Format Portable Document Format (PDF) to sposób, w jaki większość dokumentów jest tworzona do dystrybucji, współpracy i archiwizacji na całym świecie. Został on ujednolicony przez Międzynarodową Organizację Normalizacyjną (ISO) i przez rządy w ponad 75 krajach jako ich format wyboru dla ich dokumentacji. Przemysł poligraficzny wymagał użycia formatu PDF do każdego profesjonalnego zadania drukowania. Dzięki miliardom publicznie dostępnych dokumentów i niezliczonej liczbie dokumentów mieszkających w prywatnych repozytoriach żaden inny format pliku nie ma tak szerokiego zasięgu i wszechobecności, jak PDF. Jednak nawet w przypadku tych miliardów dokumentów w obiegu format PDF pozostaje słabo zrozumiały zarówno dla użytkowników, jak i programistów ze względu na brak dokumentacji poza standardem ISO 32000-1 - standardem PDF. I chociaż standard jest doskonałym dokumentem technicznym, jego rozmiar, złożoność i suchy styl sprawiają, że jest on nieczytelny dla wielu. Naszym celem jest zapewnienie przystępnego odniesienia do pliku PDF. Obejmujemy kluczowe tematy ze standardu w sposób, który umożliwi technicznie zrozumiałe zrozumienie treści zawartych w pliku PDF. Dla tych, którzy po prostu chcą zbadać wewnętrzne elementy PDF w celu zdiagnozowania problemów, znajdziesz tu potrzebne narzędzia, a ci, którzy chcą skonstruować własne, prawidłowe i dobrze sformatowane dokumenty, dowiedzą się, jak to zrobić.

Część 1

Zaczynamy od przyjrzenia się różnym obiektom, które tworzą plik PDF i połączeniu ich w spójną całość.

Część 2

Przyjrzymy się zasadniczemu aspektowi PDF - jego modelowi obrazowania. Dowiadujemy się, jak utworzyć stronę i narysować na niej grafikę.

Część 3

Kontynuując naszą dyskusję na temat podstawowego modelu obrazowania, zbadamy, jak włączyć obrazy rastrowe do treści PDF.

Część 4

Następnie dowiemy się, jak włączyć ostatni typowy typ treści PDF. Oczywiście omówienie tekstu w formacie PDF nie byłoby kompletne bez zrozumienia czcionek i glifów.

Część 5

PDF to nie tylko statyczna zawartość. Zostaną przedstawione różne sposoby, dzięki którym PDF może uzyskać interaktywność, zwłaszcza w celu umożliwienia nawigacji wewnątrz i pomiędzy dokumentami.

Część 6

Omówimy specjalne obiekty, które są adnotacjami, które są rysowane na wierzchu zwykłej zawartości, aby umożliwić wszystko, od interaktywnych linków do 3D do wideo i audio.

Część 7

Następnie przyjrzymy się, w jaki sposób udostępniane są formularze interaktywne w języku PDF.

Część 8

Ten część pokazuje, w jaki sposób plik PDF może być używany w podobny sposób jak archiwum ZIP, osadzając w nim pliki.

Część 9

Wyjaśnimy, w jaki sposób treści wideo i audio można odwoływać się lub osadzać w pliku PDF w celu odtwarzania w ramach bogatej zawartości.

Część 10

Tu przedstawiamy opcjonalną zawartość, która pojawia się tylko w określonych momentach, np. na ekranie, ale nie po wydrukowaniu lub tylko dla niektórych użytkowników.

Część 11

Omówimy sposób dodawania bogactwa semantycznego do treści poprzez oznaczanie go strukturami podobnymi do HTML, takimi jak akapity i tabele.

Część 12

Omówimy różne sposoby włączania metadanych do pliku PDF, od najprostszych ciągów na poziomie dokumentu po bogaty XML dołączony do poszczególnych obiektów.

Część 13

Wreszcie, ta część przedstawia różne otwarte standardy międzynarodowe oparte na formacie PDF, w tym sam pełny standard PDF (ISO 32000-1), różne podzbiory (takie jak PDF / A i PDF / X), a także powiązane prace (takie jak PAdES).

Rozpoczniemy naszą eksplorację plików PDF, zanurzając się bezpośrednio w elementach formatu PDF. Korzystając z tych bloków, zobaczysz, jak skonstruowano plik PDF, aby doprowadzić do znanego formatu opartego na stronach.

Obiekty PDF

Podstawowa część pliku PDF to zbiór „rzeczy", do których odnosi się standard PDF (ISO 32000) jako obiekty lub czasami obiekty COS.

COS oznacza **Carousel Object System** i odnosi się do oryginału / nazwa kodowa produktu Adobe Acrobat.

Nie są to obiekty w znaczeniu tego słowa „programowanie obiektowe"; zamiast tego są klockami, na których stoi plik PDF. Istnieje dziewięć typów obiektów: null, Boolean, integer, real, name, string, array, dictionary i stream. Przyjrzyjmy się każdemu z tych typów obiektów i ich serializacji w pliku PDF. Stamtąd zobaczysz, jak wziąć te typy obiektów i użyć ich do zbudowania konstrukcji wyższego poziomu i samego formatu PDF.

Obiekt Null

Obiekt null, jeśli faktycznie zapisany do pliku, to po prostu cztery znaki null. Jest to równoznaczne z brakującą wartością, dlatego bardzo rzadko można go znaleźć w pliku PDF. Jeśli masz powód do pracy z wartością zerową, upewnij się, że dokładnie zapoznałeś się z ISO 32000 odnośnie subtelności związanych z jego obsługą.

Obiekty boolowskie

Obiekty boolowskie reprezentują wartości logiczne true i false i są odpowiednio reprezentowane w pliku PDF, jako prawdziwe lub fałszywe

Podczas pisania pliku PDF zawsze użyjesz wartości true lub false. Jeśli jednak czytasz / analizujesz plik PDF i chcesz być tolerancyjny, pamiętaj, że źle napisane pliki PDF mogą korzystać z innych form kapitalizacji, w tym wiodących dużych liter (Prawda lub Fałsz) lub wszystkich dużych liter (PRAWDA lub FAŁSZ)

Obiekt numeryczny

PDF obsługuje dwa różne typy obiektów liczbowych - całkowite i rzeczywiste - reprezentujące ich matematyczne odpowiedniki. Podczas gdy starsze wersje PDF określiły ograniczenia implementacji, które pasowały do starszych implementacji Adobe, nie powinny już być traktowane jako ograniczenia formatu plików (ani te o jakiejkolwiek konkretnej implementacji przez jakiegokolwiek dostawcę).

Podczas gdy PDF obsługuje liczby 64-bitowe (aby włączyć bardzo duże pliki), przekonasz się, że większość plików PDF w rzeczywistości ich nie potrzebuje. Jednak jeśli ty czytasz plik PDF, możesz go rzeczywiście spotkać, więc bądź przygotowany

Obiekty liczbowe składają się z jednej lub więcej cyfr dziesiętnych, opcjonalnie poprzedzonych znakiem, reprezentujących podpisaną wartość (o podstawie 10). Przykład 1-1 pokazuje kilka przykładów liczb całkowitych.

Przykład 1-1. Liczby całkowite

```
1

-2
```

+100

612

Rzeczywiste obiekty liczbowe składają się z jednej lub więcej cyfr dziesiętnych z opcjonalnym znakiem i okresem początkowym, końcowym lub osadzonym reprezentującym podpisaną wartość rzeczywistą. W przeciwieństwie do PostScript, PDF nie obsługuje formatu naukowego / wykładniczego, ani nie obsługuje dziesiętnych radionów.

Podczas gdy termin „rzeczywisty" jest używany w pliku PDF do reprezentowania typu obiektu, rzeczywista implementacja danej przeglądarki może wykorzystywać liczby całkowite zmiennoprzecinkowe, a nawet stałe. Ponieważ implementacje mogą się różnić, liczba miejsc dziesiętnych dokładności może się również różnić. Zaleca się zatem niezawodność, a także względy rozmiaru pliku, aby nie zapisywać więcej niż cztery miejsca po przecinku. Przykład 1-2 pokazuje kilka przykładów liczb rzeczywistych w formacie PDF.

Przykład 1-2. Liczby rzeczywiste

0,05

.25

-3,14159

300,9001

Obiekt name

Obiekt name w PDF jest unikalną sekwencją znaków (z wyjątkiem kodu 0, ASCII null) normalnie używaną w sytuacjach, gdy istnieje stały zestaw wartości. Nazwy są zapisywane w pliku PDF za pomocą znaku / (SOLIDUS), po którym następuje łańcuch UTF-8, ze specjalną formą kodowania dla dowolnego nieregularnego znaku. Znaki nieregularne to takie, które znajdują się poza zakresem 0x21 (!) Przez 0x7E (~), a także każdy znak spacji. Te nieregularne znaki są kodowane zaczynając od znaku # (NUMBER SIGN), a następnie dwucyfrowego kodu szesnastkowego dla postaci. Ze względu na ich wyjątkowy charakter, większość nazw, które napiszesz do pliku PDF, jest wstępnie zdefiniowana w ISO 32000 lub będzie pochodzić z danych zewnętrznych (takich jak czcionka lub nazwa koloru). Jeśli musisz utworzyć własne niestandardowe nazwy niestandardowe oparte na danych, takie jak prywatny fragment metadanych, musisz przestrzegać zasad dotyczących nazw drugiej klasy, zgodnie z definicją w ISO 32000-1: 2008, załącznik E, jeśli życzysz sobie, aby plik został uznany za prawidłowy plik PDF Druga nazwa klasy to taka, która zaczyna się od czteroznakowego prefiksu zarejestrowanego w ISO, a następnie podkreślenia (_), a następnie nazwy klucza. Przykład jest zawarty na końcu przykładu 1-3.

Przykład 1-3. Nazwy

/Type

/ ThisIsName37

/ Lime # 20 Green

/ SSCN_SomeSecondClassName

Obiekt typu String

Ciągi, które są serializowane do formatu PDF, są po prostu seriami (zero lub więcej) 8-bitowych bajtów zapisanych albo jako znaki literowe ujęte w nawiasy (i), albo dane szesnastkowe zawarte w nawiasach kątowych (< i >). Dosłowny ciąg składa się z dowolnej liczby znaków 8-bitowych ujętych w nawiasy. Ponieważ w łańcuchu może pojawić się dowolna 8-bitowa wartość, nawiasy okrągłe ()) i odwrotny ukośnik (\) są traktowane specjalnie przez użycie odwrotnego ukośnika, aby uniknąć specjalnych wartości. Dodatkowo odwrotny ukośnik może być użyty ze specjalną notacją \\ddd, aby określić inne wartości znaków. Dosłowne łańcuchy występują w kilku różnych odmianach:

ASCII

Sekwencja bajtów zawierająca tylko znaki ASCII

PDFDocEncoded

Sekwencja bajtów zakodowanych zgodnie z kodowaniem PDFDocEncoding

Tekst

Sekwencja bajtów zakodowanych jako PDFDocEncoding lub UTF-16BE (z wiodącym znacznikiem kolejności bajtów)

Data

Łańcuch ASCII, którego format D: RRRRMMDDHHmmSSOHHmm jest opisany w ISO 32000–1: 2008, 7.9.4 . Daty, jako typ ciągu, zostały dodane do pliku PDF w wersji 1.1.

Szereg cyfr szesnastkowych (0–9, A – F) można zapisać między nawiasami kątowymi, co jest przydatne do uwzględnienia bardziej czytelnych dla człowieka dowolnych danych binarnych lub wartości Unicode (UCS-2 lub UCS-4) w obiekcie łańcuchowym. Liczba cyfr musi być zawsze równa, chociaż między parami cyfr można dodać białe znaki, aby poprawić czytelność dla ludzi.

Przykład 1-4 przedstawia kilka przykładów ciągów znaków w formacie PDF.

Przykład 1-4. Strings

```
(Testowanie)                    % ASCII
(A 033 B)                       % Tak samo jak (A + B)
(Français)                      % PDFDocEncoded
<FFFE0040>                      % Tekst z prowadzącym BOM
(D: 19990209153925-08'00 ')     % Data
<1C2D3F>                        % Dowolne dane binarne
```

Znak procentu (%) oznacza komentarz; każdy następujący po nim tekst jest ignorowany

Poprzednia dyskusja na temat ciągów dotyczyła tego, jak wartości są serializowane w pliku PDF, niekoniecznie sposobu, w jaki są obsługiwane wewnętrznie przez procesor PDF. Chociaż taka wewnętrzna obsługa wykracza poza zakres standardu, ważne jest, aby pamiętać, że różne serializacje plików mogą dawać taką samą reprezentację wewnętrzną (jak (A 053B) i (A + B) w przykładzie 1-4).

Obiekt array

Obiekt tablicowy jest heterogeniczną kolekcją innych obiektów zamkniętych w nawiasach kwadratowych ([i]) i oddzielonych białymi znakami. Możesz mieszać i dopasowywać dowolne obiekty dowolnego typu w jednej macierzy, a PDF wykorzystuje to w różnych miejscach. Tablica może być również pusta (tzn. zawierać zero elementów). Podczas gdy tablica składa się tylko z jednego wymiaru, możliwe jest skonstruowanie odpowiednika wielowymiarowej tablicy. Ta konstrukcja nie jest często używana w formacie PDF, ale pojawia się w kilku miejscach, takich jak tablica kolejności w strukturze danych znanej jako opcjonalny słownik grup treści. Nie ma ograniczeń co do liczby elementów w tablicy PDF. Jeśli jednak znajdziesz alternatywę dla dużej tablicy (takiej jak drzewo strony dla pojedynczej tablicy Kids), zawsze lepiej jest ich unikać ze względu na wydajność. Niektóre przykłady tablic podano w przykładzie 1-5.

Przykład 1-5. Tablice

```
[0 0 612 792]          % 4-elementowa tablica wszystkich liczb całkowitych
[(T) –20,5 (H) 4 (E)]  % 5-elementowa tablica ciągów, liczb rzeczywistych i liczb całkowitych
[[1 2 3] [4 5 6]]      % 2-elementowa tablica tablic
```

Obiekt dictionary

Ponieważ służy jako podstawa dla prawie każdego obiektu wyższego poziomu, najczęstszym obiektem w PDF jest obiekt słownika. Jest to zbiór par klucz / wartość, znany również jako tabela asocjacyjna. Każdy klucz jest zawsze obiektem nazwy, ale wartością może być dowolny inny typ obiektu, w tym inny słownik lub nawet wartość null.

Gdy wartość jest null, jest traktowana tak, jakby klucz nie był obecny, dlatego lepiej jest po prostu nie pisać klucza, aby zaoszczędzić czas przetwarzania i rozmiar pliku

Słownik jest ujęty w podwójne nawiasy kątowe (<< i >>). W tych nawiasach klawisze mogą pojawiać się w dowolnej kolejności, a zaraz po nich ich wartości. Które klawisze pojawiają się w słowniku zostaną określone przez definicję (w ISO 32000) obiektu wyższego poziomu, który jest tworzony. Podczas gdy wiele istniejących implementacji ma tendencję do pisania kluczy posortowanych alfabetycznie, nie jest to ani wymagane, ani oczekiwane. W rzeczywistości nie należy również przyjmować żadnych założeń dotyczących przetwarzania słownika - klucze mogą być odczytywane i przetwarzane w dowolnej kolejności. Słownik zawierający dwa razy ten sam klucz jest nieprawidłowy, a wybrana wartość jest niezdefiniowana. Wreszcie, chociaż poprawia to czytelność ludzi w celu umieszczania podziałów między parami klucz / wartość, to również nie jest wymagane i służy tylko do dodawania bajtów do całkowitego rozmiaru pliku. Nie ma limitu liczby par klucz / wartość w słowniku

Przykład 1-6 pokazuje kilka przykładów.

Przykład 1-6. Słowniki

```
% bardziej czytelny dla człowieka słownik
<<
/ Typ / Strona
/ Autor (Leonardo Rosenthol)
/ Zasoby << / Font [/ F1 / F2] >>
```

>>

% słownik ze wszystkimi usuniętymi białymi spacjami

<< / Długość 3112 / Podtyp / XML / Typ / Metadane >>

Drzewo nazw

Drzewo nazw służy do podobnego celu jak słownik, ponieważ umożliwia powiązanie kluczy z wartościami. Jednak w przeciwieństwie do słownika, klucze to obiekty łańcuchowe zamiast nazw i muszą być uporządkowane / posortowane zgodnie ze standardowym algorytmem sortowania Unicode. Ta koncepcja jest nazywana drzewem nazw, ponieważ istnieje słownik „root" (lub węzeł), który odnosi się do jednego lub większej liczby słowników / węzłów potomnych, które same mogą odwoływać się do jednego lub większej liczby słowników / węzłów podrzędnych, tworząc w ten sposób wiele gałęzi drzewa podobna struktura. Węzeł główny zawiera pojedynczy klucz, albo Name (dla prostego drzewa), albo Children (dla bardziej złożonego drzewa). W przypadku drzewa złożonego każdy z węzłów pośrednich będzie również zawierał klucz Children; końcowe / końcowe węzły każdej gałęzi będą zawierały klucz Name. Jest to wartość tablicy klucza Name, która określa klucze i ich wartości na przemian klucz / wartość, jak pokazano w przykładzie 1-7.

Przykład 1-7. Przykładowe drzew nazwy

% Proste drzewo nazw z kilkoma nazwami

1 0 obj

<<

/Nazwy [

(Apple) (Orange) % Są one posortowane, stąd A, N, Z ...

(Nazwa 1) 1 %, a wartości mogą być dowolnego typu

(Nazwa 2) / Wartość2

(Zebra) << / A / B >>

]

>>

endobj

Drzewa liczb

Drzewo liczb jest podobne do drzewa nazw, z wyjątkiem tego, że jego klucze są liczbami całkowitymi zamiast łańcuchami i są sortowane w rosnącym porządku liczbowym. Ponadto wpisy w węzłach liści (lub korzeniach) zawierające pary klucz / wartość znajdują się jako wartość klawisza Nums zamiast klucza Names.

Obiekty bezpośrednie i pośrednie

Teraz, gdy już zapoznałeś się z typami obiektów, ważne jest, aby zrozumieć, że obiekty te mogą być reprezentowane bezpośrednio lub pośrednio w pliku PDF. Obiekty bezpośrednie to te obiekty, które

pojawiają się „inline” i są uzyskiwane bezpośrednio (stąd nazwa), gdy obiekty są odczytywane z pliku. Znajdują się one zwykle jako wartość klucza słownika lub wpisu w tablicy i są typem obiektu, który widziałeś we wszystkich przykładach. Obiekty pośrednie to te, do których odwołuje się (pośrednio!) Przez odniesienie, a czytnik PDF będzie musiał przeskakiwać wokół pliku, aby znaleźć rzeczywistą wartość. W celu określenia, do którego obiektu się odnosi, każdy obiekt pośredni ma unikalny (na PDF) identyfikator, który jest wyrażany jako liczba dodatnia i numer generacji, który jest zawsze liczbą nieujemną i zwykle zero (0). Liczby te są używane zarówno do zdefiniowania obiektu, jak i odniesienia do obiektu.

Chociaż pierwotnie miały być używane do śledzenia zmian w formacie PDF, numery generacji prawie nigdy nie są używane przez nowoczesne systemy PDF, więc prawie zawsze są zerowe.

Aby użyć obiektu pośredniego, należy najpierw zdefiniować go przy użyciu identyfikatora i generacji wraz ze słowami kluczowymi obj i endobj, jak pokazano w przykładzie 1-9.

Przykład 1-9. Pośrednie obiekty wykonane w całości z bezpośrednich obiektów

```
3 0 obj % object ID 3, generation 0
<<
/ProcSet [ /PDF /Text /ImageC /ImageI ]
/Font <<
/F1 <<
/Type /Font
/Subtype /Type1
/Name /F1
/BaseFont/Helvetica
>>
>>
>>
endobj
5 0 obj
(an indirect string)
endobj
% an indirect number
4 0 obj
1234567890
endobj
```

Kiedy odwołujesz się do obiektu pośredniego, robisz to przy użyciu jego identyfikatora, jego generacji i znaku R. Na przykład, dość często można zobaczyć coś takiego jak przykład 1-10, do którego odwołują się dwa obiekty pośrednie (identyfikatory 4 i 5) .

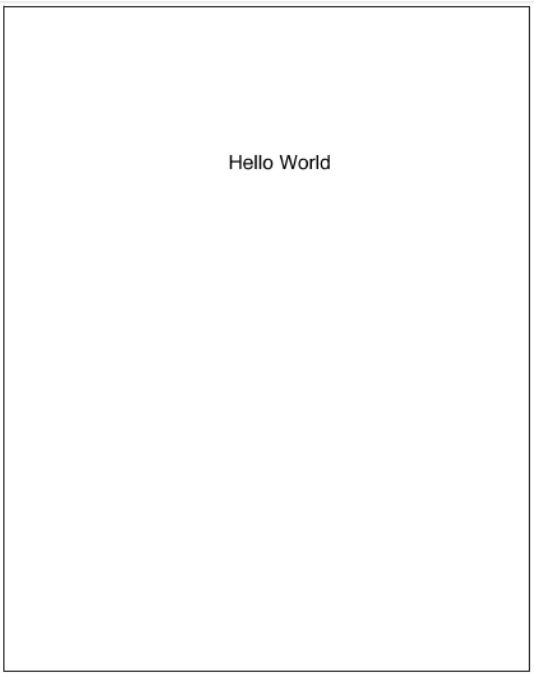
Przykład 1-10. Obiekt pośredni, który odwołuje się do innych obiektów pośrednich

```
3 0 obj % object ID 3, generation 0
<<
/ProcSet 5 0 R              % reference the indirect object with ID 5, generation 0
/Font <</F1 4 0 R >>        % reference the indirect object with ID 4, generation 0
>>
endobj
4 0 obj % object ID 4, generation 0
<<
/Type /Font
/Subtype /Type1
/Name /F1
/BaseFont/Helvetica
>>
endobj
5 0 obj % object ID 5, generation 0
[ /PDF /Text /ImageC /ImageI ]
endobj
```

Używając kombinacji ID i generacji, każdy obiekt może być jednoznacznie zidentyfikowany wewnątrz danego PDF. Korzystając z funkcji tabeli odnośników w formacie PDF, każdy obiekt pośredni można łatwo zlokalizować i załadować na żądanie z odnośnika. O ile ISO 32000 nie wskazuje inaczej, za każdym razem, gdy używasz obiektu, może on być dowolnego typu - z wyjątkiem strumieni, które mogą być tylko pośrednie.

Struktura pliku

Gdybyś widział prosty plik PDF - nazwijmy go Hello World.pdf - w przeglądarce plików PDF wyglądałby tak jak na rysunku 1-1.



Ale gdybyś zobaczył Hello World.pdf w aplikacji do edycji tekstu, wyglądałby trochę jak przykład 1-11.

Przykład 1-11. Jak wygląda „Hello World.pdf” w edytorze tekstu

```
%PDF-1.4
%%EOF
6 0 obj
<<
/Type /Catalog
/Pages 5 0 R
>>
endobj
1 0 obj
<<
/Type /Page
/Parent 5 0 R
/MediaBox [ 0 0 612 792 ]
/Resources 3 0 R
/Contents 2 0 R
>>
endobj
```

4 0 obj

<<

/Type /Font

/Subtype /Type1

/Name /F1

/BaseFont/Helvetica

>>

endobj

2 0 obj

<<

/Length 53

>>

stream

BT

/F1 24 Tf

1 0 0 1 260 600 Tm

(Hello World)Tj

ET

endstream

endobj

5 0 obj

<<

/Type /Pages

/Kids [1 0 R]

/Count 1

>>

endobj

3 0 obj

<<

/ProcSet[/PDF/Text]

/Font <</F1 4 0 R >>

>>

endobj

xref

0 7

0000000000 65535 f

0000000060 00000 n

0000000228 00000 n

0000000424 00000 n

0000000145 00000 n

0000000333 00000 n

0000000009 00000 n

trailer

<<

/Size 7

/Root 6 0 R

>>

startxref

488

%%EOF

Patrząc na to, może się wydawać błędne wrażenie, że plik PDF jest plikiem tekstowym, który można rutynowo edytować za pomocą edytora tekstu - tak nie jest! Plik PDF jest ustrukturyzowanym 8-bitowym dokumentem binarnym wyznaczonym przez serię 8-bitowych tokenów znakowych, oddzielonych białą spacją i ułożonych w (dowolnie długie) linie. Tokeny te służą nie tylko do nakreślenia różnych obiektów i ich typów, jak widzieliście w poprzedniej sekcji, ale także do określenia, gdzie zaczynają się i kończą cztery logiczne sekcje PDF.

Jak wspomniano wcześniej, tokeny w pliku PDF są zawsze kodowane (a zatem dekodowane) jako 8-bitowe bajty w ASCII. Nie można ich zakodować w żaden inny sposób, np. Unicode. Oczywiście określone dane lub wartości obiektów mogą być kodowane w Unicode; omówimy te przypadki w miarę ich pojawiania się.

Białe znaki

Białe znaki pokazane w tabeli 1-1 są używane w formacie PDF do oddzielania konstrukcji składniowych, takich jak nazwy i liczby od siebie.

Dziesiętnie	Szesnastkowo	Ósemkowo	Nazwa
0	00	000	NULL(NUL)
9	09	011	TABULATOR POZIOMY (HT)

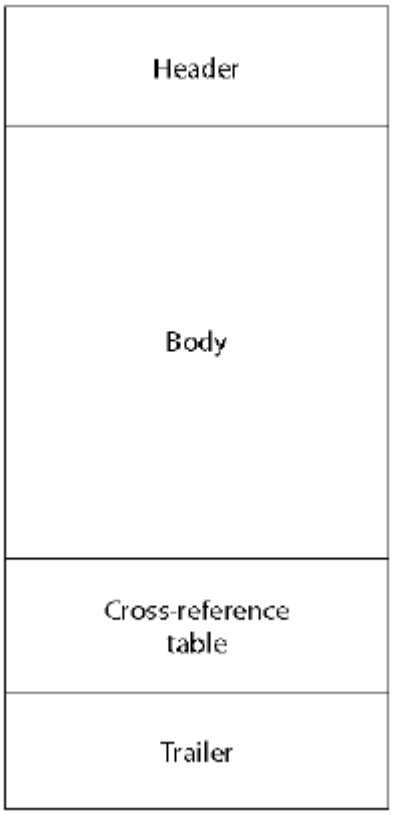
10	0A	012	LINE FEED(LF)
12	0C	014	FORM FEED (FF)
13	0D	015	CARRIAGE RETURN (CR)
32	20	040	SPACJA (SP)

We wszystkich kontekstach, z wyjątkiem komentarzy, ciągów znaków, wpisów w tabelach odsyłaczy i strumieni, PDF traktuje każdą sekwencję kolejnych białych znaków jako jeden znak.

Znaki CARRIAGE RETURN (0Dh) i LINE FEED (0Ah), zwane również znakami nowej linii, są traktowane jako znaczniki końca linii (EOL). Kombinacja CARRIAGE RETURN, po której następuje bezpośrednio LINE FEED, jest traktowana jako jeden znacznik EOL. Znaczniki EOL są zazwyczaj traktowane tak samo, jak wszystkie inne białe znaki. Czasami jednak wymagany jest znacznik EOL, poprzedzający znacznik, który pojawia się na początku linii.

Cztery sekcje pliku PDF

Rysunek 1-2 ilustruje cztery sekcje pliku PDF: header, body, cross-reference table i trailer.



Header

Nagłówek pliku PDF zaczyna się od bajtu 0 pliku i składa się z co najmniej 8 bajtów, po których następuje znacznik końca linii. Te 8 bajtów służy do wyraźnej identyfikacji pliku w formacie PDF (% PDF-) i sugeruje numer wersji standardu, z którym jest zgodny plik (np. 1.4). Jeśli Twój plik PDF zawiera rzeczywiste dane binarne (a obecnie prawie wszystkie z nich to robią), pojawi się druga linia, która zaczyna się również od znaku komentarza PDF,% (PERCENT SIGN). Poniżej % w drugiej linii będą co najmniej cztery znaki, których wartości ASCII są większe niż 127. Chociaż dowolne cztery (lub więcej) wartości są w porządku, najczęściej używane są â âÏÓ (0xE2E3CFD3).

Druga linia polega na oszukiwaniu programów wykonujących wykrywanie ASCII w porównaniu z wykrywaniem binarnym, po prostu zliczając wartości ASCII wysokiego rzędu. Włączenie tych wartości zapewnia, że pliki PDF będą zawsze uważane za pliki binarne

Trailer

Na przeciwległym końcu pliku PDF z nagłówka można znaleźć zwiastun. Prosty przykład pokazano w przykładzie 1-12. Zwiastun jest przede wszystkim słownikiem z kluczami i wartościami, które dostarczają informacji na poziomie dokumentu, które są niezbędne do zrozumienia samego dokumentu.

Przykład 1-12. Prosty trailer

trailer

<<

/Size 23

/Root 5 0 R

/ID[<E3FEB541622C4F35B45539A690880C71><E3FEB541622C4F35B45539A690880C71>]

/Info 6 0 R

>>

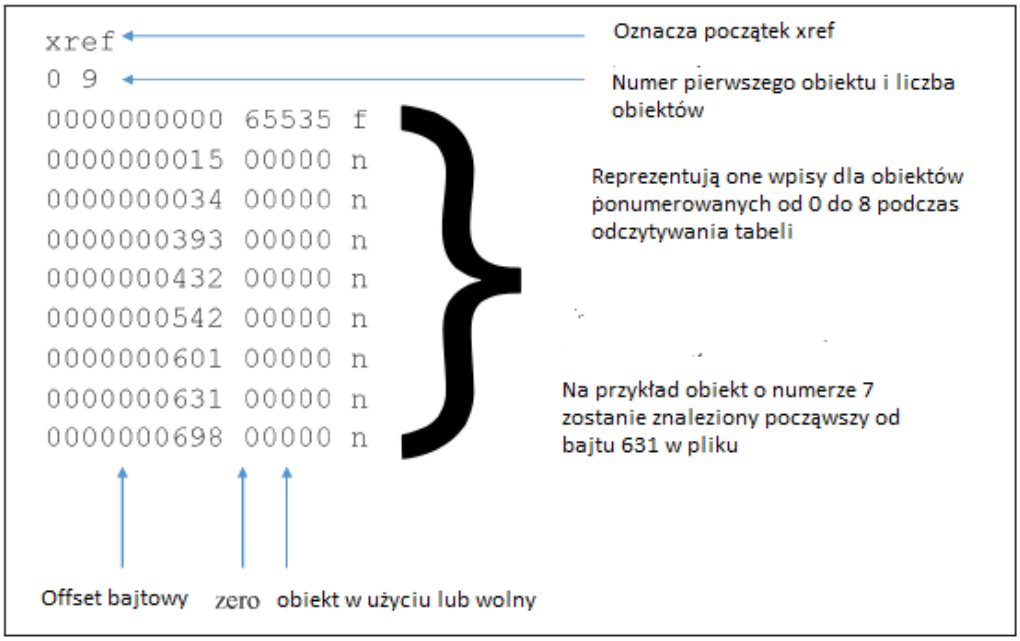
Dwa najważniejsze klucze i jedyne wymagane klucze to Size i Root. Klucz Size mówi nam, ile wpisów należy znaleźć w tabeli odsyłaczy poprzedzającej słownik zwiastunów. Kluczem do wartości jest słownik katalogu dokumentów, w którym rozpoczniesz wyszukiwanie wszystkich obiektów w pliku PDF. Inne popularne klucze w zwiastunie to klucz Encrypt , którego obecność szybko identyfikuje, że dany plik PDF został zaszyfrowany; klucz ID, który zapewnia dwa unikalne identyfikatory dokumentu; oraz klucz Info, który reprezentuje oryginalną metodę dostarczania metadanych na poziomie dokumentu.

Body

Ciało jest tam, gdzie wszystkie dziewięć typów obiektów, które składają się na sam dokument, znajduje się w pliku.

Cross-reference table

Tabela odsyłaczy jest prosta w koncepcji i implementacji, ale jest jednym z podstawowych atrybutów PDF. Ta tabela zawiera przesunięcie binarne od początku pliku dla każdego obiektu pośredniego w pliku, umożliwiając procesorowi PDF szybkie wyszukiwanie, a następnie odczytanie dowolnego obiektu w dowolnym momencie. Ten model dostępu losowego oznacza, że plik PDF można szybko otwierać i przetwarzać, bez konieczności ładowania całego dokumentu do pamięci. Ponadto nawigacja między stronami jest szybka, niezależnie od tego, jak duży jest „skok liczbowy” w numerach stron. Posiadanie tabeli odsyłaczy na końcu pliku zapewnia również dwie dodatkowe korzyści: utworzenie pliku PDF w jednym przebiegu (bez śledzenia wstecznego) jest możliwe, a obsługa przyrostowych aktualizacji dokumentu jest ułatwiona. Oryginalna forma (od PDF 1.0 do 1.4) tabeli odsyłaczy składa się z jednej lub więcej sekcji odsyłaczy, gdzie każda z tych sekcji jest serią wpisów (jedna linia na obiekt) z przesunięciem pliku obiektu, jego pokolenie i czy nadal jest w użyciu. Najpopularniejszy typ tabeli, pokazany na rysunku 1-3, ma tylko jedną sekcję z listą wszystkich obiektów



Ten typ tabeli odsyłaczy ma bardzo sztywny format, w którym pozycje kolumn są ustalone, a zera są wymagane.

Możesz zauważyć, że wartości liczb w drugiej kolumnie każdej linii tabeli odsyłaczy są zawsze zerowe, z wyjątkiem pierwszej, która wynosi 65535. Ta wartość, w połączeniu z f, daje wyraźne wskazanie, że obiekt o tym identyfikatorze jest nieprawidłowy. Ponieważ plik PDF może nigdy nie mieć obiektu o identyfikatorze 0, ten pierwszy wiersz zawsze wygląda tak, jak w tym przykładzie. Jednak, gdy plik PDF zawiera aktualizację przyrostową, może pojawić się sekcja odsyłacza, która wygląda jak ta z przykładu 1-13.

Przykład 1-13. Zaktualizowano sekcję odsyłacza

xref

0 1

0000000000 65535 f

4 1

0000000000 00001 f

6 2

0000014715 00000 n

0000018902 00000 n

10 1

0000019077 00000 n

zwiastun filmu

<< / Rozmiar 18 / Korzeń 9 0 R / Poprzednia 14207

/ ID [<86E7D6BF23F4475FB9DEED829A563FA7> <507D41DDE6C24F52AC1EE1328E44ED26>] >>

W miarę powiększania się dokumentów PDF stało się jasne, że posiadanie tego bardzo gadatliwego (i nieskompresowanego) formatu było problemem wymagającym adresowania. W związku z tym w przypadku PDF 1.5 został wprowadzony nowy typ systemu przechowywania odniesień zwany strumieniami odsyłaczy (ponieważ dane są przechowywane jako standardowy obiekt strumienia). Nowy format, oprócz możliwości kompresji, jest bardziej kompaktowy i obsługuje pliki o rozmiarze większym niż 10 gigabajtów, a jednocześnie zapewnia inne typy przyszłej rozbudowy (które nie zostały jeszcze wykorzystane). Oprócz przeniesienia tabeli odsyłaczy do strumienia, ten nowy system umożliwił również przechowywanie kolekcji obiektów pośrednich wewnątrz innego specjalnego typu strumienia zwanego strumieniem obiektów. Dzięki inteligentnemu podziałowi obiektów na wiele strumieni możliwe jest zoptymalizowanie czasu ładowania i / lub zużycia pamięci dla pliku PDF. Przykład 1-14 pokazuje, jak wygląda strumień odsyłaczy.

Przykład 1-14. Wewnątrz strumienia odsyłaczy

stream

01 0E8A 0% Wpis dla obiektu 2 (0x0E8A = 3722)

02 0002 00% Wpis dla obiektu 3 (w strumieniu obiektu 2, indeks 0)

02 0002 01% Wpis dla obiektu 4 (w strumieniu obiektu 2, indeks 1)

02 0002 02%. . .

02 0002 03

02 0002 04

02 0002 05

02 0002 06

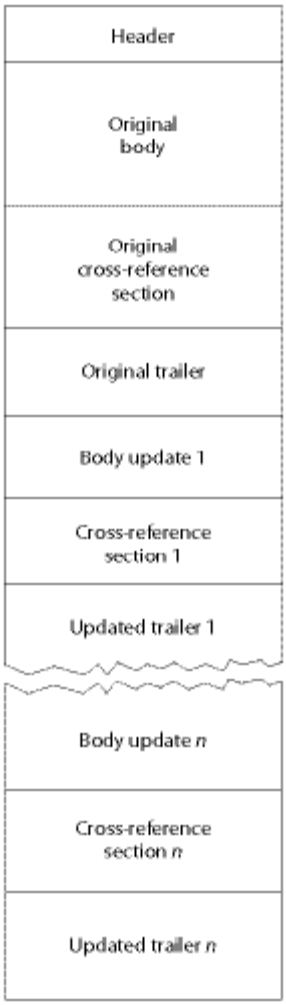
02 0002 07% Wejście dla obiektu 10 (w strumieniu obiektu 2, indeks 7)

01 1323 0% Wpis dla obiektu 11 (0x1323 = 4899)

endstream

Aktualizacja przyrostowa

Jak wspomniano wcześniej, jedną z kluczowych cech pliku PDF, która była możliwa dzięki użyciu zwiastuna i tabeli odsyłaczy na końcu dokumentu, jest koncepcja przyrostowej aktualizacji. Ponieważ zmienione obiekty są zapisywane na końcu pliku PDF, jak pokazano na rysunku 1-4, zapisywanie modyfikacji jest bardzo szybkie, ponieważ nie ma potrzeby czytania i przetwarzania każdego obiektu.



Każda sekcja odsyłacza za pierwszymi wskazuje wstecz do sekcji odsyłaczy poprzedzającej ją za pomocą klawisza Prev w słowniku zwiastunów, a następnie wyświetla tylko nowe, zmienione lub usunięte obiekty w nowej tabeli, jak pokazano w przykładzie 1-13. Chociaż widzowie w rzeczywistości nie oferują tej funkcji (za wyjątkiem podpisu cyfrowego), użycie aktualizacji przyrostowych oznacza, że można obsługiwać wiele cofnięć w przypadku zapisywania granic. Jednak to również wiąże się z niebezpieczeństwem ze strony ludzi, którzy przeglądają twoje (niepobrane) śmieci. Nawet jeśli pomyślałeś, że usunąłeś coś z pliku, nadal może tam być, jeśli zamiast pełnej składki zastosowano aktualizację przyrostową. Przy stopniowym aktualizowaniu pliku PDF bardzo ważne jest to ,że nie łączysz klasycznych odsyłaczy ze strumieniami odsyłaczy.

Jakikolwiek rodzaj odsyłacza jest używany w oryginale, musi być również użyty w sekcjach aktualizacji. Jeśli je miksujesz, czytnik PDF może zignorować aktualizacje.

Linearyzacja

Jak widzieliście, posiadanie tabeli odsyłaczy na końcu pliku oferuje różne korzyści. Istnieje jednak również jedna duża wada, a wtedy plik PDF musi zostać odczytany przez „interfejs strumieniowy”, taki jak strumień HTTP w przeglądarce internetowej. W w takim przypadku normalny plik PDF musiałby zostać pobrany w całości, zanim nawet pojedyncza strona mogłaby zostać odczytana - nie jest to wspaniałe doświadczenie użytkownika.

Aby rozwiązać ten problem, plik PDF udostępnia funkcję o nazwie linearyzacja (ISO 32000-1: 2008, załącznik F), ale lepiej znana jako „Szybki podgląd sieciowy”.

Zlinearyzowany plik różni się od standardowego pliku PDF na trzy sposoby:

1. Obiekty w pliku są uporządkowane w specjalny sposób, tak że wszystkie obiekty dla danej strony są zgrupowane razem, a następnie zorganizowane w kolejności stron (np.obiekty na stronie 1, następnie obiekty na stronie 2 itd.).

2. Obecny jest specjalny słownik parametrów linearyzacji, bezpośrednio po nagłówku, który identyfikuje plik jako linearyzowany i zawiera różne informacje potrzebne do przetworzenia go jako takiego.

3. Częściowa tabela odsyłaczy i przyczepa są umieszczone na początku pliku, aby umożliwić dostęp do wszystkich obiektów wymaganych przez obiekt główny oraz do obiektów reprezentujących pierwszą wyświetlaną stronę (zwykle 1).

Oczywiście, tak jak w przypadku standardowego pliku PDF, obiekty są nadal przywoływane w ten sam sposób, nadal umożliwiając losowy dostęp do dowolnego obiektu poprzez tabelę odsyłaczy. Fragment linearyzowanego pliku PDF przedstawiono w przykładzie 1-15.

Przykład 1-15. Zlinearyzowany fragment PDF

% PDF-1.7

%% EOF

8 0 obj

<< / Linearyzowany 1 / L 7546 / O 10 / E 4079 / N 1 / T 7272 / H [456 176] >>

endobj

xref

8 8

0000000016 00000 n

0000000632 00000 n

0000000800 00000 n

0000001092 00000 n

0000001127 00000 n

0000001318 00000 n

0000003966 00000 n

0000000456 00000 n

zwiastun filmu

<< / Rozmiar 16 / Korzeń 9 0 R / Info 7 0 R / ID [<568899E9010A45B5A30E98293

C6DCD1D> <068A37E2007240EF9D346D00AD08F696>] / Poprzedni 7264 >>

startxref

0

%% EOF

% obiektów ciała tu idzie ..

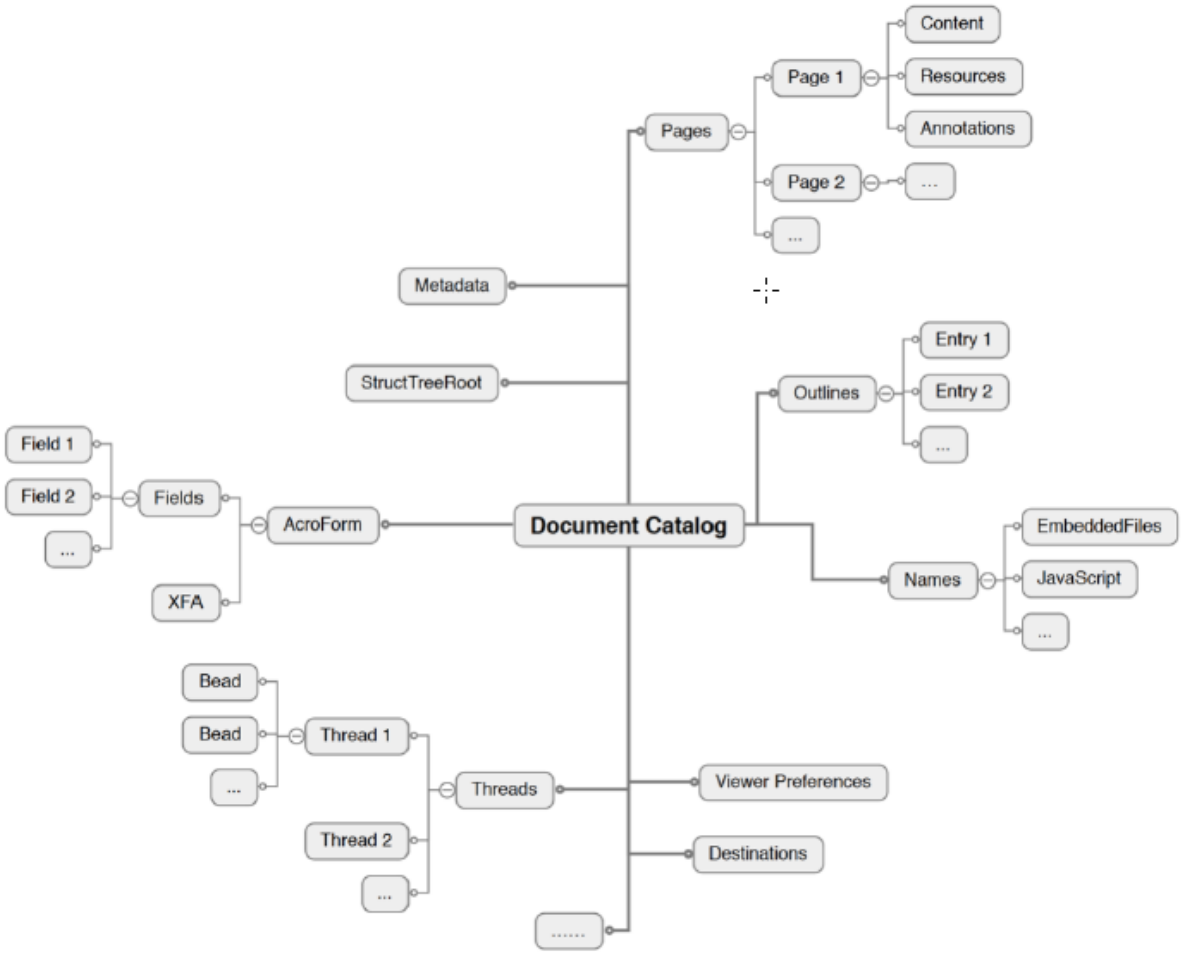
Mieszanie linearyzacji i przyrostowych aktualizacji może przynieść nieoczekiwane wyniki, ponieważ zlinearyzowana tabela odniesień będzie używana zamiast zaktualizowanych wersji, które istnieją tylko na końcu pliku. Dlatego ważne jest, aby pliki przeznaczone do użytku online były w pełni zapisane, aby usunąć aktualizacje i zapewnić prawidłowe tabele linearyzacji.

Struktura dokumentu

Teraz, gdy dowiedziałeś się o różnych obiektach w pliku PDF i sposobie ich łączenia, tworząc fizyczny układ / strukturę plików, nadszedł czas, aby połączyć je w celu utworzenia rzeczywistego dokumentu.

Słownik katalogu

Dokument PDF to zbiór obiektów, począwszy od obiektu głównego



Powodem dla którego nazywanym jest rootem jest to, że jeśli myślisz o obiektach w pliku PDF jako drzewie (lub wykresie skierowanym), obiekt ten znajduje się w katalogu głównym drzewa / wykresu. Z

tego obiektu można znaleźć wszystkie inne obiekty potrzebne do przetwarzania stron PDF i ich zawartości.

Root jest zawsze obiektem typu Catalog i jest znany jako słownik katalogu dokumentów. Ma dwa wymagane klucze:

1. Type, którego wartością zawsze będzie nazwa obiektu Katalog

2. Pages, których wartość jest pośrednim odniesieniem do drzewa stron

Mimo, że możliwość dotarcia do stron PDF jest oczywiście ważna, istnieje ponad dwa tuziny opcjonalnych kluczy, które mogą być również obecne (patrz ISO 32000-1: 2008, Tabela 28). Są to informacje na poziomie dokumentu, w tym takie rzeczy jak:

• Metadane oparte na XML

• OpenActions

• Formularze do wypełnienia

• Treść opcjonalna

• Struktura logiczna i znaczniki

Przykład 1-16 przedstawia przykład obiektu katalogu

Przykład 1-16. Obiekt katalogu

<<

/Type /Catalog

/Pages 533 0 R

/Metadata 537 0 R

/PageLabels 531 0 R

/OpenAction 540 0 R

/AcroForm 541 0 R

/Names 542 0 R

/PageLayout /SinglePage

/ViewerPreferences << /DisplayDocTitle true >>

>>

Przyjrzyjmy się kilku kluczom (i ich wartościom), które mogą okazać się przydatne do uwzględnienia w plikach PDF, aby poprawić wrażenia użytkownika:

Układ strony

Klucz PageLayout służy do informowania widza, jak wyświetlać strony PDF. Jego wartość jest obiektem nazwy. Aby wyświetlać je pojedynczo, użyj wartości SinglePage lub, jeśli chcesz, aby wszystkie strony znajdowały się w długiej, ciągłej kolumnie, użyj wartości OneColumn. Istnieją także wartości, które

można określić dla dwóch stron na raz (czasami nazywane spreadami), w zależności od tego, gdzie mają się pojawić nieparzyste strony: TwoPageLeft i TwoPageRight.

PageMode

Oprócz wyświetlania zawartości strony PDF możesz chcieć, aby niektóre elementy nawigacyjne pliku PDF były natychmiast dostępne dla użytkownika. Na przykład możesz chcieć, aby zakładki lub kontury były widoczne. Wartość klucza PageMode, który jest obiektem nazwy, określa, jakie (jeśli w ogóle) dodatkowe elementy są wyświetlane, takie jak linie UseOut, UseThumbs lub UseAttachments.

ViewerPreferences

W przeciwieństwie do poprzednich dwóch przykładów, gdzie wartości kluczy były obiektami nazw, klucz ViewerPreferences ma wartość, która jest słownikiem preferencji przeglądarki. Wśród wielu kluczy dostępnych do użycia w słowniku preferencji przeglądarki najważniejszy z nich jest pokazany w poprzednim przykładzie: DisplayDocTitle. Obecność tej wartości z wartością true powoduje, że przeglądarka plików PDF nie wyświetla nazwy pliku dokumentu na pasku tytułu okna, jak pokazano na rysunku 1-6,

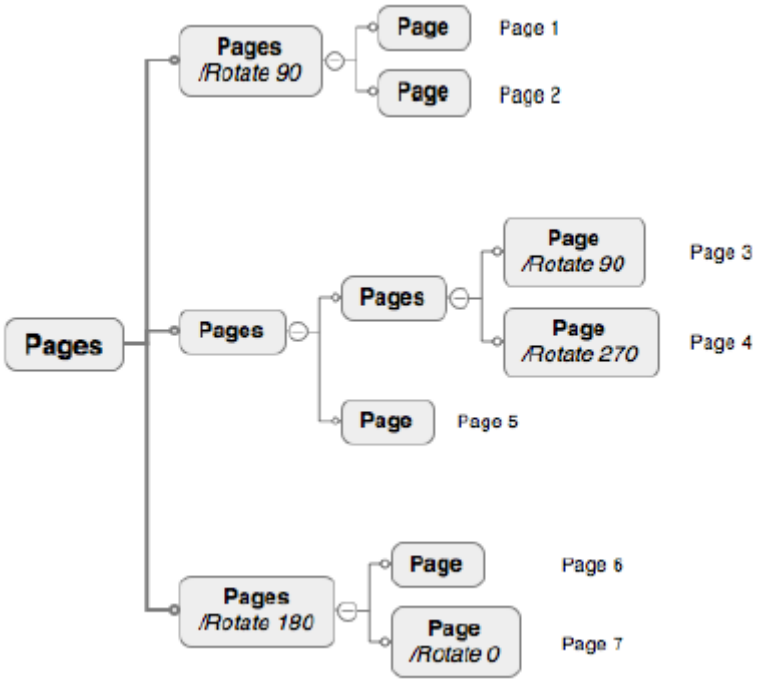
ale jego prawdziwy tytuł, jak pokazano na rysunku 1-7

Drzewo stron

Strony w pliku PDF są dostępne za pośrednictwem drzewa stron, które określa kolejność stron. Drzewo stron jest zwykle implementowane jako drzewo zrównoważone, ale może być także prostą tablicą stron.

Zaleca się, aby w jednym liściu drzewa było nie więcej niż 25–50 stron. Oznacza to, że każdy dokument większy niż ten nie powinien używać pojedynczej tablicy, ale zamiast tego powinien budować zrównoważone drzewo. Powodem tego jest to, że konstrukcja drzewa zrównoważonego oznacza, że na urządzeniach z ograniczoną pamięcią lub zasobami możliwe jest znalezienie dowolnej konkretnej strony bez konieczności ładowania całej tablicy, a następnie sekwencyjnego uzyskiwania dostępu do każdej strony w tablicy.

Jak widać na rysunku 1-8,



w drzewie strony występują dwa typy węzłów: węzły pośrednie (typu Pages) i węzły końcowe lub liściowe (typu Page). Węzły pośrednie, które zawierają początkowy węzeł drzewa, zapewniają pośrednie odniesienia do swoich rodziców (jeśli istnieją) i dzieci, a także liczbę węzłów liści w poszczególnych gałęziach drzewa. Węzeł liścia jest rzeczywistym obiektem Page.

Część rysunku 1-8, przedstawiona w składni PDF, może wyglądać jak przykład 1-17.

Przykład 1-17. Obiekty tworzące przykładowe drzewo strony

2 0 obj

<<

/Type /Pages

/Kids[4 0 R]

/Count 3

>>

endobj

4 0 obj

<<

/Type /Pages

/Parent 2 0 R

/Rotate 90

/Kids[5 0 R 6 0 R]

/Count 3

>>

endobj

5 0 obj

<<

/Type /Page

% Additional entries describing the attributes of Page 1

>>

endobj

6 0 obj

<<

/Type /Page

% Additional entries describing the attributes of Page 2

>>

endobj

Pages

Jak przed chwilą omówiono, każdy węzeł liścia w drzewie strony reprezentuje obiekt strony. Obiekt strony jest słownikiem, którego klucz typu ma wartość Page; zawiera również kilka innych wymaganych kluczy i może zawierać kilkanaście opcjonalnych kluczy i ich wartości. Przykład 1-18 pokazuje kilka słowników przykładowej strony.

Przykład 1-18. Dwa słowniki przykładowej strony

% simplest valid page object, with the four required keys

<<

/Type /Page

/Parent 2 0 R

/MediaBox [0 0 612 792] % Page Size == 8.5 x 11 in (612/72 x 792/72)

/Resources <<>>

>>

% a real-world page object

<<

/Type /Page

/Parent 532 0 R

/MediaBox [0 0 612 792]

/CropBox [0 0 500 600]

/Contents 564 0 R

/Resources <<

/ExtGState << /GS0 571 0 R /GS1 572 0 R >>

/Font << /T1_0 566 0 R >>

/XObject << /Im0 577 0 R >>

>>

/Trans << /S /Dissolve >>

/Rotate 90

/Annots 549 0 R

/AA << /C 578 0 R /O 579 0 R >>

>>

Istnieje kilka kluczy do wskazania tutaj, z których część zagłębimy się w przyszłości:

Content

Jeśli nie chcesz pustych stron w swoim pliku PDF, jest to najważniejszy klucz w słowniku strony, ponieważ wskazuje strumień treści zawierający instrukcje dotyczące tego, co ma być narysowane na stronie.

Rotate

Za pomocą tego klawisza można obracać stronę o 90 stopni. Jednak chociaż jest to poprawna i prawidłowa część pliku PDF, jest często ignorowana przez wiele narzędzi niższego rzędu. Dlatego rozważ użycie stron o odpowiedniej wielkości i (w razie potrzeby) treści przekształconych.

Trans

Jeśli jest obecny, ten klucz mówi widzowi, że podczas wyświetlania strony w „stylu prezentacji" powinien użyć zdefiniowanego przejścia podczas przechodzenia na tę stronę z poprzedniej. Szczegóły dotyczące wartości tego klucza można znaleźć w ISO 32000-1: 2008, 12.4.4.

Annots

Wartość tego klucza to tablica wszystkich adnotacji znajdujących się nad treścią strony.

AA

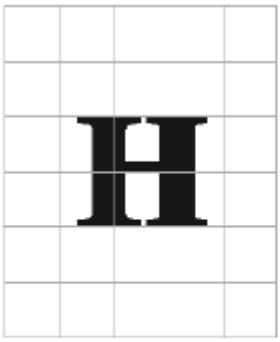
Akcje reprezentują rzeczy, które widz zrobi po niejawnych działaniach użytkownika, takich jak otwieranie lub zamykanie strony

Resource

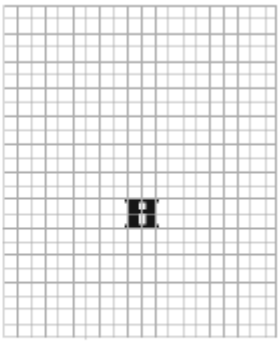
Służą one do uzupełnienia definicji obiektów graficznych, takich jak czcionka lub kolor, które mają być używane, aby rysować na stronie.

Jednostki PDF

Często podczas pracy z systemami graficznymi pracujesz bezpośrednio przy rozdzielczości urządzenia wyjściowego, takiej jak ekran 72 lub 90 dpi (punkty na cal) lub drukarka 600 dpi. Nazywa się to przestrzenią urządzenia .

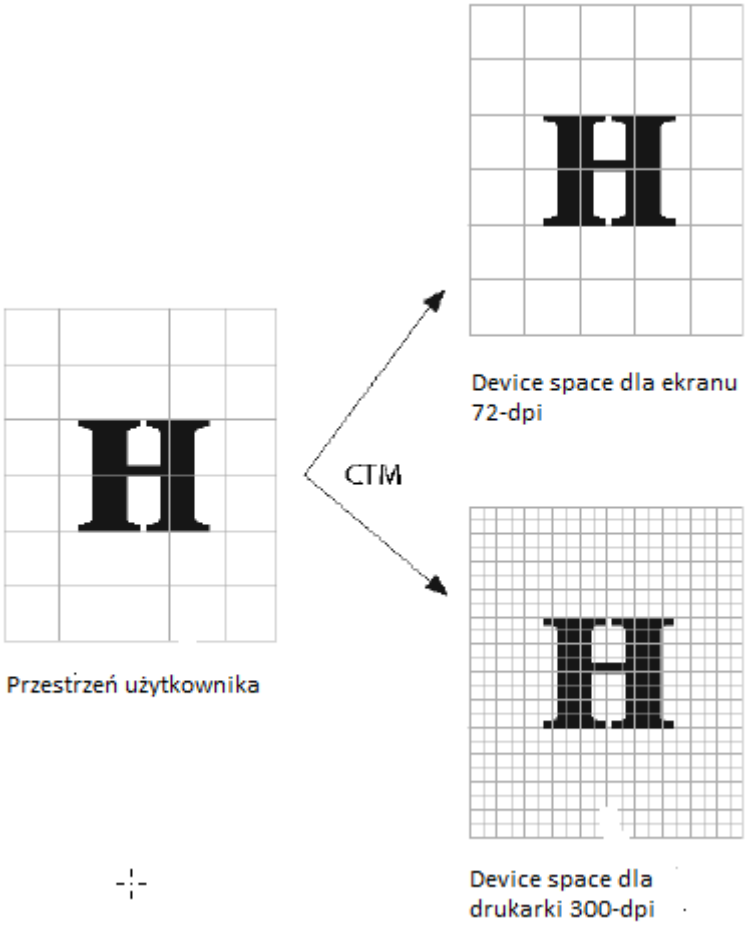


Device space dla ekranu 72-dpi



Device space dla drukarki 300-dpi

Jednak, jak pokazuje ten rysunek, jeśli chcesz, aby obiekt o tym samym rozmiarze pojawiał się niezależnie od cech urządzenia, musisz pracować w czymś innym niż przestrzeń urządzenia. W PDF, jest to nazywane przestrzenią użytkownika, pozostaje taki sam niezależnie od urządzenia wyjściowego



Domyślnie przestrzeń użytkownika wynosi 72 jednostki użytkownika na cal (zwane również „punktami”), a początek znajduje się w lewym dolnym rogu. Istnieje możliwość zmiany liczby jednostek

użytkownika poprzez użycie transformacji współrzędnych w treści strony lub obecność klucza UserUnit w słowniku strony. Początkiem układu współrzędnych będzie zawsze [0 0], ale może to nie odpowiadać dolnemu narożnikowi widocznej strony PDF, w zależności od wartości pól strony

Przykład 1-19. Przykładowe strony używające klucza UserUnit

```
2 0 obj
<<
/Type /Pages
/Kids[ 3 0 R 4 0 R 5 0 R ]
/Count 3
>>
endobj
3 0 obj
<<
/Type /Page
/Parent 2 0 R
/UserUnit 1.0 % default of 72 units/inch
/MediaBox [ 0 0 612 792 ] % 8.5 x 11 inches
% more keys here...
>>
endobj
4 0 obj
<<
/Type /Page
/Parent 2 0 R
/UserUnit 2.0 % 144 units/inch (2 * 72)
/MediaBox [ 0 0 612 792 ] % 17 x 22 inches
% more keys here...
>>
endobj
5 0 obj
<<
/Type /Page
```

/Parent 2 0 R

/UserUnit 3.14159 % something funny but perfectly valid

/MediaBox [0 0 612 792] % 26.70 x 34.56 inches

% more keys here...

>>

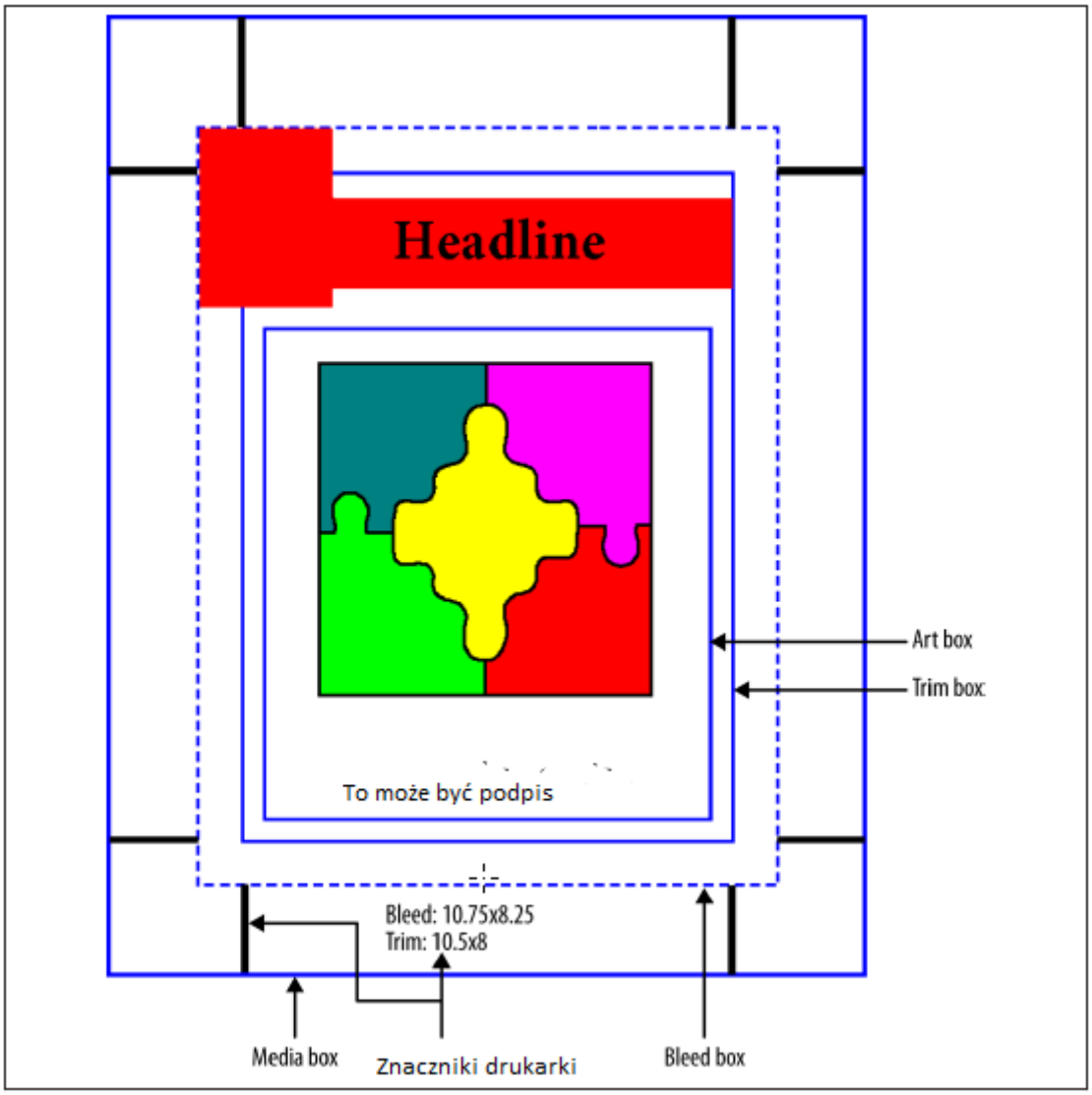
endobj

Prostokąty i pola

Podczas opisywania prostokąta w składni PDF używana jest tablica czterech liczb. Kolejność liczb to: lewy, dolny, szerokość, wysokość. Znajdziesz prostokąty używane w różnych miejscach w składni PDF, ale typ rect, którego będziesz używać najczęściej, to definiowanie rozmiarów różnych regionów na stronie - pól stron. Każde z pięciu pól strony (ISO 32000-1: 2008, 14.11.2) przedstawia prostokątny widok obszaru („pole”) dla elementów graficznych, które są rysowane na stronie, bezpośrednio lub za pomocą adnotacji. Cztery liczby w tablicy są zawsze w jednostkach użytkownika, jednostkach przestrzeni użytkownika. Ponieważ reprezentuje widok w układzie współrzędnych strony, prostokąt nie musi mieć dolnego lewego rogu przy [0 0]. MediaBox strony definiuje rozmiar strony, na której będzie rysowany. Zwykle jest to odpowiednik zwykłego rozmiaru papieru, takiego jak A4 (21 x 29,7 cm), chociaż może być dowolnego rozmiaru.

MediaBox o wartości [0 0 612 792], w domyślnych (1.0) jednostkach użytkownika, jest równoważny z kawałkiem papieru wielkości USA Letter (8,5 * 72 = 612; 11 * 72 = 792).

Oprócz MediaBoxu, na stronie znajdują się cztery inne pola stron. Są one pokazane na rysunku 1-11



Pole CropBox służy do instruowania przeglądarki plików PDF o rzeczywistym widocznym obszarze strony, gdy jest wyświetlany lub drukowany. Jest to używane głównie wtedy, gdy masz zawartość na stronie, której nie chcesz, aby użytkownik widział, więc „wyciąć” ją. W przeciwieństwie do edytora obrazów, zastosowanie pola CropBox niczego nie usuwa; po prostu ukrywa go poza widocznym obszarem

Chociaż CropBox może wykraczać poza MediaBox, przeglądarka plików PDF skutecznie przypisze wartości do MediaBox.

W branży poligraficznej TrimBox ma podobny cel. Określa, gdzie obcinarka przycina (wycina) papier po wydrukowaniu, usuwając w ten sposób zawartość poza TrimBoxem z ostatniego elementu. Jest używany, gdy masz coś, co chcesz dostać się aż do krawędzi papieru, bez żadnej białej przestrzeni lub luki. Aby to zadziałało, prawie zawsze istnieje powiązany BleedBox, który definiuje obszar poza obszarem przycinania, w którym zawartość „wykrwawia się”, dzięki czemu można go właściwie przyciąć. Ostatnie pole, zwane ArtBox, prawie nigdy nie jest używane. Pierwotnie miał być używany do reprezentowania obszaru, który obejmował tylko „grafikę” strony, z której można korzystać aby zmienić przeznaczenie, umieścić lub nałożyć na inny arkusz. Jednak nigdy tak naprawdę się nie przyjęło i nie powinieneś po prostu używać ich w swoich dokumentach.

Dziedziczenie

Jak widzieliście w „Pages”, niektóre wartości, które normalnie byłyby obecne w obiekcie strony, mogą również występować w węzłach pośrednich (obiekty stron). Gdy tak się dzieje, wartości te są dziedziczone przez wszystkie dzieci tego węzła, chyba że zdecydują się je zastąpić. Na przykład, jeśli wszystkie strony dokumentu mają ten sam rozmiar, możesz umieścić klucz MediaBox w węźle głównym drzewa stron.Nie wszystkie klucze, które mogą być obecne w obiekcie strony, mogą być dziedziczone, tylko te zidentyfikowane jako takie w ISO 32000-1: 2008, Tabela 30.

Zlinearyzowany PDF nie może używać dziedziczenia. Wszystkie wartości muszą być określone bezpośrednio w każdym obiekcie strony.

Słownik nazw

Niektóre typy obiektów w pliku PDF można odwoływać się raczej do nazwy niż do odwołania do obiektu. Ta korespondencja między nazwami i obiektami jest ustalana przy użyciu czegoś, co nazywa się słownikiem nazw dokumentów. Słownik nazw jest określany przez dołączenie klucza Nazwy w słowniku katalogu dokumentów. Każdy ze zdefiniowanych kluczy, które mogą być obecne w tym słowniku, oznacza korzeń drzewa nazw, które definiuje nazwy dla tej konkretnej kategorii obiektów.

Model obrazowania PDF

W tej części rozpoczniesz eksplorację modelu obrazowania PDF, czyli różnych rodzajów operacji graficznych, które mogą być wykonywane na stronach dokumentu. Dowiesz się nie tylko języka używanego do opisu tych operacji, ale także różnych towarzyszących mu koncepcji graficznych i obrazowych.

Strumienie treści

Jak opisano wcześniej, plik PDF składa się z jednej lub więcej stron (o ustalonym rozmiarze), a widoczne elementy na każdej stronie pochodzą z zawartości strony lub serii adnotacji znajdujących się na wierzchu (widoczne) treści. W tej części omówiono zawartość strony. Treść strony jest opisana przy użyciu specjalnej składni tekstowej (powiązanej, ale różnej od składni pliku PDF, o której dowiedziałeś się w części 1), która jest przechowywana w pliku PDF wewnątrz specjalnego typu obiektu strumienia zwanego strumieniem treści. Składnia treści pochodzi z języka Adobe Postscript i składa się z szeregu operatorów i ich operandów, gdzie każdy operand może być wyrażony jako standardowy obiekt PDF. Prosty przykład rysowania prostokąta wypełnionego kolorem niebieskim podano w przykładzie 2-1.

Przykład 2-1. Rysowanie prostego prostokąta

```
0 0 1 rg          % ustawia kolor na niebieski w RGB
0 0 100 100 re    % rysuje prostokąt 100x100, a dolny lewy 0,0
F                 % wypełnia
```

Wkrótce przyjrzymy się różnym operatorom, ale na razie najważniejszą rzeczą, którą należy wziąć z przykładu, jest to, że składnia jest wyrażona w odwrotnej notacji polskiej, gdzie operator podąża za operandami (jeśli istnieją). Drugą rzeczą, którą powinieneś pamiętać o języku składni strony, jest to, że w przeciwieństwie do Postscripta nie jest to prawdziwy język programowania, ponieważ nie ma zmiennych, pętli ani rozgałęzień warunkowych. Operatorzy treści mogą być logicznie podzieleni na trzy kategorie. Najważniejszymi z nich są oczywiście operatory rysunków, które powodują, że coś jest faktycznie rysowane na stronie. Operatory rysunku nie byłyby jednak w pełni przydatni bez możliwości ustawiania atrybutów stanu graficznego, który określa wygląd rysunku (np. Kolor lub szerokość pióra). Wreszcie istnieje zestaw operatorów nazywanych operatorami zawartości oznaczonej, które umożliwiają stosowanie specjalnych właściwości do grupy operatorów.

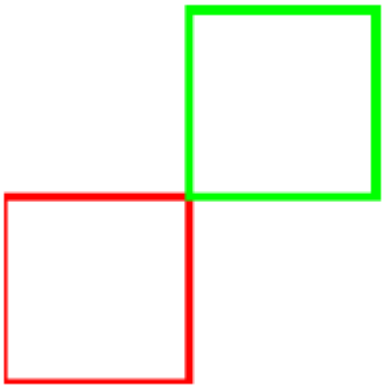
Stan graficzny

Jak już wspomniano, rysowanie nie byłoby przydatne, gdyby nie można było ustawić wszystkich atrybutów rysunku. Dlatego zaczniemy od stanu graficznego i jego operatorów. Możesz myśleć o stanie graficznym jako klasie lub strukturze z elementami lub właściwościami i powiązanymi ustawnikami do zmiany wartości. Nie ma żadnych getterów, ponieważ w składni treści nie ma zmiennych ani warunkowych przypisujących wartości lub wykonujących dowolne operacje. Jest to niezwykle ważne, ponieważ oznacza, że nie ma bezpośredniego sposobu na zmianę niektórych wartości w stanie graficznym, a następnie przywrócenie ich do poprzedniego stanu. Możesz pomyśleć, że oznacza to, że albo musisz narysować wszystkie „podobne obiekty” razem, albo zrobić wiele „resetowania”. Na

szczęście nie musisz tego robić! Procesor PDF jest wymagany do utrzymania stosu (w tradycyjnym sensie programowania) stanu graficznego, aby strumień treści mógł wypychać nowe stany na lub wyłączać ukończone stany. W ten sposób możesz zapisać stan, coś zrobić, a następnie powrócić do poprzedniego stanu. Operatorami do tego celu są q i Q.

Przykład 2-2. Rysowanie dwóch prostokątów

```
4 w             % ustawia szerokość linii na 4 dla wszystkich obiektów
q               % naciśnij / zapisz stan
1 0 0 RG        % ustawia kolor obrysu na czerwony
0 0 100 100 re  % rysuje prostokąt 100x100 z lewym dolnym punktem 0,0
S               % skok o szerokości 4 jednostek
P               % pop stan
q               % naciśnij / zapisz stan
0 1 0 RG        % ustaw kolor obrysu na zielony
100 100 100 100 re% rysuje prostokąt 100x100 z lewą dolną liczbą 100 100
S               % skok o szerokości 4 jednostek
P               % pop stan
```

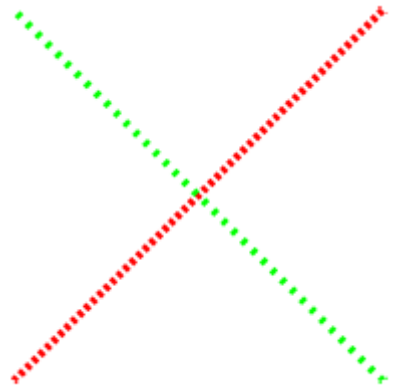


W tym przykładzie użyliśmy operatora, którego wcześniej nie widziałeś (w), aby ustawić szerokość pióra w stanie graficznym. Jak można sobie wyobrazić, jest to typowa operacja. Ponadto istnieje kilka innych typów atrybutów, które można ustawić podczas pracy z liniami (lub głaskaniem dowolnego kształtu), w tym wzorami kreski i co się dzieje, gdy dwie linie łączą się lub łączą. Przykład 2-3 to przykład rysowania dwóch linii przerywanych.

Przykład 2-3. Rysowanie linii przerywanych

```
8 w
q
1 0 0 RG
[5] 0 d                 % prosty wzór 5 na, 5 wyłączony
```

```
100 100 mln                 % przejść do 100 100
500 500 l                   % narysuj linię do 500,500
S
q
q
0 1 0 RG
[5 10] 0 d                  % 5 na, 10 z, wzór kreski
500 100 m                   % przejdzie do 500,100
100 500 l                   % narysuj linię do 100,500
S
Q
```



Przykład 2-4 przedstawia różne typy złączeń linii i zakrętek (końców linii).

Przykład 2-4. Różne połączenia linii i szczyty

```
q
0 1 0 RG
10 w
1 j                         % ustawia linię na rundę
1 J                         % ustawia zaokrąglenie linii
100 500 m
150 500 l
150 450 l
200 450 l
200 400 l
```

250 400 l

250 350 l

300 350 l

300 300 l

350 300 l

350 250 l

400 250 l

400 200 l

450 200 l

450 150 l

500 150 l

500 100 l

S

Q

q

0 1 1 RG

15 w

2 j % ustaw łączenie linii na fazę

50 500 m

100 550 l

500 550 l

550 500 l

550 100 l

500 50 l

100 50 l

50 100 l

h % zapewnia, że kształt ma kształt zamknięty

S

Q

Painter Model

Jeśli spojrzysz na poprzednie przykłady, zobaczysz, że sposób rysowania kształtów (lub ścieżek, tak jak są nazywane w terminach PDF) polega na zdefiniowaniu lub skonstruowaniu ścieżki, a następnie obrysu (S), wypełnienia, (F / f) lub obu (B) ścieżek. Każda ścieżka jest narysowana w kolejności, w jakiej pojawia się w strumieniu treści, w formie operacji „pierwsze weszło, pierwsze wyszło” (FIFO). Oznacza to, że jeśli chcesz narysować jedną ścieżkę nad drugą, po prostu ją narysuj. Połączenie tych dwóch atrybutów jest zwykle nazywane modelem malarza. Pinter Model jest również używany przez systemy graficzne, takie jak SVG i Apple's Quartz 2D. Przykład 2-5 ilustruje, jak to działa.

Przykład 2-5. Trzy nakładające się prostokąty

```
1 0 0 rg

209 426 -114 124 re

F

0 1 0 rg

237 401 -114 124 re

F

0 0 1 rg

272 373 -114 124 re

F
```

Otwarte i zamknięte ścieżki

Innym aspektem modelu jest to, że kształty mogą być otwarte lub zamknięte, co określa, w jaki sposób różne operacje głaskania i wypełniania kończą się, gdy osiągną koniec ścieżki. Rozważ ścieżkę podobną do tej z przykładu 2-6, która składa się z przejścia do punktu A, następnie linii do punktu B i innej linii do punktu C.

Przykład 2-6. Prosta ścieżka

```
100 100 m

100 200 l

200 200 l
```

Gdybyś używał operatora S, który do tej pory widziałeś w naszych przykładach, narysowałby linię w kształcie litery L, ponieważ po prostu obrysowuje ścieżkę zgodnie z definicją. Jeśli jednak użyjesz operatora s, zamiast tego będziesz miał trójkąt, ponieważ ścieżka zostanie zamknięta (tj. Połączona z ostatniego punktu z powrotem do pierwszego punktu), a następnie obrysowana. Możesz jawnie

zamknąć ścieżkę za pomocą operatora h, a następnie użyć S, aby go obrysować, dzięki czemu możesz uważać go za miły skrót.

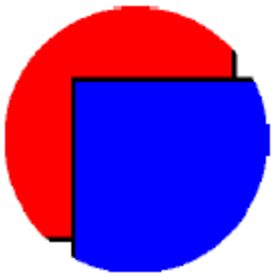
Przycinanie

Ostatnim aspektem modelu, który tutaj zbadamy, jest przycinanie. Przycinanie wykorzystuje ścieżkę (lub tekst), aby ograniczyć obszar rysowania z pełnej strony do dowolnego obszaru na tej stronie. Jest to najbardziej przydatne, gdy chcesz pokazać tylko niewielką część jakiegoś innego obiektu (zazwyczaj, ale niekoniecznie ograniczonego, grafiki rastrowej) dla określonego efektu wizualnego.

Użyj operatora W, aby oznaczyć ścieżkę jako ścieżkę przycinającą. Następnie możesz kontynuować wypełnianie lub obrysowywanie (za pomocą operatorów, które już widziałeś) lub nie rysować za pomocą operatora n.

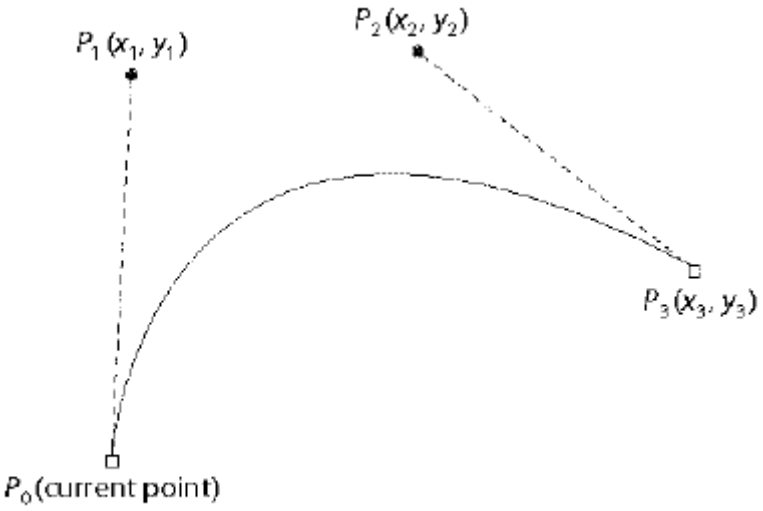
Przykład 2-7. Prostokąty przycięte kółkiem

```
q

27.738 78.358 m

27.738 56.318 45.604 38.452 67.643 38.452 c

67.643 38.452 l

89.681 38.452 107.548 56.318 107.548 78.358 c

107.548 78.358 l

107.548 100.396 89.681 118.262 67.643 118.262 c

67.643 118.262 l

45.604 118.262 27.738 100.396 27.738 78.358 c

W n %  przycięcie bez rzeczywistego rysunku

1 0 0 rg

0 0 0 RG

97.5 48.5 -98 97 re

B

0 0 1 rg

146.5 -0.5 -98 97 re

B

Q
```



Ścieżki rysunkowe

Chociaż z pewnością możesz zrobić dużo rysowania za pomocą trzech operatorów budowy ścieżek, które widziałeś do tej pory (m, l i re), możesz zrobić jeszcze więcej, jeśli możesz narysować coś, co nie jest proste - powiedz krzywą, na przykład. Operator c pozwala na narysowanie rodzaju krzywej zwanej krzywą Béziera, a konkretnie sześciennego Béziera. Taka krzywa jest definiowana od bieżącego punktu do punktu docelowego, używając dwóch innych punktów (znanych jako punkty kontrolne), aby zdefiniować kształt krzywej. Wymaga łącznie sześciu argumentów.



$x_1\ y_1\ x_2\ y_2\ x_3\ y_3$ **c**

Podczas gdy krzywe Béziera są niezwykle elastyczne i umożliwiają bardzo skomplikowane rysunki, mają podstawową wadę: nie można ich użyć do narysowania idealnego okręgu. Najbliższe, jakie można uzyskać, to połączenie czterech krzywych, które zaczynają się i kończą w czterech punktach krawędziowych okręgu, używając punktu kontrolnego o 0,6 jednostki od punktów końcowych.

Przykład 2-8 rysuje okrąg, a także pokazuje, w jaki sposób ścieżka może być obrysowana i wypełniona różnymi kolorami.

Przykład 2-8. Przerywany okrąg

1 0 0 rg

0 0 0.502 RG

2 w

[5 2] 0 d

350 200 m

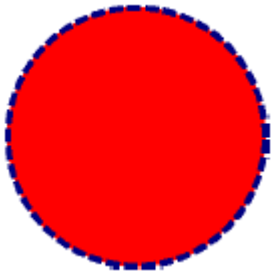
350 227.6 327.6 250 300 250 c

272.4 250 250 227.6 250 200 c

250 172.4 272.4 150 300 150 c

327.6 150 350 172.4 350 200 c

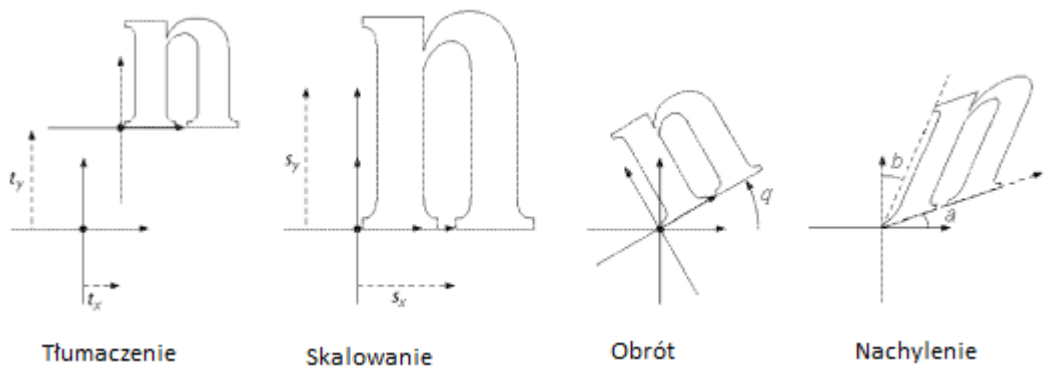
B



Transformacje

Jak omówiono w pierwszej części, każda strona definiuje obszar (w jednostkach użytkownika), w którym można umieścić treść. Zwykle początek (0,0) strony znajduje się w lewym dolnym rogu strony, a wartość y wzrasta w górę strony, a wartość x rośnie w prawo. Jest to zgodne ze standardową górną częścią układu współrzędnych kartezjańskich. Jednak w przypadku niektórych typów operacji rysowania można w pewien sposób dostosować (lub przekształcić, który jest właściwy termin) współrzędne - odwracanie / odbijanie, obracanie, skalowanie , itd. Część stanu graficznego, która śledzi to, nazywa się bieżącą macierzą transformacji (CTM). Aby zastosować transformację, należy użyć operatora cm, który pobiera sześć argumentów reprezentujących standardową macierz 3x2. Poniższy wykres przedstawia najpowszechniejsze typy transformacji i używane dla nich operandy.

Transformacja	**Operand**
Tłumaczenie	1 0 0 1 tx ty
Skalowanie	sx 0 0 sy 0 0
Obrót	cosQ sinQ -sinQ cosQ 0 0
Pochylenie	1 tanA tabB 1 0 0



Przykład 2-9 podaje kilka przykładów typowych przekształceń.

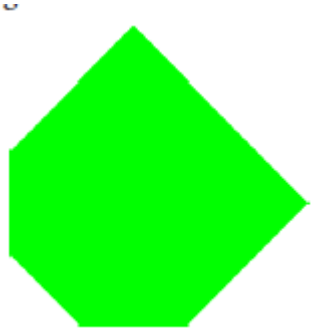
Przykład 2-9. Przekształcony kształt

```
q
1 0 0 rg
.50 0 0 .5 0 0 cm          % skala rysunku 50
0 0 100 100 re             % rysuje 100x100 rect od początku
F
Q
q
0 1 0 rg
1 0 0 1 100 100 cm         % przenieś początke do 100,100
0 0 100 100 re             % rysuje 100x100 rect w początku
F
Q
```

W niektórych przypadkach będziesz musiał wykonać wiele transformacji, operację zwaną konkatenacją macierzy. Najczęstszą operacją wymagającą konkatenacji jest rotacja. Nie tylko jest on bardziej złożony niż inne rodzaje transformacji, ponieważ wymaga użycia trygonometrii, ale wymaga wielu operacji. Powodem tego w wielu przypadkach jest to, że rotacja odbywa się zawsze wokół lewego dolnego rogu obiektu, a nie środka, czego oczekuje większość ludzi. Dlatego, aby obsłużyć obrót wokół punktu środkowego (lub dowolnego punktu), należy najpierw wykonać tłumaczenie, a następnie obrót, a następnie przekształcić z powrotem.

Przykład 2-10. Obrócony prostokąt

```
q
0 1 0 rg
1 0 0 1 50 50 cm                    % przenieś początek do 50,50 (punkt środkowy dla rect)
0.7071 0.7071 -0.7071 0.7071 0 0 cm  % 45 stopni obrotu
1 0 0 1 -50 -50 cm                  % przesuń początek z powrotem
0 0 100 100 re                      % rysuje 100x100 rect na początku
F
Q
```



Podstawowy kolor

W przykładach zawsze używaliśmy kolorów opartych na RGB (nazywanych DeviceRGB w terminach PDF). Jest to najbardziej popularny rodzaj kolorów, z którym zapoznają się użytkownicy, ponieważ używają go monitory komputerowe. W każdym przykładzie użyliśmy 1 lub 0 dla każdej wartości składowej koloru. W przeciwieństwie do innych systemów kolorów RGB, takich jak Windows GDI lub HTML, w PDF wartość dla każdego komponentu jest liczbą rzeczywistą od 0 do 1, a nie liczbą całkowitą od 0 do 255.

Jeśli chcesz przekonwertować, matematyka jest dość prosta: pdfValue = 255 - gdiValue / 255.

PDF obsługuje jednak dziesięć innych przestrzeni kolorów (lub modeli kolorów), których można użyć do określenia koloru obiektu. Ta sekcja przedstawia dwie inne przestrzenie kolorów urządzenia: DeviceGray i DeviceCMYK. Aby dowiedzieć się więcej o pozostałych ośmiu przestrzeniach kolorów, a także o tym, jak używać wzorów lub cieniowania do obrysu lub wypełnienia obiektu, patrz ISO 32000-1: 2008. W przypadku kolorów wykorzystujących tylko odcienie czerni i bieli (lub szarości) można użyć prostej przestrzeni kolorów DeviceGray. Operatory to g (dla wypełnienia) i G (dla obrysu) i przyjmują pojedynczy operand w zakresie od 0 (czarny) do 1 (biały). Ta przestrzeń kolorów powinna być używana zamiast czerni opartej na RGB, gdy tylko jest to możliwe, ponieważ powoduje to wyższą jakość operacji drukowania przy jednoczesnym oszczędzaniu miejsca w pliku PDF (3 bajty na operację vs. 8 bajtów na operację). Podczas gdy ekrany używają RGB do definiowania kolorów, drukarki używają innego modelu o nazwie CMYK dla pojemników z tuszem cyjan, magenta, żółtym i blacK, które są obecne w drukarce. Drukarki wyższej klasy mogą również mieć różne inne atramenty kolorowe, ale są używane w inny sposób. Aby opisać kolor w DeviceCMYK, używa się operatorów k lub K wraz z czterema argumentami dla każdego komponentu koloru.

Przykład 2-11 ilustruje użycie trzech przestrzeni kolorów.

Przykład 2-11. Trzy podstawowe przestrzenie kolorów

```
10 w
q
.5 G % 50                % szary,  w Gray
10 300 m
100 300 l
S
Q
q
1 0 0 RG                 % czerwony, w RGB
10 200 m
100 200 l
S
Q
q
1 0 0 0 K                % cyjan, w CMYK
10 100 m
100 100 l
S
Q
```

Operatory oznaczonych treści

W „Streamingach treści” wspomniano, że istniał zestaw operatorów, których zadaniem było po prostu oznaczenie części treści w określonym celu. Operatorzy ci nazywani są „oznaczonymi operatorami treści”, a jest ich pięciu, podzielonych na dwie kategorie. Operatory MP i DP wyznaczają pojedynczy punkt w strumieniu treści, który jest zaznaczony, podczas gdy operatorzy BMC, BDC i EMC przechwytują sekwencję elementów treści w strumieniu treści. Jak już wspomniano, oznakowanie musi znajdować się wokół kompletnych elementów treści, a nie tylko ciągu dowolnych bajtów w pliku PDF. Ponadto zaznaczona sekcja musi być zawarta w pojedynczym strumieniu treści. Aby zaznaczyć pojedynczy punkt w treści, być może w celu umożliwienia łatwego zlokalizowania go przez niestandardowy procesor PDF, operator MP lub DP jest używany w połączeniu z operandem typu Name, czasami nazywanym „tagiem”. Różnica między MP a DP oznacza, że operator DP przyjmuje także drugi argument, który jest listą właściwości. Przykład 2-12 pokazuje kilka przykładów.

Przykład 2-12. Przykład zaznaczonych punktów

```
% strumienia treści gdzieś w pliku PDF

% używamy ABCD_ jako arbitralnej nazwy drugiej klas

/ABCD_MyLine MP

q

10 10 m

20 20 l

S

Q

/ABCD_MyLineWithProps << /ABCD_Prop (Red Line) >> DP

q

1 0 0 RG

10 200 m

100 200 l

S

Q
```

Jak widać w przykładzie 2-13, operatory te są bardziej użyteczne niż wersje prostych punktów, ponieważ faktycznie wyznaczają operatorów, którzy są częścią grupy.

Przykład 2-13. Przykład zaznaczonej treści

```
%  strumienia treści gdzieś w pliku PDF

% używamy ABCD_ jako arbitralnej nazwy drugiej klasy

/ABCD_MyLine BMC

q

10 10 m
```

20 20 l

S

Q

EMC

/ABCD_MyLineWithProps << /ABCD_Prop (Red Line) >> BDC

q

1 0 0 RG

10 200 m

100 200 l

S

Q

EMC

Listy właściwości

Podczas korzystania z oznaczonych operatorów treści DP i BDC słownik jest powiązany z treścią. Słownik ten jest określany jako lista właściwości i zawiera informacje specyficzne dla wykorzystania treści (takich jak zawartość opcjonalna) lub informacje prywatne istotne dla autora tworzącego zaznaczoną treść (lub niestandardowy procesor pliku PDF). Proste listy właściwości, gdzie wszystkie wartości wszystkich kluczy są obiektami bezpośrednimi, mogą być zapisywane w strumieniu treści jako obiekty bezpośrednie (jak widać w poprzednim przykładzie). Jeśli jednak którakolwiek z wartości któregokolwiek z kluczy wymaga pośredniego odwoływania się do obiektów poza strumieniem treści, słownik listy właściwości musi zostać zdefiniowany jako nazwany zasób w słowniku podrzędnym Właściwości bieżącego słownika zasobów, a następnie przywoływany przez nazwę.

Zasoby

W przypadku treści składających się tylko ze ścieżek w prostych przestrzeniach kolorów strumień treści jest całkowicie samowystarczalny i nie wymaga żadnych zewnętrznych odniesień do innych rzeczy. Jednak w przypadku większości rzeczywistych stron PDF potrzebne będą inne rodzaje treści, takie jak obrazy bitmapowe / rastrowe i tekst. Te odnośniki zewnętrzne są zarządzane przez słownik zasobów, który jest wartością klucza Zasoby w słowniku strony (patrz Przykład 2-14). Każdy klucz w słowniku zasobów ma predefiniowaną nazwę, w zależności od typu zasobu, który jest na liście. A wartość każdego klucza sama w sobie jest słownikiem z unikalną (i dowolną) nazwą dla każdego zasobu i pośrednim odniesieniem do zasobu. Powszechną praktyką jest używanie (krótkich) identyfikatorów nazw / prefiksów (takich jak GS dla stanu graficznego, IM dla obrazu itp.), A następnie przyrostowe numerowanie w miarę postępów. Jeśli jednak chcesz nazwać je Manny, Moe i Jack - to też dobrze!

Przykład 2-14. Proste zasoby

% w słowniku strony

/Resources <<

/Font <<

/F1 10 0 obj

/F2 11 0 obj

>>

/XObject <<

/Im1 12 0 obj

>>

>>

Nie martw się zbytnio o szczegóły zasobów; będziesz je przeglądać w konkretnych przykładach, gdy będziesz kontynuować.

Zewnętrzny stan graficzny

We wszystkich dotychczasowych przykładach wszystkie atrybuty stanu graficznego zostały zastosowane bezpośrednio w strumieniu treści. Dzieje się tak głównie dlatego, że używamy ich tylko raz, a ponadto są to proste atrybuty. Jednak będą chwile, kiedy będziesz chciał zachować coś w rodzaju „predefiniowanego stylu" i po prostu odwołać się do niego po imieniu. Podobnie jak arkusze stylów w innych formatach (takich jak HTML lub DOCX), pozwala to na łatwą aktualizację stylu bez wpływu na zawartość (strumień), a jednocześnie zmniejsza rozmiar pliku i zwiększa wydajność. W pliku PDF te style są wywoływane za pomocą słownika parametrów stanu graficznego lub ExtGState (skrót od External Graphic State, zwany tak, ponieważ są one zewnętrzne względem strumienia treści). Aby go użyć, dodaj słownik do słownika zasobów strony dicitonary (oczywiście jako wpis w słowniku ExtGState), a następnie użyj operatora gs w strumieniu treści, aby go wywołać. Na przykład, aby zdefiniować stan graficzny, który używa niestandardowego wzoru kreski na dużej szerokości, można wykonać następujące czynności :

Przykład 2-15. Używanie ExtGState

% w słowniku strony

/Resources <<

/ExtGState <<

/GS0 <<

/Type /ExtGState

/LW 10 % 10 wide

/LC 1 % rounded caps

/D [[2 4 6 4 2] 2] % dash pattern

>>

>>

>>

% in the content stream

/GS0 gs

0 1 0 0 K

100 100 m

100 400 l

S

Teraz, gdy zobaczyłeś, jak używać ExtGStates jako alternatywy dla wstawiania różnych atrybutów, spójrzmy na coś, czego nie można zrobić w trybie inline, ale tylko za pomocą ExtGState.

Podstawowa przezroczystość

Model grafiki PDF obsługuje niezwykle bogaty zestaw funkcji w obszarze przezroczystości, które można szczegółowo zbadać w ISO 32000-1: 2008, klauzula 11. Na razie jednak przyjrzymy się pewnej podstawowej przejrzystości, która jest podobna do czego możesz być przyzwyczajony w innych modelach obrazowania (takich jak GDI +). Najprostszym rodzajem przezroczystości jest zastosowanie poziomu lub procentu (jako liczby od 0 do 1) do tego, jak przejrzysty (lub nieprzezroczysty) jest dany obiekt. Obiekt o wartości przezroczystości 0 jest całkowicie niewidoczny, a wartość 1 (domyślna) jest całkowicie nieprzezroczysta. Każda wartość pomiędzy oznacza, że wszystko, co znajduje się pod spodem, przejdzie przez (lub, aby użyć terminu technicznego, „zleje się" z obiektem przezroczystym) w większym lub mniejszym stopniu. Podobnie jak w przypadku innych atrybutów stanu graficznego, można ustawić wartości przezroczystości obrysu i wypełnienia oddzielnie za pomocą odpowiednio klawiszy CA i ca w słowniku ExtGState. Powodem, dla którego przezroczystość jest obsługiwana jako część słownika ExtGState, a nie bezpośrednio w strumieniu treści, było zapewnienie kompatybilności ze starszymi czytnikami w momencie wprowadzenia do PDF (wersja 1.4). Przykład 2-16 daje kilka przykładów przezroczystości.

Przykład 2-16. Trzy przezroczyste prostokąty

% w słowniku strony

/Resources <<

/ExtGState <<

/GS0 <<

/CA 1

/ca 1

>>

/GS1 <<

/CA .5

/ca .5

>>

/GS2 <<

```
/CA .75
/ca .75
>>
>>
>>
% w strumieniu treści
q
/GS0 gs          % brak przezroczystości
1 0 0 rg
209 426 -114 124 re
f
Q
q
/GS1 gs          % .5 przezroczystość
0 1 0 rg
237 401 -114 124 re
f
Q
q
/GS2 gs          % .75 przeźroczystość
0 0 1 rg
272 373 -114 124 re
f
Q
```

Co dalej

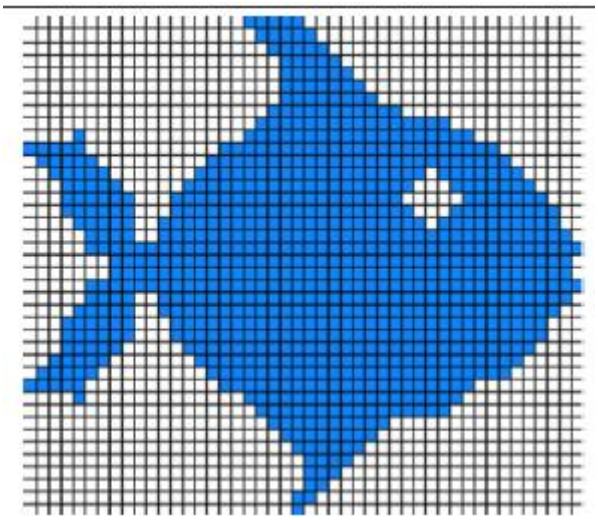
Tu dowiedziałeś się o strumieniach treści i wielu rzeczach, które można w nich znaleźć. Nauczyłeś się również, jak odwoływać się do zewnętrznych rzeczy za pośrednictwem nazwanych zasobów. Jak dotąd jednak omówiliśmy jedynie sposób rysowania grafiki wektorowej (ścieżek), a nie jakiegokolwiek innego rodzaju treści obsługiwanych przez PDF - w szczególności tekstu i obrazów. Następnie przyjrzymy się obrazom, koncentrując się na rastrze, ale także kilka dodatkowych sposobów pracy z grafiką wektorową. Następnie zajmiemy się tekstem.

Obrazy

W poprzedniej części nauczyłeś się tworzyć grafikę wektorową, szereg linii i ścieżek (a czasem tekstu), które nie mają określonej rozdzielczości i mogą składać się z wielu przestrzeni kolorów i atrybutów. Jednak w wielu przypadkach może być konieczne użycie obrazu rastrowego (czasami nazywanego obrazem bitmapowym) na stronie. Ten rozdział przedstawia je Tobie.

Obrazy rastrowe

Kiedy większość ludzi myśli o obrazach rastrowych, myśli o standardowych formatach obrazów rastrowych, takich jak JPEG, PNG, GIF lub TIFF. I chociaż formaty te zawierają dane obrazu rastrowego, zawierają również różnego rodzaju inne rzeczy w postaci „pakietu obrazu". W przypadku plików PDF nie można jednak użyć pełnego pakietu i należy go „rozpakować", aby uzyskać surową formę, której oczekuje PDF. Ta „surowa forma" jest po prostu serią pikseli, lub, mówiąc bardziej technicznie, dwuwymiarową matrycą tych pikseli (te dwa wymiary to wysokość i szerokość obrazu). Na przykład na rysunku 3-1 wysokość wynosi 40 pikseli, a szerokość 46 pikseli.



Każdy z pikseli tego obrazu, jak wspomniano wcześniej, naprawdę jest tablicą wartości - jeden na liczbę kolorów (zwanych również składnikami kolorów) w przestrzeni kolorów. Gdyby to był obraz DeviceRGB, każdy piksel miałby trzy elementy. Jednak gdyby tak było , obraz DeviceCMYK, byłyby cztery, a gdyby był to obraz DeviceGray, byłby tylko jeden składnik. To nie powinno być zaskoczeniem; dokładnie odpowiada liczbie operandów, które biorą operatorzy przestrzeni kolorów! Wreszcie, aby zrozumieć, w jaki sposób ułożone zostaną dane dla obrazu, musisz wiedzieć, ile bitów danych potrzeba dla każdego składnika każdego piksela. Większość programistów myśl tylko w kategoriach 8 bitów na komponent, dlatego obrazy RGB są czasami określane jako 24-bitowe obrazy kolorowe (8 * 3 = 24). Ale PDF obsługuje tutaj znacznie bogatszy zestaw opcji, pozwalając na 1, 2, 4, 8 lub 16 bitów na komponent (chociaż w praktyce używa się tylko 1, 8 i 16).

<u>Dodawanie obrazu</u>

Aby dołączyć obraz rastrowy do pliku PDF i wyświetlić go, musisz wykonać trzy czynności:

1. Utwórz słownik obrazów i dodaj dane obrazu.

2. Dodaj odwołanie do słownika obrazów w słowniku zasobów.

3. Odwołaj się do zasobu w strumieniu treści.

Słowniki obrazkowe

Podobnie jak w przypadku każdej innej wspólnej struktury danych PDF, potrzebny jest słownik reprezentujący wszystkie istotne informacje na temat danych obrazu. Same dane obrazu znajdują się w strumieniu, dlatego słownik, z którym będziesz pracować, będzie znajdował się w słowniku powiązanych atrybutów strumienia. Słownik obrazów jest w rzeczywistości jednym typem słownika XObject (XObject jest skrótem od eXternal Object, obiektu graficznego, który żyje poza lub na zewnątrz strumienia treści). Innym typem będzie ten, którego używasz do obrazów wektorowych. Ponieważ obiekty są zewnętrzne, można do nich odwoływać się z wielu strumieni treści bez powielania. Jest to bardzo przydatne w przypadku grafiki pojawiającej się na wielu stronach dokumentu, takich jak logo firmy / papier firmowy lub tła slajdów. Biorąc to za podstawę słownika, w połączeniu ze wszystkimi omówionymi wcześniej atrybutami, dochodzimy do słownika, który wygląda jak ten z Przykładu 3-1.

Przykład 3-1. Prosty słownik obrazków

```
1 0 obj
<<
/Type /XObject
/Subtype /Image
/Height 40
/Width 46
/ColorSpace /DeviceRGB
/BitsPerComponent 8
/Length 5520 % 40*46*3
>>
stream
% pixel data goes here
endstream
endobj
```

Dodanie go do słownika zasobów nie różni się od przykładów z ostatniego rozdziału z zasobami ExtGState. Tak więc słownik zasobów wyglądałby podobnie do przykładu 3-2.

Przykład 3-2. Słownik zasobów obrazu

```
% in the page dictionary
/Resources <<
/XObject <<
/Im1 1 0 R % reference to our 1 0 obj in the previous example
>>
>>
```

Obrazy w strumieniach treści

Prostą częścią pracy z obrazami w strumieniu treści jest znajomość (i używanie) operatora Do, który przyjmuje jeden operand: nazwę zasobu. Jeśli jednak tylko o tym wiedziałeś i próbowałeś użyć tego samego w strumieniu, tak:

/ Im1 Do

nic nie zobaczysz na stronie i zastanawiasz się, co poszło nie tak. Nieprawidłowe jest to, że rysowanie obiektów XObjects wymaga specjalnej obsługi CTM. Jeśli chcesz zrozumieć to tło, zapoznaj się z ISO 32000-1: 2008, 8.9.4 - ale na razie ważne jest, aby zrozumieć, że zamiast zwykłej matrycy tożsamości 1 0 0 1 0 0, obraz XObject ma wartość domyślną w 0 0 h 0 0 (gdzie w to szerokość obrazu, a h to wysokość w pikselach, zgodnie z definicją w słowniku obrazków). Tak więc, aby nasz obraz pojawił się w lewym dolnym rogu strony, bez żadnego skalowania ani innych przekształceń, nasz strumień treści musi wyglądać jak w przykładzie 3-3.

Przykład 3-3. Ryba FatBits

q % you don't need the q/Q, but it's a good habit!

46 0 0 40 0 0 cm

/Im1 Do

Q

Podobnie jak w przypadku ścieżek, z którymi pracowaliśmy w poprzednim rozdziale, możesz zastosować dowolną kombinację transformacji do skalowania obrazu, obrotu itp. Pamiętaj tylko, aby zawsze zaczynać od rozmiaru obrazu.

Obrazy JPEG

Jednym z różnych filtrów, które można zastosować do obiektu strumienia, jest filtr DCTDecode. Strumień DCTDecode jest równoważny plikowi JFIF, zwanemu również potocznie plikiem JPEG (lub .jpg / .jpeg). Pliki JPEG to jedyny standardowy format obrazu, który można umieścić w pliku PDF bez żadnych modyfikacji - wystarczy odczytać strumień danych z pliku, a następnie zapisz go w wartości obiektu strumienia w pliku PDF, jak pokazano w przykładzie 3-4. Oczywiście musisz albo znać z góry rozmiar, przestrzeń kolorów itp. Obrazu, albo mieć jakiś sposób na przeanalizowanie JPEG, aby uzyskać te wartości.

Przykład 3-4. Obraz PDF na podstawie danych JPEG

1 0 obj

<<

/Type /XObject

/Subtype /Image

/Height 246

/Width 242

/ColorSpace /DeviceRGB

/BitsPerComponent 8

/Length 16423

/Filter /DCTDecode

>>

stream

% image data right from the JPEG goes here

endstream

endobj

Przejrzystość i obrazy

Podczas gdy normalne zachowanie obrazu w modelu obrazowania PDF polega na tym, że wszystkie piksele są narysowane na czymkolwiek pod nimi, istnieją sposoby, dzięki którym obraz może wyrazić, że niektóre jego części są całkowicie lub częściowo przezroczyste. Oryginalne metody obsługiwane przez PDF nazywane są maskowaniem, ponieważ całkowicie „maskują" (wyłączają kontra włączają) zestaw pikseli na podstawie podanych kryteriów. Nowsze metody używają tego samego modelu przezroczystości co ścieżki, w których można uzyskać poziomy przezroczystości. Ponieważ już obejrzeliśmy ten model, omówimy go w pierwszej kolejności.

Miękkie maski

Jak się dowiedziałeś wcześniej, obrazy w formacie PDF mają zdefiniowaną przestrzeń kolorów i jako takie mają określoną liczbę składników. W przypadku RGB istnieją trzy elementy: czerwony, zielony i niebieski. Oznacza to, że nie ma miejsca na informacje dotyczące przejrzystości. Normalnym sposobem rozwiązania tego - podobnie jak w przypadku większości popularnych formatów obrazów, takich jak PNG i TIFF - jest po prostu zdefiniowanie nowego koloru (przestrzeni) z czterema składnikami (ARGB lub RGBA), gdzie czwartym składnikiem każdego piksela jest jego wartość przezroczystości . Jednak, jak się dowiedziałeś, jednym z kluczowych celów, kiedy wprowadzono przezroczystość w PDF, było utrzymanie 100% kompatybilności wstecznej z implementacjami nieprzejrzystymi. To uniemożliwiło użycie nowej przestrzeni kolorów. Dlatego zamiast nowej przestrzeni kolorów wartości przezroczystości dla każdego piksela są przechowywane w osobnym obrazie XObject. Ten obraz „miękkiej maski" jest przywoływany z oryginalnego obrazu XObject za pomocą klucza SMask w jego słowniku. Słownik obrazka z maską miękką jest standardowym słownikiem obrazów, z zastrzeżeniem kilku ograniczeń (patrz ISO 32000-1: 2008, tabela 145) - co najważniejsze, przestrzeń kolorów obrazu musi mieć postać DeviceGray. Dopuszczenie tylko szarości urządzenia ma sens, ponieważ jest to przestrzeń kolorów z jednym komponentem na piksel, która pasuje do naszego brakującego komponentu z RGBA / ARGB. Dlatego też w większości przypadków szerokość i wysokość obrazu rodzica i jego miękkiej maski będą pasować (chociaż nie jest to wymagane). Jeśli weźmiesz wcześniejszy przykład 3-3 i użyjesz miękkiej maski, aby wszystkie białe części były przezroczyste, uzyskasz coś takiego jak rysunek

% this is the soft mask

10 0 obj

<<

/Type /XObject

/Subtype /Image

/BitsPerComponent 8

/ColorSpace /DeviceGray

/Filter /FlateDecode

/Height 40

/Width 46

/Length 166 % smaller for compression

>>

stream

% masking data goes here

endstream

endobj

% this is the parent image

11 0 obj

<<

/Type /XObject

/Subtype /Image

/BitsPerComponent 8

/ColorSpace /DeviceRGB

/Filter /FlateDecode

/Height 40

/Width 46

/SMask 10 0 R

/Length 166

>>

stream

% image data goes here

endstream

endobj

Maski do szablonów

Podczas gdy miękkie maski są najsilniejsze, ponieważ pozwalają na różne poziomy przezroczystości, czasami wszystko, czego potrzebujesz, to móc wyłączyć zestaw pikseli, aby nie rysowały na wierzchu tego, co jest za nimi. Odpowiednik miękkiej maski, ale o prostych właściwościach włączania / wyłączania, nazywany jest maską szablonową. Działa prawie dokładnie tak, jak maska miękka, z tym wyjątkiem, że zamiast zamaskowanego obrazu znajdującego się w DeviceGray przy (zwykle) 8 bitach na komponent, nie ma przestrzeni kolorów i zawsze ma 1 bit na komponent. Każdy bit reprezentuje stan włączenia / wyłączenia piksela, przy czym 0 oznacza włączenie (zaznaczenie) i 1 oznacza wyłączenie (zostawienie w spokoju). W razie potrzeby wartości te można odwrócić poprzez obecność klucza dekodowania w słowniku obrazkowym maski szablonów o wartości [1 0]. Jeśli użyjesz maski szablonowej do maskowania ryby FatBits, może to wyglądać jak w przykładzie 3-5.

Przykład 3-5. Ryba FatBits z maską wzornikową

% this is the stencil mask

10 0 obj

<<

/Type /XObject

/Subtype /Image

/ImageMask true

/BitsPerComponent 1

/Height 40

/Width 46

/Length 230 % (40*46)/8

>>

stream

% masking data goes here

endstream

endobj

% this is the parent image

11 0 obj

<<

/Type /XObject

/Subtype /Image

/BitsPerComponent 8

/ColorSpace /DeviceRGB

/Filter /FlateDecode

/Height 40

/Width 46

/Mask 10 0 R

/Length 166

>>

stream

% image data goes here

endstream

endobj

Maski z kolorowymi kluczami

W niektórych przypadkach, na przykład w przypadku naszych ryb, wszystko, co naprawdę musisz zrobić, to poinformować przeglądarkę plików PDF, że powinna po prostu zignorować (zamaskować) piksele określonego koloru (w naszym przypadku białe). Ta najprostsza forma maskowania nazywa się maskowaniem klawiszem koloru (lub klawiszem chrominancji) i działa jak niebieski ekran w filmach. Wszystko, co ma zdefiniowany kolor, nie jest pociągnięty.

Maski z kolorowymi kluczami

W niektórych przypadkach, na przykład w przypadku naszych ryb, wszystko, co naprawdę musisz zrobić, to poinformować przeglądarkę plików PDF, że powinna po prostu zignorować (zamaskować) piksele określonego koloru (w naszym przypadku białe). Ta najprostsza forma maskowania nazywa się maskowaniem klawiszem koloru (lub klawiszem chrominancji) i działa jak niebieski ekran w filmach. Wszystko, co ma zdefiniowany kolor, nie jest pociągnięty. Aby użyć maski z klawiszem koloru w pliku PDF, nie jest potrzebny obraz dodatkowy; musisz tylko wiedzieć, które kolory chcesz zamaskować. Wpis Mask w słowniku obrazków będzie miał jako swoją wartość tablicę zawierającą 2 * n wpisów (gdzie n jest liczbą elementów w przestrzeni kolorów obrazu): każda para określa minimalne i maksymalne wartości dla tego elementu, który ma być zamaskowany. Dlatego do naszego obrazu RGB potrzebujemy sześciu wartości. Ponieważ chcemy tylko maskować biel, minimum i maksimum są takie same - 255 (wartość bieli). Przy takim podejściu pełny słownik obrazków wygląda jak rysunek

11 0 obj

<<

/Type /XObject

/Subtype /Image

/BitsPerComponent 8

/ColorSpace /DeviceRGB

/Filter /FlateDecode

/Height 40

/Width 46

/Mask [255 255 255 255 255 255]

/Length 166

>>

stream

% image data goes here

endstream

endobj

Obrazy wektorowe

PDF tak naprawdę nie ma pojęcia obrazu wektorowego, takiego jak plik EPS lub EMF. Zamiast tego ma sposób na kapsułkowanie strumienia treści w obiekcie wielokrotnego użytku. Podobnie jak w przypadku obrazów rastrowych, jest to inny typ XObject zwany formą XObject.

Dodanie formularza XObject

Podobnie jak w przypadku obiektów XObjects, musisz zrobić trzy rzeczy:

1. Utwórz słownik formularzy i dodaj dane.

2. Dodaj odwołanie do słownika formularzy w słowniku zasobów.

3. Odwołaj się do zasobu w strumieniu treści.

Słownik formularzy

Formularz XObject to strumień treści (omówiony w „Strumieniach treści” na stronie 35) z powiązanym słownikiem formularzy, który zapewnia dodatkową inteligencję i wsparcie dla tej zawartości. Wszystko, co możesz umieścić w strumieniu treści strony, możesz umieścić w formularzu XObject. Jest to bardzo potężne, ponieważ umożliwia ponowne wykorzystanie całej strony PDF w innym kontekście (na przykład nałożenie lub stemplowanie). W dalszej części tej sekcji dowiesz się, jak to zrobić, ale na razie spójrzmy na specjalne pola słownika formularzy. W słowniku formularzy znajdują się cztery specjalne klucze, które są ważne przy ich tworzeniu:

BBox

Najważniejszym i jedynym wymaganym jest klucz BBox. BBox to tablica reprezentująca obwiednię dla zawartości. Zawsze możesz użyć prostokąta większego niż rzeczywisty rozmiar zawartości, ale mniejszy spowoduje przycięcie zawartości do tego rozmiaru.

Matryca

Matryca jest standardową macierzą transformacji, która będzie stosowana do wszystkich instancji XObject za każdym razem, gdy zostanie narysowana. Prawie zawsze jest to matryca tożsamości, ponieważ w strumieniu treści wywołujących zostanie zastosowana jakakolwiek konkretna transformacja. Ponieważ domyślną wartością klucza jest tożsamość, nie ma powodu, aby umieszczać to w słowniku, chyba że transformacja jest czymś innym.

Zasoby

Strumień treści, podobnie jak strona, może wymagać dodatkowych zasobów (ExtGStates, czcionki itp.) I tutaj możesz się do nich odwoływać.

FormType

Ten jest nie tylko opcjonalny, ale nie służy żadnemu praktycznemu celowi, ponieważ istniała dla niego tylko jedna wartość (1). Nie zawracaj sobie głowy zapisywaniem go w plikach PDF; marnuje tylko miejsce. Przykład 3-6 pokazuje prostą formę XObject.

Przykład 3-6. Prosta forma XObject

```
1 0 obj
<<
/Type /XObject
/Subtype /Form
/BBox [0 0 100 100]
/FormType 1 % optional, only here for example
/Matrix [1 0 0 1 0 0] % optional, only here for example
/Length 180
>>
stream
0 0 1 rg % set the color to blue in RGB
0 0 100 100 re % draw a rectangle 100x100 with the bottom left at 0,0
F % fill it
endstream
endobj
```

Następnie dodajesz to do słownika zasobów strony i odwołujesz się do niego w strumieniu zawartości dokładnie tak, jak w przypadku obrazu. Jednak w przeciwieństwie do obrazów, w przypadku formularzy

XObject używasz standardowego CTM i nie musisz martwić się o rozmiar (ponieważ nie jest opisany w pikselach).

Przykład 3-7. Odwoływanie się do XObject w słowniku stron

% in the page dictionary

/Resources <<

/XObject <<

/Im1 1 0 R % reference to our 1 0 obj in the previous example

>>

>>

% in the page's content stream

q

1 0 0 1 0 0 cm % as with normal content, this means 100% at 0,0

/Im1 Do

Q

Operator Do, w połączeniu z formularzem XObject, korzysta z kluczy omówionych wcześniej w ramach renderowania lub malowania. Operacje są następujące:

• Zapisuje bieżący stan graficzny, tak jakby wywoływał operator q

• Łączy macierz z pozycji Matrix słownika formularzy z bieżącą macierzą transformacji (CTM)

• Klipy zgodnie z wpisem BBox w słowniku formularzy

• Maluje obiekty graficzne określone w strumieniu zawartości formularza

• Przywraca zapisany stan graficzny, tak jakby wywoływał operator Q.

Kopiowanie strony do formularza XObject

Jeśli budujesz narzędzie do narzucania lub stemplowania dokumentów PDF, jedną z typowych operacji jest konwersja strony PDF na formularz XObject, dzięki czemu można ją łatwo włączyć na dowolnej innej stronie. Zakładając, że masz bibliotekę PDF, która jest w stanie pracować z modelem obiektowym pliku PDF, poniższe instrukcje powinny pomóc w konwersji:

1. Jeśli strona ma tablicę strumieni treści, połącz je w jeden.

2. Oblicz BBox na podstawie klawiszy MediaBox i Rotate strony.

3. Skopiuj słownik zasobów ze strony do formularza XObject (jeśli robisz to w tym samym pliku PDF, użyj płytkiej kopii zamiast głębokiej).

Tekst

W tej części nauczysz się rysować tekst na stronie. Rysowanie tekstu jest najbardziej złożoną częścią grafiki PDF, ale także pomogło PDF pokonać konkurentów, aby stać się międzynarodowym standardem, jakim jest obecnie. Podczas gdy inni oryginalni gracze przekształcali tekst w obrazy rastrowe lub ścieżki wektorowe (w celu zachowania integralności wizualnej), wynalazca PDF wiedział, że użytkownicy potrzebują tekstu, który można przeszukiwać i kopiować, i nie wygląda po prostu ładnie na ekranie. Dzięki dogłębnemu doświadczeniu i zrozumieniu czcionek, które mieli inżynierowie Adobe, byli w stanie zintegrować rzeczywisty tekst z prezentacją wizualną.

Chociaż obsługa tekstu w formacie PDF umożliwia renderowanie dowolnych glifów z dowolnej czcionki reprezentującej dowolny język, mechanika (jak wkrótce zobaczycie) została stworzona przed Unicode. Oznacza to, że wiele rzeczy, które programiści pracujący w innych formatach plików traktują jako coś oczywistego, takie jak po prostu umieszczenie punktów kodowych Unicode i umożliwienie renderera wykonania całej ciężkiej pracy, należy wykonać ręcznie za pomocą PDF.

Teraz, gdy otrzymałeś sprawiedliwe ostrzeżenie, zacznijmy!

Czcionki

W poprzednich częściach nauczyłeś się rysować na stronie grafikę wektorową (lub ścieżki) oraz grafikę rastrową (obrazy). Tego rodzaju operacje rysowania są dość proste, ponieważ zwykle nie wymagają dodatkowych informacji - to tylko instrukcje i (w przypadku obrazów rastrowych) dane obrazu. Tekst wymaga jednak więcej elementów. Najważniejszym z tych elementów jest czcionka.

Glify

Czcionka, czasami nazywana programem czcionkowym, to zbiór unikalnych instrukcji rysunkowych zwanych glifami. Zasadniczo każdy glif nie różni się niczym od ścieżek lub rastrów, które narysowałeś w poprzednich rozdziałach. Jednak czcionka zawiera także kilka metadanych dotyczących różnych glifów, w tym coś, co nazywa się kodowaniem zapewniającym odwzorowanie znanego zestawu znaków (takiego jak ASCII, Unicode lub Shift-JIS) na glify. Nie każda czcionka ma glify dla każdej wartości w każdym kodowaniu. Podczas gdy glif dotyczy przede wszystkim jego kształtu i wyglądu, ma również zestaw powiązanych z nim wartości zwanych miernikami glifów. Te wskaźniki, z których niektóre są wyraźnie zdefiniowane w samej czcionce, podczas gdy inne są obliczane z szeregu wartości, umożliwiają oprogramowaniu wykonywanie takich czynności, jak układanie tekstu. Na przykład, aby narysować „Cześć", musisz wiedzieć, gdzie umieścić „e" po „H."

Rodzaje czcionek

Kiedy po raz pierwszy powstały interfejsy graficzne, takie jak oryginalna Lisa i Macintosh, czcionki, które były z nimi związane, były znane jako „czcionki bitmapowe", ponieważ każdy glif został opisany jako obraz bitmapowy lub rastrowy. Mimo że był dość przydatny na ekranie, nie był wystarczająco elastyczny do drukowania, więc narodziły się formaty czcionek konturowych (ścieżkowych). PDF obsługuje trzy najpopularniejsze formaty konturów:

Type 1

Był to oryginalny format konspektu stworzony przez Adobe wraz z językiem Postscript dla drukarek. Kontury glifów opisano przy użyciu uproszczonej wersji Postscript.

True Type

Jest to najbardziej znany format czcionek stworzony przez Apple i Microsoft dla ich systemów operacyjnych. Kontury glifów są opisane przy użyciu specjalnego języka, który jest unikalny dla tego formatu czcionki.

Open Type

Chociaż czcionki Type 1 i TrueType mają swoje zalety, przemysł zmęczył się „wojnami z czcionkami", dlatego narodził się OpenType. OpenType łączy najlepsze z pozostałych formatów, z opcją konturów glifów opisywanych jako Type 1 lub TrueType.

W przypadku tych typów czcionek rzeczywisty program czcionek jest zdefiniowany w osobnym pliku czcionek, który może być osadzony w obiekcie strumienia PDF lub uzyskany z zewnętrznego źródła. PDF ma także dwa typy czcionek specyficzne dla PDF, w których dane glifów muszą być zdefiniowane w pliku PDF:

Type 3

Pierwotnie udostępniony jako sposób osadzania czcionek bitmapowych, czcionka Type 3 jest tak naprawdę słownikiem PDF, w którym każdy glif jest zdefiniowany przez standardowy strumień treści. Umożliwia to nie tylko glify oparte na rastrze, ale także użycie dowolnego / wszystkich operatorów graficznych PDF do zdefiniowania glifu. Chociaż mogą być bardzo wydajne, nie są obecnie używane w większości systemów do tworzenia plików PDF.

Type 0

Znana również jako czcionka kompozytowa lub CIDFont, czcionka typu 0 jest tworzona przez pobieranie opisów glifów z jednej lub więcej czcionek i tworzenie amalgamatu lub kompozytu. Było to pierwotnie konieczne podczas pracy z czcionkami w języku chińskim, japońskim lub koreańskim (CJK), które nie miały również znaków angielskich / łacińskich i pożądana była jedna czcionka z obydwoma. Chociaż nie jest używana do łączenia glifów z wielu czcionek, jest to metoda stosowana w przypadku czcionek Unicode, szczególnie w przypadku danych dwubajtowych.

Słownik czcionek

W części 1 zobaczyłeś nasz pierwszy plik PDF, który narysował „Hello World" na stronie. Przykład 4-1 pokazuje kilka istotnych elementów, które będziemy teraz oglądać bardziej szczegółowo.

Przykład 4-1. Części „Hello World.pdf"

```
1 0 obj
<<
/Type /Page
/Parent 5 0 R
/MediaBox [ 0 0 612 792 ]
/Resources 3 0 R
/Contents 2 0 R
>>
endobj
```

3 0 obj

<<

/Font <</F1 4 0 R >>

>>

endobj

4 0 obj

<<

/Type /Font

/Subtype /Type1

/Name /F1

/BaseFont/Helvetica

>>

endobj

W tym przykładzie wiesz, że obiekt nr 1 jest stroną i ma tylko niezbędne minimum: MediaBox (dla rozmiaru strony), Treść (dla instrukcji rysowania) i Zasoby (potrzebne dla Zawartość). Istnieje jeden typ zasobu, Czcionka, z pojedynczym wpisem F1, który jest słownikiem czcionek w obiekcie nr 4.

Słownik czcionek zawiera informacje określające typ czcionki (tj. Typ 1 lub TrueType), jej nazwę PostScript, jej kodowanie oraz informacje, których można użyć jako zastępstwa, gdy program czcionek nie jest osadzony w pliku PDF. ISO 32000-1: 2008, 9.6.2.2 opisuje czcionki Standard 14 (znane również jako Base 14), o których znajomość z każdym mechanizmem renderowania plików PDF musi wiedzieć i udostępnianie ich bezpośrednio lub za pośrednictwem odpowiednich zamienników. Ponadto renderer powinien wiedzieć o swoich metrykach, co pozwoli nam (na razie) nie musieć określać tych wartości z programu czcionek i włączać ich do słownika czcionek. Obiekt nr 4 w naszym przykładzie jest dość mały, ponieważ używa jednej z tych czcionek (Helvetica). Pełna lista standardowych czcionek to:

• Times-Roman

• Times-Bold

• Times-kursywa

• Times-BoldItalic

• Helvetica

• Helvetica-Bold

• Helvetica-Oblique

• Helvetica-BoldOblique

• Kurier

• Courier-Bold

• Kurier-ukośny

• Courier-BoldOblique

• Symbol

• ZapfDingbats

Podczas tworzenia słownika czcionek dla jednej z tych czcionek wymagane są tylko trzy wymagane klucze: Typ (z wartością Font), Podtyp (z wartością Type1) i BaseName (z jedną z wartości z poprzedniej listy). Być może zauważyłeś, że niektóre z tych czcionek zawierają styl, który normalnie uważasz za styl (np. Pogrubienie lub kursywa). Jednak podczas pracy z PDF nie ma stylów, tylko czcionki. Jeśli więc chcesz, aby czcionka Times była pokazana kursywą, musisz użyć czcionki Times-Italic. Przykład 4-2 to odmiana „Hello World” przy użyciu kilku różnych czcionek.

9 0 obj

<<

/Type/Page

/Contents 24 0 R

/MediaBox[0 0 612 792]

/Parent 5 0 R

/Resources 13 0 R

>>

endobj

13 0 obj

<<

/Font <<

/F1 14 0 R

/F2 13 0 R

/F3 12 0 R

/F4 11 0 R

>>

>>

endobj

14 0 obj

<<

/Type /Font

/Subtype /Type1

/BaseFont/Helvetica

>>

endobj

13 0 obj

<<

/Type /Font

/Subtype /Type1

/BaseFont/Symbol

>>

endobj

12 0 obj

<<

/Type /Font

/Subtype /Type1

/BaseFont/Courier-Bold

>>

endobj

11 0 obj

<<

/Type /Font

/Subtype /Type1

/BaseFont/Times-Italic

>>

endobj

24 0 obj

<</Length 182>>

stream

BT

/F1 48 Tf

1 0 0 1 10 100 Tm

(Hello World)Tj

0 50 Td

/F2 48 Tf

(Hello World)Tj

0 50 Td

/F3 48 Tf

(Hello World)Tj

0 50 Td

/F4 48 Tf

(Hello World)Tj

ET

endstream

endobj

14 0 obj

<<

/Type /Font

/Subtype /Type1

/BaseFont/Helvetica

/Encoding/WinAnsiEncoding

>>

endobj

```
24 0 obj
<</Length 182>>
stream
BT
/F1 48 Tf
1 0 0 1 10 100 Tm
(English)Tj
0 50 Td
/F2 48 Tf
(Français)Tj
0 50 Td
/F3 48 Tf
(Español)Tj
ET
endstream
endobj
```

Aby obsługiwać dowolne języki inne niż łacińskie, konieczne jest stosowanie czcionek osadzonych

Stan tekstu

Teraz, gdy masz już dobrą obsługę czcionek i glifów, zobaczmy, jak możesz ich używać do rysowania tekstu na stronie

Przykład 4-4. Części Hello World.pdf

```
1 0 obj
<<
/Type /Page
/Parent 5 0 R
/MediaBox [ 0 0 612 792 ]
/Resources 3 0 R
/Contents 2 0 R
>>
endobj
2 0 obj
```

<< /Length 53 >>

stream

BT

/F1 24 Tf

1 0 0 1 260 600 Tm

(Hello World)Tj

ET

endstream

endobj

Jeśli przyjrzysz się strumieniowi treści w obiekcie nr 2, zobaczysz pięciu nowych operatorów, których wcześniej nie widziałeś. Pierwszy, BT, pojawia się sam (tj. Bez operandów) w pierwszym wierszu. Jak zapewne można się domyślić, oznacza to „Rozpocznij tekst" i jest wymagane do nakreślenia szeregu instrukcji związanych z tekstem. Jest on sparowany z operatorem ET (End Text), który można zobaczyć w ostatnim wierszu strumienia treści. Podobnie jak PDF ma stan graficzny, ma również stan tekstowy, który zawiera wszystkie atrybuty związane z rysowaniem tekstu. BT / ET parowanie deklaruje nowy stan tekstu, a następnie usuwa go, podobnie jak q i Q ze stanem graficznym. Jednak nie ma push / pop. W rzeczywistości niedozwolone jest zagnieżdżanie par BT / ET.

Czcionka i rozmiar

Chociaż w stanie tekstowym znajduje się wiele atrybutów, a poniżej kilka, trzy najważniejsze to czcionka, która ma zostać użyta, rozmiar tekstu i miejsce, w którym należy umieścić tekst. Operator Tf określa nazwę zasobu czcionek, czyli wpis w słowniku czcionek bieżącego słownika zasobów. Wartością tego wpisu jest słownik czcionek. W poprzednim przykładzie czcionka nosi nazwę F1, a jeśli wrócisz do rysunku 4-1, zobaczysz, że jest ona obecna w zasobach czcionek w obiekcie nr 3. Jak się wcześniej dowiedziałeś, PDF używa ogólnej koncepcji „jednostki użytkownika" do definiowania wielkości i położenia obiektów. W formacie PDF standardowy rozmiar glifu wynosi 1 jednostkę w przestrzeni użytkownika, a nominalna wysokość ciasno rozmieszczonych linii tekstu również wynosi 1 jednostkę. Dlatego, aby narysować glif w określonym rozmiarze, musisz go skalować. Współczynnik skalowania jest określony jako drugi argument operatora Tf, tym samym ustawiając parametr rozmiaru czcionki tekstu w stanie graficznym. W naszym przykładzie rozmiar czcionki wynosi 24. Istnieje drugi sposób ustawienia współczynnika skali dla glifów, który jest podobny do skalowania innych obiektów graficznych - przy użyciu macierzy transformacji. Do transformacji specyficznych dla tekstu używasz operatora Tm, który przyjmuje takie same parametry jak operator cm omówiliśmy w części 2. W naszym przykładzie nie ma miejsca żadne dodatkowe skalowanie, ale tekst jest ustawiany na (260,600). Jest to normalny sposób ustawiania skalowania i pozycji dla pierwszego narysowanego glifu. Można to jednak zrobić w ten sposób:

/ F1 1 Tf

24 0 0 24 260 600 Tm

(Hello World) Tj

Czy jeden sposób jest lepszy od drugiego? W nowoczesnych systemach i bieżących wdrożeniach różnice powinny być nie do odróżnienia. Jeśli próbujesz uzyskać najlepszą możliwą implementację

treści, będzie to zależeć od sposobu użycia czcionki. Jeśli używasz tekstu tylko w jednym rozmiarze (na danej stronie), najlepiej zastosować współczynnik skali jako część Tf, ponieważ wtedy glify zostaną wstępnie przeskalowane przez moduł ładujący czcionki. Jeśli jednak używasz tej samej czcionki w wielu rozmiarach na stronie, to ładowanie w 1 jednostce i używanie Tm do skalowania za każdym razem jest prawdopodobnie bardziej optymalne - ale tylko wtedy, gdy często się zmieniasz. W przeciwnym razie po prostu utwórz drugie wystąpienie czcionki za pomocą drugiego Tf, jak pokazano w przykładzie 4-5.

Przykład 4-5. Rysowanie zarówno dużego, jak i małego tekstu

```
BT
/F1 24 Tf
1 0 0 1 100 150 Tm
(Big Text)Tj
/F1 12 Tf
1 0 0 1 100 100 Tm
(Small Text)Tj
ET
```

Tryb renderowania

Dowiedziałeś się, że ścieżki narysowane w formacie PDF można wypełnić, obrysować lub oboje w oparciu o operator (np. F vs. S), który kończy opis ścieżki. W przypadku tekstu zamiast różnych operatorów istnieje jeden operator (Tr), który ustawia tryb renderowania tekstu. Rysunek pokazuje możliwe wartości argumentów, które mogą być użyte z Tr oraz ich wpływ na tekst.

	Rendering mode	Description
	0	Fill text
	1	Stroke text
	2	Fill then stroke text
	3	Text with no fill and no stroke (invisible)
	4	Fill text and add it to the clipping path
	5	Stroke text and add it to the clipping path
	6	Fill then stroke text and add it to the clipping path
	7	Add text to the clipping path

Po połączeniu trybów renderowania z atrybutami stanu graficznego, o których już wiesz, możesz utworzyć tekst w przykładzie 4-6.

Przykład 4-6. Obrysowany i wypełniony tekst

BT

/F1 80 Tf

1 0 0 1 100 100 Tm

1 0 0 RG

[2] 0 d

0.75 g

2 Tr

(ABC)Tj

ET

Rysowanie tekstu

W przypadku, gdy jeszcze tego nie rozgryzłeś, operator Tj służy do rysowania tekstu (znanego również jako „pokazywanie” ciągu) na stronie. Jest to dość proste, ponieważ operator powoduje, że moduł renderujący PDF wyrównuje „początek glifu” pierwszego glifu z bieżącą lokalizacją pisaka i rysuje glif. Następnie mechanizm renderujący przesuwa pióro o szerokość glifu do „następnego początku glifu” i rysuje następny glif i tak dalej dla całego łańcucha. W przypadku większości renderowania tekstu jest to całkowicie dopuszczalne i to, co większość użytkowników przyzwyczaja do oglądania na ekranie. Jednak w przypadkach, w których chcesz bardziej precyzyjną kontrolę nad pozycjonowaniem glifów, będziesz musiał użyć operatora TJ. Wiele czcionek zawiera informacje o tym, jak precyzyjniej umieścić niektóre glify w stosunku do siebie, znane jako kerning. Jednak, nie jest to obsługiwane przez operatora Tj podczas rysowania łańcucha. Jeśli chcesz skorzystać z tych informacji, musisz uzyskać je sam z czcionki, a następnie użyć operatora TJ, aby uzyskać bardziej atrakcyjny wizualnie wynik.

Operator TJ, zamiast brać ciąg jako operand, pobiera tablicę. Tablica składa się z jednego lub więcej ciągów znaków przeplatanych liczbami, przy czym liczby służą do regulacji położenia tekstu (Tm). Liczby są wyrażone w tysięcznych jednostkach, a wartość jest odejmowana od bieżącej współrzędnej poziomej.

Oznacza to, że w domyślnym układzie współrzędnych dopasowanie dodatnie powoduje przesunięcie następnego glifu namalowanego w lewo o podaną wartość, a dopasowanie ujemne spowoduje przesunięcie następnego glifu w prawo. Przykład 4-7 pokazuje rysowanie słowa za pomocą prostego operatora Tj i ręczne kernowanie go za pomocą TJ.

Przykład 4-7. Ręcznie kernowany tekst

BT

/F1 48 Tf

1 0 0 1 10 150 Tm

(AWAY)Tj

1 0 0 1 10 100 Tm

[(A) 120 (W) 120 (A) 95 (Y)] TJ

ET

Pozycjonowanie tekstu

We wszystkich poprzednich przykładach tekst został wyraźnie umieszczony za pomocą operatora Tm. Jest to jednak operacja dość ciężka, jeśli wszystko, co chcesz zrobić, to przesunąć pióro w jednym

kierunku (np. W dół do następnej linii lub w prawo). Dla prostszych ruchów należy użyć operatora Td. Pobiera dwa parametry, tx i ty, reprezentujące sposób poruszania piórem odpowiednio w kierunkach X i Y. Jeśli którykolwiek z parametrów wynosi 0, pióro nie przesuwa się w tym kierunku. Przykład 4-8 ilustruje użycie Td do narysowania „4 kwadratu”.

Przykład 4-8. „4 kwadrat” liczb

```
BT
/F1 48 Tf
1 0 0 1 10 700 Tm
(1)Tj
0 -50 Td
(2)Tj
50 50 Td
(3)Tj
0 -50 Td
(4)Tj
ET
```

Pamiętaj, że w PDF współrzędna y ma wartość 0 na dole strony, więc aby narysować tekst na stronie, zaczynasz od góry i odejmujesz

Nawigacja

Chociaż ostatnie trzy rozdziały spędziłeś na rozmowach na temat zawartości statycznej, w PDF jest znacznie więcej. W tym rozdziale przedstawimy różne sposoby, w jakie PDF może uzyskać interaktywność, szczególnie w zakresie umożliwiania nawigacji w obrębie dokumentów i między nimi.

Miejsca docelowe

Miejsce docelowe określa konkretny widok dokumentu. Zawsze będzie odnosić się do konkretnej strony pliku PDF i może opcjonalnie zawierać mniejszą podsekcję strony, a także współczynnik powiększenia (powiększenia). Miejsca docelowe nie są samodzielne; są to wartości kluczy w określonych słownikach związanych z częściami pliku PDF, które mogą spowodować wywołanie powiązanej akcji. Na przykład w katalogu dokumentów miejscem docelowym może być wartość klucza OpenAction. Gdy obecny, który instruuje widza, aby skoczył do tego miejsca docelowego natychmiast po otwarciu dokumentu. Powszechnym zastosowaniem OpenAction jest przeskakiwanie na pierwszą stronę rzeczywistej zawartości dokumentu, która może pojawić się po niektórych materiałach wstępnych większość użytkowników nie będzie zainteresowana czytaniem.

Jawne miejsca docelowe

Ponieważ liczba elementów, które składają się na miejsce docelowe, jest niewielka i dobrze zdefiniowana, miejsce docelowe nie jest oparte na słowniku, ale na tablicy, w przeciwieństwie do innych typów wspólnych obiektów, które do tej pory spotkałeś. Pierwszy element tablicy jest zawsze pośrednim odniesieniem do obiektu strony, do którego się odwołuje, po którym następuje obiekt nazwy opisujący rodzaj powiększenia, a następnie wszelkie dodatkowe opcje potrzebne do tego powiększenia.

Przykład 5-1. Przykłady miejsc docelowych

% Object 1 is assumed to be the page

[1 0 R /Fit] % Display entire page with horizontal & vertical magnified to

fit

[1 0 R /FitH] % Display entire page with only horizontal magnified to fit

[1 0 R /FitH 100] % FitH variant where vertical top is 100

[1 0 R /FitV] % Display entire page with only vertical magnified to fit

[1 0 R /FitV 100] % FitH variant where horizontal left is 100

In addition to Fit, FitH, and FitV, there are other ways to zoom into a specific part of

a page; the most powerful of these is XYZ, which includes the specific left and top coordinates

of the page (for the viewer to align with) along with the zoom factor. If you don’t

know them or want any of the top, left, or zoom factor values to change from what they

are when the destination is invoked, provide a 0 or null value. Example 5-2 illustrates

the use of XYZ.

Example 5-2. Examples of destinations

% Object 1 is assumed to be the page

[1 0 R /XYZ 36 36 50] % Display a portion of the page, 1/2 inch in from the top

and left and zoomed 50%

[1 0 R /XYZ null null 50] % Display the current portion of the page, zoomed 50%

Nazwane miejsca docelowe

Zamiast być definiowanym bezpośrednio przy użyciu poprzedniej składni, do miejsca docelowego można odwoływać się pośrednio za pomocą ciągu znaków. Ta funkcja jest szczególnie przydatna, gdy miejsce docelowe znajduje się w innym dokumencie PDF. Na przykład dokument (DocA), który chciałby połączyć się z początkiem części 2 w innym dokumencie (DocB), może odnosić się do tego miejsca docelowego za pomocą nazwy, a nie wyraźnego numeru strony w innym dokumencie. Umożliwiłoby to zmianę rzeczywistego fizycznego numeru strony tego rozdziału w DocB (z powodu edycji, wstawiania / usuwania stron itp.) Bez unieważniania miejsca docelowego w DocA. Nazwane miejsca docelowe działają poprzez tworzenie zgodności między nazwą (która jest reprezentowana jako obiekt łańcuchowy) a miejscem docelowym. Ta korespondencja lub mapowanie odbywa się poprzez słownik nazw dokumentu, który jest wartością klucza Nazwy w słowniku katalogu dokumentu. W słowniku nazw znajduje się klucz Dests, którego wartością jest drzewo nazw z ciągiem → mapowanie miejsca docelowego.

Drzewo nazw jest podobne do słownika, ponieważ umożliwia powiązanie klucza z wartością; jednak zamiast używać obiektu nazwy jako klucza, używa obiektu łańcuchowego i wymaga sortowania łańcuchów. Wartość docelowa powiązana z kluczem w drzewie nazw może być tablicą lub słownikiem. Gdy wartością tego wpisu jest słownik, każdy klucz jest nazwą celu, a odpowiadająca mu wartość to jawne miejsce docelowe lub słownik z wpisem D, którego wartość jest jawnym miejscem docelowym. działania

Chociaż miejsc docelowych można używać samodzielnie, czasem trzeba je włączyć do akcji, aby były użyteczne. Akcja, jak sama nazwa wskazuje, jest formą polecenia obecnego w dokumencie, która wywołuje określone zachowanie (akcję) w przeglądarce. Istnieją zarówno działania ukryte, jak i wyraźne. Działanie niejawne odbywa się za pomocą normalnej nawigacji dokumentu, takiej jak OpenAction, która jest wywoływana przy pierwszym otwarciu dokumentu, natomiast działanie jawne ma miejsce, gdy użytkownik wchodzi w interakcję z innym obiektem w pliku PDF, takim jak przycisk lub zakładka.

Słownik akcji

Słownik akcji jest wspólnym słownikiem podstawowym dla wszystkich rodzajów akcji. Składa się tylko z jednego wymaganego wpisu: klawisz S, którego wartość określa rodzaj akcji. Wszystkie pozostałe klawisze w tym słowniku będą się różnić w zależności od rodzaju akcji.

Działania GoTo

Najczęstszym działaniem jest GoTo. Działanie GoTo odpowiada celowi, ponieważ jego wywołanie spowoduje, że widz „przejdzie” do określonego celu. Jest to jasne, gdy zobaczysz, że głównym klawiszem w słowniku działań GoTo jest klawisz D, którego wartość jest miejscem docelowym

Przykład 5-3. Prosty link GoTo

% This is the action dictionary for a GoTo action

```
8 0 obj
<<
/Type /Action
/S /GoTo
/D [10 0 R /Fit] % GoTo the page referred by object 10 and fit it
>>
endobj
% This is an object called a link annotation
9 0 obj
<<
/Type /Annot
/Subtype /Link
/Rect [100 100 150 150]
/A 8 0 R
>>
endobj
```

W tym przykładzie obiekt nr 8 reprezentuje akcję GoTo, która przeniesie użytkownika na stronę dokumentu (do której odwołuje się obiekt nr 10) i wyświetli go w powiększeniu w celu dopasowania (zgodnie z jawnym miejscem docelowym, którym jest wartość klawisza D). Ta akcja nie może jednak stać sama z siebie - musi być połączona z innym obiektem, który spowoduje jej wywołanie przez widza niejawnie lub użytkownika jawnie. W tym przykładzie jest on połączony z obiektem (# 9) zwanym adnotacją łącza. Adnotacja linku to obiekt, który zapewnia klikalny obszar na stronie powiązany z akcją

Akcje URI

Drugim najczęstszym działaniem jest URI. Identyfikator URI (jednolity identyfikator zasobu) jest bardziej elastyczną koncepcją niż zwykły adres URL (jednolity lokalizator zasobów), który zna większość użytkowników, choć w zdecydowanej większości przypadków można ich używać zamiennie. Możesz więc pomyśleć o akcji URI jako o łączu internetowym. Podobnie jak akcja GoTo, akcja URI jest bardzo prosta; obejmuje dodanie jednego klucza (URI) do standardowego słownika akcji.

Przykład 5-4. Prosty link URI

```
% This is the action dictionary for a URI action
8 0 obj
<<
/Type /Action
/S /URI
```

/URI (http://www.oreilly.com)

>>

endobj

9 0 obj

<<

/Type /Annot

/Subtype /Link

/Rect [100 100 150 150]

/A 8 0 R\>>

endobj

Wartość klucza URI może być dowolnym poprawnym identyfikatorem URI. Nie ogranicza się to do popularnych schematów, takich jak http lub ftp; może użyć schematu historycznego, takiego jak suseł, lub schematu niestandardowego lub prywatnego, takiego jak książka. Powodem jest to, że czytnik PDF nie jest wymagany do obsługi określonych schematów i zwykle przekazuje identyfikator URI do systemu operacyjnego w celu przetworzenia.

GoToR i uruchom akcje

Widziałeś, w jaki sposób działania GoTo mogą być bardzo przydatne, o ile potrzebujesz tylko poruszać się po tym samym dokumencie PDF. Podobnie akcja URI umożliwia przekierowanie użytkownika do dokumentów i stron internetowych, które są online. Czasami jednak chcesz odwołać się do czegoś w innym pliku PDF lub nawet innego rodzaju dokumencie (np. Dokumencie do edycji tekstu lub arkuszu kalkulacyjnym). W takich sytuacjach konieczne jest użycie akcji GoToR (dla plików PDF) lub Uruchom (dla innych typów plików). Akcja GoToR, znana również jako akcja „zdalnego przejścia", obejmuje wszystkie aspekty akcji GoTo, ale z dodatkiem klawisza F w słowniku akcji, który wskazuje na PDF, w którym miejsce docelowe ma być zdecydowany. Powszechnym zastosowaniem tego byłoby książka podzielona na rozdziały, w których potrzebne są powiązania między rozdziałami. Przykład 5-5 ilustruje, jak używać GoToR, aby otworzyć plik PDF o nazwie „Rozdział2.pdf" (który znajduje się w tym samym katalogu co plik PDF łącza) i przejść do fizycznej strony nr 1.

Przykład 5-5. Przykład akcji GoToR

% To jest słownik akcji dla akcji GoTo

8 0 obj

<<

/ Type / Action

/ S / GoToR

/ D [0 / Fit]% UWAGA: GoToR używa zerowych stron #, a nie referencji pośrednich

/ F <<

/ Type / Filespec

/ F (Rozdział2.pdf)

/ UF (Rozdział2.pdf)

>>

endobj

Możesz się zastanawiać, dlaczego masz taki sam ciąg znaków (Część2.pdf) jak wartości zarówno dla klawiszy F, jak i UF w słowniku działań GoToR. Powodem jest to, że UF jest preferowany we współczesnych czytnikach PDF, ponieważ wartość może być łańcuchem Unicode, podczas gdy F jest tam z powodów historycznych. Tak więc, aby zapewnić najlepsze wyniki, zawsze piszesz oba. W sytuacjach, gdy konieczne jest zewnętrzne połączenie z dokumentem w formacie innym niż PDF, działanie Uruchom stanowi rozwiązanie. Jak widać w przykładzie 5-6, jego składnia jest prawie dokładnie taka sama jak w akcji GoToR.

Przykład 5-6. Przykład akcji uruchamiania

% This is the action dictionary for a GoTo action

8 0 obj

<<

/Type /Action

/S /GoToR

/D [0 /Fit] % NOTE: GoToR uses zero-based page #s, not indirect references

/F <<

/Type /Filespec

/F (Chapter2.pdf)

/UF (Chapter2.pdf)

>>

endobj

Akcje multimedialne

Dotknęliśmy tylko powierzchni działań z czterema rodzajami działań omówionymi do tej pory. Istnieją również działania, które umożliwiają PDF odtwarzanie dźwięków i filmów, pracę z osadzonymi obiektami 3D, a nawet uruchomienie programu JavaScript.

Zagnieżdżone akcje

Jedną z innych zalet używania akcji zamiast prostego miejsca docelowego jest to, że można łączyć wiele akcji, używając klawisza Dalej w słowniku akcji, aby połączyć je razem. Na przykład efekt kliknięcia adnotacji łącza myszą może polegać na odtworzyć dźwięk, przejść do nowej strony i rozpocząć film. Następny wpis nie jest ograniczony do pojedynczej akcji, ale może zawierać szereg akcji, których każda z kolei może mieć własną pozycję Next. Akcje tworzą zatem drzewo zamiast prostej listy połączonej. Działania w obrębie każdej kolejnej tablicy są wykonywane w kolejności, a po każdej następnej są akcje określone w jej następnej pozycji, i tak dalej rekurencyjnie.

Działania, w tym ich wartości Next, są zawsze przetwarzane w pierwszej kolejności, aby zapewnić spójność ich kolejności przetwarzania. Zapewnia to również ubezpieczenie, że stan dokumentu będzie zgodny z oczekiwaniami podczas przetwarzania każdego działania.

Zakładki lub kontury

Dokumenty strukturalne bardzo często zawierają drzewiastą hierarchię elementów konspektu, które czytelnik przedstawia użytkownikowi. Te elementy konturu pozwalają użytkownikowi interaktywnie nawigować z jednej części dokumentu do drugiej poprzez ich prezentację jako wizualny spis treści oddzielony od rzeczywistej zawartości strony. Gdy taki element jest aktywowany, na przykład przez kliknięcie tekstu elementu konspektu w jego interfejsie użytkownika, zgodny czytnik przechodzi do miejsca docelowego lub wywołuje akcję powiązaną z tym elementem. W przeciwieństwie do niektórych innych formatów książek lub czytania, PDF łączy wpisy nawigacyjne sterowane przez autora z dodanymi przez użytkownika linkami (zakładkami) do ich ulubionych sekcji. Podstawą hierarchii konspektu dokumentu jest słownik konspektu określony przez pozycję Kontury w katalogu dokumentów. Każdy pojedynczy element konspektu w hierarchii jest definiowany przez słownik elementów konspektu. Elementy na każdym poziomie hierarchii tworzą podwójnie połączoną listę, połączoną ze sobą za pomocą ich pozycji Poprzedni i Następny i dostęp do nich poprzez pozycje Pierwszy i Ostatni w elemencie nadrzędnym (lub w słowniku konspektu w przypadku elementów najwyższego poziomu). Wyświetlane na ekranie elementy na danym poziomie pojawiają się w kolejności, w jakiej się znajdują występują na połączonej liście. W dużym lub złożonym dokumencie hierarchia konspektu może być bardzo duża i głęboka, a wyświetlenie go w całości przez widza przyniosłoby efekt przeciwny do zamierzonego. W tym celu każdy element konturu, który ma dzieci, może również określać, czy dzieci mają być wyświetlane (otwarte), czy ukryte (zamknięte). Niestety, klucz używany do określania widoczności jest również kluczem używanym do liczby wpisów konspektu - liczba. Liczba jest zdefiniowana jako „suma liczby widocznych elementów konturu na wszystkich poziomach”. Czasami jednak wartość jest ujemna. Jak może być sumą, jeśli jest ujemna? Znak sumy wskazuje, czy element konturu ma być wyświetlany jako otwarty czy zamknięty w interfejsie użytkownika czytnika. Jeśli liczba jest dodatnia, wyświetlane są dzieci; jeśli negatywne, to dzieci są ukryte. W przykładzie 5-7 widać, że czytelnik pokaże pięć widocznych elementów konturu, ponieważ liczba w obiekcie 21 jest dodatnia. Jednak, jak widać na poniższym rysunku, nie wszystkie z tych pięciu są przedmiotami najwyższego poziomu (bezpośrednie potomki roota) - są tylko dwa z nich (obiekty 22 i 29). Możesz zobaczyć, że są to dzieci roota, podążając za pierwszym łączem z obiektów 21 do 22, a następnie następnym łączem z obiektów 22 do 29. Obiekt 22 ma troje potomków (wszystkie będą widoczne): obiekty 25, 26 i 28 I na koniec obiekt 26 ma jedno dziecko (obiekt 27), ale zostanie pokazany jako zamknięty, ponieważ wartość Count dla tego obiektu wynosi –1.

Przykład 5-7. Przykładowy kontur

On-screen appearance	Object number	Count
	21	5
Document	22	3
Section 1	25	0
Section 2	26	–1
Section 3	28	0
Summary	29	0

21 0 obj

<<

/Type /Outlines

/First 22 0 R

/Last 29 0 R

/Count 5

>>

endobj

22 0 obj

<<

/Title (Document)

/Parent 21 0 R

/Next 29 0 R

/First 25 0 R

/Last 28 0 R

/Count 3

/Dest [3 0 R /XYZ 0 792 0]

>>

endobj

25 0 obj

<<

/Title (Section 1)

/Parent 22 0 R

/Next 26 0 R

/Dest [3 0 R /XYZ null 701 null]

>>

endobj

26 0 obj

<<

/Title (Section 2)

/Parent 22 0 R

/Prev 25 0 R

/Next 28 0 R

/First 27 0 R

/Last 27 0 R

/Count −1

/Dest [3 0 R /XYZ null 680 null]

>>

endobj

27 0 obj

<<

/Title (Subsection 1)

/Parent 26 0 R

/Dest [3 0 R /XYZ null 670 null]

>>

endobj

28 0 obj

<<

/Title (Section 3)

/Parent 22 0 R

/Prev 26 0 R

/Dest [7 0 R /XYZ null 500 null]

>>

endobj

29 0 obj

<<

/Title (Summary)

/Parent 21 0 R

/Prev 22 0 R

/Dest [8 0 R /XYZ null 199 null]

>>

endobj

Poprzedni przykład jest najczęstszym rodzajem obrysu dokumentu, w którym wszystkie elementy używają miejsc docelowych do wskazywania określonych części dokumentu. Jednak zamiast klawisza docelowego dowolny element konturu może mieć klawisz A wskazujący akcję. Na przykład, jak pokazano w przykładzie 5-8, możesz mieć element konturu, który używa akcji URI, aby przenieść użytkownika do gry online powiązanej z dokumentem.

Przykład 5-8. Kontur z działaniem URI

```
29 0 obj
<<
/Title (Play Online Game)
/Parent 21 0 R
/Prev 22 0 R
/A <<
/Type /Action
/S /URI
/URI (http://www.someexamplegamesite.com)
>>
>>
endobj
```

Adnotacje

Ta część szczegółowo omówi szczególny typ obiektu w PDF - adnotację. Adnotacje to obiekty PDF, które umożliwiają klikalne czynności użytkownika, a także nowe typy treści, w tym 3D, wideo i audio.

Wprowadzenie

Treść, którą użytkownik widzi na stronie PDF, opisuje się za pomocą strumienia treści operatorów graficznych PDF. Czasami jednak konieczne jest nałożenie tej zawartości na aktywne lub „gorące" obszary lub dodatkową grafikę. Możesz także chcieć wprowadzić na swoich stronach nowe typy treści, takie jak grafika 3D lub filmy, których nie można opisać za pomocą tych operatorów graficznych. Aby to osiągnąć, używany jest obiekt adnotacji. Podobnie jak w przypadku zawartości strony, adnotacje są powiązane bezpośrednio ze stroną. W dowolnym słowniku stron dla strony zawierającej adnotacje musi istnieć klucz Annots, którego wartością jest tablica adnotacji.

Słowniki adnotacji

Obiekty adnotacji to obiekty słownika zawierające co najmniej dwa klucze: Prostokąt i Podtyp. Klucz Rect ma oczywiście wartość, która jest prostokątem reprezentującym współrzędne strony, w której adnotacja ma zostać umieszczona przez aplikację do przeglądania. Klucz podtypu ma wartość nazwy, która reprezentuje który z 27 typów adnotacji jest opisywany przez ten konkretny słownik. Wiele rodzajów adnotacji będzie również miało klucz Contents, którego wartością jest ciąg tekstowy wyświetlany bezpośrednio na stronie lub alternatywny opis zawartości adnotacji w formie czytelnej dla człowieka. W zależności od rodzaju adnotacji pozostałe klucze w słowniku adnotacji może być wymagane lub opcjonalne. Niektóre z tych kluczy są wspólne dla wszystkich typów adnotacji, podczas gdy inne są specyficzne dla określonego typu. ISO 32000 dzieli adnotacje na dwie kategorie: znaczniki (omówione poniżej) i niezaznaczające (omówione w „Adnotacje niezwiązane ze znacznikami" na stronie 102). ISO 32000-1 definiuje większość adnotacji jako adnotacje znaczników, ponieważ są one używane przede wszystkim do oznaczania w górę zawartość dokumentu PDF. Dodatkowo adnotacje te zawierają tekst, który pojawia się jako część adnotacji i może być wyświetlany w inny sposób przez zgodnego czytelnika.

Strumienie wyglądu

Słowniki adnotacji mogą zawierać pozycje opisujące kształty i kolory, które mają zostać narysowane przez przeglądarkę PDF, zamiast strumienia treści. Jednak plik PDF, który wymaga gwarantowanego wyglądu, może również kojarzyć specjalny typ strumienia treści, zwany wyglądem strumienia z adnotacją. Odbywa się to poprzez włączenie klucza AP do słownika adnotacji, którego wartością jest słownik wyglądu, który odwołuje się do jednego lub więcej strumieni wyglądu. Gdy obecny jest strumień wyglądu, przeglądarka po prostu go narysuje zamiast odtwarzać instrukcje rysowania przy użyciu innych wartości. Słownik wyglądu, który odwołuje się do strumienia wyglądu, może zawierać maksymalnie trzy klucze: N (wygląd normalny), R (wygląd najazdu) i D (wygląd dolny). Do omawianych tutaj typów adnotacji używamy tylko klawisza N, ale AcroForms, omówione w części 7, obsługuje pozostałe.

Adnotacje znaczników

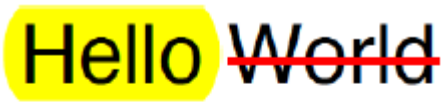
Jak sama nazwa wskazuje, te typy adnotacji służą do stosowania różnego rodzaju typowych operacji znaczników na górze strony. Istnieje wiele rodzajów znaczników, które można zastosować, od prostych wyróżnień i StrikeOut po ogólny atrament i specjalną redakcję.

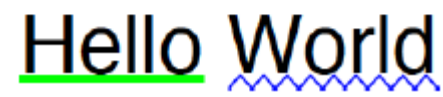
Tabela 6-1. Adnotacje znaczników

Rodzaj adnotacji	Opis
Caret	Caret annotation
Circle	Circle annotation
FileAttachment	File attachment annotation
FreeText	Free text annotation
Highlight	Highlight annotation
Ink	Ink annotation
Line	Line annotation
Polygon	Polygon annotation
PolyLine	Polyline annotation
Redact	Redact annotation
Square	Square annotation
Squiggly	Squiggly-underline annotation
Sound	Sound annotation
Stamp	Rubber stamp annotation
StrikeOut	Strikeout annotation
Text	Text annotation
Underline	Underline annotation

Oznaczanie tekstu

Są to rodzaje adnotacji, które większość użytkowników zna podczas komentowania wyróżnienia w formacie PDF, podkreślenia i przekreślenia, a także niezwykłego dodania squiggly (poszarpanego podkreślenia). W każdym przypadku podtyp będzie jednym z te cztery wartości (Podświetl, Podkreśl, StrikeOut lub Squiggly), a Prostokąt obejmie obszar na stronie (zwykle obejmujący tekst), w którym zostanie narysowana graficzna reprezentacja adnotacji. Dodatkowo chcemy zdefiniować kolor, w którym adnotacja zostanie narysowana, więc zostanie użyty klawisz C. Wartość klucza C to tablica 1, 3 lub 4 wartości reprezentujących nie tylko wartość koloru, ale także (domyślnie) przestrzeń kolorów, w której ta wartość powinna być obsługiwana. Jednoelementowa tablica reprezentuje DeviceGray, trzy elementy dla DeviceRGB i cztery elementy dla DeviceCMYK. Rysunek pokazuje kilka przykładowych adnotacji do znaczników tekstowych.





9 0 obj

<<

/Type /Page

/Annots [20 0 R 22 0 R 24 0 R 26 0 R]

% other stuff that we need for a valid page dictionary

>>

20 0 obj

<<

/Type /Annot

/Subtype /Highlight

/Rect [252.594 593.733 322.077 623.211]

/C [1 1 0]

>>

22 0 obj

<<

/Type /Annot

/Subtype /StrikeOut

/Rect [313.07 592.866 392.279 624.078]

/C [1 0 0]

>>

24 0 obj

<<

/Type /Annot

/Subtype /Underline

/Rect [251.727 528.21 322.944 559.422]

/C [0 1 0]

>>

26 0 obj

<<

/Type /Annot

/Subtype /Squiggly

/Rect [313.07 528.21 392.279 559.422]

/C [0 0 1]

>>

Jeśli adnotowany tekst jest obracany lub w inny sposób przekształcany, można użyć przycisku Quad Points, aby zdefiniować układ ośmiu punktów reprezentujących cztery rogi czworoboku, który ogranicza tekst. Chociaż nie potrzebujesz QuadPoints do prostych obrotów (0, 90, 180, 270), posiadanie jednego nie będzie bolało, ponieważ prostokąt opisany w ośmiu punktach zamiast czterech jest całkowicie do przyjęcia. Rysunek 6-2 pokazuje przykład podświetlonego tekstu, który również jest obracany.

51 0 obj

<<

/Type /Annot

/Subtype /Highlight

/Rect [258.483 291.978 368.238 401.733]

/QuadPoints [259.709 312.822 347.394 400.507 279.327 293.204 367.012

380.889]

/C [1 1 0]

>>

Chociaż tablica QuadPoints jest bardziej szczegółowa niż Rect, to zawsze musi być obecny Rect, ponieważ służy on jako „maksymalny obszar ograniczający” dla QuadPoints. W rzeczywistości ISO 32000-1 mówi „QuadPoints powinny być ignorowane, jeśli jakakolwiek współrzędna w macierzy leży poza obszarem określonym przez Rect”.

Rysowanie znaczników

PDF definiuje serię sześciu rodzajów adnotacji, które umożliwiają operacje przypominające rysowanie, których można użyć do oznaczenia pliku PDF. Chociaż niefortunnie jest, że każdy z nich ma nieco inny sposób opisywania reprezentowanego kształtu graficznego, wszystkie obsługują standardowy zestaw atrybutów, takich jak kolory, ciężary i inne.

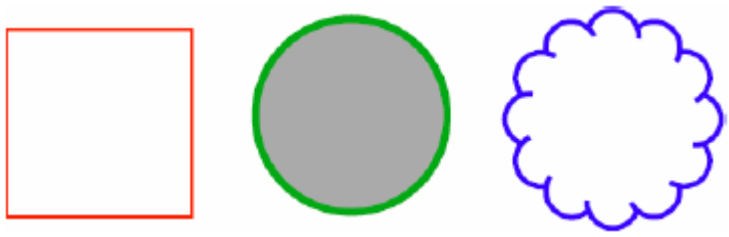
Atrybuty

Dwa najczęstsze atrybuty opisują rysowane linie (lub pociągnięcia, jeśli wolisz): kolor (i powiązaną przestrzeń kolorów), a także grubość linii i dowolny wzór kreski. Zarówno klawisze C (obrys), jak i IC (wypełnienie) używają tablicy liczb do określenia przestrzeni kolorów i wartości kolorów. Klucz BS ma słownik wartości w stylu ramki. Jak opisano w ISO 32000-1: 2008, tabela 166, słownik ten składa się tylko z dwóch możliwych kluczy: S dla stylu obramowania i D dla tablicy kresek. Styl obramowania może mieć wartości określające pełne obramowanie (S), obramowanie przerywane (D), różne rodzaje wytłoczeń (B i I) oraz styl podkreślenia (U). Tablica myślowa przybiera tę samą formę, co podczas dołączania jej do strumienia treści. Style wytłaczania są przydatne przede wszystkim do kształtów. Chociaż ty możesz podać je w prostej linii, prawdopodobnie nie otrzymasz efektu, którego pragniesz.

Kwadraty i kółka

Adnotacje kwadratowe i okręgowe są najprostszymi z adnotacji rysunkowych, ponieważ nie potrzebują żadnych dodatkowych kluczy w swoich słownikach. Zamiast tego polegają na istniejącym klawiszu Rect dla obszaru graniczącego z kształtem oraz klawiszach BS i IC dla dowolnej stylizacji. W przypadku adnotacji kwadratowych i okręgowych wartość klucza IC służy do wypełnienia całego kształtu, podobnie jak operator F wypełniłby kształt w strumieniu zawartości.

Jednym z fajnych dodatkowych kluczy, które można określić w tych adnotacjach, jest klucz BE, którego można użyć do przekształcenia prostego okręgu (lub prostokąta) w taki, który przypomina chmurę. Wartość klawisza BE to słownik efektów granicznych, który może zawierać albo klawisz S (o wartości S dla „prostej” lub C dla „pochmurnej”) lub klawisz I z liczbą całkowitą od 0 do 2, która deklaruje intensywność efektu. Dołączając efekt obramowania (BE), często przydatne jest włączenie klucza RD do słownika adnotacji. Pozwala określić „margines przesunięcia” od prostokąta ograniczającego do prostokąta używanego do narysowania kształtu. Rysunek 6-3 pokazuje kilka przykładów adnotacji w kwadracie i kole.



% A red stroked square

60 0 obj

<<

/C [1 0 0]

/RD [0.50 0.50 0.50 0.50]

/Rect [281.239990 680.687012 355.390991 754.838013]

/Subtype Square

/Type Annot

>>

% A circle stroked in (RGB) green and filled in (grayscale) gray

62 0 obj

<<

/BS << /W 3 >>

/C [0 1 0]

/IC [.50]

/RD [1.50 1.50 1.50 1.50]

/Rect [375.842987 680.411011 456.338989 760.908020]

/Subtype Circle

/Type Annot

>>

% A blue circle with a cloud effect

64 0 obj

<<

/BE << /I 2 /S C >>

/BS << /W 2 >>

/C [0 0 1]

/RD [1 1 1 1]

/Rect [483.533997 685.257019 551.442993 753.164978]

/Subtype Circle

/Type Annot

>>

Kwestia

Adnotacja linii służy do narysowania prostej linii. W słowniku ma jeden wymagany klucz: L, który jest tablicą czterech liczb określających dwa punkty końcowe linii. Dodatkowym kluczem obsługiwanym przez słownik adnotacji linii jest klawisz LE, który umożliwia specyfikację różnych typów zakończeń linii, w tym grotów strzałek, ramek i okręgów

% Simple red line

20 0 obj

<<

/Type Annot

/Subtype Line

/C [1 0 0]

/L [30.098700 755.213013 204.001999 755.213013]

/Rect [24.598700 749.713013 209.501999 760.713013]

>>

% Dashed green line

22 0 obj

<<

/Type Annot

/Subtype Line

/BS << /D [4 4] /S D /W 5 >>

/C [0 1 0]

/L [28.869200 717.310974 202.772995 717.310974]

/Rect [21.369200 709.810974 210.272995 724.810974]

>>

% Blue line with an arrowhead at the end

24 0 obj

<<

/Type Annot

/Subtype Line

/BS << /W 3 >>

/C [0 0 1]

/L [30.098700 687.211975 200.658005 687.211975]

/LE [None OpenArrow]

/Rect [23.598700 680.711975 207.158005 693.711975]

>>

Istnieje 10 różnych rodzajów możliwych zakończeń linii, jak widać w tabeli 6-2.

Tabela 6-2. Rodzaje zakończeń linii dla adnotacji linii

Nazwa	**Opis**
Kwadrat	Kwadrat wypełniony kolorem adnotacji, jeśli występuje
Koło	Koło wypełnione kolorem adnotacji, jeśli występuje
Romb	Kształt rombu wypełniony kolorem adnotacji, jeśli występuje
OpenArrow	Dwie krótkie linie spotykające się pod ostrym kątem, tworząc otwarty grot strzałki
ClosedArrow	Dwie krótkie linie spotykające się pod ostrym kątem, jak w stylu OpenArrow i połączone trzecią linią uformowanym trójkątnym zamkniętym grotem strzałki wypełniony kolorem adnotacji, jeśli istnieje
None	Brak zakończenia linii
Butt	Krótka linia w punkcie końcowym prostopadła do samej linii
ROpenArrow	Dwie krótkie linie ustawione pod kątem do tyłu w stosunku do OpenArrow
RClosedArrow	Trójkątny zamknięty grot strzałki w odwrotnym kierunku od ClosedArrow
Ukośnik	Krótka linia w punkcie końcowym około 30 stopni w kierunku zgodnym z ruchem wskazówek zegara od prostopadłej do samej linii

Wieloboki i łamane

Podczas gdy pojedyncza linia jest przydatna, w większości przypadków seria linii, które łączą się ze sobą, tworząc kształt otwarty (łamana) lub zamknięty (wielokąt), zajmie się większą liczbą przypadków rysowania i oznaczania treści na stronie. Podobnie jak w przypadku adnotacji liniowych i kwadratowych, klawisze LE, BS, BE, IC i C służą do zapewnienia pewnych atrybutów stylu. W przypadku adnotacji łamanych wartość klucza IC służy do wypełnienia tylko końca linii. Jednak w przypadku adnotacji wielokąta wartość klucza IC służy do wypełnienia całego kształtu, podobnie jak operator F wypełniłby kształt w strumieniu zawartości. Rzeczywiste punkty wzdłuż kształtu są określone jako wartość tablicowa dla klucza wierzchołków z naprzemiennymi współrzędnymi poziomymi i pionowymi. Podobnie jak w przypadku QuadPoints, mimo że tablica Vertices jest bardziej specyficzna niż Rect, Rect zawsze musi być obecny. Innym kluczem charakterystycznym dla adnotacji łamanych i wielokątów jest

klucz IT, który określa przeznaczenie obiektu. Pobiera wartość nazwy PolygonCloud, PolyLineDimension lub PolygonDimension. Rysunek 6-5 przedstawia kilka przykładów adnotacji wielokąta i łamanej.

% red polyline (as the shape is open)

105 0 obj

<<

/Type Annot

/Subtype PolyLine

/C [1 0 0]

/Rect [272.049011 536.856018 372.549011 640.978027]

/Vertices [301.295013 626.216003 273.049011 578.414978

316.505005 537.856018 371.549011 578.414978

314.332001 575.518005 323.747009 602.315979

349.096985 597.969971 350.545013 625.492004

350.545013 624.768005 323.023010 639.978027]

>>

% polygon stroked in (RGB) green and filled in (CMYK) yellow

107 0 obj

<<

/Type Annot

/Subtype Polygon

/BS << /W 5 >>

/C [0 1 0]

/IC [0 0 1 0]

/Rect [376.688995 529.958984 491.707001 638.458984]

/Vertices [399.071014 600.143005 415.729004 613.179993

446.148010 593.624023 415.005005 565.377991

388.931000 579.862976 381.688995 559.583984

429.489990 534.958984 486.707001 574.794006

459.908997 613.179993 400.519989 633.458984

399.071014 600.143005]

>>

% polygon stroked in (grayscale) black

109 0 obj

<<

/Type Annot

/Subtype Polygon

/BE << /I 2 /S C >>

/BS << /W 3 >>

/C [0]

/IT PolygonCloud

/Rect [497.958008 525.330994 598.528015 633.752014]

/Vertices [512.057007 621.146973 585.932007 601.591003

583.034973 537.856018 510.608002 553.789978

510.608002 582.036011 512.057007 621.146973]

>>

Ink

Mimo że adnotacje wielokąta i polilinii są tak elastyczne, czasem potrzebny jest całkowicie dowolny rysunek łączący serię dowolnych punktów. Dodatkowo segmenty wielokąta i polilinii są połączone prostymi liniami, ale czasami krzywa jest konieczna. W tym celu adnotacje odręczne podano w formacie PDF. Podobnie jak w przypadku innych typów adnotacji, które widzieliśmy, klawisze LE, BS i C służą do zapewnienia niektórych atrybutów stylizacji (patrz Przykład 6-1). Same punkty są przechowywane w tablicy, która jest wartością klucza InkList, a ponieważ jest to zwykle bardzo długa lista, dość często (choć nie jest to wymagane) mieć listę jako oddzielny obiekt pośredni.

Przykład 6-1. Przykłady adnotacji atramentowych

% red ink

141 0 obj

<<

/C [1 0 0]

/InkList 126 0 R

/Rect [40.283199 451.471008 158.264999 517.778992]

/Subtype Ink

/Type Annot

>>

% green ink

143 0 obj

<<

/BS << /W 5 >>

/C [0 1 0]

/InkList 192 0 R

/Rect [186.205994 427.546997 299.170013 516.556030]

/Subtype Ink

/Type Annot

>>

% magenta ink

145 0 obj

<<

/BS << /W 3 >>

/C [0 1 0 0]

/InkList 209 0 R

/Rect [323.010986 408.593994 479.394989 506.535004]

/Subtype Ink

/Type Annot

>>

126 0 obj

[

41.2832 487.808 42.0075 488.532 42.731700000000004 489.256 44.1803 490.705

45.6288 492.154 47.0773 493.602 49.9744 496.499 52.8715 498.672 55.7685

57.9413 503.018 61.5627 506.639 65.184 508.812 68.0811 510.984 69.5296

70.2539 513.882 72.4267 515.33 72.4267 516.054 73.8752 516.779 73.1509

72.4267 513.882 70.2539 511.709 68.0811 507.363 64.4597 503.018 60.1141

56.4928 493.602 55.0443 487.808 52.8715 483.462 52.1472 479.841 52.1472

52.8715 476.22 54.32 475.495 55.7685 475.495 58.6656 476.22 62.2869 478.392

66.6325 480.565 70.2539 482.738 73.8752 484.911 75.3237 487.808 76.7723

77.4965 492.154 77.4965 493.602 77.4965 494.326 77.4965 493.602 76.7723

% Lots more numbers would go here...

]

192 0 obj

[

348.372 500.845 348.372 500.12 348.372 499.396 348.372 498.672 348.372

348.372 497.223 347.648 496.499 346.924 495.775 346.199 495.775 345.475

344.751 493.602 343.302 493.602 342.578 492.154 341.854 491.429 341.13

340.405 489.981 339.681 489.981 338.957 489.981 338.233 489.981 336.784

336.06 490.705 334.611 491.429 333.163 492.154 330.99 493.602 330.266

328.817 494.326 328.093 495.051 327.369 495.051 326.644 495.775 326.644

326.644 497.223 326.644 497.948 326.644 498.672 328.093 498.672 328.817

330.266 500.12 332.438 500.845 333.887 500.845 336.784 501.569 339.681

344.027 502.293 348.372 502.293 351.994 502.293 356.339 502.293 359.236

362.133 502.293 364.306 502.293 366.479 502.293 368.652 501.569 369.376

370.825 498.672 371.549 497.223 372.273 495.775 372.997 494.326 372.997

% Lots more numbers would go here...

]

209 0 obj

[

191.206 504.466 191.206 503.742 191.206 502.293 191.206 500.845 191.206

191.931 496.499 191.931 492.878 192.655 487.808 193.379 484.911 193.379

194.828 478.392 194.828 476.22 194.828 474.047 195.552 473.323 196.276

197.001 470.426 197.001 471.15 197.725 471.874 197.725 474.047 197.725

198.449 479.117 199.173 482.738 200.622 485.635 202.07 489.981 204.243

205.692 498.672 207.14 503.018 209.313 506.639 210.762 508.812 210.762

210.762 510.984 211.486 510.984 211.486 508.812 211.486 505.915 211.486

211.486 497.223 211.486 492.878 211.486 487.808 211.486 482.738 211.486

212.21 473.323 212.934 468.977 213.659 467.528 213.659 466.08 214.383

% Lots more numbers would go here...

]

Znaczki znaczników

Gdy typy adnotacji rysunkowych nie są wystarczająco bogate, aby reprezentować grafikę, która ma zostać narysowana - na przykład, jeśli chcesz dołączyć obraz rastrowy - adnotacja stemplowa jest wyborem. Jest pod pewnymi względami najprostszą adnotacją do utworzenia, ponieważ słownik adnotacji wymaga tylko kluczy Typ i Podtyp ze słownika standardowego. Jednak to, co komplikuje to, że wymaga również klucza AP i powiązanego z nim strumienia wyglądu

25 0 obj

<<

/AP << /N 18 0 R >>

/CA 0.5 % set the opacity to 50%

/Rect [109.597 104.905 206.597 201.905]

/Subtype Stamp

/Type Annot

>>

18 0 obj

<<

/Type /XObject

/Subtype /Form

/BBox [0 0 147 147]

/FormType 1

/Length 74 % or whatever it really is

/Matrix [1 0 0 1 0 0]

/Resources << /XObject << /FRM 20 0 R >> >> % refers to subsequent XObject

stream

q

0 0 147 147 re

W n

q

0 0 147 147 re

W n

1 0 0 1 73.5 73.5 cm

/FRM Do

Q

Q

endstream

endobj

20 0 obj

<<

/Type /XObject

/Subtype /Form

/BBox [249 421 396 568]

/FormType 1

/Length 348 % or whatever it really is

/Matrix [1 0 0 1 -322.5 -494.5]

stream

0 0 1 0 k

0 0 0 1 K

0 J 0 j 6 w 4 M []0 d

/GS2 gs

1 i

322.787 425.358 m

360.798 425.358 391.612 456.173 391.612 494.184 c

391.612 532.194 360.798 563.009 322.787 563.009 c

284.777 563.009 253.962 532.194 253.962 494.184 c

253.962 456.173 284.777 425.358 322.787 425.358 c

b

0 g

353.747 517.033 m

353.747 522.313 350.147 526.153 345.107 526.153 c

340.067 526.153 335.987 522.073 335.987 517.273 c

335.987 512.473 340.307 508.633 344.867 508.633 c

349.427 508.633 353.747 512.233 353.747 517.033 c

f

307.08 517.033 m

307.08 522.313 303.48 526.153 298.44 526.153 c

293.4 526.153 289.32 522.073 289.32 517.273 c

289.32 512.473 293.64 508.633 298.2 508.633 c

302.76 508.633 307.08 512.233 307.08 517.033 c

f

1 J

278.216 481.06 m

284 468.393 298.267 456.076 323 456.076 c

347.733 456.076 362 469.06 367.783 481.06 c

S

endstream

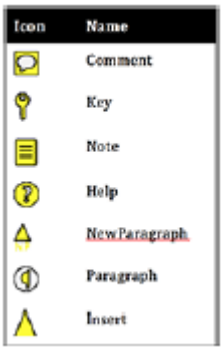
endobj

W poprzednim przykładzie forma XObject odwołuje się do innej formy XObject. Nie jest to wymagane w przypadku adnotacji na znaczku, ale jest to powszechny wzór. W tym przykładzie słownik adnotacji zawiera klucz CA, który określa stopień krycia stosowany do całego strumienia wyglądu adnotacji, gdy jest ona rysowana na stronie. Inną interesującą rzeczą, którą pokazuje ten przykład, jest to, że nawet forma XObject zastosowana w strumieniu wyglądu adnotacji może odnosić się do dodatkowych zasobów, w tym dowolnej liczby kolejnych i zagnieżdżonych postaci XObject

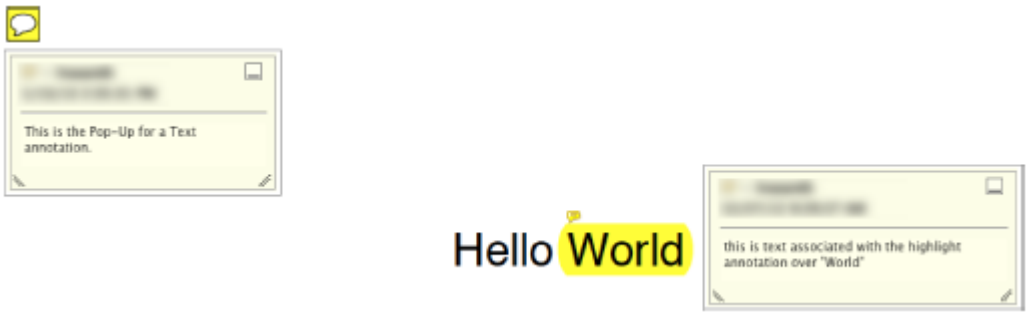
Adnotacje tekstowe i wyskakujące okienka

Adnotacja tekstowa służy do przedstawienia „karteczki" umieszczonej w określonym miejscu na stronie PDF. Tekst notatki jest wartością ciągu dla klucza Contents w słowniku adnotacji. Chociaż do dostarczenia instrukcji rysowania należy użyć standardowego strumienia wyglądu, możliwe jest

również podanie klucza nazwy z wartością, która daje przeglądarce PDF wskazówkę, jakiej ikony użyć. ISO 32000-1, tabela 172 zawiera listę możliwych nazw, które musi podać przeglądarka plików PDF. Niektóre typowe ikony używane dla tych nazw są następujące:



Wyskakująca adnotacja to specjalny rodzaj adnotacji, który istnieje, aby umożliwić powiązanie ciągu tekstu z adnotacjami znaczników. Nie ma własnego wyglądu, ale zamiast tego w słowniku adnotacji ma klucz nadrzędny, który wskazuje na inną adnotację, która jest jedną z adnotacji znaczników. Ta adnotacja nadrzędna zapewnia rzeczywisty strumień wyglądu adnotacji, jeśli istnieje. Ponadto słownik adnotacji nadrzędnej zawiera klucz Contents, którego wartość jest łańcuchem określającym tekst, który ma być wyświetlany w „notatce". Oprócz adnotacji nadrzędnej, podręczny słownik adnotacji może także zawierać klucz Open, którego wartość logiczna określa, czy przeglądarka powinna domyślnie pokazywać, że jest otwarta czy zamknięta



19 0 obj

<<

/Popup 20 0 R

/Name /Comment

/Contents (This is the pop-up for a text annotation)

/Type /Annot

/Rect [261.52 558.755 279.52 576.755]

/Subtype /Text

>>

endobj

20 0 obj

<<

/Parent 19 0 R

/Subtype /Popup

/Type /Annot

/Rect [287.427 457.785 467.428 578.222]

/Open true

>>

endobj

21 0 obj

<<

/Subtype /Highlight

/C [1 1 0]

/Popup 22 0 R

/QuadPoints [321.343 622.344 384.006 622.344 321.343 594.6 384.006 594.6]

/Contents (this is text associated with the highlight annotation over

"World")

/Type /Annot

/Rect [313.937 593.733 391.412 623.211]

>>

endobj

22 0 obj

<<

/Parent 21 0 R

/Subtype /Popup

/Type /Annot

/Rect [395.62 592.048 575.62 712.485]

/Open true

>>

endobj

Adnotacje bez znaczników

Tabela 6-3. Adnotacja bez znaczników

Rodzaj adnotacji	Opis
3D	Adnotacja 3D
FileAttachment	Załącznik pliku
Link	adnotacja Link
Movie	adnotacja Movie
Popup	adnotacja Pop-up
PrinterMark	adnotacja oznaczenia drukarki n
Screen	adnotacja Screen
Sound	Dźwięk
TrapNet	Adnotacja pułapki sieci
Watermark	adnotacja znaku wodnego
Widget	wadnotacja Widget

Adnotacje bez znaczników można podzielić na kilka podkategorii:

Elementy interaktywne

Widzieliśmy już adnotacje o linkach w naszym dyskursie na temat działań GoTo w przykładzie 5-3, a adnotacje widżetów mają dedykowany im cały parter (część 7).

Treści multimedialne

Te typy adnotacji umożliwiają wyświetlanie multimediów (w tym filmów, dźwięku i 3D) i interakcję z nimi w pliku PDF. Wszystkie zostały omówione w części 9.

Produkcja drukowania

PDF obsługuje dwa typy adnotacji dla branży poligraficznej: PrinterMark i TrapNet. Ponieważ nikt ich nie używa, to wszystko o nich powiemy.

Inne

Istnieją dwa inne rodzaje adnotacji, które nie należą do żadnej z tych kategorii:

FileAttachment and Watermark.

Adnotacje załączników do plików to jeden ze sposobów osadzania / załączania plików do plików PDF, podobnie jak w przypadku wiadomości e-mail. Są one omówione w części 8. Adnotacja znaku wodnego została wprowadzona do pliku PDF w celu wykorzystania przez inżynierów jako sposób na oznaczenie specjalnej grafiki, która nie jest skalowana podczas drukowania. Nie są one często używane, dlatego też nie będą tutaj dostępne.

AcroForms

W tej części szczegółowo omówimy specjalny rodzaj adnotacji: adnotację widgetu, która jest budulcem formularzy PDF. W PDF 1.2 wprowadzono koncepcję interaktywnego formularza (zbioru pól, które można wykorzystać do interaktywnego gromadzenia informacji od użytkownika) do formatu PDF. W pliku PDF będzie co najwyżej jedna ogólna forma; może zawierać dowolną liczbę pól, które mogą pojawiać się na dowolnej kombinacji stron. Formularzy interaktywnych nie należy mylić z obiektami XObjects. Pomimo podobieństwa ich nazw, oba są różnymi, niepowiązanymi typami obiektów.

Słownik formularzy interaktywnych

Interaktywny formularz dokumentu jest opisany przy użyciu interaktywnego słownika formularzy, który jest wartością klucza AcroForm znalezionego w słowniku katalogu dokumentów. Ponieważ taka jest nazwa klucza, interaktywny słownik formularzy jest często nazywany słownikiem AcroForm, a typ formularza - AcroForm. Ten słownik ma tylko jeden wymagany klucz, Pola, który określa tablicę słowników pól reprezentujących pola w formularzu. Istnieje jednak kilka typowych pól opcjonalnych, które mogą być konieczne w przypadku niektórych formularzy:

DR

Podczas tworzenia strumieni wyglądu dla swoich pól możesz odwoływać się do niektórych zasobów współdzielonych przez jedno lub więcej pól. Słownik zasobów określony przez ten klucz służy do dostarczania tych informacji, podobnie jak słownik zasobów strony dla zawartości strony. Jeśli klucz DR jest obecny, jego wartość (typu słownik) musi zawierać klawisz Font. Wartość czcionki to nazwa zasobu i czcionka, która ma być używana jako domyślna czcionka do wyświetlania tekstu w polach.

NeedAppearances

To pole powinno być obecne i ustawione na true, gdy jedno lub więcej pól w formularzu nie ma strumienia wyglądu. Jest to wskazówka dla widza, aby wymusić odbudowę takich strumieni; jednak wielu widzów tego nie szanuje. Dlatego zaleca się, aby zawsze podawać strumienie wyglądu dla każdego pola.

XFA

PDF 1.5 wprowadził specjalny typ formularza o nazwie XFA (eXtensible Form Architecture), który wykorzystuje gramatykę XML do opisania formy, a nie natywną składnię PDF. Wartością tego klucza, jeśli jest obecny, jest tablica fragmentów XML opartych na gramatyce XFA.

Słownik polowy

Słownik pól określa szczegóły każdego pola w interaktywnej formie dokumentu. Pola mogą być uporządkowane hierarchicznie i mogą dziedziczyć atrybuty po przodkach w hierarchii. Pole, które ma dzieci, które są polami, nazywa się polem nieterminalnym. Pole, które nie ma potomków, takie jak prosty przycisk lub pole tekstowe, nazywa się polem terminalu. Pola potomne są określone w tablicy słowników pól, która jest wartością klucza Kids, a rodzic dziecka jest określony za pomocą klucza Parent. Każde pole musi mieć przypisaną klasę za pomocą klucza FT w słowniku pól (lub dziedziczonego z pola nadrzędnego). Istnieją cztery możliwe wartości dla klasy pola:

Btn

Pola przycisków, które obejmują przyciski, pola wyboru i przyciski opcji.

TX

Pola tekstowe, w których użytkownik może wprowadzać tekst.

Ch

Pola wyboru, takie jak przewijane listy, pola kombi lub menu podręczne.

Sig

Pola podpisu, aby włączyć podpisy cyfrowe w formacie PDF.

Ponieważ celem pola jest zebranie informacji od użytkownika, potrzebuje on sposobu na przechowywanie tej wartości. Ta lokalizacja pamięci jest wartością klucza V w słowniku polowym. Rodzaj wartości i jej związek z polem różni się w zależności od typu pola.

Nazwy pól

Chociaż nie jest to wymagane, większość pól ma powiązane z nimi nazwy. W rzeczywistości pole może mieć do trzech różnych nazw, z których każda jest używana w innym kontekście. Klawisz T w słowniku polowym składa się z wartości ciągu tekstowego, który definiuje częściową nazwę pola. W pełni kwalifikowana nazwa pola nigdy nie jest wyraźnie zdefiniowana w pliku PDF, ale jest dynamicznie konstruowana na podstawie częściowych nazw pól i wszystkich jego przodków. W przypadku pola bez elementu nadrzędnego częściowe i pełne nazwy są takie same. W przypadku pola, które jest dzieckiem innego pola, w pełni kwalifikowana nazwa jest konstruowana przez dołączenie częściowej nazwy pola potomnego do w pełni kwalifikowanej nazwy rodzica, oddzielone o kropkę (2Eh). Ponieważ kropka jest używana jako separator dla w pełni kwalifikowanych nazw, nazwa częściowa nie może zawierać znaku kropki.

Na przykład, jeśli pole o częściowej nazwie UserInfo ma potomka, którego częściowa nazwa to Address który z kolei ma dziecko z częściowa nazwa ZipCode, w pełni kwalifikowana nazwa tego ostatniego pola to User-Info.Address.ZipCode.

Podczas przygotowywania pliku PDF, który może być przetwarzany przez czytnik ekranu lub inną technologię wspomagającą, ważne jest podanie klucza TU dla każdego pola. Wartością tego klucza jest ciąg tekstowy, który jest używany zamiast częściowej lub w pełni kwalifikowanej nazwy pola w celu identyfikacji go dla użytkownika. Trzecia nazwa, jaką może mieć pole, to nazwa eksportu - nazwa używana podczas eksportowania danych do pliku. Wartość ciągu klucza TM zostanie wyeksportowana jako nazwa powiązana z wartością pola, gdy zostanie wyeksportowana do różnych formatów. Jeśli nie jest obecny, zostanie użyta częściowa nazwa pola.

Flagi pola

Słownik pól może zawierać (lub może dziedziczyć) klucz Ff, którego wartość jest 32-bitową liczbą całkowitą bez znaku złożoną z szeregu „flag bitowych” lub „pozycji bitowych”, które określają różne cechy pola. Te pozycje bitów w słowie flagowym są ponumerowane od 1 (niskiego rzędu) do 32 (wysokiego rzędu). Istnieją trzy flagi

Pozycja bitu	Nazwa	Znaczenie
1	ReadOnly	Jeśli jest ustawiony, użytkownik nie może zmienić wartości pola. Ta flaga jest przydatna w przypadku pól o wartościach obliczone lub zaimportowane z bazy danych.

2	Required	Jeśli jest ustawione, pole musi mieć wartość w momencie eksportowania przez akcję Prześlij formularz.
3	NoExport	Jeśli jest ustawiony, pole nie może być eksportowane przez akcję Prześlij formularz.

, które są wspólne dla wszystkich klas pól, ale jest też wiele flag, które dotyczą tylko określonych klas pól. Wszystkie niezdefiniowane bity flagi są uważane za zastrzeżone i są ustawione na 0 przez program piszący PDF.

Pola i adnotacje

Jedną z rzeczy, o których jeszcze nie wspomniano, jest sposób zdefiniowania strony, na której pole należy i gdzie należy go narysować na stronie. W związku z tym połączymy te dwie rzeczy, aby rozwiązać nasz problem. Adnotacja widżetu jest tworzona dla każdego pola, które ma zostać narysowane, a następnie powiązana (jak każda inna adnotacja) ze stroną, na której się pojawi. Oznacza to, że pojedynczy słownik będzie zawierał klucze zarówno ze słownika polowego, jak i słownika adnotacji. Jeśli pole jest rodzicem grupy (takiej jak grupa przycisków opcji), a sama grupa nie ma reprezentacji wizualnej i dlatego nie jest rysowana na stronie, będzie to standardowy słownik pól bez żadnych dodatków do adnotacji. Ten połączony słownik jest niezbędny / wymagany tylko dla dowolnego pola formularza które będą wyświetlane na stronie.

Przykład 7-1 pokazuje adnotację widżetu, która zawiera zarówno informacje adnotacji (np. Rect, AP), jak i informacje pola (np. FT, Ff).

Przykład 7-1. Przykład adnotacji pola i widżetu

```
16 0 obj
<<
/F 4
/Type /Annot
/Rect [27.014 749.644 99.014 769.644]
/FT /Btn
/Ff 65536
/Subtype /Widget
/T (PushButton)
/AP << /N 42 0 R >>
>>
endobj
```

Klasy pól

Jak wspomniano wcześniej, do każdego pola musi być przypisana klasa. Typy pól obejmują pola przycisków, tekstu, wyboru i podpisu.

Pola przycisków

Pole przycisku reprezentuje interaktywną kontrolkę lub element na stronie, który użytkownik może „nacisnąć” za pomocą myszy lub innego urządzenia wskazującego. Istnieją trzy rodzaje pól przycisków :

% Document catalog

12 0 obj

<<

/Metadata 2 0 R

/Type /Catalog

/Pages 9 0 R

/AcroForm 22 0 R

/Outlines 6 0 R

>>

endobj

% AcroForm dictionary

22 0 obj

<<

/Fields [16 0 R 17 0 R 24 0 R]

>>

endobj

% Pushbutton field

16 0 obj

<<

/F 4

/Type /Annot

/Rect [27.014 749.644 99.014 769.644]

/FT /Btn

/Ff 65536

/Subtype /Widget

/T (PushButton)

/AP << /N 42 0 R >>

>>

endobj

% Checkbox field

17 0 obj

<<

/Type /Annot

/Rect [27.014 715.314 45.014 733.314]

/FT /Btn

/AS /Off % default state is Off (unchecked)

/Subtype /Widget

/F 4

/T (CheckBox)

/AP <<

/N << /Yes 38 0 R /Off 37 0 R >>

/D << /Yes 41 0 R /Off 40 0 R >>

>>

>>

endobj

% Radio button #3

18 0 obj

<<

/F 4

/Rect [27.014 682.711 45.014 700.711]

/Parent 24 0 R % this is the radio button group...

/AS /Off % default state is Off (unchecked)

/Subtype /Widget

/Type /Annot

/AP <<

/N << /2 33 0 R /Off 34 0 R >>

/D << /2 35 0 R /Off 36 0 R >>

>>

>>

endobj

% Radio button #2

20 0 obj

<<

/F 4

/Rect [48.7541 682.711 66.7541 700.711]

/Parent 24 0 R % this is the radio button group...

/AS /Off % default state is Off (unchecked)

/Subtype /Widget

/Type /Annot

/AP <<

/N << /1 29 0 R /Off 30 0 R >>

/D << /1 31 0 R /Off 32 0 R >>

>>

>>

endobj

% Radio button #1

21 0 obj

<<

/F 4

/Type /Annot

/Rect [73.1377 682.711 91.1377 700.711]

/Parent 24 0 R % this is the radio button group...

/AS /0 % default state is 0, which is On

/Subtype /Widget

/AP <<

/N << /0 25 0 R /Off 26 0 R >>

/D << /0 27 0 R /Off 28 0 R >>

>>

>>

endobj

% this is the radio button grouping field

24 0 obj

<<

/Kids [21 0 R 20 0 R 18 0 R]

/FT /Btn

/Opt [(RadioButton1) (RadioButton1) (RadioButton1)]

/Ff 49152

/T (RadioButton)

>>

endobj

Pushbutton

Pole z bitem Pushbutton ustawionym na 1, które nie ma żadnej wartości (a zatem nie ma klawisza V w słowniku)

Checkbox

Pole, które przełącza między dwoma stanami, włączanie i wyłączanie, i ma wyczyszczone flagi Pushbutton i RadioButton. Stany odnoszą się do nazw strumieni wyglądu powiązanych z polem i są także wartościami nazw dla klucza V.

Radio button

Jeden z zestawu powiązanych przycisków, z których tylko jeden może być włączony jednocześnie. Podobnie jak w przypadku pól wyboru, każdy przycisk może mieć dwa stany (włączony i wyłączony) i są one używane w ten sam sposób. Te pola będą miały wyczyszczoną flagę Pushbutton, ale ustawioną flagę RadioButton.

W przypadku pól przycisków bity 15, 16, 17 i 26 flag pól wskazują zamierzone zachowanie, jak opisano w tabeli 7-2.

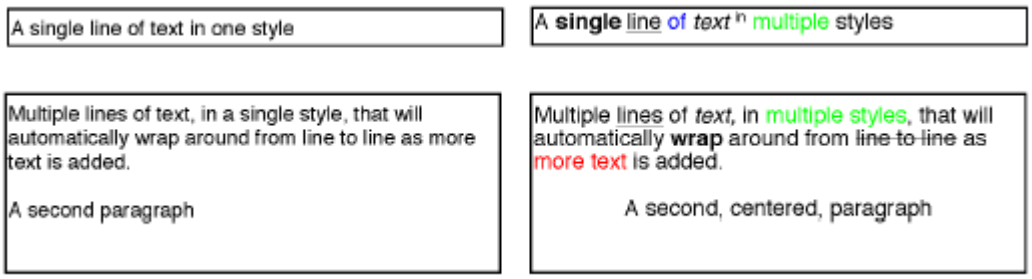
Tabela 7-2. Flagi pól przycisków

Pozycja bitu	Nazwa	Znaczenie

15	NoToggleToOff (tylko przyciski radiowe)	Jeśli jest ustawiony, należy zawsze wybierać dokładnie jeden przycisk opcji; wybierając aktualnie wybrany przycisk nie ma wpływu. Jeśli wyczyścisz, kliknięcie wybranego przycisku powoduje jego odznaczenie, nie pozostawiając wybranego przycisku.
16	Radio	Jeśli jest ustawione, pole to zestaw przycisków opcji; jeśli wyczyszczone, pole jest polem wyboru. Ta flaga można ustawić tylko wtedy, gdy flaga przycisku jest pusta.
17	Przycisk	Jeśli jest ustawiony, pole jest przyciskiem, który nie zachowuje stałej wartości.
26	RadiosInUnison	Jeśli jest ustawiony, grupa przycisków opcji w polu przycisku opcji, które używają tej samej wartości ponieważ stan włączenia będzie się włączał i wyłączał zgodnie; to znaczy, jeśli ktoś jest zaznaczony, są wszystko sprawdzone. Jeśli wyczyszczone, przyciski wzajemnie się wykluczają (takie samo zachowanie jak przyciski radiowe HTML).

Pola tekstowe

Pola tekstowe to obszary na stronie, w których dowolny tekst / ciągi znaków mogą być wprowadzane przez użytkownika lub programowo przez oprogramowanie. Tekst jest zwykle rysowany tylko jedną czcionką, rozmiarem i kolorem wzdłuż jednej linii; jednak w słowniku polowym można podawać różne flagi i dodatkowe klucze, aby umożliwić objęcie tekstu wieloma wierszami (odpowiednio zawijanie) lub użycie formatowania tekstu sformatowanego lub obu tych elementów. Rysunek 7-2 pokazuje kilka przykładów



Zwykły tekst

Łańcuch tekstowy (lub strumień) reprezentujący wartość pola jest przechowywany jako wartość klucza V w słowniku pola. Podczas tworzenia wyglądu pola tekst jest zwykle rysowany w jednym stylu (czcionka, rozmiar, kolor itd.), Który jest zdefiniowany przez wartość klucza DA. Przykład 7-2 pokazuje kilka przykładów pliku tekstowych

Przykład 7-2. Proste pola tekstowe

```
16 0 obj
<<
/DA (/Helv 12 Tf 0 g)
/F 4
/FT Tx
/Rect [ 9.526760 680.078003 297.527008 702.078003 ]
/Subtype /Widget
/Type /Annot
/T (SimpleText)
/V (A single line of text in one style)
>>
endobj
17 0 obj
<<
/DA /Helv 12 Tf 0 g
/F 4
/FT Tx
/Ff 4096
/Rect [ 8.184650 550.416992 296.184998 653.090027 ]
/Subtype /Widget
/Type /Annot
/T (MultilineText)
/V (Multiple lines of text in a single style will automatically wrap
around from line to line as more text is added to a second paragraph)
>>
endobj
```

Tekst sformatowany

Jeśli wartość tekstowa pola powinna być wyświetlana w wielu stylach, wówczas w słowniku pola podawany jest ciąg tekstu sformatowanego jako wartość klucza RV i należy ustawić bit 26 flag pola tekstowego. Łańcuch tekstu sformatowanego to w pełni sformatowany dokument XML, który jest zgodny z gramatyką zdefiniowaną dla tekstu sformatowanego w specyfikacji XFA, która jest oparta na

określonych podzbiorach standardów XHTML 1.0 i CSS2. Element <body> jest katalogiem głównym dokumentu, który jest następnie dzielony na akapity (<p>) i zakresy (), które są stylizowane przy użyciu standardowych atrybutów CSS, takich jak rodzina czcionek i kolor. Przykład 7-3 pokazuje przykład pola tekstu sformatowanego.

Przykład 7-3. Pole tekstu sformatowanego

```
20 0 obj
<<
/F 4
/FT Tx
/Ff 33554432
/Rect [ 312.427002 680.749023 600.427002 702.749023 ]
/Subtype /Widget
/Type /Annot
/T (RichText)
/V (A single line of text in multiple styles)
/RV (<?xml version="1.0"?> \
<body xfa:APIVersion="Acroform:2.7.0.0" xfa:spec="2.1" \
xmlns="http://www.w3.org/1999/xhtml" \
xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"><p \
style="margin-top:0pt;margin-bottom:0pt;text-valign:top;font- \
family:Helvetica;font-size:13pt"><span>A </span><span \
style="font-weight:bold">single</span><span \
style="font-weight:normal"> </span><span \
style="font-weight:normal;text-decoration:underline">line</span>< \
span style="font-weight:normal;text-decoration:none"> \
</span><span \
style="color:#0000ff;font-weight:normal;text-decoration:none">of< \
/span><span \
style="color:#000000;font-weight:normal;text-decoration:none"> \
</span><span \
style="color:#000000;font-weight:normal;font-style:italic;text- \
decoration:none">text</span><span \
```

style="color:#000000;font-weight:normal;font-style:normal;text- \
decoration:none"> <span \
style="vertical-align:4.03pt;font-size:8.58pt;color:#000000;font- \
weight:normal;font-style:normal;text-decoration:none">in< \
span \
style="vertical-align:baseline;color:#000000;font-weight:normal; \
font-style:normal;text-decoration:none"> <span \
style="vertical-align:baseline;color:#00ff00;font-weight:normal; \
font-style:normal;text-decoration:none">multiple<span \
style="vertical-align:baseline;color:#000000;font-weight:normal; \
font-style:normal;text-decoration:none"> styles</p></body>)
>>
endobj
21 0 obj<<
/F 4
/FT Tx
/Ff 33558528
/Rect [311.756012 550.416992 599.755981 653.090027]
/Subtype /Widget
/Type /Annot
/T (MultilineRichText)
/V (Multiple lines of text, in multiple styles,\
automatically wrap around from line to line as more text is added. \
A second, centered, paragraph)
/RV (<?xml version="1.0"?>
<body xfa:APIVersion="Acroform:2.7.0.0" xfa:spec="2.1" \
xmlns="http://www.w3.org/1999/xhtml" \
xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"><p \
style="margin-top:0pt;margin-bottom:0pt;font-family:Helvetica;font- \
size:13pt">Multiple <span \
style="text-decoration:underline">lines<span \

style="text-decoration:none"> of <span \

style="font-style:italic;text-decoration:none">text,<span \

style="font-style:normal;text-decoration:none"> in <span \

style="color:#00ff00;font-style:normal;text-decoration:none">multiple \

styles<span \

style="color:#000000;font-style:normal;text-decoration:none">, that \

will automatically <span \

style="color:#000000;font-weight:bold;font-style:normal;text- \

decoration:none">wrap<span \

style="color:#000000;font-weight:normal;font-style:normal;text- \

decoration:none"> around from <span \

style="color:#000000;font-weight:normal;font-style:normal;text- \

decoration:line-through">line to line<span \

style="color:#000000;font-weight:normal;font-style:normal;text- \

decoration:none"> as <span \

style="color:#ff0000;font-weight:normal;font-style:normal;text- \

decoration:none">more text<span \

style="color:#000000;font-weight:normal;font-style:normal;text- \

decoration:none"> is added.</p><p \

style="margin-top:0pt;margin-bottom:0pt;font-family:Helvetica;font- \

size:13pt"> </p><p \

style="margin-top:0pt;margin-bottom:0pt;text-align:center;font-family \

:Helvetica;font-size:13pt">A second, centered, \

paragraph</p></body>)

>>

endobj

Flagi pól tekstowych

W ISO 32000-1, tabela 228 wymieniono wszystkie flagi pól, które można określić w polu tekstowym, ale najczęściej używane są następujące:

Bit 13

Flaga wielowierszowa jest ustawiana, gdy wartość tekstowa pola powinna być zawinięta na końcu adnotacji.

Bit 14

Flaga hasła określa, że wartość tego pola nie powinna być nigdy rysowana ani przechowywana w pliku PDF, lecz przechowywana tylko lokalnie do celów przesyłania danych.

Bit 26

Flaga tekstu sformatowanego określa, że wartość tego pola nie jest zwykłym ciągiem, ale w rzeczywistości jest tekstem o bogatej stylistyce.

Pola wyboru

Pole wyboru służy do przedstawienia użytkownikowi wielu opcji, z których może wybrać jedną lub więcej, aby stać się wartością pola.

Flaga MultiSelect

Zwykle pole wyboru pozwala użytkownikowi wybrać tylko jeden element z listy opcji / opcji. Są jednak chwile, w których wybór wielu przedmiotów może być przydatny. Aby to włączyć, ustawiana jest flaga wielokrotnego wyboru (bit 22) flag pól.

Opcje

Lista ciągów tekstowych, które będą wyświetlane w polu jako opcje, z których użytkownik może wybrać, są przechowywane w tablicy jako wartość klucza Opt w słowniku polowym. W większości przypadków tablica jest prostą tablicą samych ciągów tekstowych; jeśli jednak nazwa eksportu danego ciągu tekstowego musi być inna, wówczas ten element tablicy sam będzie tablicą dwóch elementów - ciąg tekstowy do wyświetlenia, po którym następuje nazwa eksportu. Przykład 7-4 ilustruje oba podejścia.

Przykład 7-4. Przykłady opcji

% Simple options

/Opt [(France) (Belgium) (Germany) (United Kingdom) (Spain)]

% Options with alternate export names

/Opt [(France) (Belgium) [(Germany) (DE)] [(United Kingdom) (UK)] (Spain)]

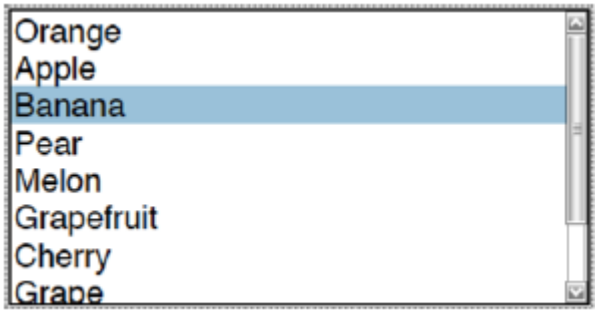
Wartości

Wartość, podobnie jak w przypadku innych typów pól, jest zapisywana jako wartość klucza V w słowniku pól. Jednak po ustawieniu flagi wielokrotnego wyboru wartość ma tablicę typu zamiast ciągu tekstowego. Wartość V może być również typu null, co jest wartością domyślną, co oznacza, że nic nie zostało wybrane z opcji. Lista wybranych pozycji zostanie zapisana jako wartość klawisza I w słowniku polowym. Wartość będzie się składać z listy liczb całkowitych reprezentujących (zerowe) indeksy wybranych pozycji. Zwykle jest to używane tylko w przypadku pól z ustawioną flagą wielokrotnego wyboru, ale można go również stosować w przypadku pól pojedynczego wyboru. Jeśli elementy wymienione w wartości I różnią się od wartości V w słowniku polowym, pozycja V będzie miała pierwszeństwo.

Przewijane listy

Domyślnym typem pola wyboru jest przewijana lista (Przykład 7-5), która wyświetla listę pozycji w przewijanej liście (jeśli jest ich zbyt wiele, aby pokazać je wszystkie naraz). Użytkownik może nie mieć nic zaznaczonego lub przynajmniej jednego wybranego elementu.

Przykład 7-5. Przewijana lista



16 0 obj

<<

/DV (Prune) % the default value if another is not specified

/F 4

/FT Ch

/I [2]

/Opt [(Orange) (Apple) (Banana) (Pear) (Melon) (Grapefruit) (Cherry)

(Grape) (Prune)]

/Rect [59.053501 606.116028 275.135986 716.169983]

/Subtype /Widget

/T (ListBox)

/Type /Annot

/V (Banana)

>>

endobj

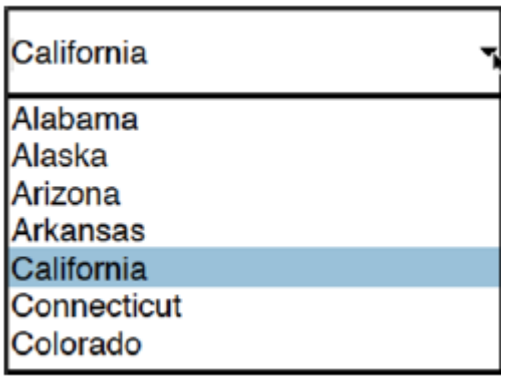
Pola kombi

Zamiast zajmować dużą część strony z pełną listą elementów, mogą one być przedstawione w mniejszym polu kombi, z którego użytkownicy mogą wybierać elementy, które chcą (Przykład 7-6). Jeśli ustawiony jest bit 18 flag pola, pole wyboru zostanie przedstawione jako pole kombi.

Pola kombi

Zamiast zajmować dużą część strony z pełną listą elementów, mogą one być przedstawione w mniejszym polu kombi, z którego użytkownicy mogą wybierać elementy, które chcą (Przykład 7-6). Jeśli ustawiony jest bit 18 flag pola, pole wyboru zostanie przedstawione jako pole kombi.

Przykład 7-6. Pole kombi



17 0 obj

<<

/DV (California)

/F 4

/FT Ch

/Ff 131072

/I [4]

/Opt [(Alabama) (Alaska) (Arizona) (Arkansas) (California) (Connecticut)

(Colorado)]

/Rect [341.571014 682.617004 526.112976 715.499023]

/Subtype /Widget

/T (Dropdown)

/Type /Annot

/V (California)

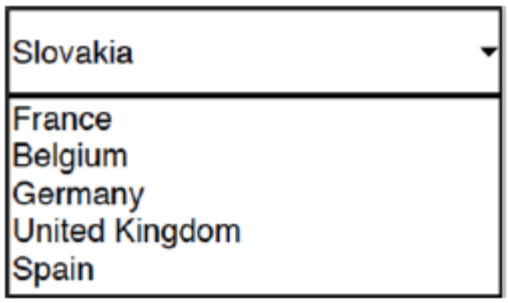
>>

endobj

Edytowalne pola kombi

Czasami nie jest możliwe uwzględnienie wszystkich możliwych opcji na liście i konieczne będzie wprowadzenie własnych wartości przez użytkowników. Ustawienie bitu 19 flag pola powoduje, że wartość pola kombi jest edytowalna.

Przykład 7-7. Edytowalne pole kombi



<<

/DV (Germany)

/F 4

/FT Ch

/Ff 393216

/Opt [(France) (Belgium) (Germany) (United Kingdom) (Spain)]

/Rect [342.242004 606.116028 526.783997 638.997986]

/Subtype /Widget

/T (DropdownE)

/Type /Annot

/V (Slovakia)

>>

Pola podpisu

Pole podpisu umożliwia identyfikację, czy użytkownik powinien zastosować podpis cyfrowy lub elektroniczny do pliku PDF. Jako takie, pole może być uważane za podpisane lub nie podpisane. Podpisane pole podpisu będzie zawierało klucz V w słowniku polowym.

Przed faktycznym podpisaniem pola przez użytkownika służy on jako symbol zastępczy na stronie, na której zostanie narysowana graficzna reprezentacja podpisu (jeśli jest obecny), wraz z wszelkimi informacjami na temat rodzaju technologii podpisu, które należy zastosować w danym momencie podpisywania. Pola podpisu, które nie mają być widoczne, będą miały prostokąt adnotacji o zerowej wysokości i szerokości lub z bitem Hid den lub bitem NoView klawisza F w ustawionych słownikach adnotacji.

Formularz akcji

Oprócz akcji nawigacyjnych istnieją trzy typy akcji specyficzne dla pól formularza: SubmitForm, ResetForm i ImportData. Podobnie jak akcje nawigacyjne Częściej jednak kojarzy się je z polem przycisku w formularzu.

Prześlij formularz

Akcja Prześlij formularz informuje przeglądarkę plików PDF o przesłaniu nazw i wartości określonych pól do określonego adresu URL w określonym formacie. Informacje te znajdują się w słowniku akcji, którego klucz S będzie miał wartość Prześlij formularz. Adres URL, do którego zostaną przesłane dane formularza, jest określony jako specyfikacja pliku URL, która jest wartością klawisza F w słowniku akcji. Pola, które mają być zawarte w przesyłanych danych, są określone w wartości tablicy klucza Fields. Każdy element tablicy może być albo pośrednim odniesieniem do słownika pól lub łańcuchem tekstowym reprezentującym w pełni kwalifikowaną nazwę pola. Elementy obu rodzajów można mieszać w tej samej tablicy. Klucz pola nie jest wymagany. Jeśli go nie ma, oznacza to, że wszystkie pola formularza zostaną przesłane.

Formaty przesyłania

PDF zawiera specyfikację czterech różnych formatów, w których dane formularza można przesłać na podany adres URL. Wybór formatu zależy od szeregu flag, które są określone jako wartość klucza Flags w słowniku akcji:

FDF

Forms Data Format (FDF) to podzbiór składni PDF, którego można użyć do przedstawienia danych formularza. Jest opisany w ISO 32000-1, 12.7.7 i jest domyślnym formatem przesyłania danych. Aby użyć tego formatu, upewnij się, że bity 3 (ExportFormat), 6 (XFDF) i 9 (SubmitPDF) są czyste.

HTML

Jest to ten sam format, którego używa HTML 4.01 do przesyłania danych formularza. Aby użyć tego formatu, ustaw bit 3 (ExportFormat) i upewnij się, że bity 6 (XFDF) i 9 (SubmitPDF) są czyste.

XFDF

Format XML Forms Data Format (XFDF) to wersja FDF oparta na XML. XFDF jest opisany w uwadze technicznej Adobe. Aby użyć tego formatu, ustaw bit 6 (XFDF) i upewnij się, że bity 3 (ExportFormat) i 9 (SubmitPDF) są czyste.

PDF

W tym formacie przesyłany jest cały dokument, a nie poszczególne pola i wartości. Aby użyć tego formatu, ustaw bit 9 (Prześlij PDF) i upewnij się, że bity 3 (ExportFormat) i 6 (XFDF) są czyste.

Przykład 7-8 pokazuje przykładową akcję SubmitForm.

Przykład 7-8. Akcja Prześlij formularz

<<

/ F 4

/ FT Btn

/ Ff 65536

/ Rect [358.347992 725.026001 430.347992 745.026001]

/ Podtyp / Widżet

/ T (Prześlij)

/ Type / Annot

/ A <<

/ S SubmitForm

/ Fields [(Address) (By) (Date) (Email) (Name) (TelNum) (Title)]

%, ponieważ nie ma klucza * Flagi *, co oznacza użycie domyślnego FDF

>>

>>

Zresetować formularz

Akcja ResetForm jest dość podobna do akcji Prześlij formularz, ponieważ działa na liście pól podanej w kluczu Pola słownika akcji. Jednak zamiast przeglądarki plików PDF przesyłającej dane określonych pól pod określony adres URL, wartość V) każdego pola jest resetowana do wartości jego klucza DV (wartość domyślna). Klawisz S w słowniku akcji będzie miał wartość ResetForm. Jeśli jest obecny, wartość klucza Flagi może wynosić tylko 0 lub 1. Wartość 0 (lub brak klucza) oznacza, że lista pól w polach to pola, które należy zresetować. Wartość 1 oznacza, że są to pola, których nie należy resetować (i resetować wszystkie inne). Przykład 7-9 ilustruje to drugie podejście.

Przykład 7-9. Działanie ResetForm

<<

/ A <<

/ Fields [(By_2) (Date_2) (Email_2) (Name_2) (TelNum_2) (Title_2)]

/ Flagi 1% resetują wszystkie ALE te ...

/ S ResetForm

>>

/ F 4

/ FT Btn

/ Ff 65536

/ Rect [447.501007 725.026001 519.500977 745.026001]

/ Podtyp / Widżet

/ T (Reset)

/ Type / Annot

>>

Zaimportować dane

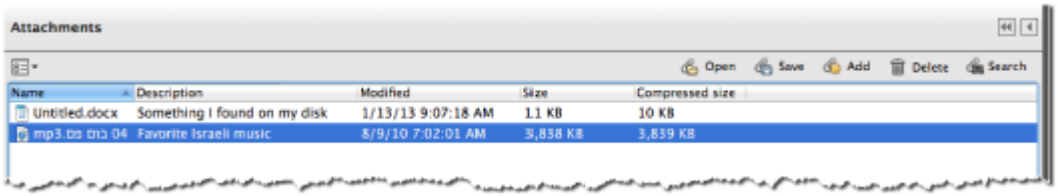
Akcja ImportData umożliwia import danych z określonego pliku FDF do pól pliku PDF. Słownik akcji ma dwa wymagane klucze: klawisz S (z wartością ImportData) i klawisz F, którego wartością jest słownik specyfikacji plików wskazujący, gdzie znajdują się dane FDF.

Pliki osadzone

W tej częci wyjaśniono, w jaki sposób plik PDF może służyć jako kontener na inne pliki, podobnie jak plik ZIP, a jednocześnie zapewnia bogatą zawartość strony, która im towarzyszy. W większości przypadków formaty plików (takie jak .docx lub .xslx) zostaną przekonwertowane na format PDF w celu dystrybucji. Czasami jednak przydatne może być również posiadanie oryginalnego pliku. Niestety istnieje duża szansa, że oba pliki zostaną rozłączone, więc tak się dzieje ,że przydatny jest sposób na osadzenie lub załączenie oryginału w pliku PDF. Ponadto możesz osadzić inne pliki związane z plikiem PDF, które nie są rzeczywistą zawartością, takie jak dane XML. Z tych i innych powodów PDF obsługuje możliwość osadzania innych plików wewnątrz siebie, a następnie wyświetlania ich w interfejsie przeglądarki PDF.

Specyfikacja pliku

Sercem osadzania plików jest słownik specyfikacji plików. Słownik faktycznie obsługuje zarówno pliki osadzone, jak i pliki referencyjne, ale skupimy się ściśle na formie osadzonej (patrz Rysunek 8-1). Aby zapewnić identyfikację słownika, musi on zawierać klucz Type, którego wartość to Filespec. Dodatkowo muszą być jeszcze trzy inne klucze obecne w słowniku: F, UF i EF (przykład: patrz Przykład 8-1). Klawisz F zawiera nazwę pliku w specjalnym kodowaniu specyficznym dla ciągów specyfikacji pliku (ISO 32000-1: 2008, 7.11.2), który jest „standardowym kodowaniem dla platformy, na której przeglądany jest dokument”. większość nowoczesnych systemów operacyjnych, to UTF-8, ale nie jest to wymagane. Jednak klucz UF zawiera nazwę zakodowaną jako standardowy 16-bitowy Unicode. Klucz EF odnosi się do słownika osadzonych plików, który jest prostym słownikiem z pojedynczym kluczem F, którego wartością jest osadzony strumień plików, w którym żyją rzeczywiste dane dla osadzonego pliku, wraz z dodatkowymi metadanymi o pliku. Można podać opcjonalny klucz opisu, którego wartością jest czytelny opis pliku.



Przykład 8-1. Przykładowe słowniki specyfikacji plików

% file specification for a file with a simple ASCII name

20 0 obj

<<

/F (Untitled.docx)

/UF (Untitled.docx)

/EF << /F 22 0 R >>

/Type /Filespec

/Desc (Something I found on my disk)

>>

endobj

% file specification for a file with a name requiring Unicode

31 0 obj

<<

/F (04 ... ...mp3)

/Type /Filespec

/Desc (Favorite Israeli music)

/EF << /F 32 0 R >>

/UF (þÿ.0.4. .Ñ.Õ.Ý. .ä.Ý...m.p.3) % 04 פם בום .mp3

>>

endobj

Osadzone strumienie plików

Strumień osadzonego pliku to po prostu obiekt strumienia, który zawiera dane dla osadzonego pliku. Jako taki, można go przechowywać i kompresować za pomocą filtrów, takich jak Flate - ta sama technologia, którą zastosowano w pliku ZIP. W słowniku osadzonego strumienia plików może znajdować się wiele dodatkowych informacji, takich jak typ multimediów internetowych pliku (inaczej typ MIME), jako wartość klucza podtypu. Inne informacje, takie jak data i godzina utworzenia lub ostatniej modyfikacji pliku, mogą zostać uwzględnione w słowniku parametrów osadzonego pliku (który jest wartością klucza Params). Przykład 8-2 pokazuje przykład osadzonego strumienia plików.

Przykład 8-2. Przykład osadzonego strumienia plików

32 0 obj

<<

/Subtype /audio/mpeg

/Filter /FlateDecode % compressed using Flate/ZIP technology

/Length 1000830 % encoded length

/Params <<

/ModDate (D:20100809110201)

/CheckSum <1E2AFAC553A11A00E20A02774BA42EBF>

/CreationDate (D:20130113152115-05'00')

/Size 3930112 % decoded length

>>

>>

stream

% Flate-compressed stream data goes here....

endstream

endobj

Wartość klucza CheckSum w słowniku parametrów osadzonego pliku to 16-bajtowy ciąg znaków, który jest sumą kontrolną bajtów nieskompresowanego osadzonego pliku, obliczoną przez zastosowanie standardowego algorytmu trawienia komunikatów MD5 do bajtów osadzonego strumienia plików . Specyfikacja pliku i związany z nim osadzony strumień plików to tylko jeden element układanki; nadal musi być połączony z czymś w strukturze PDF, aby można go było znaleźć w przeglądarce plików PDF. Jeśli plik jest powiązany z określoną treścią na określonej stronie, adnotacja FileAttachment byłaby odpowiednia. Jeśli jednak plik jest bardziej globalny w stosunku do dokumentu, drzewo nazw plików osadzonych byłoby właściwym miejscem

Specyfikacje pliku URL

Chociaż nie jest używany w przypadku plików osadzonych, istnieje specjalny typ specyfikacji pliku zwany adresem URL, który jest używany w innych częściach pliku PDF jako standardowy sposób określania, że strumień danych pliku powinien być pobierany z danego jednolitego lokalizatora zasobów (adres URL). Aby zadeklarować specyfikację pliku jako specyfikację pliku URL, klucz FS będzie miał wartość (typu Nazwa) URL (patrz Przykład 8-3). Ponadto wartość klucza F nie będzie ciągiem specyfikacji pliku, ale zamiast tego będzie adresem URL formy zdefiniowanej w RFC 1738, „Uniform Resource Locators”.

Ponieważ wymagania dotyczące kodowania znaków określone w RFC 1738 ograniczają adres URL do 7-bitowego ASCII w USA, który jest ścisłym podzbiorem PDFDocEncoding, wartość można również uznać za zawartą w tym kodowaniu.

Przykład 8-3. Przykładowa specyfikacja pliku URL

<<

/ FS / URL

/ F (http://www.adobe.com/devnet/acrobat/pdfs/PDF32000_2008.pdf)

>>

Sposoby osadzania plików

Pliki można połączyć z plikiem PDF na dwa sposoby, w zależności od tego, czy mają być powiązane z określoną zawartością w określonej lokalizacji, czy globalnie z plikiem PDF jako całością. W pierwszym przypadku użyjemy adnotacji załączników. W tym drugim przypadku podejście polega na dodaniu klucza EmbeddedFiles do słownika nazw dokumentu.

Adnotacje FileAttachment

Adnotacja załącznika pliku jest prostym typem adnotacji; jest podobny do adnotacji tekstowej, z tym wyjątkiem, że zamiast klucza zawartości z tekstem, który ma zostać wyświetlony, zawartość jest tekstem opisowym o pliku, takim jak nazwa pliku. Zawiera również klucz FS, który wskazuje na słownik specyfikacji dołączonego pliku, a jego klucz podtypu ma wartość FileAttachment.

Przykład 8-4 pokazuje wynik umieszczenia adnotacji załącznika do pliku, która określa ikonę spinacza obok tekstu w Hello World.pdf.

Przykład 8-4. Przykład pliku Adnotacja załącznika

% the annotation object/dictionary

41 0 obj

<<

/C [0.25 0.333328 1]

/Type /Annot

/Contents (world.jpg)

/Name /Paperclip

/Subtype /FileAttachment

/FS 42 0 R

/Rect [390.162 599.772 397.162 616.772]

>>

endobj

% the file specification dictionary

42 0 obj

<<

/F (world.jpg)

/Type /Filespec

/UF (world.jpg)

/EF << /F 43 0 R >>

>>

endobj

% and the embedded file stream

43 0 obj

<<

/Subtype /image/jpeg

/Length 25531

/DL 20172

/Params <<

/ModDate (D:20121020024106-04'00')

/CheckSum <19D579AB5B7C8F46B63C37F385707872>

/CreationDate (D:20121020024106-04'00')

/Size 20172

>>

>>

stream

% Stream data goes here...

endstream

endobj

Drzewo nazw EmbeddedFiles

Osadzone strumienie plików są powiązane z dokumentem jako całością poprzez dodanie do słownika nazw dokumentu klucza EmbeddedFiles, którego wartością jest drzewo nazw. To drzewo nazw odwzorowuje ciągi nazw na specyfikacje plików, które odnoszą się do osadzonych strumieni plików poprzez ich wpisy EF (patrz przykład 8-5).

Przykład 8-5. Przykładowe drzewo nazw EmbeddedFile

8 0 obj

<<

/Type /Catalog

/Names 16 0 R

/PageMode /UseAttachments

/Metadata 1 0 R % not included in the sample

/Pages 5 0 R % not included in the sample

>>

endobj

16 0 obj

<<

/EmbeddedFiles 17 0 R

>>

endobj

17 0 obj

<<

/Names [

(Some Embedded File) 21 0 R

(Untitled.docx) 20 0 R

]

>>

endobj

20 0 obj

<<

/F (Untitled.docx)

/UF (Untitled.docx)

/EF << /F 22 0 R >>

/Type /Filespec

/Desc (Something I found on my disk)

>>

endobj

21 0 obj

<<

/F (Some Embedded File)

/Type /Filespec

/Desc (Something else on my disk)

/EF << /F 32 0 R >>

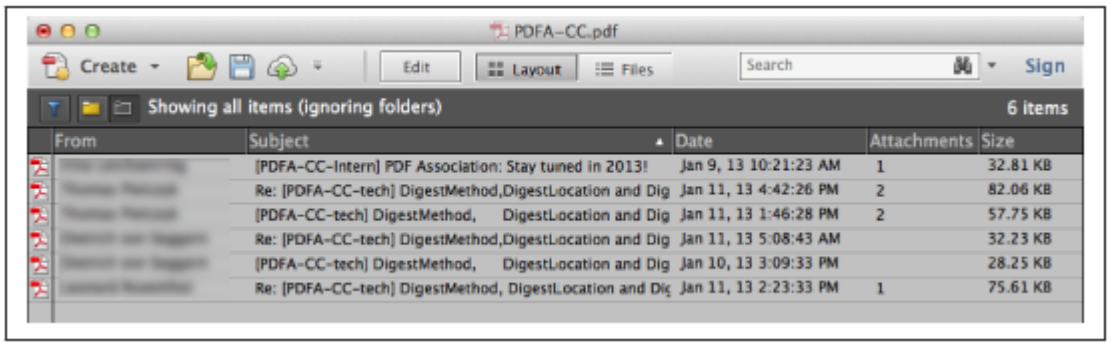
/UF (Some Embedded File)

>>

endobj

Kolekcje

Plik PDF z osadzonymi plikami jest przydatny, gdy treść strony jest głównym celem osoby, która będzie czytać dokument. Czasami jednak masz zbiór dokumentów, które należy zgrupować, ale żaden z nich nie ma wyższego priorytetu niż inny. Dlatego też osadzone pliki są w centrum uwagi. Na przykład mogą to być wszystkie materiały do sprawy sądowej lub licytacji na zlecenie inżynierskie. W takich przypadkach chcesz, aby przeglądarka plików PDF wyświetlała listę plików i powiązane z nimi metadane, a nie zwykły widok zawartości strony dokumentu głównego. W tym celu używana jest funkcja przenośnych kolekcji (lub po prostu „kolekcji”) pliku PDF. Rysunek 8-2 pokazuje przykład



Nie ma wymogu, aby dokumenty w kolekcji miały ukryty związek lub nawet podobieństwo; wykazanie różnicujących cech powiązanych dokumentów może być jednak pomocne w nawigacji dokumentów.

Słownik kolekcji

Zawartość kolekcji to pliki wymienione w EmbeddedFiles. Dowolny plik w drzewie nazw będzie częścią kolekcji, natomiast pliki osadzone, które nie znajdują się w drzewie, nie będą. Aby te pliki były kolekcją, a nie tylko luźnym zestawem plików osadzonych, w pliku PDF musi znajdować się słownik kolekcji o wartości Collec klucz słownika w słowniku dokumentu (prosty przykład znajduje się w Przykładzie 8-6). Chociaż żaden z kluczy w słowniku kolekcji nie jest wymagany, przydatny słownik kolekcji zawiera co najmniej dwa klucze: D i View.

D

Klucz D ma wartość ciągu, która jest nazwą pliku PDF w drzewie nazw EmbeddedFiles, którą przeglądarka PDF ma wyświetlać na początku. Zaleca się, aby był to kluczowy dokument w kolekcji lub instrukcje dotyczące poruszania się po kolekcji.

View

Klucz View ma wartość (typu nazwa), która powie przeglądarce PDF, czy ma wyświetlać listę plików z kolekcji w trybie szczegółów (D), trybie kafelkowym (T), czy początkowo ukrytym (H).

Przykład 8-6. Prosty słownik kolekcji

```
43 0 obj
<<
/Type /Catalog
/Collection 44 0 R
/Names 42 0 R % this would be a standard EmbeddedFiles name tree
/Pages 39 0 R % this would be a standard page dictionary
>>
endobj
44 0 obj
<<
/Type /Collection
```

/D (Index)

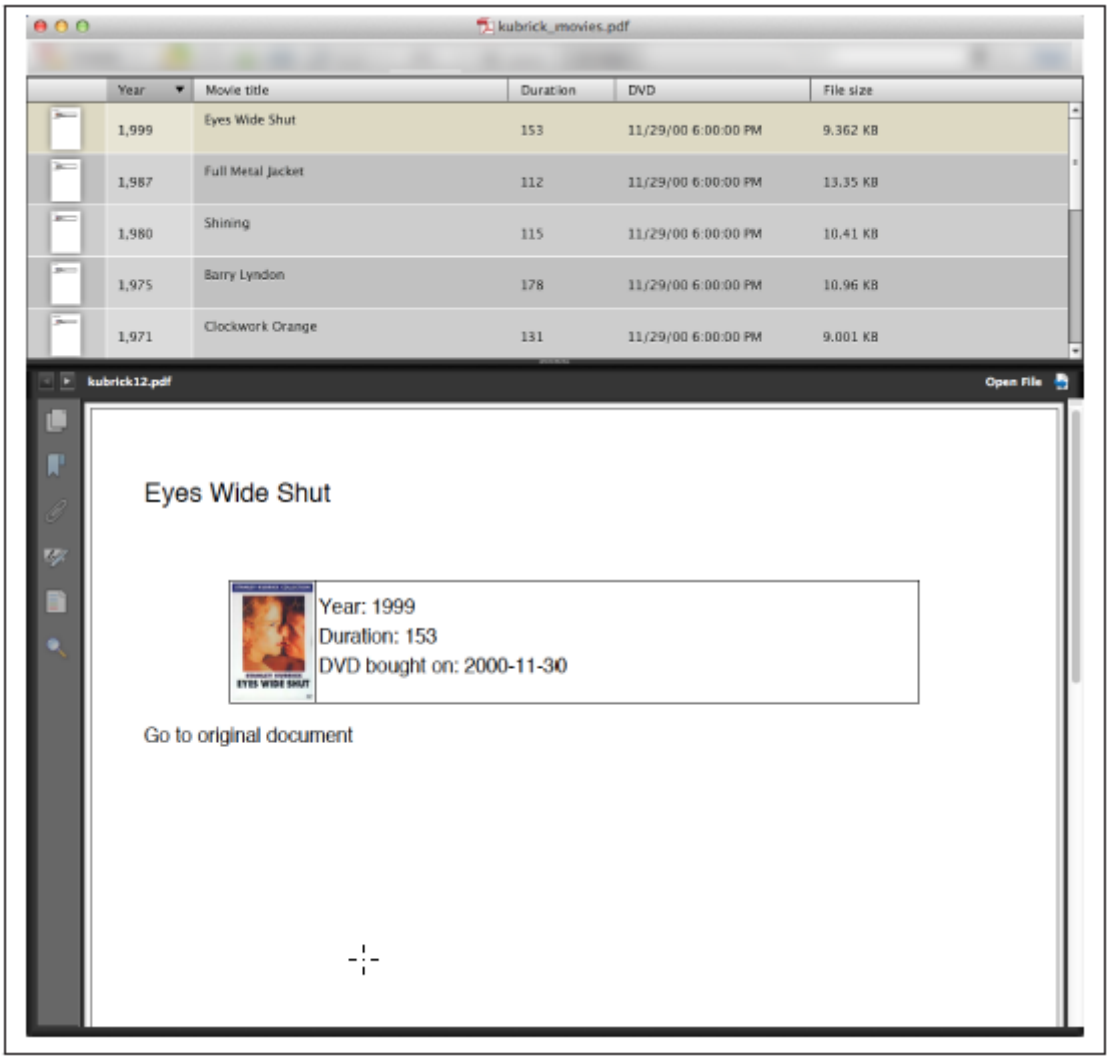
/View /D

>>

endobj

Schemat kolekcji

Chociaż prosta lista może być przydatna, bardziej prawdopodobne jest, że istnieją dodatkowe informacje o każdym pliku, które mogą być wyświetlane jako część interfejsu kolekcji prezentowanego przez przeglądarkę PDF. Na przykład, jeśli pliki reprezentowały katalog filmów, przydatne mogą być daty i czasy premiery filmów, jak na rysunku 8-3.



Aby utworzyć zestaw pól, takich jak te na przykładowym obrazie, słownik schematu kolekcji jest zawarty w słowniku kolekcji jako wartość klucza schematu, przy czym każdy klucz w słowniku ma wartość, która jest słownikiem pola kolekcji. Coś co by wyglądało jak w przykładzie 8–7.

Przykład 8–7. Przykładowy schemat kolekcji

<<

/Type /CollectionSchema

/YEAR <<

/Subtype /N % type of the data is a name

/N (Year)

/Type /CollectionField

/O 0

>>

/DURATION <<

/Subtype /N % type of the data is a name

/N (Duration)

/Type /CollectionField

/O 2

>>

/TITLE <<

/Subtype /S % type of the data is a string

/N (Movie title)

/Type /CollectionField

/O 1

>>

/DVD <<

/Subtype /D % type of the data is a date

/N (DVD)

/Type /CollectionField

/O 3

>>

>>

W schemacie przykładowym są cztery pola - ROK, CZAS TRWANIA, TYTUŁ i DVD - reprezentujące nie tylko nazwy pól, ale także ich typy. Pola te zostaną następnie skojarzone z każdym plikiem określonym w drzewie nazw EmbeddedFiles poprzez dodanie klucza CI w każdym słowniku specyfikacji plików. W naszym przykładzie nazwy są pisane wielkimi literami, ale nie jest to w żaden sposób wymagane. Użycie wszystkich wielkich liter gwarantuje, że wartości będą unikatowymi nazwami w pliku PDF. Przykład 8-8 pokazuje przykładową specyfikację pliku.

Przykład 8–8. Specyfikacja pliku z powiązanym słownikiem elementów kolekcji

<<

/F (kubrick12.pdf)

/CI

<<

/Type /CollectionItem

/YEAR 1999

/DURATION 153

/TITLE (Eyes Wide Shut)

/DVD (D:20001130000000+01'00')

>>

/EF

<<

/F 35 0 R

/UF 35 0 R

>>

/UF (kubrick12.pdf)

/Type /Filespec

/Desc (Eyes Wide Shut)

>>

Mając do dyspozycji wszystkie te dane, możemy również sortować listę plików na podstawie dowolnego elementu schematu zamiast domyślnej kolejności drzewa nazw plików osadzonych. Odbywa się to poprzez włączenie klucza sortowania do słownika kolekcji, którego wartością jest powiązany słownik sortowania kolekcji, jak pokazano w przykładzie 8-9.

Przykład 8-9. Przykładowy słownik sortowania kolekcji

% Sort the collection based on the YEAR field, in descending order

<<

/Type /CollectionSort

/S /YEAR

/A false

>>

Działania GoToE

Wcześniej dowiedziałeś się o działaniach, które pozwoliły użytkownikowi nawigować w obrębie istniejącego dokumentu (GoTo) lub do dokumentu zewnętrznego (GoToR). Teraz, gdy już wiesz, jak osadzać dokumenty w pliku PDF, zobaczmy, jak nawigować do osadzonego dokumentu. Akcja GoToE (lub „embedded go-to") jest dość podobna do zdalnej akcji go-to, ale umożliwia przeskakiwanie do osadzonego pliku PDF. Obsługiwane są zarówno adnotacje do załączników, jak i wpisy w drzewie nazw EmbeddedFiles. Te osadzone pliki mogą z kolei zawierać osadzone pliki, a działanie GoToE może wskazywać jeden lub więcej nadrzędnych plików PDF do końcowego docelowego pliku PDF (zwanego również docelowym plikiem PDF) za pośrednictwem słownika docelowego. Słownik akcji dla akcji GoToE będzie składał się z tych samych trzech kluczy, które znajdują się zarówno w akcji GoTo, jak i GoToR - Typ (z wartością Action), S (z wartością GoToE) i D (którego wartość jest miejscem docelowym w docelowy plik PDF). Wartość klawisza T w słowniku akcji jest słownikiem docelowym, który lokalizuje cel w stosunku do źródła, podobnie jak ścieżka względna opisuje fizyczną relację między dwoma plikami w systemie plików. Słowniki docelowe mogą być zagnieżdżane rekurencyjnie, aby określić jeden lub więcej celów pośrednich przed osiągnięciem ostatecznego. „Ścieżka względna" opisana w słowniku docelowym musi nie tylko schodzić w dół hierarchii, ale może również iść w górę, tak jak wpis „.." oznaczałby w ścieżce DOS lub Unix. „Kierunek" jest określony kluczem R (relacja) i ma wartość P (rodzic) lub C (dziecko). Przykład 8-10 pokazuje kilka przykładowych działań GoToE.

Przykład 8-10. Przykładowe działania GoToE

```
% Simple target of just a single embedded file
1 0 obj
<<
/Type /Action
/S /GoToE
/D [ 0 /FitH 794 ]
/T <<
/N (Our First PDF.pdf)
/R /C
>>
>>
% Target that navigates up and then back down into another PDF
1 0 obj
<<
/Type /Action
/S /GoToE
/D [ 0 /FitW 612 ]
/T <<
/R /P % navigate up to the parent
```

```
/T <<
/R /C % now down to one of its children
/N (Target.pdf) % named Target.pdf
>>
>>
>>
% Target that navigates up twice and then back down twice
1 0 obj
<<
/Type /Action
/S /GoToE
/D [ 0 /Fit ]
/T <<
/R /P % navigate up to the parent
/T <<
/R /P % and up again
/T <<
/R /C % now down to one of its children
/N (Intermediate.pdf) % named Intermediate.pdf
/T <<
/R /C % and one of its children
/N (Final.pdf) % named Final.pdf
>>
>>
>>
>>
>>
```

Multimedia i 3D

Przez 20 lat istnienia PDF świat multimediów przeniósł się z prostych dźwięków i animacji na dzisiejsze interaktywne doświadczenia w 2D i 3D. PDF obsługuje różne sposoby włączenia tych różnych rodzajów mediów. Ta część szczegółowo omówi serię typów adnotacji, które umożliwiają włączenie multimediów i treści 3D w formacie PDF.

Proste media

PDF 1.2 wprowadził typy adnotacji dźwiękowych i filmowych, które przeniosły PDF poza pierwotną wizję „statycznego elektronicznego papieru 2D" do pełnoprawnego formatu bogatego dokumentu, jakim jest obecnie.

Adnotacje dźwiękowe

Adnotacja dźwiękowa została pierwotnie dodana do pliku PDF, aby zapewnić analogię do adnotacji tekstowej, z tym wyjątkiem, że zamiast notatki tekstowej zawierałby dźwięk nagrany z mikrofonu komputera lub zaimportowany z pliku, który byłby odtwarzany po aktywacji adnotacji. Słownik adnotacji dla adnotacji dźwiękowej składa się z Podtypu dźwięku, strumienia danych dźwiękowych jako wartości klawisza Dźwięk, a także wszelkich wymaganych wspólnych informacji adnotacyjnych. Dodatkowo może być obecny klucz nazwy, którego wartość to Głośnik lub Mikrofon; deklaruje to predefiniowaną ikonę do użycia, gdy nie ma strumienia wyglądu. Strumieniowe dane dla dźwięku powinny mieć wspólny, samoopisujący format, taki jak AIFF, RIFF / wav lub snd / au, a częstotliwość próbkowania danych musi być uwzględniona (jako wartość klucza R) w słownik strumieniowy. Dodatkowe informacje o danych dźwiękowych, takie jak liczba kanałów lub bitów na wartość próbki na kanał, mogą być również zawarte w kluczach i wartościach w słowniku strumieniowym. Chociaż najczęściej osadza się dźwięki, ponieważ są one zwykle małe, nie jest to wymagane. Zamiast tego można użyć słownika specyfikacji plików do pliku zewnętrznego

Przykład 9-1. Przykładowa adnotacja dźwiękowa

1 0 obj

<<

/C [0 1 0]

/Contents (Presentation about nothing)

/F 28

/M D:20010213120212-05'00'

/Name (Speaker)

/Rect [22 529 42 549]

/Sound 2 0 R

/Subtype /Sound

/T (Leonard Rosenthol)

/Type /Annot

>>

endobj

2 0 obj

<<

/Type /Sound

/Length 1000 % or whatever the real length is

/Filter /FlateDecode % compression is good

/R 11025 % sampling rate

>>

stream

% binary stream data of the sound would go here...

endstream

endobj

Ze względu na różne ograniczenia w adnotacji dźwiękowej uważa się ją za przestarzałą i chociaż przeglądarki PDF nadal ją obsługują, nie zaleca się już używania tego typu adnotacji w przypadku dźwięków. Zamiast tego rozważ użycie adnotacji ekranowej

Akcje dźwiękowe

Dźwięk nie zawsze musi być powiązany z adnotacją. Czasami dźwięk może być odtwarzany w ramach interakcji użytkownika z plikiem PDF. W tym przypadku przewidziano akcję dźwiękową. Podobnie jak w przypadku innych akcji, klawisz S zapewnia typ akcji Dźwięk, w tym przypadku. Drugim wymaganym kluczem w słowniku akcji, podobnie jak w przypadku adnotacji dźwiękowej, jest Dźwięk, którego wartością jest ten sam strumień danych i powiązany słownik jak w adnotacji. Możliwe jest również określenie, czy dźwięk ma być odtwarzany synchronicznie, czy asynchronicznie (Synchroniczny) i czy powtórzyć (Powtórz). Przykład akcji dźwiękowej pokazano w przykładzie 9-2.

Przykład 9-2. Przykładowa akcja dźwiękowa

1 0 obj

<<

/S Sound

/Sound 2 0 R

>>

endobj

2 0 obj

<<

/B 16

/C 2

/E /Signed

/Filter /FlateDecode

/Length 1281270

/R 44100

/Type /Sound

>>

stream

% lots of stream data

endstream

endobj

Adnotacje filmowe

Adnotacja do filmu umożliwia odtwarzanie popularnych formatów wideo lub animacji, które mogą również obejmować dźwięk lub dźwięk. Obsługiwane formaty są niezdefiniowane przez PDF, a zatem wybór należy do przeglądarki. Aby zdefiniować słownik adnotacji dla adnotacji filmowej, wymagane są tylko dwa klucze: klucz podtypu o wartości Film i film, którego wartością jest słownik filmowy. Dodatkowo może być obecny klucz A, którego wartością jest słownik aktywacji filmu. Słownik filmowy Słownik filmowy definiuje rzeczywisty film do odtworzenia za pomocą słownika specyfikacji plików, umożliwiając tym samym osadzanie danych lub odwoływanie się do nich z zewnątrz. Ponadto współczynnik proporcji filmu można podać, gdy Rect adnotacji został przeskalowany, ale chcesz, aby film był odtwarzany z określonym współczynnikiem proporcji. Możliwe jest również dostarczenie predefiniowanego plakatu do filmu, który jest standardowym obrazem XObject wyświetlanym na stronie.

Słownik aktywacji filmu

Ten słownik zawiera przeglądarkę plików PDF z pewnymi informacjami na temat wizualnej prezentacji filmu. Aby odtworzyć film we własnym pływającym oknie zamiast bezpośrednio na stronie, podaj klucz FWScale wraz z tablicą współczynników skalowania (takich jak [1 1] dla skali 100%). Aby wyświetlić elementy sterujące dla użytkownika w celu dostosowania odtwarzania filmu, podaj klucz ShowControls o wartości true. Innym typem informacji, które można podać w tym słowniku, jest to, jak to zrobić aby odtworzyć wideo (lub przynajmniej rozpocząć odtwarzanie), w tym szybkość i głośność oraz czy odtworzyć go tylko raz, czy powtórzyć (Tryb). Przykład adnotacji do filmu pokazano na rysunku 9-1.

1 0 obj

<<

/A << /ShowControls true >>

/Movie <<

/Aspect [308 210]

/F <<

/F (SampleMovie.mov) % simple relative file path

/Type FileSpec

>>

/Poster 2 0 R

>>

/Border [0 0 1]

/C [1 1 1]

/F 1

/Rect [95.062149 496.936981 258.025818 608.048584]

/Subtype /Movie

/T (iPod Support)

/Type /Annot

>>

endobj

Ze względu na różne ograniczenia w adnotacji filmowej uważa się ją za przestarzałą. Chociaż przeglądarki PDF będą go nadal obsługiwać, nie zaleca się już używania tego typu adnotacji w przypadku filmów. Zamiast tego rozważ użycie adnotacji ekranowej.

Akcje filmowe

Podobnie jak w przypadku dźwięków, możliwe jest również wywołanie odtwarzania filmu za pomocą akcji. Operacja jest połączona z adnotacją filmu w tym samym pliku PDF, albo przez pośrednie odniesienie poprzez wartość klucza adnotacji, albo przez nazwę za pomocą wartości klawisza T. Ta akcja nie tylko pozwala na odtwarzanie filmu, ale może także określać inne operacje, takie jak Zatrzymaj lub Pauza, za pomocą klawisza Operacji. Przykład 9-3 pokazuje przykładową akcję filmową.

Przykład 9-3. Przykładowa akcja filmu

```
1 0 obj
<<
/S Movie
/T sample_iTunes.mov % identify by name
/Operation /Play % not necessary, but here for example
>>
endobj
```

Multimedia

W wersji PDF 1.5 obsługa multimediów w formacie PDF została objęta jednym nowym typem adnotacji - adnotacją na ekranie. Nazywa się to screen, ponieważ jego zadaniem jest określenie regionu strony (która będzie wyświetlana na ekranie), na którym będzie odtwarzany klip multimedialny. W rzeczywistości, tak naprawdę nie ma to nic wspólnego z multimediami; wszystkie rzeczy związane z mediami odbywają się poprzez akcję Rendition

Adnotacja na ekranie

Chociaż adnotacja na ekranie może być tak prosta, jak tylko klucz podtypu o wartości Screen, nie byłaby bardzo przydatna. Najważniejszą częścią jest wartość klucza A lub AA, w której określono działanie Rendition. Ponadto adnotacja zwykle ma klucz MK, którego wartością jest słownik cech wyglądu, który opisuje, jak będzie wyglądać adnotacja (renderowana bezpośrednio lub używana do tworzenia strumienia wyglądu). Przykład 9-4 pokazuje przykładową adnotację na ekranie.

Przykład 9-4. Przykładowa adnotacja na ekranie

```
1 0 obj
<<
/A 2 0 R % the Rendition action
/BS <<
/S /S
/Type /Border
/W 1
>>
```

/MK <<

/BC [0 0 1]

>>

/F 6

/Rect [498.316437 702.674866 549.301331 735.843201]

/Subtype /Screen

/T (A Movie)

/Type Annot

>>

endobj

Słownik cech wyglądu

Słownik cech wyglądu jest używany przez adnotacje ekranowe (a także AcroForms) do opisania ich wyglądu. Wartości można użyć do bezpośredniego renderowania wyglądu, ale częściej służą one do określenia grafiki, która ma być obecna w strumieniu wyglądu adnotacji. Niektóre klucze, które mogą występować w tym słowniku:

R

Wielokrotność 90, która reprezentuje liczbę stopni obrotu adnotacji

BC

Kolor, który ma być użyty na granicy adnotacji, opisany przy użyciu tego samego formatu tablicy co kolor tekstu

BG

Kolor tła adnotacji opisany przy użyciu tego samego formatu tablicy co kolor tekstu

Akcje renderowania

Działanie Rendition kontroluje odtwarzanie treści multimedialnych, bezpośrednio lub za pomocą JavaScript. Jest to zawsze związane z adnotacją na ekranie; w rzeczywistości jednym z wymaganych kluczy w słowniku operacji renderowania jest klucz AN, którego wartość jest pośrednim odniesieniem do takiej adnotacji. Pozostałe wymagane klucze to typ akcji (S), którego wartość to Rendition; operacja (OP) do wykonania (odtwarzanie, zatrzymanie itp.); oraz obiekt interpretacji jako wartość klucza R. Zamiast OP można użyć klucza JS z wartością JavaScript, która ma zostać wykonana po uruchomieniu akcji. Przykład 9-5 pokazuje przykładową akcję Rendition.

Przykład 9-5. Przykład akcji renderowania

1 0 obj

<<

/ S Rendition

/ OP 0

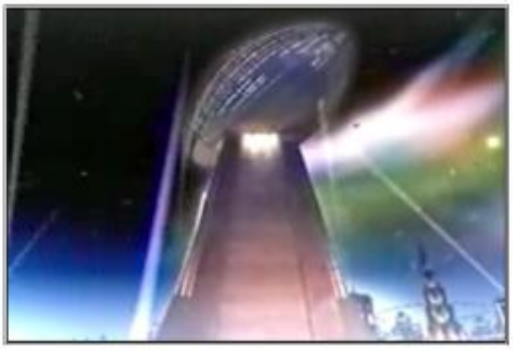
/ R 2 0 R% odwołanie do obiektu renderowania

>>

endobj

Obiekty renderowania

Podstawowy typ tłumaczenia nazywany jest przekazem medialnym; określa, w co grać, jak grać i gdzie grać. Możliwe jest również utworzenie uporządkowanej listy wersji multimediów (zwanej wersją wyboru), aby zapewnić przeglądarce plików PDF opcje dotyczące obsługi multimediów



o typ wersji. W przypadku słownika odtwarzania multimediów konieczne jest podanie informacji o tym, co grać za pomocą (klawisz C), jak go odtwarzać (klawisz P) i gdzie go odtwarzać (klawisz SP). Wartość klawisza C to słownik klipów multimedialnych, który opisuje, co ma być odtwarzane. Słownik klipów multimedialnych może określać, że są to pełne dane dla klipu lub tylko jego fragmentu, ale tutaj będziemy patrzeć tylko na pełne dane, ponieważ jest to zdecydowanie najczęstszy przypadek. Dane mogą być przechowywane w pliku (jako wartość klucza D) za pomocą prostego strumienia, lub specyfikacja pliku może być wykorzystana do odwołania do pliku osadzonego lub zewnętrznego. Typ danych jest zadeklarowany przy użyciu standardowych typów MIME jako wartości klucza CT. Wartość klawisza P to słownik parametrów odtwarzania multimediów, który określa sposób odtwarzania multimediów. Chociaż może określać konkretny odtwarzacz multimedialny, który ma być używany (choć nie jest to zalecane, ponieważ ogranicza on możliwości przeglądarki plików PDF do zastąpienia), jest częściej używany do opisania, czy zapewnić interfejs użytkownika do sterowania wideo (C), czy skalować wideo podczas odtwarzania (F) i ile razy (jeśli w ogóle), aby powtórzyć odtwarzanie wideo (RC). Pozostałym kluczowym elementem słownika interpretacji jest klucz SP, którego wartością jest słownik parametrów ekranu multimediów, który opisuje, czy odtwarzać multimedia na stronie, czy w ruchomym oknie (W), jak pokazano na rysunku 9-3, oraz, jeśli używasz pływającego okna, które monitoruje (lub monitoruje), można lub nie można w nim odtwarzać na (M) oraz przy jakim rozmiarze i lokalizacji należy grać (F). Przykład 9-6 pokazuje przykładowy obiekt renderowania.

Przykład 9–6. Przykładowy obiekt do interpretacji

```
2 0 obj
<<
/S MR
/C <<
/S /MCD
/CT (video/mpeg)
/D <<
/F (http://www.steppublishers.com/sites/default/files/step.mov)
/FS /URL
/Type /Filespec
>>
>>
/P <<
/BE <<
/C true
/F 2
/RC 1
>>
>>
/SP <<
/BE <<
/W 0 % use a floating window
```

/B [0.50 0 0] % background color for the floating window

/F <<

/D [352 288] % width and height of the window

/R 0 % user cannot resize it

/T false % no title bar

>>

>>

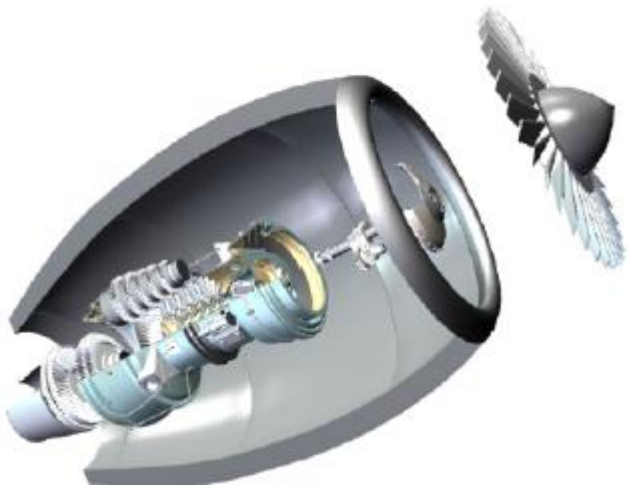
>>

>>

endobj

3D

Możliwość dodania wideo lub audio do pliku PDF przenosi go poza statyczny dokument elektroniczny do bardziej interaktywnego i bogatego w treść. Pozostaje jednak w sferze dwóch wymiarów. Dzięki adnotacjom 3D plik PDF może przejść do trzeciego wymiaru dzięki obecnej zawartości, którą można obracać i manipulować wzdłuż wszystkich trzech osi



Adnotacje 3D

Kompozycje 3D mogą być prezentowane czytelnikowi za pomocą adnotacji 3D i jej słownika adnotacji 3D. Dodatkowo adnotacja 3D zapewnia strumień wyglądu, który ma normalny wygląd (N) dla aplikacji, które nie obsługują adnotacji 3D oprócz reprezentowania początkowego wyświetlania grafiki 3D.

Słownik adnotacji 3D

Słownik adnotacji 3D to standardowy słownik adnotacji, którego podtypem jest 3D. Ponadto musi zawierać klucz 3DD, którego wartością jest strumień 3D zawierający dane 3D, a także wszelkie dodatkowe informacje, takie jak widoki 3D. Wartością 3DD może być również słownik referencyjny 3D. Ta opcja jest używana tylko w rzadkim przypadku, gdy w dokumencie jest wiele adnotacji, które muszą wyświetlać te same dane 3D.

Przykład 9–7. Przykładowa adnotacja 3D

1 0 obj

<<

/3DD 2 0 R

/AP << /N 3 0 R >>

/Contents (A 3D Model)

/Rect [289.174988 99.371803 764.690002 493.700989]

/Subtype /3D

/Type /Annot

>>

endobj

Widoki 3D

Kiedy czytnik prezentuje dane w strumieniu 3D w sposób widoczny dla człowieka, wykorzystuje szereg parametrów zastosowanych do wirtualnej kamery w celu ich renderowania. Ta seria parametrów, w tym orientacja i pozycja kamery oraz opis tła, nazywa się widokiem 3D (lub po prostu widokiem). Widoki te mogą również określać sposób renderowania, kolorowania, oświetlania i przekrojów kompozycji 3D. Widok może nawet zawierać listę węzłów (trójwymiarowych obszarów) grafiki 3D, aby uczynić ją niewidoczną. Zwykle widoki te są dynamiczne, tworzone po prostu przez użytkownika interakcyjnie manipulującego różnymi parametrami, takimi jak swobodny obrót i tłumaczenie. Możliwe jest jednak również skojarzenie zestawu predefiniowanych widoków z kompozycją 3D. Na przykład rysunek mechaniczny części może mieć określone widoki pokazujące górę, dół, lewo, prawo, przód i tył obiektu

Różne parametry widoku 3D są zachowane w słowniku widoku 3D. Jedynym kluczem wymaganym w słowniku jest XN, czyli nazwa widoku, który może zostać przedstawiony użytkownikowi. Spośród różnych opcjonalnych parametrów, które można ustawić, najczęstszą parą są klucze MS i C2W. Wartość klucza C2W to 12-elementowa macierz transformacji 3D, która określa pozycję i orientację kamery we współrzędnych światowych, podczas gdy klawisz MS zawiera wartość M, która instruuje czytelnika, aby używał wartości C2W. Przykład słownika widoku 3D pokazano w przykładzie 9-8.

Przykład 9–8. Przykładowy słownik widoków

<<

/C2W [1.0 0.0 0.0 0.0 0.0 -1.0 0.0 1.0 0.0 0.000006 -387.131989 -0.099388]

/MS /M

/Type /3DView

/XN (Default)

>>

Strumienie 3D

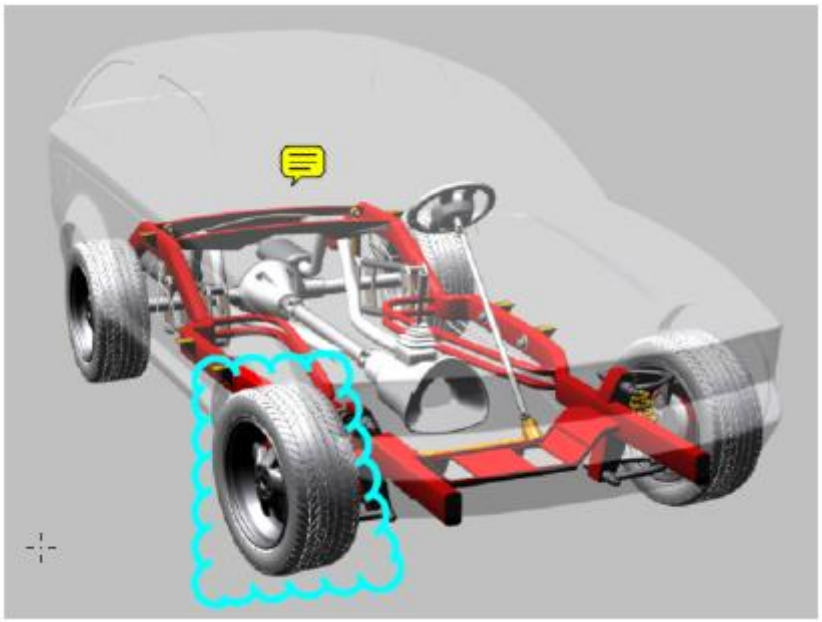
Strumień 3D to strumień, którego zawartość jest w formacie U3D lub PRC. Ponadto jego powiązany słownik musi mieć klucz podtypu, którego wartość deklaruje format danych (U3D lub PRC). Słownik strumienia 3D może również zawierać tablicę widoków 3D związanych z grafiką 3D jako wartość klucza VA. Klucz DV służy do określenia, który widok jest widokiem domyślnym lub początkowym, albo jako indeks liczb całkowitych w tablicy VA, albo jako ciąg pasujący do jednej z nazw dostarczonych widoków. Przykład 9-9 pokazuje przykład strumienia 3D z widokami.

Przykład 9–9. Przykładowy strumień 3D z widokami

```
1 0 obj
<<
/Type /3D
/Subtype /U3D
/VA [ 2 0 R 3 0 R 4 0 R ]
/DV 0 % the first one is the default
>>
stream
% U3D data goes here...
endstream
endobj
2 0 obj
<<
/BG << /C [ 0.752945 0.752945 0.752945 ] /Subtype /SC >>
/C2W [
-0.399527 -0.916721 0.0 -0.238227
0.103825 0.965644 -0.885226 0.385801
-0.259869 758.682983 -202.897003 207.556000
]
/CO 727.596008
/MS M
/Type /3DView
/XN (Default)
>>
```

Znaczniki na 3D

Różne adnotacje znaczników, które zostały wcześniej wprowadzone dla normalnej zawartości strony PDF, można również zastosować do określonych widoków grafiki 3D. Aby określić, że dana adnotacja znaczników jest powiązana z adnotacją 3D, do standardowego słownika adnotacji dodawany jest klucz ExData, którego wartością jest słownik znaczników 3D. Słownik znaczników 3D określa adnotację 3D i widok 3D, z którym skojarzony jest znacznik, i może także zawierać skrót MD5 danych 3D, aby umożliwić widzowi upewnienie się, że grafika 3D nie zmieniła się od momentu zastosowania adnotacji. Rysunek 9-6 pokazuje przykładowe znaczniki 3D.



% Polygon annotation

1 0 obj

<<

/BE << /I 2.0 /S C >>

/BS << /W 3.0 >>

/C [0.0 1.0 1.0]

/ExData 2 0 R

/Rect [302.412994 403.898987 399.747986 520.927979]

/Subtype /Polygon

/Type /Annot

/Vertices [

315.097992 506.010010 364.582001 508.194000

387.140991 423.052002 315.097992 416.502014

315.097992 506.010010]

>>

% 3D markup dictionary

2 0 obj

<<

/Type /ExData

/Subtype /Markup3D

/3DA 2 0 R % this is the 3D annotation

/3DV 3 0 R % and this is the view

>>

Treść opcjonalna

Opcjonalna zawartość to funkcja PDF, która umożliwia wyświetlanie określonych obiektów graficznych i / lub adnotacji tylko wtedy, gdy spełniony jest określony zestaw kryteriów. Kryteria te może określić autor treści - na przykład, aby ta treść pojawiła się tylko na ekranie i nigdy nie drukowała - lub może być określona przez użytkownika poprzez interakcję z widzem. Ta funkcja jest przydatna do różnych rzeczy, od rysunków CAD, map po dokumenty wielojęzyczne i wiele innych.

Opcjonalne grupy treści

Podstawowym elementem składowym do definiowania opcjonalnej treści jest opcjonalna grupa treści (OCG), która jest słownikiem składającym się z wymaganego Typu (którym zawsze jest OCG), wymaganej Nazwy grupy (która może być wyświetlana przez przeglądającego), oraz klucz Użycie, który deklaruje sposób użycia grupy.

Stan treści

Grupa ma przypisany stan, który jest WŁĄCZONY lub WYŁĄCZONY. Stany mogą być ustawiane automatycznie przez przeglądarkę (na podstawie Użycie), programowo lub przez interfejs użytkownika przeglądarki. Treść należąca do grupy jest widoczna, gdy grupa jest WŁĄCZONA, a niewidoczna, gdy jest WYŁĄCZONA. Treści mogą należeć do wielu grup, które mogą mieć sprzeczne stany.

Stosowanie

Treść jest zwykle pogrupowana, ponieważ ma pewną wspólną cechę. Może to być język treści lub może być używany tylko na ekranie. Opcjonalny słownik użytkowania zawartości, który jest wartością klucza użycia w opcjonalnym słowniku grupy treści, deklaruje podobieństwo. Te informacje o użyciu są następnie wykorzystywane przez przeglądarkę plików PDF w celu ustalenia, czy oceniać stan grupy na podstawie czynników zewnętrznych w połączeniu z wartością klucza AS w opcjonalnym słowniku konfiguracji treści. Słownik użycia może zawierać dowolną liczbę kluczy, jeśli grupa ma wiele wspólnych cech. Najczęściej używane klucze to:

Eksport

Wartością tego klucza jest słownik zawierający pojedynczy klucz ExportState, którego wartość (typu Name) jest WŁĄCZONA lub WYŁĄCZONA, deklarując, czy przeglądający powinien wyeksportować tę treść do formatu nieopcjonalnego rozpoznającego treść (takiego jak raster obrazy lub Postscript).

Drukowanie

Wartością klucza jest słownik zawierający pojedynczy klucz PrintState, którego wartość (typu Nazwa) jest WŁĄCZONA lub WYŁĄCZONA, deklarując, czy treść ta powinna zostać wydrukowana.

Widok

Wartością tego klucza jest słownik zawierający pojedynczy klucz ViewState, którego wartość (typu Nazwa) jest WŁĄCZONA lub WYŁĄCZONA, deklarując, czy treść ta powinna być wyświetlana na ekranie (lub jakikolwiek domyślny „widok” przeglądarki).

Powiększenie

Słownikowa wartość tego klucza zawiera albo klucz min, klucz maks, albo oba. Wartości tych kluczy, jeśli występują, określają minimalne i maksymalne powiększenie / powiększenie (w procentach), przy których grupa powinna być uznana za WŁĄCZONĄ.

Język

Wartością tego klucza jest słownik, który deklaruje naturalny język treści w grupie za pomocą klucza Lang.

Chociaż język to świetny sposób na grupowanie treści, nie jest on automatycznie wykrywany i przypisywany do stanu grupy przez zwykłych widzów.

Przykład 10-1. Niektóre przykładowe opcjonalne grupy treści

```
10 0 obj
<<
/Name (Watermark)
/Type OCG
/Usage <<
/Print << /PrintState /ON >>
/View << /ViewState /OFF >>
>>
>>
endobj
11 0 obj
<<
/Name (Do Not Print Or Export)
/Type OCG
/Usage <<
/Export << /ExportState /OFF >>
/Print << /PrintState /OFF >>
/View << /ViewState /ON >>
>>
>>
endobj
12 0 obj
<<
/Name (zoom = {0% 100%})
/Type OCG
/Usage <<
```

/Zoom <<

/max 1

>>

>>

>>

endobj

13 0 obj

<<

/Name (zoom = {100% 200%})

/Type OCG

/Usage <<

/Zoom <<

/max 2

/min 1

>>

>>

>>

endobj

Opcjonalne członkostwo w treści

Podczas gdy większość treści musi należeć tylko do jednej opcjonalnej grupy treści i powiązanego z nią słownika użytkowania, czasami treść może należeć do wielu grup, a grupy te mogą mieć stany sprzeczne. Aby zapewnić widzowi informacje niezbędne do rozwiązania takich potencjalnych konfliktów, ten rodzaj treści powinien być powiązany z opcjonalnym słownikiem członkostwa w treści (OCMD) zamiast z opcjonalną grupą treści. OCMD jest w istocie listą OCG, które określają różne potencjalne widoczności stwierdza wraz z polityką widoczności lub wyrażeniem widoczności opisującym sposób określania stanu.

Zasady widoczności

Polityka widoczności jest najprostszym sposobem na określenie sposobu rozwiązania różnych stanów OCG. Jest to preferowana metoda i powinna być stosowana na rzecz wyrażeń widoczności, jeśli to możliwe. Strategia jest określona przez obiekt o pojedynczej nazwie z jedną z czterech możliwych wartości. Jest to wartość klucza P w OCMD. Dostępne wartości zasad są następujące:

AllOn

Widoczny tylko wtedy, gdy wszystkie wpisy w OCG są WŁĄCZONE

AnyOn

Widoczny, jeśli którykolwiek z wpisów w OCG jest WŁĄCZONY

AnyOff

Widoczny, jeśli którykolwiek z wpisów w OCG jest WYŁĄCZONY

AllOff

Widoczny tylko wtedy, gdy wszystkie wpisy w OCG są WYŁĄCZONE

Przykład 10-2. Przykładowa polityka widoczności

<<

/Type /OCMD % Content belonging to this optional content

% membership dictionary is controlled by the states

/OCGs [12 0 R 13 0 R 14 0 R] % of three optional content groups.

/P /AllOn % Content is visible only if the state of all three

% groups is ON; otherwise it's hidden.

>>

Wyrażenia widoczności

Wyrażenie widoczności, jak sama nazwa wskazuje, pozwala na bardziej złożone wyrażenia boolowskie przy definiowaniu sposobu rozwiązania różnych stanów OCG. Gdy przeglądający ocenia wyrażenie widoczności, jeśli wyrażenie ma wartość true, opcjonalna grupa treści jest w stanie WŁĄCZENIA. Jeśli ma wartość false, stan jest WYŁĄCZONY. Wyrażenie to tablica, której pierwszym elementem jest nazwa reprezentująca operator boolowski (I, Lub lub Nie), podczas gdy kolejne elementy to opcjonalne grupy treści lub inne wyrażenia widoczności. Jeśli pierwszym elementem wyrażenia jest Nie, wówczas jest tylko jeden kolejny element; w przeciwnym razie (dla And lub Or) może mieć jeden lub więcej kolejnych elementów. Może wydawać się dziwne, że wyrażenie And lub Or może mieć tylko jeden powiązany element, ale ten element może sam być wyrażeniem widoczności zamiast odwołania do OCG. Przykład 10-3 pokazuje, jak powinna wyglądać polityka widoczności jako wyrażenie widoczności.

Przykład 10-3. Prosta widoczność

<<

/Type /OCMD

/VE [/And 12 0 R 13 0 R 14 0 R]

>>

Przykład 10-4 pokazuje bardziej złożone wyrażenie, które odnosi się do pięciu różnych grup, reprezentowanych przez obiekty od 1 do 5 w pliku PDF, o nazwach odpowiednio OCG 1 do OCG 5. Jeśli zostanie zapisany, przykład brzmi „OCG 1"; LUB (NIE „OCG 2") LUB („OCG 3" I „OCG 4" ORAZ „OCG 5") „.

Przykład 10-4. Złożone wyrażenie widoczności

<<

/Type /OCMD

/VE [/Or % Visibility expression: OR

1 0 R % OCG 1

[/Not 2 0 R] % NOT OCG 2

[/And 3 0 R 4 0 R 5 0 R] % OCG 3 AND OCG 4 AND OCG 5

]

>>

Opcjonalna konfiguracja treści

Opcjonalny słownik konfiguracji treści (OCCD) reprezentuje wstępnie skonfigurowaną konfigurację stanu dla jednej lub więcej grup. Plik PDF (zawierający OCG) może zawierać kilka OCCD, ale musi zawierać co najmniej jeden. Powodem tego jest to, że gdy dokument jest otwierany po raz pierwszy przez zgodny czytnik, stany grup są inicjowane na podstawie domyślnego OCCD dokumentu. Domyślne OCCD jest określone jako wartość klucza D w opcjonalnym słownik właściwości treści, do którego odwołuje się sam słownik katalogu dokumentów Chociaż wszystkie klucze w OCCD są opcjonalne, najczęściej występuje przynajmniej pewna kombinacja BaseState i ON lub OFF. BaseState to nazwa (WŁĄCZONA, WYŁĄCZONA lub Niezmieniona), która reprezentuje stan wszystkich grup, od których należy zacząć. Ten „stan bazowy" można następnie dostosować za pomocą klawiszy ON i / lub OFF, które wyświetlają listę określonych OCG, których stan należy ustawić na ON lub OFF. Powszechne jest również nadawanie OCCD nazwy za pomocą klucza Nazwa. Przykład 10-5 pokazuje prosty opcjonalny słownik konfiguracji treści.

Przykład 10-5. Proste OCCD

2 0 obj

<<

/Name (Example)

/BaseState /ON % turn them all on

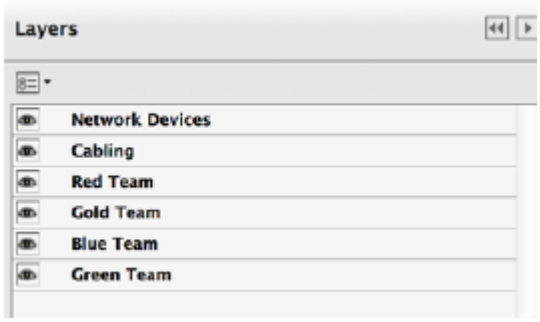
/OFF [1 0 R] % except this one

>>

endobj

Kolejność kluczy

W niektórych przypadkach wybory, które treści powinny być widoczne, a które nie, są dokonywane przez autora treści i pozostają stabilne przez cały okres życia treści. Jednak większość treści opisanych za pomocą opcjonalnych grup treści jest wykonywana, aby użytkownik mógł ręcznie zmienić widoczność treści. Na przykład przy złożonym schemacie architektonicznym lub elektrycznym użytkownik może potrzebować włączyć lub wyłączyć różne grupy elementów graficznych, aby zobaczyć tylko te ważne. Aby zażądać, aby przeglądarka plików PDF przedstawiła użytkownikowi listę opcjonalnych treści konfigurowanych przez użytkownika, używany jest klucz zamówienia. W przypadku prostej listy grup, takich jak ta z rysunku 10-1, wartością klucza jest jednowymiarowa tablica OCG.



% Present 6 OCGs in the list of OCGs

<<

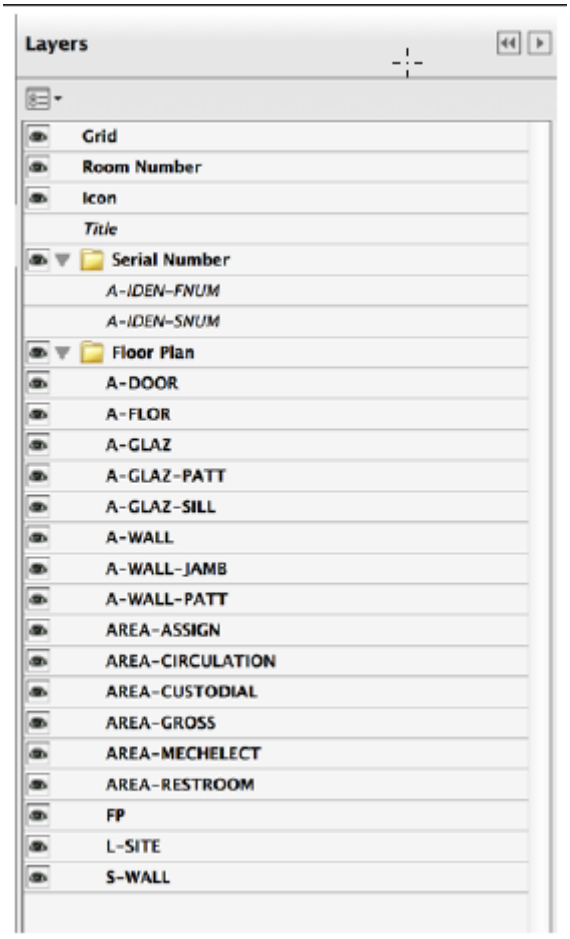
/Order[

40 0 R 42 0 R 38 0 R 32 0 R 36 0 R 34 0 R

]

>>

Możliwe jest również grupowanie grup OCG w różne hierarchiczne grupy, w sposób, który może być pomocny dla użytkownika lub który po prostu reprezentuje logiczne grupy treści. Na przykład, jeśli istnieje wyspecjalizowana podkategoria elementów, można je zebrać razem. Nazwę grupy można pobrać z OCG, która bezpośrednio poprzedza podtablicę elementów, lub można ją określić za pomocą ciągu tekstowego w podtablicy. Rysunek 10-2 pokazuje kilka przykładów hierarchicznych grup OCG.



% Using the Name of an OCG for the name of the group

<<

/Order[

24 0 R 37 0 R 34 0 R 25 0 R

30 0 R [

32 0 R 31 0 R

]

22 0 R [

27 0 R 28 0 R 29 0 R 36 0 R 40 0 R 23 0 R 26 0 R 35 0 R 41

0 R 43 0 R 45 0 R 44 0 R 46 0 R 42 0 R 38 0 R 33 0 R 39 0 R

]

]

>>

% Using a text string for the name of the group

<<

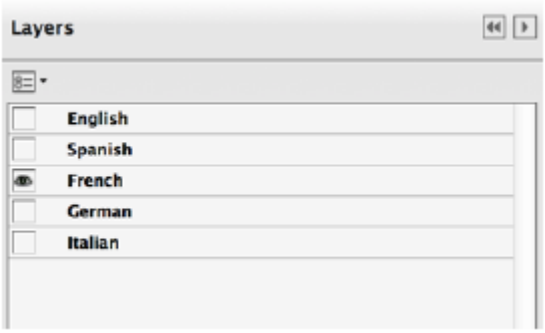
/Order [

[(Group 1) 1 0 R 2 0 R]

[(Group 2) 3 0 R 4 0 R]

RBGroups

OCG, które są wymienione w wartości tablicy klucza zamówienia, są po prostu przedstawiane jako (możliwie hierarchiczna) lista. Użytkownicy mogą je włączać i wyłączać w dowolnej kombinacji, którą chcą. W wielu przypadkach, takich jak schematy architektoniczne lub elektroniczne, działa to dobrze. Jeśli korzystasz na przykład z opcjonalnych grup treści, aby umożliwić użytkownikowi wybranie języka dla treści dokumentu, nie ma sensu, aby tekst w wielu językach był wyświetlany jednocześnie (z jedną wersją nad drugą!). Aby zastosować logikę, że tylko jeden z grupy elementów może być „włączony" na raz w interfejsie użytkownika widok, używany jest klucz RBGroups. RB w RBGroups oznacza przycisk opcji. Wartość klucza RBGroups to tablica jednej lub więcej tablic, przy czym każda z nich reprezentuje kolekcję OCG, których stany powinny być zgrupowane przy użyciu metafory przycisku opcji - tylko jedna na raz. Rysunek 10-3 ilustruje użycie RBGroups.



<<

/ RBGroups [

[55 0 R 57 0 R 59 0 R 61 0 R 63 0 R]

]

>>

Możliwe jest również użycie złożonych wyrażeń widoczności, aby osiągnąć to samo, ale grupy RBG są łatwiejsze i bardziej kompatybilne z różnymi przeglądarkami; dlatego są one bardziej powszechne i preferowane.

AS (stan automatyczny)

Klawisz automatycznego stanu (AS) w OCCD deklaruje, które z dostępnych wartości użytkowania mają zostać sprawdzone przez przeglądarkę PDF i użyte do automatycznej regulacji stanu różnych OCG, jak pokazano w przykładzie 10-6. Wartością klucza AS jest tablica słowników aplikacji użytkowania, z których każda określa, które OCG mają zostać sprawdzone pod kątem rodzaju zdarzenia (lub sytuacji) i kategorii użycia.

Przykład 10–6. Przykład OCCD ze stanem automatycznym

<<

/Order[17 0 R]

/ON[17 0 R 36 0 R]

/AS [

<<

/Event/View

/OCGs[36 0 R]

/Category[/View]

>>

<<

/Event/Print

/OCGs[37 0 R]

/Category[/Print]

>>

<<

/Event/Export

/OCGs[38 0 R]

/Category[/Export]

>>

]

>>

endobj

Opcjonalne właściwości treści

Nawet po dodaniu wszystkich plików OCG, OCMD i OOCD do pliku PDF wymagany jest jeszcze jeden słownik. To jest opcjonalny słownik właściwości treści; jest to wartość klucza OCProperties w słowniku katalogu dokumentów. Bez tego przeglądający nie będzie wiedział, że w pliku PDF jest jakakolwiek opcjonalna treść. W słowniku właściwości znajdują się dwa wymagane klucze:

OCG

Jest to tablica każdego OCG w pliku PDF, niezależnie od sposobu jego użycia, wymieniona w dowolnej kolejności. Jeśli nie ma go w tej tablicy, przeglądarka może zignorować go, gdy zostanie napotkany później.

D

Wartość tego klucza jest domyślnym opcjonalnym słownikiem konfiguracji treści dla pliku PDF. Opcjonalnie, jeśli plik PDF zawiera wiele OCCD, można je wymienić jako wartość klucza Configs w tym słowniku. Przykładowy słownik opcjonalnych właściwości treści pokazano w przykładzie 10-7.

Przykład 10–7. Przykładowy słownik OCProperties

<<

/OCGs [1 0 R 2 0 R 3 0 R] % we have 3 OCGs in this PDF

/D <<

/BaseState /ON % and they are all on

>>

>>

Oznaczanie treści jako opcjonalne

Do tej pory w tym rozdziale widzieliśmy, jak tworzyć opcjonalne grupy treści i całą infrastrukturę, aby udostępnić je przeglądarce PDF i jej użytkownikom. Jednak w pliku PDF zawierającym wszystkie wcześniej opisane obiekty nadal brakuje jednego kluczowego komponentu - w jaki sposób elementy treści są połączone z opcjonalnymi grupami treści lub opcjonalnymi słownikami członkostwa, które

mogą wpłynąć na ich widoczność? Każda treść, na którą widoczność będzie miała wpływ dana opcjonalna grupa treści, należy do tej grupy (lub należy do niej).

Opcjonalne treści w strumieniach treści

Aby określić, które określone elementy treści strumienia treści mają być skojarzone z opcjonalną grupą treści lub opcjonalnym słownikiem członkostwa w treści, muszą być zawarte między oznaczonymi operatorami treści BDC i EMC. Operator BDC użyje znacznika OC, a powiązana lista właściwości określa OCG lub OCMD, do którego należy treść. Ponieważ odwołanie do grupy odbywa się za pośrednictwem jej obiektu pośredniego, lista właściwości musi używać nazwanego formularza zasobu dla list właściwości. Przykład 10-8 pokazuje prosty przykład oznaczania zawartości w strumieniu jako opcjonalnej.

Przykład 10–8. Prosty przykład treści opcjonalnej

```
18 0 obj
<<
/Type /Page
/Contents 19 0 obj
/CropBox [ 0 0 612 792 ]
/MediaBox [ 0 0 612 792 ]
/Resources <<
/Properties << /MC0 20 0 R >>
>>
>>
endobj
% (partial/incomplete) content stream object
19 0 obj
<<>>
stream
/OC /MC0 BDC % this group is part of the MC0 OCG
0 g
1 i
BT
/T1_0 1 Tf
12 0 0 12 234 364 Tm
(This page intentionally blank\000)Tj
ET
```

EMC

endstream

endobj

% optional content group

20 0 obj

<<

/Intent /View

/Name (Alternate Content)

/Type /OCG

>>

endobj

Treść może być powiązana z co najmniej jednym OCG i / lub OCMD, a zawartość grupy nie musi być ciągła w strumieniu treści, a nawet w tym samym strumieniu treści. To sprawia, że opcjonalna treść jest niezwykle elastyczna do wielu celów

% Content stream with each color in a separate OCG

/OC /MC0 BDC % Red rectangles

1 0 0 rg

279 159.356 -211 86 re

f

EMC

/OC /MC1 BDC % Green rectangles

0 1 0 rg

755 160.356 -211 86 re

f

EMC

/OC /MC2 BDC % Blue rectangles

0 0 1 rg

519 159.356 -211 86 re

f

EMC

/OC /MC0 BDC % Red rectangles

1 0 0 rg

279 62.356 -211 86 re

f

EMC

/OC /MC1 BDC % Green rectangles

0 1 0 rg

755 63.356 -211 86 re

f

EMC

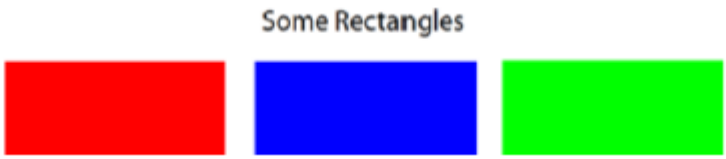
/OC /MC2 BDC % Blue rectangles

0 0 1 rg

519 62.356 -211 86 re

f

EMC



% Content stream

% This content doesn't belong to any group

BT

0 0 0 rg

/T1_0 1 Tf

27 0 0 27 314.2861 142.3242 Tm

[(S)-6(ome R)-4(ec)-12.9(tangles)]TJ

ET

/OC /MC3 BDC % everything here is part of MC3

/OC /MC0 BDC % this belongs to MC3 *AND* MC0

1 0 0 rg

279 159.356 -211 86 re

f

EMC

/OC /MC1 BDC % this belongs to MC3 *AND* MC1

0 1 0 rg

755 160.356 -211 86 re

f

EMC

/OC /MC2 BDC % this belongs to MC3 *AND* MC2

0 0 1 rg

519 159.356 -211 86 re

f

EMC

EMC

Opcjonalna treść dla formularzy XObjects

W przypadku formularzy XObjects lub XObjects obrazkowych, które są używane tylko w jednym strumieniu treści, wystarczy łatwo zaznaczyć strumień wywołujący. Jednak gdy takie obiekty są używane w wielu strumieniach treści i będą konsekwentnie w tej samej grupie (grupach), łatwiej jest skorzystać z klucza OC w słowniku XObject. Wartość klucza OC w słowniku XObject jest pośrednim odniesieniem do opcjonalnej grupy treści lub opcjonalnego słownika członkostwa w treści, do którego należy. Przykład 10-9 ilustruje to użycie.

Przykład 10–9. Przykład formularza XObject z powiązanym OCG

35 0 obj

<<

/FormType 1

/Subtype/Form

/Length 91

/Matrix[1.0 0.0 0.0 1.0 0.0 0.0]

/BBox[0.0 -31.7999 79.9918 0.143921]

/Resources<</Font<</TT0 32 0 R>>>>

/OC 37 0 R

>>

stream

0 0.75 1 rg

0 i

BT

/TT0 1 Tf

0 Tc 0 Tw 0 Ts 100 Tz 0 Tr

24 0 0 24 0 -24 Tm

(DRAFT)Tj

ET

endstream

endobj

Jeśli formularz XObject lub obraz XObject ma klucz OC, a ten sam XObject jest również częścią opcjonalnej grupy treści lub opcjonalnego słownika członkostwa treści skonfigurowanego w strumieniu treści, zostanie użyta kombinacja dwóch stanów (traktowana jako AND) określić widoczność obiektu. Na przykład, jeśli wartość OCG klucza OC ma wartość ON, ale strumień treści ma wartość OFF, wówczas stan obiektu będzie OFF. Możliwe jest również, aby indywidualna treść formularza XObject była częścią OCG lub OCMD. W takim przypadku widoczność XObject zostanie oceniona, a jeśli jego stan jest WŁĄCZONY, określone obiekty graficzne zostaną ocenione.

Opcjonalna treść adnotacji

Chociaż poszczególne elementy treści strumienia wyglądu adnotacji można oznaczyć i powiązać z określonymi opcjonalnymi grupami treści lub opcjonalnymi słownikami członkostwa w treści, bardziej powszechne jest po prostu skojarzenie całej adnotacji. To nie tylko ukrywa adnotację przed widokiem, ale także zapewnia, że plik PDF przeglądarka nie zapewni interakcji użytkownika z elementem. Jest to bardzo przydatne w przypadku interaktywnych formularzy, jako sposób ukrywania lub pokazywania elementów na podstawie innych kryteriów Podobnie jak w przypadku obiektów XObject w formie lub obrazie, wartość klucza OC w słowniku adnotacji jest pośrednim odniesieniem do opcjonalnej grupy treści lub opcjonalnego słownika członkostwa w treści, do którego należy. Przykład 10-10 pokazuje.

Przykład 10-10. Przykład adnotacji z powiązanym OCG

20 0 obj

<<

/Type /Annot

/Subtype /Highlight

/Rect [252.594 593.733 322.077 623.211]

/C [1 1 0]

/OC 37 0 R

>>

Tagowanie i struktura

Zorganizowany PDF

Jak widać we wszystkich poprzednich rozdziałach, PDF umożliwia rysowanie tekstu, wektorów, obrazów rastrowych, a nawet wideo i 3D na stronie, którą można wyświetlić lub wydrukować. Jednak treść jest taka: seria instrukcji rysowania. Nie ma kontekstu semantycznego ani strukturalnego. Nic nie wyróżnia jednego akapitu od drugiego ani jednego obrazu od drugiego. W rzeczywistości nie ma nawet pojęcia akapitu lub słowa - tylko garść glifów i związane z nimi kodowanie. Ograniczeniem tym jest funkcja PDF zwana strukturą logiczną. Umożliwia powiązanie hierarchicznej grupy obiektów, zwanych elementami struktury, z różnymi obiektami graficznymi na stronie i wszelkimi dodatkowymi atrybutami potrzebnymi do wystarczającego opisu tego obiekty. Jest to dość podobne pojęcie do języków znaczników, takich jak HTML lub XML, ale w PDF ta struktura i treść znajdują się w oddzielnych logicznych obszarach PDF, a nie są mieszane (jak na przykład w HTML). Ta separacja pozwala na uporządkowanie i zagnieżdżenie elementów logicznych całkowicie niezależnie od kolejności i położenia obiektów graficznych na stronach dokumentu. Chociaż istnieje szereg predefiniowanych typów elementów struktury, które umożliwiają organizację dokumentu w rozdziały i sekcje lub identyfikację specjalnych elementów, takich jak ryciny, tabele i przypisy, udogodnienia zapewniane przez PDF są dość rozszerzalne. Ta rozszerzalność pozwala autorom wybrać, jakie informacje strukturalne mają być zawarte i jak je reprezentować, umożliwiając procesorom nawigację po pliku bez znajomości określonych konwencji strukturalnych.

Drzewo Strukturalne

Jak wspomniano wcześniej, elementy strukturalne są ułożone w hierarchiczną strukturę zwaną StructTree lub drzewem struktury. W katalogu głównym drzewa znajduje się struktura drzewa, słownik, którego klucz Type ma wartość StructTreeRoot (patrz Przykład 11-1). Istnieją dwie inne rzeczy, które muszą być obecne w katalogu głównym: pierwsze z dzieci w drzewie i grupowanie elementów struktury według strony. Klawisz K w katalogu głównym wskazuje na pierwszy element konstrukcyjny w drzewie struktury. Jego wartością może być słownik pojedynczego elementu struktury lub tablica elementu struktury słowniki. Większość otagowanych plików PDF będzie miała pojedynczy wpis, który jest elementem struktury typu Dokument. Klucz ParentTree to drzewo liczb, które grupuje wszystkie elementy strukturalne na stronie wraz z powiązanym numerem / indeksem. Chociaż bardziej logiczne jest posiadanie porządkowej numer strony reprezentuje numer / indeks w drzewie liczb, który nie jest wymagany, jak zobaczymy, kiedy nauczymy się kojarzyć strukturę ze stroną

Przykład 11-1. Przykładowa struktura drzewa

```
1 0 obj

<<

/ K 3 0 R% pierwszy element konstrukcji

/ ParentTree 2 0 R% drzewo liczb elementów

/ Type / StructTreeRoot

>>

endobj

2 0 obj
```

<<

% jednostronicowy dokument z dwoma elementami

/ Nums [0 [4 0 R 5 0 R]]

>>

endobj

Elementy konstrukcji

Każdy element strukturalny jest reprezentowany przez słownik, którego klucz Type ma wartość StructElem. Określony typ elementu strukturalnego, który reprezentuje, jest określony jako wartość klucza S. Ta wartość jest nazwą obiektu i może być dowolna, chociaż zaleca się trzymanie się wartości omówionych w „Standardowych typach struktur". Jeśli zdecydujesz się użyć własnej nazwy dla elementu struktury, pamiętaj o użyciu mapy ról, aby odwzorować ją na najbliższy standardowy typ struktury. Klucz P w słowniku elementów struktury ma jako wartość element nadrzędny w drzewie, dzięki czemu procesor może poruszać się po drzewie we wszystkich kierunkach. W przypadku pierwszego dziecka rodzicem będzie StructTreeRoot. Podobnie jak w przypadku StructTreeRoot, elementy potomne każdego elementu można znaleźć jako wartość klucza K. Wartość K może być elementem struktury, tablicą elementów struktury lub liczbą całkowitą reprezentującą oznaczony identyfikator treści (MCID) na stronie docelowej treści. Ponadto możliwe jest odwołanie do adnotacji lub obiektu XObject, jeśli odwołujesz się do całego obiektu jako zawartości tego konkretnego elementu struktury. Chociaż możliwe jest bezpośrednie odniesienie do obiektu XObject, bardziej powszechne jest po prostu włączenie obiektu XObject do zaznaczonej sekwencji treści Chociaż nie jest to wymagane, zwykle w słowniku elementu struktury występuje klucz Pg, którego wartością jest słownik stron, na którym wyświetlana jest zawartość reprezentująca element. Innym wspólnym kluczem w słowniku elementu struktury jest klucz Lang, którego można użyć do jednoznacznej identyfikacji języka naturalnego mającego zastosowanie do danego elementu struktury (i jego elementów potomnych, chyba że zostanie to zastąpione inaczej). Wartością tego klucza jest standardowy kod RFC 3066. Przykład 11-2 pokazuje kilka przykładowych elementów struktury.

Przykład 11-2. Przykładowe elementy struktury

2 0 obj

<<

/K [3 0 R 4 0 R] % there are two children to the document

/Lang (en-US)

/P 1 0 R % back to the struct root

/S /Document

/Type /StructElem

>>

endobj

3 0 obj

<<

/K 0 % this is MCID 0 on the page

/P 2 0 R

/Pg 5 0 R % and here is the page

/S /P % P(aragraph)

/Type /StructElem

>>

endobj

4 0 obj

<<

/K 1 % MCID 1

/P 2 0 R

/Pg 5 0 R % and here is the page

/S /Figure

/Type /StructElem

>>

endobj

Standardowe typy konstrukcji

Definiując standard dla typów - czyli wartość klawisza S w słowniku elementów struktury - PDF zapewnia wspólne słownictwo, z którego mogą korzystać autorzy plików PDF, aby zapewnić, że procesory są w stanie zrozumieć zintegrowaną semantykę. Te typy są podzielone na elementy grupujące, elementy strukturalne na poziomie bloków (BLSE) i elementy strukturalne na poziomie liniowym (ILSE), w zależności od tego, czy element odnosi się do rzeczywistej treści i tego, jak normalnie treść ta byłaby umieszczona na stronie.

Grupowanie elementów

Elementy grupujące nie mają koncepcji prezentacji, lecz służą wyłącznie do grupowania innych zestawów elementów. Na przykład dokument będzie pojedynczym głównym elementem strukturalnym dla większości plików PDF. Inne typowe elementy grupujące są podobne do elementów występujących w innych systemach znaczników, takich jak Sect, Div i BlockQuote.

Blokowe elementy konstrukcyjne

Element struktury na poziomie bloku to dowolny obszar tekstu lub innej treści ułożonej w kierunku postępu bloku, taki jak akapit, nagłówek, pozycja na liście lub przypis. Tabela 11-1 przedstawia niektóre z tych typów treści i powiązane z nimi elementy struktury.

Typ struktury	**Opis**
H (Nagłówek)	Etykieta podziału dokumentu.
H1 – H6	Nagłówki z określonymi poziomami.

P (akapit)	Niski poziom podziału tekstu.
L (lista)	Sekwencja elementów o podobnym znaczeniu i znaczeniu. Bezpośrednimi elementami potomnymi będą elementy listy (LI).
LI (element listy)	Indywidualny element listy.
Lbl (etykieta)	Nazwa lub numer, który odróżnia dany element od innych na tej samej liście lub w innej grupie podobnych przedmiotów. Na przykład na liście słownikowej zawiera zdefiniowany termin; na liście wypunktowanej lub numerowanej to zawiera znak punktora lub numer pozycji na liście oraz wszelkie powiązane znaki interpunkcyjne.
LBody (Treść listy)	Treść opisowa elementu listy. Na przykład na liście słownikowej zawiera definicję termin.
Tabela (tabela)	Dwuwymiarowy układ prostokątnych komórek danych, prawdopodobnie o złożonej strukturze. Zawiera jeden lub więcej wierszy tabeli (TR) lub opcjonalny nagłówek tabeli (THead), po których następuje jeden lub więcej elementów tabeli elementy (TBody) i opcjonalną stopkę tabeli (TFoot).
TR (wiersz tabeli)	Wiersz nagłówków lub danych w tabeli.
TH (komórka nagłówka tabeli)	Komórka tabeli zawierająca tekst nagłówka opisujący jeden lub więcej wierszy lub kolumn tabeli.
TD (komórka danych tabeli)	Komórka tabeli zawierająca dane, które są częścią zawartości tabeli.
THead (grupa wierszy nagłówka tabeli)	Grupa wierszy stanowiących nagłówek tabeli.
TBody (grupa wierszy treści tabeli) tabeli.	Grupa wierszy, które stanowią główną część treści
TFoot (grupa wierszy stopki tabeli)	Grupa wierszy tworzących stopkę tabeli.

Elementy konstrukcyjne na poziomie liniowym

Element strukturalny na poziomie wbudowanym zawiera część tekstu lub innej zawartości o określonych cechach stylistycznych lub odgrywających określoną rolę w dokumencie. W obrębie zawierającego BLSE kolejne ILSE (ewentualnie zmieszane z innymi elementami treści) są uważane za ułożone kolejno w kierunku liniowym (np. Od lewej do prawej w zachodnich systemach pisma). ILSE może również zawierać BLSE. Tabela 11-2 przedstawia niektóre typowe typy elementów konstrukcyjnych na poziomie wbudowanym.

Typ struktury	**Opis**

Span (rozpiętość)	Ogólna wbudowana część tekstu, która nie ma żadnych szczególnych nieodłącznych cech.
Quote (cytat)	Wewnętrzna część tekstu przypisana komuś innemu niż autor otaczającego tekstu.
Note (Uwaga)	Pozycja tekstu objaśniającego, na przykład przypis dolny lub przypis końcowy, do którego odwołuje się treść dokumentu.
Refernece (odniesienie)	Odniesienie do treści w innym miejscu dokumentu.
BibEntry (pozycja bibliograficzna)	Odnośnik identyfikujący zewnętrzne źródło niektórych cytowanych treści.
Code (kod)	Fragment tekstu programu komputerowego.
Link (Link)	Powiązanie między częścią treści ILSE a odpowiednią adnotacją linku.
Annot (adnotacja)	Powiązanie między częścią treści ILSE a odpowiednią adnotacją

Artefakty

Artefakty to obiekty graficzne, które są dodawane przez system tworzenia treści, ale niekoniecznie reprezentują oryginalną treść autora, taką jak numery stron, Batesa lub obrazy tła. Obiekty graficzne, które nie są konieczne do zrozumienia treści autora, takie jak powtarzające się nagłówki lub reguły przypisów, są również identyfikowane jako artefakty.

Artefakt odróżnia się od rzeczywistej treści poprzez umieszczenie go w oznaczonej sekwencji treści za pomocą znacznika Artefakt. Przykład pokazano w przykładzie 11-3.

Przykład 11-3. Przykładowy artefakt numeru strony

```
% a part of a content stream

/Artifact

BMC

(Page 1) Tj

EMC
```

Mapowanie ról

Podczas korzystania z niestandardowych wartości dla typu struktury ważne jest, aby zapewnić słownik map ról opisujący, który ze standardowych typów struktur najbardziej przypomina. Słownik map ról to po prostu lista kluczy odpowiadających niestandardowym typom używanym dla każdego klucza, wartością jest nazwa standardowego typu struktury. Słownik ten jest określony jako wartość klucza RoleMap w katalogu głównym drzewa struktury. Przykład RoleMap pokazano w przykładzie 11-4.

Przykład 11-4. Przykład RoleMap

```
1 0 obj

<<
```

/K 3 0 R % the first structure element

/ParentTree 2 0 R % number tree of the elements

/Type /StructTreeRoot

/RoleMap 6 0 R % map the custom elements

>>

endobj

2 0 obj

<<

% a one-page document with two elements on it

/Nums [0 [4 0 R 5 0 R]]

>>

endobj

3 0 obj

<<

/K [4 0 R 5 0 R] % there are two children to the document

/Lang (en-US)

/P 1 0 R % back to the struct root

/S /Document

/Type /StructElem

>>

endobj

4 0 obj

<<

/K 0 % this is MCID 0 on the page

/P 3 0 R

/Pg 10 0 R % and here is the page

/S /Para % Para(graph)

/Type /StructElem

>>

endobj

5 0 obj

<<

/K 1 % MCID 1

/P 3 0 R

/Pg 10 0 R % and here is the page

/S /Chap % Chap(ter)

/Type /StructElem

>>

endobj

6 0 obj

<<

/Para /P

/Chap /Sect

>>

endobj

Kojarzenie struktury z treścią

Zidentyfikowanie, którzy operatorzy grafiki w parach treści są powiązani z określonym elementem struktury, odbywa się po prostu poprzez umieszczenie tych elementów w parze oznaczonych operatorów treści - w szczególności BDC i EMC - oraz powiązanej listy właściwości. Prosty przykład przedstawiono w przykładzie 11-5.

Przykład 11-5. Prosty przykład oznaczonej treści

BT

/TT0 1 Tf

-0.018 Tw 60 0 0 60 158.1533 714.3984 Tm

/P <</MCID 0 >>BDC

[(Hello W)80.2(orld)]TJ

EMC

ET

/Figure <</MCID 1 >>BDC

q

541 0 0 407 36 189.4000244 cm

/Im0 Do

Q

EMC

Ta treść odnosi się do elementów struktury z przykładu 11-2, które składały się z dwóch ponumerowanych elementów, 0 i 1, liczb, do których odnoszą się klucze MCID na listach właściwości. Chociaż nazwa użyta w tym przykładzie dla znacznika wokół obrazu to Figure, może to być Foo lub dowolny inny ciąg. Jest to wartość klucza S w słowniku elementów struktury, która faktycznie określa typ struktury. Używanie tej samej nazwy jest bardzo dobrym pomysłem i jest wysoce zalecane! Chociaż stosowanie struktury do operatorów graficznych w strumieniu treści strony jest najczęstszym podejściem, możliwe jest również stosowanie struktury w innych typach strumieni treści, takich jak ten powiązany z formularzem XObject. W większości przypadków cała forma XObject reprezentuje kompletny element struktury i można po prostu zamknąć operator Do wewnątrz zaznaczonej treści, jak w poprzednim przykładzie. Możliwe jest jednak także zastosowanie tego samego typu oznaczonych operatorów treści do poszczególnych operatorów graficznych w strumieniu treści XObject. Przy stosowaniu oznaczonych operatorów treści do poszczególnych grafik operatorów w strumieniu treści XObject, niedozwolone jest również dołączanie Do dla tego XObject wewnątrz jakiegoś innego elementu struktury.

Oznaczone pliki PDF

Chociaż dodanie struktury do pliku PDF może być bardzo przydatne, istnieją dodatkowe reguły, które można zastosować podczas pisania zawartości pliku PDF, aby umożliwić jeszcze bogatszy zestaw semantyki w końcowym pliku PDF. Po zastosowaniu tych reguł plik PDF jest nazywany tagowanym plikiem PDF. Oznaczony dokument PDF jest zgodny z następującymi zasadami:

• Cały tekst powinien być reprezentowany w formie, którą można przekonwertować na Unicode.

• Podziały słów powinny być wyraźnie przedstawione.

• Rzeczywistą treść należy odróżnić od artefaktów układu i stronicowania.

• Treść powinna być podana w kolejności związanej z jej pojawieniem się na stronie, określonej przez program piszący PDF.

• Należy zastosować podstawowy model układu opisujący rozmieszczenie elementów struktury na stronie.

• Do zdefiniowania znaczenia elementów konstrukcji należy zastosować zestaw standardowych typów konstrukcji.

Jednym z najważniejszych celów tych reguł jest zapewnienie wiarygodnego określenia całego tekstu w treści strony

Tagowany dokument PDF będzie również zawierał słownik informacji o znaku z wartością true dla klucza Marked. Słownik informacji o znaku jest wartością klucza MarkIn fo w słowniku katalogu dokumentów.

Metadane

W tej części poznasz różne sposoby włączenia metadanych do pliku PDF, od najprostszych ciągów na poziomie dokumentu po bogate XML dołączone do poszczególnych obiektów.

Słownik informacji o dokumencie

Nawet w oryginalnej wersji 1.0 PDF było jasne, że obecność metadanych była wymagana dla każdego formatu pliku, a na pewno takiego, który reprezentowałby dokumenty do elektronicznej dystrybucji i przechowywania. W tym celu informacje o dokumencie słownika (lub słownik informacyjny, a nawet tylko słownik informacyjny) został utworzony. Jak sama nazwa wskazuje, słownik informacyjny jest standardowym obiektem słownika PDF. Jednak w przeciwieństwie do każdego innego obiektu, z którym się zetknąłeś, do tego obiektu nie ma odniesienia w katalogu, ale w zwiastunie. Oryginalna specyfikacja PDF 1.0 udokumentowała cztery (opcjonalne) klucze do tego słownika, z których każdy dopuszcza tylko zakodowaną wartość ciągu w PDFDocEncoding.

Autor

Nazwisko osoby, która utworzyła dokument.

Data utworzenia

Data i godzina utworzenia dokumentu, sformatowana jako data.

Daty, jako rodzaj ciągu, zostały dodane do pliku PDF w wersji 1.1, więc bardzo wczesne pliki PDF mogą mieć wartość tego klucza jako prosty ciąg.

Autor

Oprogramowanie użyte do stworzenia oryginalnego dokumentu, który posłużył jako podstawa konwersji do formatu PDF. Jeśli plik PDF został utworzony bezpośrednio, wartość może pozostać pusta lub może być taka sama jak producenta.

Produkt

Nazwa produktu, który utworzył plik PDF.

W PDF 1.1 dodano cztery dodatkowe (opcjonalne) klucze, każdy zezwalający na wartość ciągu zakodowaną w PDFDocEncoding:

Tytuł

Tytuł dokumentu.

Przedmiot

Temat dokumentu.

Słowa kluczowe

Wszelkie słowa kluczowe powiązane z tym dokumentem.

ModDate

Data i godzina ostatniej modyfikacji dokumentu, sformatowana jako data.

Podczas gdy specyfikacja PDF dokumentowała tylko te osiem kluczy, programiści i użytkownicy pierwotnie mieli swobodę dodawania dodatkowych kluczy do słownika, których wartości mogą być dowolnego typu. Później specyfikacja PDF (a teraz sama ISO 32000-1) ograniczyła wartości tylko do ciągów tekstowych typu

Przykład 12-1. Przykładowy słownik informacyjny

1 0 obj

<<

/Title (PostScript Language Reference, Third Edition)

/Author (Adobe Systems Incorporated)

/Creator (Adobe FrameMaker 5.5.3 for Power Macintosh®)

/Producer (Acrobat Distiller 3.01 for Power Macintosh)

/CreationDate (D:19970915110347-08'00')

/ModDate (D:19990209153925-08'00')

/DEV1_CustomKey (Here is a sample custom key using a proper second class name)

/CustomKey2 (Here is a sample custom key improperly using a first class name)

>>

endobj

Strumienie metadanych

Jak widać, słownik informacyjny ma wiele ograniczeń, w tym typowanie danych i obsługę złożonych struktur (takich jak tablice lub słowniki), nie wspominając już o powiązaniu tylko z dokumentem jako całością, a nie z pojedynczymi obiektami w pliku PDF. Aby rozwiązać te problemy, wprowadzono nowy typ metadanych zwany strumieniem metadanych (ponieważ, jak zapewne można się domyślić, jest przechowywany jako obiekt strumienia). Strumienie te można powiązać nie tylko z dokumentem, ale z dowolnym zawartym w nim obiektem (chociaż niektóre z nich mają większe prawdopodobieństwo, że je będą mieć niż inne). Ponieważ są to strumienie, do danych mogą być zastosowane standardowe filtry kompresji lub kodowania. Jednak zdecydowanie zaleca się, aby przynajmniej strumień metadanych na poziomie dokumentu był przechowywany jako zwykły tekst. W rzeczywistości niektóre standardy PDF wymagają przechowywania strumienia metadanych na poziomie dokumentu w postaci zwykłego tekstu. Chociaż powodem zalecania strumieni metadanych w postaci zwykłego tekstu jest umożliwienie narzędziom nieobsługującym plików PDF sprawdzanie, katalogowanie i klasyfikowanie dokumentów, Okazuje się, że może to być problematyczne, jeśli dokona się aktualizacji przyrostowych dokumentu. Gdy dokument zostanie zaktualizowany, pojawi się drugi (zaktualizowany) strumień metadanych, który dezorientuje narzędzie nieobsługujące plików PDF. Zawartość strumienia metadanych znajduje się w określonej gramatyce Extensible Markup Language (XML) znanej jako Extensible Metadata Platform (XMP), która została znormalizowana jako ISO 16684-1.

XMP

W XMP metadane składają się z zestawu właściwości. Właściwości są zawsze powiązane z konkretną jednostką (zwaną zasobem). Oznacza to, że właściwości dotyczą zasobu. Każda dana właściwość ma nazwę i wartość. Pod względem koncepcyjnym każda właściwość zawiera oświadczenie dotyczące

zasobu w postaci „Nazwa właściwości zasobu to wartość właściwości”. Na przykład „Autorem Moby Dick jest Herman Melville”. Oświadczenie to jest reprezentowane przez metadane, w których zasobem jest książka Moby Dick, nazwa nieruchomości to autor, a wartość nieruchomości to Herman Melville.

Przykład 12-2. Przykład XMP

<xmp:CreateDate>1851-08-18</xmp:CreateDate>

<xmp:CreatorTool>Ink and Paper</xmp:CreatorTool>

<dc:creator>

<rdf:Seq>

<rdf:li>Herman Melville</rdf:li>

</rdf:Seq>

</dc:creator>

<dc:title>

<rdf:Alt>

<rdf:li xml:lang="x-default">Moby Dick</rdf:li>

</rdf:Alt>

</dc:title>

Wszystkie właściwości, pola struktury i nazwy kwalifikatorów w XMP muszą być prawnie kwalifikowanymi nazwami XML. Oznacza to, że muszą to być poprawnie sformułowane nazwy XML oraz przestrzeń nazw XML - dotyczy to właściwości najwyższego poziomu, pól struktury i kwalifikatorów. Jest to wymaganie odziedziczone po RDF (Resource Definition Framework), technologii opartej na XMP na podstawie.

Schematy

Schemat XMP to zestaw nazw właściwości najwyższego poziomu we wspólnej przestrzeni nazw XML, wraz z ich typami danych i informacjami opisowymi. Zazwyczaj schemat XMP zawiera właściwości, które są istotne dla określonych typów dokumentów lub dla niektórych etapów przepływu pracy. Istnieje zestaw standardowych schematów, a także mechanizm definiowania nowych schematów. Termin „schemat XMP” jest tutaj używany do wyraźnego odróżnienia tej koncepcji od innych zastosowań terminu „schemat”, a zwłaszcza od języka schematu XML W3C. Schemat XMP jest zwykle mniej formalny i definiowany przez dokumentację zamiast pliku schematu odczytywalnego maszynowo. Schemat XMP jest identyfikowany przez jego identyfikator URI przestrzeni nazw XML. Ma również powiązany prefiks przestrzeni nazw, który może przyjmować dowolną wartość, chociaż są w użyciu często używane. Takie użycie przestrzeni nazw pozwala uniknąć konfliktu między właściwościami w różnych schematach, które mają tę samą nazwę, ale różne znaczenia. Na przykład dwa niezależnie zaprojektowane schematy mogą mieć właściwość twórcy: w jednym może oznaczać osobę, która utworzyła zasób; aw innym aplikacji użytej do utworzenia zasobu. Termin „najwyższy poziom” odróżnia właściwości katalogu głównego w schemacie XMP od nazwanego pola struktury w ramach wartości właściwości. Zgodnie z konwencją schemat XMP definiuje swoje właściwości najwyższego poziomu, ale nazwy pól struktury są częścią informacji o typie danych. Typy danych, które mogą reprezentować wartości właściwości XMP, dzielą się na trzy podstawowe kategorie: proste typy,

struktury i tablice. Ponieważ metadane XMP są przechowywane w formacie XML, wartości wszystkich typów są zapisywane jako ciągi znaków Unicode. Przykład 12-3 pokazuje prosty strumień metadanych.

Przykład 12-3. Przykładowy strumień metadanych

```
157 0 obj
<<
/Length 4520
/Subtype/XML
/Type/Metadata
>>
stream
<?xpacket begin="Ôªø" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/"
x:xmptk="Adobe XMP Core 5.1-c004 1.136136, 2010/05/14-18:06:40">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about=""
xmlns:xmp="http://ns.adobe.com/xap/1.0/">
<xmp:ModifyDate>2010-07-06T19:33:16-03:00</xmp:ModifyDate>
<xmp:CreateDate>2010-07-06T19:33:03-03:00</xmp:CreateDate>
<xmp:MetadataDate>2010-07-06T19:33:16-03:00</xmp:MetadataDate>
<xmp:CreatorTool>Acrobat PDFMaker 10.0 for Word</xmp:CreatorTool>
</rdf:Description>
<rdf:Description rdf:about=""
xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/">
<xmpMM:DocumentID>uuid:483d2fca-113d-4c81-b650-a39a67866aa6</xmpMM:DocumentID>
<xmpMM:InstanceID>uuid:83008e27-bcc3-4480-a03a-13dc46d7f1f5</xmpMM:InstanceID>
<xmpMM:subject>
<rdf:Seq>
<rdf:li>127</rdf:li>
</rdf:Seq>
</xmpMM:subject>
</rdf:Description>
```

<rdf:Description rdf:about=""

xmlns:dc="http://purl.org/dc/elements/1.1/">

<dc:format>application/pdf</dc:format>

<dc:title>

<rdf:Alt>

<rdf:li xml:lang="x-default">ISO TC 171/SC 2/WG5</rdf:li>

</rdf:Alt>

</dc:title>

<dc:description>

<rdf:Alt>

<rdf:li xml:lang="x-default">ISO/WD 19005-2</rdf:li>

</rdf:Alt>

</dc:description>

<dc:creator>

<rdf:Seq>

<rdf:li>Leonard Rosenthol</rdf:li>

</rdf:Seq>

</dc:creator>

</rdf:Description>

<rdf:Description rdf:about=""

xmlns:pdf="http://ns.adobe.com/pdf/1.3/">

<pdf:Producer>Adobe PDF Library 10.0</pdf:Producer>

</rdf:Description>

<rdf:Description rdf:about=""

xmlns:pdfx="http://ns.adobe.com/pdfx/1.3/">

<pdfx:SourceModified>D:20100706222950</pdfx:SourceModified>

<pdfx:Company>AIIM</pdfx:Company>

<pdfx:Manager>Betsy Fanning</pdfx:Manager>

</rdf:Description>

<rdf:Description rdf:about=""

xmlns:photoshop="http://ns.adobe.com/photoshop/1.0/">

<photoshop:headline>

<rdf:Seq>

<rdf:li>ISO/WD 19005-2</rdf:li>

</rdf:Seq>

</photoshop:headline>

</rdf:Description>

</rdf:RDF>

</x:xmpmeta>

<?xpacket end="w"?>

endstream

endobj

W tym przykładzie można zobaczyć różne aspekty XMP, o których wspomniano wcześniej: RDF, wiele przestrzeni nazw (dc, xmp, pdf i xmpMM) oraz zarówno proste (xmp: Data utworzenia), jak i typy tablic (dc: creator).

XMP w formacie PDF

Główny strumień metadanych dotyczy samego dokumentu i jest wartością pozycji metadanych w katalogu dokumentów. Ponadto do dowolnego strumienia lub obiektu słownikowego mogą być dołączone metadane za pośrednictwem wpisu Metadane. Zaleca się umieszczenie pozycji Metadane w słowniku lub strumieniu reprezentującym same dane (takie jak czcionka lub obraz). Wzdłuż tych linii metadane mogą być również powiązane z oznaczoną treścią w strumieniu treści. To powiązanie jest tworzone przez włączenie wpisu do słownika listy właściwości, którego kluczem są metadane i którego wartością jest słownik strumieni metadanych

Przykład 12-4. Przykładowy katalog z metadanymi

485 0 obj

<<

/Type/Catalog

/Metadata 54 0 R

/Pages 466 0 R

/ViewerPreferences<</Direction/L2R>>

>>

endobj

54 0 obj

<<

/Type/Metadata

/Subtype/XML

/Length 71746

>>

stream

<?xpacket begin="Ôªø" id="W5M0MpCehiHzreSzNTczkc9d"?>

<x:xmpmeta xmlns:x="adobe:ns:meta/"

x:xmptk="Adobe XMP Core 5.2-c001 63.139439,

2010/09/27-13:37:26 ">

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<rdf:Description rdf:about=""

xmlns:xmp="http://ns.adobe.com/xap/1.0/">

<xmp:CreateDate>2011-04-25T15:33:20Z</xmp:CreateDate>

<xmp:CreatorTool>Microsoft PowerPoint</xmp:CreatorTool>

<xmp:ModifyDate>2011-04-25T10:34:09-05:00</xmp:ModifyDate>

<xmp:MetadataDate>2011-04-25T10:34:09-05:00</xmp:MetadataDate>

</rdf:Description>

<rdf:Description rdf:about=""

xmlns:pdf="http://ns.adobe.com/pdf/1.3/">

<pdf:Keywords/>

<pdf:Producer>Adobe Mac PDF Plug-in</pdf:Producer>

</rdf:Description>

<rdf:Description rdf:about=""

xmlns:dc="http://purl.org/dc/elements/1.1/">

<dc:format>application/pdf</dc:format>

<dc:creator>

<rdf:Seq>

<rdf:li>Rick</rdf:li>

</rdf:Seq>

</dc:creator>

<dc:title>

<rdf:Alt>

<rdf:li xml:lang="x-default">Presentation2.pptx</rdf:li>

</rdf:Alt>

</dc:title>

</rdf:Description>

<rdf:Description rdf:about=""

xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/">

<xmpMM:DocumentID>uuid:51b92418-85ef-f843-ba4d-9c6ea3482287</xmpMM:DocumentID>

<xmpMM:InstanceID>uuid:5506218e-628b-8046-8af4-f2eb28096824</xmpMM:InstanceID>

</rdf:Description>

</rdf:RDF>

</x:xmpmeta>

<?xpacket end="w"?>

endstream

endobj

XMP a słownik informacji

Chociaż oparte na XMP metadane na poziomie dokumentu, do których odwołuje się słownik katalogu, są kanonicznymi metadanymi w dokumencie, można je zastąpić zaktualizowanymi metadanymi w słowniku informacji o dokumencie. Jest tak w przypadku, gdy starszy procesor PDF aktualizuje tylko słownik informacji. Zarówno XMP, jak i słownik informacyjny zapewniają datownik. Jeśli znacznik daty w XMP jest równy lub późniejszy niż data modyfikacji w słowniku informacji o dokumencie, metadane XMP są uznawane za wiarygodne. Jeśli jednak data modyfikacji w słowniku informacji o dokumencie jest późniejsza niż znacznik daty metadanych XMP, informacje zapisane w słowniku informacji o dokumencie zastąpią wszelkie semantycznie równoważne elementy w metadanych XMP. Pisząc metadane, zawsze używaj XMP.

Standardy PDF

Adobe Systems wprowadził na rynek PDF w 1993 roku, w tym publiczną specyfikację formatu. Jednak podczas gdy wielu programistów było w stanie stworzyć własne narzędzia do czytania i pisania dokumentów PDF, tylko Adobe mogło dodawać lub zmieniać funkcje w samym języku PDF. Zmiany te były korzystne dla firmy Adobe, ale nie zawsze dla różnych branż i segmentów rynku. Z tego powodu przemysł drukarski realizował pomysł opracowania podzbioru plików PDF, który mógłby następnie zostać znormalizowany przez międzynarodowy organ, taki jak Międzynarodowa Organizacja Normalizacyjna (ISO). Rezultatem tej pracy, a także pierwszym z podzestawów, był PDF / X. Kilka lat później różne interesy rządowe i biznesowe połączyły się, aby stworzyć PDF / A, podzbiór PDF koncentruje się na długoterminowej archiwizacji dokumentów PDF. Była to bardzo publiczna praca nad opracowaniem formatu PDF / A, która skłoniła inne branże do wprowadzenia innych wymienionych tutaj standardów. W 2007 r. Adobe uznało, że nadszedł czas na wprowadzenie pełnej specyfikacji PDF do ISO. Doprowadziło to do publikacji ISO 32000-1, która przekształciła PDF 1.7 w całkowicie otwarty standard międzynarodowy. PDF (ISO 32000) ISO 32000-1 reprezentuje formalizację i publikację pełnej wersji PDF 1.7 Adobe PDF Reference oraz otwartego standardu międzynarodowego. Komitet ISO (TC171 / SC2 / WG8) spędził wiele lat na tworzeniu standardu, który był technicznie identyczny z poprzednim Adobe PDF 1.7, ale przeszedł obszerny proces przepisywania w celu wyjaśnienia wielu kwestii. Ten standard jest podstawą wszystkich przyszłych generacji standardów PDF. Po ujednoliceniu „Adobe's PDF" następnym krokiem komitetu było rozpoczęcie prac nad wprowadzeniem funkcji, których chcieli od lat w formacie PDF. Niektóre przykłady obejmują włączenie danych geoprzestrzennych lub danych GIS, kompensację punktu czerni dla bogatszego odwzorowania kolorów oraz ulepszone znakowanie i strukturę. Obecnie ISO 32000-2 jest w fazie rozwoju i ma zostać opublikowany w 2015 roku.

PDF / X (ISO 15930)

Pierwszy ze standardów podzbioru PDF, PDF / X, koncentrował się na potrzebach profesjonalistów drukarskich, grafików i twórców, zapewniając specyfikacje dotyczące tworzenia, przeglądania i drukowania końcowych stron gotowych do wydruku lub gotowych do druku. PDF / X zawiera wytyczne dotyczące kluczowych aspektów drukowania, takich jak przestrzenie kolorów, osadzanie czcionek i stosowanie zalewkowania. Ogranicza także inne treści - takie jak osadzone multimedia - które nie służą bezpośrednio do produkcji wysokiej jakości wydruków. PDF / X-1a (ISO 15930-1), pierwszy z tej rodziny, został opublikowany w 2001 roku. Określono w nim podzbiór PDF, który wymagał osadzenia wszystkich czcionek, nie zawierał adnotacji, a dozwolone były tylko przestrzenie kolorów były DeviceCMYK, DeviceGray i Separacja. Wymagało również, aby plik określał, czy został uwięziony, czy nie. W 2002 r. Wprowadzono PDF / X-3 (ISO 15930-3); zbudowano go w formacie PDF / X-1a, ale dopuszczono kolory koloryzowane za pomocą przestrzeni kolorów, takich jak ICCBased i Lab. Te oryginalne wersje PDF / X-1a i PDF / X-3 były oparte na PDF 1.3. W 2003 r. Wprowadzono aktualizacje zarówno do formatu PDF / X-1a (15930-4), jak i PDF / X-3 (15930-6), które dostosowały je do formatu PDF 1.4, ale nadal nie pozwalały na przezroczystość. Minęło trochę czasu, zanim branża drukarska zrozumiała korzyści i obawy związane z wykorzystaniem przezroczystości w formacie PDF, więc dopiero w 2008 roku opublikowano pierwszą wersję PDF / X-4 (ISO 15930-7). Wersja wprowadziła model PDF / X-3 zarządzany kolorami dla PDF do wersji PDF (1.6) zawierającej przezroczystość, lepszą kompresję (np. JPEG2000 i ObjectStream; patrz „Tabela powiązań" na stronie 16) i więcej. Jednak w tej oryginalnej wersji z 2008 roku odkryto kilka problemów, które zostały zastąpione przez aktualizację w 2010 roku. Pliki PDF / X-1a, PDF / X-3 i PDF / X-4 stanowią całkowicie samodzielne pliki. Wszystkie czcionki, przestrzenie kolorów i obrazy muszą znajdować się w pliku PDF - niedozwolone są odniesienia zewnętrzne. Chociaż jest to z pewnością normalny przypadek drukowania, istnieją sytuacje, w których

zdolność odwoływania się do treści zewnętrznych lub zasobów zewnętrznych może być korzystna, na przykład w światach danych zmiennych i transakcyjnych. Dla tych branż utworzono PDF / X-5 (ISO 15930-8), aby zapewnić standard, który umożliwia albo pojedyncze pliki, albo zestawy wielu plików wykorzystujących odwołania zewnętrzne. Był też plik PDF / X-2 (ISO 15930-2), który próbował dostarczyć metodę obsługi treści zewnętrznych. Niestety projekt był dość kiepski i nigdy nie został publicznie wdrożony. Więc przemysł udaje, że to się nigdy nie wydarzyło!

PDF / A (ISO 19005)

W 2003 r. Przedstawiciele rządu USA zwrócili się do Adobe Systems z prośbą o utworzenie podzbioru plików PDF, który byłby bardziej niezawodny i spójny niż generowane wówczas przez producentów plików PDF. Potrzebowali tej niezawodności, aby móc poprawnie przechowywać dokumenty PDF w swoich archiwach przez 10, 20, 50 lub więcej lat. PDF / A-1, opublikowany w 2005 r. I oparty na PDF 1.4, stanowi standard tworzenia, przeglądania i drukowania dokumentów cyfrowych w celu długoterminowego przechowywania. Dokumenty te są całkowicie samodzielne z osadzonymi czcionkami i jednolity kolor i bez szyfrowania, dzięki czemu mogą służyć jako ostateczne dokumenty zapisu. Odniesienia do treści zewnętrznych są niedozwolone, ponieważ elementy te mogą nie istnieć w przyszłości. Ponadto wymagane są metadane oparte na XMP, aby upewnić się, że plik samopisuje się. PDF / A-2, opublikowany w 2010 roku i oparty na PDF 1.7 (ISO 32000-1), niesie ze sobą wiele wymaganych funkcji, takich jak przezroczystość i ulepszona kompresja wykorzystująca JPEG2000 i strumienie obiektów. Wykorzystując ISO 32000-1 jako podstawowy standard, PDF / A-2 stał się pierwszym standardem dla podzbiorów PDF, który jest całkowicie oparty na ISO. Jedną z funkcji, o które prosiło wiele osób, ale nie przekształcono ich do formatu PDF / A-2, była możliwość posiadania dowolnych załączników, takich jak dane XML. Aby zapewnić rozwiązanie dla tych przepływów pracy, komitet (ISO TC171 / SC2 / WG5) opracował PDF / A-3 w 2012 r. Każdy ze standardów PDF / A ma co najmniej dwa poziomy zgodności, a i b. A poziom zgodności można uznać za poziom „wszystko" lub „dostępny", ponieważ wymaga on zgodności z pełnym zestawem wymagań dla normy, w tym oznaczenia pliku i uporządkowania go pod kątem dostępności. Poziom zgodności b lub „podstawowy" jest powszechnie stosowany w przypadku prostszych treści, takich jak zeskanowane dokumenty lub dokumenty, których oryginalne źródło cyfrowe nie jest już dostępne. PDF / A-2 i PDF / A-3 wprowadził trzeci poziom zgodności, który można postrzegać jako znajdujący się pomiędzy innymi: poziom u, dla „Unicode". Wymaga to, aby cały tekst w pliku mógł być odwzorowany na Unicode.

PDF / E (ISO 24517)

Mimo że PDF od zawsze był wykorzystywany na rynku inżynieryjnym, wprowadzenie obsługi opcjonalnej zawartości, 3D i pomiarów spowodowało znaczne upowszechnienie formatu wśród architektów, inżynierów, specjalistów budowlanych i zespołów produkujących produkty. PDF / E był bezpośrednim wynikiem chęci społeczności inżynierów do opracowania specyfikacji opartej na PDF / A. Koncentruje się na ich potrzebach związanych z wymianą dokumentacji i rysunków w łańcuchu dostaw w celu udostępniania dokumentów lub usprawnionego przeglądu i znaczników. Określa wymagania dla plików PDF, które sprawiają, że jest on bardziej odpowiedni do budowania, produkcji i geoprzestrzennych przepływów pracy dzięki obsłudze interaktywnych mediów, animacji i 3D. Ponieważ jednym z kluczowych przypadków użycia standardu było umożliwienie współdzielenia treści, umożliwia ono szyfrowanie i zarządzanie prawami cyfrowymi. Opublikowano w 2007 r., PDF / E-1 oparty jest na PDF 1.6. PDF / E-2, który jest opracowywany przez ISO TC171 / SC2 / WG7, ma zostać opublikowany w 2014 roku i będzie oparty na ISO 32000-2.

PDF / VT (ISO 16612-2)

Podczas gdy PDF / X-4 i PDF / X-5 zaspokajają większość potrzeb przemysłu drukarskiego, osoby pracujące z drukowaniem zmiennym i transakcyjnym potrzebowały pewnych szczególnych dodatków. W swoich dużych przepływach pracy obejmujących wyciągi bankowe i faktury biznesowe, włączenie bogatych metadanych i identyfikowalnych części dokumentów było koniecznością. Ponadto wiele części tych dokumentów jest ponownie wykorzystywanych i wymagany był optymalny sposób ich identyfikacji i ponownego wykorzystania w procesie drukowania. PDF / VT, oparty na PDF / X, został opublikowany w 2010 r. I spełnił te potrzeby. Istnieją trzy wersje PDF / VT. PDF / VT-1 jest całkowicie samodzielnym formatem pliku opartym na PDF / X-4, podczas gdy PDF / VT-2 pozwala na odwołanie do poszczególnych formularzy XObject w innych plikach i jest zbudowany na PDF / X-5. Istnieje również szczególny przypadek pliku PDF / VT-2 o nazwie PDF / VT-2, który można obsługiwać w całości w strumieniu, zamiast wymagać faktycznego zapisu do plików.

PDF / UA (ISO 14289)

Wiele rządów na całym świecie ma prawa, które wymagają, aby ich publikacje były dostępne dla wszystkich ich obywateli, niezależnie od ich niepełnosprawności. PDF od dawna ma funkcje (zwłaszcza poprzez oznaczanie i strukturę), które umożliwiają dowolny dokument zgodny z międzynarodowymi standardami dostępności ogólnego przeznaczenia, ale do niedawna nie istniał żaden standard skoncentrowany na PDF, który nadawałby konkretny kierunek. PDF / UA-1, opublikowany w 2012 r., Jest oparty na ISO 32000-1 i zawiera zestaw wytycznych do tworzenia plików PDF, które są powszechnie dostępne, a zatem bardziej czytelne dla osób niepełnosprawnych, takich jak osoby z zaburzeniami widzenia lub ograniczoną mobilnością.

Inne standardy związane z plikami PDF

Należy pamiętać o kilku innych standardach związanych z plikami PDF. PAdES (ETSI TS 102 778) Unia Europejska (UE) od dawna jest silnym zwolennikiem podpisów cyfrowych lub elektronicznych, w tym standaryzacji podpisów w oparciu o inne serializacje, takie jak CMS (CAdES) i XML (XAdES). W 2008 r. Europejski Instytut Norm Telekomunikacyjnych (ETSI) opublikował ETSI 102778, który opiera się na normie ISO 32000-1 w celu ułatwienia bezpieczniejszych transakcji biznesowych w formie papierowej. Ta norma określa serię profili dla PAdES (zaawansowane podpisy elektroniczne PDF), które są zgodne z europejską dyrektywą 1999/93 / WE.

PDF i Opieka zdrowotna

Inicjatywa PDF Healthcare, choć nie jest standardem formatu plików, zapewnia najlepsze praktyki i wytyczne dotyczące wdrażania, aby ułatwić przechwytywanie, wymianę, konserwację i ochronę informacji medycznych. Przestrzeganie tych wytycznych zapewnia bezpieczniejszy pojemnik elektroniczny, który może przechowywać i przekazywać informacje związane ze zdrowiem, w tym dokumenty osobiste, dane XML, obrazy i dane DICOM, notatki kliniczne, raporty laboratoryjne, formularze elektroniczne, zeskanowane obrazy, zdjęcia, cyfrowe zdjęcia rentgenowskie i EKG