various features in the program to create sophisticated, dynamic, and complex eForms.

Working with Preferences

The very first thing you need to do after launching Acrobat XI is visit the application Preferences. You find many toggles and switches in a long list of preferences in Acrobat. Most important are preferences located in the General and Forms categories.

Launch Acrobat and open the preferences by pressing the CTRL (Windows) or the Command (Mackintosh) key + K for a keyboard shortcut. Alternately you can open the Acrobat menu (Macintosh) and click Preferences or Edit menu (Windows) and click Preferences.

Setting General Preferences

The Preferences window opens and you find a list on the left side of the window with Categories listed. Click General in the left pane and the right pane changes. Click the check box where you see *Use single-key accelerators to access tools*.

Checking this box enables you to use keys on your keyboard to access tools. It will take some time to remember all the keyboard shortcuts but there are a few you can easily commit to memory that help speed up your work in Acrobat. These include:

- **Hand tool (H).** After you check the item in the preferences, individual keys you press on the keyboard select a tool. When pressing the H key the Hand tool is selected. You use this tool to move a document around the document pane. If you zoom in on a form, you see only a portion of the form in the Document pane. Click and drag the Hand tool around the page to move the page content within the document pane. Additionally, when you fill-in form fields, you use the Hand tool.

- **Select Object tool (R).** As the name implies, you select objects with this tool. A field is an object. You can use the Select Object tool to select field objects for copying/pasting, resize field objects, manage fields by aligning and distributing, and double-clicking a field to easily access a field's properties. This is perhaps the most essential tool you find for authoring forms.

Just remembering these two tool keyboard accelerators can help you work much faster in Acrobat when designing forms. The remaining items in the General Preferences pane can be learned as you become more knowledgeable about Acrobat and other features in the program.

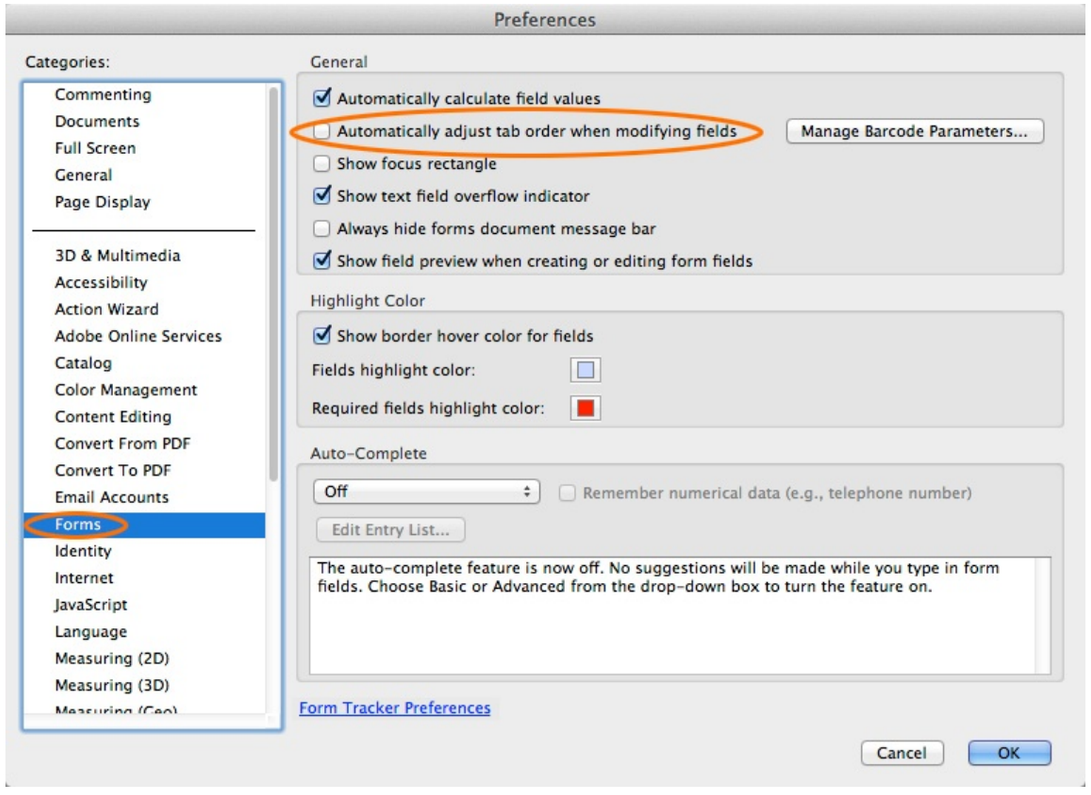
Setting Forms Preferences

Acrobat provides a special category in the Preferences for toggles and switches related specifically to forms. In the left pane click Forms and the right pane changes to reflect options you have for setting forms preferences.

There is one item in this category that's paramount for forms designers and a new preference option that was introduced in

Acrobat XI. Click Forms in the left pane and notice the item denoted as *Automatically adjust tab order when modifying fields*. You must remove the checkmark for this preference item. If you leave this item checked, Acrobat will operate painfully slow when working on complex forms. Always keep this item unchecked. If you need to adjust tab orders on your forms you have many other alternatives I discuss in Chapter 10.

Figure 8.1 Forms Preferences



Press CTRL/Command + D. Click Forms in the left pane and remove the checkmark

The remaining items in the forms preferences are personal choices for viewing forms and fields. You can explore these items independently. For now, just realize the most important item to address is turning off the auto tab order setting.

Setting Language Preferences

If your forms are assembled in a language other than US English, you can click the Language item in the category list and choose a language from the Application Language drop-down menu.

The remaining preferences can be left at defaults when you installed Acrobat. As needs arise you can periodically visit the preferences to make changes. For help with preference choices you can open the Help menu and click Adobe Acrobat XI Pro Help.

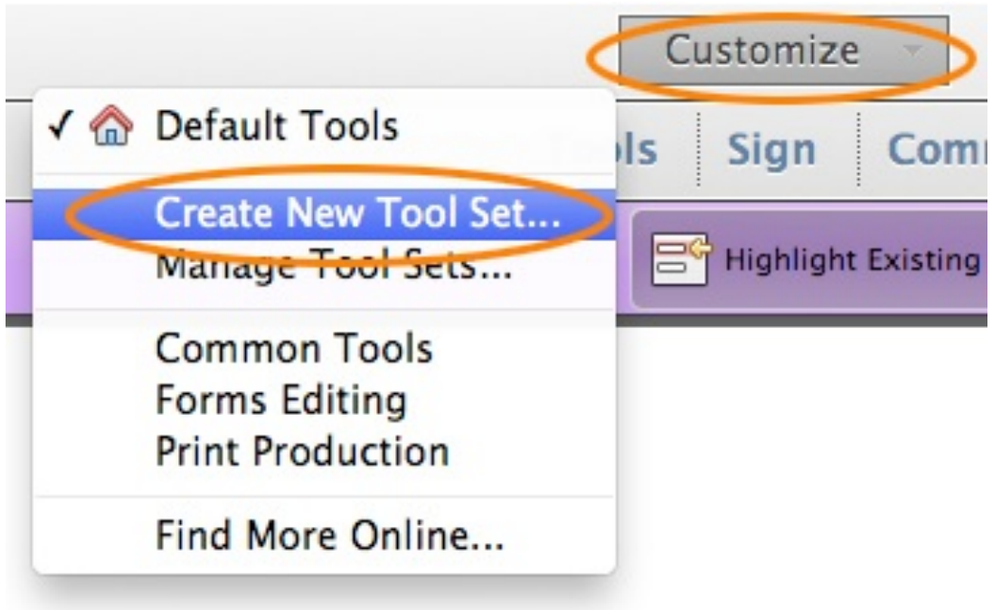
Adding Tool Sets

If you're familiar with other Adobe applications you know that most programs support creating Workspaces. A workspace is a collection of tools and menu choices that a user personally defines. You make choices for the contents of your workspace and save it. You can create additional workspaces and save each one independent of the others. If using a program like Adobe InDesign you might create a workspace for designing print documents and another for designing web documents. You might use tools that support CMYK color for print and RGB color for web and save to the respective workspace.

In Acrobat terms, workspaces are known as Tool Sets. You likewise determine what tools you want in a given tool set and save the set. You can likewise create several different sets. If you use Acrobat regularly and you work with reviewing documents and you also design forms, you might save two different sets of tools —one for Comment & Review and the other for Forms.

For our purposes we're talking about forms. Therefore, you'll want to create a tools set that contains all the tools you use when designing forms. To begin developing a tool set you start by clicking the Customize button in the top right corner of the Acrobat window to open a drop-down menu. Choose Create New Tool Set from the menu options.

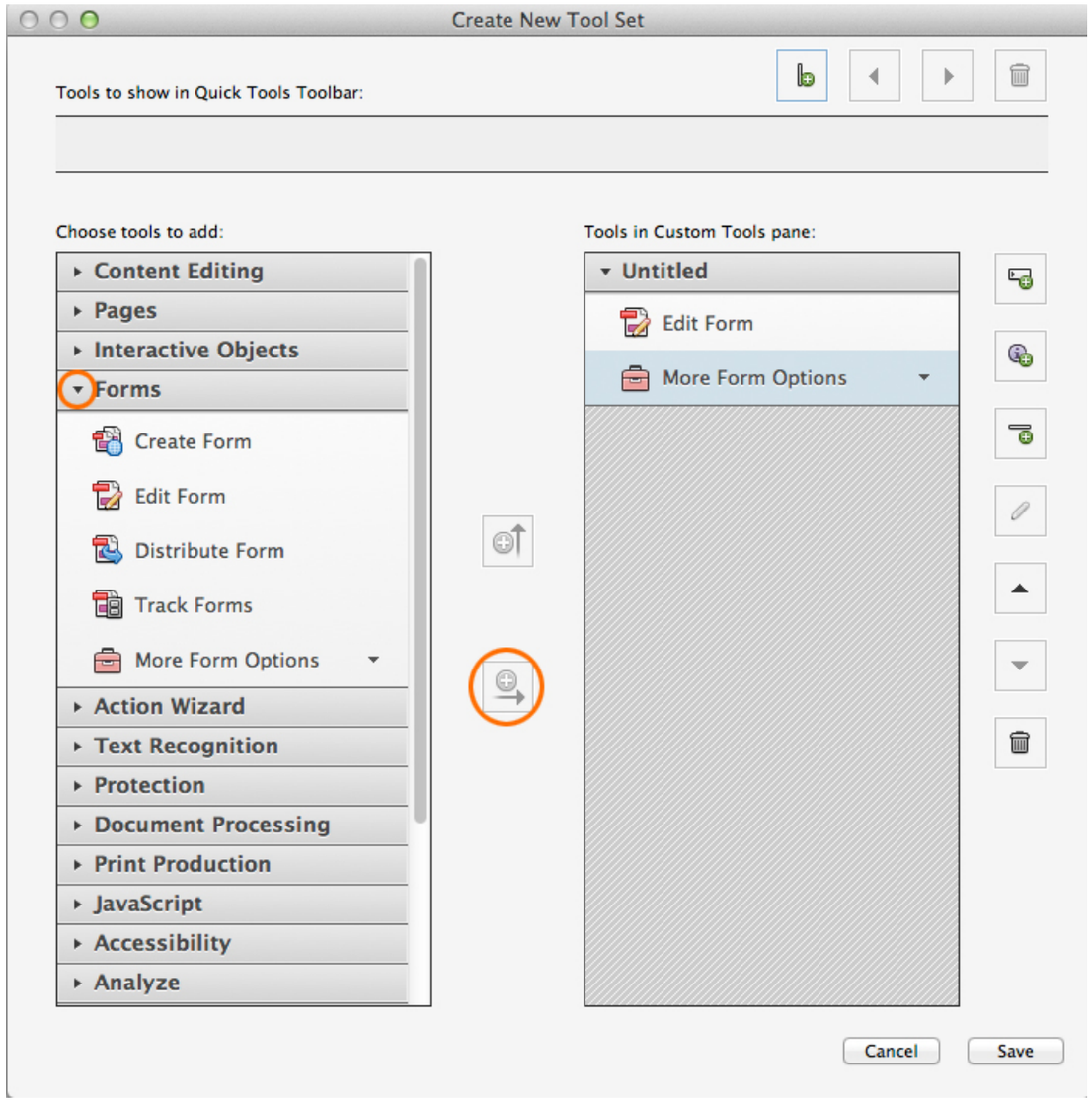
Figure 8.2 Create New Tool Set



Click Customize and choose Create New Tool Set

The Create New Tool Set window opens. Here you find a list of all available Acrobat tools listed in the left panel, and by default you find an empty panel on the right where you see the caption *Tools in Custom Tools pane*. The right pointing arrowheads adjacent to the tool categories in the left pane are used to expand/collapse a category.

Figure 8.3 Add Tools to New Tool Set



The Create New Tool Set Window

Obviously you want to add all the items listed in the Forms section. Click the right pointing arrow and click a tool in the list. To move the tool to the right pane, click the right arrow icon appearing between the two panes. Follow the same steps and add all the tools so they appear listed in the right pane.

Other tools you'll want to add to your Forms Editing Tool Set include:

- **Select Object tool:** You find this tool located in the Interactive Objects category.
- **Save:** This is optional if you happen to know the keyboard shortcut (CTRL/Command + S) to save and update a file.
- **Save As:** Likewise this may be an optional choice. This command is important when creating complex forms. You may want to save a copy of a form to test JavaScripts and formulas without disturbing your final form.
- **Replace:** This command is used for replacing pages. I use this command virtually every time I create a new form. I find there are many times I need to return to the original document, make some edits, and convert to PDF. If I add field objects on a form, I want the fields to remain but I want the background design with my new changes appearing on the form. If this is the case, I choose Replace and a dialog box opens asking me to locate the file I want to use to replace the existing page or a range of pages.

  You find the Replace tool in the Pages category.
- **Insert from File:** Also included in the Pages category is the Insert Pages from File tool. You can use this tool to insert a blank page when you want to test calculations, JavaScripts, and button actions.
- **Page Templates:** Page Templates are used when you want to add new pages to a document that are spawned from template pages. I cover more on spawning pages from templates in Chapter 18.

  You find the Page Templates tool in the Document Processing category.
- **JavaScript:** Open the JavaScript category and add all the tools contained in this category.
- **Accessibility:** To comply with Section 504 in the USA you'll want to add accessibility to all your forms for USA government sites. You may need to do quite a bit of editing for tagging files properly, adding alt text, and insuring that the reading order is correct for screen readers. To add Accessibility tools, open the Accessibility category and add all the tools to the right pane.

TIP

Acrobat has a Read Out Loud feature that can help you determine reading order before forms are tested on screen readers. Choose ***View ➤ Read Out Loud ➤ Activate Read Out Loud****. Your form is read out loud. You can control the reading speed by pressing the 0 to 9 keys on your keyboard.*

For creating government eForms, the above listed items are the most important. Acrobat provides other tools that might be used by forms designers such as distributing a form to Office 365 or a Microsoft Sharepoint server. It's more likely though that in government offices you won't use the distribution choices from Adobe.

In the Save & Export category you find the Reader Extended PDF tool. Since you should require all users to upgrade to Reader XI, you won't need to enable PDFs with Reader Usage Rights. If, for some reason, you need to support the needs of Reader users below Reader XI, you may want to add this tool to the right pane.

Adobe has spent much engineering time and effort in simplifying the forms editing process and has tossed in some items such as the Create Form tool that takes you to Adobe Forms Central where forms can be designed from an assortment of templates. Again this is an item not likely used by government forms designers.

If you stick with those items recommended for adding to a Forms Editing tool set, you have all the essential tools necessary for designing dynamic forms.

If the order in the right pane in the Create New Tool Set window is not the order you want, you can select items in the right pane and click the up/down arrows to reorder the tools. When finished, click the Save button and your new tool set is added to the Customize drop-down menu.

Once you select a tool set, the tool set remains open in all your Acrobat work sessions. If you want to change to another tool set, just reopen the Customize menu and select another tool set.

Adding Tools to the Toolbar Well

Above the Document pane you find a number of tools in a toolbar spanning the width of the Acrobat window. A few tools not loaded by default can help when designing multi-page forms. Open a context menu (right click anywhere on the Toolbar Well or with a one-button mouse on the Macintosh, press Control and click to open a context menu) and choose from the menu *Page Navigation ➤ Previous View*. Return to the same menu and choose Next View. If you view a page such as page 1 in a document and jump to page 5 (or another page), click the Previous view tool to return to the page before the jump. This saves you time over manually scrolling pages.

If you find additional tools that work well for you, you can examine tools in a context menu and add them to the Toolbar Well in similar fashion.

Zooming Views

A common task when editing forms is zooming the view. You may want to zoom in on a field to precisely size a field box, then return to a Fit View to examine the entire form page.

You find the minus (-) and plus (+) symbols loaded in the Toolbar Well by default. Additionally you find a text box for typing a zoom view, a drop-down menu to choose fixed zoom views, and page icons for Fit Window to width and enable scrolling and Fit one full page to window. You have a number of other tools you can add for zooming views that can be added to the Toolbar Well.

Perhaps the easiest options for zooming views is just remember a few keyboard shortcuts. Using the shortcuts enables you to work much faster in Acrobat. The most important for zooming views include:

- **Ctrl/Command + (plus):** Press the Control key (Windows) or Command key (Macintosh) and the plus (+) key to zoom the view.
- **Ctrl/Command - (minus):** Press the Control key (Windows) or Command key (Macintosh) and the minus (-) key to zoom out.
- **Ctrl/Command 0 (zero):** Press the Control key (Windows) or Command key (Macintosh) and the zero key to zoom to a full-page view.

Try to commit these shortcuts to memory and you can quickly zoom in and out of pages in the Document pane.

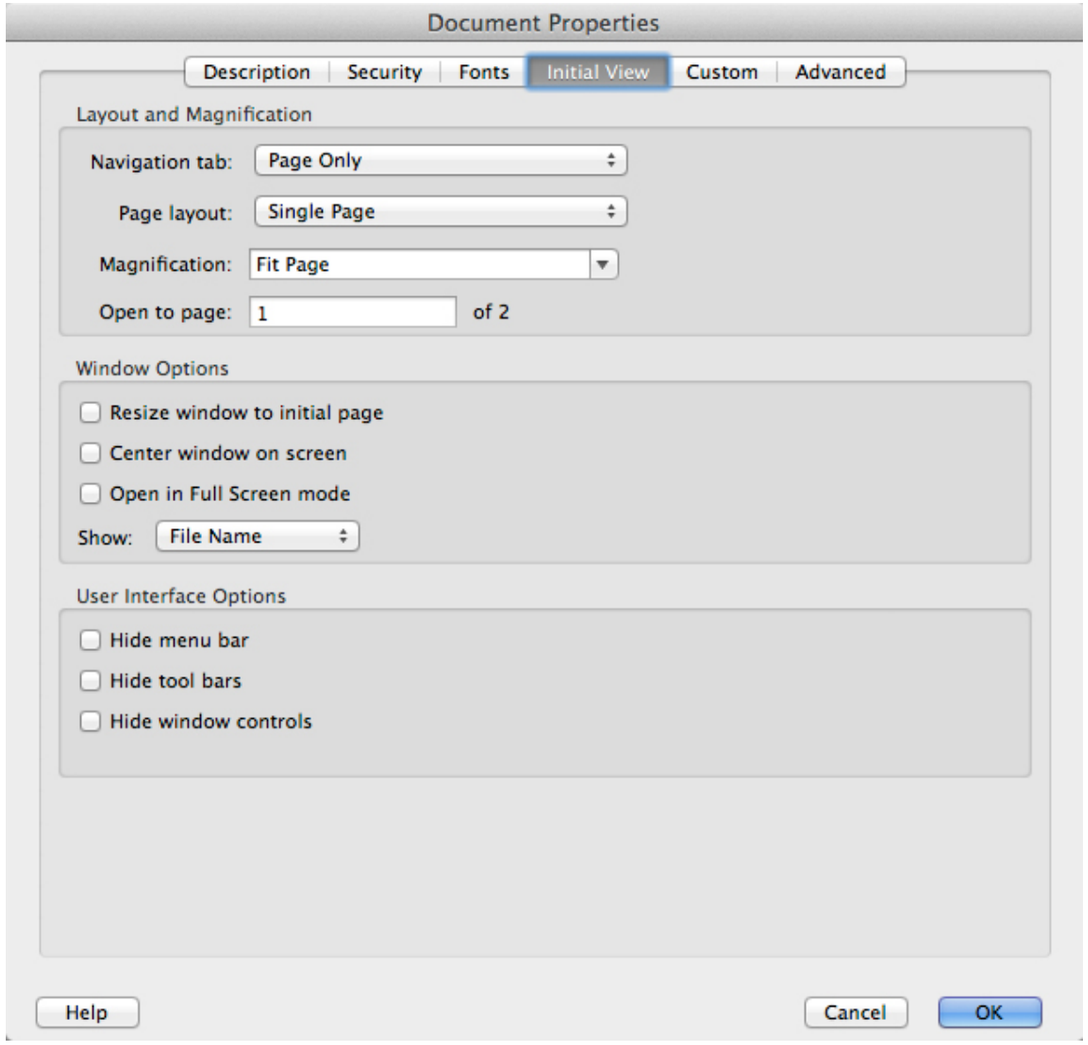
Setting Initial Views

An initial view is the zoom view when a document is opened in Acrobat or Reader. The view can also show or hide toolbars and show or hide panels. You can make choices for how you want constituents to view your form when the file is opened.

The toggles you set are strictly subjective. You may want all your forms to appear zoomed in so the form filler can clearly see the fields on small monitors. You could also set the view to Fit Width so the form zooms to a screen width. Another choice is to Fit View where an entire page is shown when the file opens. Regardless of the choice you make this item requires some discussion in a staff meeting and some standards should be set forth so all forms hosted by your government office are set to the same zoom view.

Setting the opening view is handled in the Initial View window. To access the Initial View settings, choose *File* ➤ *Properties* or press CTRL/Command + D. When the Document Properties window opens, click the Initial View tab.

Figure 8.4 Initial View Properties



Set the Initial View for forms that you want appear as the default view when the form is opened.

When you first open the Initial View window you find a number of drop-down menu items are set to Default. Be certain to override all the Default choices. The Default choice pertains to Preference choices made by individual Reader users. Some users may have defaults set to open Fit Width, some to Fit Page, and others to Automatic. The result is that your document is displayed differently on different computers according to the preference choices each user makes.

Default views would be fine if all Adobe Reader users knew they could change a viewing preference. However, most users are not aware of the toggles. It may be best for your department to make the choice for what viewing option is best for most users. Experienced users can always change the view once inside Reader or Acrobat using keyboard shortcuts and tools.

In the Initial View window you have several choices to make that can help the viewing experience for users. These include:

- **Navigation tab:** Typically you might set this option to Page only. However, if you have bookmarks created for long forms, you may want to choose Bookmarks Panel and Page from the Navigation tab drop-down menu. You may have some forms that are multi-lingual and the different languages may be on separate layers in your form. If this is the case you may want Layers Panel and Page opened when the user views the file in Reader. Again, you make the choice in the Navigation tab drop-down menu.

- **Page Layout:** Should the form be viewed as a single page or should continuous pages be enabled? I personally prefer using a single page view. When viewing pages in this view, the entire page snaps to view. When you scroll pages each page snaps to a view in the Document pane. If you choose

Single Page Continuous, users can scroll pages without snapping to a full page. In essence, one can see the bottom half of the top page and the top half of the bottom page in the Document pane. This can be an advantage for a user if they need to see a previous set of fields that might be helpful in completing the next set of fields appearing on a following page.

Page layout is another subjective judgment and warrants discussion in a committee meeting for standardizing forms.

- **Magnification:** This is where you set the open default zoom view. You have several choices from the Magnification drop-down menu. Make a choice that is the standard for your government office.

- **Open to Page:** By default, the number is set to 1 where the first page in the file appears in the Document pane when a file is opened. You may want to change this item to open another page where a user may logically start but the form is best viewed after populating a form from page one.

  As an example, I created a health statistics form for the Botswana Africa Ministry of Health. The form opened page 225 where a registration card for a new in-patient was recorded. However the important part of the form was the discharge information that resided on page 2.

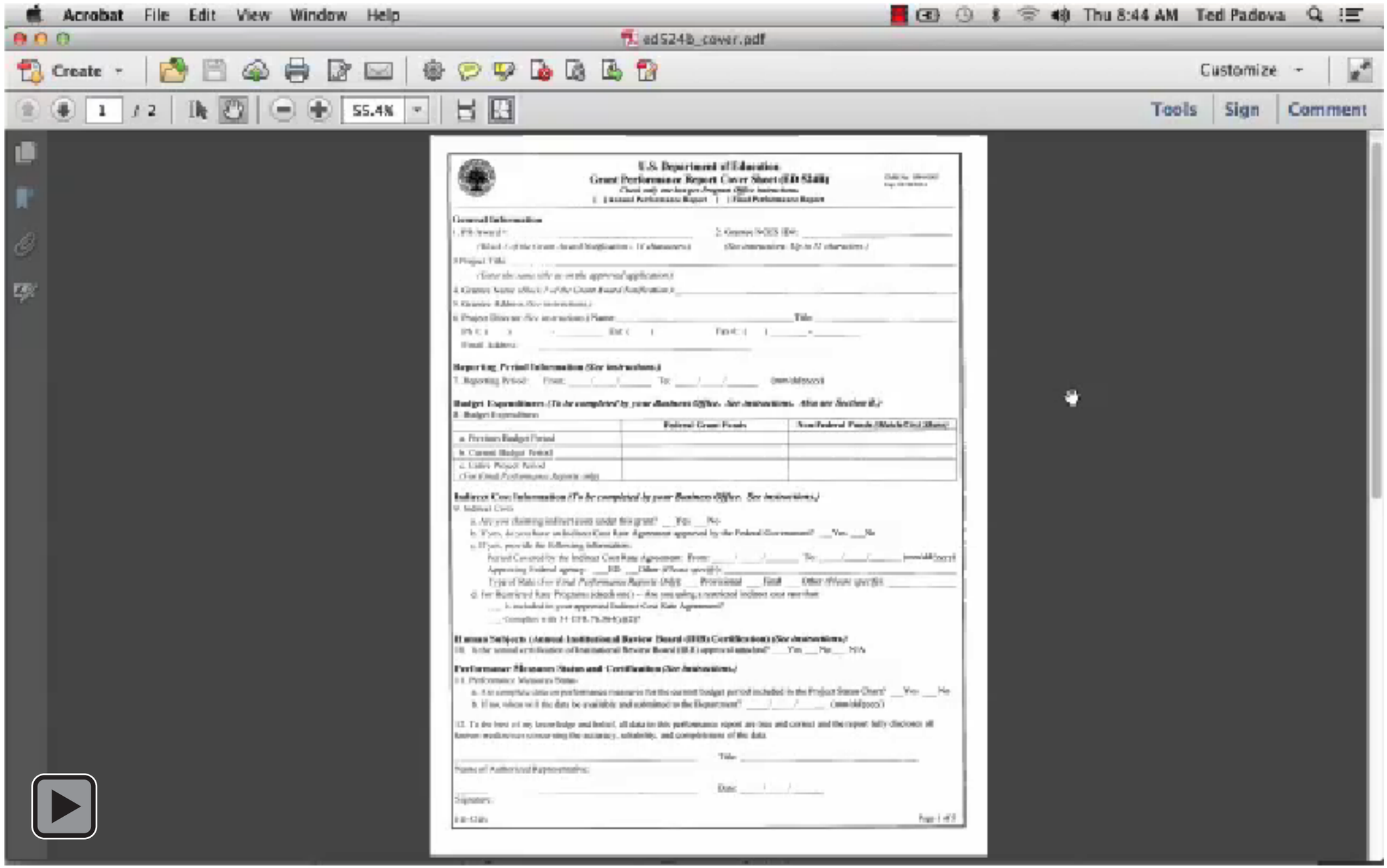
  Discharges occurred after the registrations. When a patient was discharged the data from the registration card was transposed to the patient discharge form. Data always had to be sent from the registration card to the discharge form. There was no way to populate registration information manually on the discharge form. Hence, since registration cards were always viewed first, either to add a new patient or discharge a patient, it made sense to open the PDF on the registration card page first —page 225.

  In terms of the design for this example, I could have designed the document with the registration cards appearing at the beginning of the form and the discharge at the end of the form. However, the registration cards were an afterthought when the health professionals met and requested additional information on the form. Extensive JavaScripts and calculations were made that referenced page numbers so it was much easier to modify the form by adding registration cards at the end of the form. All of this was transparent to the workers since pages opened automatically when data were transposed.

- **Window Options:** For eForms, all these items should be left unchecked and at defaults.

- **User Interface Options:** You can choose to hide items listed in this section. You may want to avoid hiding menu bars. Users may want to work with menu commands for saving and closing your forms. Hiding tool bars and window controls provides more real estate for users to work with forms. For eForms, the tool bars and window controls may be unnecessary for most forms. If you do choose to hide items, be consistent for all forms your agency deploys.

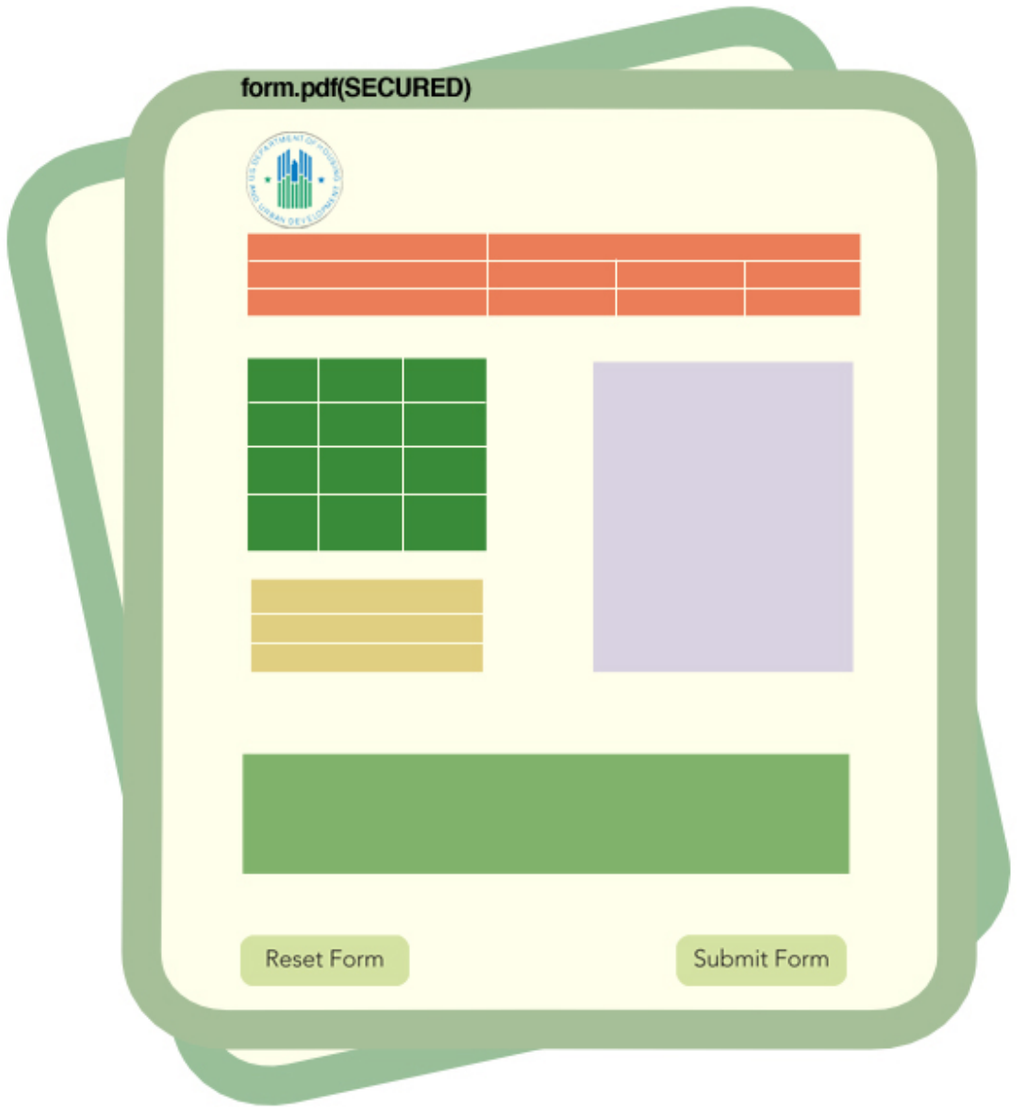
Movie


Acrobat File Edit View Window Help
Thu 8:44 AM Ted Padova
ed524b_cover.pdf
Create
Customize
1 / 2
55.4%
Tools Sign Comment
U.S. Department of Education
Grant Performance Report Cover Sheet (ED 524B)
General Information
Budget Expenditures
Federal Grant Funds
Non-Federal Funds
a. Previous Budget Period
b. Current Budget Period
c. Cumulative
Indirect Cost Information
Human Subjects
Performance Measures Status and Certification
Signature:
Date:
Page 1 of 3

Chapter 9

Understanding Form Elements

Field objects contained in PDF forms



Understanding Form Elements

Chapter 9

In the previous chapters we looked at some issues related to the status of government forms and an overall view of tools and elements in the Acrobat interface. Now it's time to get down to understanding precisely how forms are constructed. You have the background information in the first eight chapters. From this chapter forward I talk about creating forms in Acrobat.

IN THIS CHAPTER

Entering Form Edit Mode

Acrobat has what we might think of as two separate interfaces. All the edits and work you do with everything other than form editing is handled in the default space I refer to as Normal view.

When you first begin creating a form, you enter a different workspace referred to as Form Edit mode.

Switching to Form Edit Mode

To begin creating a form you need to start with a PDF file. As I stated earlier in previous chapters you first create a form design in another program such as MS Word, Adobe InDesign, or some other authoring application. You then convert your layout to a PDF file. Finally you open the PDF in Adobe Acrobat.

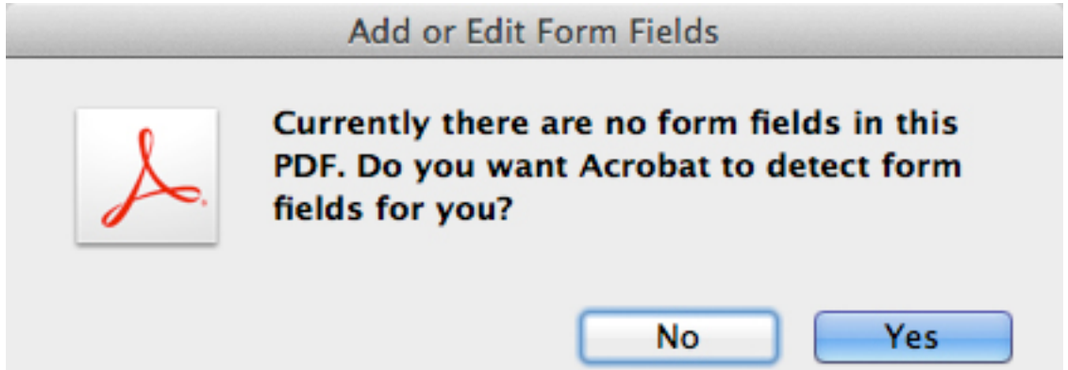
Once in Acrobat, you use the Tools panel to address form editing. If you created a custom tool set you should see the Edit Form item in your tool set. If you haven't created a tool set you can find the Edit Form tool in the Default tool set in the Forms category.

Note

From this point forward I assume you have made a custom tool set with the tools added as described in Chapter 8. If you have not created a custom tool set for forms, look over Chapter 8 and add the tools as recommended in that chapter.

Open a PDF document and click Edit Form in the Tools panel. If this is the first time you enter Form Edit mode on a new form a dialog box opens asking you if you want Acrobat to detect fields for you.

Figure 9.1 Add or Edit Fields Dialog Box



When you first enter Form Edit mode on a form containing no fields, Acrobat prompts you for auto-detecting fields.

Understanding Auto Field Recognition

The dialog box appearing when you first enter Form Edit mode on a new PDF document offers you an option for Acrobat to automatically recognize fields. Through several recent versions of Acrobat, Adobe has attempted to offer novice users easy methods for working with forms. Auto field recognition is an example of one effort to help novice users create simple forms.

You, however, are now or soon to be, a professional form designer. As such you don't need Adobe's simplistic methods for working with forms. As a matter of fact, it will be

counterproductive for you to permit Acrobat to automatically create fields.

There are a few reasons why you should avoid using auto field recognition. First, the fields Acrobat automatically creates on a form are not perfect. Acrobat may add fields in locations where no field should appear. Radio buttons and checkboxes may not be created properly. Secondly, all the fields Acrobat automatically creates are named without a hierarchical order. If you leave the field creation to Acrobat you'll need to change the names on most of the fields.

In short, NEVER, EVER click Yes when you start a new form. Always begin a new form with field objects you create manually. When creating basic to complex forms you can actually move faster through the form editing process by creating fields manually.

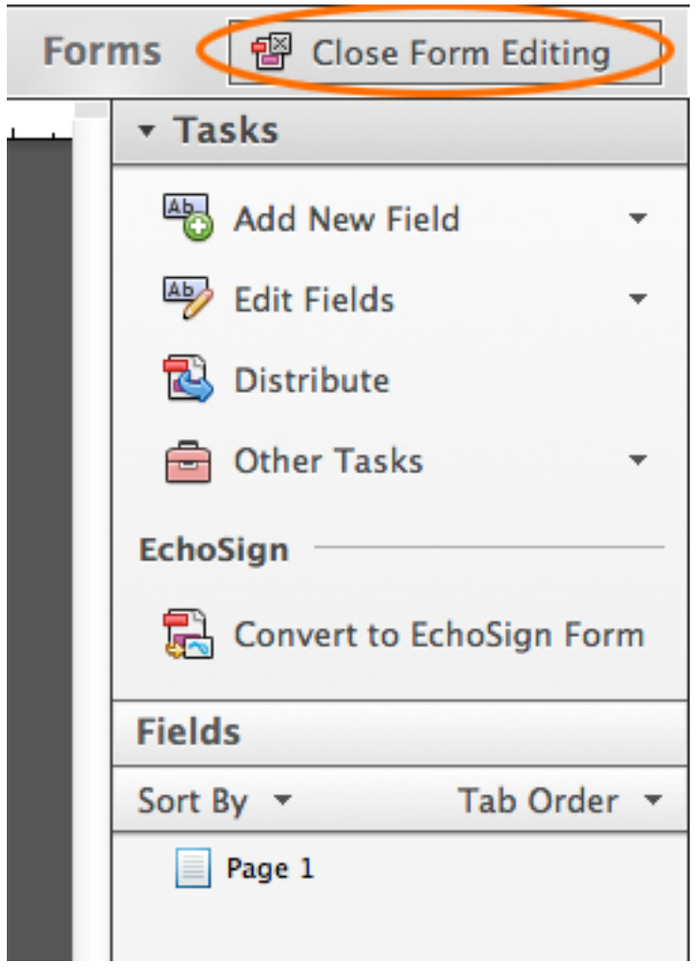
That said, click No in the dialog box that opens when you first open a form. After you add a field to a form, the dialog box won't appear when you open the form. This is a one-time experience that happens when you open a new form that does not contain any field objects.

Understanding the Form Edit Mode Interface

The first thing to notice when you enter Form Edit mode is the Close Form Editing button in the top right corner of the Acrobat window. You click this button to return to Normal view. Be aware that you need to exit Form Edit mode when you want to test your form. During an editing session you may frequently toggle between Form Edit mode and Normal mode. Be certain to remember how you enter Form Edit mode and how you return to Normal view.

There are two primary areas you need to understand when editing forms. You use field object tools and you use the Tasks panel to complete your forms.

Figure 9.2

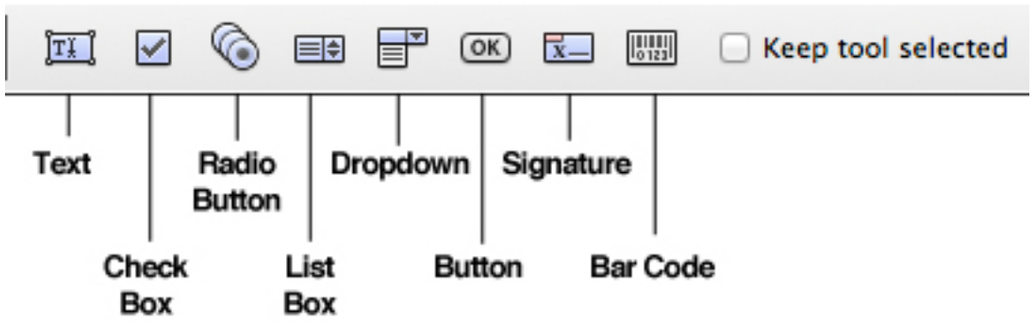


To return to Normal view, click the Close Form Editing button.

Using the Form Tools

At the top of the Form Edit mode window you find a row of tools used for creating field objects. There are eight different tools available for creating different field types. These tools include:

Figure 9.3 The Form Edit Toolbar



The Form Edit toolbar located at the top of the Form Edit window

- **Text Fields:** Text fields are used for entering both text and numeric information. On most forms, text fields occupy the majority of fields on a form.

- **Check boxes:** Check boxes can be used as independent items within a group or they can be mutually exclusive fields. For example if you wish to obtain information where a constituent should check all items they possess for identification such as driver's license, birth certificate, credit card, registration identify card, income tax report, etc. The form filler would check all items that apply. The check boxes would appear within a group for a single question where each box is available for marking without affecting the other boxes.

  Mutually exclusive check boxes enable form fillers to select only one box within a group. As one box is checked all other boxes appear unmarked. You may ask a question such as highest level of education. The responses might include grammar school, high school, baccalaureate degree, graduate degree, and doctorate. The form filler would check only one box within the group. If one item is checked and then another item is checked, the first mark is cleared leaving only one mark in the group.

  Mutually exclusive fields can help you acquire responses that help prevent user error. If you have a question where a Yes or No response is required, you don't want the user to check both boxes. When you set up mutually exclusive fields, the user is restricted to supplying only a single response.

- **Radio Buttons:** Contrary to popular belief, radio buttons are identical to checkboxes with two exceptions. Radio buttons can have a circle appearance as a field border and checkboxes are restricted to squares/rectangles for field borders. The second exception is that a check box can be unchecked and radio buttons can only be cleared in a mutually exclusive set by clicking another radio button.

Everything else concerning check boxes and radio buttons is equal.

To simplify learning creating form fields, you can avoid using radio buttons on your forms. You don't need them since check boxes provide you with everything you need for designing forms and collecting responses.

- **List Boxes:** List boxes provide you with a field type where responses are added in a scrollable window. If the space on your form is limited, you can create a list box where response information appears in a scrollable window.

- **Dropdown lists:** This field was formerly labeled Combo Box in earlier versions of Acrobat. As the name implies the field appears as a drop-down with response information opening in a list. If space is limited on a form and you want to collect responses for something like civil status, you might create a dropdown list with menu items such as single, married, divorced, separated, widowed, etc. The form filler opens the menu and clicks the item that applies.

- **Buttons:** In previous versions of Acrobat, the Button tool appeared both in Form Edit mode and by default in the Toolbar Well in Normal view. In Acrobat XI you can only access the Button tool in Form Edit mode or by editing a tool set and adding the button at the top of the Acrobat window in the Toolbar Well.

You can invoke actions by clicking a button and you can add content in the form of a button icon. An action might be resetting a form and clearing data. You can add a button field that invokes the Reset Form action.

Button icons are images and the button can be sized to a full page if needed. A button icon might be something like a passport photo added to a form. Rather than the user attaching a photo to the form, the button icon can be set up to import a photo making it visible on the form.

- **Digital Signatures:** For forms hosted on government websites you typically won't add digital signature fields. Creating a digital signature for end users is a complicated process and most users are not likely to create a digital signature in Reader. If a user did have a digital signature, you would need a certificate to verify the signature. This would be prohibitive if collecting forms routinely from constituents.

Where digital signatures might be used effectively is with in-house forms within a branch of government by employees. If this is of interest to you, look at Chapter 20 where I discuss creating forms for in-house use.

- **Barcodes:** Barcodes are added to forms for retrieving form data using barcode readers. Since our focus is creating forms with an emphasis on electronic routing, you won't likely need to work with barcode fields.

While Acrobat provides you with eight different field types, you need only be familiar with five of the eight field types (*Text, Check Box, List Box, Dropdown List,* and *Buttons*) for just about any form you host on government websites.

Understanding JavaScript

Acrobat uses a special implementation of JavaScript. If you know JavaScript it's easy to modify code for working in Acrobat and with PDF files. If you don't know any JavaScript you can begin by copying and pasting code in forms.

JavaScript is beneficial for all eForms and many forms require no more than a simple routine you can copy and paste. More sophisticated scripts take time to learn or require assistance from programmers. For now, it's important to know where JavaScripts are added to a form.

The areas in forms where you find JavaScripts include:

- **Text Fields:** Text fields have a calculation property and often you find JavaScripts used for field calculations such as summing data. Some math routines do not require JavaScript and can be handled with some easier tools available in Acrobat. However, more sophisticated math routines do require using JavaScript. Later in Chapters 16 and 17 I cover multiple methods for calculating data and offer code samples for easy copy/paste methods.

  In addition to the Calculate properties you also find JavaScripts in text fields for Actions, Formatting, and Validating field data.

- **Buttons:** Buttons invoke actions and there are a series of different action types you can apply to buttons such as hiding fields, resetting forms, executing menu items, etc. Among the various action types you can assign to buttons is running a JavaScript. You can use buttons and JavaScript for many dynamic features in your forms.

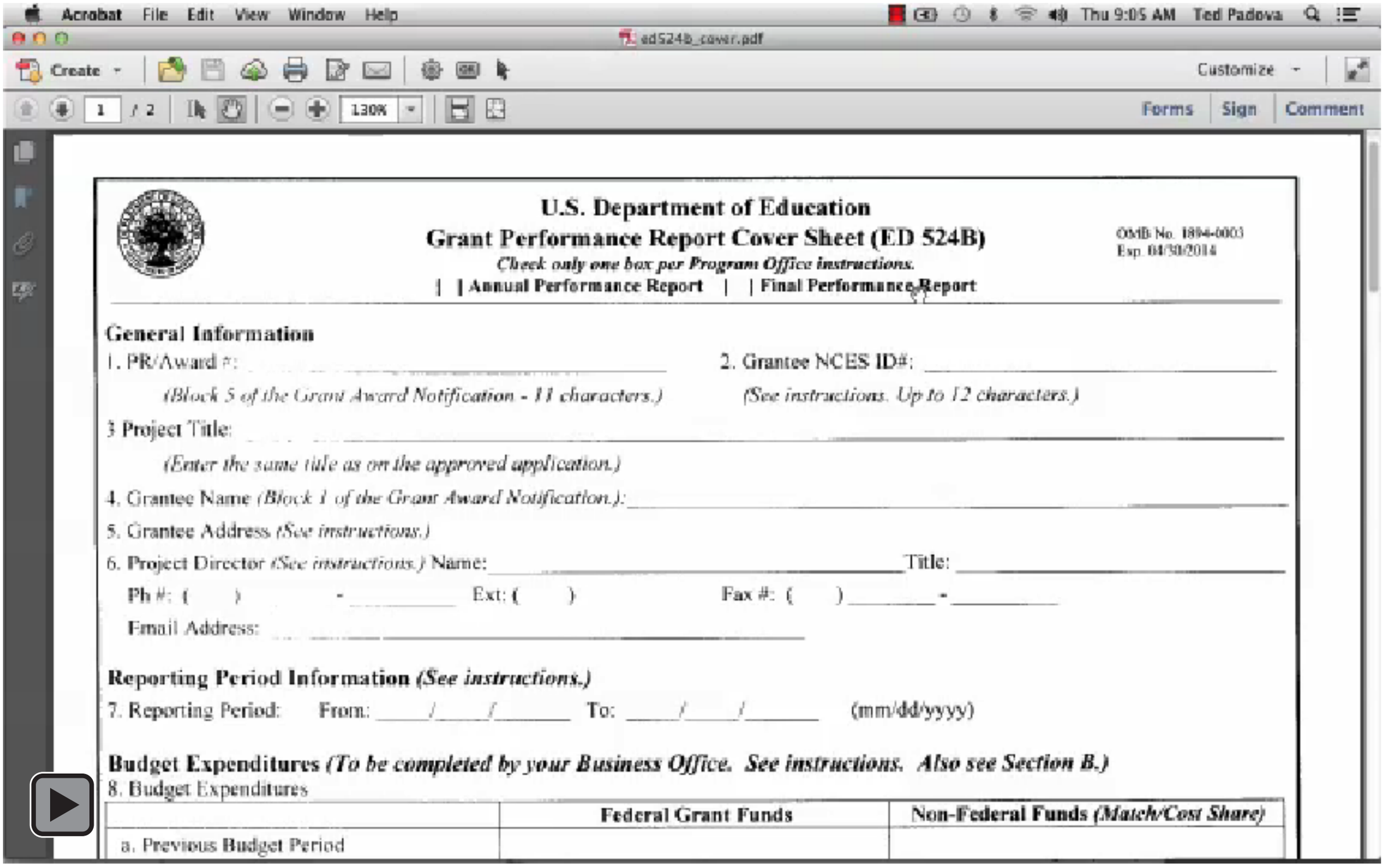
- **Document Level JavaScripts:** Functions can be accessed multiple times in a form. These functions are added as Document JavaScripts. You can also use document level JavaScripts to execute a routine just after the PDF opens. One helpful script you can add at the document level is assessing the viewer version the end user is working with. If the user is opening your form in Reader prior to version XI you can invoke a message that informs the user they need Reader XI in order to complete the form. For more on assessing viewer versions see Chapter 17.

- **Document Actions:** Document actions include printing and saving a document. When the user saves your form you might want to add a message in the form of a JavaScript that reminds the user to submit the form via a submit button on the form.

- **Bookmarks:** You can add JavaScripts to bookmarks. When the user clicks a bookmark in a long form, you might want to invoke an action such as opening a secondary file where instructions are found.

- **Page Actions:** Page actions are added in the Page Thumbnails Navigation pane. You can add JavaScripts for a Page Open action and a Page Close action. When the user scrolls pages in a form you might want to add a Page Close action that offers a message in a dialog box or some other action via a JavaScript.

Movie


Acrobat File Edit View Window Help
Thu 9:05 AM Ted Padova
ed524b_cover.pdf
Create
Customize
1 / 2
130%
Forms Sign Comment
U.S. Department of Education
Grant Performance Report Cover Sheet (ED 524B)
OMB No. 1894-0003
Exp. 04/30/2014
Check only one box per Program Office instructions.
| | Annual Performance Report | | Final Performance Report
General Information
1. PR/Award #:
(Block 5 of the Grant Award Notification - 11 characters.)
2. Grantee NCES ID#:
(See instructions. Up to 12 characters.)
3 Project Title:
(Enter the same title as on the approved application.)
4. Grantee Name (Block 1 of the Grant Award Notification.):
5. Grantee Address (See instructions.)
6. Project Director (See instructions.) Name: Title:
Ph #: ( ) - Ext: ( ) Fax #: ( ) -
Email Address:
Reporting Period Information (See instructions.)
7. Reporting Period: From: / / To: / / (mm/dd/yyyy)
Budget Expenditures (To be completed by your Business Office. See instructions. Also see Section B.)
8. Budget Expenditures
Federal Grant Funds
Non-Federal Funds (Match/Cost Share)
a. Previous Budget Period

Chapter 10

Working with Text Fields

Adding text fields and tables to forms

Working with Text Fields

Chapter 10

If you followed along from the beginning of this book you know that forms begin in Form Edit mode and you have five field types that are commonly used with eForms. When you open a form in Form Edit mode, it's time to add field objects.

IN THIS CHAPTER

Creating Text Field Objects

Text fields are the most common field objects you find on all forms. To add text fields on a form you must open the PDF form design in Acrobat and click the Edit Form tool in the Tools panel.

To create text fields do the following:

1. **Enter Form Edit mode**.
   With a form open in Acrobat click the Edit Form button. If a dialog box opens asking you if you want Acrobat to auto-detect fields click No.

2. **Select the Text tool.**
   From the tools at the top left of the Form Edit mode window, click the Add text field tool (I simply refer to this as the Text tool in this chapter).

3. **Create a text field.**
   With the Text tool selected click and drag open a rectangle where you want to add the first field.

4. **Adjust the field to size.**
   If the text field does not precisely fit in the area where the text field is to appear, click one of the handles (black squares at the corners and sides of the rectangle) and drag in or out to reshape the rectangle.

The first text object is added to your form. Before you add additional fields, you want to set all field attributes for the first field. Setting field attributes is handled in the Field Properties window.

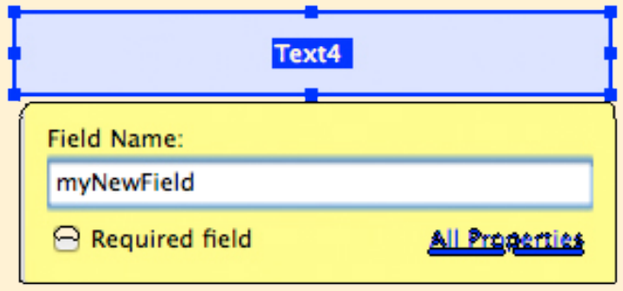
Opening the Field Properties Window

When you first create a field in Form Edit mode a pop-up appears (Figure 10.1). If you want to adjust properties, you can immediately click Properties in the pop-up before you make any other adjustment or before you deselect the field. If you want to adjust properties after the fact you have several ways to open the Field Properties window. These include:

Figure 10.1 Mini Popup Window

Note

When you click a handle and size the rectangle the small yellow pop-up disappears. This item is used to name a field and easily access the Text Field Properties window. Don't worry if the item disappears, you can easily open the Text Field Properties window and later edit field names.



Mini Popup window opens where you can name a field and open the field Properties window

- **Double click the field.**
  Double clicking the field while the Text tool is selected opens the Field Properties window.

- **Open a context menu and choose Properties.**
  Right click the field to open a context menu and choose Properties. On a Macintosh with a one-button mouse, press Control and click to open a context menu.

- **Exit Form Edit mode, and perform the same operations in Normal view.**
  Once you add a field in Form Edit mode, you can return to Normal view and make all the edits you need for adding fields and adjusting properties while remaining in the Normal view. However, you must first click the Select Object tool (or press the R key on your keyboard) and double click on a field or open a context menu and choose Properties.

At this point you have a decision to make. Do you want to stay in Form Edit mode or do you want to work in Normal view? I personally never work in Form Edit mode unless I work with the Fields panel. I find this entire Form Edit mode experience oversimplified by Adobe and unnecessary for experienced form authors.

An advantage you have in working in Normal view is you can easily toggle the editing and a view like the end user will see. Press the H key to select the Hand tool and all the handles and key lines on field objects disappear. When you want to use the Select Object tool, press the R key and you're back to editing the fields.

Another advantage in working in Normal view is you have access to all Acrobat tools. You may want to assess coordinates on a page for locating precise positioning for spawned fields, you may want to add a comment note, you may want to add a watermark and assess its placement in respect to fields, you may want to insert a blank page for testing scripts, or many other tasks that are not permitted in Form Edit mode.

Whether you wish to work in Form Edit mode or Normal view is a personal choice. Either is acceptable.

Setting Attributes in the Field Properties Window

When you create the first text field object you immediately need to address field properties. Some field properties include font choice, appearances such as border and fill colors, field formatting, visibility, calculations and a host of other choices you make in the Text Field Properties window.

The reason it's important to change the attributes from the defaults is that if you choose to duplicate a field, the new field copy carries with it the field attributes. One thing that helps speed up your work is creating a field, setting attributes, then duplicating the field and just changing the name and tooltip for

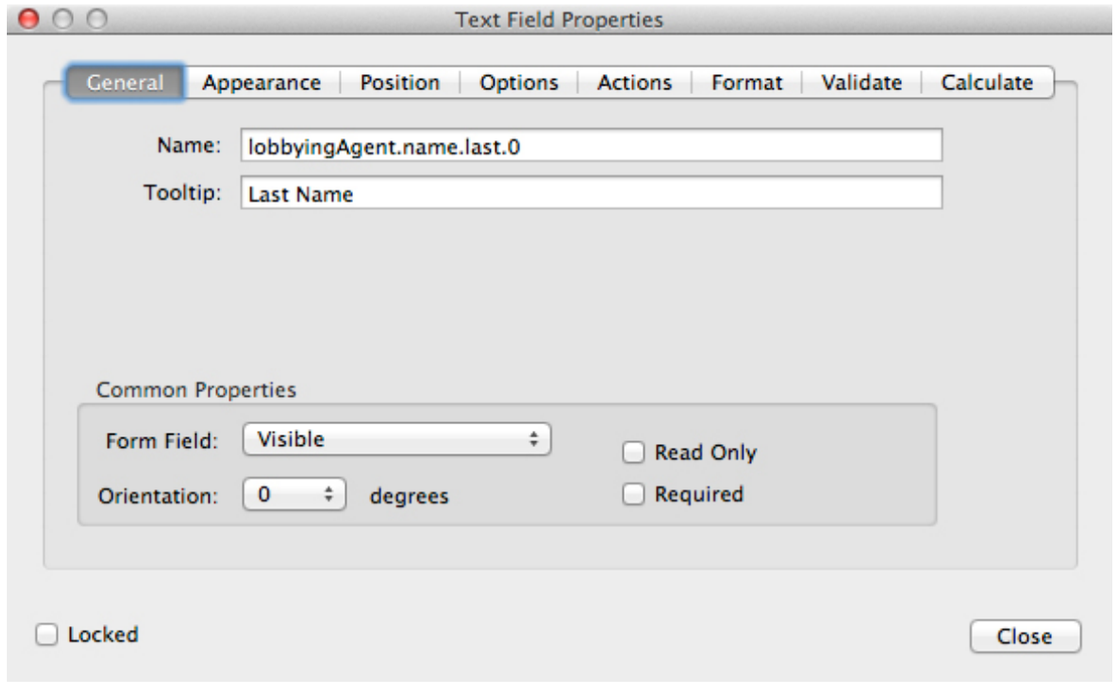
the new fields. All duplicate fields have the same font, font size, and appearance settings plus any other attributes you add to the original field.

To edit the field properties do the following:

1. **Select a field and open the Field Properties window.**
   Double click a field in Form Edit mode or use the Select Object tool in Normal view and double click the field. The Text Field Properties window opens.

2. **Select General Properties.**
   Notice the tabs at the top of the Text Field Properties window. Click General on the far left to edit the General Properties.

Figure 10.2 Text Field Properties Window



Click the General tab to open the General Properties pane.

3. **Change the field name and tooltip.**
   The first item to address always is changing the field name. If you don't change the name, all fields with the same name will have the same properties and they will contain the same user data. Each and every field in a PDF form requires a unique name. There is one exception with Button fields I explain later in Chapter 12.

4. **Use hierarchical names.**
   In my example form notice that there are two sections containing similar data. The first section *10.a Name and Address of Lobbying Agent* and the second *section 5*, If *Reporting Entity in No. 4 is…* have several similar fields. To make things easy as I move along on this form, I want to set it up so I can easily copy/paste fields. To do so, I'm going to set up field names with a root name common for each section. For example, in the 10.a area I use *lobbyingAgent* as the root name. In the section 5. area I'll use *prime* as the root name. If you're confused, don't worry. You'll see the results soon.

   For the first field name I use *lobbyingAgent.name.last*. Note that *lobbyingAgent* and *name* are separated with a period. This is important when using hierarchical names. Below the field *lobbyingAgent.name.last* you can type a field tool tip. Add a descriptor here such as *Lobbying Agent Last Name* and you finish editing the General properties.

Figure 10.3 Sample Form

DISCLOSURE OF LOBBYING ACTIVITIES

Complete this form to disclose lobbying activities pursuant to 31 U.S.C. 1352
(See reverse for public burden disclosure.)

Approved by OMB
0348-0046

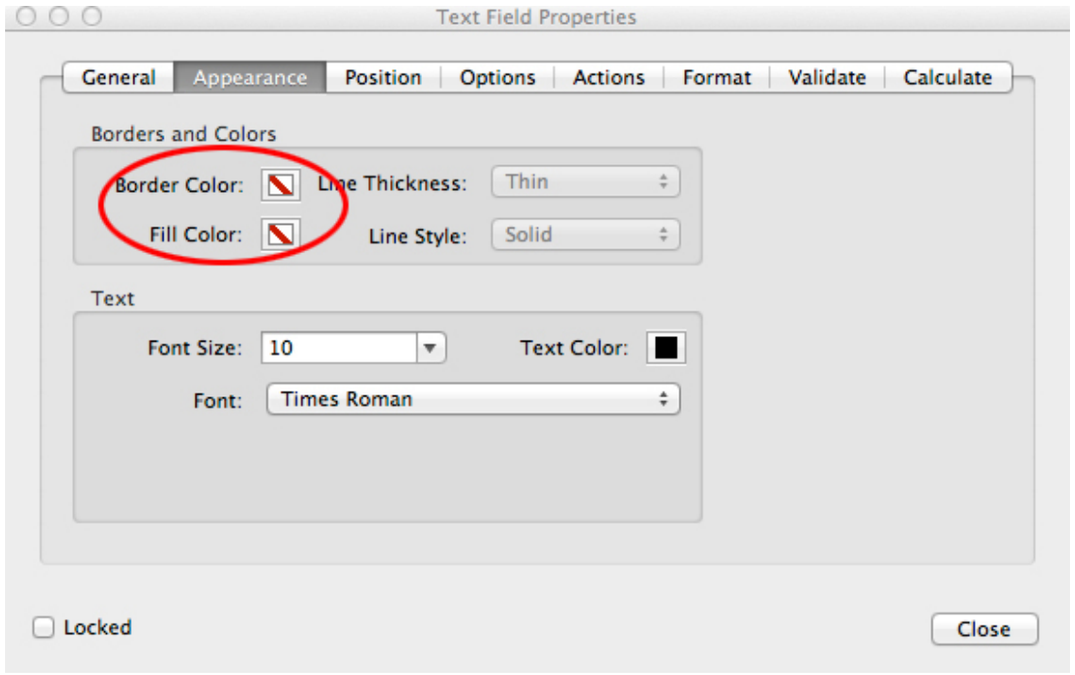
1. Type of Federal Action: ☐ a. contract b. grant c. cooperative agreement d. loan e. loan guarantee f. loan insurance	**2. Status of Federal Action:** ☐ a. bid/offer/application b. initial award c. post-award	**3. Report Type:** ☐ a. initial filing b. material change **For Material Change Only:** year _____ quarter _____ date of last report
4. Name and Address of Reporting Entity: ☐ Prime ☐ Subawardee Tier _____, *if known*: **Congressional District**, *if known*:	**5. If Reporting Entity in No. 4 is a Subawardee, Enter Name and Address of Prime:** **Congressional District**, *if known*:	
6. Federal Department/Agency:	**7. Federal Program Name/Description:** CFDA Number, *if applicable*: _____	
8. Federal Action Number, *if known*:	**9. Award Amount,** *if known*: $	
10. a. Name and Address of Lobbying Registrant (*if individual, last name, first name, MI*):	**b. Individuals Performing Services** (*including address if different from No. 10a*) (*last name, first name, MI*):	
11. [illegible] 1352. This disclosure of lobbying activities is a material representation of fact upon which reliance was placed by the tier above when this transaction was made or entered into. This disclosure is required pursuant to 31 U.S.C. 1352. This information will be available for public inspection. Any person who fails to file the required disclosure shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.	Signature: _____ Print Name: _____ Title: _____ Telephone No.: _____ Date: _____	
Federal Use Only:		Authorized for Local Reproduction Standard Form LLL (Rev. 7-97)

Sample form

5. Edit Appearances.

Click the Appearance tab in the Field Properties window to open the Appearance adjustments.

Figure 10.4 Appearance Properties



The Text Field Appearance tab

For Border Color and Fill Color click the diagonal line in the color swatches for None. You open the color swatches pop-up by clicking on the square adjacent to the Border Color and Fill Color text. If you need a border on a field you can adjust line weights in the Line Thickness drop-down menu. As a general rule, it's best to draw rules in your original authoring program so fields typically don't require adding borders and fills.

For Font Size and Font make choices from the drop down menus. As a general rule, 11-point type is a standard typewriter size that works well for most forms. You may want to set the font to 11-point type or lower to 10-point. Be certain the size is not set too small. This is an item you need to check when you exit the Field Properties window. Be certain to select a font size then type in a field to insure text characters fit within the height of the text box.

6. **Position.**
   Click the Position tab. This is a new item added in Acrobat XI. In this pane you find precise coordinates for the position of the field object on the page. In most cases you won't need to edit any of the values. Just pass this by and move on to the next tab.

7. **Options.**
   Click the Options tab and look over the field options. Here you find alignment for the data typed in the field, check spelling, permit long text to overflow, enable multi-line text, setting up the field as a comb field and more. For the current example form I leave the settings in the Options tab at the defaults and move on to the next tab.

8. **Review the Actions, Format, Validate, and Calculate tabs.**
   For the current field on the example form I leave the remaining items at the defaults. In the Actions tab you can assign a number of different actions including JavaScripts. In the Format tab you find options for choosing a format. You can, for example, format the text as a number. The default is None which is fine for text data. The Validate tab is used for validating values and the Calculate tab is used to perform calculations.

9. **Press the Enter/Return key on your keyboard to set the attributes.**
   Edits you make in the Text Field Properties window are dynamic and take effect as you make changes. As rule I always press Enter to be certain the last item I changed does indeed register in the window.

   At this point, leave the Text Field Properties window open. You can continue adding new fields and make changes without having to reopen the window each time you make an edit. However, if you test a field and type data in a field, the minute you select the Hand tool, the window closes and you need to reopen it to continue editing field properties when adding additional fields.

10. **Check the field with sample text.**
    Press H on the keyboard to select the Hand tool and type text in the field to be certain the text fits the field box height. If you're in Form Edit mode you can click Preview at the top of the window and type text or you can click Close Form Editing to exit Form Edit mode and type text in the field.

Your first text field object should be formatted properly and you're ready to add additional fields. If you want to use the attributes you assigned to the previous field and have them appear as new defaults you can open a context menu on the field and choose *Use Current Properties as New Defaults*.

Duplicating Fields

The easiest way to populate a form, especially when you have a number of fields all having the same field attributes, is to duplicate fields. To duplicate fields do the following:

1. **Enter Normal view** if you're in Form Editing mode and **press the R key on your keyboard to choose the Select Object tool.**

2. **Click the field you want to duplicate.**
   Press the CTRL key (Windows) or the Option key (Macintosh) and click the mouse button. With the mouse button and modifier key pressed, press the Shift key and drag the field vertically (or horizontally).

   The CTRL/Option key permits you to duplicate a field. Pressing the Shift key constrains the movement —in this example to a horizontal movement.

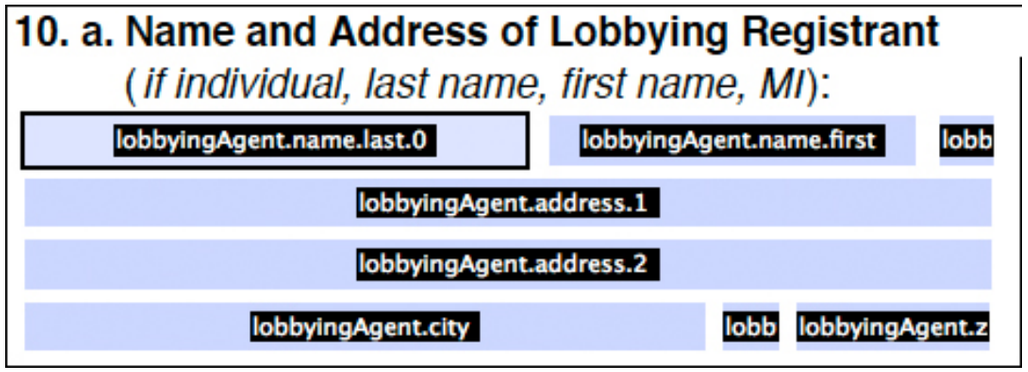
3. **Size the field and fit to position.**
   Drag handles on the field rectangle to size the field for the next position.

> **Note**
>
> *In the first release of Acrobat XI version XI.0 you may experience a problem with moving the field. If you press the keys and move a field and you don't see the duplicate field, deselect the field then click and press shift and drag to position. The field gets duplicated in the first step but may not move.*

4. **Open the Field Properties and change the field name.**
   You must name each field with a unique name. In this example I name the field *lobbyingAgent.name.first*. The same root name is used and the name changes after the root. Below the field, edit the Tooltip text box.

5. **Continue adding additional fields.**
   Follow the same steps to duplicate fields, change field names, and edit tooltips to populate all the fields in Section 10.a. As I said earlier, you can leave the Field Properties window open while adding new fields.

Figure 10.5 Fields Added to Form



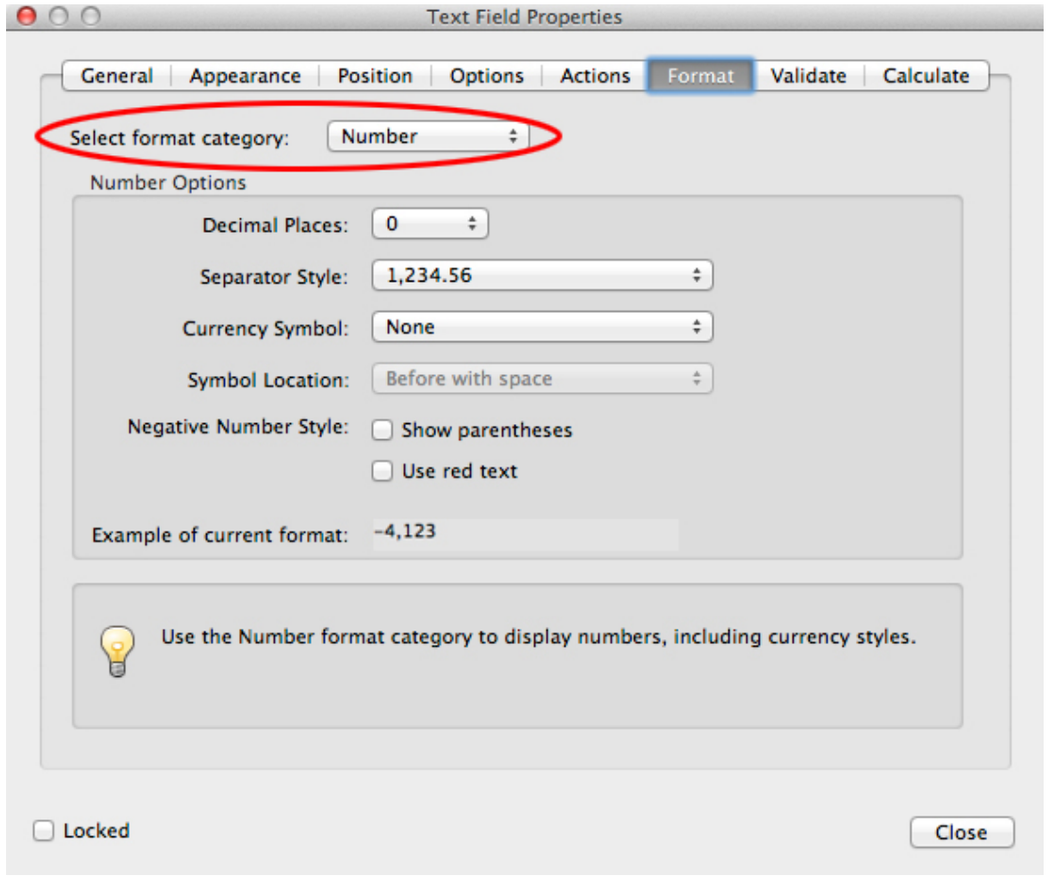
First group of fields added to form

Changing Field Formats

All fields in Section 10.a are text fields and work fine with no additional formatting. The text field in Section 9 for a dollar amount needs formatting. In addition, the telephone and date fields can use a little help with some special formatting.

When you have text fields that require numbers and especially if you want to use numeric data in calculations, you need to format the text fields as numbers. Open the Text Field Properties window and click the Format tab (Figure 10.6). The default you see in the Select format category drop-down menu is None. Open the menu and choose Number. Where you see the Number Options section, make choices for how you want the number formatted. In my example I choose 0 (zero) for Decimal Places and use a Separator Style with commas. You can choose a symbol for a currency number from the Currency Symbol drop-down menu, however the sample form displays a US dollar sign, therefore in this example the Currency Symbol is left at None.

Figure 10.6 Text Field Format Properties

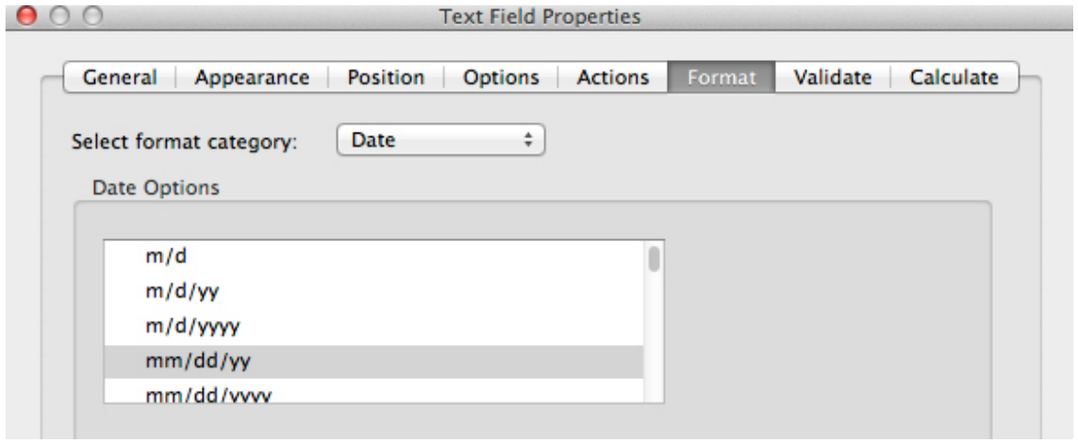


Format numbers in the Format tab

Telephone numbers and Date formats are easily formatted in the Field Properties window in the Format tab. Double click the Telephone number field and click the Format tab in the Field Properties. Open the Select format category drop-down menu and choose Phone Number.

Double click the Date field, click Format and open the Select format category drop-down menu. Chose Date from the menu choices. From the Date Options choose the date format you want to use.

Figure 10.7 Date Field Properties

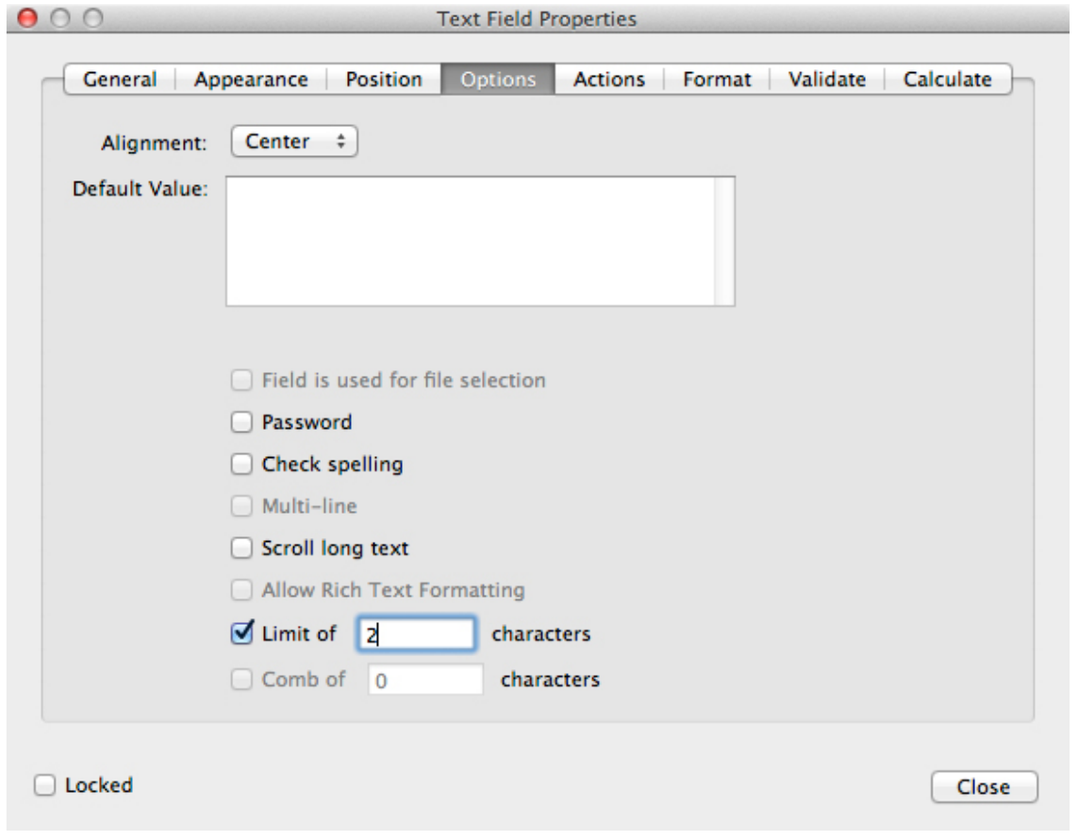


Choose Date from the Select format category drop-down menu to format date fields

When formatting the date field, use the same date format on all your forms. Reader requires you type date field data that specifically matches the date format. If you use *dd/mm/yy* for example, an application alert dialog box opens if the user types *dd/mm/yyyy*. It's also a good idea to add the accepted format in the Tooltip field for text fields —something like *Today's Date (dd/mm/yy)*.

Another item you may want to format is the State field. In the USA, states are abbreviated with two characters. You can limit user responses to two characters in the Text Field Properties Options settings. When you create a state field or other field type where you want to limit the number of characters, click Options in the Text Field Properties window. Type 2 in the *Limit of ____ characters* text box.

Figure 10.8 Options Tab



The Options tab

Something else that you may want to change in the State field is alignment. You might want the text in this field center aligned. To change alignment open the Alignment drop-down menu in the Text Field Properties Options tab and choose Center.

Examining the Sample Form

The sample I use here (Figure 10.10) is a form hosted by the USA Federal Government. The form was populated with form fields. Upon examining the form, we can make some assumptions about how the form was populated with field objects and how we might improve the construction of the form.

Understanding How the Fields Were Created

The sample form was not populated with Acrobat's auto field detection when the form was opened in Form Edit mode. We know this because the field names appear to be custom names and Acrobat could not create five separate fields in Sections 10.a and 5. If you click the Edit Form tool, Acrobat prompts you to auto detect fields. If the form author clicked Yes, Acrobat would have created a single field for Section 10.a and Section 5.

We also know that the form author used manual methods for naming fields. If the form author automatically populated the form, Acrobat would have named fields Text1, Text 2, Text3, etc.

This means that the form author created the fields in Section 10.a and Section 5 separately. The form is okay for how it is intended for use, but we can simplify the field creation process a bit — especially for the fields in Section 10.a and Section 5.

To add fields for Section 10.a we might want the fields constructed to follow the description where you see (*if individual, last name, first name, MI*). As it stands, the form is not very clear about what data needs to appear in the first line of fields. If the user adds a company name, then perhaps the last, first, and MI should not be completed. If this is the case a check box might be added for Individual and Company. If the user checks the Individual check box, three fields are shown. If the user clicks the Company check box only a single field is shown. Unfortunately there's not much room on the form to specify differences for company vs. last, first, and MI.

The way the form was originally designed, five fields were added in Section 10.a. I'm assuming the first field is for Company, the second field for last, first, MI, followed by two address fields and the city, state, and zip code in one field at the bottom of the section.

Rather than sort this out, let's use last, first, and MI for three fields at the top and forget about company name. If it is required as shown in Figure 10.5, the user might use the last name field to write the company name.

To duplicate fields in Section 10.a do the following:

Copying and Pasting Fields

Look over the sample form. At this point the 10.a Section contains fields properly formatted. Formatting for the State field was set to 2 characters in the Options tab, the Zip Code field was

defined as a Zip Code in the Format properties, and the fields all had tooltips that were added in the General Properties.

We could continue duplicating additional fields, change field names and tool tips, and edit field attributes as we did in Section 10.a to populate fields in Section 5. However there's a much easier way to create the Section 5 fields. We can copy and paste fields and easily change field names. This is where you find some of the power for using hierarchical names on forms.

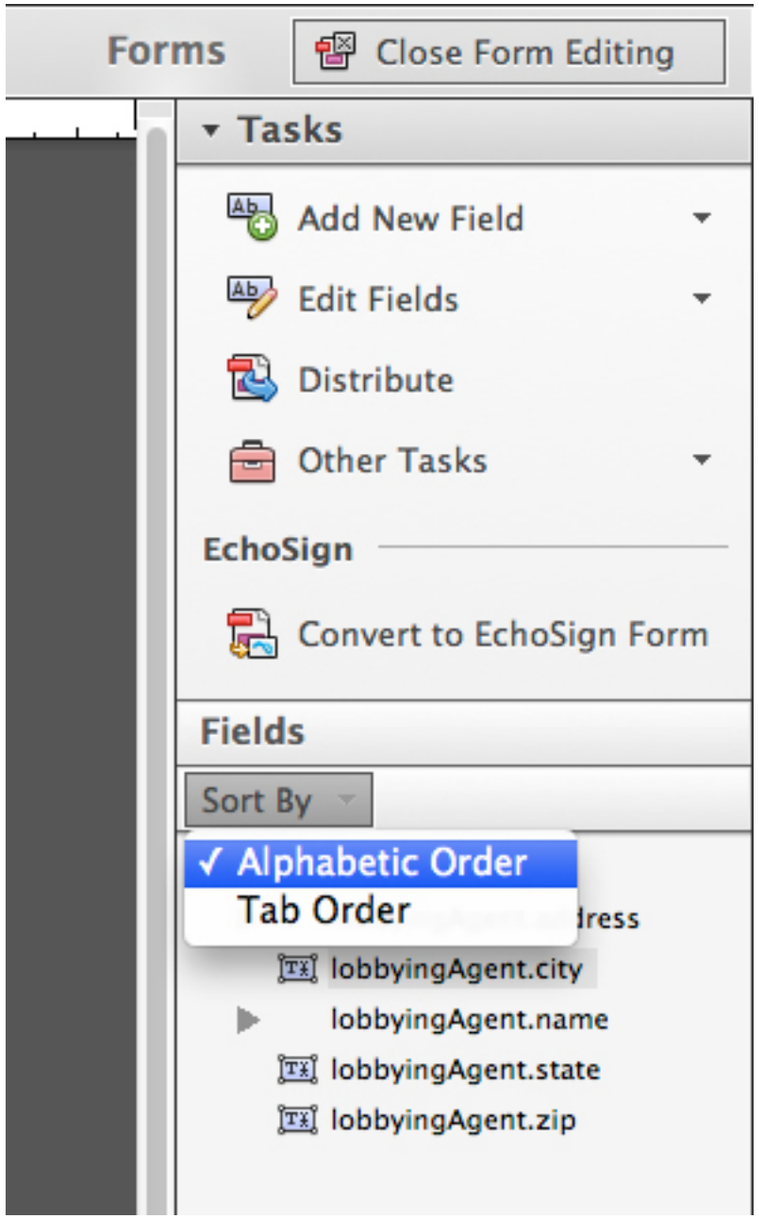
1. **Open Form Edit mode.**
   We first need to edit the root field names. To globally change root field names you must enter Form Edit mode.

2. **Sort fields alphabetically.**
   In the Fields panel, open the Sort By drop-down menu and choose Alphabetical Order.

3. **Change the root name for the lobbyingAgent fields.**
   Notice fields having the same root name are grouped together and displayed in a hierarchy. *lobbyingAgent* appears followed by the individual fields having the same root value. Click *lobbyingAgent* in the Fields panel and all fields beginning with the same root name (*lobbyingAgent*) are selected. Click again on *lobbyingAgent* and you can edit the root name.

4. **Change the root name to *prime*.**

Figure 10.9 Sorting Fields



In Form Edit mode sort fields alphabetically in the Fields panel.

5. **Copy fields.**
   On the form (while still in Form Edit mode) drag the cursor through the Section 10.a fields. Choose Edit ➤ Copy or press CTRL/ Command + C.

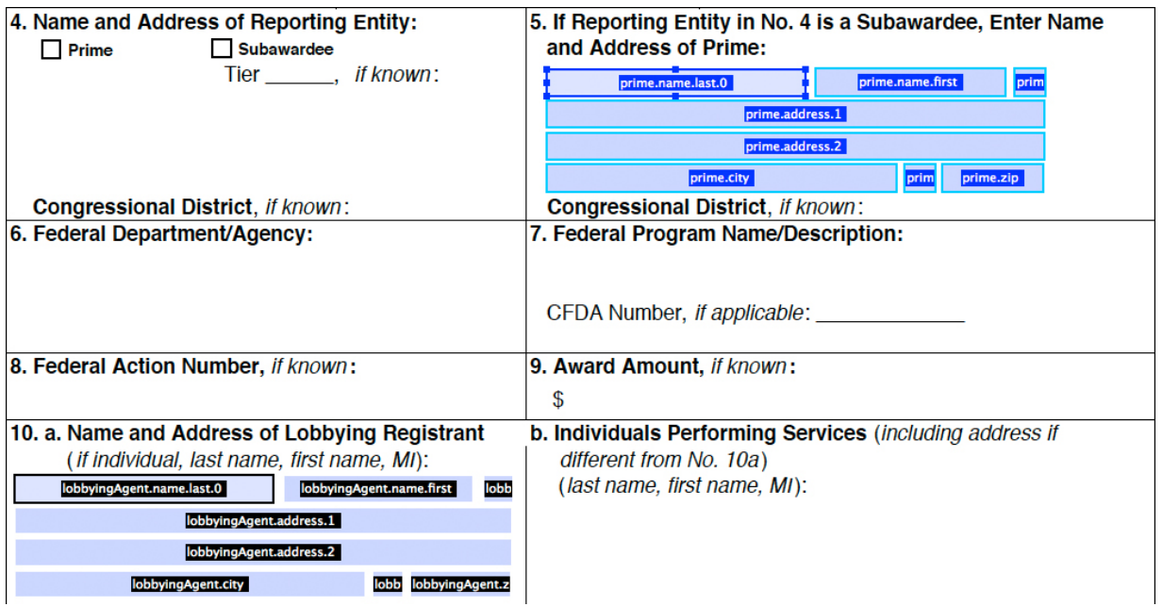
6. **Immediately undo the edit.**
   Choose Edit ➤ Undo or press CTRL/ Command + Z. The last edit Acrobat recorded was changing the field name. Acrobat didn't record you copying fields. At this point you have fields on the clipboard having the *prime* root name. When you

Undo, the fields in Section 10.a return to the original field names that have the root value *lobbyingAgen*t.

7. **Paste the fields. Choose Edit ➤ Paste or Press CTRL/ Command + V.**
   When you paste the fields on the clipboard, you paste the fields having the root value *prime*. Section 10.a has fields with a root value of *lobbyingAgent*. Your form now has fields in two sections with unique field names.

8. **Move the pasted fields to position.**
   Be certain to avoid deselecting fields. Move the cursor to a field and drag the selected fields to Section 5. Deselect the fields by clicking away from the selection and individually drag fields to position and size them to fit within the form design.

Figure 10.10 Move Pasted Fields



Pasted fields and positioned in Section 5

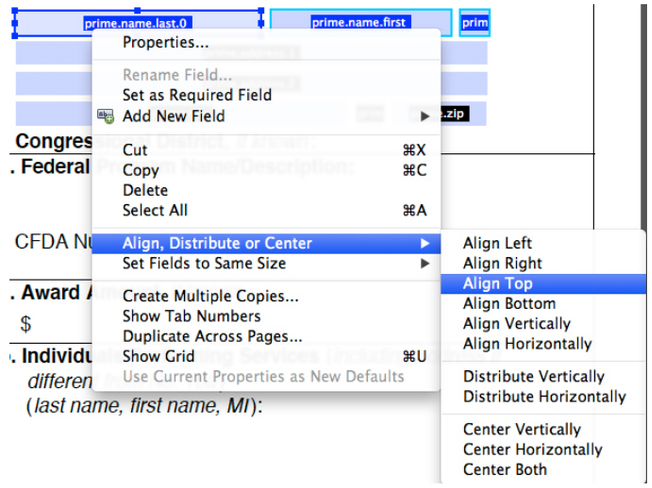
Aligning Fields

When you drag fields individually around a form, the field alignment is not precise and requires polishing up the alignment. Acrobat has a number of alignment commands to help you move fields to various alignment positions.

To align fields do the following:

1. **Select a row of fields by dragging the cursor through all fields in a row.**
   If you want to align columns you can drag through fields in a column.

Figure 10.11 Alignment Menu



Open a context menu on selected fields and make menu choices for alignment

2. **With fields selected open a context menu.**
   Right click on a selected field (Control + click with a one-button mouse on Macintosh) to open a context menu.

3. **Align the top of the selected fields.**
   From the context menu choose Align, Distribute, or Center ➤ Align Top. The selected fields align across the row. Continue aligning additional rows as needed.

Note

In the context menu you have various choices for aligning and distributing fields. These menu commands are helpful when you need to add some precision for the field placements.

The remaining text fields on the sample form require you adding fields individually. Be certain to name fields appropriately, add tool tips, and position the fields using alignment commands.

Working with Comb Fields

At some point in time you have probably filled out a form that contains comb fields. A comb field contains hash marks or boxes where separate characters are typed on a form. One of the mistakes I see on many forms containing comb fields is that PDF authors often create separate individual field boxes for comb fields.

Designing comb fields with separate fields presents a few problems. First, the user needs to tab through the fields each time a character is typed. For many credit cards you find 19 numbers. This requires the user to type and press tab 19 times. A second problem is that if data are exported, the export routine requires a subroutine to concatenate the numbers so the end result is a single number.

These are not monumental problems but they do require a bit of extra work for the user and the agency staff retrieving forms. All of this can be avoided by using the Comb field feature in Acrobat.

Designing Comb Fields

When working with comb fields you first need to make a choice for where you want to design the field appearance. In an authoring application you can create boxes or hash marks. In Acrobat you can apply a border to the field and Acrobat displays a series of boxes.

One problem I see consistently on forms having comb fields is the divisions between lines are most often off-center. If you rely on copying and pasting lines and moving to new locations in an authoring program the result is more often than not uneven spaces between lines. A better solution in a design application is to use a program with a step and repeat, blend, or duplicate feature that spaces lines perfectly. Programs like Adobe InDesign and Adobe Illustrator provide these options.

If you decide to let Acrobat define the field appearance, the Thin border width is not a hairline. Therefore if your form contains hairlines, the comb fields will appear larger than boxes and rules you might add in the authoring application for other fields and sections divisions.

The variances between border widths is slight when combining designs in both an authoring application and Acrobat. Most users

won't notice much difference. Graphic artists and forms designers will immediately notice the inconsistency in the designs.

Personally I find designing forms in Adobe InDesign the best solution. InDesign provides all you need for designing attractive forms. If you have to make a choice though between precise line widths and equidistant spacing, you need to test field attributes to find the best compromise.

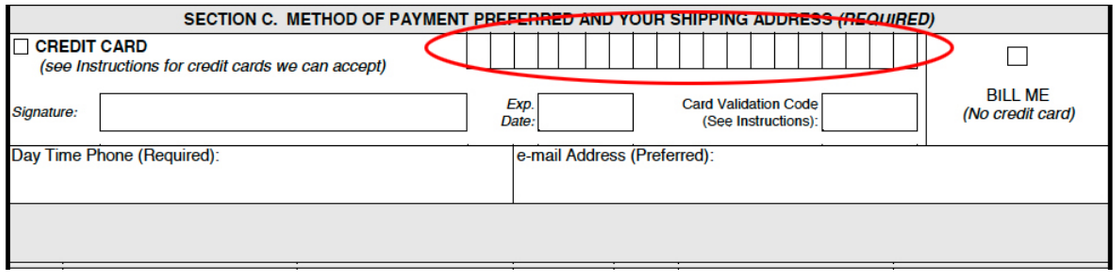
If you don't use an application that can easily proportionally space lines, test appearances for using the design background and using Acrobat's field border appearances. In some cases the appearance may be best with the original design if the non-proportional spacing is slight. In other cases you may find using Acrobat's borders provides a better solution.

Creating Comb Fields

In the sample form from the US National Archives Office you find a comb field in Section C. At first glance you can see some slight proportional spacing problems in the field design. This form originated in Microsoft Word where the form design was constructed.

In this example I create a comb field and examine differences between using no field border and using a border.

Figure 10.12 Comb Field in Design



A comb field on the form

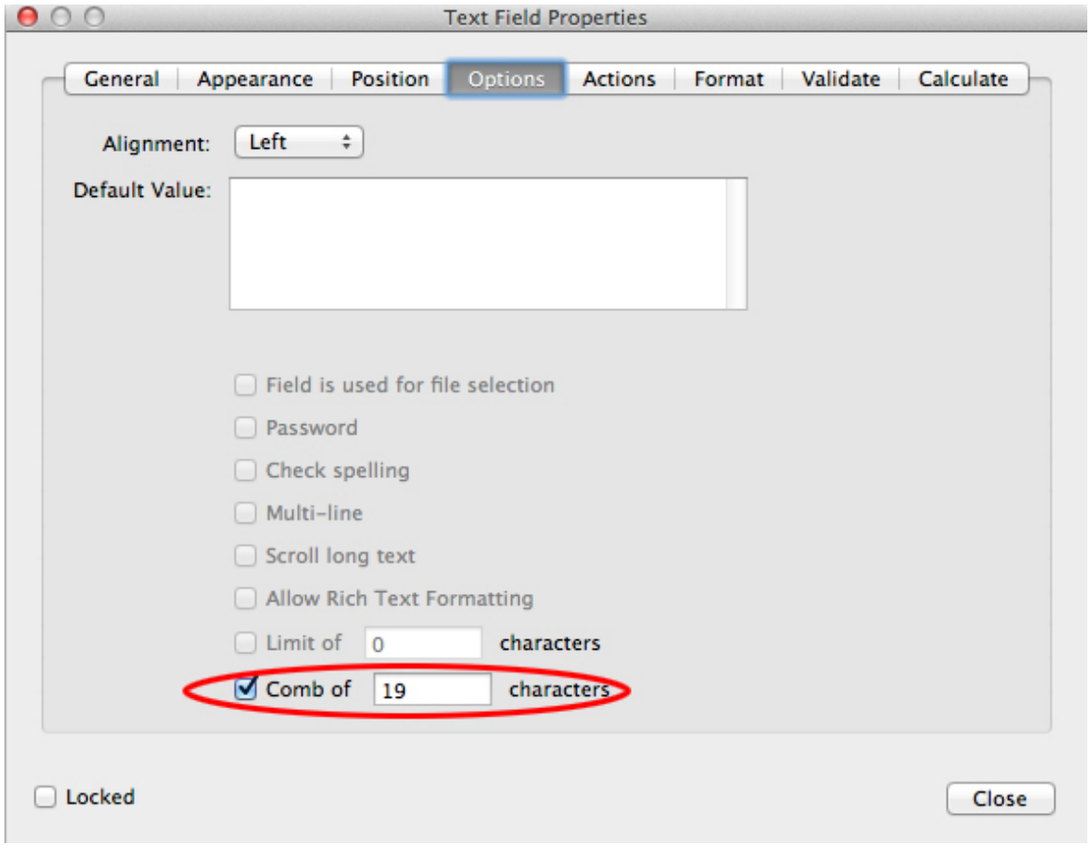
To create a comb field, do the following:

1. **Use the Text field tool and create a field.**
   Size the field to fit the comb field design.

2. **Edit the field properties.**
   Open the Text Field Properties window and begin in the General tab where you name the field and add a tool tip. In the Appearance tab, set the border and fill to none.

3. **Click the Options tab.**
   The Options properties is where you set the comb field attributes. By default the *Comb of* check box is grayed out. Comb fields cannot be spell checked, scroll text, be multi-lined, etc. You must remove all check boxes in the Options tab in order to create a comb field. Uncheck the boxes then check the *Comb of* check box. Type in the text field the number of characters for the field.

Figure 10.13 Options Properties

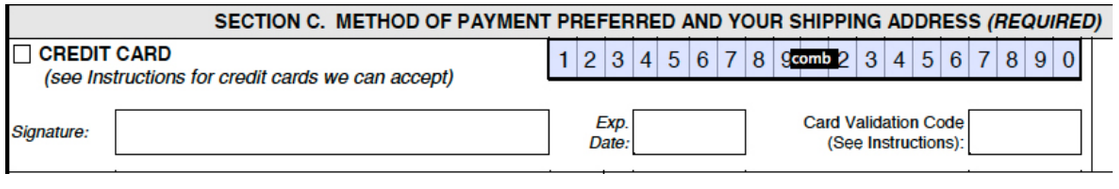


Click Options and uncheck all boxes. Click the Comb of check box and enter the number of characters to fit within the hash marks

4. **Test the field.**

   Type values in the field and look over the result. Be certain the number of characters needed match the number you defined in the Options properties.

Figure 10.14 Comb Field



Comb field populated

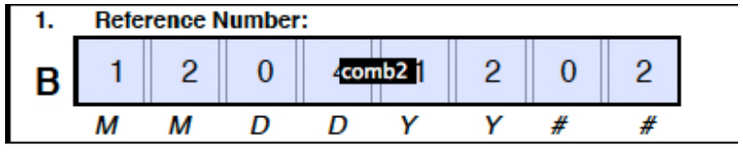
Although the separator lines are not proportionally spaced the end result doesn't look too bad on the sample form. To compare the field having no border with a field having a border, open the Text Field Properties window. Click the Appearance tab and click the color swatch to select a color. Be certain Thin is selected in the Line Thickness drop-down menu.

The edits are dynamic while the Text Field Properties window is open. You can change back and forth between a border and no border to compare the differences in visual appearance.

The sample design uses hairlines for rules and boxes. When you use a thin border for the field, the difference between the comb field appearance and other elements on the page is much more obvious than the slight difference in the uneven proportional spacing created in the authoring program. For this form it's best to use a comb field with no border.

If you look at Section 1 you find another area where a comb field might be used effectively. If the form author creates eight separate boxes the user needs to tab trough the fields. Adding a comb field, even though the design is not clearly a comb field, works well and provides a better solution.

Figure 10.15 Comb Field



Comb field at top of the form

Working with Tables

You find tables containing rows and columns on many eForms. Populating tables is made easy by using some additional features in Acrobat. Let's begin by creating a simple table. In this example I use a US Department of Agriculture form for recording employee overtime. Within the form, a table appears with ten columns and eight rows.

Figure 10.16 Table

USDA-APHIS

REQUEST AND AUTHORIZATION FOR OCCASIONAL OR IRREGULAR UNSCHEDULED OVERTIME

This request shall be used to obtain approval for "occasional or irregular unscheduled overtime". Whenever possible, overtime shall be approved in advance. Use a separate request for each overtime job. *This request is not for Regularly scheduled overtime or reimbursable overtime.* Prepare in a sufficient number of copies to allow one approved copy to be filed with each employee's Time and Attendance Report, and one copy to be sent to the next headquarters of the requesting unit.

1. **REQUESTING UNIT** (Branch, division, area office, district office, etc.)

2. LOCATION OF REQUESTING UNIT *(City and State)*

3. NAME OF EMPLOYEE	4. X if Applicable: Comp Time Auth	Nonexempt Under FSLA	Title	Grade	No. of Hours	Clock Hours	Date(s)	Pay Period	Location (City and State)
(1)									
(2)									
(3)									
(4)									
(5)									
(6)									
(7)									
(8)									

12. JUSTIFICATION: *Explain why overtime is necessary; alternatives available; consequences of non approval; other explanation).*

REQUESTED BY

The above overtime is ordered and/or approved for reasons shown in item 12. Justifications

13. SIGNATURE AND TITLE OF REQUESTING OFFICER	14. DATE REQUESTED	15. SIGNATURE AND TITLE OF APPROVING OFFICIAL	16. DATE APPROVED

APHIS FORM 1 (MAR 83)

Form containing a table

Rather than creating fields and duplicating them individually we can let Acrobat do most of the work for us. For a simple table like the one on the sample form, do the following:

1. **Create a field in the top row first column.**
   When working with tables you need to manually create the first row or column. When you add field names it's important to NOT use hierarchical names.

2. **Name the fields and set attributes in the Text Field Properties window.**
   In this example I name the first field *employee*. Add a tooltip, set the font and font size, choose no border and no fill, and leave the remaining settings at defaults.

3. **Create additional fields to populate the first row.**
   Duplicate the first field (CTRL/Option + Shift and drag right) and change the field name and tooltip. Change the name to the same name appearing in the columns at the top of the table. Continue adding fields across the first row.

Figure 10.17 First Row in Table

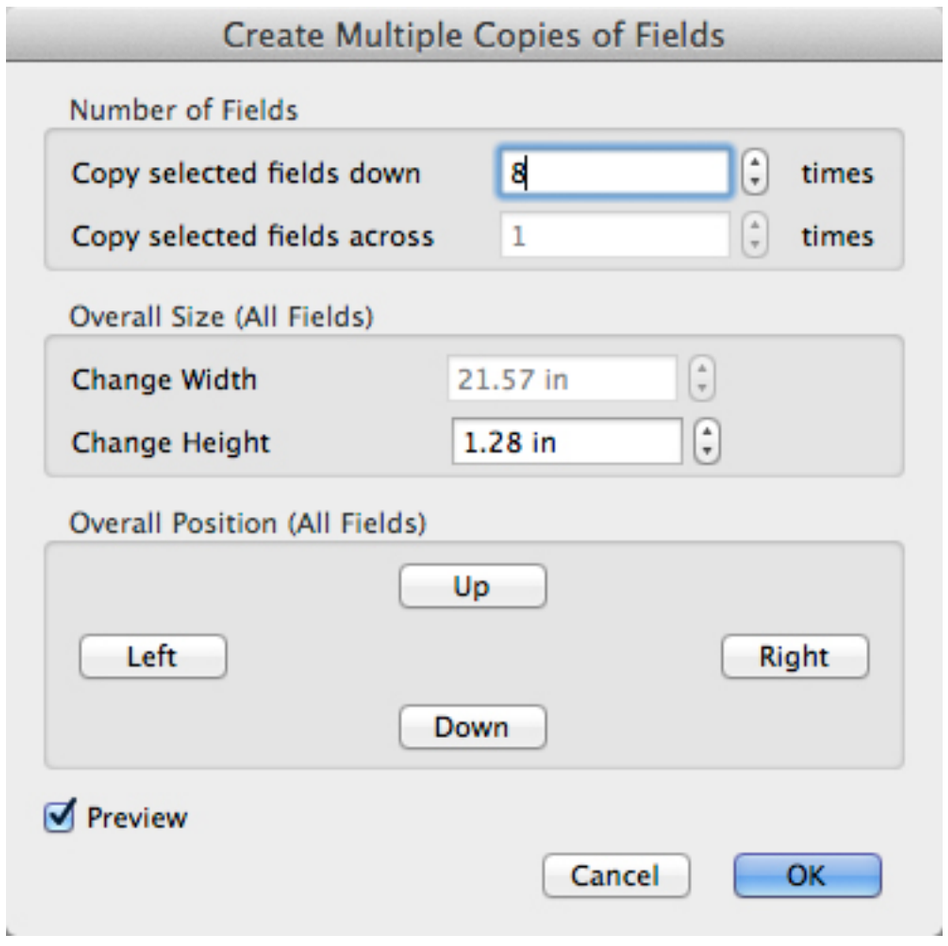
3. NAME OF EMPLOYEE	4. X if Applicable: Comp Time Auth	Nonexempt Under FSLA	Title	Grade	No. of Hours	Clock Hours	Date(s)	Pay Period	Location (City and State)	
(1) employee	authorized	nonexempt	title	grade	numHours	clockHours	dates	payPeriod	city	state

First row in table populated with fields across columns.

4. **Populate the table.**
   Select the first row of fields. Open a context menu and choose Create Multiple Copies. In the Create Multiple Copies window (Figure 10.18) enter 8 for the *Copy selected fields down* text box and 1 for the *Copy selected fields across* text box. Press OK and the table is populated with fields.

> **Note**
>
> *Acrobat takes care of naming fields in a hierarchical order. In this example the first column of fields are named employee.0, employee.1, employee.2, etc.*

Figure 10.18 Create Multiple Copies



5. **Drag through the fields in the last row and move the fields to fit in the last row of cells.**
   For small incremental movements you can use the arrow keys on your keyboard.

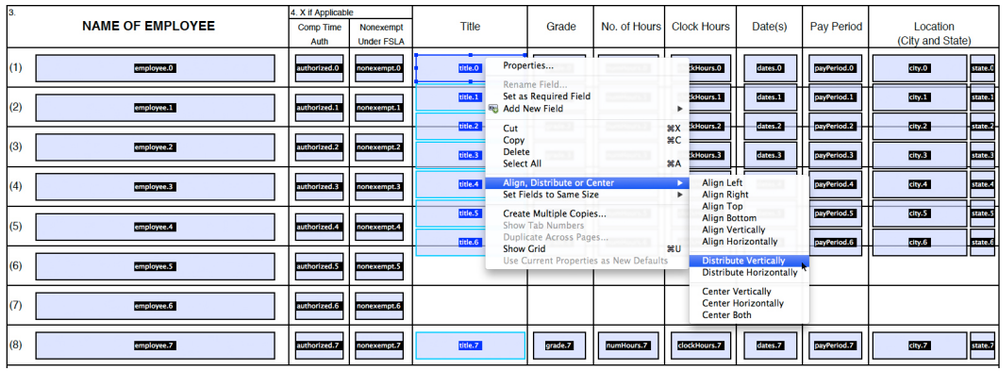
6. **Distribute fields.**
   Unfortunately Acrobat can't distribute rows evenly for the entire table. You need to distribute the columns individually.

   Drag through fields in the first column to select the fields. Open a context menu and choose Align, Distribute, or Center ➤ Distribute Vertically (Figure 10.19). The fields in the first column move and fit nicely within the Name of Employee column.

7. **Repeat the steps for distributing fields for each column.**

If you work with tables containing check boxes or radio buttons you have some additional considerations when it comes to populating tables. Look to Chapter 11 for more on creating tables with check boxes.

Figure 10.19 Distribute Fields



Select fields in a column and open a context menu.

Setting the Tab Order

When a user opens your form, they should be able to press the tab key and the cursor should jump to the first field in the top left corner of the form unless there's some good reason why another field should be the first field in the tab order. As the user presses the tab key the cursor should move to the next logical field.

As one of the final steps in completing a form, you should open the form in Acrobat, press the tab key and tab through the form. If the tab order is not logical, you need to make some changes in Form Edit mode.

There are several ways you can edit the tab order. To begin with you should look at the Fields panel in Form Edit mode and open the Tab Order drop-down menu. In this menu you have several choices. You can set the tab order to follow the structure of the document, set the tab order by rows and columns, set the tab order manually, or not specify a tab order.

Generally, it's best to start with setting the tab order by row. If you have a form where part of the form appears logical with a row order and other parts of the form appear logical with a column order you need to make adjustments manually after you set the tab order by row.

For manually adjusting tab order, do the following:

1. **Open a populated PDF form in Acrobat.**
   After opening a form, press H to select the Hand tool and tab through the document so you understand how the tabs are set and what edits need to be made.

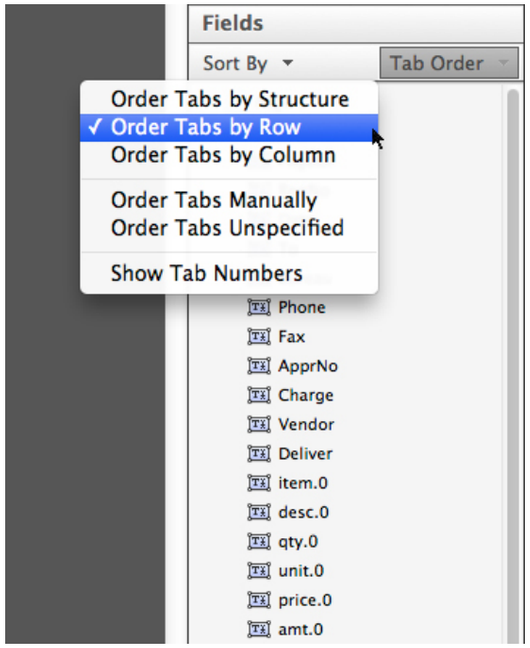
2. **Enter Form Edit mode and open the Tab Order drop-down menu.**

3. **Choose Order tabs by Row in the menu.**

4. **Test the tab order again.**
   Return to Normal view, press H to select the Hand tool and tab through the document. Be certain the cursor does not appear in any field and verify that the first tab moves the cursor to the first logical field. Tab through the document and note any problems with the tab order.

5. **Return to Form Edit mode.**

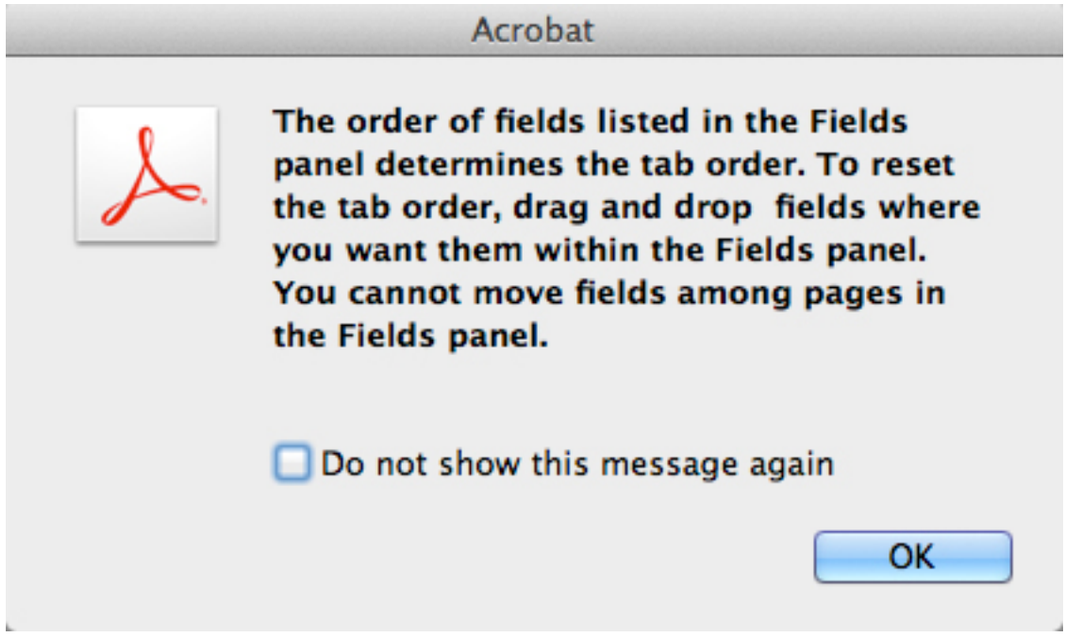
Figure 10.12 Row Tab Order



Open Form Edit mode and open the Tab Order drop-down menu in the Fields panel. Choose Order Tabs by Row.

6. **Adjust the tab order manually.**
   Open the Tab Order drop down menu and choose Order tabs Manually. A dialog box opens providing you instructions for manually ordering tabs. As denoted in the dialog box, you reorder tabs by clicking fields in the Fields panel and moving the fields up/down to change the tab order.

Figure 10.21 Alert Message



Read the message and click OK

In earlier versions of Acrobat prior to Acrobat XI you could only move one field at a time. Reordering tabs was a painfully slow process. Now in Acrobat XI you can select multiple fields and drag them in the Fields panel.

To select a group of fields in a contiguous order, click the top field you want to move, press the Shift key and select the last field you want to move. All fields between the first field and the last field are selected. Click and drag the fields up or down to change the tab order.

You can also select fields in a noncontiguous order. Click a field, press CTRL/Command and click another field. Continue using the modifier key and click each field you want to move. Click on a selected field and move up or down to change the order.

Moving the Cursor to Another Field

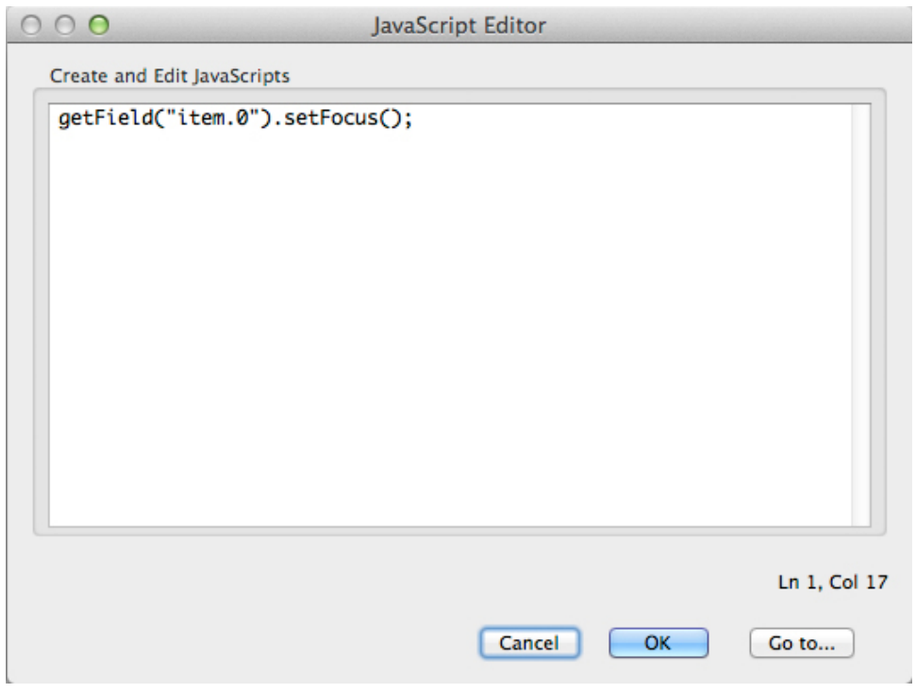
You may want the cursor to jump to another field after the user completes filling in a field with data. For example, suppose you have a section on a form where fields are visible by default. Suppose further that checkboxes appear at the top of a section. If the user clicks the Yes box the cursor needs to move to the next section. If the user clicks the No box, the following section fields are hidden and the user needs to skip the section. If this is the case you may want the cursor to automatically appear in the next section where fields are visible —even on another page.

In order to move the cursor to a field from a check box action, you need to write a JavaScript. (Look to chapters 16 to 19 for more on writing JavaScripts).

On a check box field, open the Actions tab in the Check Box Properties window and choose Run a JavaScript from the Select Action drop-down menu. Click Add and write the following code in the JavaScript Editor window:

```
1. getField("field name").setFocus();
```

Figure 10.22 JavaScript Editor



Open the Actions tab and choose Run a JavaScript from the Select Acton drop-down menu to open the JavaScript Editor.

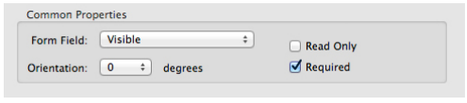
After writing the script click OK in the JavaScript window and close the Check Box Properties window. Ignore the number 1 in the above script when you type the code. Line numbers are used for annotating JavaScripts in this book to help explain different lines of code.

In the script above you need to add the name of the field you want the cursor to jump to within the quote marks. When the user clicks a check box, the cursor automatically moves to the field you identify in the script. If you have a Yes check box and a No check box you can add separate scripts so the cursor moves respective to the fields you identify in the scripts.

Setting Up Required Fields

In the General properties you find a check box for Required. The Required item is used to make certain all required fields contain data before a form is submitted. As you create new fields you can check the box in the General tab in the Properties window.

Figure 10.23 Required

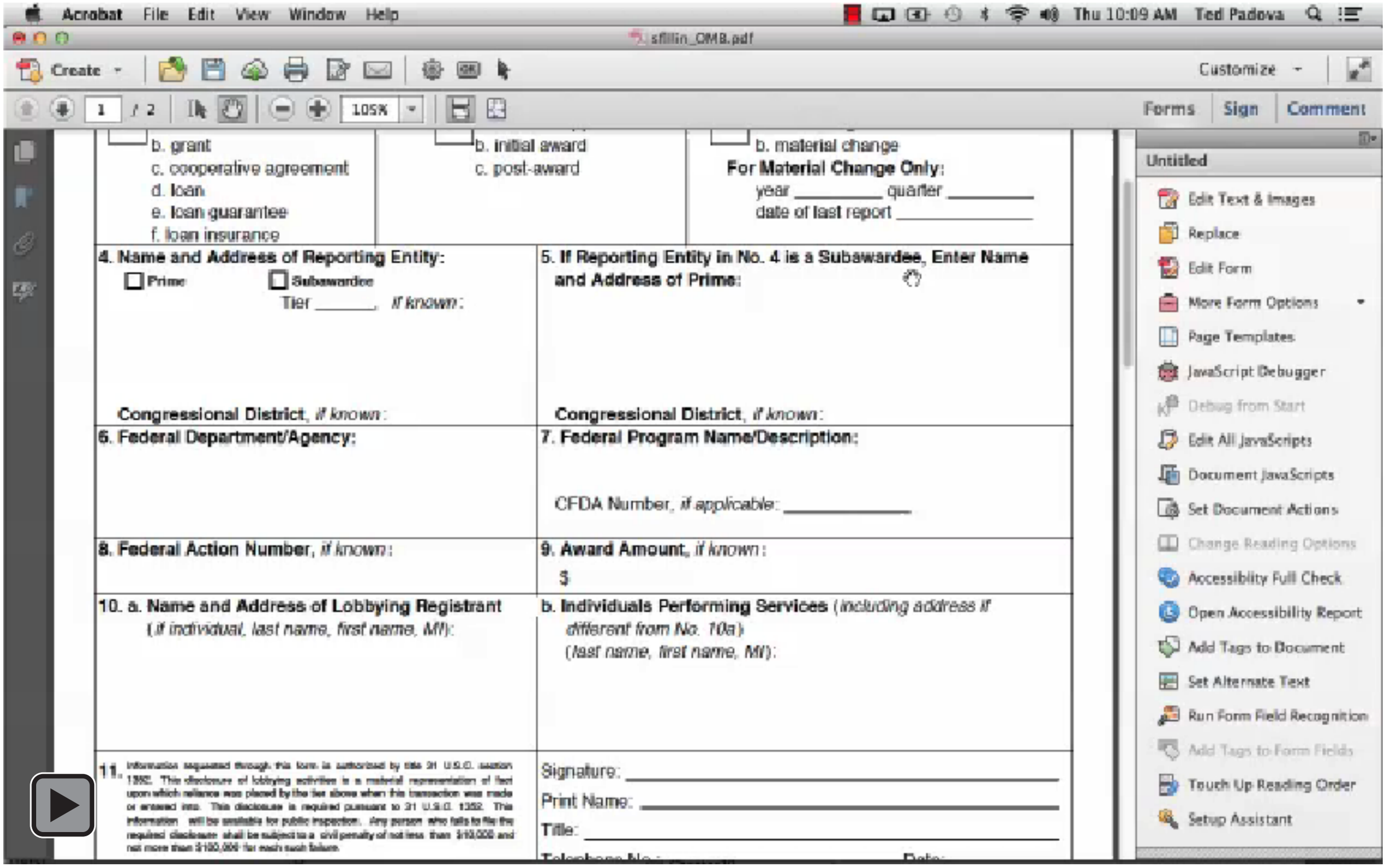


Check the Required check box in the General Properties

However, I find that it is often best to add all fields on a form, then later mark those fields that are required after the fact.

If all fields on a form are required, you can select all fields (with the Select Object tool active press CTRL/Command + A to select all fields), open the Properties window, click the General tab, and check Required. If you want to eliminate some fields that are not required, you select all fields, then press CTRL/Command and click the fields with the Select Object tool to deselect individual fields. With just the right fields selected open the General Properties and check Required.

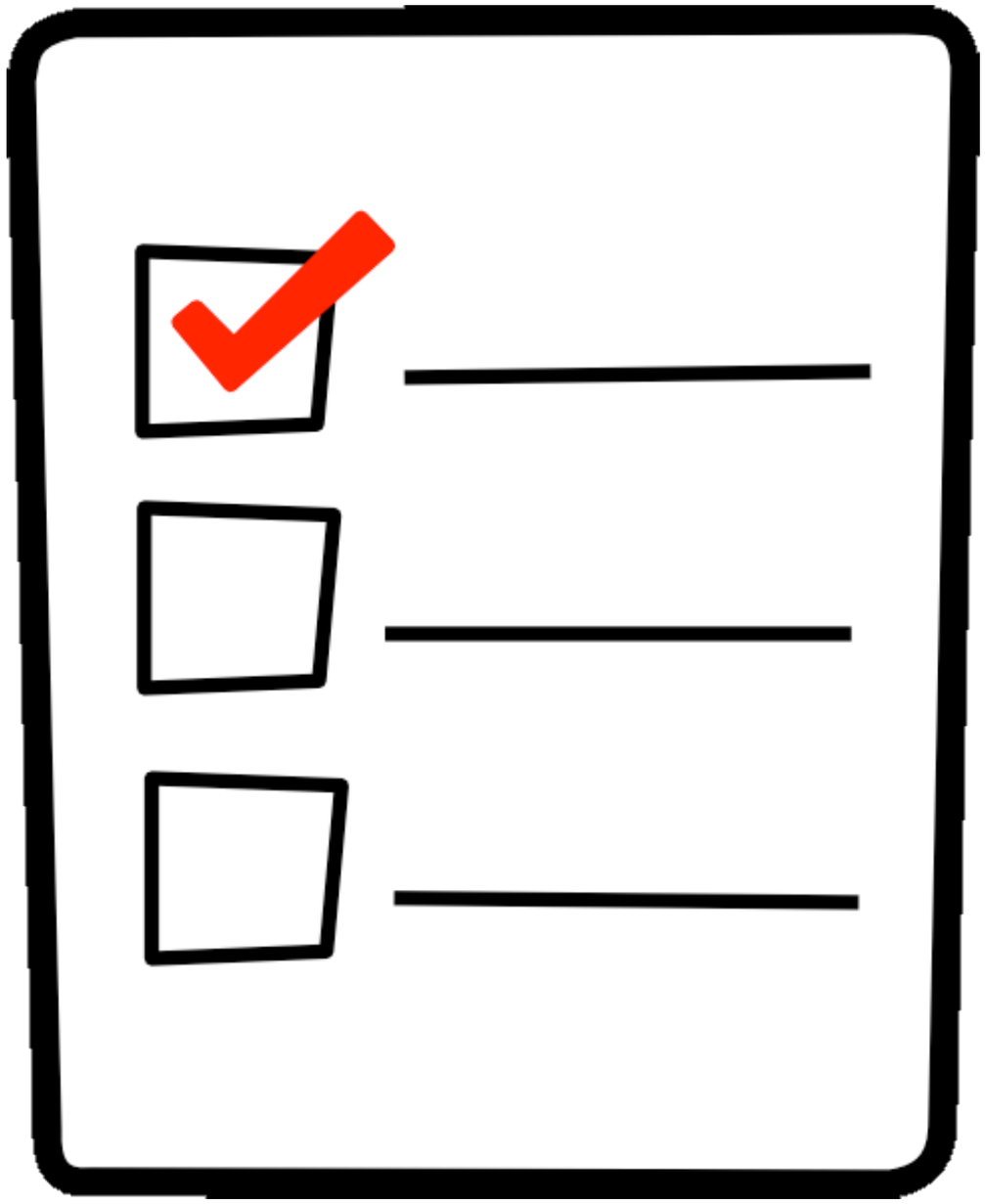
Movie


Acrobat File Edit View Window Help
Thu 10:09 AM Ted Padova
sflllin_OMB.pdf
Create
Customize
1 / 2
105%
Forms Sign Comment
b. grant
c. cooperative agreement
d. loan
e. loan guarantee
f. loan insurance
b. initial award
c. post-award
b. material change
For Material Change Only:
year _____ quarter _____
date of last report _____
4. Name and Address of Reporting Entity:
Prime Subawardee
Tier _____, If known:
5. If Reporting Entity in No. 4 is a Subawardee, Enter Name and Address of Prime:
Congressional District, if known:
Congressional District, if known:
6. Federal Department/Agency:
7. Federal Program Name/Description:
CFDA Number, if applicable: _____
8. Federal Action Number, if known:
9. Award Amount, if known:
$
10. a. Name and Address of Lobbying Registrant (if individual, last name, first name, MI):
b. Individuals Performing Services (including address if different from No. 10a) (last name, first name, MI):
11. Information requested through this form is authorized by title 31 U.S.C. section 1352. This disclosure of lobbying activities is a material representation of fact upon which reliance was placed by the tier above when this transaction was made or entered into. This disclosure is required pursuant to 31 U.S.C. 1352. This information will be available for public inspection. Any person who fails to file the required disclosure shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.
Signature: _____
Print Name: _____
Title: _____
Untitled
Edit Text & Images
Replace
Edit Form
More Form Options
Page Templates
JavaScript Debugger
Debug from Start
Edit All JavaScripts
Document JavaScripts
Set Document Actions
Change Reading Options
Accessibility Full Check
Open Accessibility Report
Add Tags to Document
Set Alternate Text
Run Form Field Recognition
Add Tags to Form Fields
Touch Up Reading Order
Setup Assistant

Chapter 11

Working with Check Box Fields

Creating nonexclusive and mutually exclusive check box fields



Working with Check Box Fields

IN THIS CHAPTER

Chapter 11

As I mentioned in Chapter 9, there are only two slight differences between check boxes and radio buttons. As a matter of choice you can use check boxes for all fields requiring mutually exclusive or non-exclusive responses.

In terms of form designs you should place check boxes to the left of text and questions. In some cases check boxes may work best on the right side of a line of text —particularly for long lines of text with check boxes placed at the end of the lines. If so, use the right side sparingly. Most importantly, you should be consistent. If you have some check boxes to the left of a line of text and some check boxes to the right of a line of text, users can easily become confused when filling out a form.

Creating Mutually Exclusive Check Boxes

A series of check boxes where you solicit responses for each item in a group are not, by default, mutually exclusive. A mutually exclusive group of check boxes permit only a single response from among the group. When one check box is checked, all other check boxes within a group are cleared of data and remain unmarked.

There's just one simple rule to follow when you create mutually exclusive check boxes. The field names need to be identical and the export values need to be different. If field names are different, each box in a group can be checked regardless of the export value.

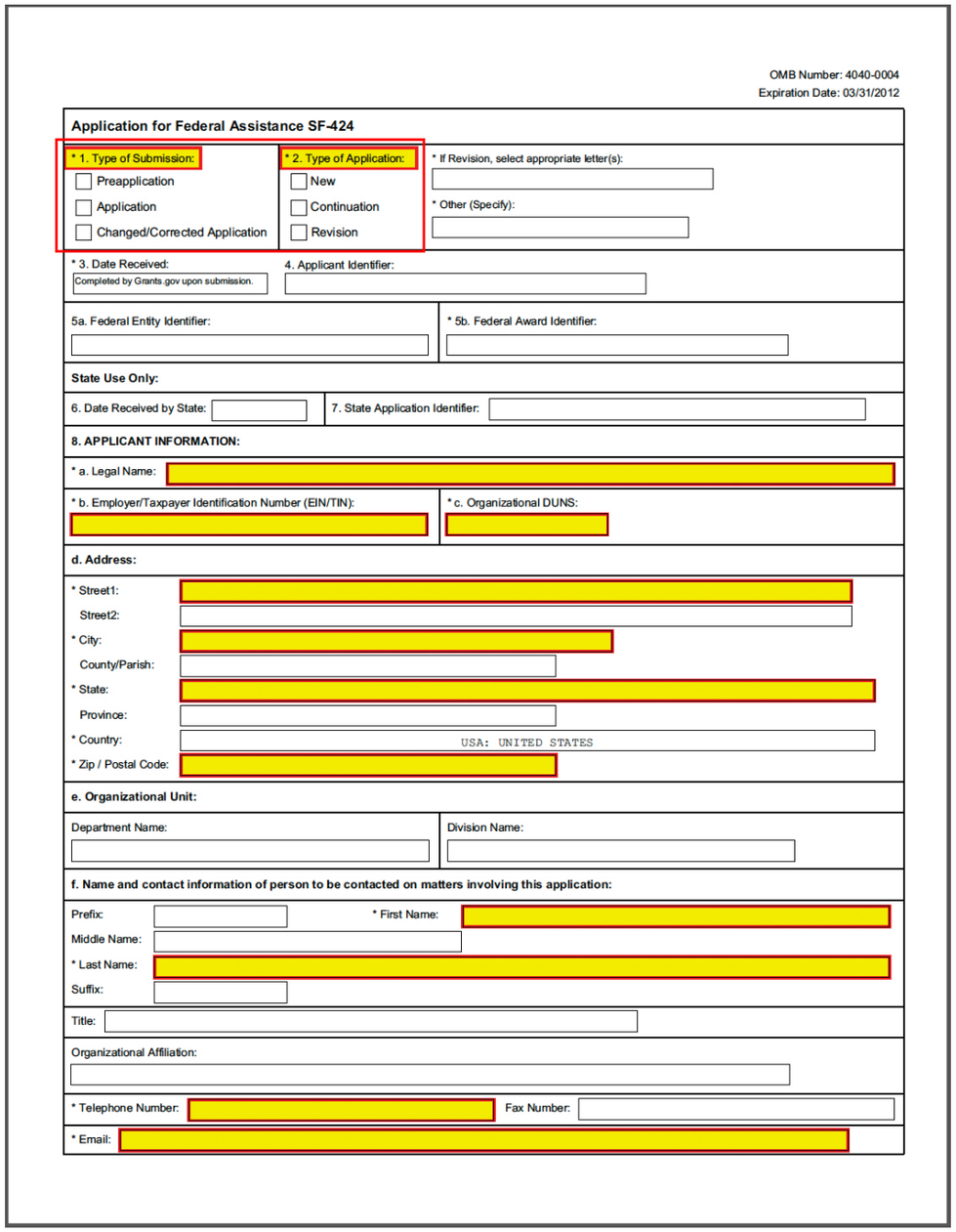
Note

On the sample form the check boxes are clearly visible and were created in the authoring application. Typically it's best to add strokes and rules in the design program rather than rely on Acrobat for borders. You have many more choices for line weights in your authoring application than what is available in Acrobat.

To create a group of mutually exclusive check boxes, do the following:

1. **Open a form containing check boxes.**
   In the sample form used by the US Department of Education (Figure 11.1) the first two sections (1 and 2) have three

Figure 11.1 Sample Form from US Dept. of Education


OMB Number: 4040-0004
Expiration Date: 03/31/2012

Application for Federal Assistance SF-424

* 1. Type of Submission:
☐ Preapplication
☐ Application
☐ Changed/Corrected Application

* 2. Type of Application:
☐ New
☐ Continuation
☐ Revision

* If Revision, select appropriate letter(s):
* Other (Specify):

* 3. Date Received:
Completed by Grants.gov upon submission.

4. Applicant Identifier:

5a. Federal Entity Identifier:

* 5b. Federal Award Identifier:

State Use Only:

6. Date Received by State:

7. State Application Identifier:

8. APPLICANT INFORMATION:

* a. Legal Name:

* b. Employer/Taxpayer Identification Number (EIN/TIN):

* c. Organizational DUNS:

d. Address:

* Street1:
Street2:
* City:
County/Parish:
* State:
Province:
* Country: USA: UNITED STATES
* Zip / Postal Code:

e. Organizational Unit:

Department Name:

Division Name:

f. Name and contact information of person to be contacted on matters involving this application:

Prefix:
* First Name:
Middle Name:
* Last Name:
Suffix:
Title:
Organizational Affiliation:
* Telephone Number:
Fax Number:
* Email:

Two sections at top left of form require check box fields

checkboxes each. The three checkboxes in each section need to be set up as mutually exclusive fields so only a single response is permitted.

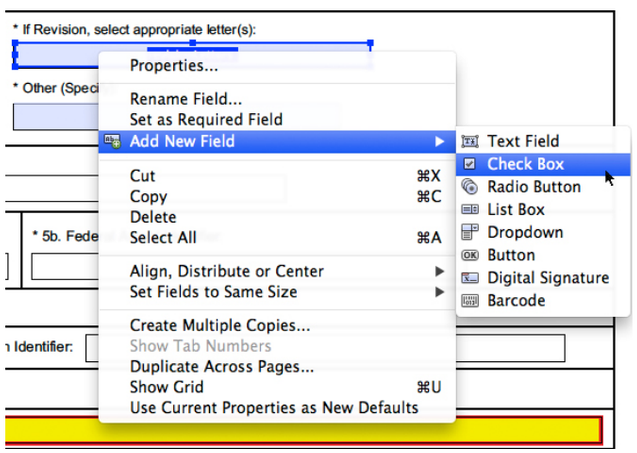
2. **Create a check box field.**
   If you elect to work in Normal view and you already have a field (any field) on a page, you can remain in Normal view while adding additional fields.

   The sample form contains text fields. Now I want to add a check box field. To do so, open a context menu on any field. From the context menu choose Add New Field. A submenu opens displaying a list of the field types. Click Check Box and the cursor is loaded with a check box. Click on the form to place the check box field.

   If you wish to add fields in Form Edit mode, click Edit Form in the Tools panel and click the Check Box tool in the toolbar at the top of the window. Click on the form to place the check box field.

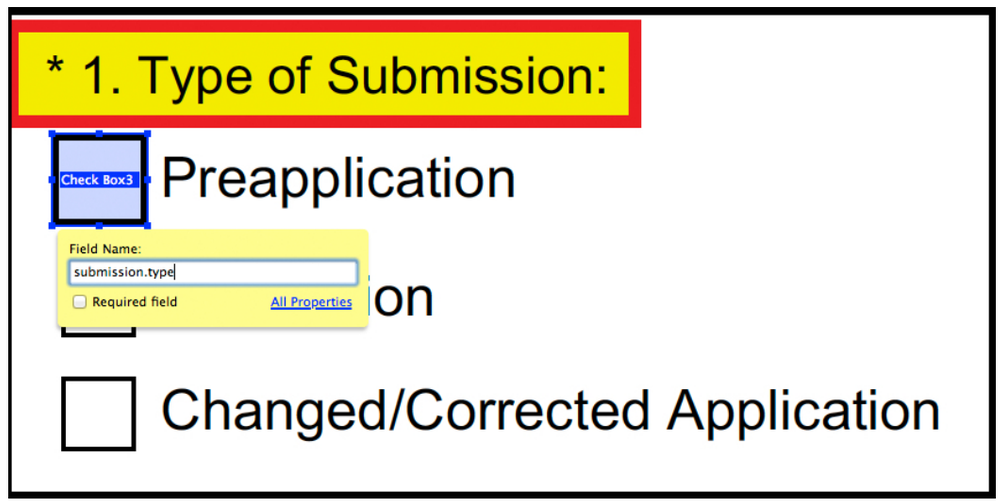
3. **Name the field.**
   You can type a field name in the mini yellow popup or you can type a field name in the Check Box Properties window. If you type a name in the mini yellow popup, click Properties to open the Check Box Properties window.

Figure 11.2 Add Check Box



Open a context menu on an existing field and choose Add New Field.

Figure 11.3 Mini Popup Window



You can type the field name in this window or in the Field Properties window.

In this example, I used *submission.type* for the field name. I use a hierarchical name however I need to edit the names separately when I change the field names.

Add a tooltip and click the Required button if at least one check box needs to be marked before the form is submitted.

Note

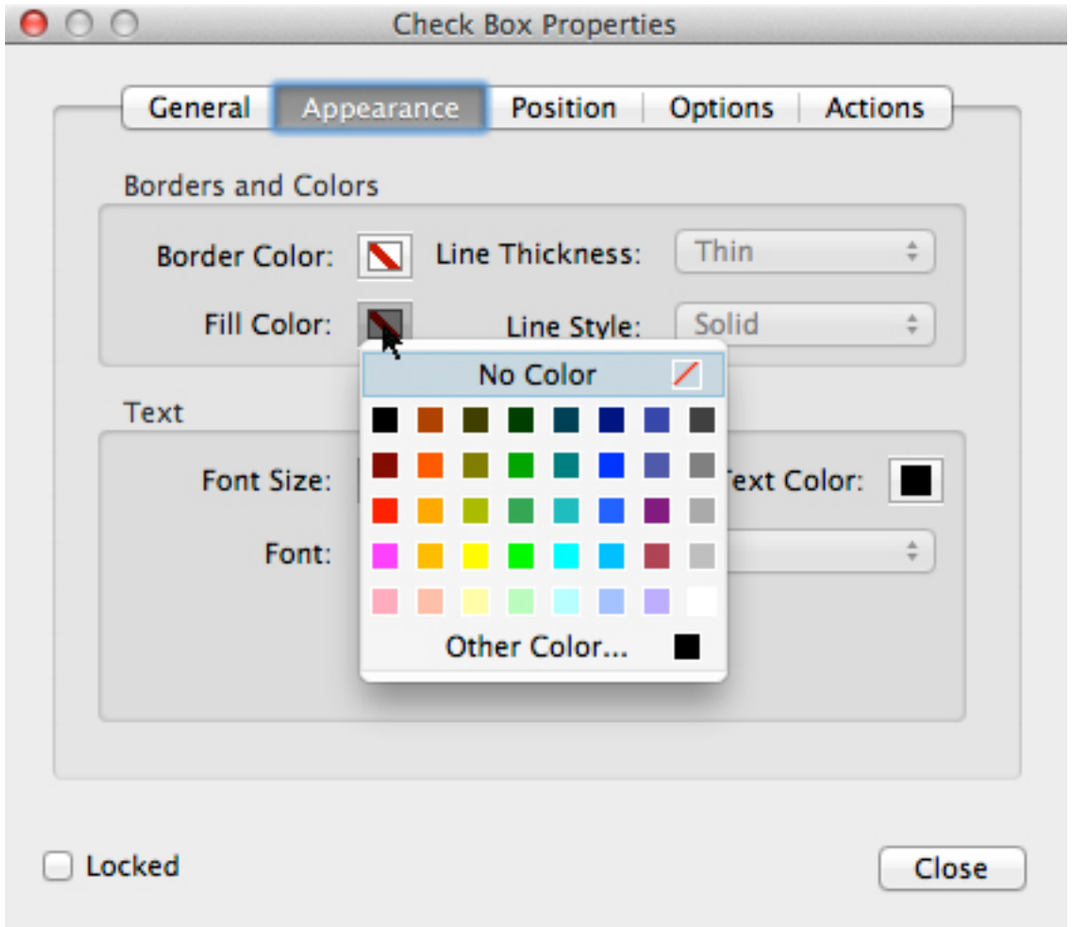
The Check Box Properties window appears very different than the Text Field Properties. As you work with the different field types the Properties window changes to reflect options specific to the field type in question. However the General properties for all field types contain the same information where you supply field names, tool tips, and choose field display options.

4. **Adjust appearances.**
   Click the Appearance tab. Remove the default Black stroke and white fill. If you design your forms with graphics in the authoring program you don't need any strokes or fills applied in Acrobat.

   For Font Size you can leave the default at Auto or you can exaggerate the mark by increasing the font size. If the check boxes are small you may want to have the response marks appear larger than the check boxes. The best way to determine what works well is try a size and examine the results by marking a check box. You can return to the Check Box Properties window and make changes.

Once you make appearance settings for Check Box fields open a context menu on a field and choose Use Current Properties as New Defaults. When you add additional check boxes on a form, the Appearance settings remain without a border and fill.

Figure 11.4 Check Box Appearance Properties



Click the color chip to open the popup menu.

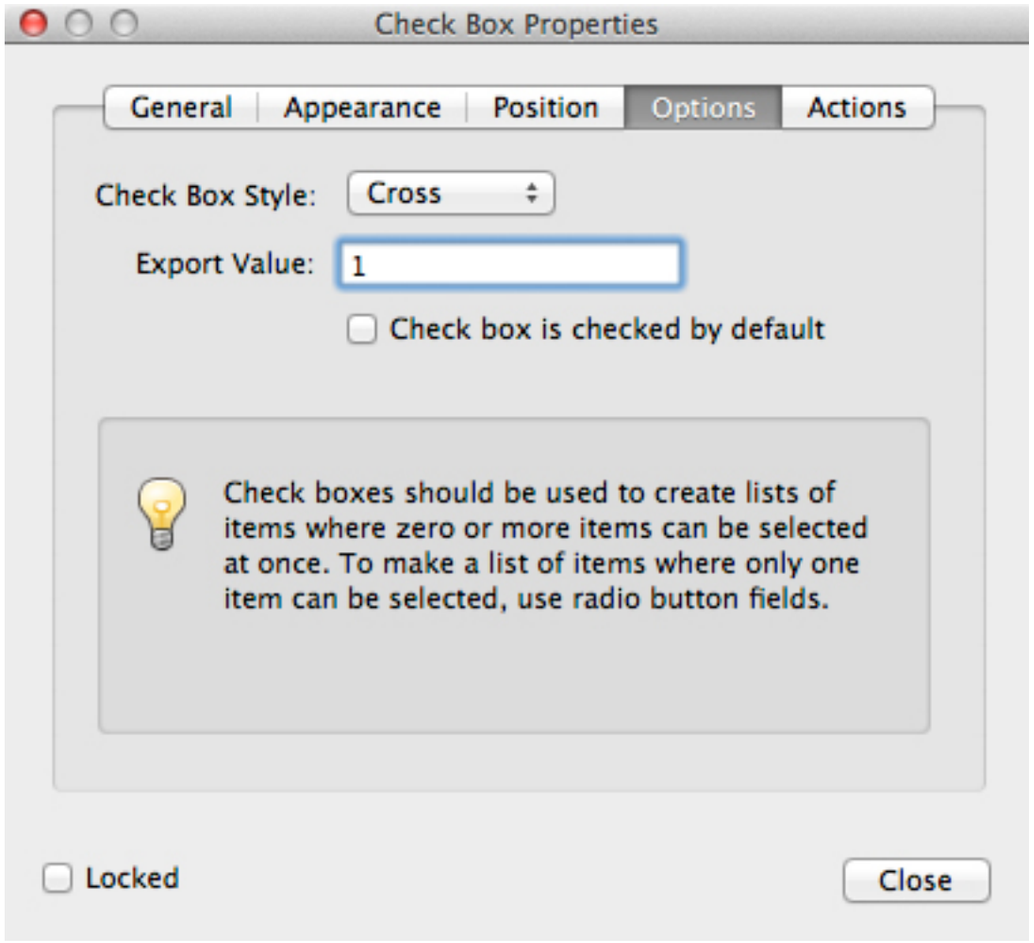
5. **Edit the field Options.**
   Click the Options tab. In the Options tab you find settings for Check Box Style and Export Value.

   For Check Box Style you have a number of choices for displaying a character when a check box is marked. You can use a Check Mark, Cross (X), Diamond, Square, Star, etc. What's important is be consistent. You should use the same style for all your eForms. I find the Cross style works well for the forms I design. The Cross places an X in the check box when marked. However, this is a personal choice and should be discussed in committee so all forms authors use the same style for all forms hosted on you agency site.

   The Export Value is critical for mutually exclusive fields. You must add unique text for each field within a group. The default is Yes. You can change the export value by typing text in the Export Value text box. For the sample form I use 1 for the first check box (Figure 11.5).

   When data are exported to a database system, export values can easily be changed in the export routine. By using a numeric value here I can easily copy/paste the check box fields and not need to worry about changing export values after I paste fields.

Figure 11.5 Check Box Options Tab



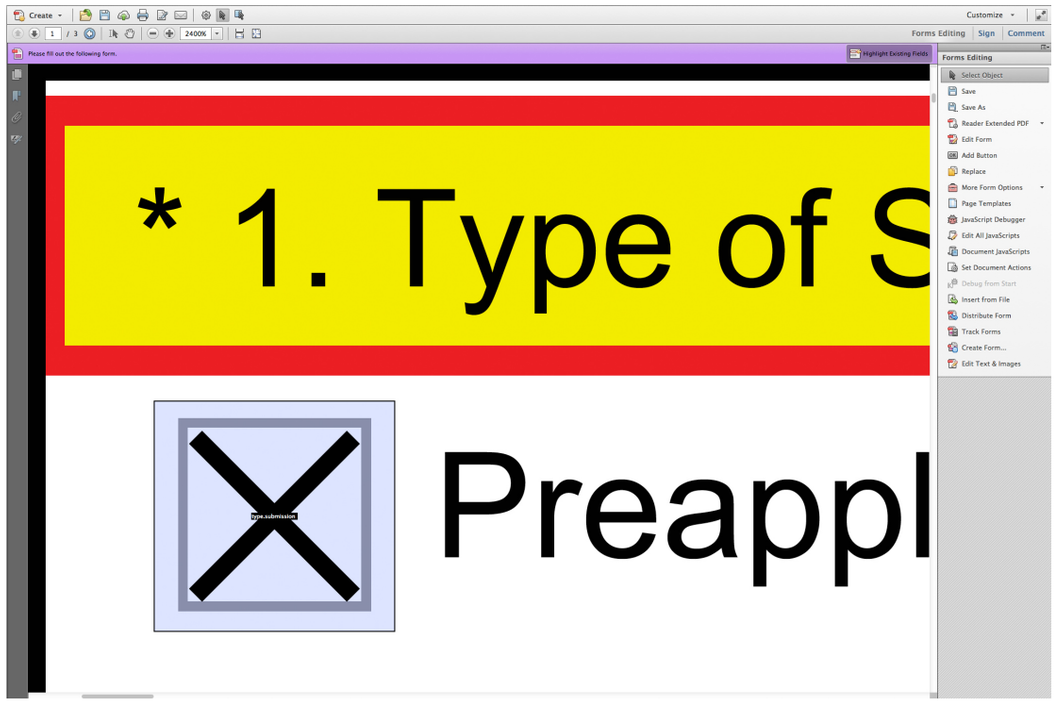
For this form I use 1 or the Export Value

If I choose to add export values such as Pre-application, Application, and Changed/Corrected Application for the check boxes in Section 1 on the sample form, I would need to change the export values when copying the checkboxes and pasting for Section 2. By using 1, 2, and 3 for export values, I won't need to change settings in the Options tab.

6. **Examine the field size.**

   Zoom in on the field so you can clearly see it in a zoomed view. You can press CTRL/Command + (plus) several times to zoom in on the field. Click the check box with the Hand tool and examine the mark. The field should be sized a little larger than the box displayed on the page. Notice in my example the field box is slightly larger than the graphic box.

Figure 11.6 Zoom View



Zoom in on the Check Box fields to clearly view the field object against the background box

When you add check boxes on forms you want the marks clearly legible and clearly in view if the user zooms out to a Fit in Window view.

After examining the mark, zoom out slightly so you can see the remaining check boxes on the form.

7. **Duplicate the field.**

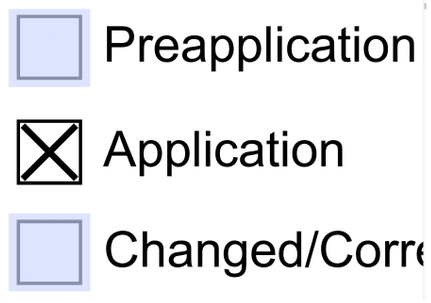
   After all property settings are made to the first check box field and the field is sized properly, you can duplicate the field and just make a change to the export value. All other settings remain the same for check boxes within a common group.

   Press CTRL/Option + Shift and drag the field vertically to duplicate it. In the Check Box Properties window type a new value for the Export Value. In this example I type 2 for the export value.

   Duplicate once again and add 3 for the export value for the third field.

When you examine the results only a single check box can be marked within the group.

Figure 11.7 Box Checked



Only one box in the group can be checked

Copying/Pasting Check Box Fields

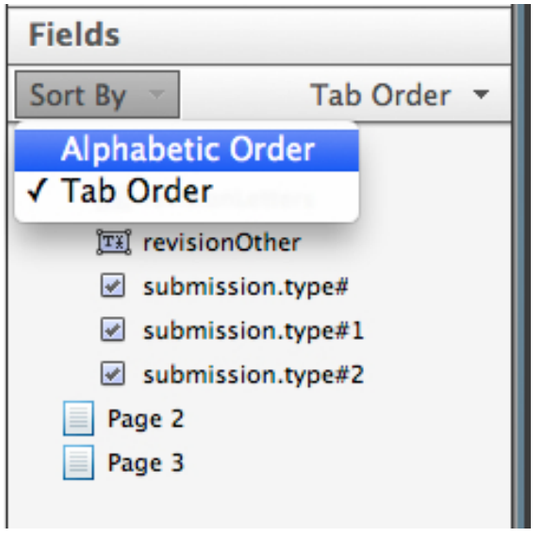
In my example I have three check box fields added in Section 1 on the sample form. Section 2 also requires 3 mutually exclusive check boxes. All the field attributes including the export values can be copied along with the check box fields. The only item that needs a change is the field names.

To copy the Section 1 fields and paste them for Section 2, do the following:

1. **Open Form Edit mode.**
   We need to first change field names in the Fields panel. Zoom out of the document so you can see the form in a zoomed out view.

2. **In the Fields panel in Form Edit mode, open the Sort By drop-down menu and choose Alphabetical Order**.

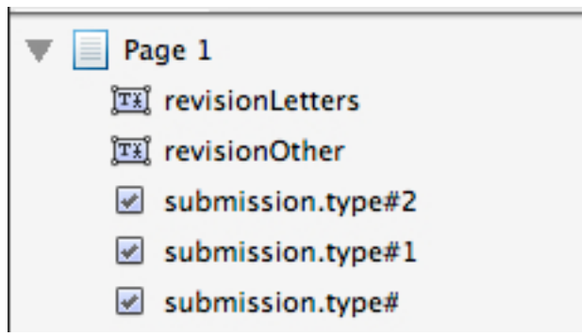
Figure 11.8 Sort Fields Alphabetically



In the Fields panel open the Sort By drop-down menu and chose Alphabetical Order

3. **Change the field names.**
   Click the check box field name. In my example three fields are named *submission.type*. Rename the three fields. In this example I use *application.type*.

Figure 11.9 Edit Field Names



Type new names for the three fields.

4. **Copy the fields.**
   Use the Select Object tool and drag through the three check box fields. Choose Edit ➤ Copy or press CTRL/Command + C on your keyboard.

5. **Undo the last edits.**
   The copied fields on the clipboard are named *application.type*. When you Undo, the last check box field returns to the original field name. Undo three times to revert the field names.

6. **Paste in Place.**
   If you choose Edit ➤ Paste the fields are pasted in the center of the form. You can paste the fields precisely from where they were copied and easily drag to the right constraining the

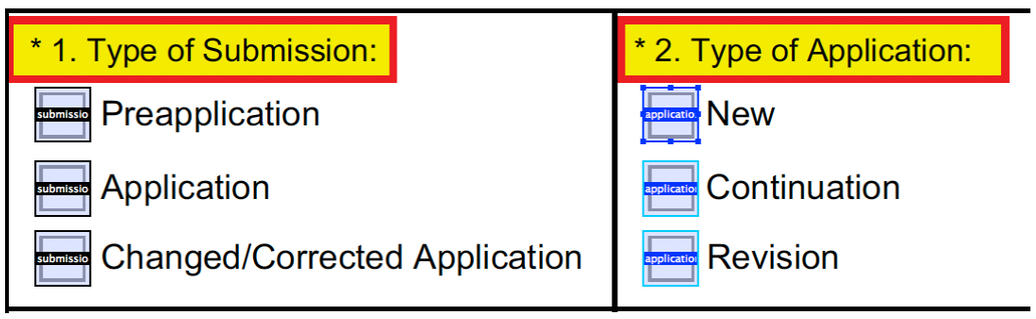
movement. This guarantees a perfect vertical alignment.
To Paste in Place, press the Shift key down on your keyboard and choose Edit ➤ Paste. The fields are pasted on top of the existing fields and remain selected. Be certain to click within a field box and press the Shift key down to constrain movement. Drag the selection to the right and move the fields in place.

Note

If you work in Normal view always keep the Select Object tool active when pasting fields. If the Hand tool is active and you paste fields, the fields are deselected when pasted. You must then attempt to select the pasted fields that may be surrounded by other fields making it difficult to select just the fields you need to move in place. If you accidentally paste with the Hand tool active, press CTRL/Command + Z to Undo. Click the Select Object tool and paste again.

7. Move the pasted fields to position.

Figure 11.10 Fields Moved to Position



After pasting press the Shift key and move the pasted fields to position.

Creating Non-exclusive Check Box Fields

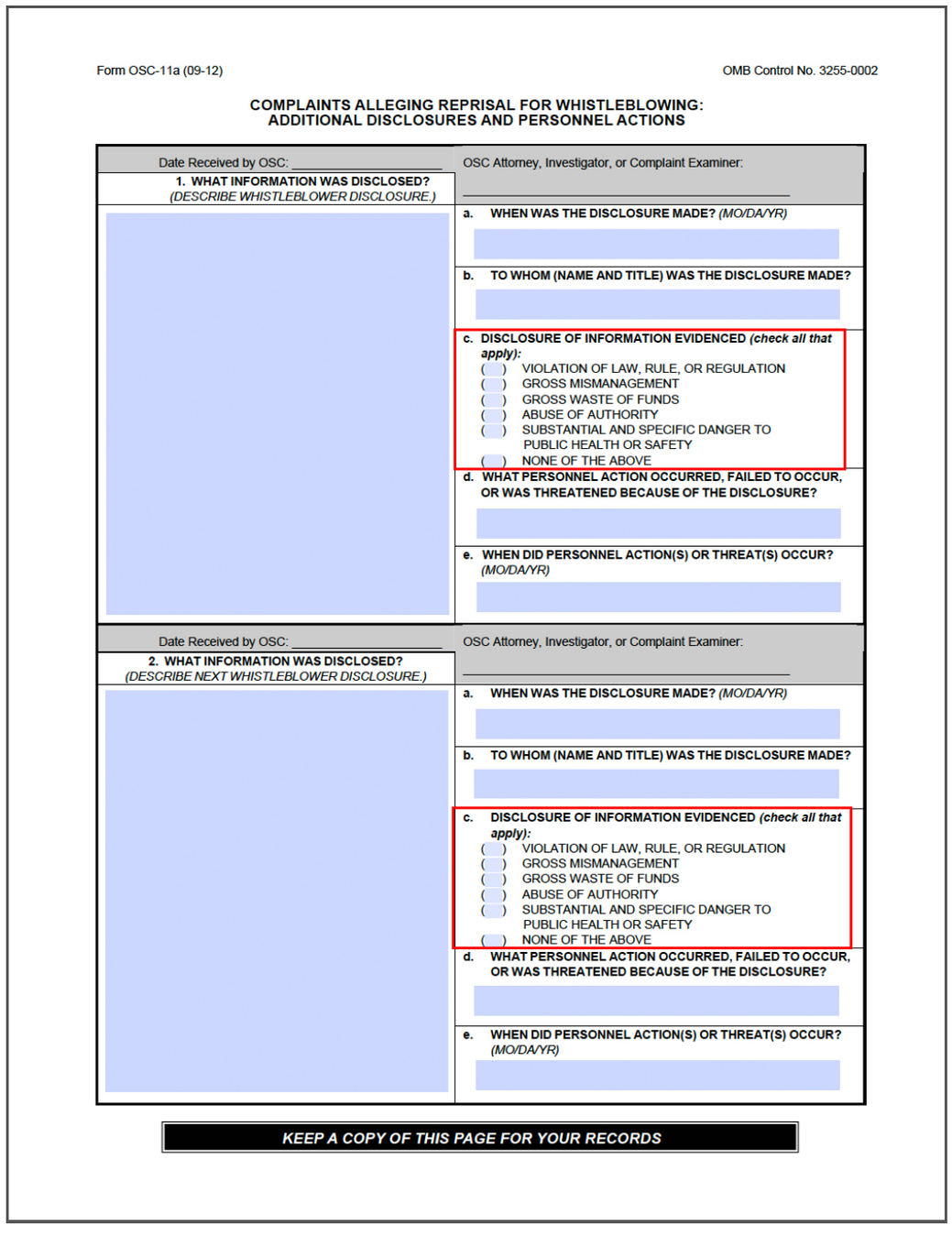
As you might suspect, non-exclusive check box fields follow the same steps as when creating mutually exclusive fields with the one exception that all fields need a unique name.

If you have a series of check boxes that require calculating data —something like a questionnaire where the user rates an item as good, average, poor, etc., you want to use unique names such as responseOne, responseTwo, responseThree, etc. for the different rows and hierarchical names for the columns such as responseOne.0, responseOne.1, responseOne.3, etc. fields across columns. Each name in a hierarchical order is a common name for the root followed by a number to make the field names unique. You use the hierarchical structure so the calculation formulas are much easier to write when you add JavaScripts.

More often than not you find an abundance of mutually exclusive check boxes on eForms. However, occasionally you find a series of check boxes where more than one response is needed within a common group of fields.

The US Office of Special Counsel hosts a complaint form where you find two sections containing non-exclusive check box fields. Notice in Section 1.c. and 2.c. there are six check box fields. At the bottom of each field group there is a check box for NONE OF THE ABOVE.

Figure 11.11 Sample Form



Form OSC-11a (09-12) OMB Control No. 3255-0002

COMPLAINTS ALLEGING REPRISAL FOR WHISTLEBLOWING:
ADDITIONAL DISCLOSURES AND PERSONNEL ACTIONS

Date Received by OSC: _____ OSC Attorney, Investigator, or Complaint Examiner: _____

1. WHAT INFORMATION WAS DISCLOSED? *(DESCRIBE WHISTLEBLOWER DISCLOSURE.)*

a. WHEN WAS THE DISCLOSURE MADE? *(MO/DA/YR)*

b. TO WHOM (NAME AND TITLE) WAS THE DISCLOSURE MADE?

c. DISCLOSURE OF INFORMATION EVIDENCED *(check all that apply):*
( ) VIOLATION OF LAW, RULE, OR REGULATION
( ) GROSS MISMANAGEMENT
( ) GROSS WASTE OF FUNDS
( ) ABUSE OF AUTHORITY
( ) SUBSTANTIAL AND SPECIFIC DANGER TO PUBLIC HEALTH OR SAFETY
( ) NONE OF THE ABOVE

d. WHAT PERSONNEL ACTION OCCURRED, FAILED TO OCCUR, OR WAS THREATENED BECAUSE OF THE DISCLOSURE?

e. WHEN DID PERSONNEL ACTION(S) OR THREAT(S) OCCUR? *(MO/DA/YR)*

Date Received by OSC: _____ OSC Attorney, Investigator, or Complaint Examiner: _____

2. WHAT INFORMATION WAS DISCLOSED? *(DESCRIBE NEXT WHISTLEBLOWER DISCLOSURE.)*

a. WHEN WAS THE DISCLOSURE MADE? *(MO/DA/YR)*

b. TO WHOM (NAME AND TITLE) WAS THE DISCLOSURE MADE?

c. DISCLOSURE OF INFORMATION EVIDENCED *(check all that apply):*
( ) VIOLATION OF LAW, RULE, OR REGULATION
( ) GROSS MISMANAGEMENT
( ) GROSS WASTE OF FUNDS
( ) ABUSE OF AUTHORITY
( ) SUBSTANTIAL AND SPECIFIC DANGER TO PUBLIC HEALTH OR SAFETY
( ) NONE OF THE ABOVE

d. WHAT PERSONNEL ACTION OCCURRED, FAILED TO OCCUR, OR WAS THREATENED BECAUSE OF THE DISCLOSURE?

e. WHEN DID PERSONNEL ACTION(S) OR THREAT(S) OCCUR? *(MO/DA/YR)*

KEEP A COPY OF THIS PAGE FOR YOUR RECORDS

Form containing non-exclusive choices

Before we create the fields we need to think about what we hope to accomplish. If a user clicks a check box then clicks the NONE OF THE ABOVE check box, we need to add a little error correction. We should set up the fields so a user can click any or all of the first five fields, but when NONE OF THE ABOVE is checked the first five fields should be cleared. We also need to be certain that if a user clicks any one of the five fields the NONE OF THE ABOVE field is cleared of a check mark. In other words we want to make it impossible for a user to mark both the first five fields AND the last field in each section.

Another issue we need to contend with is copying/pasting fields. We can't simply copy paste fields because attributes we use in the first set of fields to clear the fields can't be used in the second set of fields. For example, if we clear the first five fields with a Reset a form action, and copy/paste the fields, the action remains for clearing the first set. When we paste fields we need to change the Reset a form action to clear the second set of fields.

Taking into consideration the controls we need to implement on the form, the following explains how to tackle the design for these check boxes.

1. **Create a Check Box field for NONE OF THE ABOVE field.** The first five fields need to clear the NONE OF THE ABOVE field. Therefore I want this field placed on the form first so I can clear the field in the Check Box Field Properties before I duplicate fields.

In Normal view you can open a context menu on an existing field and choose Add New Field ➤ Check Box. If no field exists on the form open Form Edit mode and click the Check Box tool. When the cursor loads the Check Box click in the area where the NONE OF THE ABOVE check box needs placement. Size the check box by dragging handles in/out.

Note that you should zoom the view so you can place the first check box field precisely in position.

2. **Name the field.**
   In my example I use *disclosure1cNone* for the field name. This field is named without a hierarchical name.

3. **Set the field attributes.**
   In the Check Box Field properties window click the Appearance tab. Set the Border and Fill color to None (diagonal line icon). Click the Options tab and choose a check box style from the Check Box Style drop-down menu. My preference is the Cross choice however, you can choose any mark your agency decides to use for check box field styles. The default Export Value can remain as Yes.

4. **Create a second check box field.**
   In my example I want a check box at the top of the section adjacent to the first question. You can CTRL/Option + Shift and drag the NONE OF THE ABOVE field to duplicate it and carry the field attributes or you can use the Check Box tool and create a new check box. If you create a new check box, you need to define the same attributes.

5. **Change the field name.**
   Before you do anything, change the field name. If you duplicate a field and change any attribute before changing the field name, the new attribute setting is applied to the duplicate field and the original field.

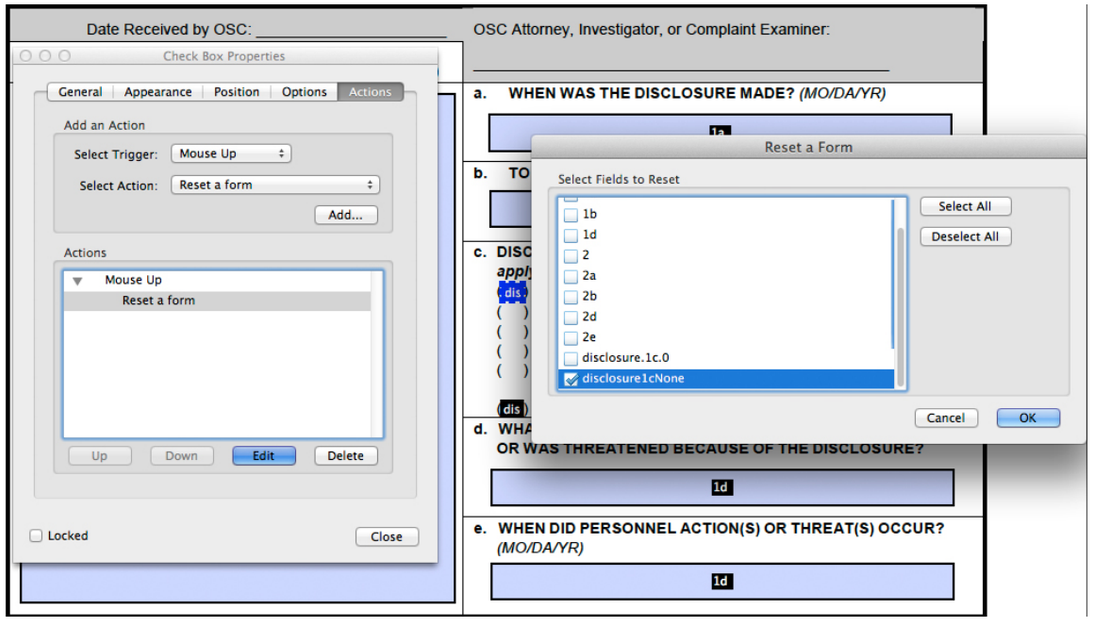
5. **Reset the Form.**
   In this example we want to clear the NONE OF THE ABOVE field. When this check box is checked, the last field in the group is cleared.

> **Note**
>
> *You can clear data by using an Action or by writing a JavaScript. When you become familiar with JavaScript, as I explain in Chapters 16-19, using a JavaScript will be the preferred method. However, this form is not complicated and doesn't contain many fields so using the Reset a form action works well.*

6. **Click the Actions tab in the Check Box Properties window.** From the Select Action drop down menu choose *Reset a form*. Click the Add button and the Reset a Form window opens.

Scroll the window and locate the field name you want to clear. In my example I want to clear disclosure1cNone (the NONE OF THE ABOVE field I first added to the form). Check the box adjacent to the field name and click OK.

Figure 11.12 Reset a Form Window



In the Actions tab choose Reset a form from the Select Action menu and click Add to open the Reset a Form window.

7. **Create multiple copies of the field.**
   The attributes for the first check box in the group are set including clearing data on the last field in the group. When we create multiple copies the copies likewise have the same action to clear data on the last field.

8. **Select the field you want to duplicate.**
   In my example the *disclosure.1c* field is the one I want to duplicate. Open a context menu on the selected field and choose *Create Multiple Copies*. In the Create Multiple Copies of Fields window enter the number of fields down/across. In my example I add 5 for Copy selected fields down and 1 for Copy selected fields across. Click OK and the fields are duplicated.

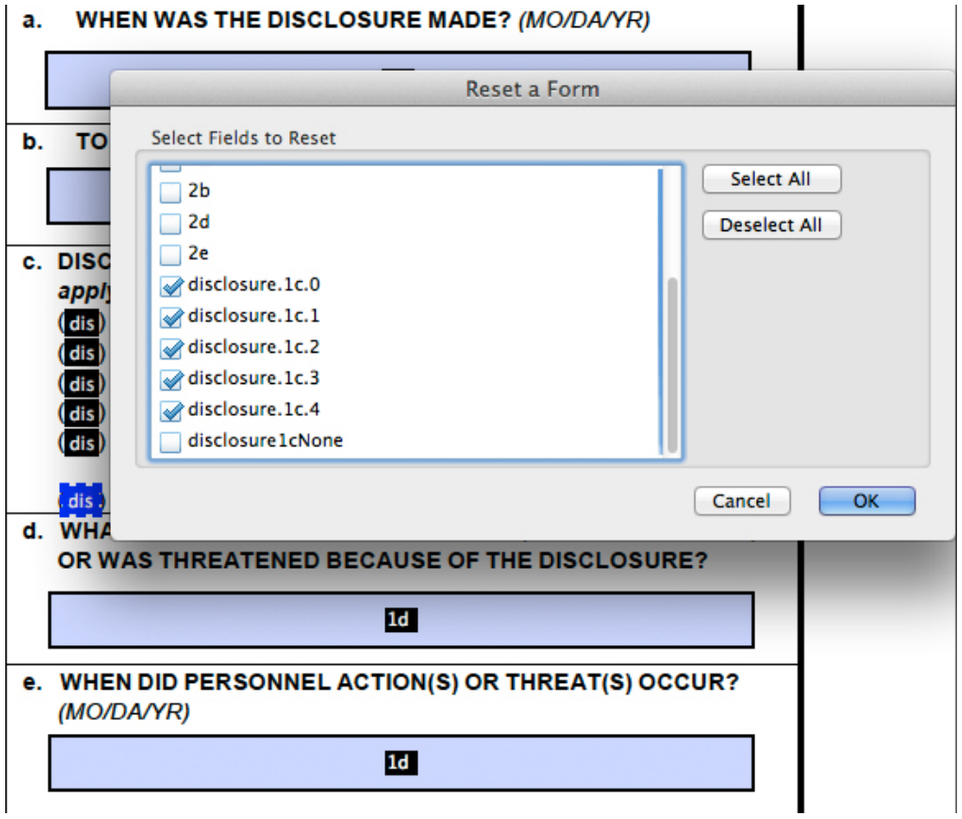
9. **Distribute fields.**
   Move the last field in the group to position and select all fields in the group. Open a context menu and choose Align, Distribute or Center ➤ Distribute Vertically. The fields should align to position.

10. **Set the attributes for the NONE OF THE ABOVE field.**
    At this point our first five fields are non-exclusive and when any of the fields are checked, the NONE OF THE ABOVE field is cleared of data. The user is prevented from choosing the last field then another field where both check marks are visible. However, the user can click one of the first five fields, then click the last field resulting in a check mark for one of the first five fields and the last field.

We want to be certain that if a user clicks the NONE OF THE ABOVE field, all fields above this field are cleared of data. We had to first create all the fields in order to set attributes for clearing the NONE OF THE ABOVE field.

11. **Select the last field and open the Check Box Properties window.**
    Click the actions tab and choose Reset a form from the Select Action drop-down menu. Click Add and check the first five fields in the Reset a Form window. Click OK and test the fields.

Figure 11.13 Resetting Multiple Fields



Check the fields to be reset

The user can click any or all of the first five fields and the NONE OF THE ABOVE check box remains unchecked. If the user clicks the NONE OF THE ABOVE check box, all fields are cleared and a mark appears only in the NONE OF THE ABOVE check box. If the user returns to one of the first five fields and checks a field, the NONE OF THE ABOVE field is cleared. Designing your forms to prevent user error is something you want to control whenever possible.

For creating the second section of fields, do not copy/paste the fields. You need to properly address all the field attributes and be certain clearing data is correct for the second group and doesn't interfere with the first group of check boxes. The best way to handle creating the second group of fields is to start over by adding new check boxes with the Check Box tool and follow the same steps as above

Working with Mutually Exclusive Fields in Tables

In Chapter 10 I talked about creating tables using the Create Multiple Copies command to populate a table with fields. The caveat when using this command is that it doesn't work when you have fields with identical names. As you know, mutually exclusive fields have identical field names but different export values. Therefore when you populate a table with mutually

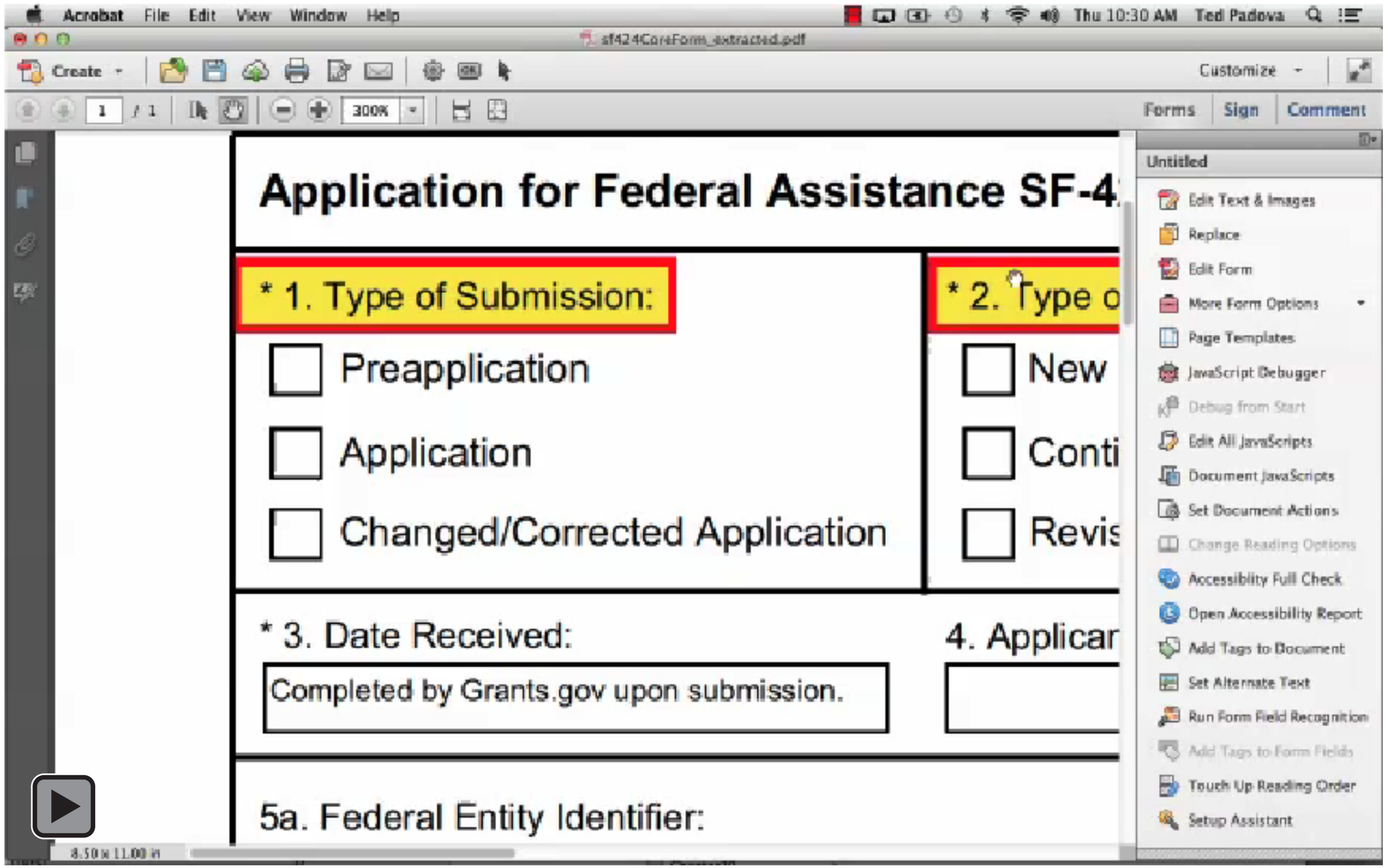
exclusive check box fields, you need to work within some of the limitations of Acrobat.

To create a table containing some mutually exclusive fields, follow the same steps as outlined in Chapter 10 for creating tables. However, exclude one of the mutually exclusive check boxes. For example if you have Yes/No columns. Create fields for all columns except the No columns. Populate the table using methods described in Chapter 10.

After the table is completed, Copy and paste the Yes columns, position the pasted fields in the table, and while the fields are selected, open the Check Box Properties. Click the Options tab and change the export value. For a No column change the export value from Yes to No.

There are many field attributes you need to change individually for a group of fields. However some attributes such as Export Value can be changed and applied globally to multiple fields.

Movie


Acrobat File Edit View Window Help
Thu 10:30 AM Ted Padova
sf424CoreForm_extracted.pdf
Create
Customize
1 / 1
300%
Forms Sign Comment
Application for Federal Assistance SF-4
* 1. Type of Submission:
Preapplication
Application
Changed/Corrected Application
* 2. Type o
New
Conti
Revis
* 3. Date Received:
Completed by Grants.gov upon submission.
4. Applicar
5a. Federal Entity Identifier:
8.50 x 11.00 in
Untitled
Edit Text & Images
Replace
Edit Form
More Form Options
Page Templates
JavaScript Debugger
Debug from Start
Edit All JavaScripts
Document JavaScripts
Set Document Actions
Change Reading Options
Accessibility Full Check
Open Accessibility Report
Add Tags to Document
Set Alternate Text
Run Form Field Recognition
Add Tags to Form Fields
Touch Up Reading Order
Setup Assistant

Chapter 12

Working with Buttons

Creating Buttons, adding labels, and adding button faces

Working with Buttons

Chapter 12

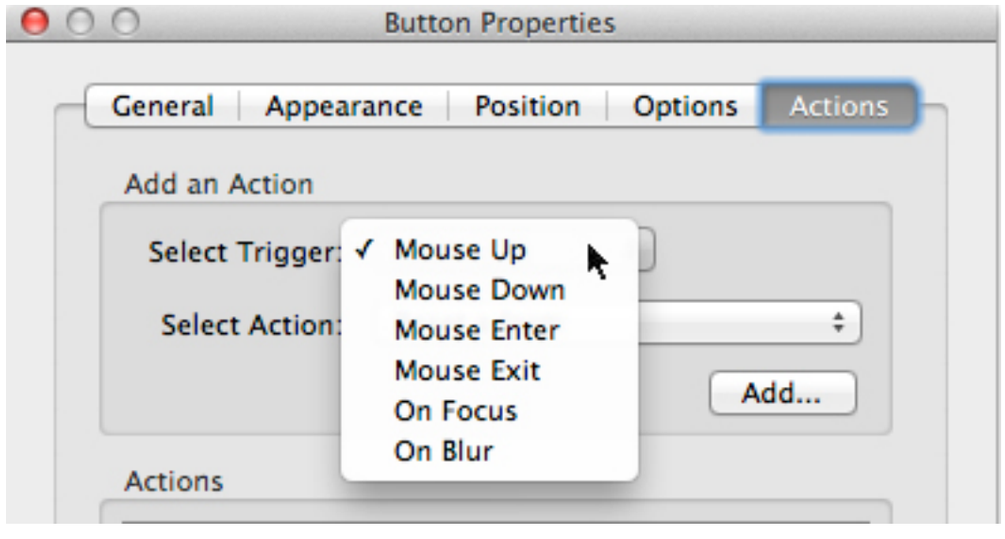
Buttons are an important part of every form. Most common are buttons used for clearing data on a form, printing a form, and submitting data to a host agency. Buttons can also be used for navigation through document pages, linking to other forms, and performing actions that add more dynamic functions for the end user.

IN THIS CHAPTER

Understanding Button Actions

Buttons, and similarly all form fields, include an Actions tab in the Field Properties window. Open the properties on any field type and click the Actions tab. In the Actions tab you find two items to address. The first choice is the trigger. You choose what mouse click or navigation to a field that triggers the action. Most common is the Mouse Up action. When you assign an action to the Mouse Up trigger, the action is invoked when the user releases the mouse button. You can use the Mouse Down action so the action is invoked when a user clicks but users do not expect this behavior. Hyperlinks on websites typically use a Mouse Up trigger. Therefore, if using a mouse click to invoke an action, always use the Mouse Up trigger.

Figure 12.1 Action Triggers



Select a Mouse Trigger for the assigned Action.

The remaining triggers are available for entering and exiting fields either through a mouse click (Enter/Exit) or via the tab key (Focus/Blur). These mouse triggers are very helpful when creating forms.

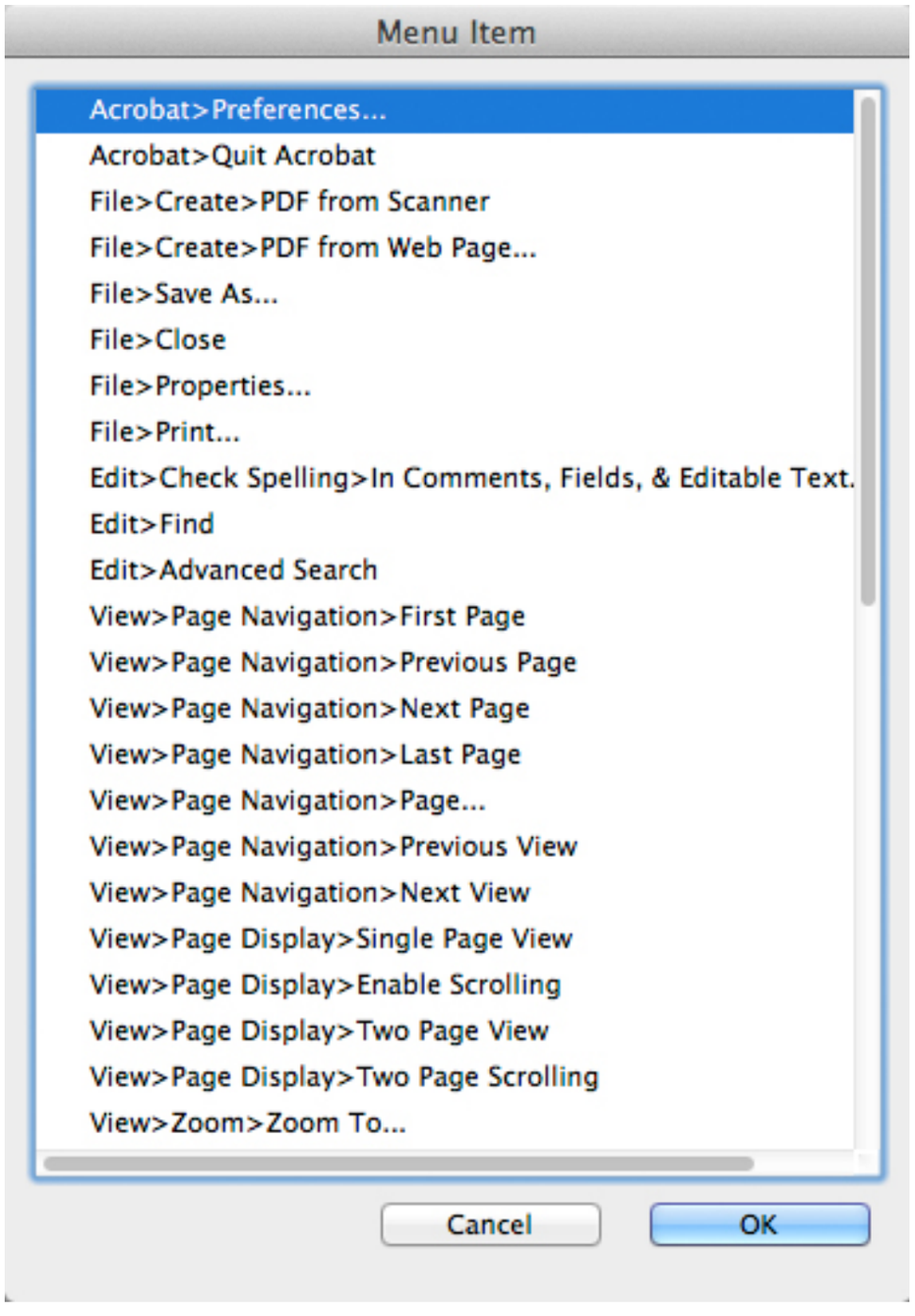
You may have a Mouse Enter trigger that opens a dialog box warning a user that responses in two or more fields are not filled in properly, then send the cursor back to a field and disallow the user to continue until proper responses are completed. You might use an On Focus trigger (when the user tabs to a field) to open a message box that informs the user what data are acceptable in a field. For example, you might inform users that no dashes are needed for credit card, telephone, or social security numbers.

After you choose a mouse trigger, you select what action is invoked when the user clicks the button. From the Select Action drop-down menu you find a long list of action types. Not all the actions are needed for eForms. Those actions of most interest to you include:

- **Execute a Menu Item.**
  This action invokes a menu command. In earlier versions of Acrobat we had access to all menus. Through recent Acrobat revisions many commands have been eliminated from the menu item choices.

  You might design a form where a key of information is contained within the form. The key might be many pages with codes and descriptors. It would be easier for the end

Figure 12.2 Menu Items



Choose Execute a menu Item and click the Add button to open the Menu Item window.

user if they use Acrobat Search to find an item where the descriptor and a code are listed on a page. You can use the Execute a Menu Item action that automatically opens the Advanced Search window. The user types search information and the search results are displayed in a window. One click takes the user to the page where the result is found.

There are a number of other menu item choices at your disposal (Figure 12.2). When you select Execute a Menu Item, click the Add button and review the menu choices Acrobat provides you.

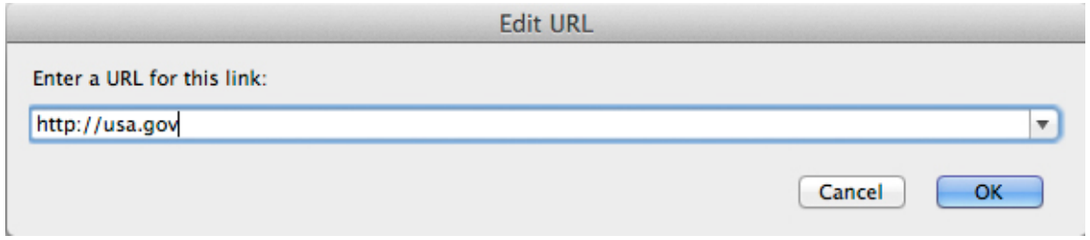
- **Go to a Page View.**
  This action is used for navigation. You can link to pages within a document or a page in another document. If you link to pages in other documents you must make certain the user downloads both pages from your website.

- **Import Form Data.**
  Although an action related to forms, this action is useless for Adobe Reader users. You won't need this action for forms you host on your website. You might however, use this action internally among Acrobat users in your office for importing data on a form.

- **Open a Web Link.**
  As you might suspect, this action opens a webpage. You might add a button on a form that opens a webpage where a

user finds instructions in an HTML document for filling in a given form or a link to an additional PDF that one can download from the agency's website. In most cases however, you can add text to your form, change the text color to blue to help the user discover the text as a link, and the text, written as a web link (*such as: http://www.county.gov*) when clicked, can open the respective webpage.

Figure 12.3 Open a Web Link



Type the complete URL in the dialog box.

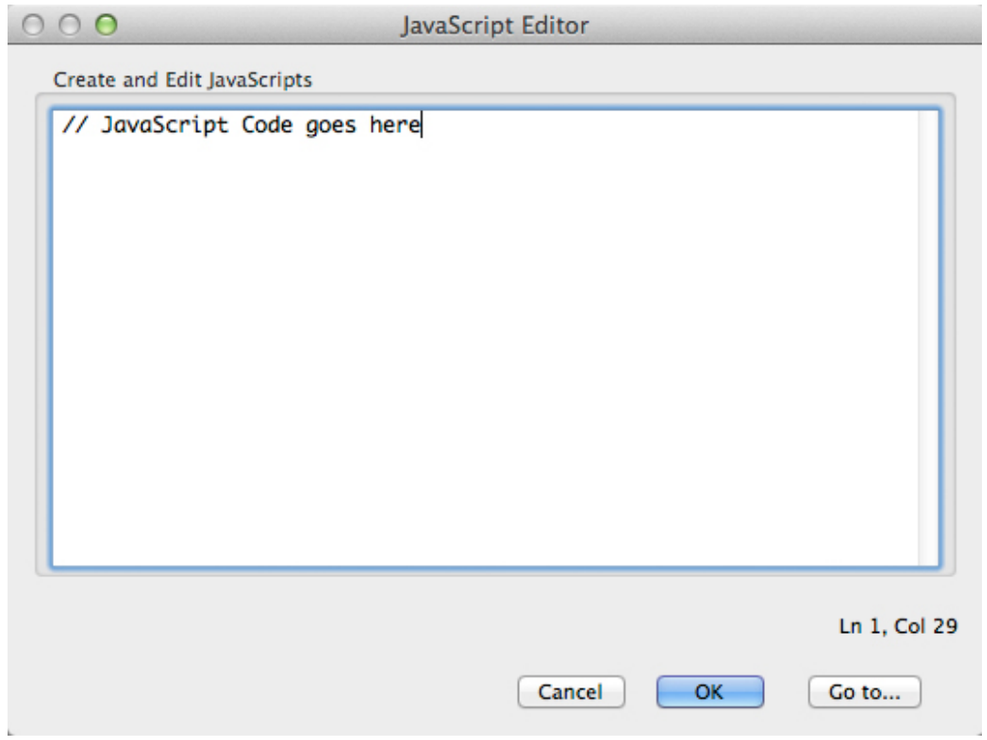
✦ **Reset a Form.**

This action is very common for forms. You use this action to clear data. When you add the action, a window opens where you select exactly what fields you want cleared.

✦ **Run a JavaScript.**

Once you become familiar with JavaScript this action is likely to be the most frequent action you'll use. When you add the action, a window opens where you type code to write the JavaScript.

Figure 12.4 JavaScript Editor



You can type and paste code in the JavaScript Editor window.

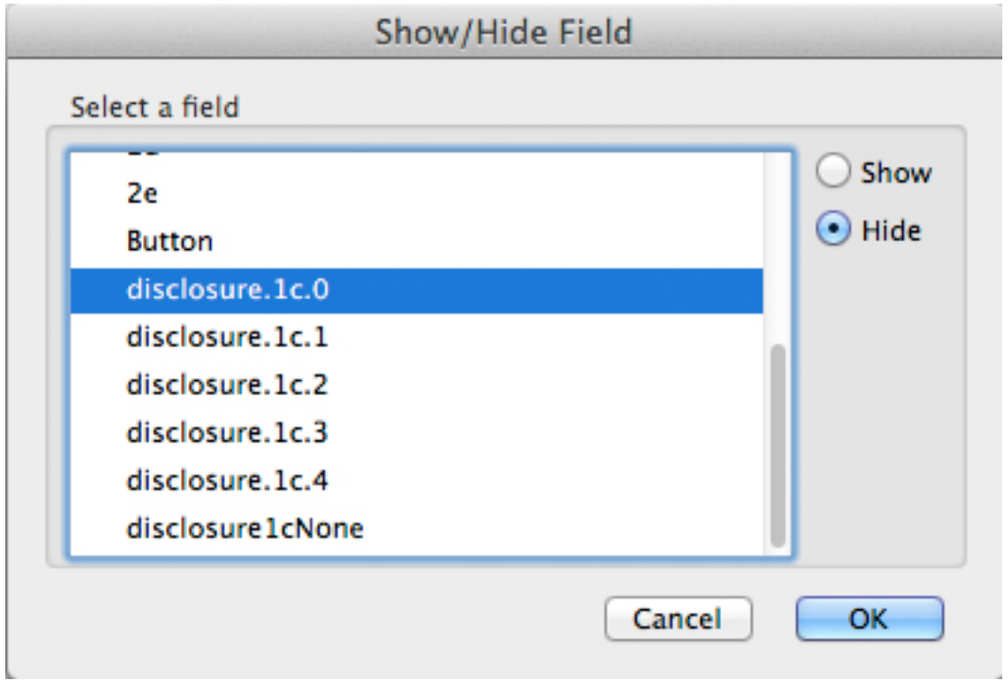
✦ **Set Layer Visibility.**

For multi-lingual forms this action is very helpful. You can design forms so all the graphic elements appear on a common layer. Each language text appears on separate layers and the fields appear on all layers regardless of layer visibility. You can use a button to invoke an action that displays the language layer of choice. For example, click English and the English text is visible. Click French and the English layer hides while the French text layer is made visible.

In order to design multi-layer forms you need an authoring program like Adobe InDesign that supports creating multiple layers. When you export from InDesign you have an option to export a layered PDF document.

- **Show/Hide a Field.**
  Showing and hiding fields is an action type that can be used in many ways for many purposes. You can show fields based on a user response (something like fields that pertain to gender, age, or other criteria). In this case a user response invokes the show/hide fields action.

Figure 12.5 Show/Hide a Field Dialog



Click the Show & Hide buttons then click the field names.

You might wish to hide fields that don't require user responses (something like official use only fields). You might also use show/hide fields for added security on a form. For example, after a user completes populating a form and before submitting a form, personal data are hidden on the form.

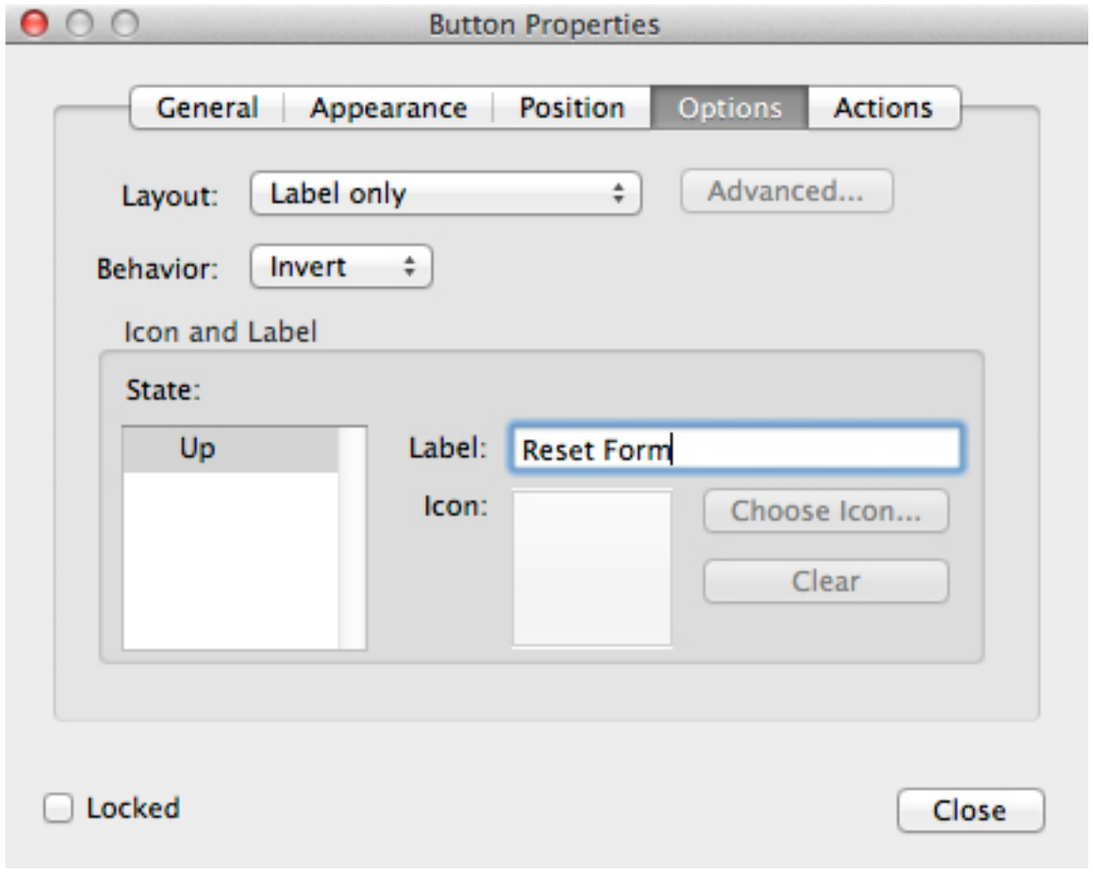
There are a number of uses where the show/hide field action is applicable. When you add the action, a list of all fields on your form are displayed in a window and a choice for showing or hiding the fields. You can individually check which fields are affected when the action is invoked.

- **Submit a Form.**
  For eForms this is a critical action type and should be included on all forms hosted on government websites. When you add the action you are presented with a window where you make choices for where the data are sent. You can choose to have data emailed back to the host or data submitted to a website. If you choose to submit data to a website additional programming is needed from your IT department to set up data collection and routing.

Using Button Labels & Faces

When you add a button field to a form, the Options tab in the Button Properties window offers you three choices for a button appearance. You can use a label (a line of text), an image, or both a label and image. In the Options tab you have choices for what label you want to use, how you want to position the objects, and some behavior settings.

Figure 12.6 Button Label



In the Options tab I chose Label only from the Layout menu. Text typed in the Label text field appears inside the button.

Designing Forms with Button Labels

A common use for text labels is when you want to add reset and submit form buttons. You don't need a graphic or button design in order to create a button field with a label that informs the user what action is invoked when the user clicks the button.

Inasmuch as Acrobat does permit you to use text labels that describe the button uses, you might wish to create graphic images for common buttons used on your forms. You can create such images in illustration, photo editor, and layout programs and import the graphics for the button labels. Using graphics helps you insure that all the labels are identical on your forms and all departments designing forms use the same objects.

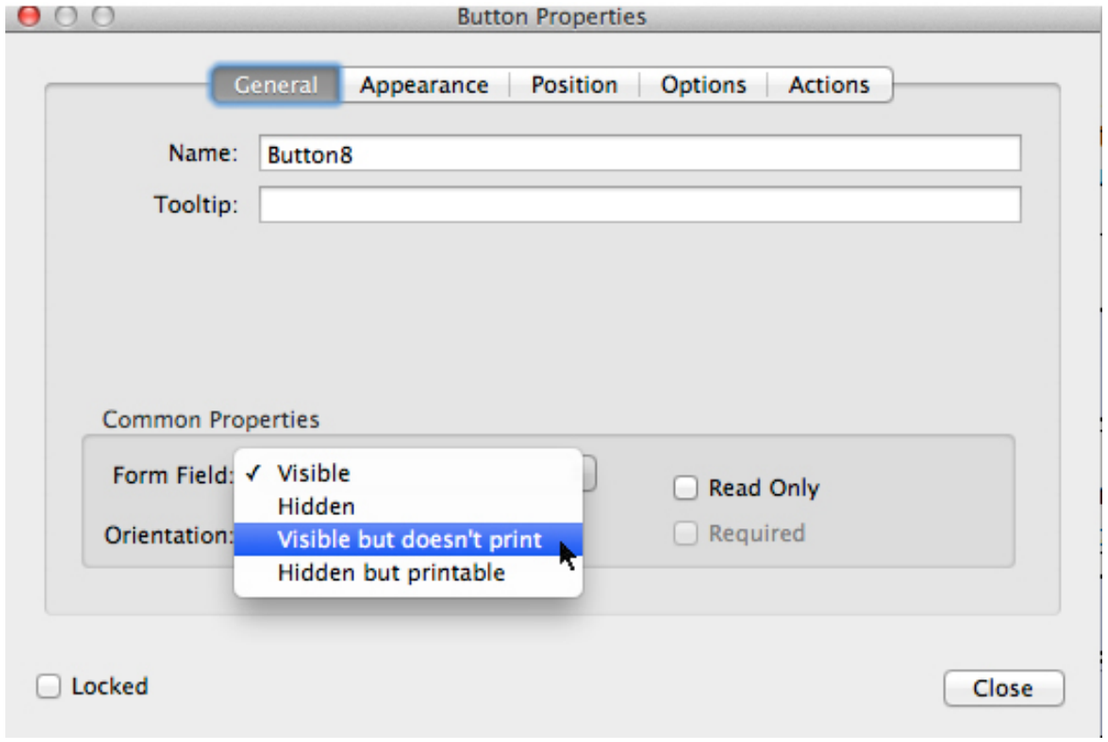
At the authoring stage, these button labels are added in Acrobat before the form is deployed. The end user has no control over changing the labels.

Setting Button Visibility

If you do need to print a form, in most cases the form should be printed without showing buttons. You can easily determine what fields are displayed on printed forms in the General Properties for each field type.

Open the field properties and from the Form Field drop-down menu choose Field is visible, but doesn't print. All fields marked with this choice won't appear on printed forms.

Figure 12.7 Showing/Hiding Fields on Printing



Open the Form Field drop-down menu in the General Properties and choose an option for field visibility.

Importing Button Faces

Prior to Acrobat XI, Adobe Reader users could not change button faces. If you designed a form in Acrobat with a button and JavaScript you could permit the Acrobat users to click a button and import a file —thus changing the button face. The file types you can import in Acrobat include most image formats such as JPEG, GIF, PNG, TIFF, etc., and PDF.

In Adobe Reader XI we find support for importing button faces similar to what has been available only to Acrobat users. However, importing a button face in Reader XI is limited to PDF documents. Any file (illustration files, image files, text documents etc.) converted to PDF can be imported as a new button face in Adobe Reader XI. But the files must be in PDF format.

You may have forms that require a passport photo. In the past, government offices needed to acquire photos as attachments to forms or delivered in an office as a printed photo. Now, you can design forms so the photos appear within a form. Likewise any other type of graphic can be imported as long as the graphic is saved as PDF.

Adobe may soon change the limitation for Reader users and allow more formats acceptable when importing button icons. Currently with Adobe Reader XI.0, you can host a webpage informing users that they can save image files as PDF and import the images in forms using Adobe Reader XI.

As is the case with saving form data, using the full complement of comment tools and several other features, Adobe Reader XI users do not need to work with enabled files. Save a PDF with a button and the appropriate JavaScript, and the Reader user can import content for the button face.

I discuss how to accomplish writing the script and setting up the buttons for importing content using Adobe Reader XI later in Chapter 19.

Duplicating Button Fields

Unlike other field types, button fields do not require unique names. You can use a button field with the same name on a page or several pages in a form and vary content and actions within each field independently. Although it is acceptable, it is best to use unique names unless duplicating fields across pages. If you wish to delete or copy/paste multiple buttons you can use hierarchical names. If you name several fields with names like *button.0*, *button.1*, *button.2*, etc. you can open the Fields panel, select the root name *button* and delete the fields or copy them.

When you want to duplicate items across pages, Acrobat provides you a command that enables you to add duplicates within specified page ranges. If you have a button that performs an action on a given page (such as opening the Advanced Search window) and you want the same action applied to buttons on a range of pages, you can duplicate the button across the pages where you want to invoke the action.

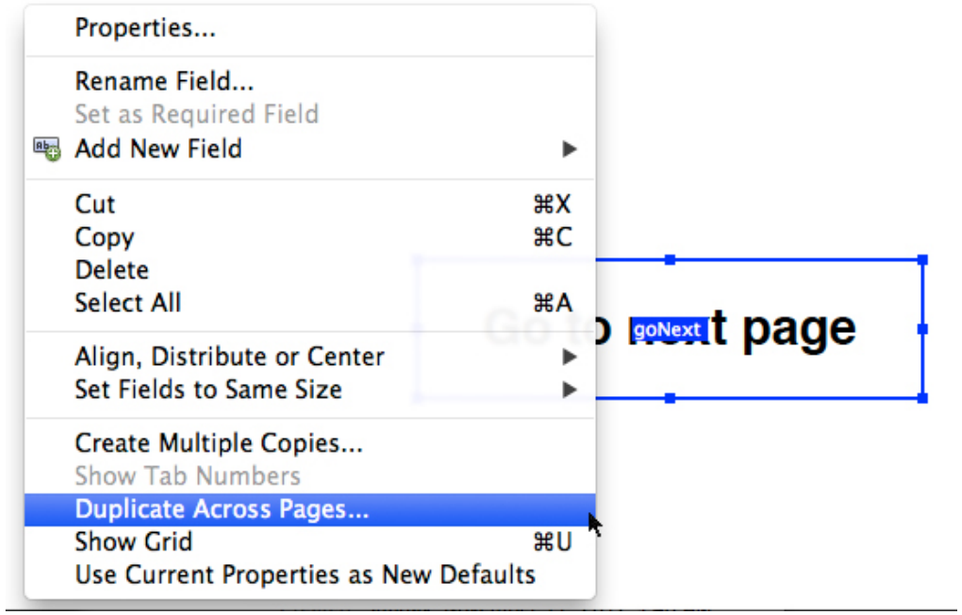
To duplicate buttons (or any field type) across a range of pages, do the following:

1. **Open a form containing several pages.**

2. **Add a button field to a page and set the attributes in the Button Field Properties window.**
   You should visit the Options tab in the Button Properties window and add a label, an icon, or both to the field.

3. **Duplicate the field.**
   Open a context menu on the field and choose Duplicate Across Pages.

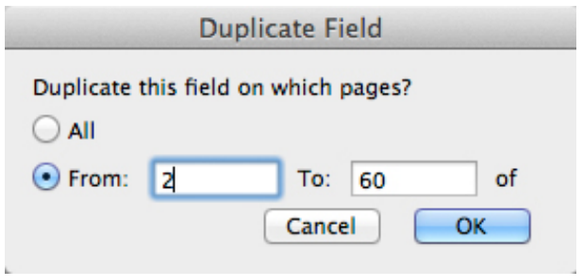
Figure 12.8 Duplicate Field Across Pages



Open a context menu and choose Duplicate Across Pages

The Duplicate Field dialog box opens. You have choices for duplicating the field across all pages or within a specific range of pages.

Figure 12.9 Duplicate Fields



Type the page range in the From/To text boxes.

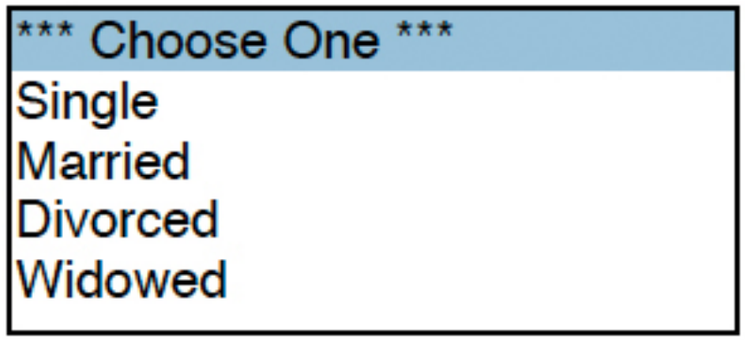
4. **Choose the option you want and click OK.**
   The field is duplicated across the pages, and all the attributes assigned to the original button are equally assigned to the duplicate buttons.

Be aware that the field names won’t change for the duplicated fields. Buttons can have different attributes assigned without changing the field names. Other fields like text fields however, require changing the field names if you want unique data in the fields.

Chapter 13

Working with List Boxes and Dropdown Menus

List Box and Dropdown fields have some similar characteristics and some distinct differences



Chapter 13

IN THIS CHAPTER

List Boxes and Dropdown menus (formerly called Combo Boxes in earlier versions of Acrobat) afford you two advantages when populating forms with fields. First, both items can help conserve space on your form while supplying a number of fixed response options. Items are listed in a box or menu requiring much less space than separate fields placed on a page. List Boxes are scrollable lists in a window and Dropdowns function similar to drop-down menus in application programs where a list falls down on a mouse click and the user makes a selection from the list.

Secondly these field types also offer you a means for restricting responses to options you determine in the list or menu. A question for type of business on a taxpayer form, for example, can restrict responses such as Cooperative, Corporation, Partnership, Limited Liability Company, and Sole Proprietorship. You can also determine if you want the user to supply additional responses (Dropdowns only) or you can lock out any editing and limit responses to those you provide.

List Boxes and Dropdowns have some common features when setting field properties that include:

- Both lists are created in the Options tab and the items in the lists can be sorted alphabetically or arranged manually.
- List Boxes are fixed to the items appearing in the box and no user-supplied data can be added.
- Dropdowns can be restricted to a list or you can choose to permit users to type additional information.
- List Boxes can be set up so users can select multiple items in a list or they can be restricted to a single choice.
- Dropdowns cannot be set to accept multiple responses. Only a singe item in the list can be selected for a response.
- List Boxes require more space on a form. They should be large enough for several responses to appear within the list window.
- Dropdowns can be designed with a single item visible in the field box. When the user clicks the field the list drops down to reveal more choices.
- The item you last selected in the Options tab in a list becomes the default shown on the form for both List Boxes and Dropdowns. For example, if you add items in a list such as *Apprentice*, *Journeyman*, and *Master* and you have *Master* selected in the Options properties when you close the Properties window, that item appears selected in the list as the default. If you reset a form, the default always prevails.

For Dropdowns especially, more often than not, you want the first item selected so the list falls down revealing all items below the selected item. You might add an item at the top of your list described as *Choose one..*, *Select one…*, etc. If you add an option like this, be certain to select the item in the Options tab before you exit the properties window so it becomes the default.

Choosing the first item in a list is not always true however. You may have a List Box or Dropdown listing countries for place of residence. If most of your forms come from your country, you might select your country name in the list before you exit the Options tab. Each time the form is opened, your country appears selected in a List Box or Dropdown menu.

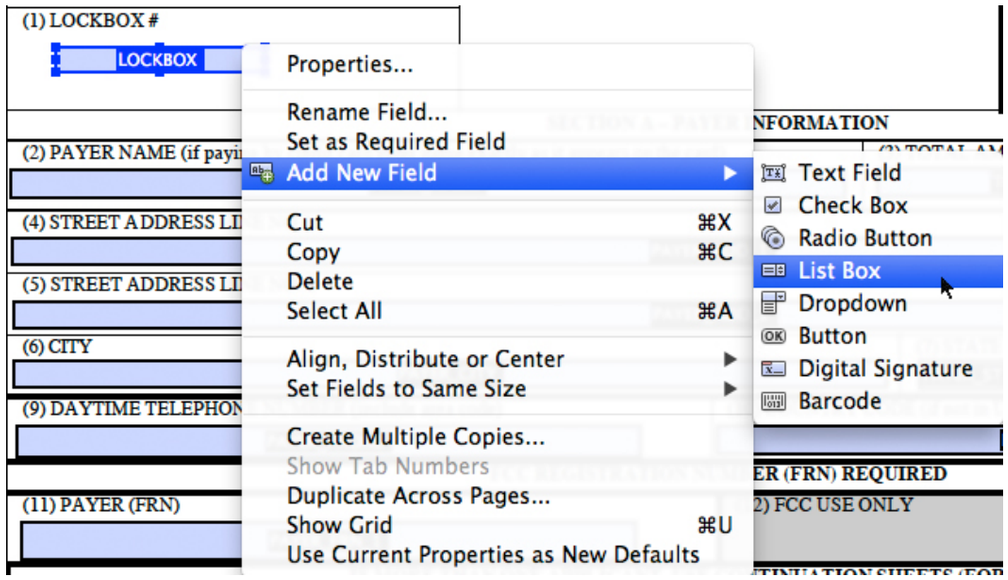
Creating List Boxes

When creating List Boxes you want to examine the form and decide whether you have enough room for a box. A list box displaying a single response in the field rectangle is awkward and makes it difficult for users to scroll the list. You should plan on sizing the list box large enough so at least three responses appear on the form.

To create a list box, do the following:

1. **Select the List Box tool and draw open a rectangle where you want the field placed on the form.**
   You can work in Form Edit mode or Normal view. If you're in Normal view, open a context menu on an existing field and choose Add New Field ➤ List Box.

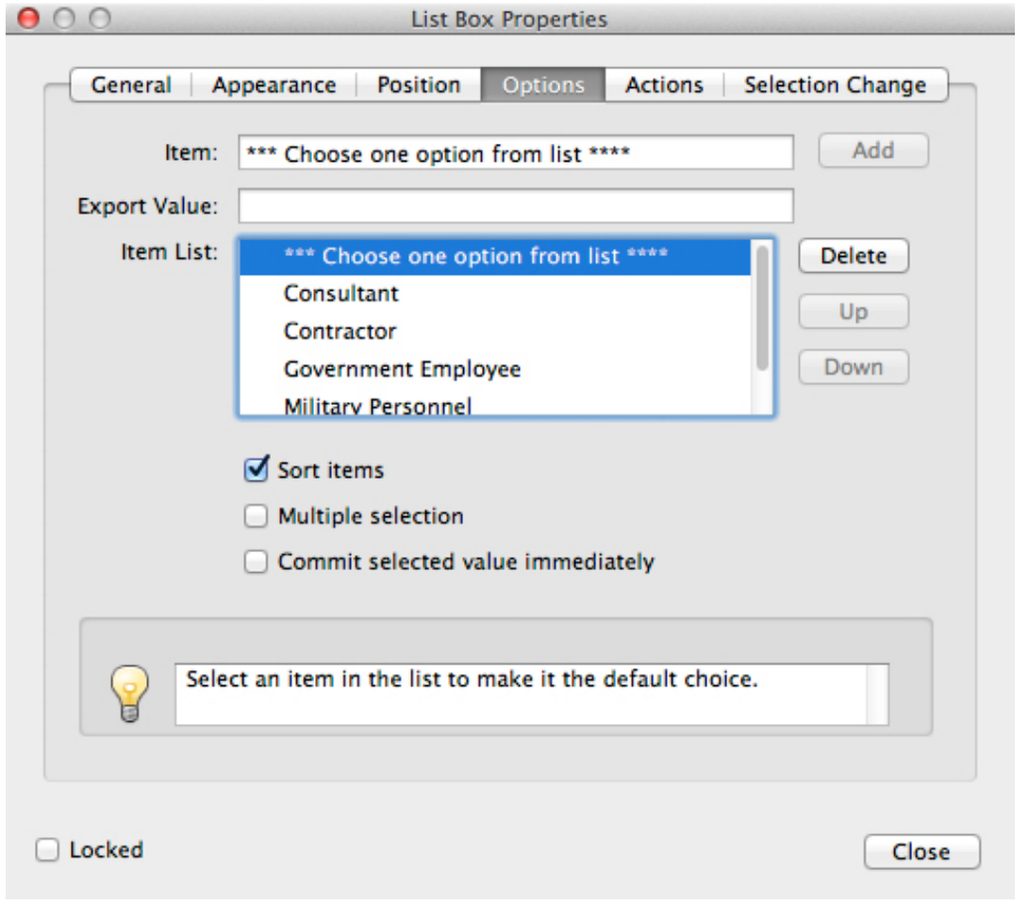
Figure 13.1 Add New Field Options



Open a context menu in Normal view on a field and choose the Add New Field command, then choose the field you want to add.

2. **Add the list items.**
   Name the field, set Appearance properties and click the Options tab in the List Box Properties window. To add items in the list, type the item name in the Item field. If you want a value such as a numeric value when the data are exported from your form, you can type a value in the Export Value text box. If you don't add an export value, the Item name in the list is exported. After adding an Item, click the Add button. Your new item is added to the list appearing at the bottom of the Options tab. Continue adding additional items to complete your list.

Figure 13.2 List Box Properties Options Tab



Type an item for the list in the Item text box and click the Add button.

3. **Managing the list.**
   If your list is best viewed in an alphabetic order, you can click Sort Items in the Options tab. If you want to manually move items in the list you can click an item and click either the Up or Down buttons to reorder the list.

4. **Enable/disable multiple selections.**
   Ask yourself, do you want the user to make multiple responses or limit the response to a single choice? If you have a question such as personal interests, the user may have several personal interests and want to make multiple selections. If you ask a user his/her country of birth, you want only a single country response on the form.

5. **Select the default.**
   Before you exit the List Box Properties, be certain to click the item in the list you want appearing as the default. You may wish to add a line of dashes or asterisks for the first line or text such as *Choose one…* Move the item to the top of the list and select it before exiting the List Box Properties window. When the form is completed, the list box displays the item you chose for the default.

Creating Dropdown Menus

Dropdowns follow many of the same rules you apply to list boxes. After you create a dropdown field, you supply the list information in the Dropdown Properties window in the Options tab. You type the name of the item, click the Add button and the item is listed at the bottom of the Options tab. You can sort and manage the list in the same manner as you do with List Boxes.

The one item you can choose with dropdowns that doesn't appear for List Boxes is *Allow user to enter custom text*. If you check this box the user is not restricted to the list items. However, Dropdown menus are limited to single selections unlike List Boxes where users can select multiple items in a list.

Before you exit the Dropdown Properties window you need to select the item you want to appear as the default. Dropdowns most often display a single item in the field box with an arrowhead appearing to the right of the default selection. Be certain the item appearing in the field box is the one you want displayed on your form before the user interacts with it.

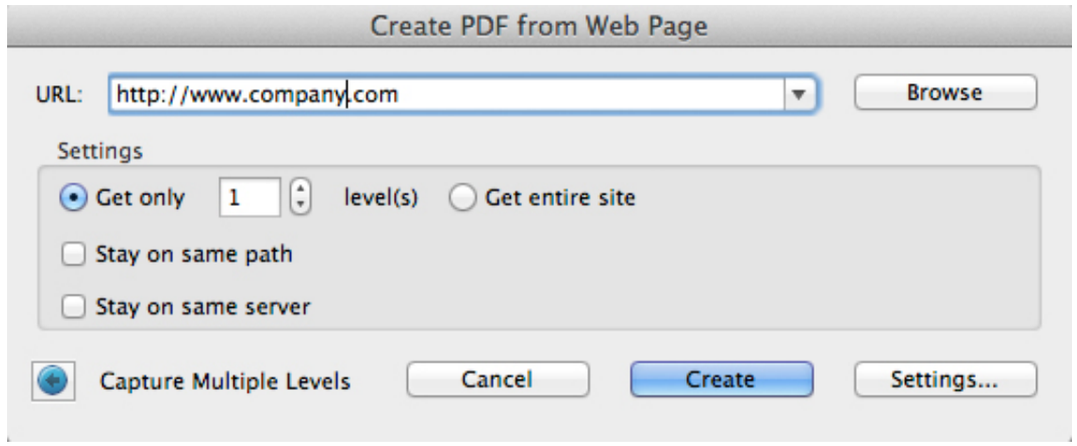
Dropdown menus are particularly helpful when you need to identify states, regions, countries, calling codes, and similar kinds of lists. One disadvantage you have with Acrobat is that you need to type each item in a dropdown list. Acrobat provides no means for importing text or MS Excel files to populate lists for either list boxes or dropdowns.

You do have one workaround however when you want to use lists commonly found on websites such as country lists. When you arrive at a website that contains a dropdown menu displaying

countries, regions, states, etc., you can convert the HTML file to a PDF and capture form fields.

On Windows you can use the Create PDF tool in the MS Explorer toolbar. On the Macintosh you need to use the Create PDF From Web Page tool (File ➤ Create PDF ➤ From Web Page. When the PDF is opened in Acrobat click the field you want to copy and press CTRL/Command + C. Open your form and press CTRL/Command + V to paste the field. Dropdown menus carry field attributes when you copy from an HTML page converted to PDF and paste in your form.

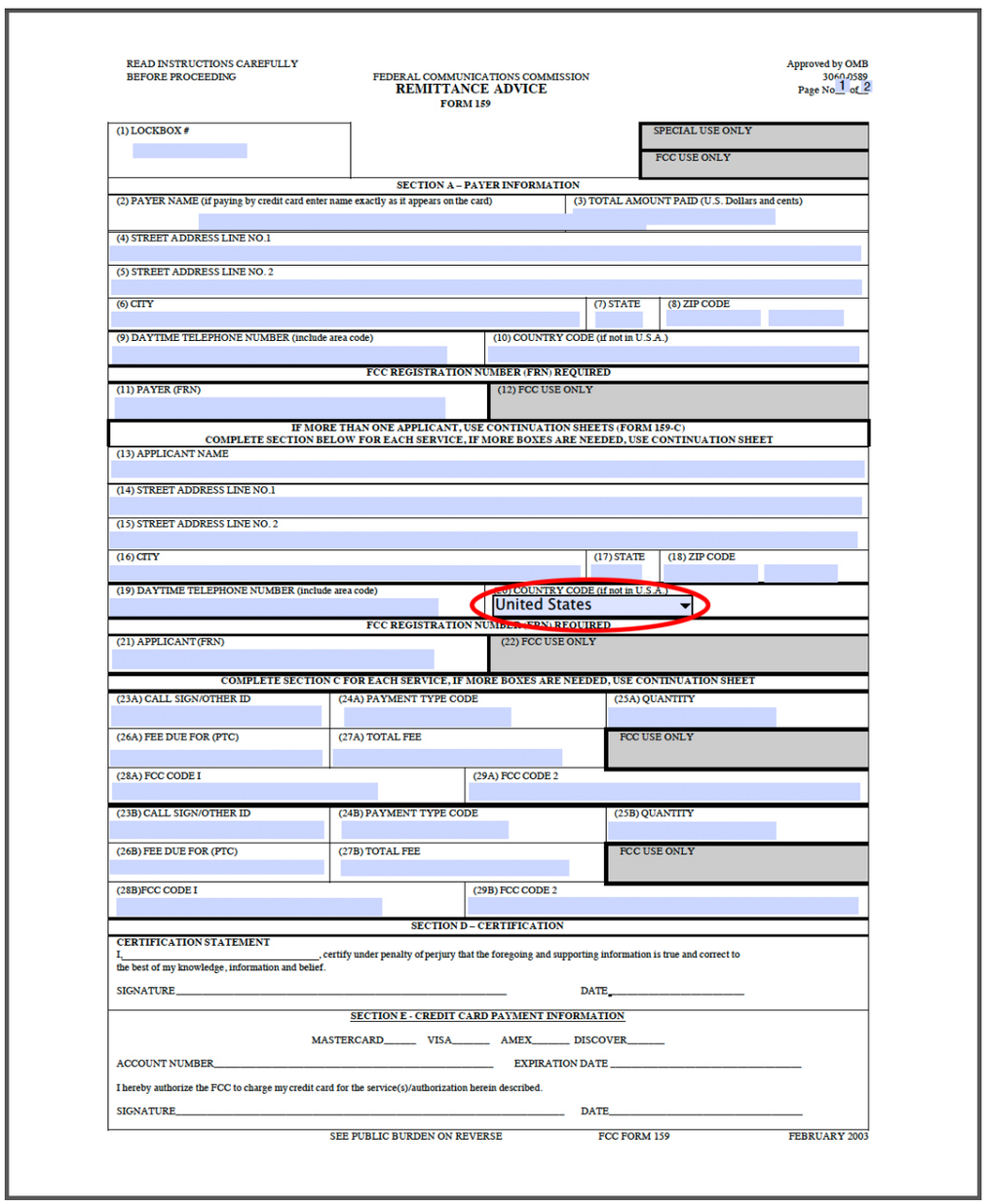
Figure 13.3 Create PDF From Web Page



Capture a web page that has form fields for countries, states, or other dropdown menus. When converted to PDF you can copy/paste fields.

The USA Office of Management and Budget uses the example form shown here. A field appears on the form for the user to identify a country calling code. Many government forms need to address some data for an international market. Departments of state, embassies, diplomatic services, etc. require using international country and calling codes on forms. The field for identifying a country calling code is a good candidate for a dropdown menu.

Figure 13.4 OMB Sample Form



READ INSTRUCTIONS CAREFULLY BEFORE PROCEEDING

FEDERAL COMMUNICATIONS COMMISSION
REMITTANCE ADVICE
FORM 159

Approved by OMB
3060-0589
Page No. 1 of 2

(1) LOCKBOX #

SPECIAL USE ONLY
FCC USE ONLY

SECTION A – PAYER INFORMATION

(2) PAYER NAME (if paying by credit card enter name exactly as it appears on the card)
(3) TOTAL AMOUNT PAID (U.S. Dollars and cents)
(4) STREET ADDRESS LINE NO.1
(5) STREET ADDRESS LINE NO. 2
(6) CITY
(7) STATE
(8) ZIP CODE
(9) DAYTIME TELEPHONE NUMBER (include area code)
(10) COUNTRY CODE (if not in U.S.A.)

FCC REGISTRATION NUMBER (FRN) REQUIRED

(11) PAYER (FRN)
(12) FCC USE ONLY

IF MORE THAN ONE APPLICANT, USE CONTINUATION SHEETS (FORM 159-C)
COMPLETE SECTION BELOW FOR EACH SERVICE, IF MORE BOXES ARE NEEDED, USE CONTINUATION SHEET

(13) APPLICANT NAME
(14) STREET ADDRESS LINE NO.1
(15) STREET ADDRESS LINE NO. 2
(16) CITY
(17) STATE
(18) ZIP CODE
(19) DAYTIME TELEPHONE NUMBER (include area code)
(20) COUNTRY CODE (if not in U.S.A.)
United States

FCC REGISTRATION NUMBER (FRN) REQUIRED

(21) APPLICANT (FRN)
(22) FCC USE ONLY

COMPLETE SECTION C FOR EACH SERVICE, IF MORE BOXES ARE NEEDED, USE CONTINUATION SHEET

(23A) CALL SIGN/OTHER ID
(24A) PAYMENT TYPE CODE
(25A) QUANTITY
(26A) FEE DUE FOR (PTC)
(27A) TOTAL FEE
FCC USE ONLY
(28A) FCC CODE 1
(29A) FCC CODE 2
(23B) CALL SIGN/OTHER ID
(24B) PAYMENT TYPE CODE
(25B) QUANTITY
(26B) FEE DUE FOR (PTC)
(27B) TOTAL FEE
FCC USE ONLY
(28B)FCC CODE 1
(29B) FCC CODE 2

SECTION D – CERTIFICATION

CERTIFICATION STATEMENT

I,_____, certify under penalty of perjury that the foregoing and supporting information is true and correct to the best of my knowledge, information and belief.

SIGNATURE_____ DATE_____

SECTION E - CREDIT CARD PAYMENT INFORMATION

MASTERCARD_____ VISA_____ AMEX_____ DISCOVER_____

ACCOUNT NUMBER_____ EXPIRATION DATE _____

I hereby authorize the FCC to charge my credit card for the service(s)/authorization herein described.

SIGNATURE_____ DATE_____

SEE PUBLIC BURDEN ON REVERSE FCC FORM 159 FEBRUARY 2003

A field exists for calling codes

To create a dropdown menu for calling codes, do the following:

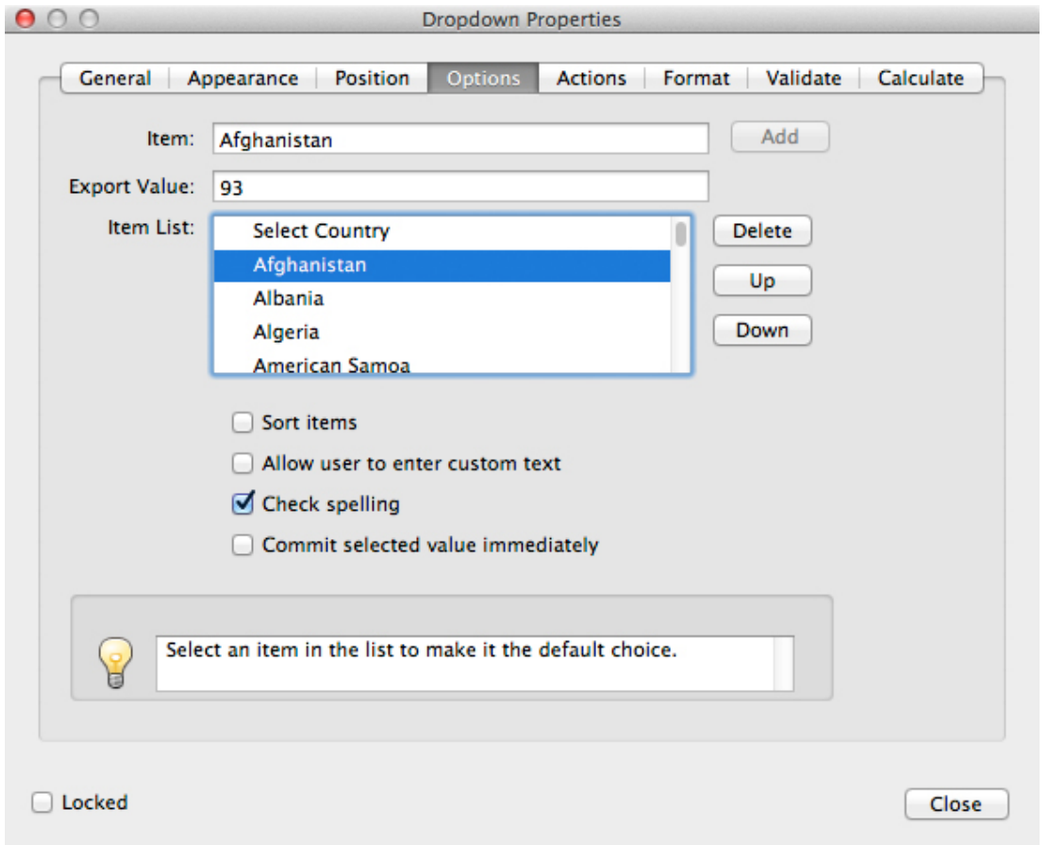
1. **Search the Web for a form containing a dropdown menu listing calling codes.**
   A webpage may include a dropdown menu listing calling codes however, I found this task difficult. I couldn't find a menu with the data I needed for a dropdown field on the example form. As an alternative, I found dropdown menus listing country names and decided to modify the Options.

2. **Copy a dropdown field from a webpage file converted to PDF.**
   In this example I first converted a webpage to PDF using the File ➤ Create PDF ➤ From Web Page) command. I copied a dropdown field containing country names and pasted the field in the example form.

3. **Open the Dropdown Properties window and click the Options tab.**
   Since countries were listed in the Dropdown Properties Options tab when I pasted the field, I decided to use the country names and change the export values. Each country is listed and the export value is the country calling code. I made edits by clicking a country name and typed the corresponding code in the Export Value text box.

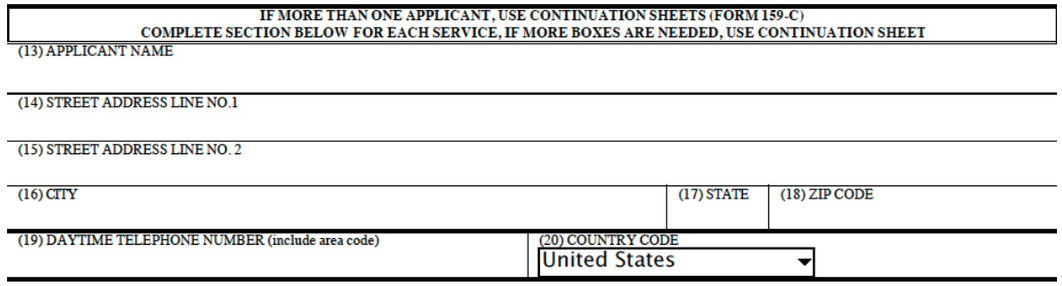
Using this method makes it easy for the user. Users simply choose the country from the menu rather than searching for codes matching country names. When the data are exported the code is exported to the data file.

Figure 13.5 Dropdown Menu Listing Countries



Click a country and add an Export Value. In this example I added calling code numbers for the Export Values.

Figure 13.6 Dropdown List on Form



The selected item in the Dropdown Options Properties appears as the default selection in the menu.

4. **Select the default.**

   After editing fields in the Options tab, be certain to select the item in the list that you want to appear as the default. In this case I selected United States. Choose the item that you believe most users are likely to select when filling out the form. Click OK and the selection becomes the new default.

List Boxes and Dropdowns can be helpful when you need room on a form for long response lists. These fields are also helpful for users when you provide closed response answers to questions. The users won’t need to struggle to type an answer in an open field.

Chapter 14

Securing Forms & Copyright Protection

Adding security to forms to prevent users from changing the form. Should you copyright your forms?

Chapter 14

IN THIS CHAPTER

Securing a form is protecting the form against changing its appearance and content. For eForms, all forms should be secured without exception. Even the most basic forms requiring no more than a signature should be secured and protected against changing the form contents. Any content displayed on government websites and available for public consumption should be considered a target for security and protection. This rule should extend to every document a government website hosts even those documents not considered forms.

Securing a PDF Document

Acrobat provides several opportunities to secure PDF files. You can use Acrobat Security, you can use a 3rd party solution for securing files, you can use in-house security options developed by your IT department, and you can certify PDF files.

The simple and easy method for applying security is handled by Acrobat Security. For the purposes of this book I'll address Acrobat Security but I encourage all readers to search for solutions acceptable for your branch of government. There may be mechanisms in place now in your agency for securing documents including PDFs. You should spend some time researching security options available to your department.

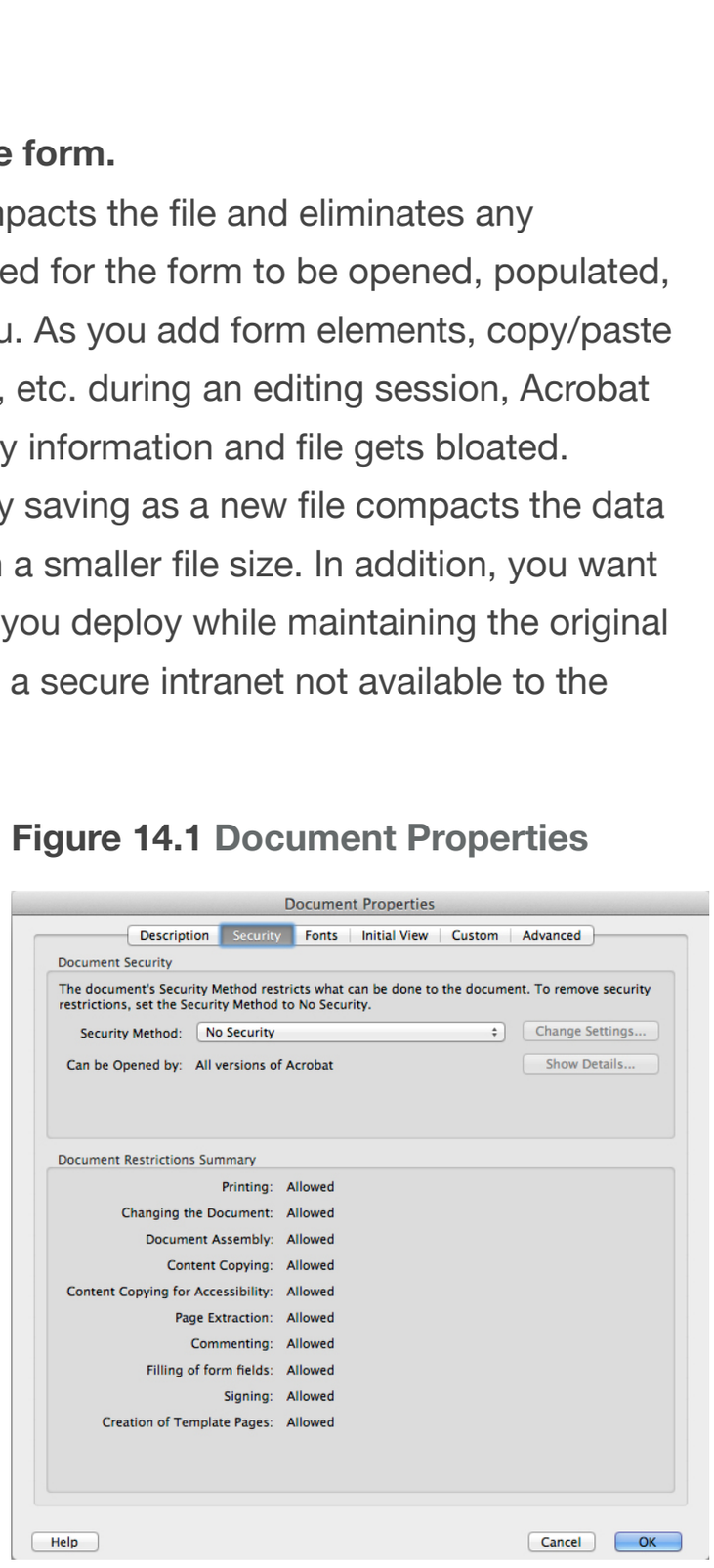
Applying security to a PDF is handled in the Document Properties. You can secure files protecting certain activities while granting permissions for a number of other activities. These choices are found in the Acrobat Security window.

To secure a PDF form, do the following:

1. **Open the form in Acrobat.**
   Securing a document is the last step you perform before deploying a form. You should test the form, be certain the form is assembled properly and ready to deploy.
2. **Save a copy of the form.**
   Saving a copy compacts the file and eliminates any overhead not needed for the form to be opened, populated, and returned to you. As you add form elements, copy/paste fields, delete fields, etc. during an editing session, Acrobat retains unnecessary information and file gets bloated. Rewriting the file by saving as a new file compacts the data and often results in a smaller file size. In addition, you want to secure the form you deploy while maintaining the original unsecured form on a secure intranet not available to the public.
3. **Open the copy you save and open the Document Properties.**

   Press CTRL/ Command + D or choose File ➤ Properties to open the Document Properties window.

Figure 14.1 Document Properties



Click the Security tab in the Document Properties window.

4. **Choose a Security Method.**
   Click the Security tab. From the Security Method drop-down menu you see several options for securing the document. For the purposes of this discussion we look at Password Security. Unless you have another security method in place, this security option will work well for the eForms you deploy except documents related to national security.

   From the menu options choose Password Security. The Password Security window opens.

5. **Adding Restrictions.**
   Be certain to not check the box for Require a password to open the document. Checking this box requires the constituent to know a password in order to open the PDF in Adobe Reader. The eForms you host won't require people knowing a password to open the forms.

Check the box for Restrict editing and printing of the document. A password will be required in order to change these permission settings. When you check the box, options you find in the Password Security – Settings window are made available to you.

6. **Disallowing Printing.**
   The first item to address is whether you want the end user to be able to print the form. This may be a matter that needs some discussion in committee. You can force constituents to follow your lead in achieving sustainable measures by disallowing printing. Or you can provide users the ability to print files for their own personal filing system. If you do restrict printing, you should create a webpage that explains the fact that no forms can be printed, they must be handled electronically, and your reasoning for not permitting printing.

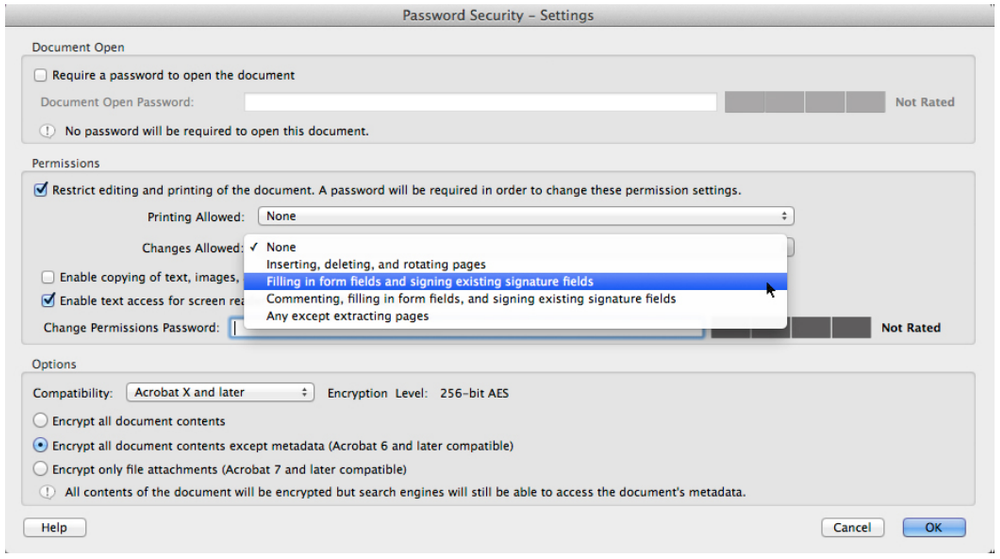
7. **Enable Form Fill-In.**
   From the Changes Allowed drop-down menu choose *Filling in form fields and signing existing signature fields*. You must enable this permission or users won't be able to fill in the form.

8. **Enable text access.**
   By default the item for *Enable text access for screen reader devices for the visually impaired* is checked. You should leave this item enabled so your forms meet with accessibility standards and permit users with screen readers the ability to complete your forms.

9. **Change Permissions Password.**

   This is where you add password security to your form. On the right side of the text box is a scale that measures the strength of your password. Typing a few characters results in a weak password. Using upper and lowercase characters, numbers, and certain symbols results in a password with more strength.

Figure 14.2 Password Security Settings



Be certain to enable form fill in and signing existing signature fields.

10. **Compatibility.**

    From the menu choose Acrobat X compatibility. This encryption level is much stronger than earlier versions of Acrobat compatibility.

    Check the box for *Encrypt all document contents except metadata (Acrobat 6 and later compatible)*. This box should be checked so your forms can be found using Internet search engines.

11. **Click OK.**

    When you click OK you are prompted to re-enter your password. Type the password exactly the same as the password you supplied in the Change Permissions Password text box.

12. **Save the file.**

    The security settings are not added to the document until you save the file.

Cataloging Passwords

One of the best ways to keep a record of passwords for PDF files in your department is to create a PDF form where new passwords are stored. You can encrypt the file with password security for opening the document. Once open, you can perform a search with Acrobat Search to find document form numbers and passwords associated with them. Keep this file on a secure intranet server where only the forms design departments have access.

Storing passwords in an encrypted file requires users to know only one password to open the PDF. When a user opens the file, the user has access to all passwords used on the forms you list in the file.

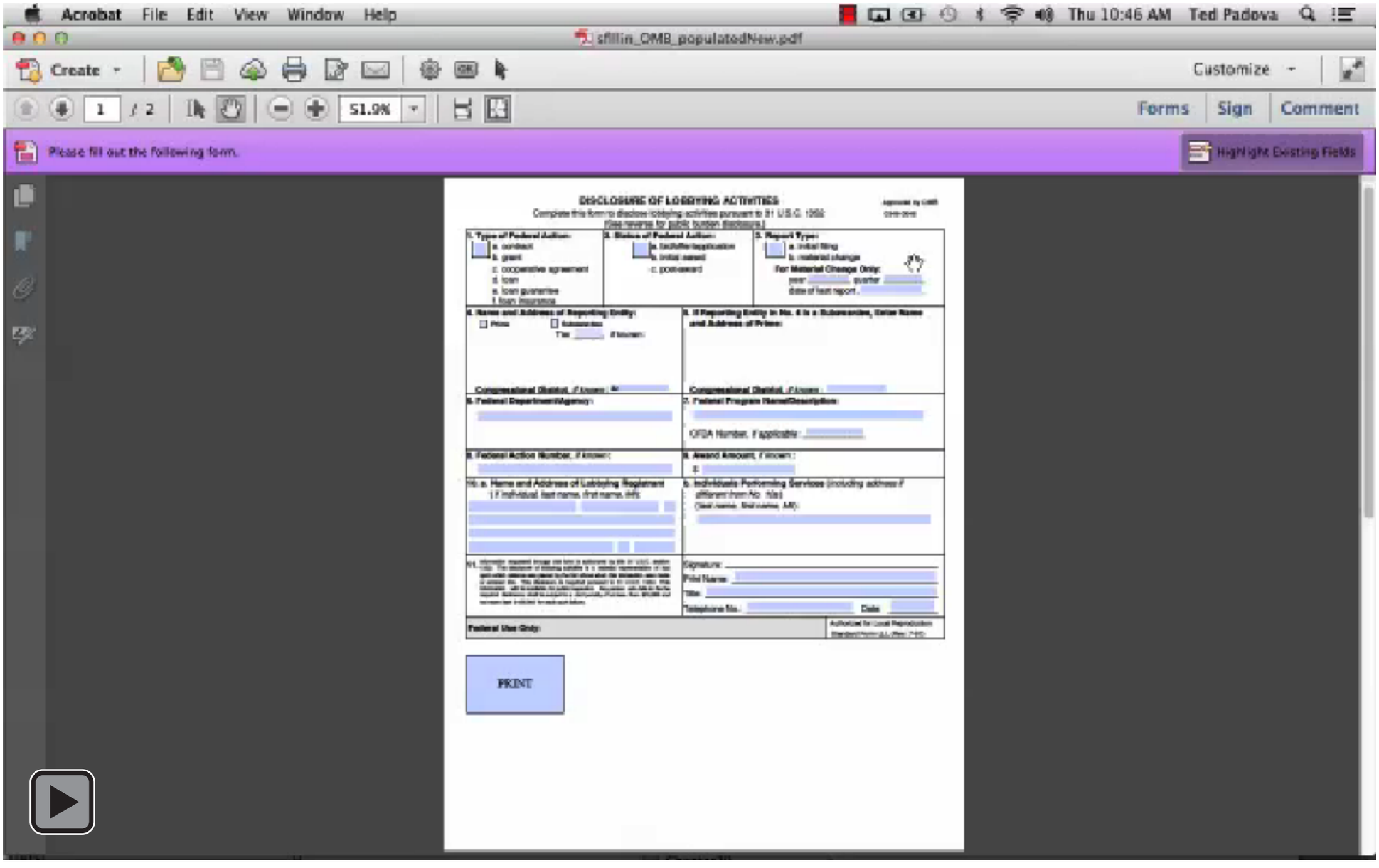
Copyrighting Forms

Almost all forms on USA Federal government websites are considered to be in the public domain and therefore do not carry copyright protection unless specifically noted on the form. US States vary according to policies established by the states. Many US State forms are copyrighted and cannot be reproduced or displayed in any content whether printed or online.

This is a matter for your office to decide. Unless you have specific reasons for doing so I would recommend you not copyright forms —especially if you're proud of the forms hosted on your site. Your web content may be assessed and evaluated by magazines, reporting agencies, and studies that examine developments in use of eForms. Independent researchers can benefit by showing the community of government offices examples of well-designed and functional eForms.

If your forms are scanned documents, have no fields, no department seals, and designed poorly, by all means, copyright your forms. You don't want samples of your forms appearing in a study by ComputerWorld!

Movie


Acrobat File Edit View Window Help
Thu 10:46 AM Ted Padova
sflllin_OMB_populatedNew.pdf
Create
Customize
1 / 2
51.9%
Forms Sign Comment
Please fill out the following form.
Highlight Existing Fields
DISCLOSURE OF LOBBYING ACTIVITIES
Complete this form to disclose lobbying activities pursuant to 31 U.S.C. 1352
(See reverse for public burden disclosure.)
1. *Type of Federal Action:
a. contract
b. grant
c. cooperative agreement
d. loan
e. loan guarantee
f. loan insurance
2. Status of Federal Action:
a. bid/offer/application
b. initial award
c. post-award
3. Report Type:
a. initial filing
b. material change
For Material Change Only:
year quarter
date of last report
4. Name and Address of Reporting Entity:
Prime
Subawardee
Tier , if known:
Congressional District, if known:
5. If Reporting Entity in No. 4 is a Subawardee, Enter Name and Address of Prime:
Congressional District, if known:
6. Federal Department/Agency:
7. Federal Program Name/Description:
CFDA Number, if applicable:
8. Federal Action Number, if known:
9. Award Amount, if known:
$
10. a. Name and Address of Lobbying Registrant
(if individual, last name, first name, MI):
b. Individuals Performing Services (including address if different from No. 10a)
(last name, first name, MI):
11.
Signature:
Print Name:
Title:
Telephone No.:
Date:
Federal Use Only:
Authorized for Local Reproduction
PRINT

Chapter 15

Signing and Notarizing Forms

Adding signatures to forms and working with eNotaries

Chapter 15

Perhaps the largest obstacles in preventing eForms from proliferating throughout government offices are related to signing and notarizing forms. In the USA the Digital Signature Act of 1999, by law, granted agencies the right to recognize digital signatures as a means of authenticating documents much like they do with written signatures. In the UK, electronic signatures regulations were established in 2002.

IN THIS CHAPTER

In more than a decade of legal acts permitting agencies rights for using and accepting digital signatures for legal and binding documents, little progress has been made and this still remains an area of caution by many agencies.

However some progress is slowly making its way with eSignatures. The US Federal Housing Administration (FHA), for example, in 2010 began accepting eSignatures from third party documents including real estate forms. Progress has been slow though some movement is occurring. As more government agencies begin using eSignatures we should see much wider acceptance.

Another area that has kept eForms from proliferating is electronic notarization. There are eNotaries in existence today throughout the USA. However they are not present in some US states. Of the 50 US states, 29 have eNotaries and 21 states do not have notaries that can electronically notarize a document.

Like digital signatures, eNotarization has been slow but we are seeing progress.

The important thing to consider is that digital signatures and electronically signing documents as well as electronic notarization are both available in more than half the US States and moving to acceptance in other countries. Services do exist and those government websites that do host eForms can remarkably improve growth with supporting electronic signing and notarizing documents. The more government supports eSignatures and eNotarizations, the more users will take advantage and the more eNotary services will be made available.

Signing PDF Forms

As a PDF author, you need not be as concerned with adding digital signature fields on forms, but more interested in supporting signatures from constituents. There are two types of signatures that users most often apply. These are adding a digital signature and adding a wet signature.

The lion's share of Adobe Reader users are not familiar with creating a digital signature. A digital signature must first be created in Reader. After a user creates a digital signature, the user can digitally sign a form containing a digital signature field.

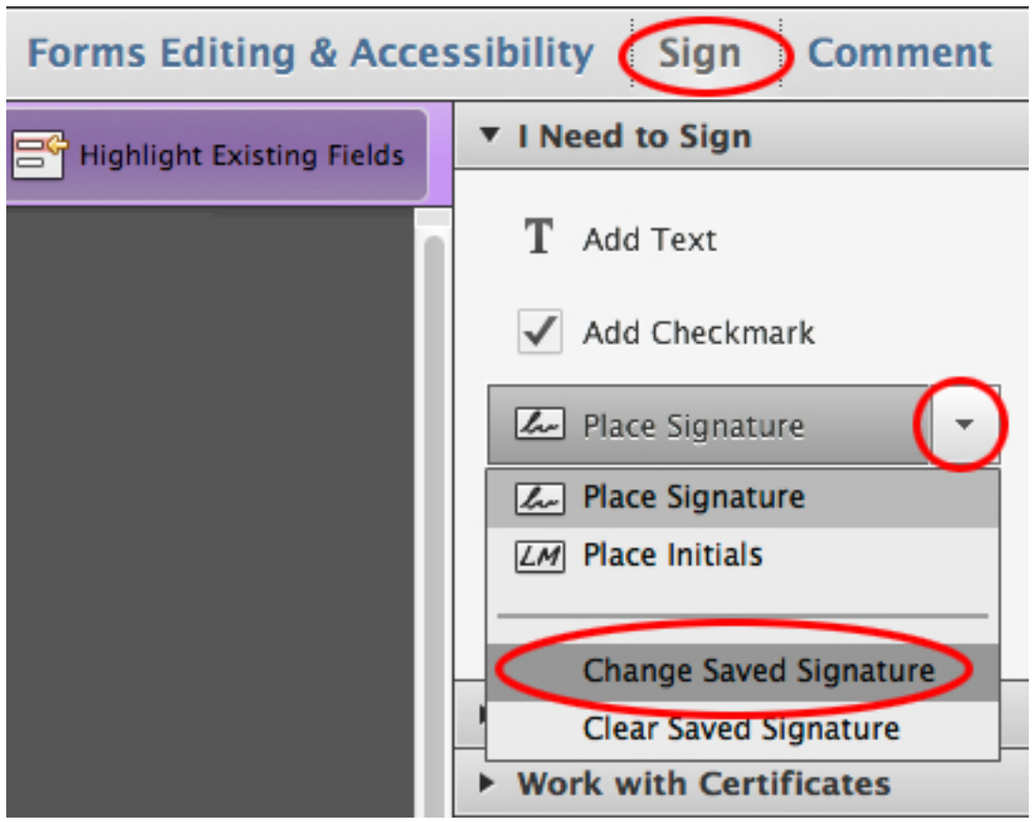
In controlled environments such as within your office, digital signatures can be beneficial. However for the masses, not only is digital signature creation often misunderstood, validating signatures requires obtaining public keys from the users. Even if the end users understood how to add a signature and include a public key, the job is enormous when it comes to validating signatures.

For a more practical application users can place a wet signature. The user simply uses a mouse, stylus, or finger (on a mobile or tablet device) and signs a document much like they would handwrite a signature on a piece of paper. Wet signatures also

include creating a signature by typing the user's name. Both methods are handled in the Sign panel ➤ Place Signature menu in Adobe Reader.

Placing a signature requires the user to open the Sign panel. From the Place Signature drop-down menu the user needs to first click Change Saved Signature. The Place Signature dialog box opens. Here you find a drop-down menu for Typing a Signature, Drawing a Signature, or Using an Image.

Figure 15.1 Sign Panel Options

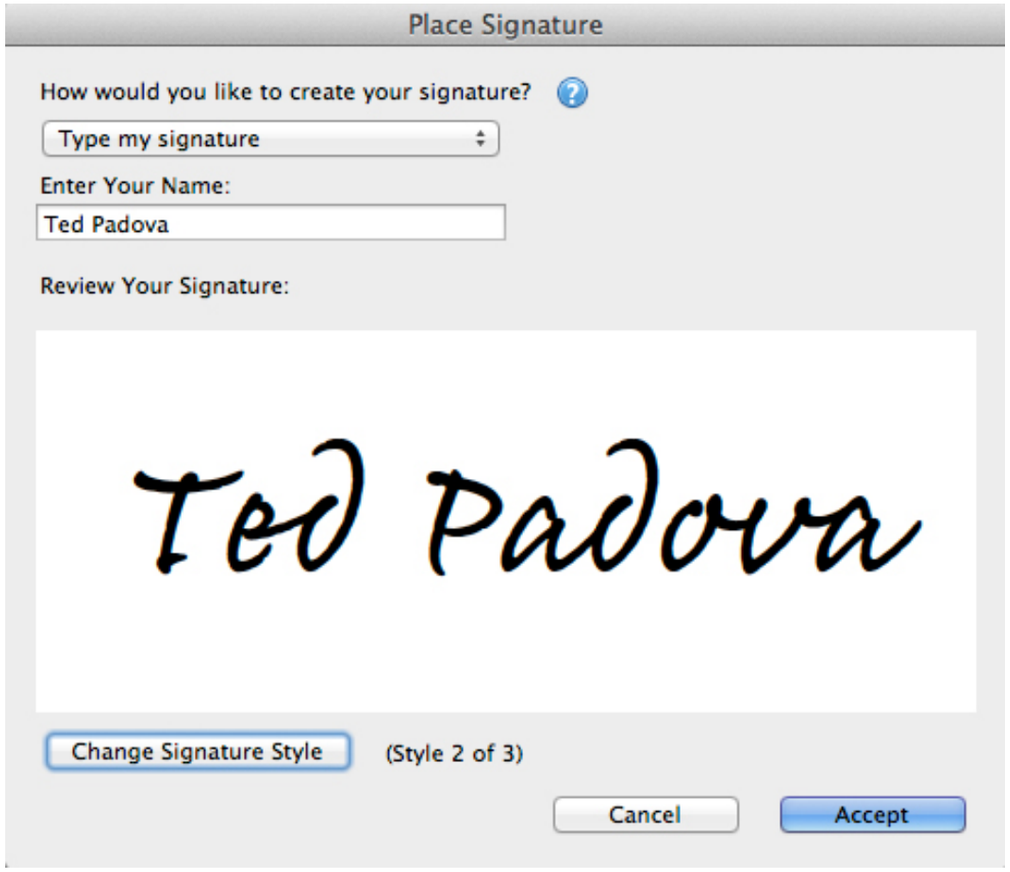


Click the Sign panel and from the Place Signature drop-down menu choose Change Saved Signature.

Typing a Signature.

This option requires the user to type their name in a text field box in the *Place Signature* dialog box. After typing a name, the user can change the text appearance by clicking the *Change Signature Style* button. Continue clicking and Reader displays different fonts and styles in the Review Your Signature window. When the style looks acceptable, the user clicks the Accept button.

Figure 15.2 Typed Signature



Choose Type My Signature from the drop-down menu and type your name in the Enter Your Name text box.

Draw my Signature.

This option permits the user to write a signature much like handwriting on paper. The signature may appear unpolished since the user needs to work with a mouse or finger. If a user has a stylus, the signature can look very close to a handwritten signature.

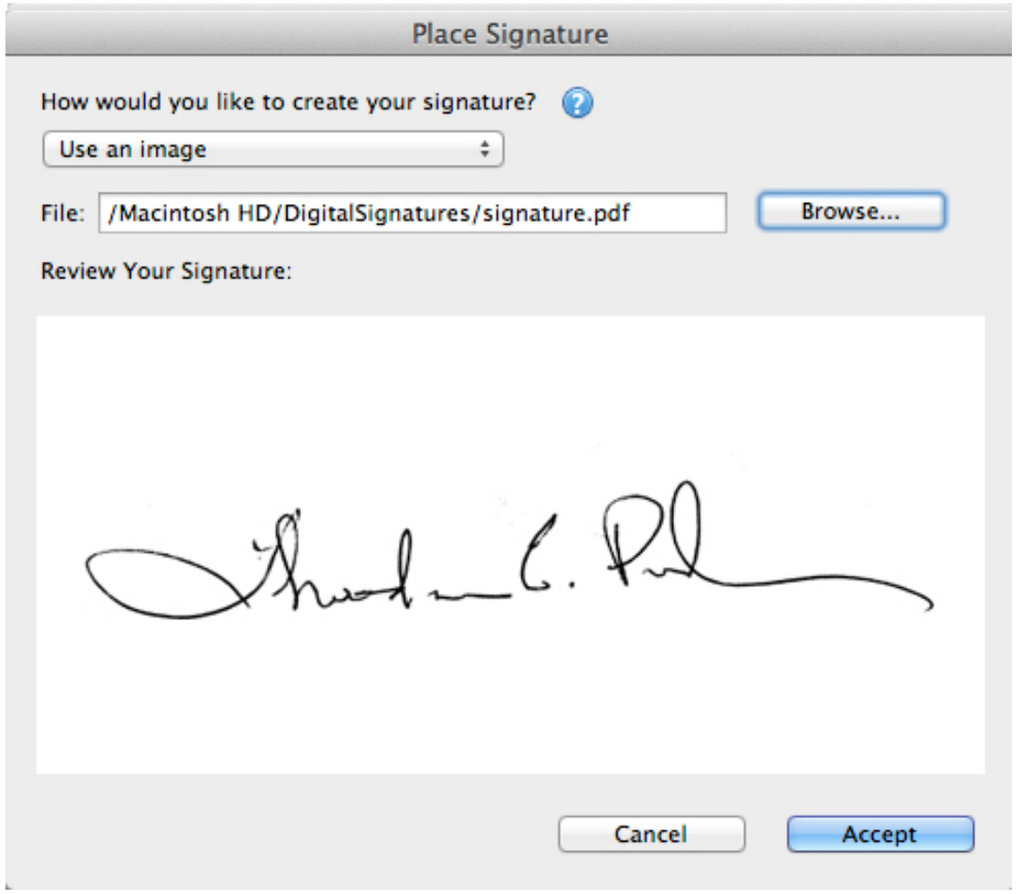
Use an Image.

A handwritten signature can be scanned and saved as JPG, PNG, GIF, BMP, TIFF, or PDF. When the user chooses Use an Image from the drop-down menu they click the Browse button to locate the scanned signature on their hard drive. A preview is displayed in the Review Your Signature window. If the preview looks right, click the Accept button.

Once a Reader user sets attributes for a signature, the same settings are used when filling in additional forms and working in different Reader sessions. No further changes are needed until one wishes to change his/her signature appearance.

Creating a signature is relatively simple, but very clear steps should be posted on a website for users to follow and the user community should be informed what signature is acceptable by your agency. This entire area of electronic signing documents should be discussed in committee and decisions should be made relative to acceptable standards.

Figure 15.3 Place an Image



Choose Use an Image from the drop-down menu and click the browse button to locate a scan of your signature.

eNotarizing PDF Forms

Notarization of all forms is not necessary but there are a significant number of forms within many agencies that do require notarization. For this reason any government office requiring notarization of documents should host information about eNotary services within an agency's catchment area.

PDF authors don't need to be concerned with adding content to a form that permits eNotarization. The job at hand is research and clearly stating processes on agency websites for constituents to follow for gaining access to eNotaries. For eForms requiring notarization you might add a URL link to a webpage detailing information regarding eNotaries in your area to make it easier for constituents to find eNotary services.

Your agency would be well advantaged in locating existing eNotaries, promoting expansion of eNotary services, and meeting with groups such as GlobalSign, American Society of Notaries (USA), The National Notary Association (USA), The Notaries Society org in the UK, and various other notary societies that exist in many countries.

Chapter 16

Working with Calculations

Knowing how to use Acrobat's calculations options

Chapter 16

IN THIS CHAPTER

When calculations are needed on a form, it's always best to let Adobe Reader perform the calculations rather than the end users. If you develop correct calculation formulas, you can prevent user error and save both the constituent and your agency personnel time in processing a form.

There are three methods you can use to create calculations in PDF documents. These include 1) using *Acrobat's built-in formulas*, 2) using *Simplified Field Notation*, and 3) using *JavaScript*. The easiest method and one that even the most novice Acrobat user can apply is adding a built-in calculation formula provided by Acrobat. A little more difficult is using Simplified Field Notation. The most difficult for non-programmers is using JavaScript.

Knowing Where Calculations are Placed in PDFs

The Calculate tab in text field boxes is the only place where you can add any one of the three methods. JavaScript however, can be added to any PDF elements where you find Actions. You can perform calculations with JavaScript on any field type. In text fields you can add JavaScripts in the Actions, Format, Validate, and Calculate tabs.

When creating calculations in tables for summing data or writing other formulas, most of your work involves using the Calculate tab in text fields. Many eForms are designed with fields where calculations can be performed. For most of these forms you use a few simple math operations and don't need to worry about writing complex formulas.

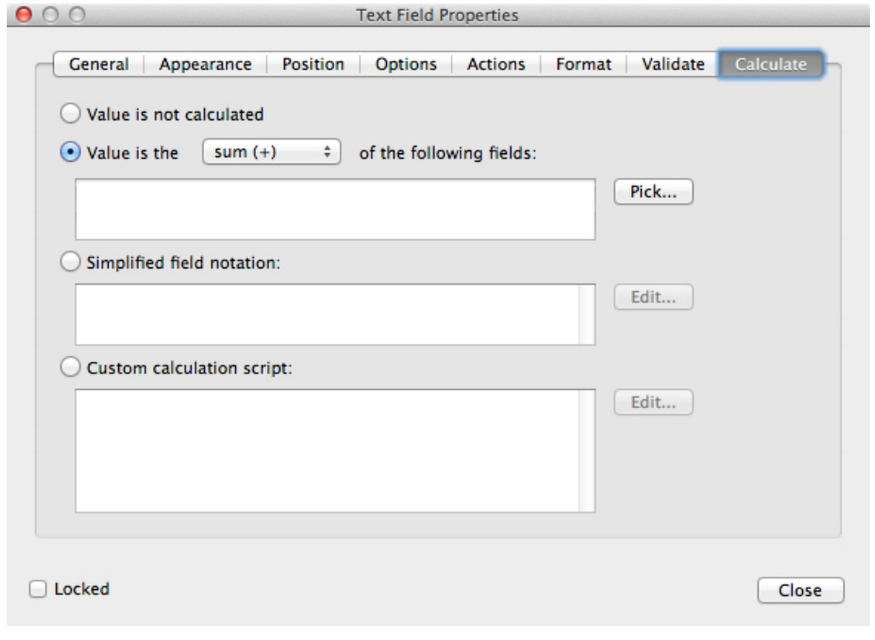
Calculating Data with Acrobat Built-in Formulas

Acrobat supports several formulas. You can sum contents of fields, you can calculate a product (multiply), calculate an average, calculate a minimum within a range of values, and calculate a maximum within a range of values. Acrobat's built-in formulas are limited to these five calculation methods. If you want to subtract or divide values you need to use either Simplified Field Notation or JavaScript.

Before you dive into the Calculate Properties on a text field, there are a few things you should know about field design. The *B6J* sample form from the US Department of Justice (see Figure 16.2) makes the work for adding a calculation formula difficult for the PDF author. The form is not well designed. Inasmuch as tool tips provide the user feedback on the data required for a given field, the alignment of text and fields is awkward. However more importantly, the field names in the column that needs tabulation are unique names without a proper hierarchical structure. Fields are named *Utilities.Fuel.Amt*, *Telephone.Amt*, *Transportation.Amt, Clothing.Amt*, etc.

When you open the Text Field Properties window and click the Calculate tab for the *AverageMonthlyExp.Amt* (the total field), you select *Value is the* and choose sum (+) from the drop-down menu.

Figure 16.1 The Calculate Text Field Properties



Click the Value is the radio button & choose sum(+)

Figure 16.2 Form BJ6 From US Department of Justice

B6J (Official Form 6J) (12/07)

In re Debtor.Name.1 , Debtor

Case No. Case.Num.1 (if known)

SCHEDULE J - CURRENT EXPENDITURES OF INDIVIDUAL DEBTOR(S)

Complete this schedule by estimating the average or projected monthly expenses of the debtor and the debtor's family at time case filed. Prorate any payments made bi-weekly, quarterly, semi-annually, or annually to show monthly rate. The average monthly expenses calculated on this form may differ from the deductions from income allowed on Form22A or 22C.

Debt Check this box if a joint petition is filed and debtor's spouse maintains a separate household. Complete a separate schedule of expenditures labeled "Spouse."

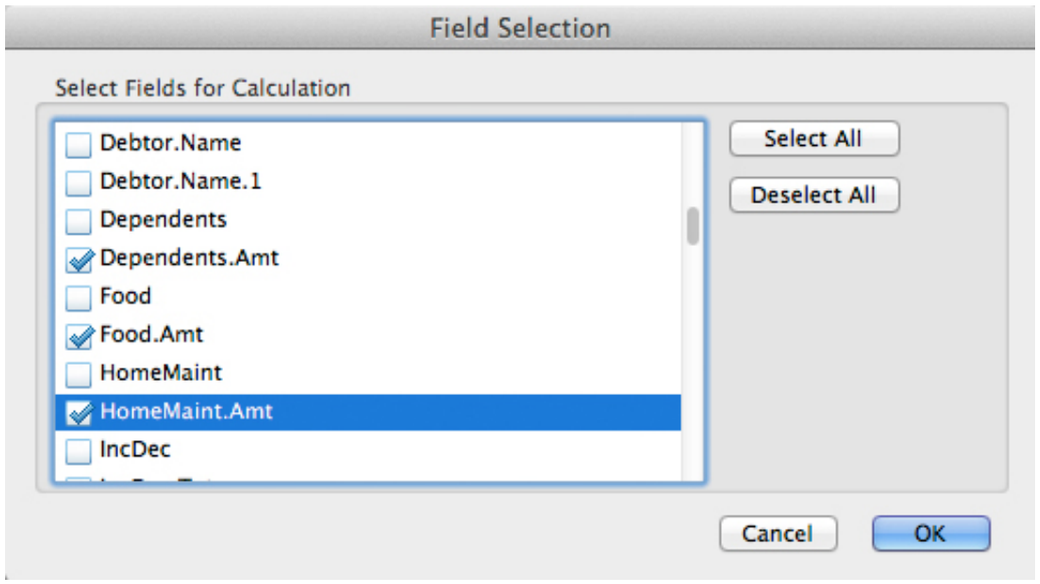
1. Rent or home mortgage payment (include lot rented for mobile home)	$ Mortgage.Amt
a. Are real estate taxes included? Yes Mortgage. No Mortgage.	
b. Is property insurance included? Yes Mortgage. No Mortgage.	
2. Utilities: a. Electricity and heating fuel	$ Utilities.Fuel.Amt
b. Water and sewer	$ Utilities.Water.Amt
c. Telephone	$ Utilities.Telephone.Amt
d. Other Utilities.Other.Txt	$ Utilities.Other.Amt
3. Home maintenance (repairs and upkeep)	$ HomeMaint.Amt
4. Food	$ Food.Amt
5. Clothing	$ Clothing.Amt
6. Laundry and dry cleaning	$ Laundry.Amt
7. Medical and dental expenses	$ Medical.Amt
8. Transportation (not including car payments)	$ Transportation.Amt
9. Recreation, clubs and entertainment, newspapers, magazines, etc.	$ Recreation.Amt
10.Charitable contributions	$ Charity.Amt
11.Insurance (not deducted from wages or included in home mortgage payments)	
a. Homeowner's or renter's	$ Insurance.Home.Amt
b. Life	$ Insurance.Life.Amt
c. Health	$ Insurance.Health.Amt
d. Auto	$ Insurance.Auto.Amt
e. Other Insurance.Other.Txt	$ Insurance.Other.Amt
12. Taxes (not deducted from wages or included in home mortgage payments) (Specify) Taxes.Txt	$ Tax.Amt
13. Installment payments: (In chapter 11, 12, and 13 cases, do not list payments to be included in the plan)	
a. Auto	$ InstallPymts.Auto.Amt
b. Other InstallPymts.Other1.Txt	$ InstallPymts.Other1.Amt
c. Other InstallPymts.Other2.Txt	$ InstallPymts.Other2.Amt
14. Alimony, maintenance, and support paid to others	$ Alimony.Amt
15. Payments for support of additional dependents not living at your home	$ Dependents.Amt
16. Regular expenses from operation of business, profession, or farm (attach detailed statement)	$ Business.Amt
17. Other Other.Txt	$ Other.Amt
18. AVERAGE MONTHLY EXPENSES (Total lines 1-17. Report also on Summary of Schedules and, if applicable, on the Statistical Summary of Certain Liabilities and Related Data.)	$ AvgMonthlyExp.Amt
19. Describe any increase or decrease in expenditures reasonably anticipated to occur within the year following the filing of this document: IncDec.Txt	
20. STATEMENT OF MONTHLY NET INCOME	
a. Average monthly income from Line 15 of Schedule I	$ Statement.Income
b. Average monthly expenses from Line 18 above	$ Statement.Expense
c. Monthly net income (a. minus b.)	$ Statement.Net

Button.Reset Button.SaveAs Button.Print

Fields are not properly named for easy calculations

You next click Pick and the Field Selection dialog box opens. In this dialog box you need to check every field in the column that you want to use in the formula.

Figure 16.3 Field Selection Dialog



You need to check all fields that are used in the formula

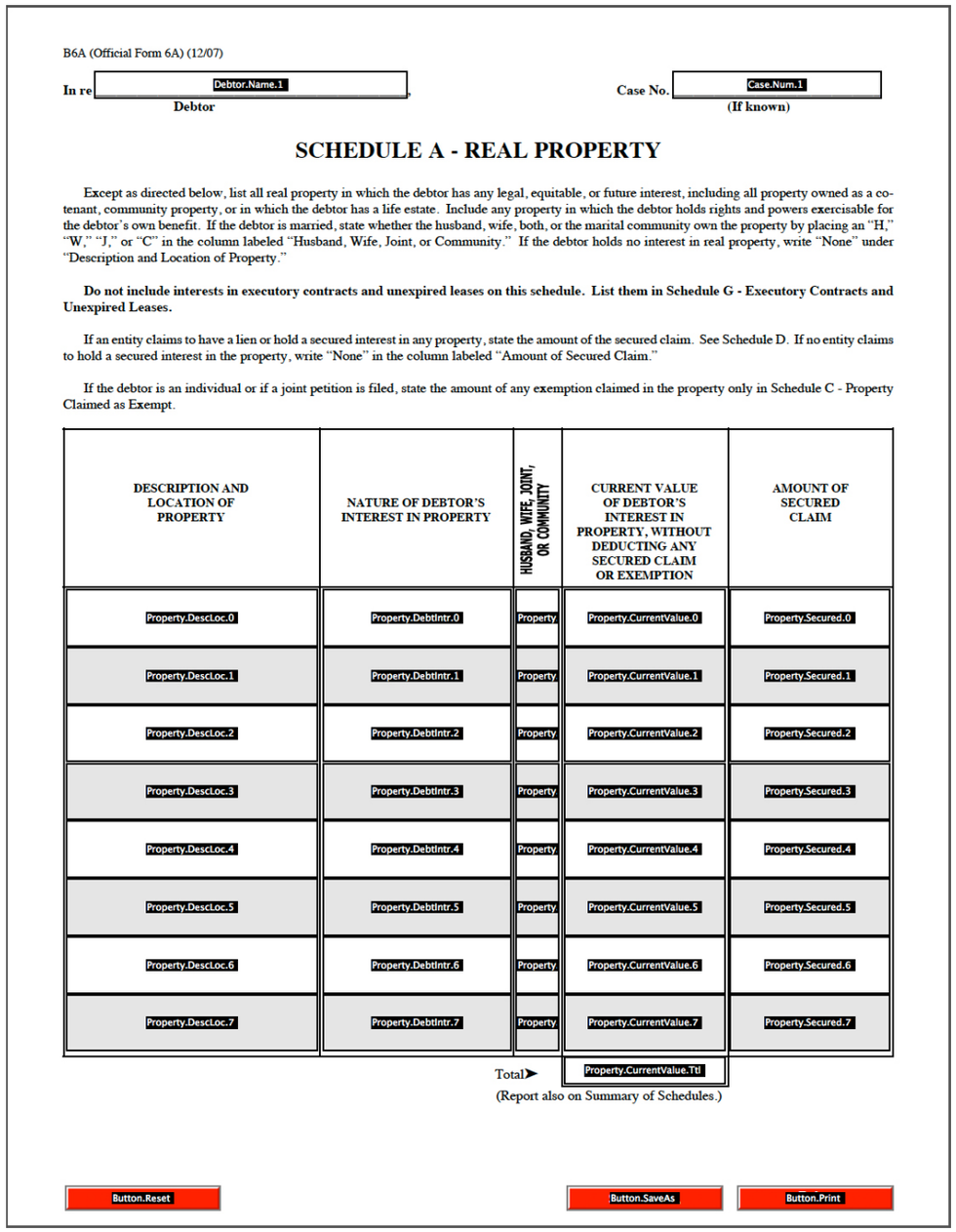
Since there are fields throughout the form you need to carefully read the field names and determine what fields are required for the calculation. On this particular form it would be easy to miss including a field in the formula and the calculation would render incorrect values.

After you select the fields, click OK and the formula is ready to perform the calculation.

On form B6A (also from the US Department of Justice) you find a table containing fields named in hierarchical order. However

EVERY field in the table has the same root name: *Property*. There is no way to distinguish the fields in the column that needs to be tabulated from the other fields in the table.

Figure 16.4 Form with a Column that Needs Summing



B6A (Official Form 6A) (12/07)

In re Debtor.Name.1, Debtor — Case No. Case.Num.1 (If known)

SCHEDULE A - REAL PROPERTY

Except as directed below, list all real property in which the debtor has any legal, equitable, or future interest, including all property owned as a co-tenant, community property, or in which the debtor has a life estate. Include any property in which the debtor holds rights and powers exercisable for the debtor's own benefit. If the debtor is married, state whether the husband, wife, both, or the marital community own the property by placing an "H," "W," "J," or "C" in the column labeled "Husband, Wife, Joint, or Community." If the debtor holds no interest in real property, write "None" under "Description and Location of Property."

Do not include interests in executory contracts and unexpired leases on this schedule. List them in Schedule G - Executory Contracts and Unexpired Leases.

If an entity claims to have a lien or hold a secured interest in any property, state the amount of the secured claim. See Schedule D. If no entity claims to hold a secured interest in the property, write "None" in the column labeled "Amount of Secured Claim."

If the debtor is an individual or if a joint petition is filed, state the amount of any exemption claimed in the property only in Schedule C - Property Claimed as Exempt.

DESCRIPTION AND LOCATION OF PROPERTY	NATURE OF DEBTOR'S INTEREST IN PROPERTY	HUSBAND, WIFE, JOINT, OR COMMUNITY	CURRENT VALUE OF DEBTOR'S INTEREST IN PROPERTY, WITHOUT DEDUCTING ANY SECURED CLAIM OR EXEMPTION	AMOUNT OF SECURED CLAIM
Property.DescLoc.0	Property.DebtIntr.0	Property	Property.CurrentValue.0	Property.Secured.0
Property.DescLoc.1	Property.DebtIntr.1	Property	Property.CurrentValue.1	Property.Secured.1
Property.DescLoc.2	Property.DebtIntr.2	Property	Property.CurrentValue.2	Property.Secured.2
Property.DescLoc.3	Property.DebtIntr.3	Property	Property.CurrentValue.3	Property.Secured.3
Property.DescLoc.4	Property.DebtIntr.4	Property	Property.CurrentValue.4	Property.Secured.4
Property.DescLoc.5	Property.DebtIntr.5	Property	Property.CurrentValue.5	Property.Secured.5
Property.DescLoc.6	Property.DebtIntr.6	Property	Property.CurrentValue.6	Property.Secured.6
Property.DescLoc.7	Property.DebtIntr.7	Property	Property.CurrentValue.7	Property.Secured.7
			Property.CurrentValue.Ttl	

Total➤
(Report also on Summary of Schedules.)

Button.Reset Button.SaveAs Button.Print

Fields on this form all have the same root name.

Let's fix this form to demonstrate an example for how the form should have been designed:

1. **Create a text field**.

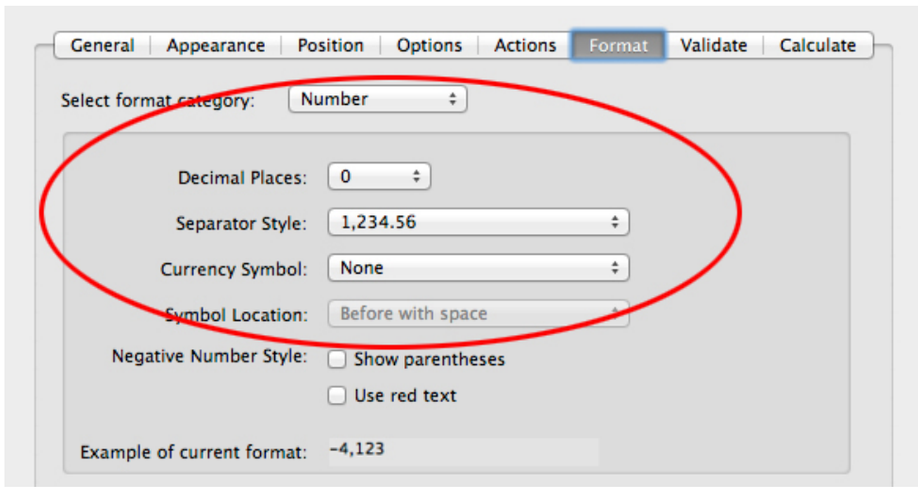
   In this sample form I deleted the fields in the 4th column where you see *Current Value…* and started by creating a new field in the first row.

   For tables having columns and rows you create the first field without a hierarchical name. In this example I use *currentValue* as the field name. Since all other fields in the table have a root name of Property, my new column has a unique name for the root value.

2. **Format the field**.

   When calculating numeric values always format the text as a number. In the Text Field Properties window click Format.

Figure 16.5 Format the Field as a Number



Choose Number from the Select Format Category drop-down menu.

Open the Select format category drop-down menu and choose Number. Edit the Decimal places and Separator style options. In this example I choose 0 (zero) for Decimal places and use the Separator style with commas.

3. **Populate the column with fields**.
   When you're certain you have the field attributes properly assigned for the first field in a column, open a context menu and choose Create Multiple Copies. In this example I use 8 for Copy selected fields down and 1 for Copy selected fields across.

Figure 16.6 Table Populated with Fields

DESCRIPTION AND LOCATION OF PROPERTY	NATURE OF DEBTOR'S INTEREST IN PROPERTY	HUSBAND, WIFE, JOINT, OR COMMUNITY	CURRENT VALUE OF DEBTOR'S INTEREST IN PROPERTY, WITHOUT DEDUCTING ANY SECURED CLAIM OR EXEMPTION	AMOUNT OF SECURED CLAIM
Property.DescLoc.0	Property.DebtIntr.0	Property.I	currentValue.0	Property.Secured.0
Property.DescLoc.1	Property.DebtIntr.1	Property.I	currentValue.1	Property.Secured.1
Property.DescLoc.2	Property.DebtIntr.2	Property.I	currentValue.2	Property.Secured.2
Property.DescLoc.3	Property.DebtIntr.3	Property.I	currentValue.3	Property.Secured.3
Property.DescLoc.4	Property.DebtIntr.4	Property.I	currentValue.4	Property.Secured.4
Property.DescLoc.5	Property.DebtIntr.5	Property.I	currentValue.5	Property.Secured.5
Property.DescLoc.6	Property.DebtIntr.6	Property.I	currentValue.6	Property.Secured.6
Property.DescLoc.7	Property.DebtIntr.7	Property.I	currentValue.7	Property.Secured.7
		Total➤	Property.CurrentValue.Ttl	

New fields in the Current Value column.

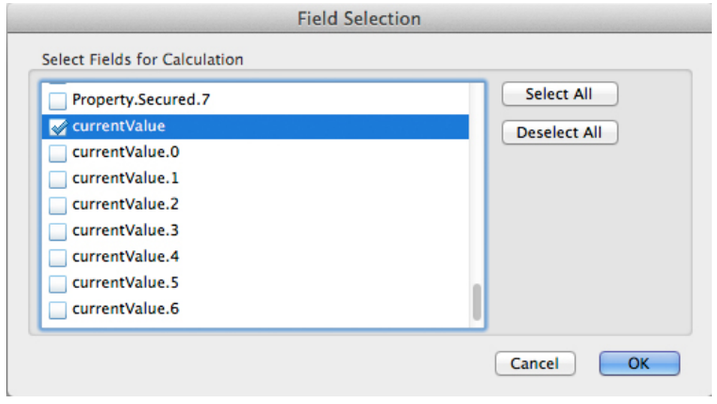
4. **Click OK and Acrobat creates eight fields down.**
   The fields are named with a hierarchical structure. In my example the field names are *currentValue.0*, *currentValue.1*, *currentValue.2*, etc.

5. **Choose a calculation formula.**
   The *Property.CurrrentValue.Ttl* field is used for the sum of the currentValue fields. In the Text Field Properties window, click the Calculate tab.

   Select the *Value is the* radio button and from the drop-down menu and choose *sum (+)*.

6. **Add the fields to be calculated.**
   Click Pick and the Field Selection window opens. In the Field Selection dialog box, scroll the list until you find *currentValue*.

Figure 16.7 Field Selection Window



Click just the root name and all fields with the same root name are aded to the formula.

> **Note**
>
> *The root name appears first followed by currentValue.0, currentValue.1, currentValue.2, etc.*

When you select a root value among a group of fields, the calculation is performed for all fields having the same root name. You do not need to specifically identify each field in the Field Selection dialog box.

7. **Click OK in the Field Selection dialog box and the calculation formula is added to the field.**

Calculating Data with Simplified Field Notation (SFN)

Simplified Field Notation (SFN) is a scripting language that resembles writing formulas similar to the way formulas are written in spreadsheet programs. The disadvantage you find when using SFN is that you cannot use hierarchical field names. SFN is helpful when you want to subtract and divide values since these formulas are not part of Acrobat's built-in formulas.

An SFN calculation might be something like (*item0* + *item1*) / *2*. Note that *item0* and *item1* are not hierarchical field names (no period exists in the names). In this example a field named *item0* is added to a field named *item1* and the result is divided by 2.

When you have fields that need some simple math routines and the field names are not described with a hierarchical name structure you can use SFN. However, for more complicated formulas, your best solutions are available with JavaScript.

Calculating Data with JavaScript

Acrobat JavaScript is not limited to calculating numeric values. You can open application alert messages, manage data, show/hide fields, show/hide layers, open/close files, create new objects, delete objects, run analyses on documents and much more using JavaScript.

In terms of calculating data, JavaScript is added to the Calculate properties like we previously added Acrobat built-in and SFN formulas.

The same field attributes need to be assigned when using JavaScript to make your forms authoring much easier. Just like we found a way to simplify calculating sums with Acrobat's built-in formulas, JavaScript is much easier when you use hierarchical names.

To compare using other formulas with JavaScript, let's use the same form discussed earlier in this chapter where we calculated a sum using Acrobat built-in formulas. The steps to follow include:

1. **Follow steps 1-3 you use for calculating sums with Acrobat JavaScript.**
   The fields should be formatted for numbers and should have hierarchical names.

2. **Select the total field and open the JavaScript Editor.**
   Click the Calculate Properties and click the Custom calculation script radio button. Click Edit to open the JavaScript Editor. The JavaScript Editor is like a text editor where you write code for the JavaScript (See Figure 16.8).

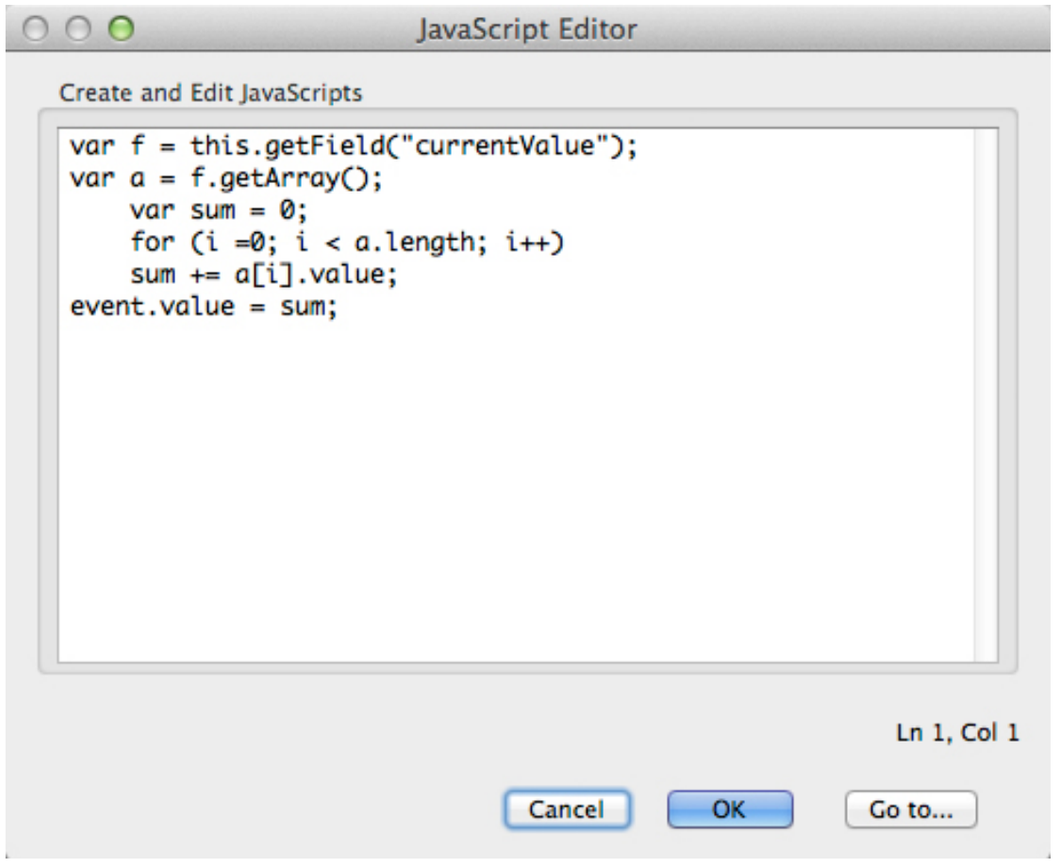
3. **Enter the following code in the JavaScript Editor:**

```
1. var f = this.getField("currentValue");
2.var a = f.getArray();
3.    var sum = 0;
4.        for (i =0; i < a.length; i++)
5.        sum += a[i].value;
6.    event.value = sum;
```

4. **Click OK in the JavaScript Editor.**
   The formula is set to calculate the *currentValue* fields in the column.

In the JavaScript for this example the first line of code sets variable f to the field *currentValue*. This is a root name of the fields in the column we need to total. In the second line of code we set up an array. We set a variable sum to a value of 0 (zero) and then follow that line with a loop. The loop runs through the entire column adding values as it loops through the fields. The result is the event.value (within the field where this script is written) and sum in line 6 is the sum total of the *currentValue* fields.

Figure 16.8 Type Code in the JavaScript Editor



Type the code and click OK.

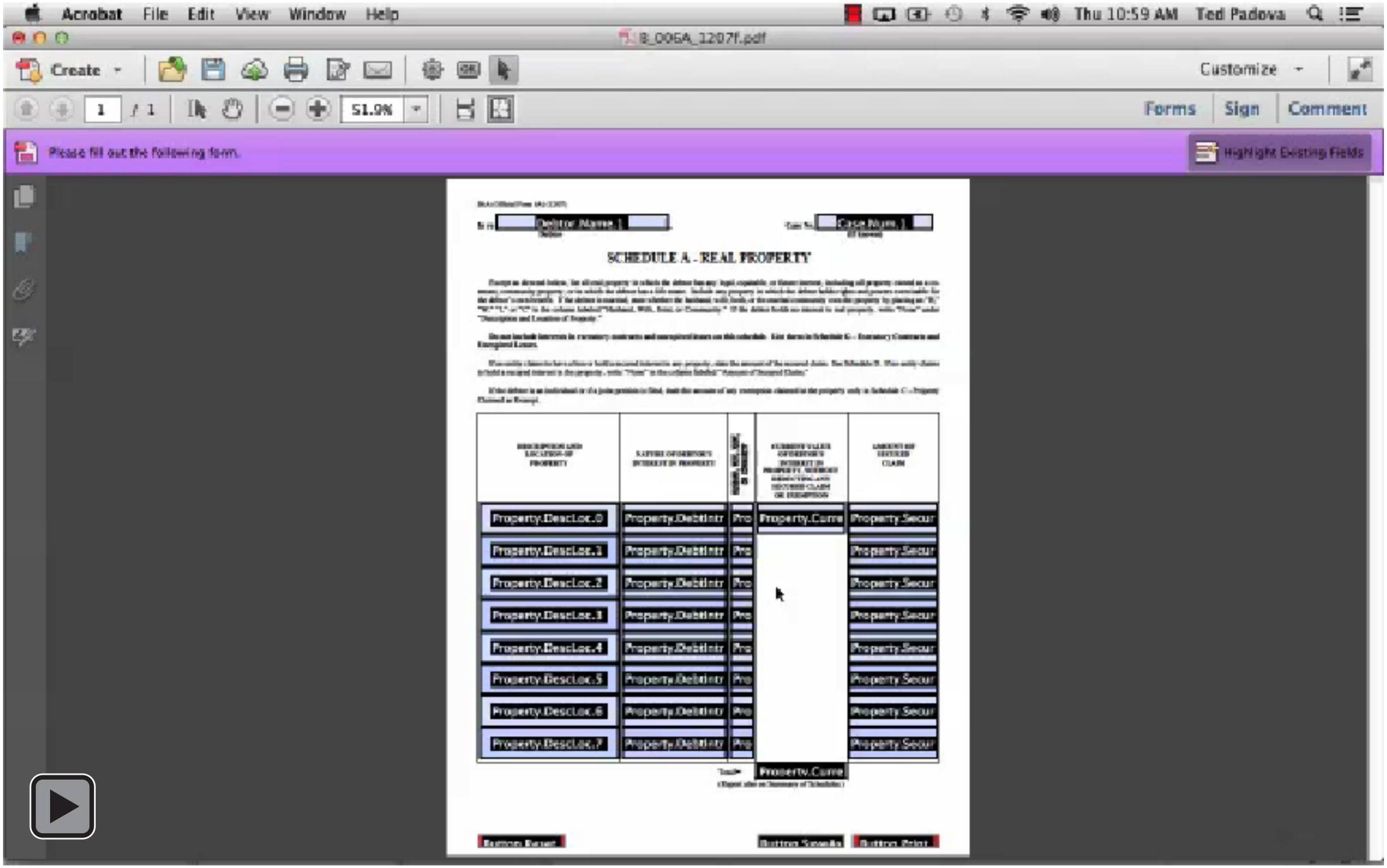
If you never looked at a JavaScript and you are not a programmer, don't be concerned. You can copy this exact same code and paste it in the JavaScript Editor for any total field where you want to sum two or more rows (or columns) of data. The only item you need to change is the *currentValue* text in the first line of code. If you use a root field name such as *total* so the fields in the

column appear as *total.0*, *total.1*, *total.2*, etc., you replace *currentValue* with *total*.

It doesn't matter how many fields you have in a column or row. You could have 2,000 or more fields. The script still works for calculating a total. In line 4 we run a loop through the length of the array (*a.length*). As long as there are additional fields, the loop keeps running until it reaches the last field.

Remember. Most eForms require only a few different JavaScripts that you can easily copy and paste. Don't be concerned about not having skills as a programmer. With just a few copy/paste JavaScripts you can make your forms dynamic and satisfy needs for your agency.

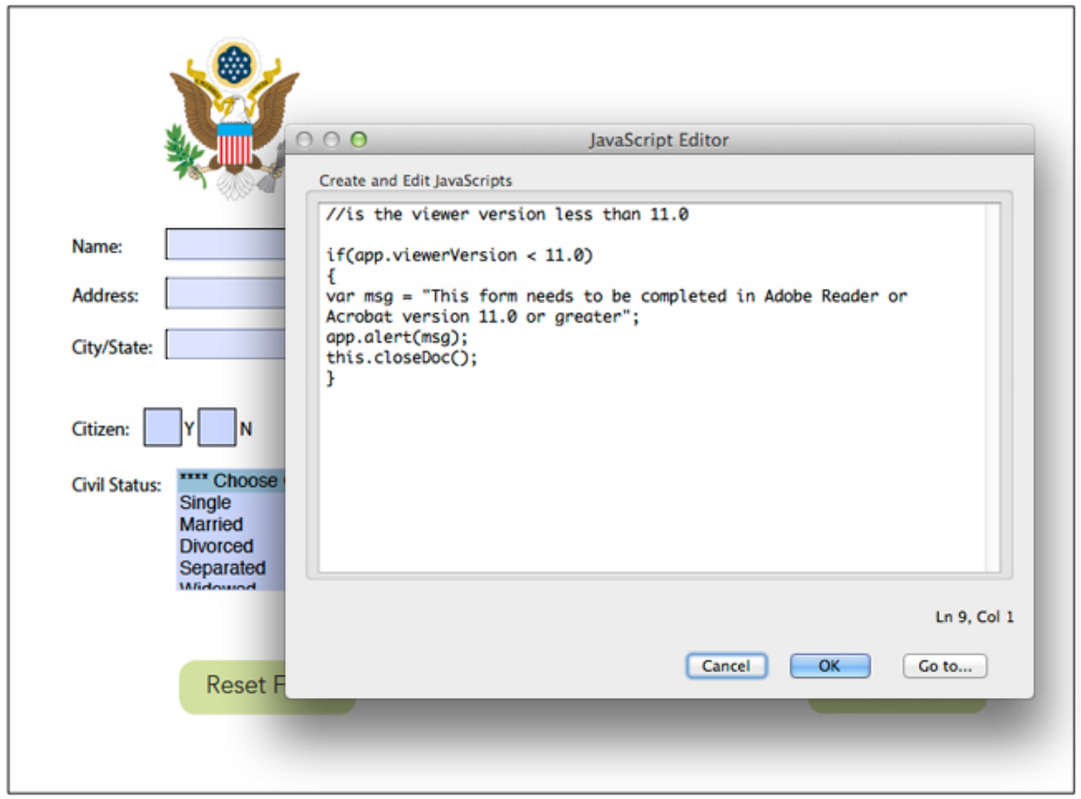
Movie


Acrobat File Edit View Window Help
Thu 10:59 AM Ted Padova
B_006A_1207f.pdf
Create
Customize
1 / 1
51.0%
Forms Sign Comment
Please fill out the following form.
Highlight Existing Fields
SCHEDULE A - REAL PROPERTY
Property.DescLoc.0
Property.DescLoc.1
Property.DescLoc.2
Property.DescLoc.3
Property.DescLoc.4
Property.DescLoc.5
Property.DescLoc.6
Property.DescLoc.7
Property.Curre
Property.Curre

Chapter 17

Working with JavaScript

Using easy copy/paste JavaScripts



Working with JavaScript

By far the most complicated aspect of creating eForms is working with JavaScript. For non-programmers, adding JavaScripts in forms can be a challenge. Fortunately though, with a fundamental understanding for where scripts are added in forms and what results you want to accomplish you can handle some of your JavaScript needs by copying and pasting scripts.

IN THIS CHAPTER

JavaScript Resources

The more you work with eForms the more you'll want to know about JavaScript. To gain some information and learning how to write scripts, there are help documents and manuals online available free for download. Information and material can be found at:

Adobe Systems.

Adobe Systems provides manuals for JavaScript they host on the Adobe website. JavaScript manuals are typically dated however much of the information provided for an earlier version of Acrobat works well for Acrobat XI. For material on JavaScript from Adobe Systems look at: http://www.adobe.com/devnet/acrobat/javascript.html.

The JavaScript API Reference manual can be downloaded from: http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/js_api_reference.pdf.

PDF Scripting.

Thom Parker, one of the world's leading authorities on Acrobat JavaScript, hosts a website www.pdfscripting.com where you can find code snippets, video tutorials, and articles on Acrobat JavaScript.

JavaScript Tips and Techniques.

I host a PDF document on my website that has 101 tips for creating PDF forms. More than half the tips address JavaScript. This free tips manual can be downloaded from http://tedpadova.com/downloads.html.

Search the Internet.

The Internet has a wealth of information related to PDF forms and Acrobat JavaScript. Do a Google or Bing search for tips and help related to JavaScript and code samples.

Using the JavaScript Console

You add JavaScripts in the JavaScript window on form fields, at the document level, on page actions, bookmarks, etc. To test and debug scripts you can use the JavaScript Console. This special window is used for temporarily testing code snippets and debugging code.

You open the JavaScript Console from the Tools ➤ Forms panel by clicking JavaScript Debugger. The keyboard shortcut is CTRL/Command + J. When working with JavaScript you'll want to remember this keyboard shortcut to quickly access the JavaScript Console/Debugger.

In the Console window you can type code and execute the code and errors are reported when scripts contain bugs. You continually interact with this panel when writing JavaScripts so

it's a good idea to keep the window open as you write and test JavaScripts.

To write a script and execute it, let's take a look at creating an application alert message. Application alerts are routines that you'll want to frequently add to eForms. You may want a message to appear when the form is opened instructing the user on how to complete or process a form, you may want an alert message to appear if data are not properly encoded, you may want to inform users on proper encoding for form fields or a host of other circumstances where you find a message alert helpful for the user.

To create and execute an application alert message, do the following:

1. **Open the JavaScript Console.**
   Press CTRL/Command + J on your keyboard.

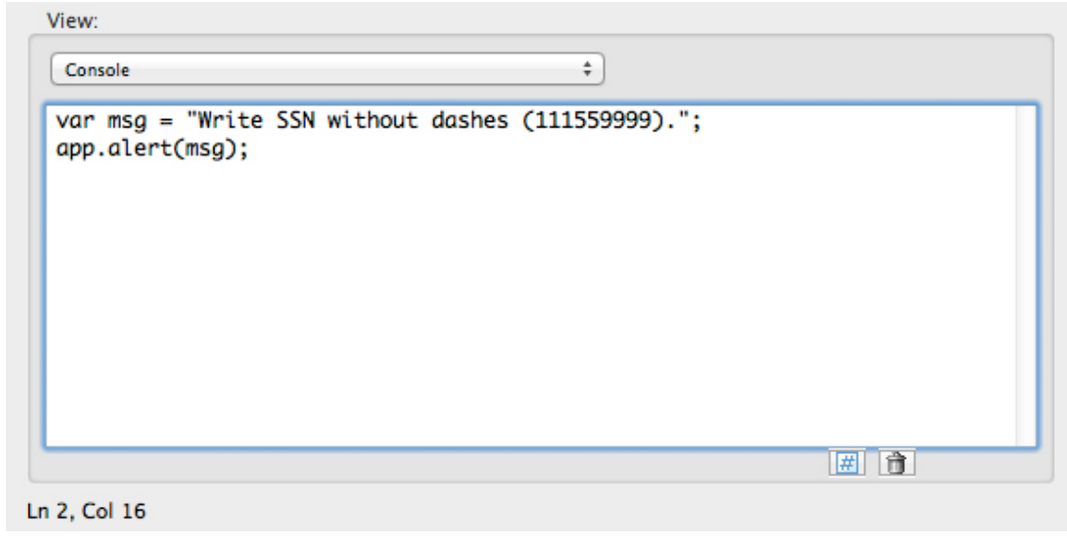
2. **Clear the default text.**
   When the console opens you find a few lines of text appearing in the Console window. Click the trash icon at the bottom right corner of the JavaScript Debugger window.

3. **Add a script.**
   For this example type in the following code:

```
1. var msg = "Write SSN without dashes
   (111559999).";

2.    app.alert(msg);
```

Figure 17.1 The JavaScript Console



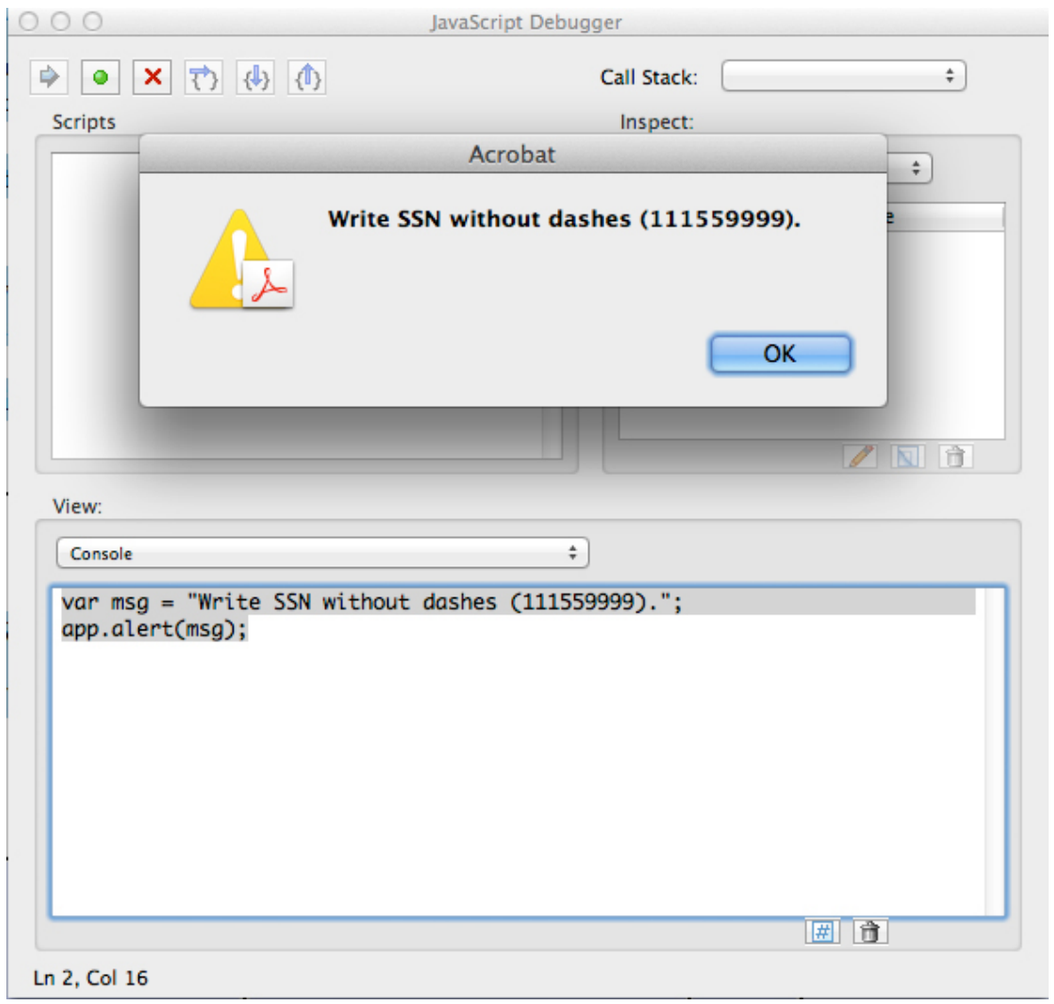
Type the code in the JavaScript Console.

3. **Select the text.**
   You can select a few lines of code in a long script to test a subroutine or you can select the entire script when testing the routine. In this case there are only two lines of code and both lines need to be selected to test the routine.

4. **Press Enter on the numeric keypad.**
   Do not press the Enter/Return key. You need to use the Enter key on your numeric keypad. If using a laptop computer you may need to add the Function key when pressing Enter.

After pressing Enter, you should see an application alert dialog box. In the JavaScript code above, the variable *msg* was assigned to the line of text within the quote marks.

Figure 17.2 The Routine Executed



Select both lines of text and press Enter (num pad key).

You could use any variable name you like such as x, message, etc. In this example we use *msg* for the variable name. The second line of code instructs Acrobat to open an alert dialog box and displays the message.

This routine is one that you can easily copy and paste in the JavaScript window when you want to add a script to a button, field, bookmark, etc. All you have to do is change the text appearing within the quote marks.

Document Level JavaScripts

When you want a script to execute upon opening a form, you add the script as a Document Level JavaScript. More complicated scripts using functions are also written at the document level.

You can write scripts that instruct Reader to open application alert messages, clear data on a form, show/hide fields, set parameters for fields, and a number of other conditions in a document level JavaScript. The document level scripts execute before the user begins to interact with a form.

Assessing Viewer Versions

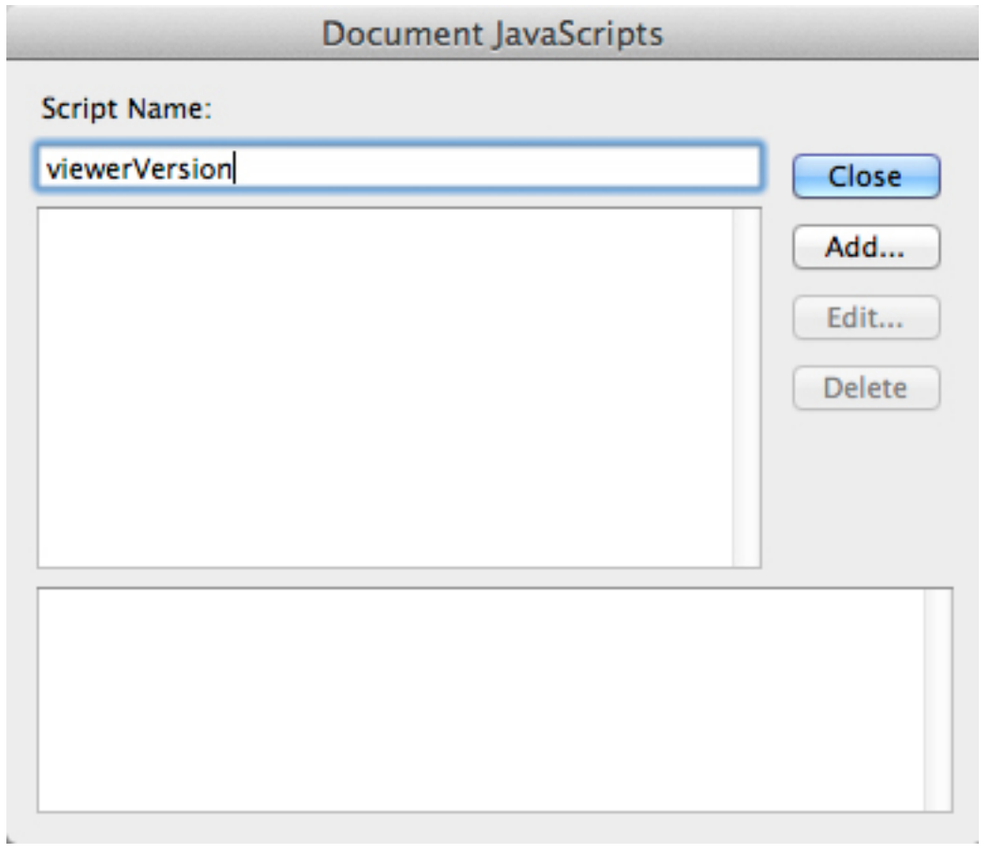
A good use for writing a document level script is assessing the user's viewer version. Suppose your form can only be completed in Adobe Reader XI. You can write a script that assesses the version of Adobe Reader on the user's computer. If the version is lower than version XI, an application alert opens. If the Reader

version is XI or greater, then nothing happens. To write a script that assesses the user's version of Adobe Reader, do the following:

1. **Open a form in Acrobat.**

2. **Open the Document JavaScript Editor.**
   In the Tools ➤ JavaScript panel click Document JavaScripts.

Figure 17.3 Document Level JavaScripts



Type a name and click the Add button to open the JavaScript Editor

3. **Open the JavaScript Editor.**
   Type a name for the script in the Script Name text box. Click the Add button to open the JavaScript Editor. In the JavaScript Editor window clear the default text (select the text and press the Backspace/Delete key on your keyboard).

4. **Type the JavaScript code.**

```
1. //is the viewer version less than 11.0?

2. if(app.viewerVersion < 11.0)

3. {

4. var msg = "This form needs to be completed
   in Adobe Reader or Acrobat version 11.0 or
   greater";
   app.alert(msg);

5. }
```

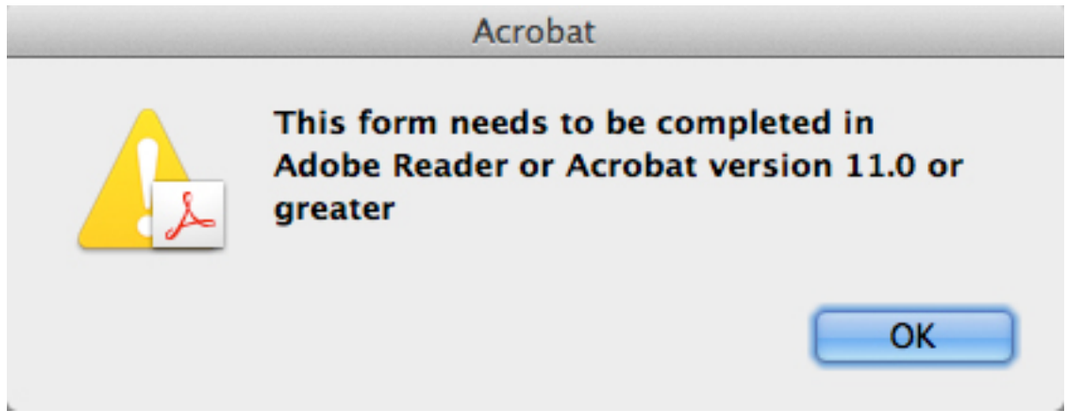
5. **Close the windows.**
   Click OK in the JavaScript Editor. When you return to the Document JavaScripts window click Close.

The first line of code is a comment line and ignored by Acrobat when the script is executed. It's a good idea to add comment lines throughout your routines where some notation is helpful for you and others to understand when modifying the form. Comment lines begin with //. You can also keep a subroutine from executing by adding /* in the first line and */ in the last line of code where you want Acrobat to ignore code when the routine is executed. This can be helpful when you debug scripts and want to isolate a subroutine to test it.

The second line of code looks at the application viewer. Regardless of whether the form is opened in Reader or Acrobat, if the viewer is less than version 11, the code within the braces is executed. If the viewer is 11.0 or greater, nothing happens.

The code within the braces sets up the message and executes an application alert message.

Figure 17.4 Application Alert Dialog Box



If the viewer version is less than version XI.0, the dialog box opens displaying the message.

If a user tries to open your form in a Reader version below XI, the dialog box opens displaying the message.

A routine like this should be included on all eForms. Some PDF forms can be populated in a viewer version 7.5 and above. However, forms may not work properly in viewer versions less than 7.5. Many users are still working with Reader versions less than 7.0 therefore it's best to give the users a heads up when a form won't work properly in viewers less than a given viewer version.

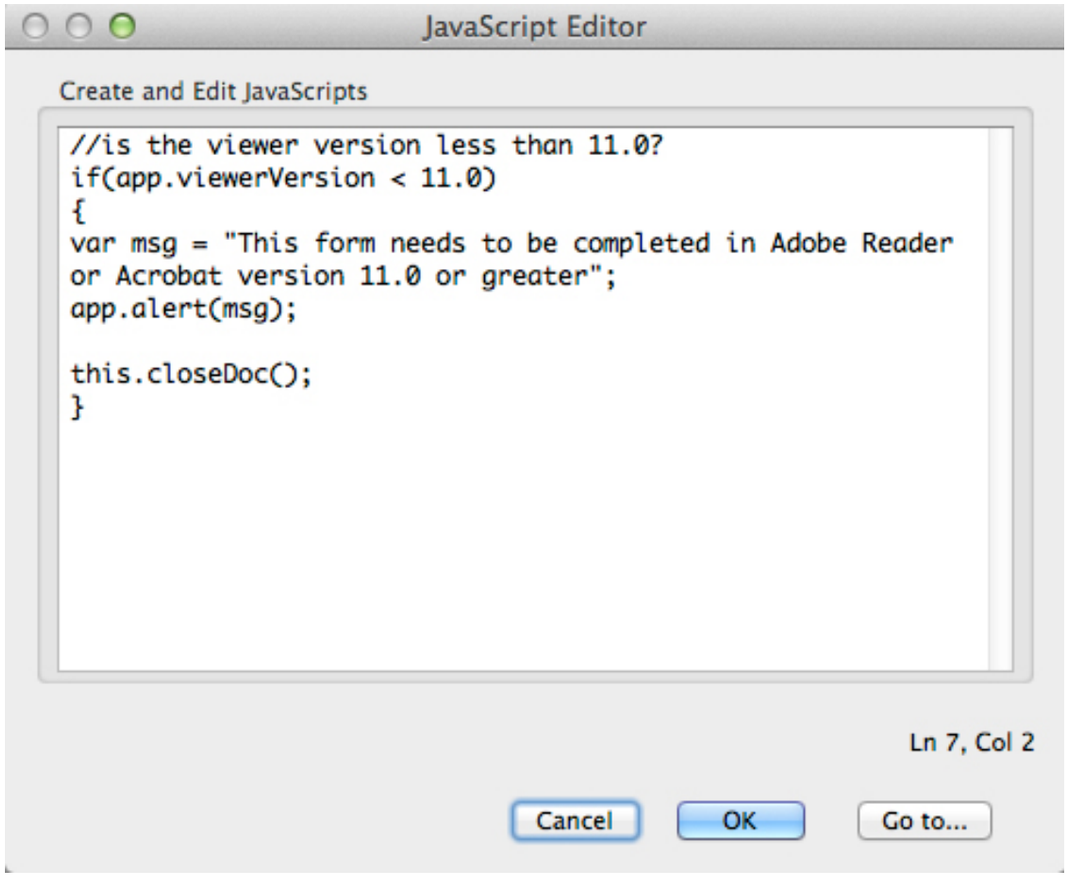
New features have been added in Reader XI that do not work in any previous version of Reader, although the routines work fine in Acrobat viewers less than XI.0. You must insure that Reader users are properly informed about what viewer version is needed to complete a form.

Force Quitting the Viewer

Suppose you want all your eForms completed in Acrobat XI and above. Some of your forms may be completed fine in earlier versions of Reader, but many forms may require using Reader XI. To standardize on a viewer version for all forms, you want to insure users are required to upgrade to Reader XI or above before they interact with your forms.

You can easily prevent users from working in a version less than the one you prescribe by adding one more line of code to the previous example.

Figure 17.5 Code for Force Closing a Form



add the last line of code before the closing brace.

Right after the line 5:

```
app.alert(msg);
```

Enter the following:

```
this.closeDoc();
```

This line of code must be within the braces and appear after the application alert opens. When the Reader user opens the form in a version less than version XI, the alert dialog box opens informing the user that the form must be completed in Reader XI or above. After they click the OK button, the file automatically closes. No editing is permitted in a viewer version less than the one you prescribe in the routine.

Adding this line of code insures that all your eForms are completed in the version you prescribe in the script. This kind of forced condition assures the constituents that when they complete forms the data are reliable and all aspects of the form work properly.

Writing Button Scripts

Button fields are contained on many forms. You might use buttons to clear data, send data from one group of fields to another or from one form to another, navigate pages such as jumping to instructions in a multi-page form and returning back to the form, submitting form data to a host agency, show/hide fields and just about anything else you can imagine. With JavaScript you are only limited by your imagination. In addition you can add other actions such as Execute a Menu item and nest JavaScripts with other actions on the same button.

Writing JavaScripts for buttons is yet one more reason you need to use hierarchical field names. If you want to hide all fields on a form that has 100 fields or more, 2 lines of JavaScript will perform the action. If every field name has a unique name it would require 202 lines of code. If you want to clear fields in one section on a form, 4 lines of code will do the job when fields have hierarchical

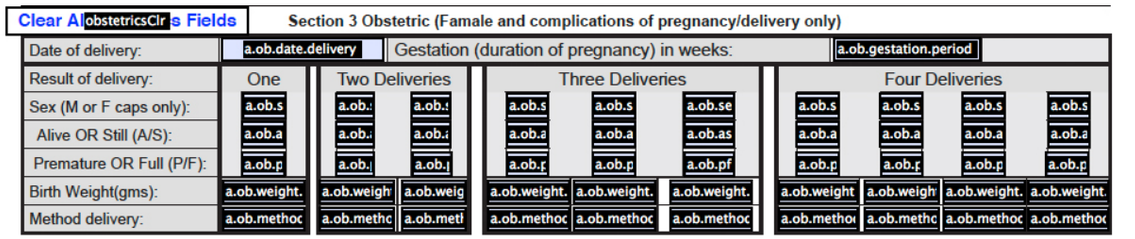
names. If fields have unique names each field name in the section needs to be referenced in the JavaScript.

Before adding buttons and writing JavaScripts, you need to be certain the form is constructed properly and all fields are properly named.

Showing/hiding fields

If fields are not visible on a form then the user isn't able to interact with the form. You may want a user to interact with a form based on conditions such as checking a box for gender. If a gender response is F (female) then questions specific to females are shown. If the response is M (male) then fields specific to males are shown. When the form opens, fields are hidden and only appear after the user checks the gender check box. In another scenario you may want all fields hidden on a form until the user checks a box agreeing to terms and conditions. If the user checks the Yes field, all fields are shown. If the user clicks No then all fields remain hidden. As an example for a situation where you may want to show/hide fields, the sample form shown here is a health form. On this form there is a section for obstetrics. Obviously the fields in the obstetrics section apply to only female patients. By default all obstetrics fields are hidden when the form opens. If the user marks the sex field with F (female) then the obstetrics fields are made visible. If the response is M (male) the fields are again hidden.

Figure 17.6 Fields in Obstetrics Section



All fields in the Obstetrics section have a root field name of a.ob,

Notice that the fields in the obstetrics section have a root name of *a.ob* and you find fields named *a.ob.sex.0*, *a.ob.weight.0*, *a.ob.date.delivery*, etc. the fields in this section have unique names but they all have a common root value of *a.ob*.

Most of the fields on this form all begin with *a*. Therefore, on this form we can add a JavaScript that clears all the *a* fields and/or a JavaScript that shows/hides only the *a.ob* fields.

Let's look at how we add a JavaScript to the Sex field that shows/hides the obstetrics fields based on the user response.

1. **Open the field properties on a field where you want to add an action to show/hide fields**.
   In this example I start with a check box for F (female).

2. **Open the Actions tab.**
   For this field we won't use the Calculate properties on the text field. The Actions tab is where we write the JavaScript.

3. **Set the mouse trigger to Mouse Up.**
   The JavaScript executes when you click the checkbox and release the mouse button.

4. **Choose Run a JavaScript from the Select Action drop-down menu.**

5. **Click The Add button to open the JavaScript Editor window.**

6. **Add the following code in the JavaScript Editor:**

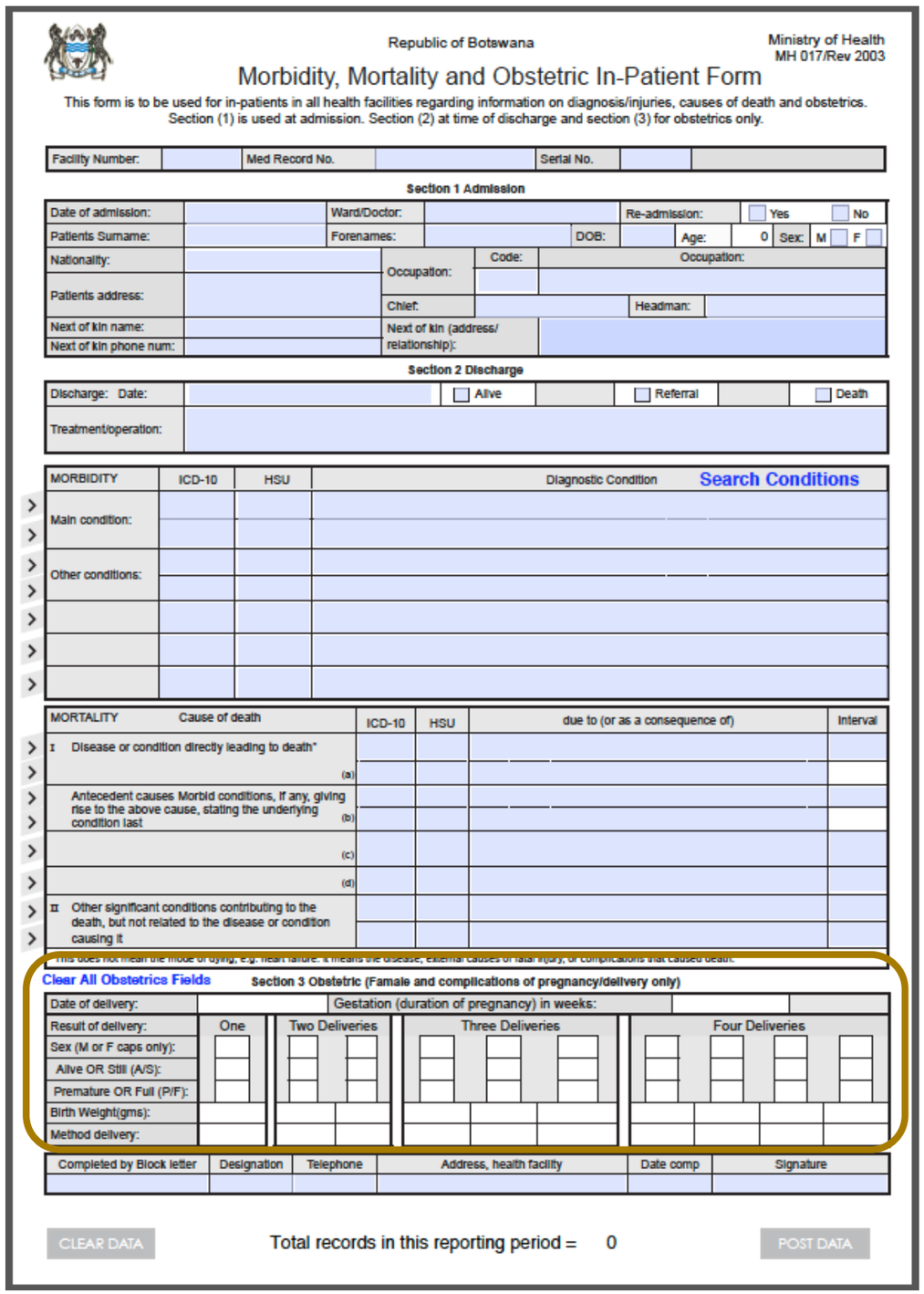
```
1. var j = this.getField("a.ob.");

2. j.hidden = false;
```

This script with only two lines of code shows all fields having a root value of *a.ob* when the F (female) check box is selected. By default the obstetric fields are hidden when the form opens. The only way to show the fields is when the user clicks the F check box. In line 1 we set the variable j to the *a.ob fields*. *this.getField("field name")* is a common term you'll use in many scripts. The second line of code takes all fields set to the variable name (in this case all the *a.ob fields*) and shows them (*j.hidden = false*).

To prevent user error you need to add a routine in the field that hides the obstetrics fields. If a user clicks the F check box first and then clicks the M checkbox, the first action shows the

Figure 17.7 Sample: Botswana Health Form


Republic of Botswana

Ministry of Health
MH 017/Rev 2003

Morbidity, Mortality and Obstetric In-Patient Form

This form is to be used for in-patients in all health facilities regarding information on diagnosis/injuries, causes of death and obstetrics.
Section (1) is used at admission. Section (2) at time of discharge and section (3) for obstetrics only.

Facility Number: | Med Record No. | Serial No.

Section 1 Admission

Date of admission: | Ward/Doctor: | Re-admission: Yes No
Patients Surname: | Forenames: | DOB: | Age: 0 | Sex: M F
Nationality: | Occupation: | Code: | Occupation:
Patients address: | Chief: | Headman:
Next of kin name: | Next of kin (address/ relationship):
Next of kin phone num:

Section 2 Discharge

Discharge: Date: | Alive | Referral | Death
Treatment/operation:

MORBIDITY | ICD-10 | HSU | Diagnostic Condition | Search Conditions
Main condition:
Other conditions:

MORTALITY Cause of death | ICD-10 | HSU | due to (or as a consequence of) | Interval
I Disease or condition directly leading to death* (a)
Antecedent causes Morbid conditions, if any, giving rise to the above cause, stating the underlying condition last (b)
(c)
(d)
II Other significant conditions contributing to the death, but not related to the disease or condition causing it

*This does not mean the mode of dying, e.g. heart failure. It means the disease, external causes of fatal injury, or complications that caused death.

Clear All Obstetrics Fields — Section 3 Obstetric (Famale and complications of pregnancy/delivery only)

Date of delivery: | Gestation (duration of pregnancy) in weeks:
Result of delivery: | One | Two Deliveries | Three Deliveries | Four Deliveries
Sex (M or F caps only):
Alive OR Still (A/S):
Premature OR Full (P/F):
Birth Weight(gms):
Method delivery:

Completed by Block letter | Designation | Telephone | Address, health facility | Date comp | Signature

CLEAR DATA | Total records in this reporting period = 0 | POST DATA

Sample form having two check boxes for gender choice at the top of the form and fields for obstetrics choices at the bottom of the form.

obstetric fields. When M is checked you need to re-hide the obstetric fields. The same script is used except line 2 is edited as follows:

```
1. var j = this.getField("a.ob.");

2. j.hidden = true;
```

Remember that when a form is secured against making changes and form fields are hidden there is no way for a user to interact with the hidden fields. You might find that hiding fields used specifically for official use helpful on a form. You could add a button that opens a dialog box prompting the user for a password. If the password is entered correctly, the fields are shown. This method requires those who retrieve forms to simply enter a password rather than having to edit the security settings. (See Chapter 20 for more on working with Official Use Only fields and adding passwords to unlock the fields).

Locking Fields (readonly)

Locking fields is helpful when you want the data in a field displayed but you don't want users to edit the field(s). Calculation formulas are a good example for when you want the results displayed but the field locked against editing the results. Another use for read only fields is when you want a user to populate a section of a master form from another form and use a button to send the completed data to the master form. If certain fields on a master form are read only, you can force users to open another form, populate the form, and click a button that sends the data to the master form.

As an example, look at the sample from the Botswana Ministry of Health form here. This form contains in-patient registration cards. The idea is that patients are registered when they are admitted. Another form is used to discharge a patient. A third form collects data from the discharge form in a summary form that is submitted to the central office monthly.

Figure 17.8 Registration Card

Facility Code:	10E 45	Med Rec Num	116729	Serial no.:	65174 A	Date Admin	17/05/12	
Patients Surname:	Mogambi			Forenames:	Eugene			
Ward/Doctor:	Dr. Michael Barko	Re- admin	☐ Yes ☒ No	DOB	22/11/87	Sex:	M	
Nationality:	Botswana	Code: 247	Occupation	Mining Engineers, Metallurgists & Related Professionals				
Patients address 1:	234 Gabarone Avenue							
Patients address 2:	Apt 45	Chief:	Festus Khama	Headman:	Ketumile Mogae			
Next of kin name:	Naledi Barko	Next of kin Address/ Relationship:	Adress same as patient	234 Gabarone Avenue, Apt 45			Occ Codes	
Next of kin Phone:	212-7684						Send Data	

The user completes the registration information then clicks the Send Data button (lower right corner) to send the data to the discharge form.

When a patient is ready for discharge the health worker clicks the Send Data button. The action on this button opens the master discharge form and sends the data from the registration form to the master discharge form.

After the data are sent to the second form, the registration information is cleared guaranteeing that the registration file contains only those patients who are currently in a hospital or health clinic. This helps heath officials tabulate total patients in

hospitals/clinics at any given time and the data collected from the discharge forms is accumulated for analysis.

Figure 17.9 Discharge Form

Republic of Botswana

Ministry of Health
MH 017/Rev 2003

Morbidity, Mortality and Obstetric In-Patient Form

This form is to be used for in-patients in all health facilities regarding information on diagnosis/injuries, causes of death and obstetrics.
Section (1) is used at admission. Section (2) at time of discharge and section (3) for obstetrics only.

Facility Number:	10E 45	Med Record No.	116729	Serial No.	65174 A	

Section 1 Admission

Date of admission:	17/05/12	Ward/Doctor:	Dr. Michael Barko			Re-admission:	☐ Yes	☒ No			
Patients Surname:	Mogambi	Forenames:	Eugene	DOB:	22/11/87	Age:	0	Sex:	M ☒	F ☐	
Nationality:	Botswana	Occupation:	Code:	Occupation:							
Patients address:	234 Gabarone Avenue, Apt 45		247	Mining Engineers, Metallurgists & Related Professionals							
		Chief:	Festus Khama			Headman:	Ketumile Mogae				
Next of kin name:	Naledi Barko	Next of kin (address/ relationship):	234 Gabarone Avenue, Apt 45								
Next of kin phone num:	212-7684										

Section 2 Discharge

Discharge: Date:		☐ Alive		☐ Referral		☐ Death
Treatment/operation:						

MORBIDITY	ICD-10	HSU	Diagnostic Condition Search Conditions
Main condition:			
Other conditions:			

MORTALITY Cause of death	ICD-10	HSU	due to (or as a consequence of)	Interval
I Disease or condition directly leading to death*				
(a)				
Antecedent causes Morbid conditions, if any, giving rise to the above cause, stating the underlying condition last (b)				
(c)				
(d)				
II Other significant conditions contributing to the death, but not related to the disease or condition causing it				

*This does not mean the mode of dying, e.g. heart failure. It means the disease, external causes of fatal injury, or complications that caused death.

Clear All Obstetrics Fields **Section 3 Obstetric (Famale and complications of pregnancy/delivery only)**

Date of delivery:		Gestation (duration of pregnancy) in weeks:									
Result of delivery:	One	Two Deliveries		Three Deliveries			Four Deliveries				
Sex (M or F caps only):											
Alive OR Still (A/S):											
Premature OR Full (P/F):											
Birth Weight(gms):											
Method delivery:											

Completed by Block letter	Designation	Telephone	Address, health facility	Date comp	Signature

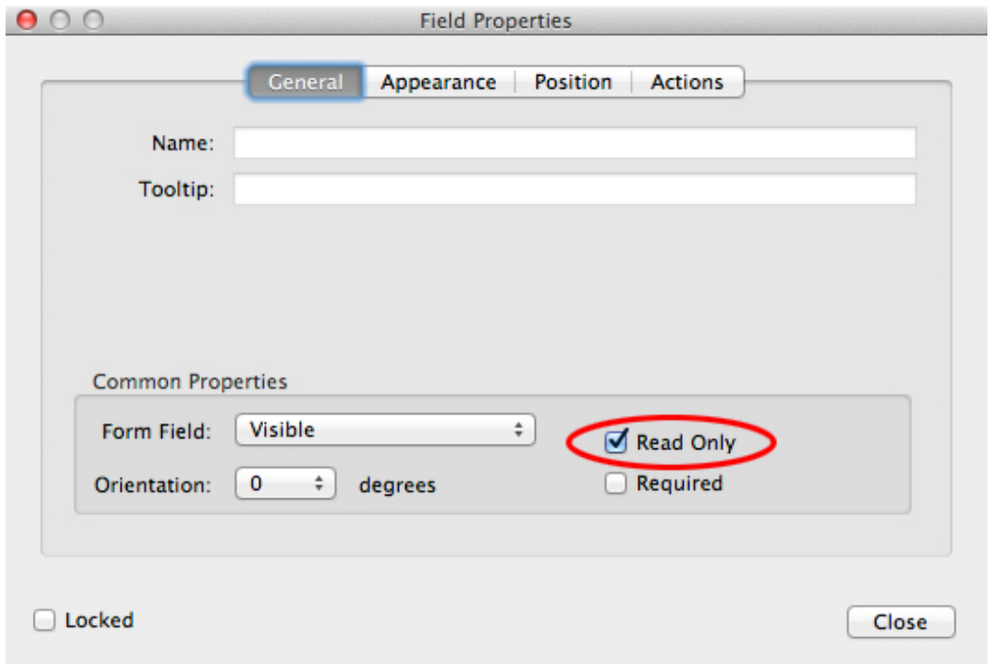
CLEAR DATA Total records in this reporting period = 0 POST DATA

The top section contains read only fields. Data are sent from the Registration Cards to this form.

Rather than have health workers rekey data that are contained on a registration form, a button is used to send the existing registration data to the final discharge form. To prevent users from changing the original registration data, the registration data on the master discharge form are sent to read only fields.

The script to send the data from the registration form to the discharge form is extensive so I won't bother you with the script information. For now realize that you can set up fields in a form to read only, so certain fields are locked when a form opens. This setting is in the General properties for all field types. Notice the checkbox for Read Only in the Common Properties in the General tab.

Figure 17.10 General Properties



Read Only checked.

If you have a form that requires a user response to make the fields active, you can globally change the field attributes and turn off (or on) Read Only.

Notice in the master discharge form most of the fields have a root name of *a*. Suppose these fields are locked when the form opens and suppose further that the user needs to click a button to agree to terms and conditions in order to interact with the *a* fields.

An Agree button would have a JavaScript action for unlocking the fields written as follows:

```
1. var f = this.getField("a");

2. f.readonly = false;
```

Once again notice the benefits for using hierarchical field names. When you begin many forms you may not immediately know what interactivity you wish to control on a form. Therefore, even if you initially feel that you won't require hierarchical names, get in the habit of using them. Later, if you modify a form that requires using hierarchical names, you won't need to completely redesign the form.

Navigating Pages with JavaScript

Buttons can be helpful to assist with page navigation on long forms where you may have instructions included on many separate pages. You may wish to place buttons to assist the user in finding help information related to filling out a given section.

You might add a button action to send the user to a specific page in a document. A script like the following is used if you write a JavaScript to navigate pages.

```
this.pageNum = 25;
```

JavaScript is zero based meaning that JavaScript sees the first page in a document as page 0, followed by 1, 2, 3, etc. The above script takes the user to page 26. You can use other actions besides JavaScripts to navigate pages such as the Go to a page view. However, writing JavaScript is much faster. You can copy/paste the script and edit the page number much faster than moving through pages to set the page result in the Create Go to View dialog box.

Other page navigation scripts that are handy include the following:

```
this.pageNum++;

this.pageNum--;
```

When a user completes a form on one page and needs to move to the next page the (++) takes the user to the next page. The (--) takes the user to the previous page.

You may want to spawn a page from a template (see Chapter 18 for spawning pages from templates). When you write a script that spawns a page from a template, the new page is added to the

document but the user remains on the same page where the button used to spawn a new page exists. Typically it helps users to move to the new page when a new page is added. After spawning a page from a template you'll want to add page navigation at the end of the script.

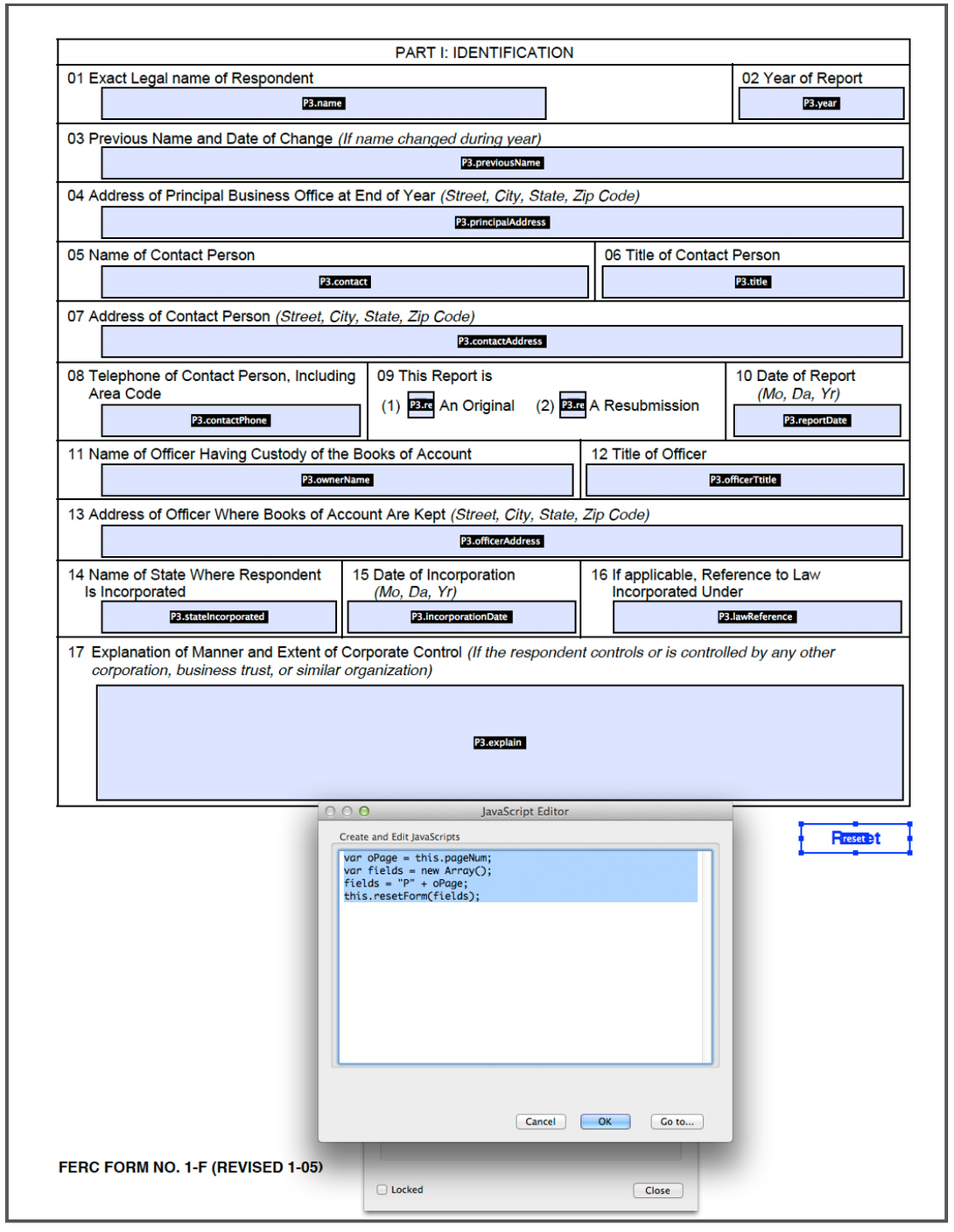
Clearing Data (reset form)

There's an action for Reset a form in the Select Action drop-down menu. When you use this action type, a dialog box opens and you have a choice to clear all fields or only selected fields you check in the dialog box.

You might have a form where a user adds identifying information followed by several sections to fill out. Perhaps a button action that clears all data except the identifying information helps the user start again in a section. In another situation, you may have forms with many different pages and want to add a reset button to clear data only on a page in view.

The sample form from the US Federal Regulatory Commission has 60 pages. Fifty-seven of the pages contain form fields. It would be a laborious task to add a reset button on 57 pages using the Reset a form action. Using JavaScript however, we can greatly simplify the task and add a reset button on each page that clears only the data on the respective page. We can create reset buttons for clearing page data in a matter of minutes.

Figure 17.11 Sample Form



Field names for the 4th page in the document have a common root value of P3.

When you design a form with multiple pages you must first think about the interactivity you want in the form before you begin adding field objects. On the sample form I want to place a reset form button to clear data on each page. Therefore I need to think about how to name the fields properly to facilitate creating the reset buttons with as few steps as possible.

JavaScript permits you to use the current page number and assign it as a variable. Therefore if the root name for fields is something like *P3*, *P4*, *P5*, etc., you can use *P(n)* where *n* is a variable. On the sample form I use a variable name *oPage* to make the variable name a little more descriptive and assign it to the page number.

The fields on the pages need to have a root name of *P* plus the page number such as *P3*, *P4*, *P5*, etc.

Note

You don't have to use P to begin the field root name. You can use Page or any other name you want for the root name. You just need to be certain the name follows with the variable we assign to the page number.

On the sample form used in this example, the first three pages contain information and there are no form fields on these pages. The actual form begins on page 4. However, remember that JavaScript is 0 based, hence, in JavaScript terms, page 4 is actually page 3. Therefore, I begin on the fourth page and use the root name *P3* (JavaScript sees the page as page 3 while Acrobat sees the page as page 4). The first two fields at the top of the form on page 4 are named *P3.legalName*, *P3.reportYear*. The next page in the form uses the root name *P4* and so on.

When I add a button to the P3 page I add a JavaScript in the Actions tab and open the JavaScript Window where I add the following script:

```
1. var oPage = this.pageNum;

2. var fields = new Array();

3. fields = "P" + oPage;

4. this.resetForm(fields);
```

In the above script the variable oPage is assigned to the current page in view. Therefore when on the P3 page, the *oPage* value is 3 (*this.pageNum*). In line 3 the fields variable is set to P + the *oPage* value or P3. The last line of code clears the data for only the P3 fields.

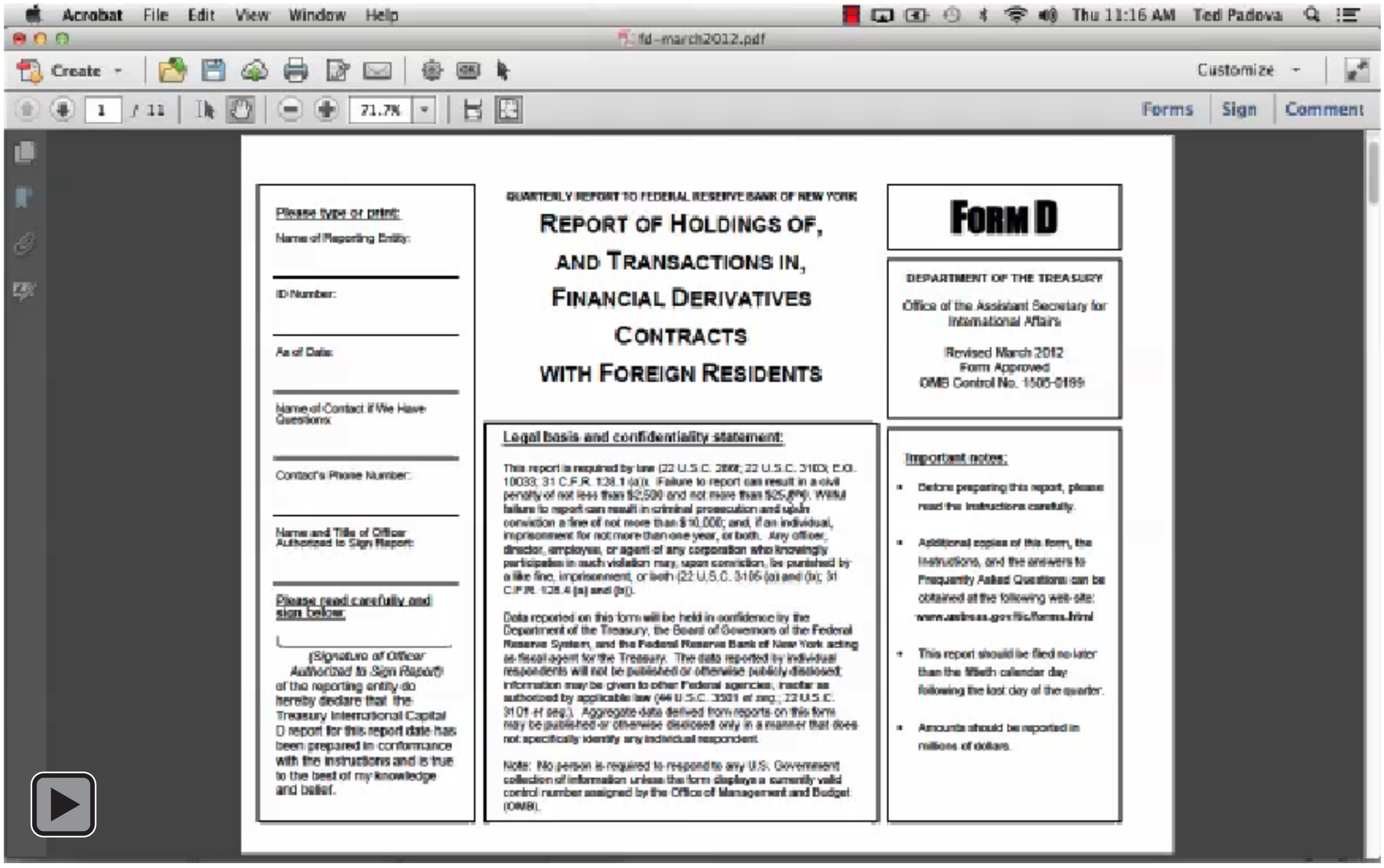
Since we use a variable for the page number and name fields containing the page number you can duplicate this button across the remaining pages. No separate code needs to be written for each page. Simply test the script to be certain it works properly and open a context menu on the button. From the menu options choose Duplicate Across Pages. When the Duplicate Field dialog box opens enter the page range. In my example I click the From radio button and type 5 in the first text box and 60 in the second text box.

Note

In the Duplicate Field dialog box we don't use 0 (zero) for the first page number. Since Acrobat displays the current page as Page 4, the duplicate fields need to start on page 5 —the next logical page.

Creating a reset button with 4 lines of JavaScript code and duplicating the button across 57 pages can be accomplished in a few minutes. If we don't use hierarchical names it would virtually take hours of development time to create individual reset buttons on 57 pages.

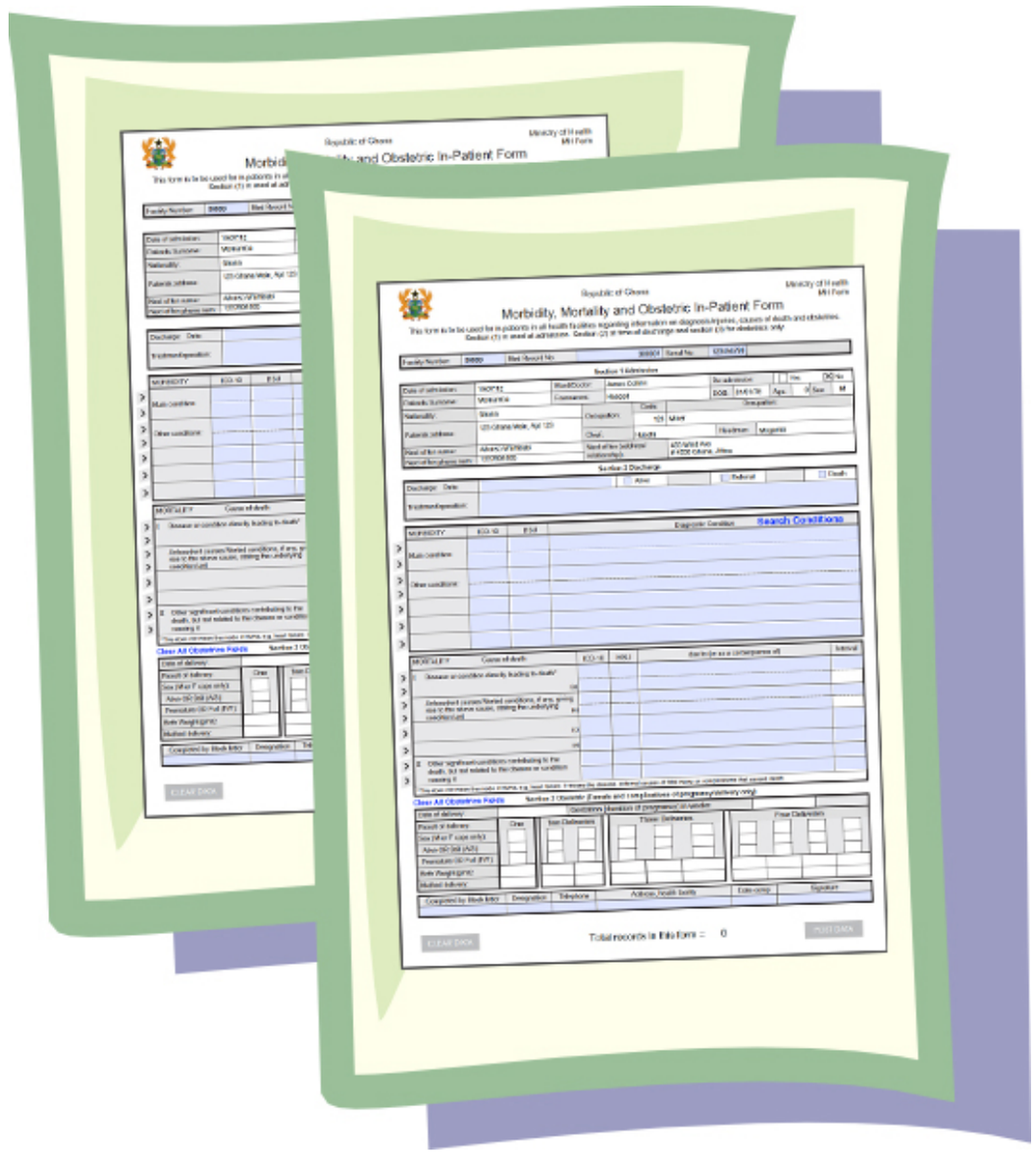
Movie


Acrobat File Edit View Window Help
Thu 11:16 AM Ted Padova
fd-march2012.pdf
Create
Customize
1 / 11
71.7%
Forms Sign Comment
Please type or print:
Name of Reporting Entity:
ID Number:
As of Date:
Name of Contact if We Have Questions:
Contact's Phone Number:
Name and Title of Officer Authorized to Sign Report:
Please read carefully and sign below:
(Signature of Officer Authorized to Sign Report)
of the reporting entity do hereby declare that the Treasury International Capital D report for this report date has been prepared in conformance with the instructions and is true to the best of my knowledge and belief.
QUARTERLY REPORT TO FEDERAL RESERVE BANK OF NEW YORK
REPORT OF HOLDINGS OF, AND TRANSACTIONS IN, FINANCIAL DERIVATIVES CONTRACTS WITH FOREIGN RESIDENTS
Legal basis and confidentiality statement:
This report is required by law (22 U.S.C. 286f; 22 U.S.C. 3105; E.O. 10033; 31 C.F.R. 128.1 (a)). Failure to report can result in a civil penalty of not less than $2,500 and not more than $25,000. Willful failure to report can result in criminal prosecution and upon conviction a fine of not more than $10,000; and, if an individual, imprisonment for not more than one year, or both. Any officer, director, employee, or agent of any corporation who knowingly participates in such violation may, upon conviction, be punished by a like fine, imprisonment, or both (22 U.S.C. 3105 (a) and (b); 31 C.F.R. 128.4 (a) and (b)).
Data reported on this form will be held in confidence by the Department of the Treasury, the Board of Governors of the Federal Reserve System, and the Federal Reserve Bank of New York acting as fiscal agent for the Treasury. The data reported by individual respondents will not be published or otherwise publicly disclosed; information may be given to other Federal agencies, insofar as authorized by applicable law (44 U.S.C. 3501 et seq.; 22 U.S.C. 3101 et seq.). Aggregate data derived from reports on this form may be published or otherwise disclosed only in a manner that does not specifically identify any individual respondent.
Note: No person is required to respond to any U.S. Government collection of information unless the form displays a currently valid control number assigned by the Office of Management and Budget (OMB).
FORM D
DEPARTMENT OF THE TREASURY
Office of the Assistant Secretary for International Affairs
Revised March 2012
Form Approved
OMB Control No. 1505-0189
Important notes:
Before preparing this report, please read the instructions carefully.
Additional copies of this form, the instructions, and the answers to Frequently Asked Questions can be obtained at the following web site: www.ustreas.gov/tic/forms.html
This report should be filed no later than the fifteenth calendar day following the last day of the quarter.
Amounts should be reported in millions of dollars.

Chapter 18

Dynamically Adding Pages to Forms

In Adobe Reader XI you can now spawn pages from templates



Chapter 18

Acrobat provides a means where you can dynamically add new pages in a form. Adding new pages requires you to first set up a page template then write a script to spawn the page from a template.

IN THIS CHAPTER

Spawning pages from templates was not possible in Adobe Reader prior to version XI. The addition of this feature and importing button faces adds much more functionality for eForms. In terms of spawning pages you may have forms where amplified details help provide additional space for more lengthy responses. A job application form, for example, may be brief for job entry-level people but may need much more room for professional vitae items. In this scenario you can design a brief form and let the user add more pages as needed.

In another example you may want to add summary data from one form to aggregate data over a period of time. Even if you know the maximum number of pages required in a form, you would have to add the maximum pages before deploying the form. Since you can now spawn pages from templates in Adobe Reader you can permit the user to add pages as needed. This makes deploying forms much easier since a single page form would require less disk space than a form with many pages and having many fields. The more fields you add to a form the larger the file sizes grow.

Creating Templates

To spawn a page from a template you must first have a page in a form identified as a template. You add fields to template pages just like you would add fields on any form. You can use hierarchical names and follow the same rules you use for designing any form.

Templates can be created as separate documents and inserted in a master form or you can start with a page already included in a form and convert the page to a template. To easily add a page to an existing form, open the Page Thumbnails panel and drag a form from the desktop to the Page Thumbnails panel.

It doesn't matter where the page you intend to covert to a page template resides in a form. Template pages can be hidden and are not identified in the total pages for a form. For example, if a form has 3 pages and you turn one page into a template, you can choose to hide the template page. If you hide the page template, Acrobat/Reader sees the file as having only two pages.

The first thing you want to do is be certain the form to be used as a template works properly, fields are formatted properly, and all the JavaScripts, if any exist, work properly. When you're certain the form field elements are properly set up, you then open the Page Templates tool from the Document Processing panel.

To create a page template, do the following:

1. **Open the page you want to add as a page template in the Document Pane.**
   You must have the page in view to make the page a template.

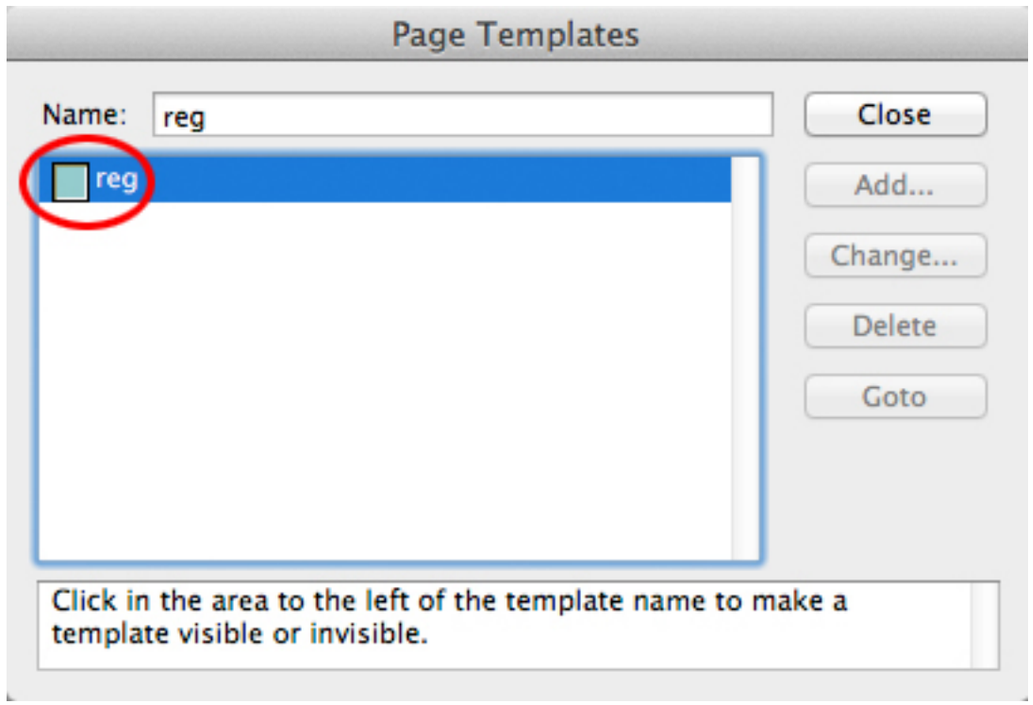
2. **Open the Page Templates window.**
   In the Tools ➤ Document Processing panel click Page Templates. The Page Templates window opens.

3. **Provide a name for the template.**
   Type a Name in the Name text box in the Page Templates window to add a name. Names you provide here should be simple names. When you write JavaScripts for interacting with a page template you can use the template page name.

   Returning to the example of a registration form I used in the last chapter, I add the name *reg* for my template name.

4. **Add the template.**
   Click the Add button in the Page templates window to add the page as a template.

Figure 18.1 Page Templates Window



Click the eye icon inside the box to the left of the page template name to hide the template.

5. **Hide the template.**
   It would be rare to have a template visible in your form. Typically you always want to hide a template page especially if the template contains form fields. This prevents users from entering data in form pages converted to templates. When you spawn a page from a template you'll want to start with a fresh clean page containing no data in the data fields.

   Click the eye icon to the left of the template name to hide the template.

6. **Close the Page Templates window.**

When you close the window the page template is hidden. The only way you can edit the template is to open the Page Templates window and make the template visible.

After the page template is created you can write a script to add duplicate pages in your form. This can be handled with button actions or you can tab out of the last field on a page and run the script on an On Blur action to automatically spawn a page. The latter is transparent to the user since the user doesn't need to click a button to spawn a page.

Writing a Script to Spawn a Page From a Template

Let's use a button click to spawn a page from a template. Create a button for adding a new page. Use the Run a JavaScript action and click Add to open the JavaScript window. In the JavaScript window, write the following script:

```
1. var a = this.getTemplate("reg");

2. a.spawn ({

3.  nPage:this.pageNum+1,

4.  bRename:true,

5.  bOverlay:false,

6. })
```

The first line of code sets the template (*reg*) to the variable *a*. The name you added in the Page Templates window should be within the quote marks. The second like of code instructs Acrobat to spawn a page from the named template. Lines 3-5 describe attributes for the spawned page. In line 3 we determine where the new page is placed in the document. In the code above it's the current page number plus 1. If you want the page placed at the end of the document you would use
this.pageNum = this.numPages-1;

In line 4 we instruct Acrobat to rename the fields. If you set bRename to false, the fields will have the same field names and you won't be able to add unique data to the fields.

The overlay is set to false. You can spawn pages so the spawned page overlays the current page or creates a new page in the document. We want a new page with unique field names so the overlay switch is turned off (*false*).

When you click a button with this kind of script a new page is added to the document that carries with it all the characteristics of the template page.

Understanding Field Names in Spawned Pages

Acrobat needs a mechanism to rename fields with unique names. The way renaming fields is handled is using the page number were the spawned page resides. If you have two pages in a document and spawn a new page, the new page fields are renamed with a root name of *P2*. (Remember, JavaScript is zero based so it sees the third page as page two. Spawn another page and the root name changes to *P3*.

If a template page has a field name like *nameLast*, the spawned page renames the field to *P2.nameLast*. The numeric value is a description of the page number. You can use a variable to describe page numbers if you need to loop through fields on several pages. You might add a line of code like:

```
var oPage = this.pageNum;
```

The above code picks up the page number and assigns it to the variable *oPage*.

Suppose you want to perform an action that retrieves a field name on a spawned page. If you added the line of code above to the script you would follow the first line with:

```
this.getField("P" + oPage + ".nameLast";
```

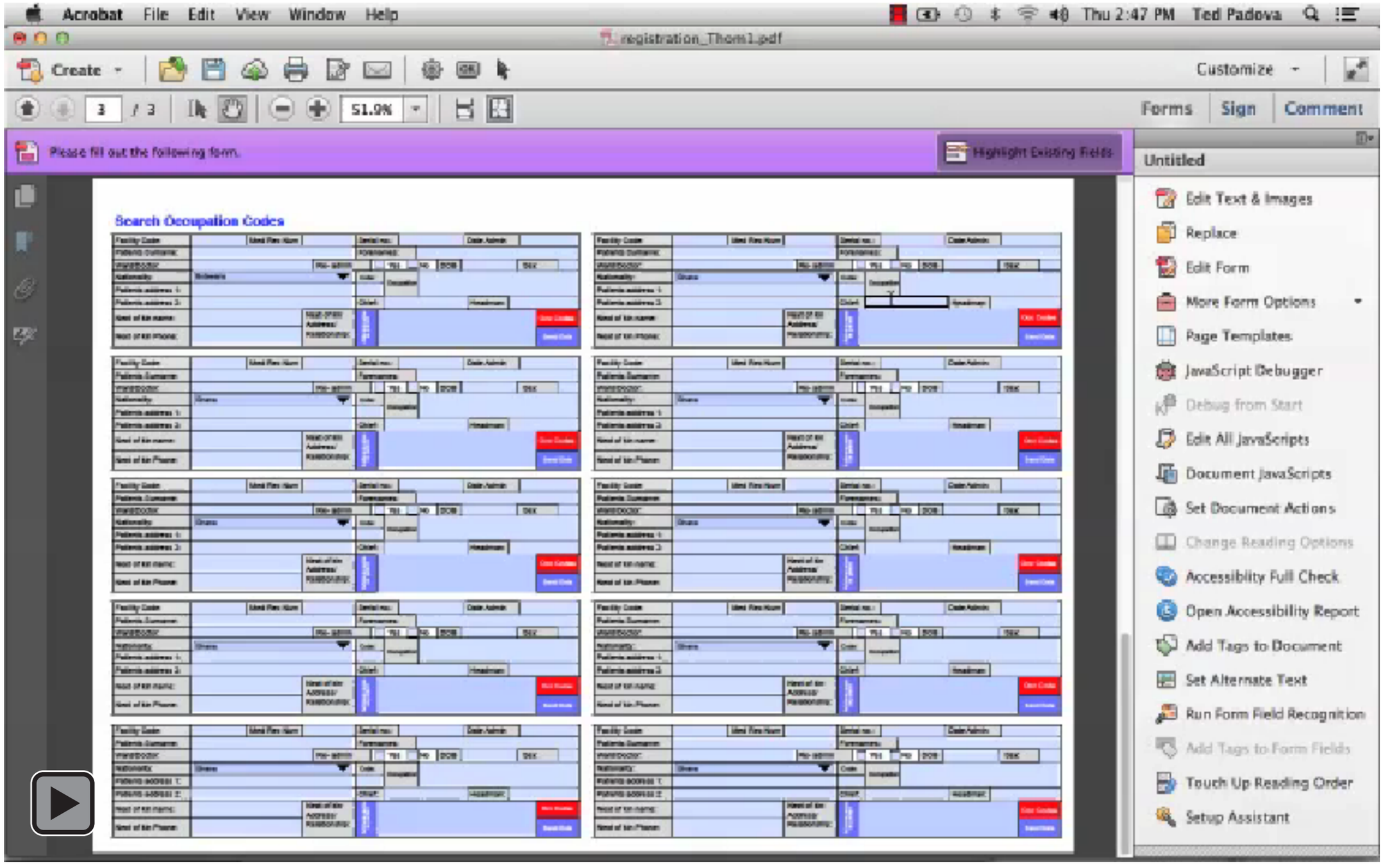
Notice the period inside the quote marks. This is essential since the field name is *P3.nameLast*.

If you have multiple pages and a button to perform an action such as sending data to another form or another page in a form you could use the same script on a button placed on every page. Since the oPage value is different on each page, such as on page 9 in a document the oPage value is 8 therefore the code: `this.getField("P" + oPage + ".nameLast";` gets the field `P8.nameLast`.

In short, you can create a script for a button field, test the script to be certain it works properly. Then add the button to the template page and all spawned pages have a button to spawn a new page. The same script works on all pages since the oPage value is determined by the page number where the button is activated.

For more about using page numbers assigned to variables see Chapter 17.

Movie


Acrobat File Edit View Window Help
Thu 2:47 PM Ted Padova
registration_Thom1.pdf
Create
Customize
3 / 3
51.9%
Forms Sign Comment
Please fill out the following form.
Highlight Existing Fields
Search Occupation Codes
Untitled
Edit Text & Images
Replace
Edit Form
More Form Options
Page Templates
JavaScript Debugger
Debug from Start
Edit All JavaScripts
Document JavaScripts
Set Document Actions
Change Reading Options
Accessibility Full Check
Open Accessibility Report
Add Tags to Document
Set Alternate Text
Run Form Field Recognition
Add Tags to Form Fields
Touch Up Reading Order
Setup Assistant

Chapter 19

Adding Button Faces with JavaScript

You can add a button to a form and permit the Adobe Reader user to change the button face using a simple JavaScript

Adding Button Faces with JavaScript

Chapter 19

IN THIS CHAPTER

Creating Buttons

Writing the JavaScript

As I explained earlier in Chapter 12, Adobe Reader XI now supports importing button faces. Adding a button field in Acrobat and opening the form in Reader doesn't permit the Reader user the ability to change a button face. You need to have a JavaScript written in the Actions tab in order to import a graphic.

Changing button appearances is particularly helpful when you want the user to add a photo on a form. Many countries require photos attached to application forms for schools, government offices, employer applications, and such similar organizations.

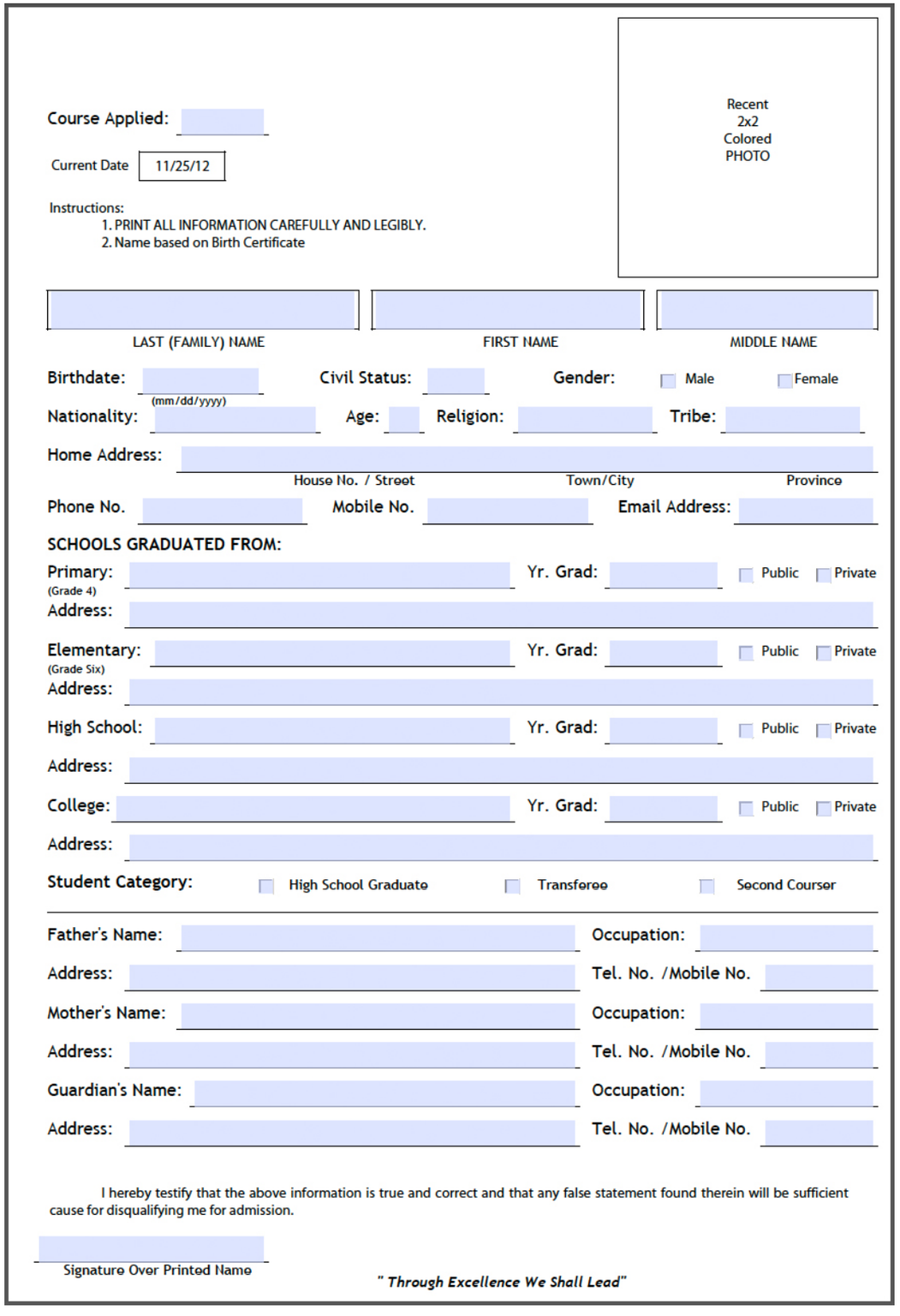
> **Note**
>
> *As I mentioned in Chapter 12, button faces are imported from PDF files only. Acrobat supports many different image types and signature appearances support many different image types, but button faces, as of the first release of Reader XI, require importing PDF files.*

Users need to save photo image files as PDF documents. Programs like Photoshop, Photoshop Elements and various authoring programs support exporting to PDF. Users not having image editing programs may need to import a photo taken with a cell phone or a camera in an authoring program and export to PDF.

Creating Buttons

To begin you need to create a button field. The sample form used here is a college application form. The form requires a photo affixed to the form so we can add a button field where the Reader user can import a photo.

Figure 19.1 College Application Form



This college application form requires the applicant to submit a photo.

In the Options tab in the Button Field Properties choose Appearance and set appearances for Border and Fill. You may want a keyline border around the image and you may want the background surrounding the photo set to white. If you want the background to be transparent use None for the fill.

After setting appearances, click the Options tab and choose Icon only from the Layout drop-down menu. At this point you don't need to import an image in the button field.

Click the Advanced button in the Options tab and the Icon Placement window opens. Check the Fit to Bounds check box and click OK.

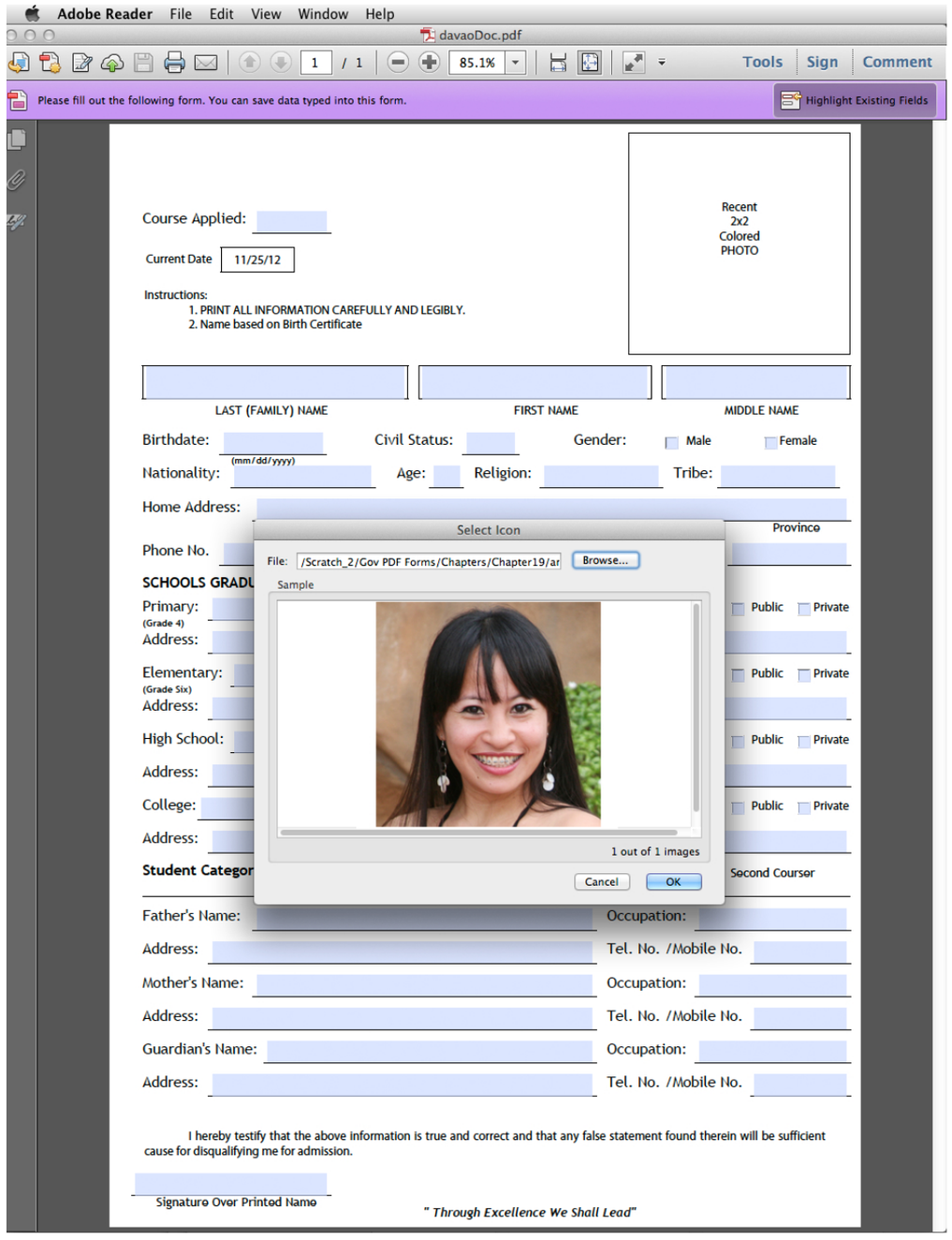
Writing the JavaScript

Click the Actions tab in the Button Field Properties window and choose Run a JavaScript for the Action type. Click the Add button and type the following in the JavaScript window:

```
event.target.buttonImportIcon();
```

This single line of code is all you need for the Reader user to click the button and open a dialog box where the user can browse a hard drive and locate a PDF file to import. Once the file is found, the user clicks OK and the PDF is imported as a button face.

Figure 19.2 Importing a Photo in Adobe Reader

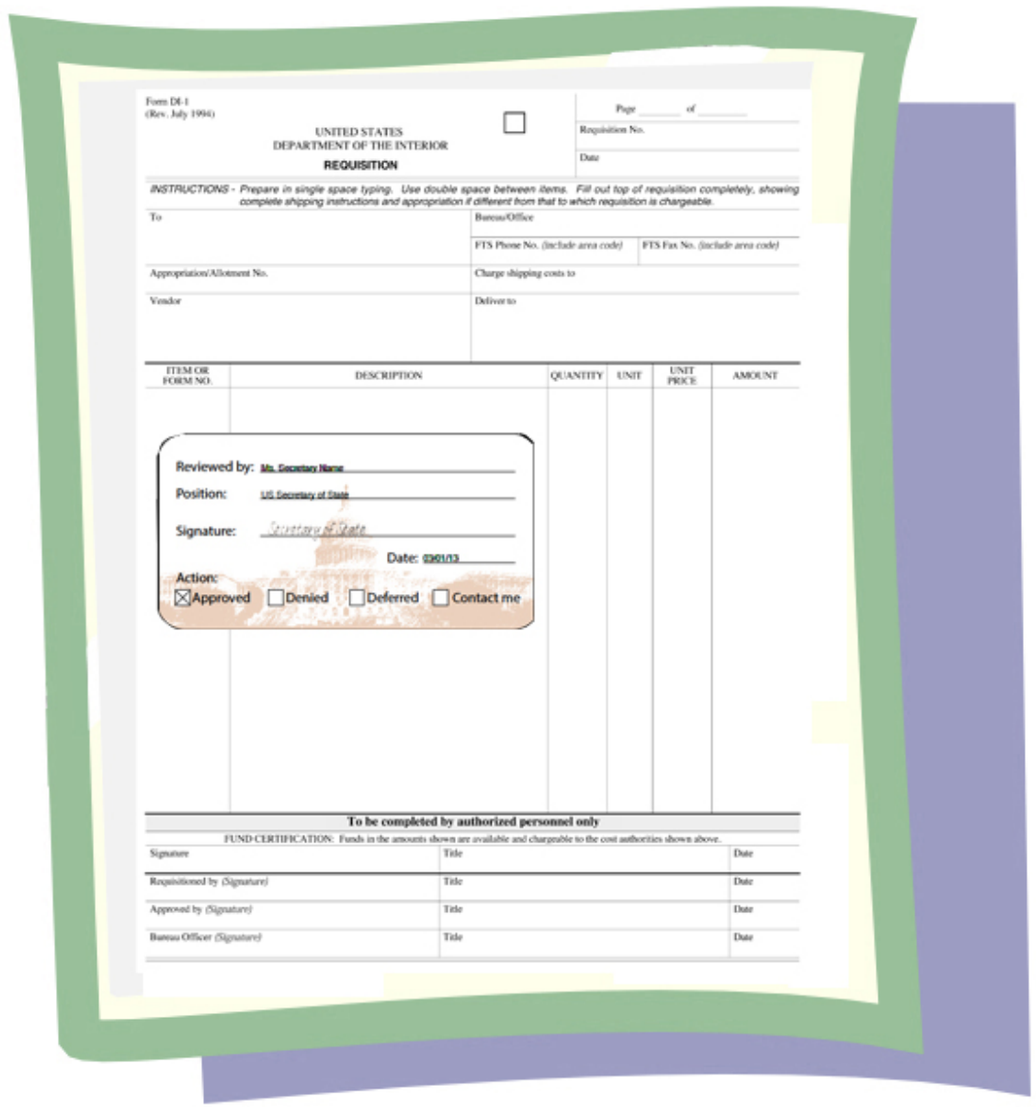


Click the button field in Adobe Reader and click Browse in the Select Icon dialog box. Navigate to the photo you want to import, select the photo, and click OK to add the photo as a button face.

Chapter 20

Working with In-Office Forms

Designing forms for in-house use



Chapter 20

Government offices work with two types of forms —those forms that are hosted on eGovernment websites for the public and those forms used internally within offices by employees. Just about any kind of office whether it be government or enterprise with 5 or more employees is likely to distribute forms and collect populated forms from the local staff.

An abundance of forms within offices are often found in HR departments where you find travel reimbursement forms, request for leave forms, employee evaluation forms, grievance forms, insurance forms, and many more.

In-house operations are actions the host performs when collecting distributed forms from office personnel or processing forms from constituents. You may find Official Use Only sections on forms that are intended for local agency use where approvals are granted, administrative signatures are applied, issuance and expiration dates are specified, identifying information related to authorized personnel reviewing the form, and more.

IN THIS CHAPTER

In this chapter we look at forms where personnel within offices need to interact with the forms either collected from the constituency or used exclusively by internal staff.

Using PDF Stamps

If you create a PDF workflow in your office for approving forms and documents one of the best approval tools you have in Acrobat and Adobe Reader are PDF Stamps. PDF Stamps are similar to analog stamps used for approving documents, marking sections on a form where signatures and initials are needed, marking documents for Draft, Final, Completed, Confidential, Void, For Public Release, and more.

PDF Stamps can be custom designed and they can contain fields with JavaScripts. You can create custom dynamic stamps with signatures for individual users and route documents electronically showing who has approved the documents before they are deployed. Documents can be certified to protect the PDFs from being altered after a user has reviewed and commented on the document.

There's so much to working with PDF Stamps that Acrobat JavaScript guru Thom Parker wrote a book that exclusively covers working with PDF Stamps. I can't begin to get into all the details related to stamps here however we can look at some simple, but worthwhile solutions.

PDF Stamps Essentials

There are two kinds of PDF Stamps —static stamps and dynamic stamps. A static stamp is a graphic, text, or both graphic and text created in an authoring application and converted to PDF. The PDF is then imported using the PDF Stamps menu in the Comments panel as a new stamp. You can determine a category and a stamp name when you import a PDF and create a new stamp.

When a static stamp is used, no aspect of the stamp is changed. The stamp merely has an appearance and used to support a comment note or simply reside on a page displaying the stamp design.

Dynamic stamps are created the same as static stamps and imported as a stamp using the same menu options in the Stamp drop-down menu. However after the stamp is created, and before the stamp is imported in the Stamp menu, dynamic features such as a date, identity information, and some other JavaScript actions can be added. Dynamic stamps can be made available to staff within a department and when stamps are placed on documents the dynamic features are derived from each user's Acrobat/ Reader configuration. For example, when you open the Preferences window and click Identity in the left pane, you can add an Identity name and title. This information can be imported in a dynamic stamp.

Stamps can also be secured or certified using a variety of security options. You can authenticate signatures from users to insure reliability of stamps.

Creating advanced dynamic stamps and adding security requires some additional work and training among members in a workgroup. Without delving into complex tasks, I'll keep it simple in this chapter and illustrate some examples of static stamps and easy to create dynamic stamps. For more advanced work with stamps and security look at Thom Parker's PDFScripting website at www.pdfscripting.com.

Creating a Static PDF Stamp

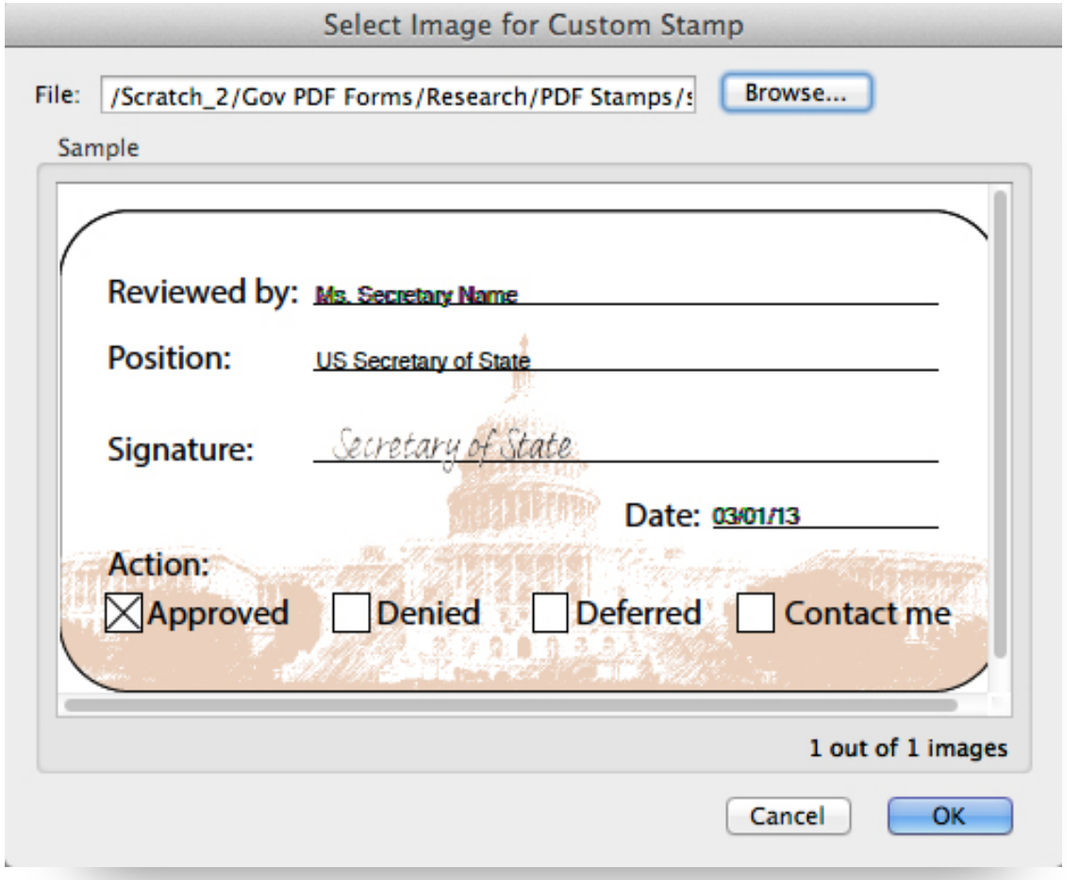
Creating a static PDF Stamp is much easier than creating a dynamic stamp. If working with stamps and JavaScript is new to you, you may want to begin using static stamps in your workflow. If you want to use a graphic image for approving form drafts, and further wish to use the same graphic for disapproving documents, marking documents as draft or ready for deployment and other conditions, you can create separate stamps for each notation you wish to add to forms and other documents.

To create a static stamp, do the following

1. **Create the stamp design**.
   Use an authoring application such as Adobe Illustrator, Adobe InDesign, MS Word or the application you use for assembling text and graphics. You design the stamp in an authoring program and convert the design to PDF.

2. **Create a new Static PDF Stamp**.
   Open a document (any document) in Acrobat or Adobe Reader. Click the Comments panel and open the Annotations panel. From the Stamp drop-down menu choose Custom Stamps ➤ Create Custom Stamps.

Figure 20.1 Select Image for Custom Stamp



After choosing Create Custom Stamp, the Select Image for Custom Stamp dialog box opens. Click Browse and locate the stamp design you want to import.

The Select Image for Custom Stamp dialog box opens (See Figure 20.1).

3. **Import the Stamp Design.**
   In the Select Image for Custom Stamp dialog box click the Browse button. In the Browse/Open dialog box locate the PDF document and click Select. Click OK in the Select Image for Custom Stamp dialog box and the Create Custom Stamp dialog box opens.

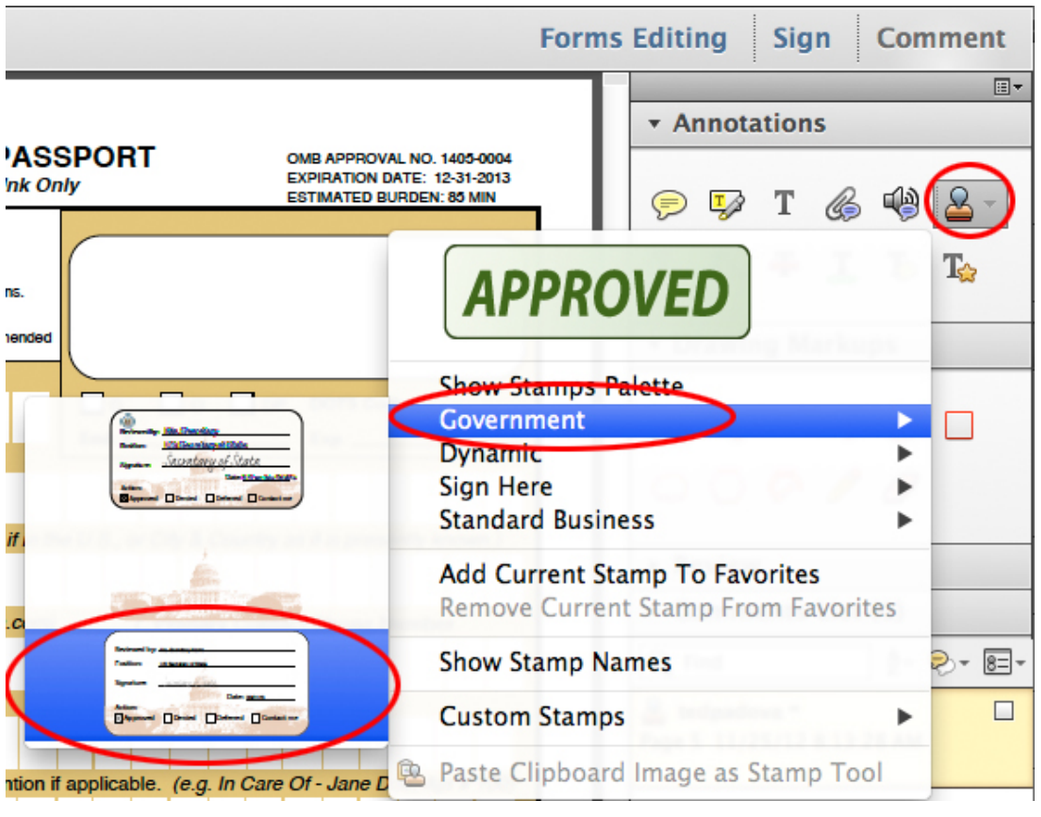
4. **Select a stamp Category and name the stamp.**
   In the Create Custom Stamp dialog box open the Category drop-down menu and select the category where you want to place the stamp. If you want to create a new category, type a category name in the Category text box. Type a name for the stamp in the Name text box and click OK. Your new stamp is added to the Stamps menu listed in the category you identified for the stamp.

Figure 20.2 Create Custom Stamp Dialog

choose a Category and type a name for the new custom stamp.

To use a stamp on a document you open the Stamp menu, click the category name and click the stamp you want to use. The stamp is loaded in the cursor. Move the cursor to a location on a form and click to place the stamp. If you want to add a comment note, double click the stamp after placing it on a page and a comment note window opens. Type text in the window as desired.

Figure 20.3 Open the Stamp Menu



Click the down arrow adjacent to the Stamp tool and choose a category, then click the stamp within the category to load the stamp.

Note

Stamps can be placed on secure PDF files if the permissions include Commenting, filling in form fields, and signing using existing signature fields. If the form grants permission for only Filling in form fields and signing using existing signature fields, users cannot add stamps or other notations on the form. For more on securing PDF forms see Chapter 14.

Creating a Dynamic PDF Stamp

Dynamic stamps can be configured using a lot of bells and whistles that require some extensive knowledge of JavaScript. For simple tasks you can easily copy/paste fields with JavaScripts without writing any code. In this section I'll cover some basic steps in creating dynamic steps. For more advanced dynamic stamp development see Thom Parker's www.pdfscripting.com website.

The first thing to know about creating dynamic stamps is that you first begin your development with a static stamp. You follow the steps outlined in the previous section for first creating a static stamp and importing the stamp with two exceptions. When you convert your design to PDF, open the PDF and press CTRL/Command + D to open the Document Properties. Click the Description tab and type a title for the stamp. What you add to the Title document metadata becomes the stamp name when you add the PDF to the Stamps menu.

The second exception is adding field objects before you import the PDF in the stamps menu. You cannot add field objects after the fact. When creating dynamic stamps, fields and JavaScripts need to be added before the document becomes a stamp.

The next thing to understand about dynamic stamps is that the dynamic nature of the stamp is not executed until the stamp is added to a PDF page. Any JavaScripts included with the stamp won't be executed until you add a stamp to your document.

To use fields with JavaScripts that you can copy and later paste in your stamp design you need to locate the Dynamic stamp file that was installed with your Acrobat installation. Stamp files reside in two locations on your computer. In an application user location and inside the Acrobat folder. The easiest way to locate the dynamic stamps file is by using a short JavaScript routine.

Open the JavaScript console (CTRL/Command + J) in Acrobat. (For more on using the JavaScript Console see Chapter 17). In the Console window you can type:

```
app.getPath ("app", "stamps")
```

Select the text in the JavaScript Console and press Enter (num pad key) and the above script returns the directory path for the Acrobat folder location where the stamp files are stored.

Or you can locate the Stamps folder in the user data folder by typing the following in the JavaScript Console:

```
app.getPath ("user", "stamps")
```

Likewise select the text in the JavaScript Console and press Enter (num pad key) and the above script returns the directory path for the User folder location where the stamp files are stored.

In terms of reliability, the Application folder is more reliable. On the Macintosh you have to do a little more work to find the Stamps folder in the Acrobat folder. You need open your Applications folder, then open the Adobe Acrobat XI Pro folder. Click Adobe Acrobat Pro and open a context menu. From the menu options choose Show Package Contents. The directory path in the JavaScript Console reports the location inside the Package Contents folder.

To copy/paste fields and make your stamp a dynamic stamp, do the following:

1. **Create a static stamp design and save as PDF.**
   Open the Properties window (CTRL/Command + D) and Type a title in the Description tab. Leave the file open in Acrobat.

2. **Open the Dynamic.pdf file installed by Acrobat.**
   Use the JavaScript Console to locate the Dynamic.pdf file.

3. **Copy the field on the Revised Stamp.**
   When you open the Dynamic.pdf file the first page is blank. Scroll to the second page and you find the Revised stamp. Press R on your keyboard to activate the Select Object tool. Click the Revised text box and press CTRL/Command + C to copy the field. This field contains a JavaScript in the Calculate tab that retrieves the current date automatically. Each time a document is stamped the date the file was stamped appears in the text box.

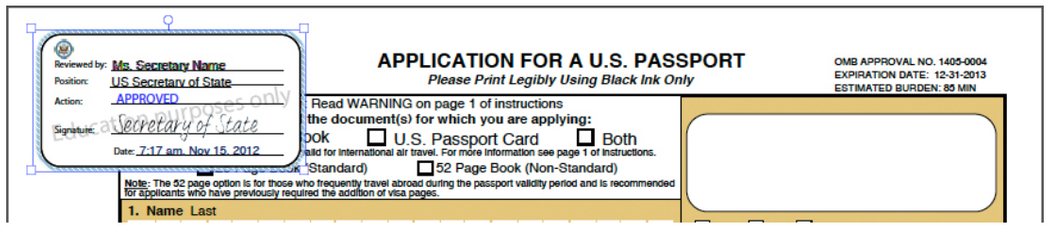
4. **Paste the field in your stamp design.**
   Bring your design forward in the Acrobat Document Pane and press CTRL/Command + V. Size and position the field to the appropriate area on the design.

5. **Save the file and add it as a new custom Stamp.**

There are other fields you can copy/paste in stamp designs. When you open the Acrobat or Reader Preferences window and click Identity, the user's login name is derived from the login that was set up when the operating system was installed. If computers in a department office are custom installed for users you can copy fields in the Dynamic.pdf file where the user login is added to a stamp.

Acrobat has some limitations with the Identity information. It has never worked properly for adding JavaScripts like `event.value = identity.name`. This script should work properly and retrieve

Figure 20.4 Dynamic Stamp



The dynamic stamp uses Identity data and a date field to automatically populate the first two fields and the date field at the bottom of the stamp.

the Name field in the Identity Preferences. Likewise you should be able to use the Identity Title. However, this doesn't work well in Acrobat either. There are workarounds using different scripts and adding Trusted Functions, but that's a much more complex issue.

Working with Application Response Dialog Boxes

On many government forms you find sections restricted for official use only. It's easy to conclude that many forms hosted on government websites are designed for analog routing. The reason being that most forms you download do not contain fillable fields for office users.

If fields are contained for office workers and the fields are editable for end users, you run the risk of retrieving forms with data added by the end users. If the form is signed or certified, this presents a problem. Signed forms should not be edited after a user populates a form. If the form is certified, editing the form nullifies the certification.

The best way to handle fields used for official use or by authorized personnel is to hide the fields. If the forms are secured, the end user isn't able to interact with the office use fields.

Another consideration that you need to entertain is in regard to digital signatures. In Chapter 13 I talk about digital signatures for the constituency and we look at using simple signatures for users to sign forms. In a more controlled environment such as within an office, you might consider using Digital Signature fields. You can secure documents and authenticate signatures within your office.

If you choose to use digital signatures among office personnel you need to provide some instruction for those who retrieve and sign forms. Digital signatures can be applied using Adobe Reader but creating signatures and distributing public keys is a little complex. You'll need to offer some training for personnel to adequately handle working with signatures.

That said, let's look at adding fields on a form for authorized personnel use only and making the fields accessible to only personnel who interact with the forms.

1. **Open a form where a section exists for official use or authorized personnel.**
   In this example I use a form from the US Department of Interior. The original form had data fields for the Title and Date columns in the table at the bottom of the page but there were

no signature fields. The Title and Date fields were open so users could interact with these fields.

2. **Add form fields to the section.**
   On the sample form I deleted the fields in the table for the authorized personnel only area. Fields on this form were not named with hierarchical names so I needed to start over with new fields using appropriate field naming conventions.

   The new fields I created in this section have a common root name of *oAuthority*. Using a common root name enables us to easily show/hide fields (See Figure 20.5).

   Before populating the table, I created the first row fields and used a digital signature field for the signature column. After creating the first Digital Signature field I added a JavaScript.

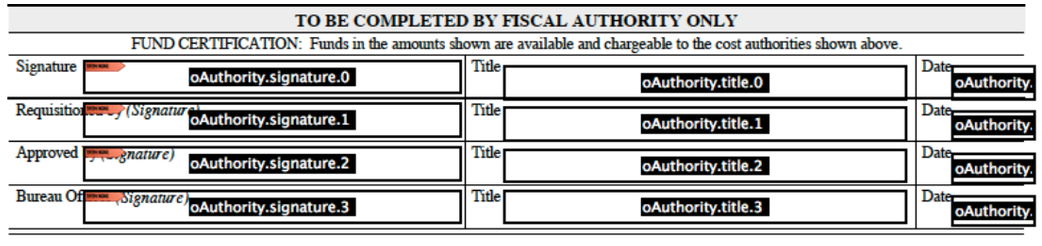
3. **Add a JavaScript to the Signature Field.**
   Open the Digital Signature Properties and click the Signed tab. In the Signed tab click *This script executes when field is signed*. Assuming this form is routed to several individuals, it might be best to keep fields hidden until they need to be signed. Therefore the following script was added in the JavaScript Window:

```
1.    var f = this.getField("oAuthority");
2.    f.hidden = true;
```

   This script hides the fields with the common root of *oAuthority* after the document is signed.

4. **Create the table.**
   After the first row of fields are created and the JavaScript is added to the first Digital Signature field, open a context menu and choose Create Multiple Copies. Distribute and position the fields as explained in Chapter 11.

Figure 20.5 Table Created at Bottom of Form



To make it easy to write a script that shows/hides fields, the fields have a common root name of oAuthority.

5. **Hide the fields.**
   Drag through the new fields with the Select Object tool and open a context menu. Choose Properties to open the Properties Window. Note that some properties can be adjusted when both Digital Signature and Text fields are selected.

6. **Click the General tab and from the Form Field drop-down menu select Hidden.**

At this point the table for authorized use only is populated and the field attributes are properly assigned. If the staff signing the form all have Acrobat the users can open the field properties and show

the hidden fields when the form is returned. To do so, the staff would need the password that was used to secure the form and have an understanding for working with field objects in Acrobat.

To simplify matters, we can add a button that we can password protect. When a user clicks the button, the correct password needs to be typed. When the password is correct, the fields are made visible. If the password is not correct, the fields remain hidden.

Here's how to do it:

1. **Continuing with the sample form, add a button field.**
   In this example I made a button with a label that reads *To be completed by authorized personnel only*. The button appears transparent to the user however if a user clicks the button we need to add an application alert in the JavaScript that informs the user that authorized personnel only use the fields.

2. **Define the Button Properties.**
   Open the Button Properties window and set the border and fill color to none. Click the Options tab and add a label. (See Chapter 12 for more on working with button properties).

3. **Add a JavaScript.**
   Click the Actions tab and choose Run a JavaScript from the Select Action drop-down menu. Click Add and the JavaScript Window opens. Type the following script in the JavaScript Window:

```
1. var t = this.getField("oAuthority");
// target field
2. var cResponse = app.response({
3.    cQuestion: "Enter password?",
4.    cTitle: "Password Request"});
5. if ( cResponse != "password"){ // if the
password is not entered properly
6.    app.alert("The fields below are for
authorized personnel only");
7. else
8.    t.hidden = false;
```

4. **Click OK in the JavaScript window and close the Button Properties window.**

The first line of code assigns the field *oAuthority* to the variable *t*. Since the table fields all begin with the root name oAuthority it's easy to show/hide these fields with a few lines of JavaScript. Note the // in this line and in line 5. Text following the // is a comment or programmer notation and ignored by Acrobat.

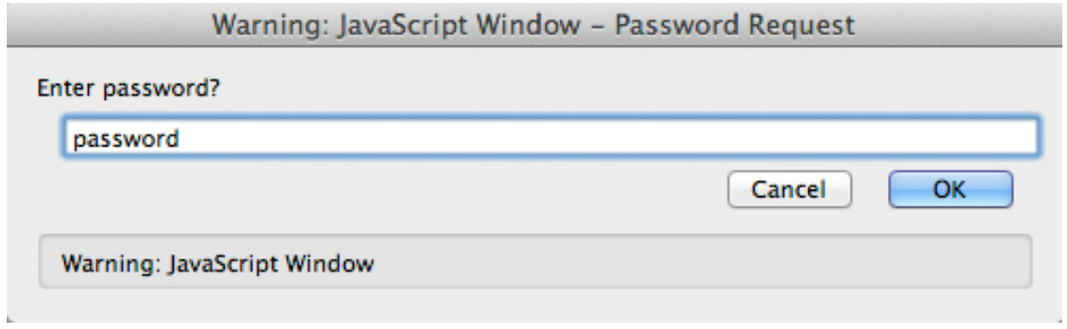
The second line of code sets the variable *cResponse* to the *app.response object*. Lines 3 and 4 relate to information displayed when the app.response window opens. A question appears and a title for the dialog box.

In line 5 we specify the password. For this example I simply use *password*. If you use this routine in your forms you'll want to change "*password*" to the password you want to use. Be certain to use more than 8 characters and mix up the characters with numbers and/or symbols. But make it easy enough for your staff to type in the response dialog box. The if statement in line 5 checks the response. If the password typed does not precisely match the password identified between the quote marks, an alert dialog box opens and displays the message in line 6.

If the password is correct the *oAuthority* fields are not hidden as show in line 7. Since we added a script in the signature fields to hide the oAuthority fields after a signature is added to the form, each authority needs to use the same password to make the fields visible.

When the form is completed, the person in charge of filing the form can click the button to show fields, save the form, and all fields remain visible.

Figure 20.6 Application Response Dialog



When a user clicks the button, a dialog box opens prompting the user for a password.

Sending Data to Secondary Forms

If you design forms and host the forms for users where data in one form needs to be sent to another form, you must make the user aware that both forms need to reside in the same folder location on a hard drive. If forms are in different folders, Reader cannot open the secondary form. It may be rare that you would use such forms for constituents but you can find many uses for intra-agency staff.

In your office you may collect forms from constituents and use summary forms to aggregate data. You might use summary forms to collect a week, a month or a quarter worth of data. Then export the summary form that ultimately gets introduced into your database system. If you don't use a server to collect and route data, this method can be effective to handle data without purchasing server systems —especially for small government offices where the costs for servers are prohibitive.

There are several things you need to know about setting up documents where you want data sent from one form to another. The first thing you need to do is write a Document Level JavaScript. At the document level add the following on both forms:

```
this.disclosed = true;
```

If you write the script for a button to send data from one form to another you begin by adding the following:

```
1. this.slave =
   app.openDoc("document.pdf",this);

2. this.bringToFront();
```

The above script looks for the secondary PDF form (in this example *document.pdf*). Again, this file must reside in the folder where you have the form with a button that contains the script.

Code that follows identifies the fields in the current document and exports field data to the slave document. The script might look like:

```
1. //medical record number

2. var record =
   this.slave.getField("s.medRecord.0");

3. var j = this.getField("c.medRecord.0");

4. record.value = j.value ;
```

In the first line I use a comment so I know what field is retrieved from the current form and sent to the slave document. In this case a medical record number is used.

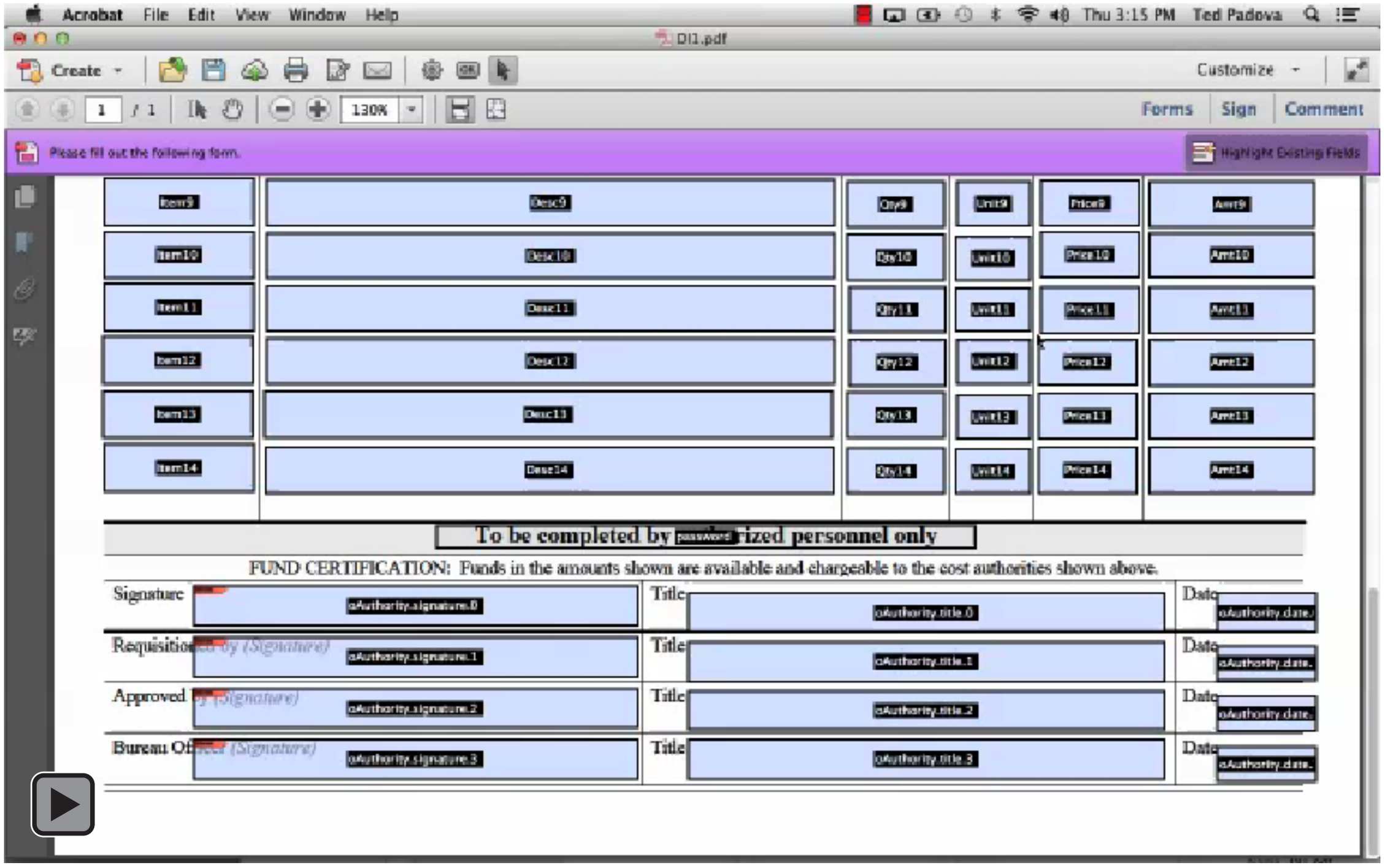
In line 2 I assign the field "*s.medRecord.0*" to the variable *record*. The root name is *s* to make it easy for discerning the differences for the slave document (s) and the current document (c).

In line 3 we assign the "*c.medRecord.0*" field to the variable *j*. In line 4 we take the *j.value* which is the current data and make it equal to the *record.value* in the slave document.

You continue adding the same 4 lines of code while changing field names for additional fields where you want to send data from one form to another. It's helpful to add a comment line for every field you address. When it comes time to debug the script it will be much easier to correct the code.

If you host forms for constituents, this routine can run effectively in Adobe Reader.

Movie


Acrobat File Edit View Window Help
Thu 3:15 PM Ted Padova
DI1.pdf
Create
Customize
1 / 1
130%
Forms Sign Comment
Please fill out the following form.
Highlight Existing Fields
Item9 Desc9 Qty9 Unit9 Price9 Amt9
Item10 Desc10 Qty10 Unit10 Price10 Amt10
Item11 Desc11 Qty11 Unit11 Price11 Amt11
Item12 Desc12 Qty12 Unit12 Price12 Amt12
Item13 Desc13 Qty13 Unit13 Price13 Amt13
Item14 Desc14 Qty14 Unit14 Price14 Amt14
To be completed by authorized personnel only
FUND CERTIFICATION: Funds in the amounts shown are available and chargeable to the cost authorities shown above.
Signature
aAuthority.signature.0
Title
aAuthority.title.0
Date
aAuthority.date.
Requisitioned by (Signature)
aAuthority.signature.1
Title
aAuthority.title.1
Date
aAuthority.date.
Approved by (Signature)
aAuthority.signature.2
Title
aAuthority.title.2
Date
aAuthority.date.
Bureau Official (Signature)
aAuthority.signature.3
Title
aAuthority.title.3
Date
aAuthority.date.

Chapter 21

Making Forms Accessible

Forms deployed on government websites should all be made accessible for persons with vision and motion challenges.

Chapter 21

USA Federal and state forms need to comply with Section 504 of the US Rehabilitation Act that guarantees certain rights and liberties for persons with disabilities. The act was legislated in May 1977 and through the years more rights were granted to persons with disabilities. Today, disability law has become an international human rights issue with the United Nations, some regions in the world, and many individual countries. As a matter of practice all eForms in the USA and many other countries should be made accessible.

IN THIS CHAPTER

What are Accessible PDF Forms?

Accessible forms contain the structure and design that optimizes readability on screen reading devices for the vision and motion challenged users. Certain tags need to be defined in PDF forms to make the reading experience optimal for screen readers. Images, for example, need alternative text tags that describe the appearance of an image. Form fields need detailed tooltips that describe fields.

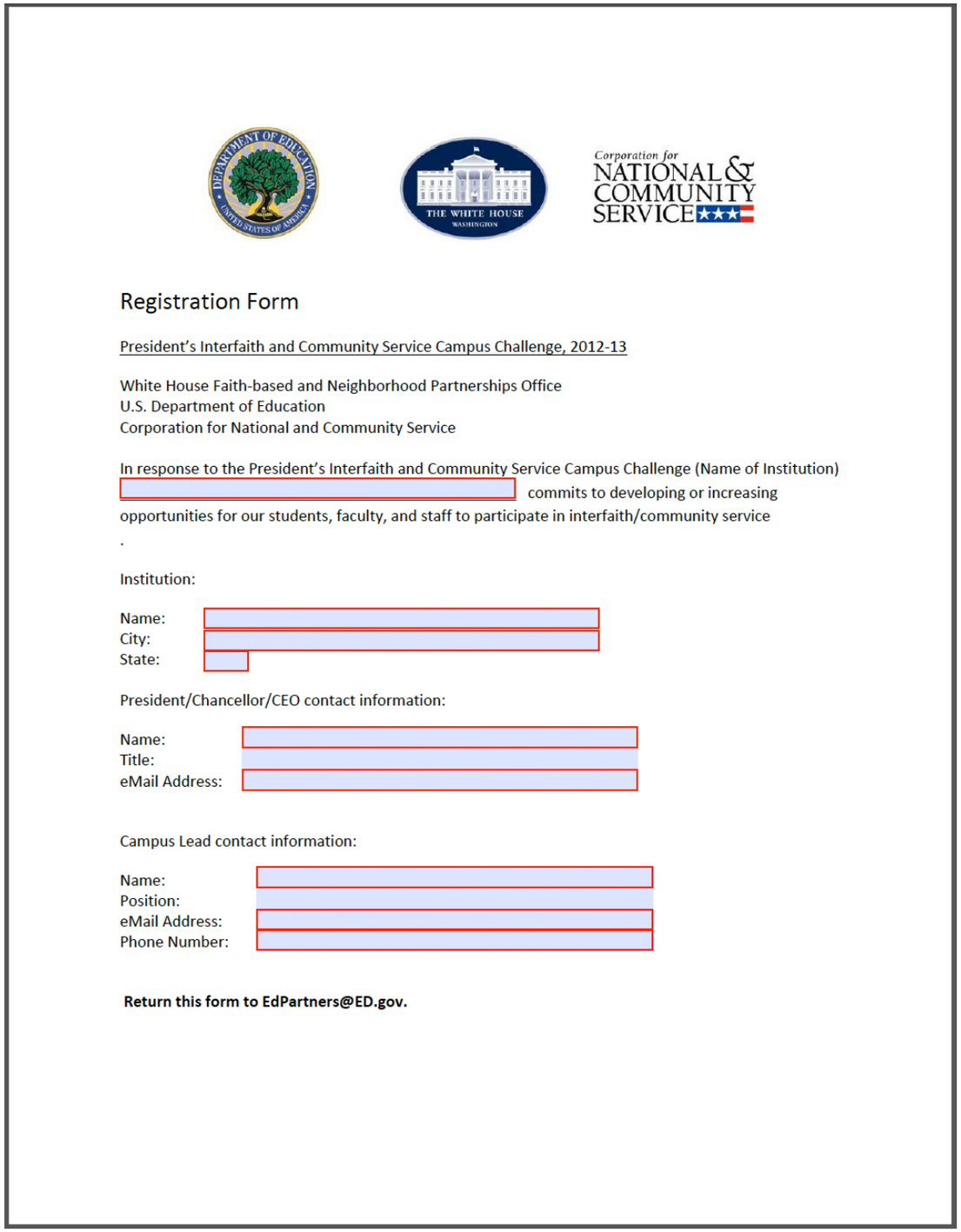
Designing for Accessibility

The sample form used for this example is a US Department of Education form. The form is constructed fairly well however there are a few problems that need to be corrected before making this form accessible.

The first item presenting a problem is the underscores below the first text field. When designing forms, stay away from using underscores to designate field positions. When this form is read aloud, each underscore character is read as a separate item. Therefore you hear underscore, underscore, underscore, etc. from the screen reader. The first task therefore is to delete the underscores before making this form accessible.

The second item that needs attention is the *mailto* link at the bottom of the form. When this form was originally designed, a link was created for the EdPartners@ED.gov text. The action set for the link was to open a web link. Clicking the link merely opens the

Figure 21.1 Sample Form



Form from US Department of Education

users mail client. The form is not attached when clicking the link. A better solution is to add a submit button or change the link action to a Submit Form Action.

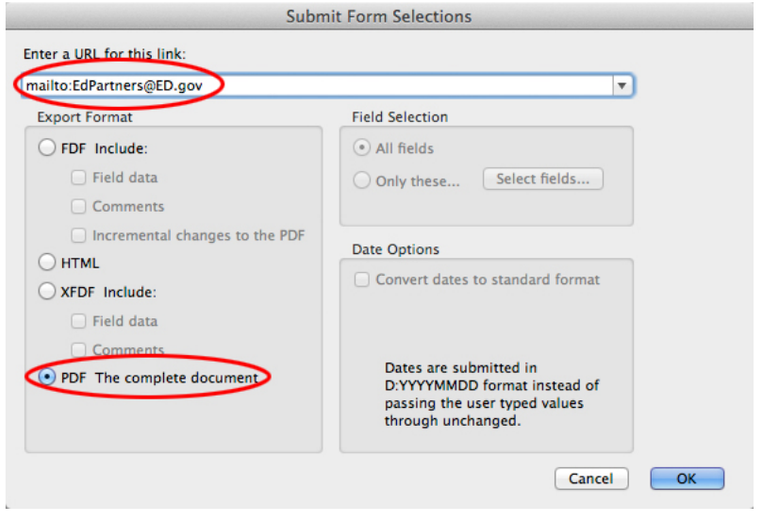
To add a submit button, do the following:

1. **Create a button field.**

2. **Adjust the field properties.**
   In this example you can simply use a label such as Submit Form

3. **Add a Submit Form Action.**
   Click the Actions tab and choose Submit a form. The Submit Form Selections window opens. At the top of the window enter the email address. In this example *mailto:EdPartners@ED.gov*. Click the PDF the complete document radio button.

Figure 21.2 Submit Form Selections



Add mailto: and the email address in the top line and check the PDF The Complete document radio button

When the user clicks the Submit Form button, the default mail client opens and the PDF document is automatically attached to a new mail message.

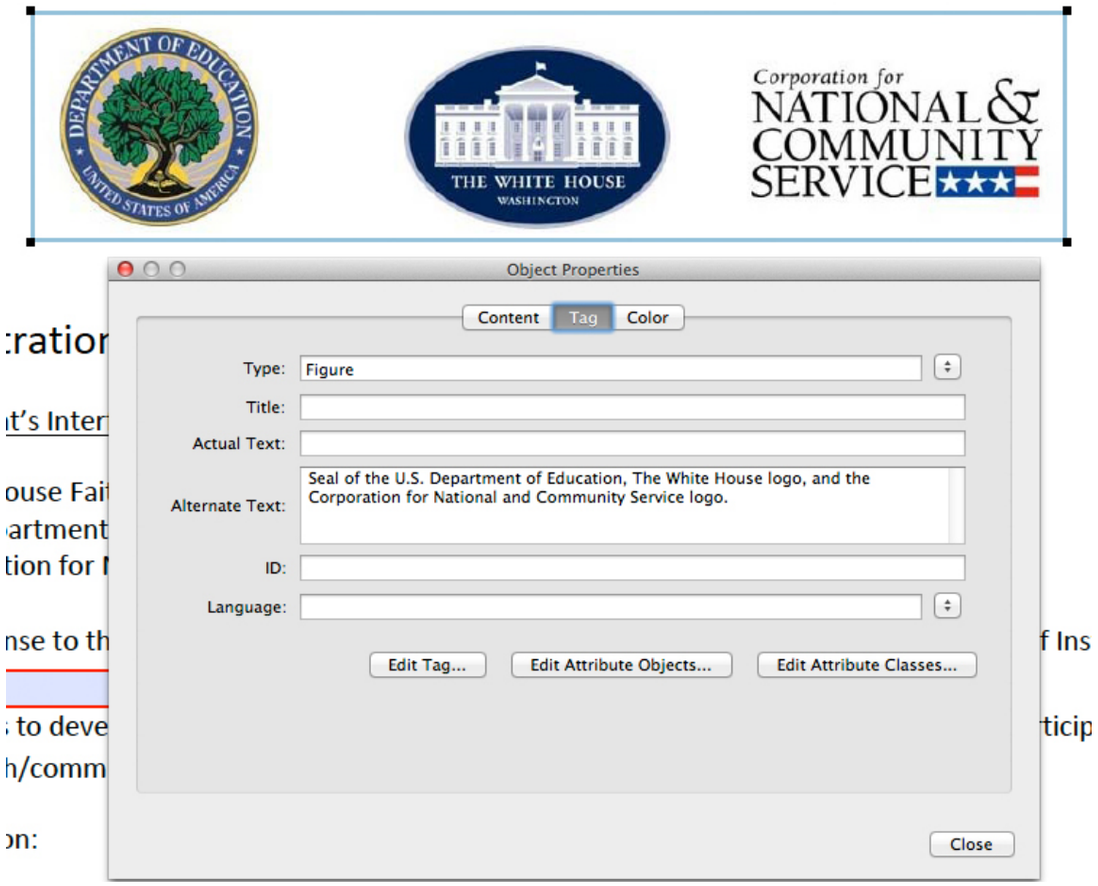
A third item that needs attention in the original form is the document metadata. This form had no metadata for the Title, Author, Subject, and Keywords fields. The accessibility checker picks this problem up after checking accessibility so we can handle adding document metadata later.

Before running the Accessibility Checker, you may want to review all images in a file. On the sample form I use here the alternate tags were properly added to the seals at the top of the form. You can add alt tags to images easily by right clicking a graphic with the Edit Object tool (found among the Print Production tools) and choose Properties from a context menu (Figure 21.3). When the Object Properties window opens click the Tag tab. Here you can type alternate text.

Checking for Accessibility

After you design a form and you're confident that the fields are designed properly, the fields have tooltips describing all the fields, and JavaScripts work properly, you're ready to check the form for accessibility. This is the first step in making a form accessible.

Figure 21.3 Image Tag Properties

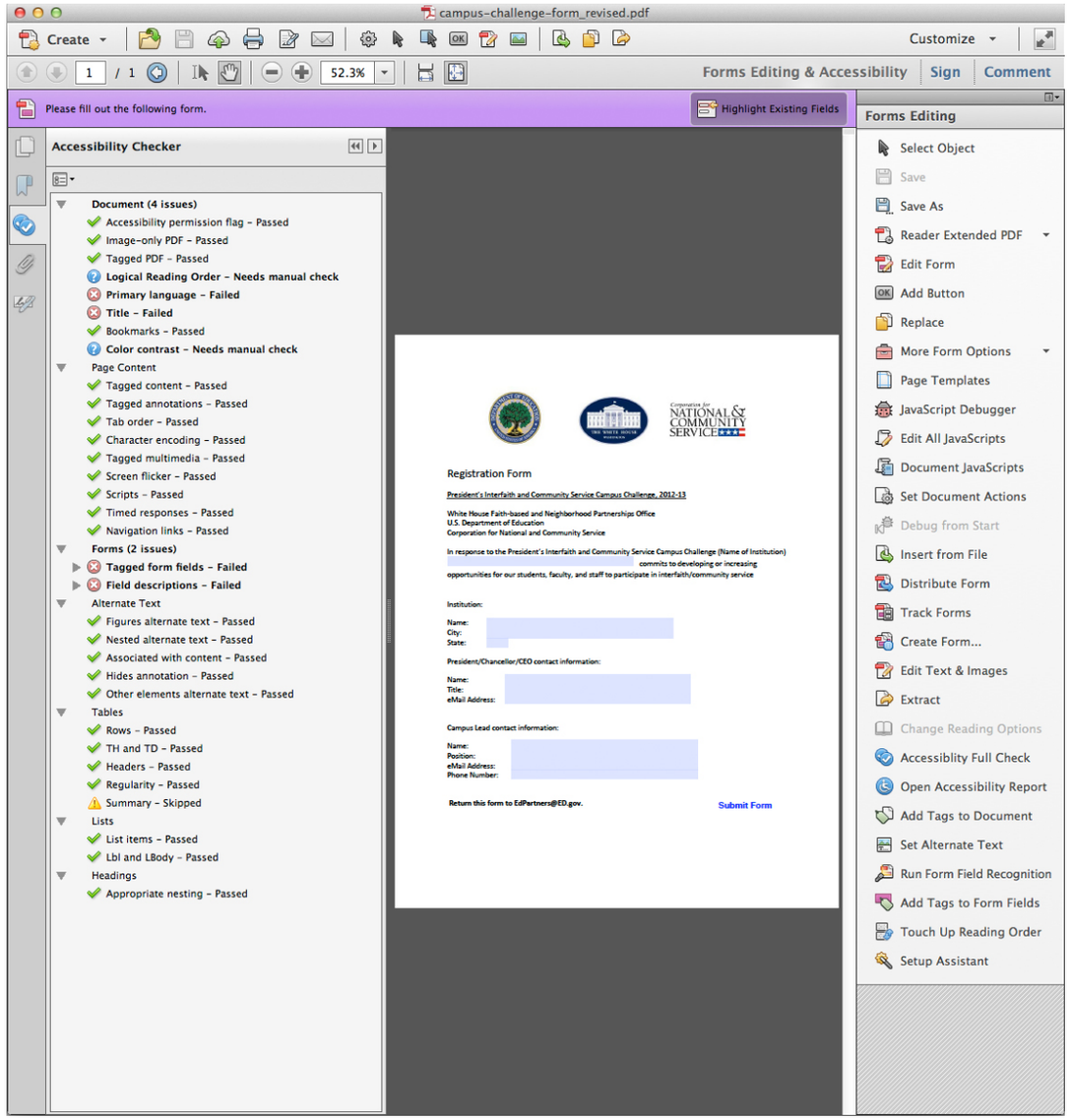


Type the tag information in the Alternate Text box.

At this time you may want to add the Accessibility tools to your Forms Editing toolset. Click the Customize button in the top right corner of the Acrobat window and choose Manage Tool Sets from the drop-down window. Click the Edit button and add all the Accessibility tools to your toolset.

With the tools in place, click the Accessibility Full Check tool in the Tools panel. The Accessibility Checker reports back errors in the Navigation pane (left side of the Acrobat window).

Figure 21.4 After Running the Accessibility Check



Errors are reported in the Accessibility Checker pane in the Navigation pane.

The items in the list report errors where you see a circle with an X. On many of the errors listed in the Accessibility Checker pane you can right click to open a context menu and choose Fix from the menu options. Where you see Title – Failed in the Accessibility Checker pane, open a context menu and choose Fix. The

Document Properties are displayed where you can add Title, Author, Subject, and Keywords.

Checking the Reading Order

When your form is read on a screen reader you want to be certain that the reading order is the same as the view you see in the Acrobat window. To check the Reading Order click the Touch Up Reading Order tool in the Tools panel. When the Touch Up Reading Order panel opens, click the Show Order Panel button. The Order pane opens in the Navigation pane.

Figure 21.5 Reading Order

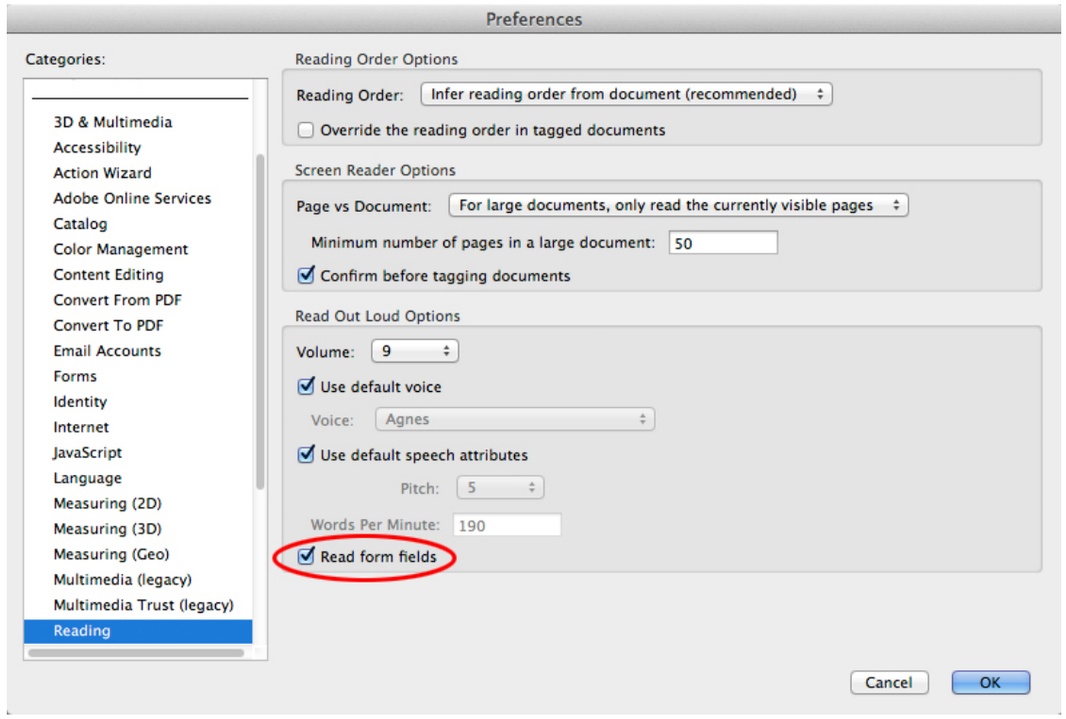
Click the Show Order Panel button to open the Order pane in the Navigation pane.

You can click and drag items in the Order pane to change the reading order.

Testing the Form with Read Aloud

The first step to take to test your file is open the Preferences (CTRL/Command + K). Click Reading in the left pane and check the Read Form Fields check box at the bottom of the Reading Preferences.

Figure 21.6 Reading Preferences



Check Read form fields before reading aloud.

To test the form as it might be read on a screen reader, choose View ➤ Read Out Loud ➤ Activate Read Out Loud. Return to the same submenu and choose Read This Page Only or Read to End of Document.

Test the overall reading of the document to be certain the reading order is correct and the graphics all have alt tags. After the document is read aloud, press the tab key. The cursor should jump to the first field and the tool tip is read out loud. Continue tabbing through all fields and check for accurate reading of the tool tips.

After you check the form using Acrobat's Read Out Loud feature, check the form on a screen reader before deploying the form.

Chapter 22

Hosting Forms on eGovernment Websites

Organizing forms and making them easily accessible to constituents

Chapter 21

Creating great forms means nothing if visitors to a government website can't find forms. Forms need to be accessed quickly and easily, they must be saved in a common format for consistency, and they must be easy to download.

IN THIS CHAPTER

Constituents should be no more than three clicks away from locating a form needed to process a transaction. Today, however, most government websites do not have easy and intuitive means for finding forms.

Ideally, the best solution for helping constituents find forms is placing a link on the top banner of the eGovernment website. Quite simply a button for Forms should be as visible as the Home button. If all government websites place a link to a forms page on the banner, visitors would soon begin to look at the banner for quick access to forms.

Most countries have national government websites, regional government office websites, and city websites. As citizens move around a country or foreigners migrate, they need to acquire forms for business licenses, school applications, taxpayer information, residency and immigration forms, and more.

Acquiring forms should involve the same navigation steps on all government websites within a country. It's very frustrating for users to locate forms from one website to another within the same country if they have to follow different navigation links. I contend that the website designs of national, regional, and LGUs should have common design elements. If we agree that processing transactions is one of the top three purposes of eGovernment, then eGovernment websites need consistency in organizing content.

Unfortunately, today website visitors searching for forms on one city or regional office find no consistency when searching through other LGUs within the same country. Little consistency exists between websites among national government offices, regional offices, and LGUs.

Organizing Forms

Government first needs to understand the difference between a form and brochures, instructions, pamphlets, etc. A form is a document used for processing a transaction. It should be made fillable. All other documents are not forms. When designing forms webpages logical divisions should be made between forms and other documents. A good many documents such as pamphlets, brochures, etc. have no business being listed on webpages listing forms.

There are so many eGovernment websites today hosting pages for forms while a good portion of the content doesn't resemble a form. On some government websites the visitor needs to sort through long lists of downloads to find forms integrated with everything from brochures, instructions, slide presentations, and many other documents —some related to forms and others that have no relevance for filing in an eForm.

There are several considerations that need to be entertained when developing uniform standards for developing and organizing forms:

1. **Create a Forms Standards Committee.**
   At the national level a committee should be assembled to develop a set of standards that require all government websites to follow within a country. Committee participation should include graphic design people as well as IT specialists. Leaving forms designs to IT departments can result in a horrendous set of form designs in terms of the forms structure and appearances.

2. **Develop standards and best practices.**
   The appearances of forms should be consistent and well designed. A book of standards should be developed for placement of text, graphics, and fields for all forms.

3. **Approval officers.**
   After forms are developed, individuals or committees should approve forms before they are deployed. Leaving forms approvals to individuals throughout a county leads to inconsistency and much disparity in form designs.

4. **Training forms authors.**
   PDF authors should be well trained and knowledgeable about forms construction, Acrobat JavaScript, and fundamentals of form design. Form authors should be knowledgeable about document metadata, accessibility, security, and data routing.

5. **Develop a forms organization schema.**
   All forms should be numbered with form numbers. Identifiers should be agreed upon in committee. A form might be identified with a national department abbreviation and number or an LGU number and department plus a number. The standards should be agreed upon in committee so all agency personal can look at a form number and immediately know where the form originated.

6. **Develop forms webpages.**
   From a Home page link, the user should be taken to a well-organized forms webpage. In committee, standards for forms keys and webpages should be developed and implemented throughout the country. A user should be able to immediately arrive at a forms page and find consistency in the design no matter what national office or LGU the user explores.

There may be other considerations that need to be explored. However, if a central standards committee is developed, additional matters can be discussed and standardized via committee resolutions.

Creating New Revenue Streams

Governments can assess the current workflows for how constituents are submitting forms and the costs to the constituents that currently exist. How much time and expense is required for constituents to submit a form? Does the constituent need to travel to a government office? How far, how much cost is associated with the travel?

Take something like submitting an application to attend a public institution of higher learning as an example. Suppose a constituent needs to travel from a rural location to a capitol city. In developing countries a prospective applicant might take a bus accompanied by a parent, spend money for lodging for one night and make a return trip.

Add up the costs now associated with submitting forms to institutions. Even within a city where time and public transportation is necessary for hand delivering a form and compare nominal costs to process a form submitted electronically.

Government can implement processing fees for most forms submitted electronically that actually saves the consumer money, time, and aggravation. The more people processing forms electronically the less burden is placed on government offices for personnel and facilities that need to accommodate long lines and waiting periods.

This is a win-win-win situation. Consumers win because less money is required to visit an office to process a form. Government wins because offices have a new revenue stream. Government wins again by using revenues to offset costs for implementing additional servers to collect and route data.

In developing countries Internet penetration can be as low as 6%. However, these countries typically have Internet cafés readily accessible in some of the most remote locations. Some of the most impoverished countries in the world can take advantage of electronic forms processing for many constituents.

Chapter 23

Using PDF Forms on Mobile Devices

The current state of supporting PDF forms on mobile devices

Chapter 22

IN THIS CHAPTER

In most first world countries we find Internet penetration over 70%. In third world countries we find Internet penetration less than 10%. Many third world countries have Internet penetration as low as 6%.

In terms of smart phones and Internet hot spots, third world and developing countries have much higher levels of Internet access. Many of the countries having penetration of 6% have over 28% access via mobile phones, tablets and hot spots.

Technology is continually growing to support needs of people using various devices throughout the world. One needs to view technology as evolutionary and something that eventually supports needs of most users. Inasmuch as support for filling in and submitting eForms is not readily available to the masses in many countries, we are in a rapid growth cycle. Eventually most people throughout the world will be able to take advantage of filling in forms and submitting them regardless of the device they use.

PDF Readers and eForms on Handheld Devices

There are a number of PDF readers today supported on iOS, Android, and Windows handheld devices. I hesitate to name any here because in the mobile market it's a moving target. What's available today can easily be replaced by a new product tomorrow. Suffice it to say, there now exists a number of PDF readers supported on every handheld and tablet.

Of the many PDF readers available on handhelds, only a few readers permit filling in a form. However, not all Acrobat features are currently supported on these devices. Various signature attributes are limited, importing button faces are not supported, spawning pages are not supported, and sending data to secondary forms is not supported.

Nevertheless, some forms can be completed and signed on handhelds and submitted to government offices.

Supporting Handheld Users

To support users on various devices government offices need to research and investigate limitations with devices and software to complete and submit forms. With rapid growth in technology, offices need to continually keep abreast of developments.

Inasmuch as support for some of the forms you create is not yet available, it will soon arrive. The time to start is now. Websites should have separate downloads made available for forms where preferred usage is a computer and where alternate usage can be handled with other devices. As technology supports more features among readers for handhelds, changes can be addressed on websites for changing form designs and providing constituents with information on what devices can be used to complete and submit forms.

Chapter 24

Exploring Some Sample Forms

Sample forms used in real world scenarios

Chapter 24

In my work with government offices, particularly in developing countries, I found many interesting needs that offices expressed for using PDF forms. In this chapter I explain some of those needs and look at some solutions.

IN THIS CHAPTER

Philippine Unified Form

The Philippine national office for the Department of Trade and Industry (DTI) wanted a form to help constituents apply for business licenses. The Philippines was ranked 165 in the world for the ease of doing business in 2010. The complicated applications for sole proprietorships and corporations took as much as 120 days to process a new business application.

Problem

The Department of Trade and Industry set up local offices to help facilitate the business application process but they needed a form for constituents to complete before visiting an office. No servers were in place and no means of collecting and routing data existed so applicants needed to print forms and personally deliver the forms at local offices.

The Unified Form was a collection of forms that were required by 7 different government agencies. Standard agency forms such as internal revenue, health, SEC, and others were required to complete a business application. At the beginning of the project all forms were PDF documents but none were fillable forms.

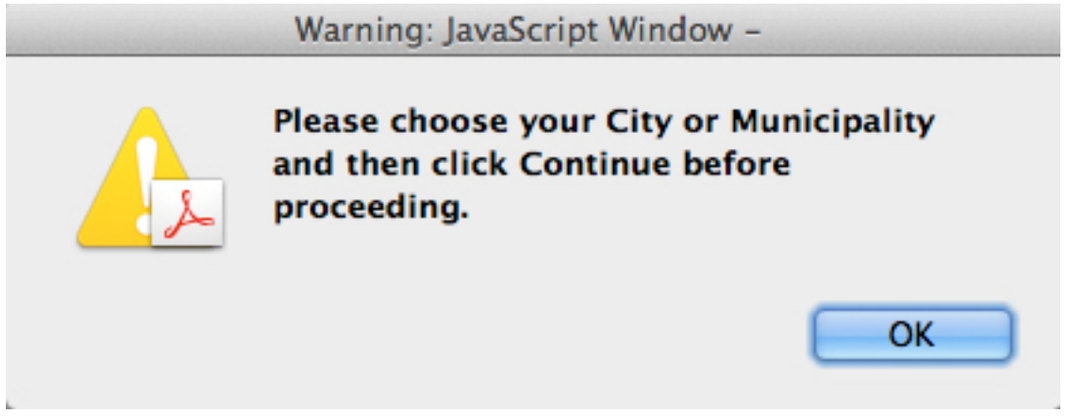
The task for this job was to create a single document containing the 7 different agency forms, make the forms fillable, and implement as many error checks as possible for improving user accuracy.

Designing the Unified Form

Each agency form required for a business application was included in the Unified Form. Rather than have the user fill in common data fields, it made sense to identify common fields on the the individual forms and design a master form at the beginning of the document. The user would populate the master form and data would appear on the common fields in other forms.

The form was designed for use throughout the country in fifteen regions. Some regions had more than 100 LGUs. Each form needed to identify a given LGU at the top of the form. To add an identifier, a document level JavaScript opened an alert dialog box prompting a user to select a given local area from a drop-down menu when the form opened.

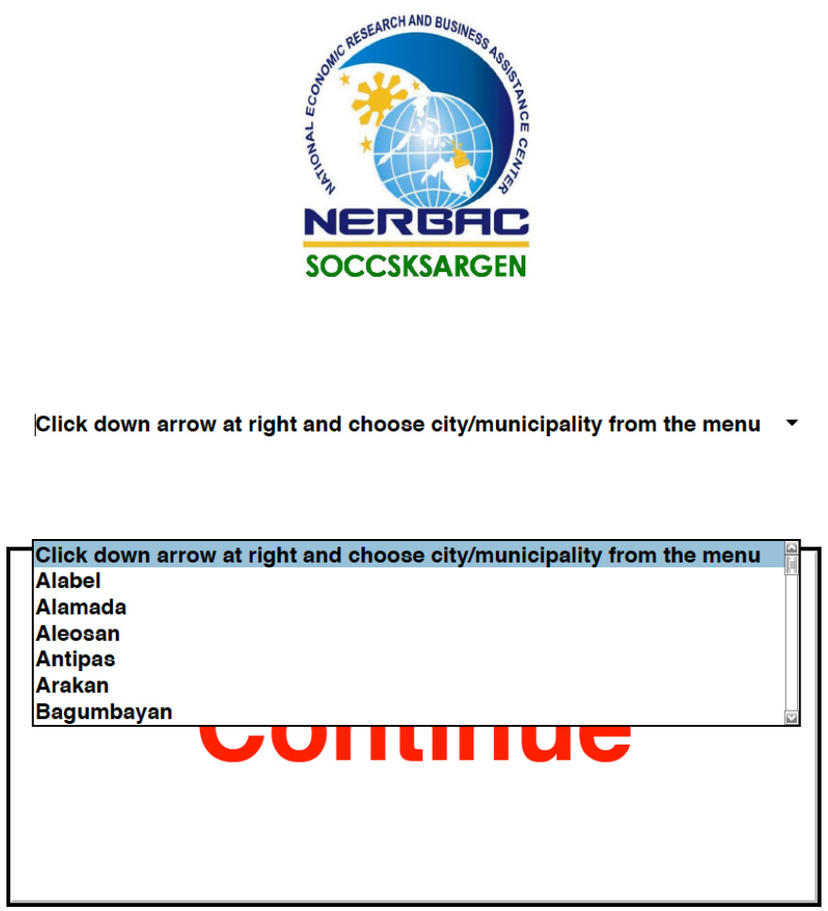
Figure 24.1 Open Alert Dialog



Informs the user to make a selection in a drop-down menu

After the user makes a selection in the drop-down menu for identifying a given municipality a button is made visible. When the user clicks the button, the selection from the drop-down menu is sent to all forms in the document. This insures easy identification for the area where the form originated and makes the form useable in every LGU in the country.

Figure 24.2 Drop-down Menu for Selecting a Municipality



Before a user can proceed a selection from the menu must be made for identifying a given municipality.

A field such as taxpayer ID was required on all forms. Therefore when the user populated the taxpayer ID on the master form the taxpayer ID fields on the other forms were automatically populated. Common fields such as business name, address, proprietor name, address, etc., were locked on all forms except the master form. Therefore the user needs to return to the master to make corrections. When a correction is made the forms with the common data are updated. This method insures the user and the government agency collecting the form that data are consistent throughout the form.

Figure 24.3 Master Form Page

Common data are sent from this form to the remaining forms in the document.

Fields on the master form have different field names than the common fields on the other forms. Therefore data needs to be sent from the master form to the other forms for the common fields. This is easy enough with a simple JavaScript. However, the script is not in the Calculate properties as you might imagine. For this task I add the script to the Validate tab in the Text Field Properties window.

The script to send data from the current field to target fields was written as:

```
this.getField("global.bus.name").value =
event.value;
```

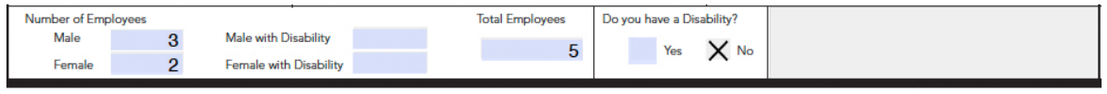
This script simply gets the target field name and whatever is typed in the current field (the *event value*) is sent to the target field.

A script like this helps users eliminate the need for typing redundant data. If a name is required on one page and repeated on several other pages, why ask the user to type a name several times? You can help the users ease through your forms by implementing shortcuts like this.

Additionally it helps prevent user error. If fields are locked using *read only*, the user needs to return to the page that accepts data input. If an error is found on the data input page, one change insures that data on all pages are changed accordingly for the common fields.

On page two in the Unified Form a question exists for adding the total number of employees.

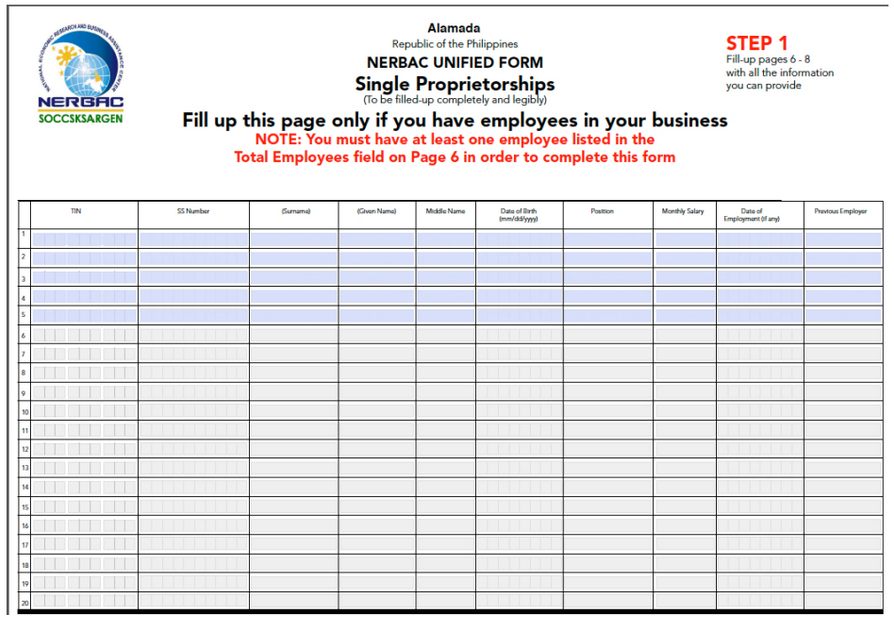
Figure 24.4 Number of Employees



User identifies the number of employees

Following page two is a page where employee information is supplied. By default the fields are hidden. When the user supplies data for total employees, the fields on the following page are made visible. But only the number of rows that match the total number of employees are made visible. This is another measure that insures proper completion of the form. A user cannot claim 5 employees and fill in data for less than or more than five.

Figure 24.5 Identify SSS Number for Employees



The fillable rows are available for only the total number of employees.

The Java Script that shows the number of rows respective to the total employees is written as follows:

```
1. var w =
   this.getField("sole.bus.male.employees");

2. var y =
   this.getField("sole.bus.female.employees");

3. event.value = w.value + y.value;

4. var j = event.value;

5. var k = this.getField("employee");

6. k.hidden = true;

7. if (j.value != 0)

8. for (var i = 0; i < j; i++){

9. var tin = this.getField("employee.tin." +
   i);

10.    tin.hidden = false;

11. }
```

To simplify the script for purposes of discussion I added just the first column (*tin).* The actual script continues after line 10 to include the remaining columns (SS Number, Surname, Given Name, Middle Name, Date of Birth, Position, Monthly Salary, Date of Employment, and Previous Employer). Each of these fields need to be added to the script the same as lines 9 and 10 to unhide the fields.

In lines 1-3 the result in the total employees field is the sum of the male and female employees. The variable *j* is assigned to the total number of employees.

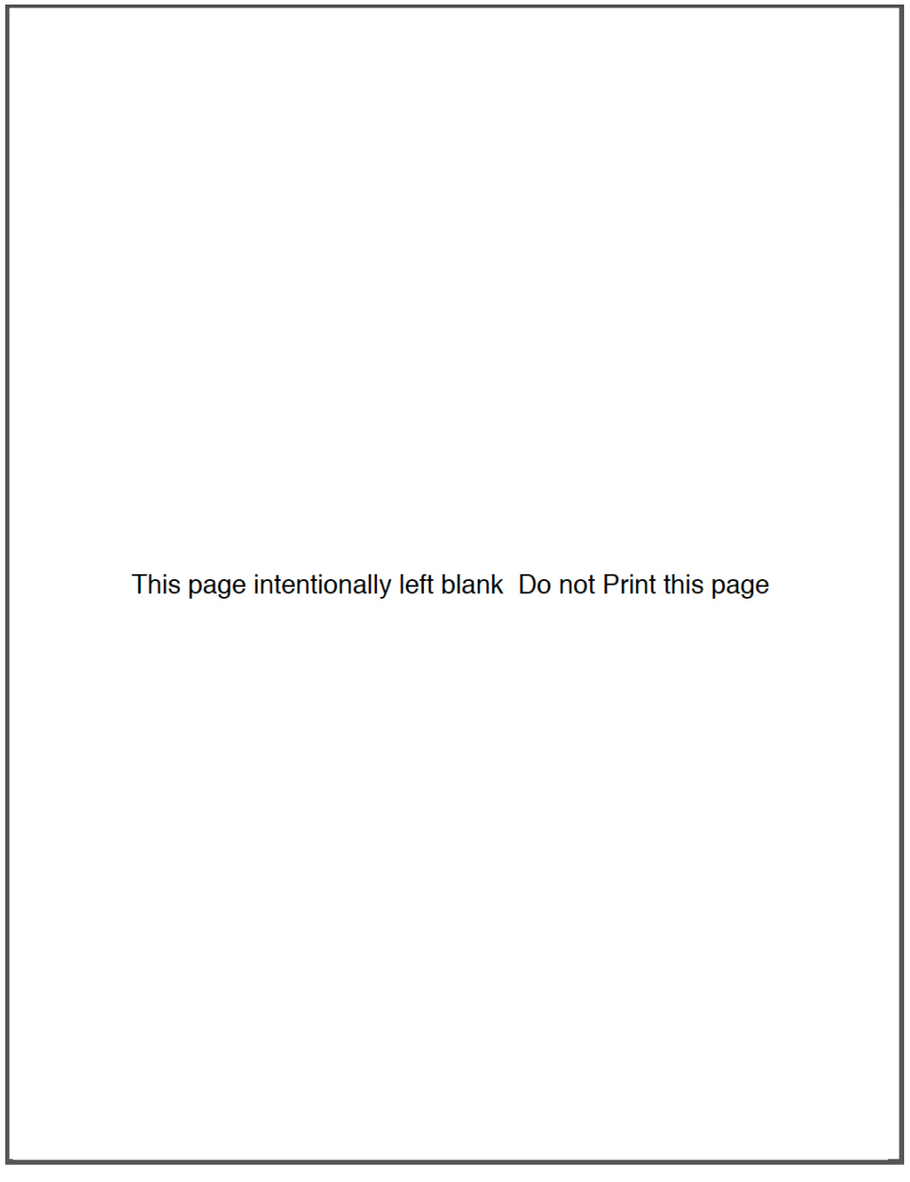
In line 7, if the total number is not 0 (zero) the script executes the lines following line 7. In line 8 a loop is set up to loop through the total number of employees (*j* value). Therefore if the total number of employees is five, the loop runs 5 times (the length of j).

Line 9 takes the first field in the table in column one and assigns the variable *tin* to the field *employee.tin + i.* The first time the loop is run the tin value is *employee.tin.0*. The next time the routine loops through, the tin value is *employee.tin.1*, and so on.

Another item that helps users fill in the form is a question asked about foreign investors. If the answer is yes, additional pages and fields need to be completed. If the answer is no, there's no reason to show the fields. By default, the fields for questions related to foreign investors are hidden. The check boxes on the master form ask if the user is a foreign investor (Yes) or not (No). JavaScripts are contained in the check box fields to show/hide fields.

The page containing questions related to foreign investors contains a text field with the appearance set to fill white. Text in the field displays a message that the page is intentionally left blank and not to be printed. Additionally, fields on the page are hidden.

Figure 24.6 The Default View



By default the page contents and fields are hidden.

The background design is hidden from view by the huge text box covering the page. In order to display the page design and the fields on the page, the text box and the fields need to be in view. When a user checks the Yes check box, a JavaScript shows the page design and the fields. The script is written as follows:

```
1. var f = this.getField("global.foreignText");

2. f.hidden = true;

3. var g = this.getField("global.foreign");

4. g.hidden = false;
```

The text field covering the page is named: *global.foreignText*. In the first line of code the variable *f* is assigned to this field. In line 2, *f* is hidden.

All the fields on the page have a common root value of *global.foreign*. Line 3 assigns the variable *g* to the *global.foreign* fields. In line 4 the fields are made visible (*`g.hidden = false`).*

On the master form the No check box reverses the fields visibility. The *`global.foreignText`* field is shown (`f.hidden = false;`). The global.foreign fields are hidden (*`f.hidden = true;`*)

Figure 24.7 Page Contents and Fields Visible

BTRCP Form No. 17

APPLICATION TO DO BUSINESS OR ENGAGE IN AN ECONOMIC ACTIVITY IN THE PHILIPPINES PURSUANT TO THE PROVISIONS OF REPUBLIC ACT 7042

1. Name of Applicant:

Romano Noel Henry
Last Name First Name Middle Name

2. Alien Certificate of Registration No. (if any)

ACR No.

3. Residence Address:

Blk 1 Waling Waling Street
Talomo
Davao City

Telephone Number: 213 0856479

4. Business Address:

Telephone Number:

5. Sources of Financing

	Total	Local Sources	Foreign Sources	Country
Owner's Capital				*** Choose one****
Loans:				
Short-Term				*** Choose one****
Long-Term				*** Choose one****
Others:				*** Choose one****

SPECIFY ACTIVITY E.G. MANUFACTURER, IMPORTER, EXPORTER, CONTRACTOR, DEALER, ETC., OR ANY COMBINATION OF THESE SPECIFYING TO SERVICE PRODUCT COLUMN THE SPECIFIC SERVICE AND PRODUCTS INVOLVED. (NOTE: RETAIL BUSINESS NOT ALLOWED.

6. SOURCES OF FINANCING

	Total Project Costs			
	Amount of Pesos			Country
Owner's Capital	Total	Local Sources	Foreign Sources	
Loan :				
- Short Term				*** Choose one****
- Long Term				*** Choose one****
Others				*** Choose one****

Projection Of Sales for the next 3 years

Year	Local	Export	Total
			0.00
			0.00
			0.00

I hereby agree to submit annual and other reports that may be required by the Department of Trade & Industry.

Signature of the Applicant or his Authorized Representative

Republic of the Philippines)
City/Province of _____)
Subscribed and sworn to before me this __day of _____20__ in the Province/City of _____, affiant exhibited his residence certificate/passport no. _____ issued at _____ on _____ 20___.

NOTARY PUBLIC

Document No. ____
Page No. ____
Book No. ____
Series of ____

When the Yes check box is checked, the huge text field is hidden and the fields are made visible.

Certificates Form

Another project requested by a local government office in the Philippines was a form needed to print certificates. A variety of different shops need inspection and approval such as fire extinguisher, emissions control, auto repair shops, etc. After the shops are approved by a local government office, a certificate is printed and needs to be on display at the shop.

The certificates are renewed annually and new certificates are printed for each shop every year.

Figure 24.8 Certificates for For Fire Extinguishers

Republika ng Pilipinas
Department of Trade and Industry
BUREAU OF TRADE REGULATION & CONSUMER PROTECTION

No. T-XI-99-77-555 (N)

LICENSE CERTIFICATE
FOR
FIRE EXTINGUISHER BUSINESS

Pursuant to the provisions of Rule 37, P.D. 1185 known as the Fire Code of the Philippines and Bureau of Trade Regulations and Consumer Protection Administrative Order No. 2, Series of 1979 dated March 2, 1979.

City Fire Extinguisher Shop

is hereby certified as qualified and authorized as

TECHNICIAN

of fire extinguisher and supplies and equipment of the same in the Philippines, subject to the provisions of laws, rules and regulations which are now or may hereafter be in force relating to fire extinguishers, their supplies and equipment.
This license certificate expires on December 31, 2013 unless sooner cancelled, revoked or suspended for cause.

IN WITNESS WHEREOF, I have hereunto set my hand and caused the seal of the DEPARTMENT OF TRADE AND INDUSTRY to be affixed at Davao City, Philippines this 20 day of March Two Thousand and Thirteen.

Clear Data | Print Certificate

Current Number of Certificates in this file: 1

Documentary Stamp Tax Paid PhP 15.00

TEOLULO T. PASAWA
City Director

Certificates are printed annually for a variety of different businesses after being certified by local government.

Problem

The local government office printed certificates each year for each type of shop. The office was burdened with rising printing costs and they always printed more certificates than used for any given year. The additional certificates were trashed at the end of each year.

Printed certificates were given to shops and a copy of each certificate was kept by the government office and filed.

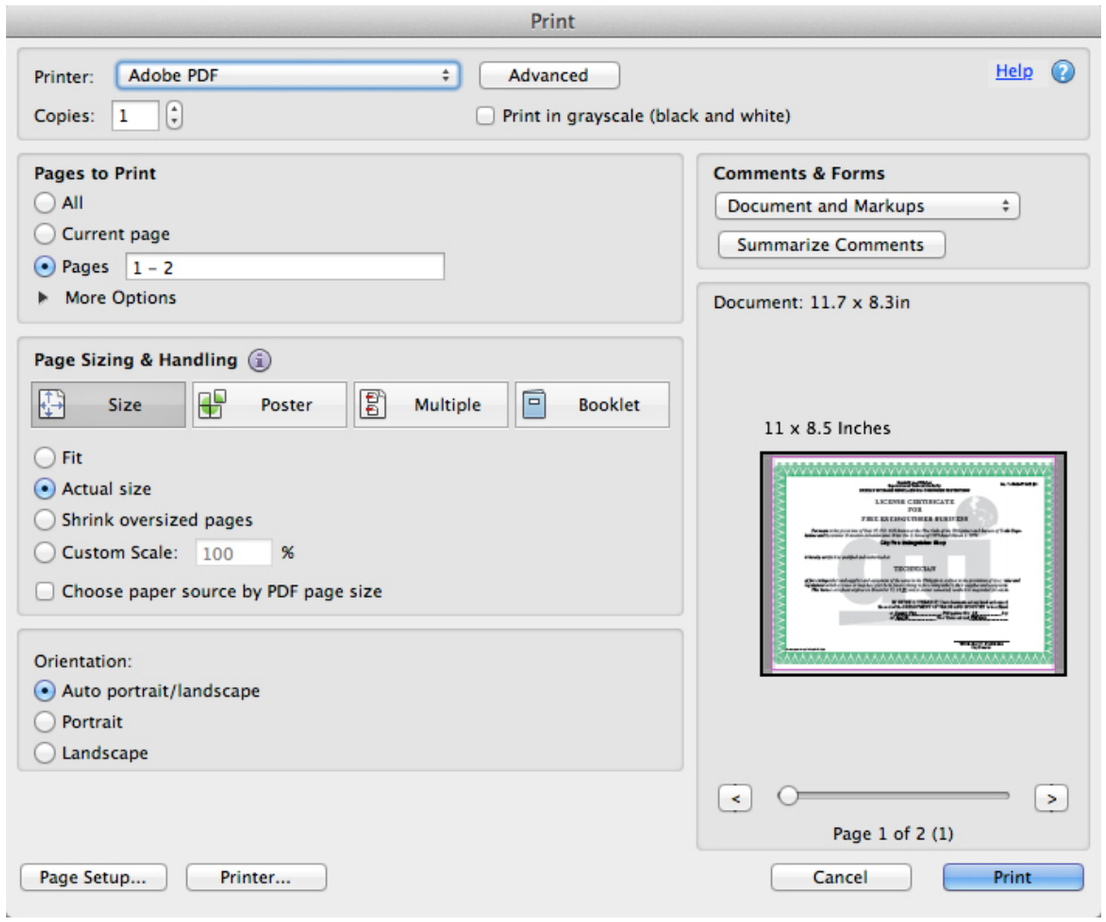
Solution

Rather than print certificates for an anticipated number and print duplicates for a government office to file, it made more sense to print certificates on demand. Additionally, the government office could keep electronic records and eliminate the need for filing paper records.

The forms served several purposes. The certificate design was used in the form, the form was printed by the government office, and data were aggregated on separate summary pages.

When a worker completed the form, the Print dialog box opened via a JavaScript on the Print button. The worker clicks the Print button to print the form.

Figure 24.9 Print Dialog Box



After a form is populated with data, the Print button opens the Print dialog box.

After printing, a button appears on the form to send the data to a summary page. The user clicks the button before they can continue. The button JavaScript sends the data from the certificate to a summary form contained in the same document.

The JavaScript behind the Send Data button was written as follows:

Figure 24.10 Send Data Button

After the form is printed a button appears to send the data to a summary form.

```
1.  var h = this.getField("s.counter");

2.  var oPage = this.getField("page");

3.  var c = this.getField("holder");
```

The first 3 lines of code set up some counters. As new records are added to the file, the counters increment so the data sent to the summary page falls after the previous record.

```
4.  //name

5.  var j = this.getField("P" + h.value +
    ".summary.list.name." + c.value);

6.  var name = this.getField("a.name");

7.  j.value = name.value;
```

Lines 4 to 7 is a partial list. Not all fields are included in this abbreviation of the script. In line 5 var *j* is assigned to the target field on the summary form. Line 6 assigns the variable *name* to the respective field on the certificate. Line 7 sends the data from the certificate form to the respective field on the summary form. The actual routine on the form follows getting fields from the certificate form and sending data to the summary form.

```
8.  //increment the counters

9.  oPage.value = oPage.value + 1;

10. c.value = c.value + 1;
```

Lines 8-10 increment the counters.

```
11. var max = this.getField("maxRecord");

12. var msg = "You cannot add any more
    businesses to this file.";
```

```
13. var f = this.getField("printForm");

14. if (oPage.value >= 300){

15. app.alert(msg);

16. max.hidden = false;

17. f.readonly = true;

18. }
```

Lines 11 and 12 set up variables for assessing the value of a field *max.record* and a message alert. If 300 pages are populated on the form, the message box opens and informs the user that no more records can be added to the form. The maximum number of records is made visible on the form and the print field is set to read only. This prevents the user from printing additional certificates.

```
19. var oHide = this.getField("sendData");

20. oHide.hidden = true;
```

The last lines of code hide the Send Data button field

Figure 24.11 Summary Form

	Name	Year (expire)	Signed Location	Signed (day)	Signed (month)	Signed (yr)	Cert Number
	City Fire Extinguisher Shop	2014	Davao City	20th	March	2013	No. T-XI-99-77-555 (N)

Summary form collects data for each new certificate.

The summary form is a history file and users in the department office can review the form for duplicate requests, reprinting lost forms and keeping an electronic record of certificates printed in a given year. There remains no need for office personnel to print duplicate certificates for the office files.

Health Statistics Form

The Botswana, Africa Ministry of Health needed a form for in-house use to accumulate data on patients in hospitals and clinics. They needed improvement in their data collection system to help reduce backlogs and stay current with accumulating data on patients' diagnoses.

Problem

The Botswana Ministry of Health central statistical unit collected forms from hospitals and clinics throughout the country. The printed forms were used for encoding data in their database management system (SPSS).

The statistical unit was backlogged more than 3 years and the staff time was devoted exclusively to data entry rather than data analysis.

Solution

Rather than use a single central office to encode data, a more productive measure was to develop forms that could be used by facility's staff in local clinics and hospitals, send the forms to the central office, and the central statistical office could export the data to their SPSS system. The statistical staff could then be freed from data entry and focus more time on data analysis.

Among the challenges with developing a form for facilities use was properly identifying diagnostic conditions. Each primary and secondary diagnosis required identifying the precise diagnostic term accompanied by an international ICD-10 code. In addition, a local office code (HSU) was also required for each diagnosis.

There are over 14,000 ICD-10 codes. The central office needed to manually search through manuals to find a diagnostic condition, the accompanying ICD-10 code, and reference the internal HSU code to populate the data. Using manual methods required extensive staff time.

Designing the Forms

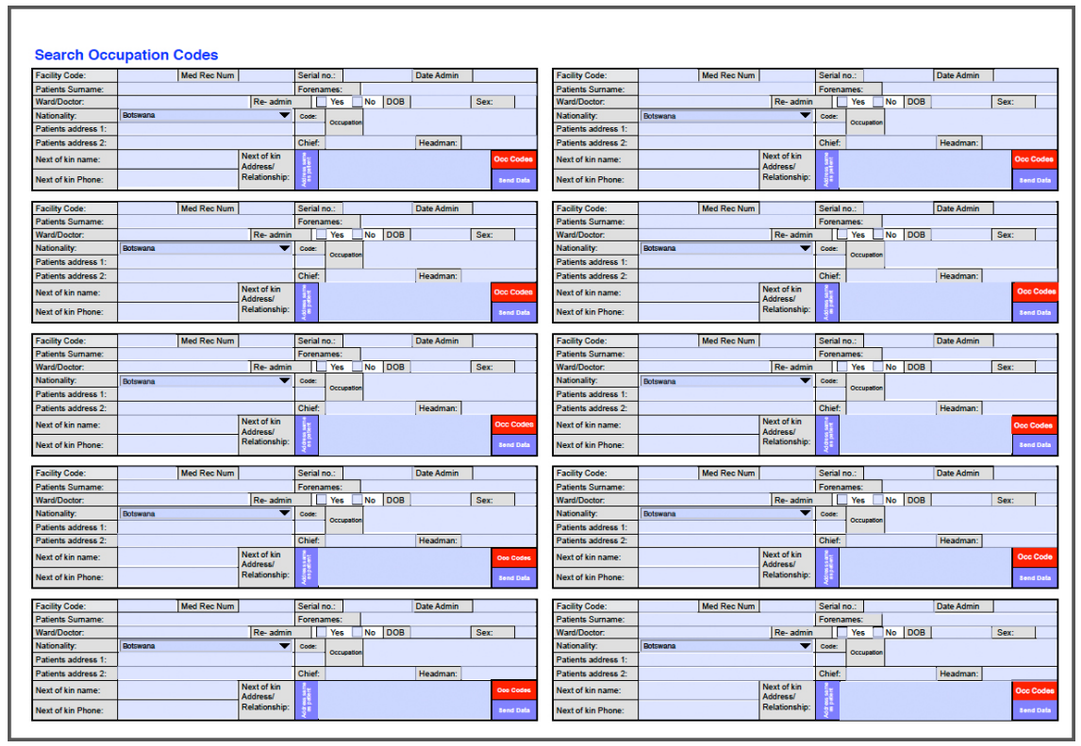
Three forms were designed for the project —a registration form, a discharge form, and a summary form. The registration form was used for inpatient admittances in clinics and hospitals.

The registration form is used when a patient is admitted to a clinic or hospital. The default form has a single page. Each page holds 10 records.

Records remain on this form until a patient is discharged. When a patient is discharged, a button on each registration card sends the registration data to the discharge form. When the data are sent to the discharge form, the registration card is cleared of data where a new registration can be added to the old card.

If all 10 cards are populated on a page, a JavaScript automatically spawns a new page from a template. Pages are dynamically created as needed.

Figure 24.12 Patient Registration Form



The registration form accommodates 10 registrations per page.

On the first page in the form a button exists prompting the user to create a new registration. The JavaScript that spawns a page from a template is placed on this button.

The JavaScript looks at the first field on the registration cards. If the Facility Code field (first field on each card) contains data, the JavaScript looks at the next field. The script loops through the 10 fields on a page. If the first field contains no data, the cursor is placed in the field. If all fields on a given page contain data, a new page is spawned from a template.

The script is written as follows:

```
1. var nTemplateNum = 1;
2. var bDone = false;
3. do{
4.   // Get field-set for a specific template
   page
5.   var oFieldSet = this.getField("P" +
   nTemplateNum +".reg.r.facility");
6.   if(oFieldSet != null)
7.   {// If field set exists, then iterate to
   find empty field
8.     var aFlds = oFieldSet.getArray();
9.     for (i =0; i < aFlds.length; i++)
10.  {
11.     if(aFlds[i].value =="")
12.     {// Empty field found, set focus to
   this field.
13.     // Set the zoom level
14.     this.zoom = 150;
15.     aFlds[i].setFocus();
16.     bDone = true;
17.     break;
18.    }
19.  }
20. }
21. else
```

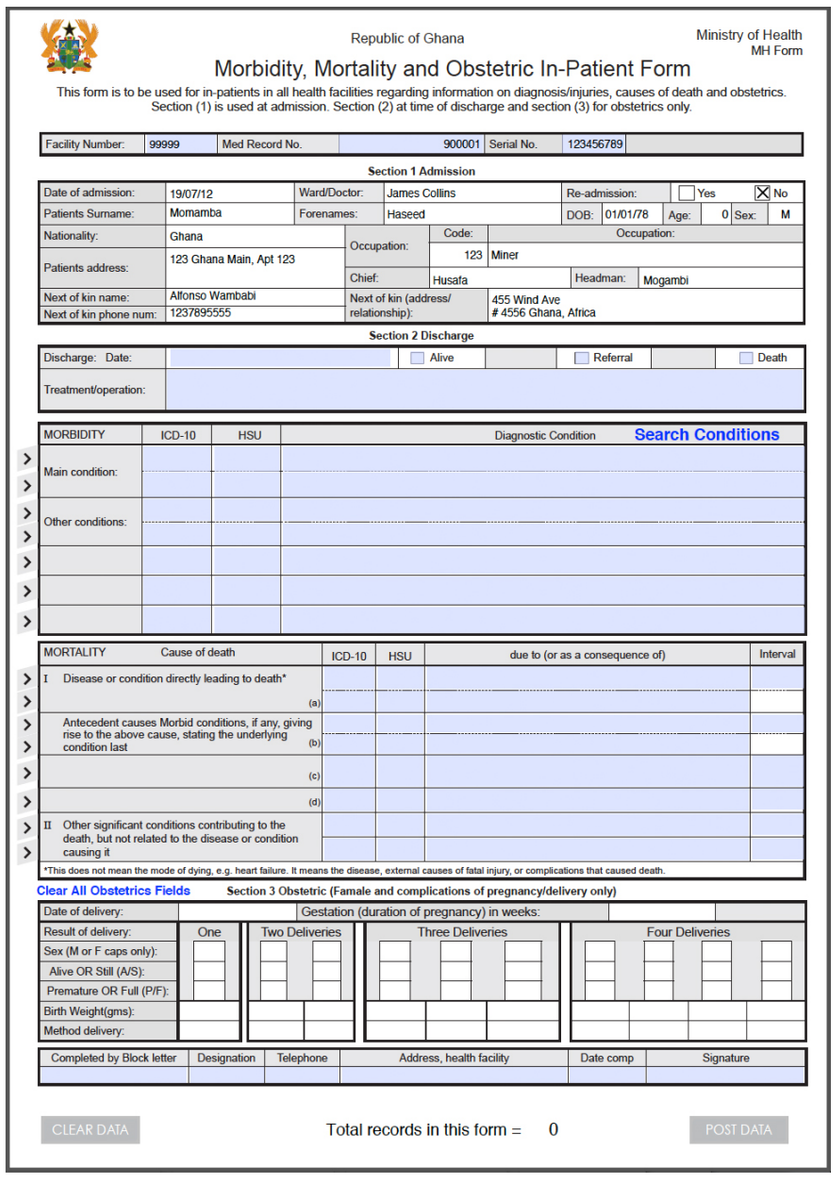
```
22. {// Template does not exist, Time to spawn
    a new one
23.  bDone = true;
24.   var a = this.getTemplate("reg");
25.   a. spawn ({
26.      nPage:this.numPages+1,
27.      bRename:true,
28.      bOverlay:false,
29.      });
30. this.zoom = 150;
31. this.getField("P" + nTemplateNum
    +".reg.r.facility.0").setFocus();
32.  }
33.  nTemplateNum++;
34.}while(!bDone);
```

The first part of the routine in lines 5 - 20, assesses fields for data. If a field contains data line 15 sets the focus (moves the cursor) to the empty field.

If all 10 records have data in the first field, lines 22 - 33 are executed. A new page is spawned and the cursor is placed on the first field in the new page. All of this is transparent to the user. One simply clicks the button for a new registration and the cursor appears in a field (either on an existing page or on a new spawned page.

When a patient is discharged, the clinic worker clicks the Send Data button on a given registration card. The JavaScript on the Send Data button opens the discharge form (form MH-017) and populates the Section I data. The registration card is cleared of data after the data are sent to form MH-017.

Figure 24.13 Discharge Form



The Section I fields are read only. Data needs to be introduced from the registration form.

Form MH-017 contains a one-page form followed by 225 pages of a lookup table. In order to precisely record diagnostic codes the diagnostic descriptions and codes are contained within the file and data are transposed from the table to the form. This eliminates a need for workers to use manuals and attempt to record data from a manual to the form.

To make it easy for workers to search the table, a button on the form appears as *Search Conditions*. This button has an Execute Menu Item action specified for Edit ➤ Advanced Search. When the user clicks the button the Advanced Search window opens.

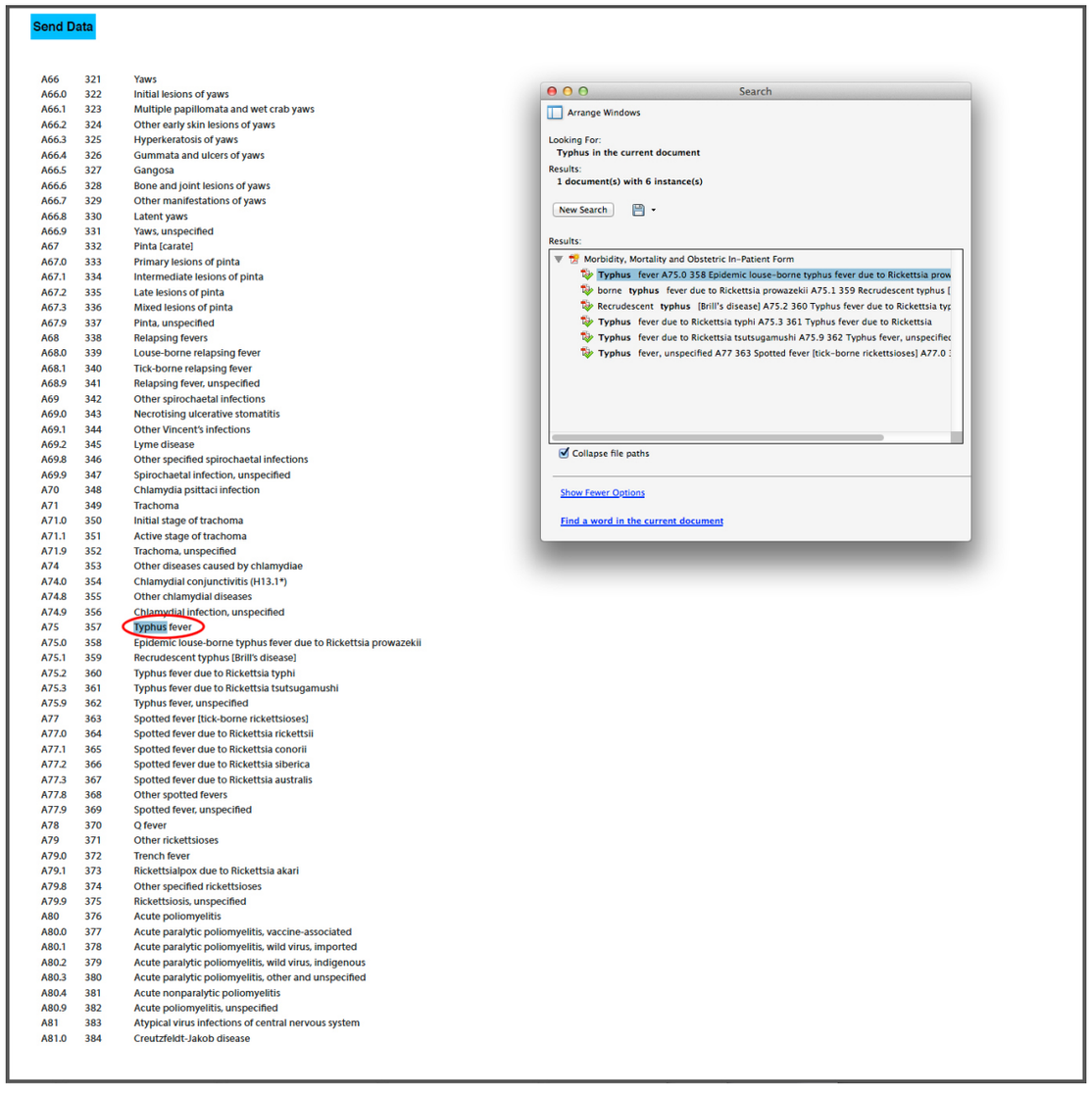
The user types search criteria in the Search window and clicks Search. Results are displayed in the Search window. In this example I typed Typhus and the results are shown for all conditions related to Typhus.

In the Search window the user can click any one of the results and the corresponding page where a given result is found opens in the Document pane.

The found result is highlighted on the page so the user knows precisely where the found result appears.

Each page in the lookup table contains a button. The lookup table does not contain separate fields for each condition and code. Rather, the JavaScript behind the button assesses the x,y coordinates of a mouse click, looks below the mouse click and sends the text on the page to form fields at the top of the page.

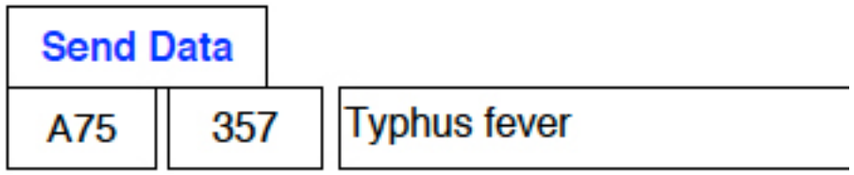
Figure 24.14 Search Results



The searched word(s) appear in the Search window and the selected result is highlighted on the corresponding page.

For example, if I click on Typhus highlighted in this example, the text below the highlight is *A75 357 Typhus fever*. The JavaScript reads the text and sends A75 to one field, 357 to a second field, and Typhus fever to a third field at the top of the page. If the data appear correct to the user, the user clicks the Send Data button.

Figure 24.15 Data in Three Fields



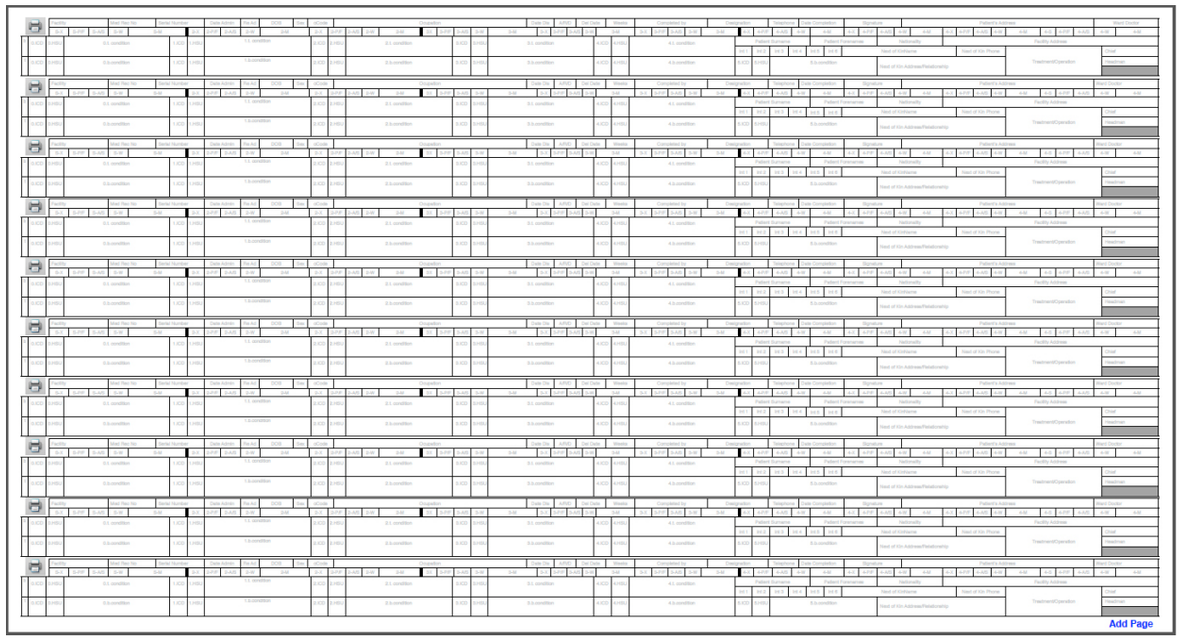
When the user clicks a condition on a page, the text is parsed into three data fields.

The Send Data button has a JavaScript that sends the data in the three fields to form MH-017. Before sending the data a number of error checks are made. Diagnostic conditions that are gender specific need to match the gender choice on the MH-017 form. Age is checked for conditions related to pregnancy and need to be consistent with ages of child bearing years. If conditions are not appropriate for sex and age choices on the MH-017 form, application alert dialog boxes open with a message and the data are cleared. Only when the data meet appropriate conditions are the data sent to the form.

When form MH-017 is completed, the user clicks the Post Data button in the lower right corner of the form. The JavaScript for this button opens the summary file where a new record is created for the discharge. Data are cleared on MH-017 and ready for a new discharge.

A button at the lower right corner of the summary page is used to spawn a new page from a template. By default, the summary form is a one-page document. New pages are added as needed.

Figure 24.16 Summary Form



Summary form collects discharge data.

The summary form is used to collect a month's worth of data. At the end of the month the summary form is submitted to the central statistical office where personnel integrate the data into their SPSS analysis system. The entire process alleviates a heavy workload from central office personnel who can spend their time in data analysis as opposed to data entry.

Farm Products Order Form

The area of Cotabato on the island of Mindanao in the Philippines is rich with agricultural products. The regional government office wanted a form that could be hosted on a website to facilitate ordering agricultural products from agricultural cooperatives.

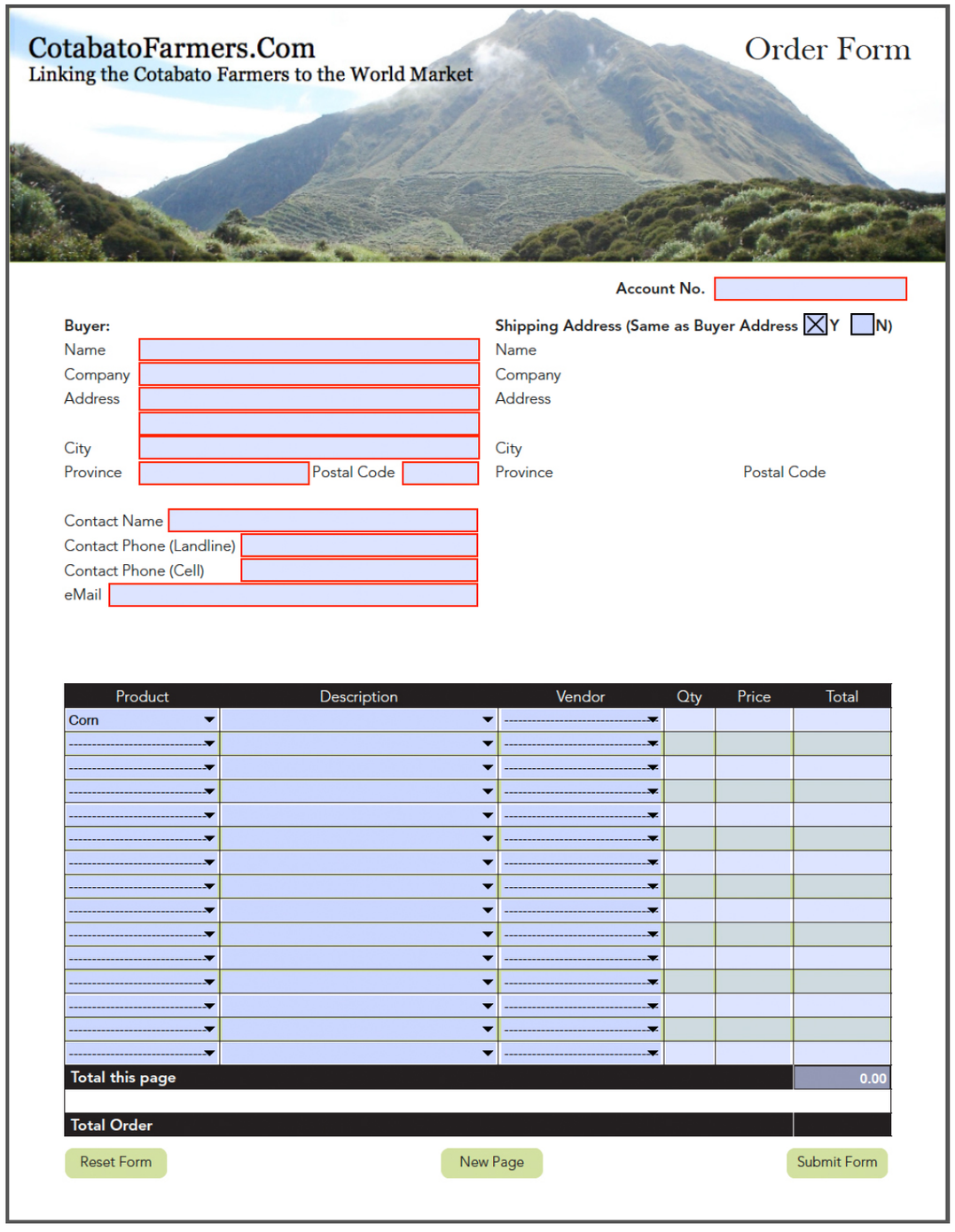
Problem

The regional government office wanted a simple, easy to use form they could host on a website and collect form data electronically. There was no such form at the time to perform electronic ordering. The office also wanted the form to be flexible for purchasers of large quantities of farm products as well as a short form for purchasers of limited items. They further wanted error checking and a means of limiting common problems they frequently experience when collecting order forms.

Solution

The most common problem for the farmers was clients not properly identifying descriptions for products. Placing an order for corn requires the purchaser to describe what product they wish to purchase when ordering corn (canned, whole ear, or creamed corn). If a form requires the user to type text in a field, users may not complete the field for the description or they may add a term not within the product range of the farmers.

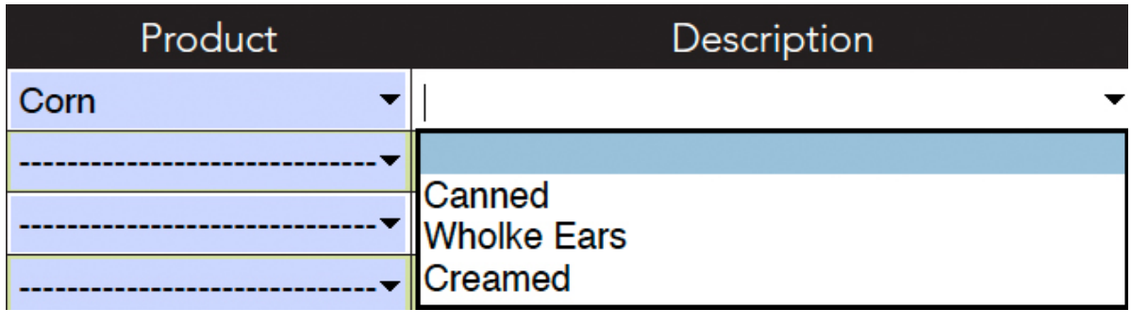
Figure 24.17 Cotabato Farmers Order Form



A form was designed to help facilitate proper product descriptions for items ordered.

To resolve the problem a form was designed that required the description field to be completed. The only way a user could complete the required field is to choose an option from a drop-down menu. The choices in the drop-down menu however are specific to the Product choice. Therefore if a user selects Corn as the product, the description dropdown field offers choices for canned, whole ear, or creamed.

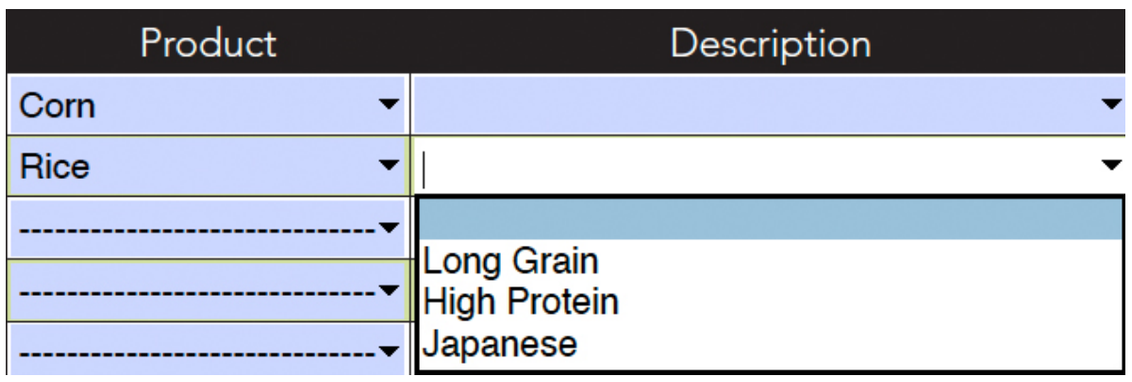
Figure 24.18 Corn Selected in First Dropdown Menu



When Corn is selected the Description dropdown menu restricts the options to corn.

If the user selects another product such as Rice, the choices are limited in the description drop-down menu to Long Grain, High Protein, and Japanese.

Figure 24.19 Rice Selected in First Dropdown Menu



When Rice is selected the Description dropdown menu restricts the options to rice.

To restrict the choices in the Description field, a JavaScript was written in the Product dropdown menu field.

The script was added to the Format tab and written as follows:

```
1. if (!event.willCommit) {
2.     var a = this.getField("description.0");
3.     a.clearItems();
4. switch (event.change) {
5.   case "Corn":
6.   a.setItems(["", "Canned", "Whole Ears",
   "Creamed"]);
7.   break;
8.   case "Milk":
9.   a.setItems(["", "Alaska Filled",
   "Condensed", "Powered"]);
10.  break;
11.  case "Rice":
12.   a.setItems(["", "Long Grain", "High
   Protein", "Japanese"]);
13.  break;
14.  case "Sardines":
15.   a.setItems(["", "In Tomato Sauce",
   "Canned", "Rolled", "Flat"]);
16.  break;
17.   }
18.}
```

In line 2 we identify the target field which is also a dropdown menu. Line 3 clears the target field. Lines 4 to 18 set up the options for the target field. Each line where you see Case "*item*" identifies the items in the Product drop-down menu. The lines are followed by the *a.setItems...* that describes the options ultimately shown in the Description dropdown menu.

This kind of restricting a user to complete fields before data can be submitted and limiting responses within a specific range can greatly help the reliability of the data.