# PDF
# Inside and Out


Leonard Rosenthol
PDF Standards Architect
Adobe Systems

# What's on the Agenda?

- PDF Overview
  - What's in there and what's not
  - Core Properties of PDF
- PDF Internals
  - PDF Objects
  - Example PDFs
    - Simplest PDF
    - Example Form
    - Updating
- Wrap Up
- Q & A

# PDF Overview

# PDF - Portable Document Format

- File format designed specifically for electronic distribution of "final form documents"
- Created by Adobe in 1992-1993, as part of their Acrobat product.
- PDF is an open international standard from the ISO
  - PDF 1.7 is now ISO 32000-1 <http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502>
  - Earlier versions of the PDF Reference are available from Adobe at <http://partners.adobe.com/public/developer/en/pdf/>

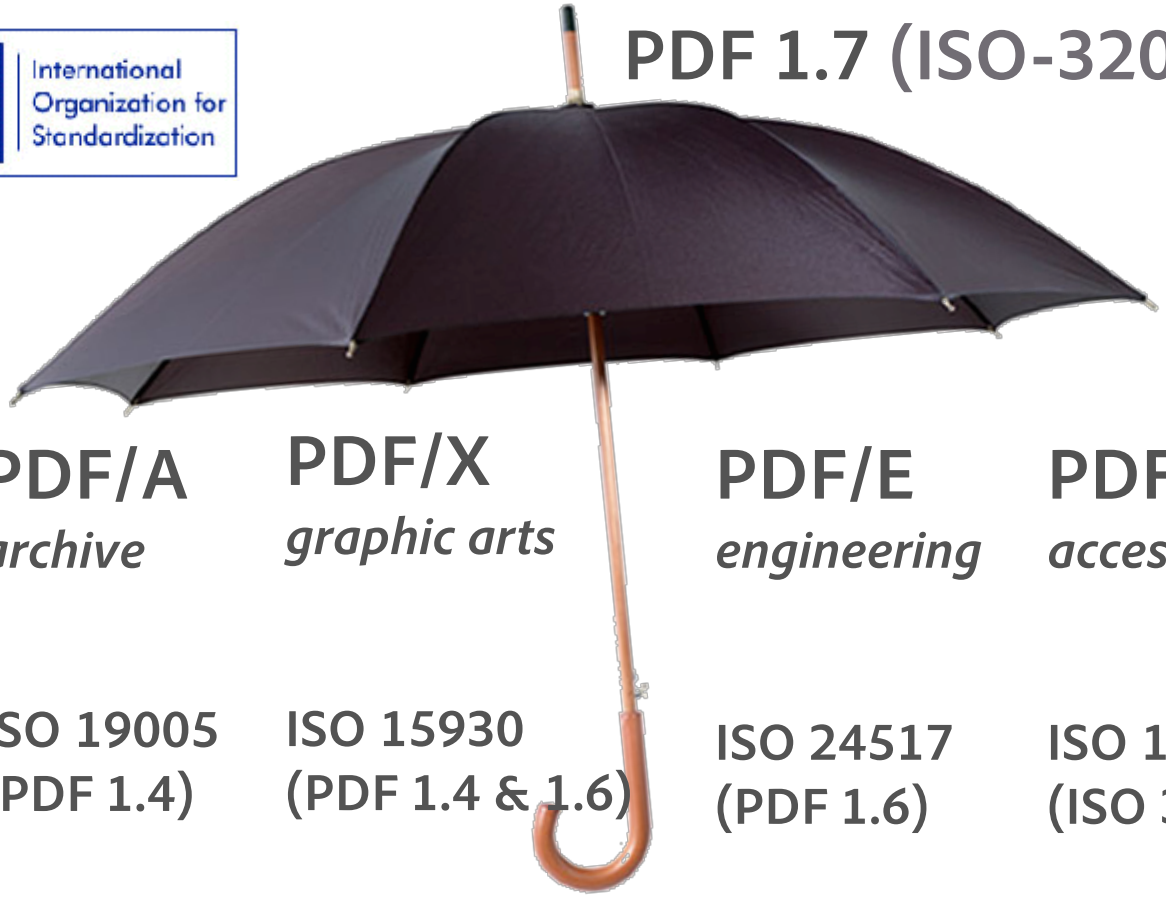
# PDF becomes an ISO Standard

- *"SAN JOSE, Calif. — **Jan. 29, 2007** — Adobe Systems Incorporated (Nasdaq:ADBE) today announced that it intends to release the full Portable Document Format (PDF) 1.7 specification to AIIM, the Enterprise Content Management Association, for the purpose of publication by the International Organization for Standardization (ISO)."*

- Establish a baseline standard
  - cover the millions of PDF files that already exist
  - describe the behavior of existing 'defacto standard" viewers
- Evolution of PDF specification as a public and open activity
  - ISO is widely recognized and respected & experienced with PDF-based standards
  - Adobe is just one of many participants in the working group

- **ISO 32000-1 was ratified in January 2008! (published 7/2008)**

# Establishing the ISO PDF Umbrella



PDF 1.7 (ISO-32000)

**PDF/A**
*archive*
ISO 19005
(PDF 1.4)

**PDF/X**
*graphic arts*
ISO 15930
(PDF 1.4 & 1.6)

**PDF/E**
*engineering*
ISO 24517
(PDF 1.6)

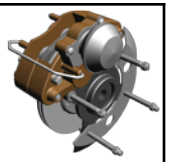
**PDF/UA**
*accessibility*
ISO 14289
(ISO 32000)

# ISO PDF Subset Standards

## PDF/X (ISO 15930)

- ✓ Predictable printing of digital files anywhere in the world
- ✓ Reliable blind transmission with reduced processing errors
- ✓ Removes things that can be problematic for printing
  - ✓ Annotations, forms, links, security, etc.
  - ✓ All fonts embedded – self contained
  - ✓ Consistent and reliable color


## PDF/E (ISO 24517)



- ✓ Reliable exchange of engineering documentation
  - ✓ Incorporation of complex data into compact PDF (3D, object-level data, etc.)
  - ✓ Accurate printing of engineering drawings
- ✓ Secure distribution of intellectual property
- ✓ Support for exchange/management of annotation and comment data

## PDF/A (ISO 19005)


- ✓ An archival replacement for paper
- ✓ Consistent, long-term retrieval and rendering
- ✓ All data self-contained – no scripting or encryption
- ✓ Includes document metadata (XMP) for search and retrieval
- ✓ Engineered for reliable migration and future compatibility
- ✓ Digital signatures for verification

## PDF/UA (ISO 14289)

- ✓ Definition of accessible PDF documents
  - ✓ Specifies required structural elements for PDF document content such as tables, forms, fonts
- ✓ Definition of accessible Readers and AT devices
- ✓ Includes participants from government, education & private sector

# PDF - What's in there?

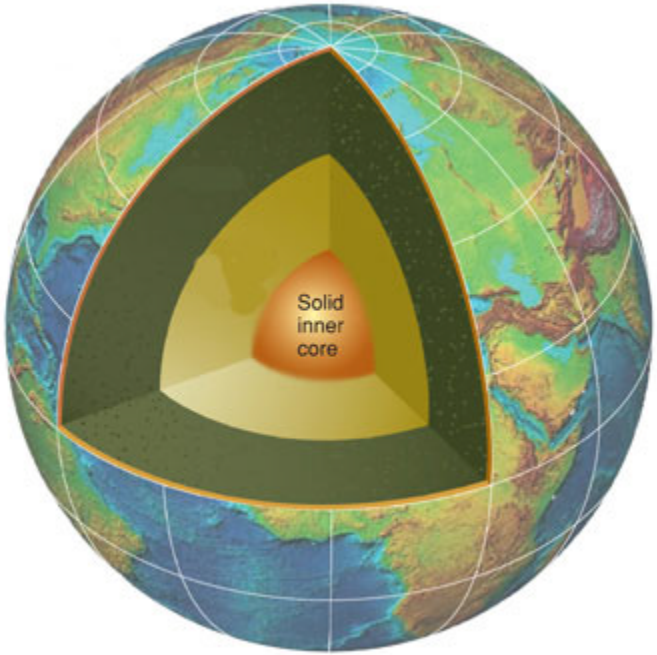
- PostScript/Adobe imaging model
  - Text & graphics in a device & resolution independent manner
- Bitmap Images
- Prepress Features (trapping, bleed, etc.)
- Navigation Tools (Bookmarks, Hyperlinks, etc.)
- Annotations - Text notes, "MarkUp"
- Forms
- Multimedia – movies/video, sounds/audio, Flash/SWF
- 3D
- Security & Authentication
- And more…

# PDF - What's NOT in there?

- PDF is NOT Postscript!
  - Transparency
  - ICC-based Color Management
  - Non-printable elements (hyperlinks, etc.)
  - Interactive elements (multimedia, 3D, forms)
  - Modern compression technology (JBIG2, JPEG2000)
  - No programming language constructs
  - Strict file structure allowing for random access
  - Presence of font metrics for viewing fidelity


- They do, of course, share a common heritage of a rich precise graphics model and core language features. PDF has simply evolved past it.

# Core Properties of PDF

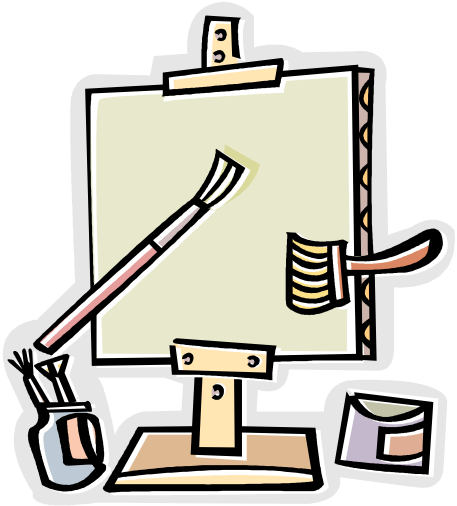


# Core Properties of PDF

- Adobe Imaging Model
- Portability
- Compression/Encryption
- Font Independence
- Random Access
- Incremental Update
- Extensibility

# Adobe Imaging Model

- Same model as Postscript, where a page is drawn by "placing paint" on a selected area
  - "figures" can be letter shapes, regions defined by lines and curves or sampled images
  - Paint can be any color (specified in variable color spaces)
  - Figures can be clipped to any other figure/shape
  - Figures are "overlaid" on each other, in the order they exist in the page description.
- Plus Transparency/Opacity (PDF 1.4)

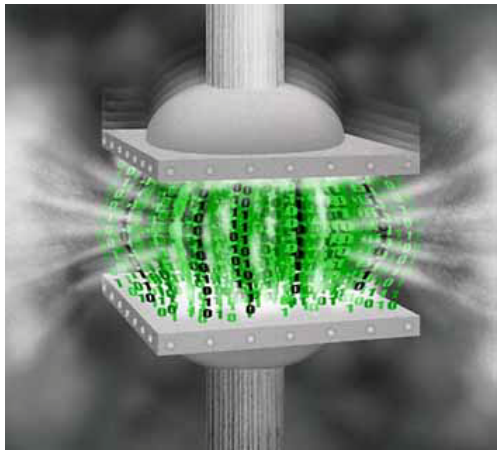


# Portability

- PDF files are binary, all 8 bits can be used
  - Support for 7 bit files exists
    - ASCII-85 when needing to encode to 7 bits
- Single document format regardless of platform
- Non-Roman language support via standard encodings as well as Unicode

# Compression/Encryption

- Support for a number of industry standard algorithms
- Compression
  - JPEG (for color & grayscale images)
  - CCITT Group 3 & 4, and RLE for monochrome images
  - Flate (ZIP) for text, graphics, etc.
  - JBIG2, JPEG2000
- Encryption
  - RC4
    - 40 bits (4.0)
    - 56 bits (4.0.5)
    - 128 bits (5.0 and later)
  - RSA Public Key Cryptography – Digital Signatures
    - PKCS#7, ASN.1, etc.
  - AES (7.0)



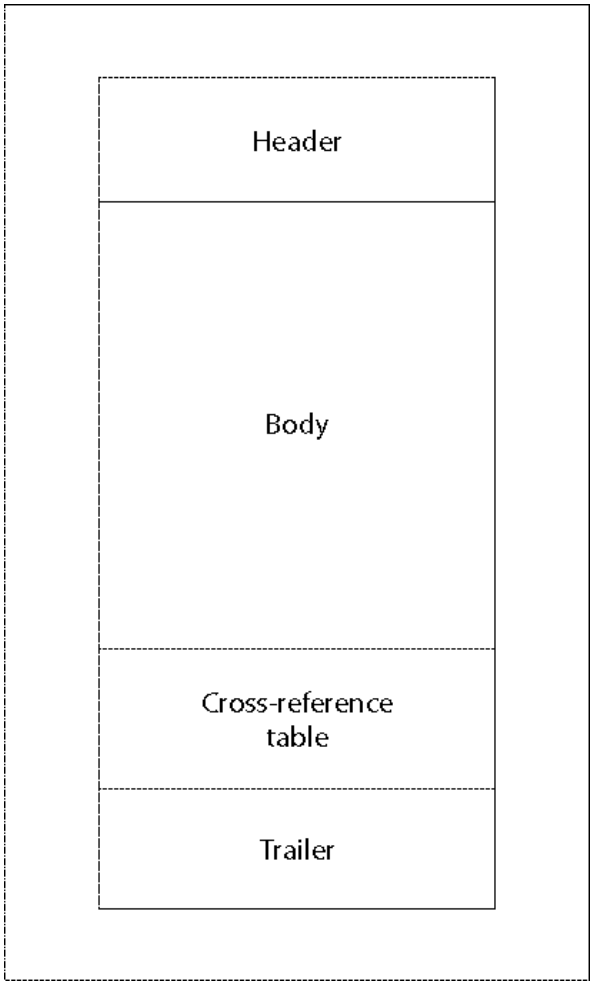
# Font Independence

- Support for all standard font formats
  - Type 1
  - TrueType
  - OpenType (both SFNT & CFF based)
- Support for font embedding
  - full & subset embedding
- Use of Font Descriptors
  - Name, character metrics (width, height), etc.


- To ensure correct display on all platforms and PDF viewers, you should embed.
  - Adobe Acrobat, however, provides font substitution facilities.

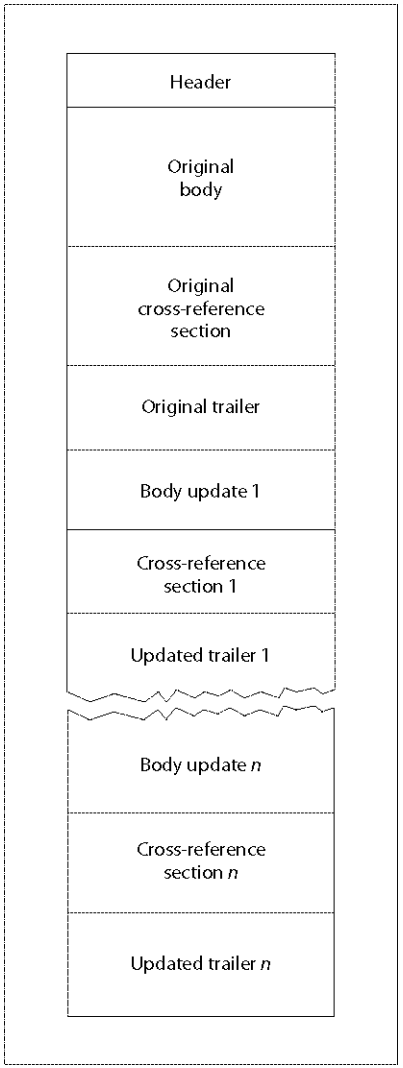
# Random Access

- Cross reference table maintains lists of pages, objects on a page, etc.
- XRef is stored at the end of the document, allowing for single pass creation and ease of location
  - Except in the case of linearized documents designed for byte-serving (ie. dynamic serving via the web)
- PDF 1.5 introduced a newer style of XRef "streams" which can be compressed, support more objects, etc.



# Incremental Update

- Modifications are written to the end of the file, leaving the original data intact
- A new xref table is written containing the new/modified data, and a link back to the old xref.
- Since original data is still present, support for multiple undos across save boundaries can be supported.
  - However Acrobat only provides a UI for this feature when Digital Signatures are used.



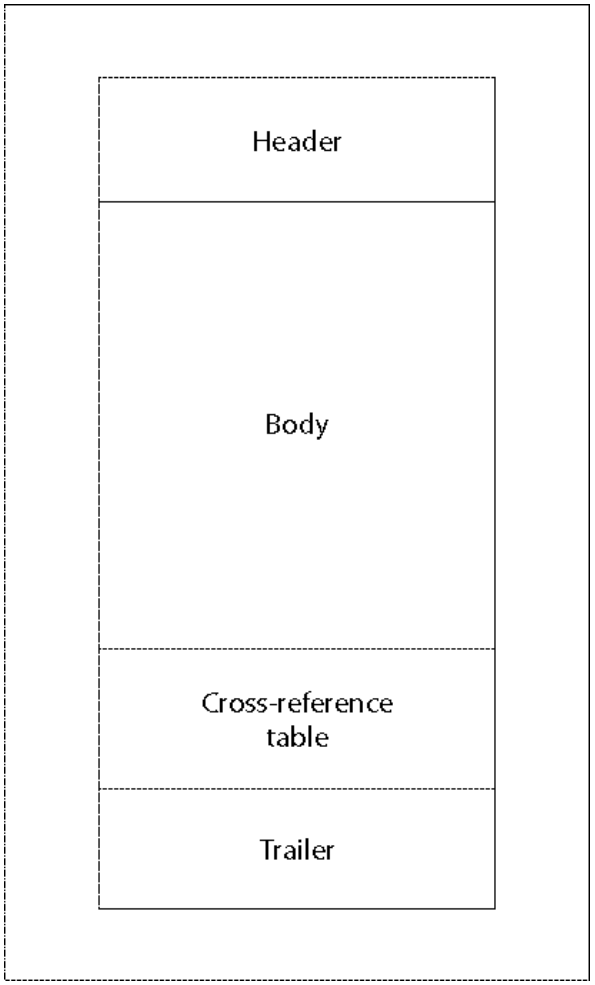
# Extensibility

- As seen by the features added to PDF since 1.0, you can see that new features can easily be added to PDF w/o breaking backwards compatibility.
- A viewer will simply ignore an object that it doesn't understand.

# PDF Internals

# PDF Document Layout

- Header
  - Specifies PDF version
- Body
  - Sequence of objects
- XREF
  - Where to find each object
- Trailer
  - Tells where to find XREF



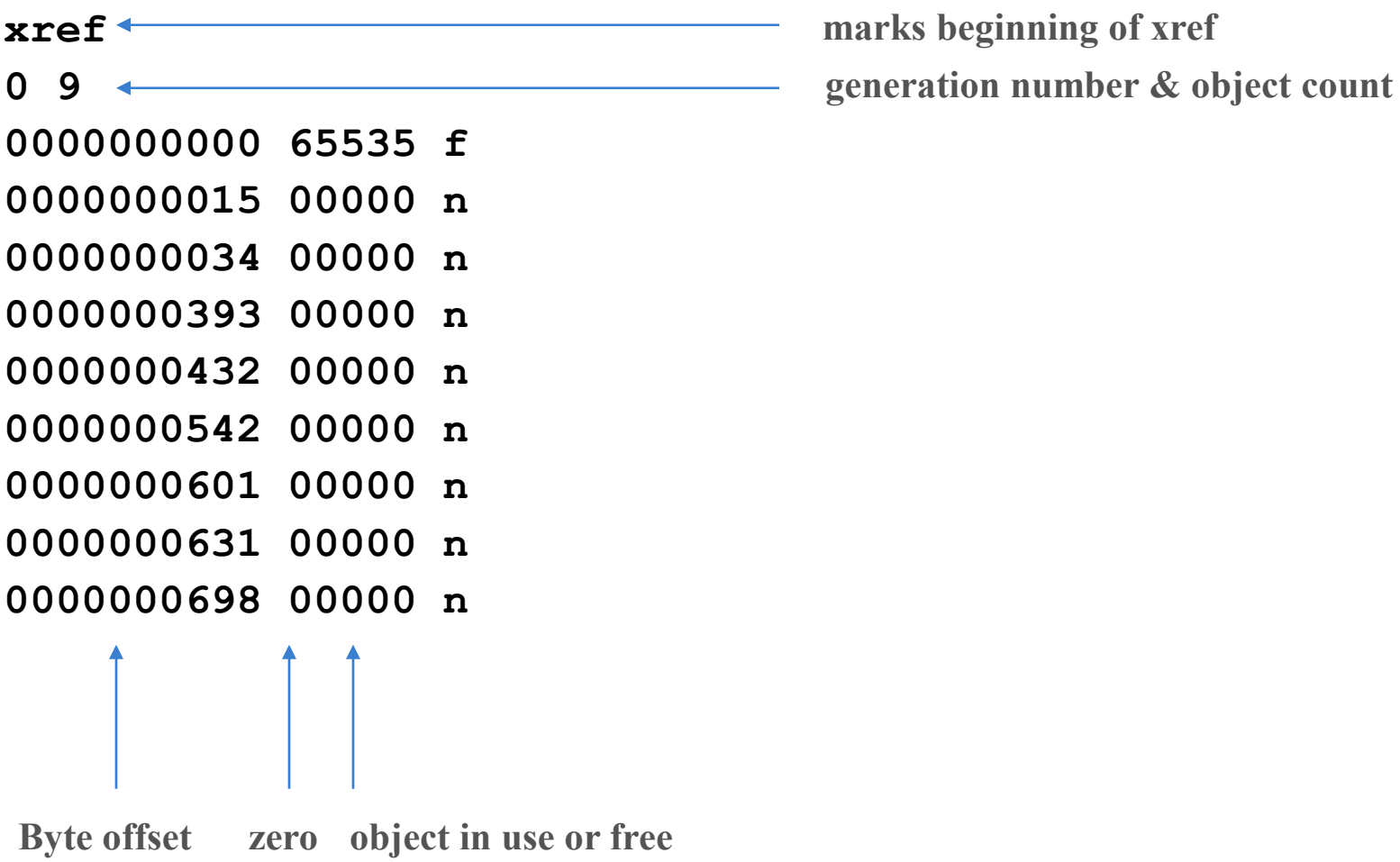
# Header

- Version of the PDF (%PDF-1.4)
- Specially placed "garbage bytes" (%âãÏÓ)
  - % - comment start
  - 4 bytes of "high order ASCII"
    - Can be anything, but most choose that sequence
  - Used to fool older systems into thinking PDF is binary vs. text

# Trailer

- MUST include references to the main object
  - Catalog or "Root" object (/Root)
  - Number of xref entries (/Size)
- It can also optionally include
  - Unique Identifier for the PDF (/ID)
  - Encryption information (/Encrypt)
  - Metadata (/Info)

# XREF



# XRef Streams

- PDF 1.5 introduced a (optional) new form of XREF
  - Compression
  - Compact format
  - Support >10GB docs
  - Future expansion

```
stream
01 0E8A 0 % Entry for object 2 (0x0E8A = 3722)
02 0002 00 % Entry for object 3 (in object stream 2, index 0)
02 0002 01 % Entry for object 4 (in object stream 2, index 1)
02 0002 02 % …
02 0002 03
02 0002 04
02 0002 05
02 0002 06
02 0002 07 % Entry for object 10 (in object stream 2, index 7)
01 1323 0 % Entry for object 11 (0x1323 = 4899)
endstream
```

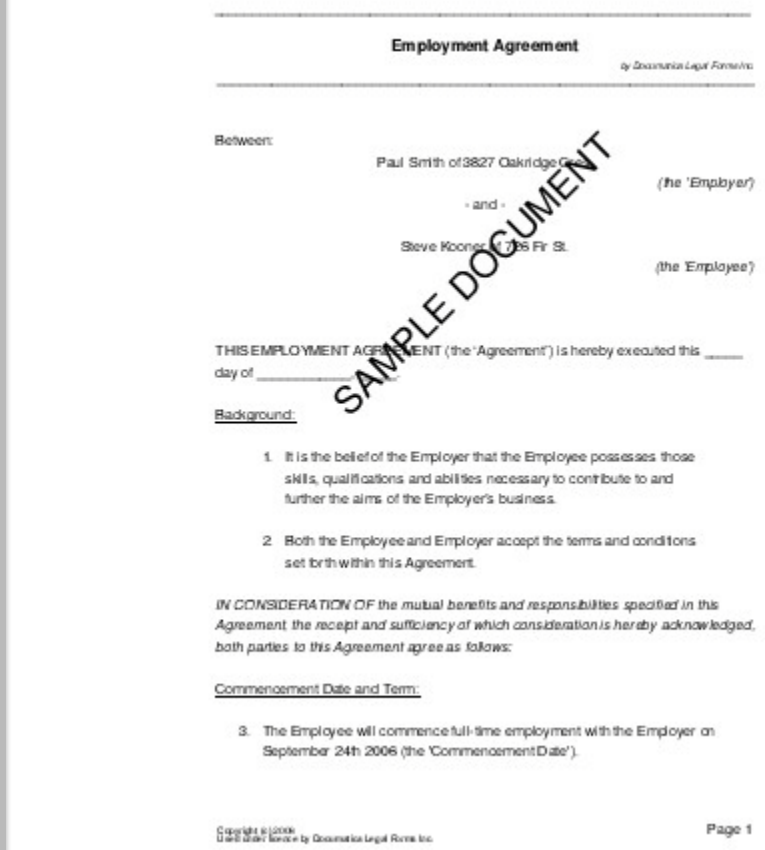
# The Body - Objects everywhere

- A PDF file is a collection of objects
  - 8 types of objects
- They are called “CosObjects”
  - COS = Carousel Object System
  - Pronounced either Cos or Coz
- Object are arranged into a logical structure that are documented in the PDF Reference Manual or ISO 32000-1.
- Objects are either direct or indirect
  - Direct objects are just “inline values”
  - Indirect objects have an object ID and a generation ID.
    - The object ID is the index into the xref table
    - The generation ID is almost always 0

# PDF Objects

| Object Type | Description | Example(s) |
|---|---|---|
| Null | | null |
| Boolean | | true or false |
| Integer | | 1, 2, 100, 613 |
| Real | | 0.05, .25, 300.90001 |
| Name | Used for key names & labels | /Type, /Page, /ThisIsName37, /UTF8Name#007 |
| String | PDDocEncoding, UTF16BE or Hex | (Testing), <FFFE0040>, <1C2D3F> |
| Array | heterogeneous ordered set of objects | [ 0 0 612 792 ], [ (T) –20.5 (H) 4 (E) ] |
| Dictionary | Name/value pairs | << /Type /Page<br>/Author (Leonard Rosenthol)<br>/Resources << /Font [ /F1 /F2 ] >> >> |
| Stream | Large blocks of data + descriptive dictionary.<br>Commonly holds content operators & images | << /Type /XObject /Subtype /Image<br>/Filter /FlateDecode >><br>stream …. endstream |

# A Sample PDF



**Employment Agreement**

*by Documents Legal Forms Inc.*

Between:

Paul Smith of 3827 Oakridge Cres. *(the 'Employer')*

- and -

Steve Kooner of 7?8 Fir St. *(the 'Employee')*



THIS EMPLOYMENT AGREEMENT (the 'Agreement') is hereby executed this _____ day of _____.

Background:

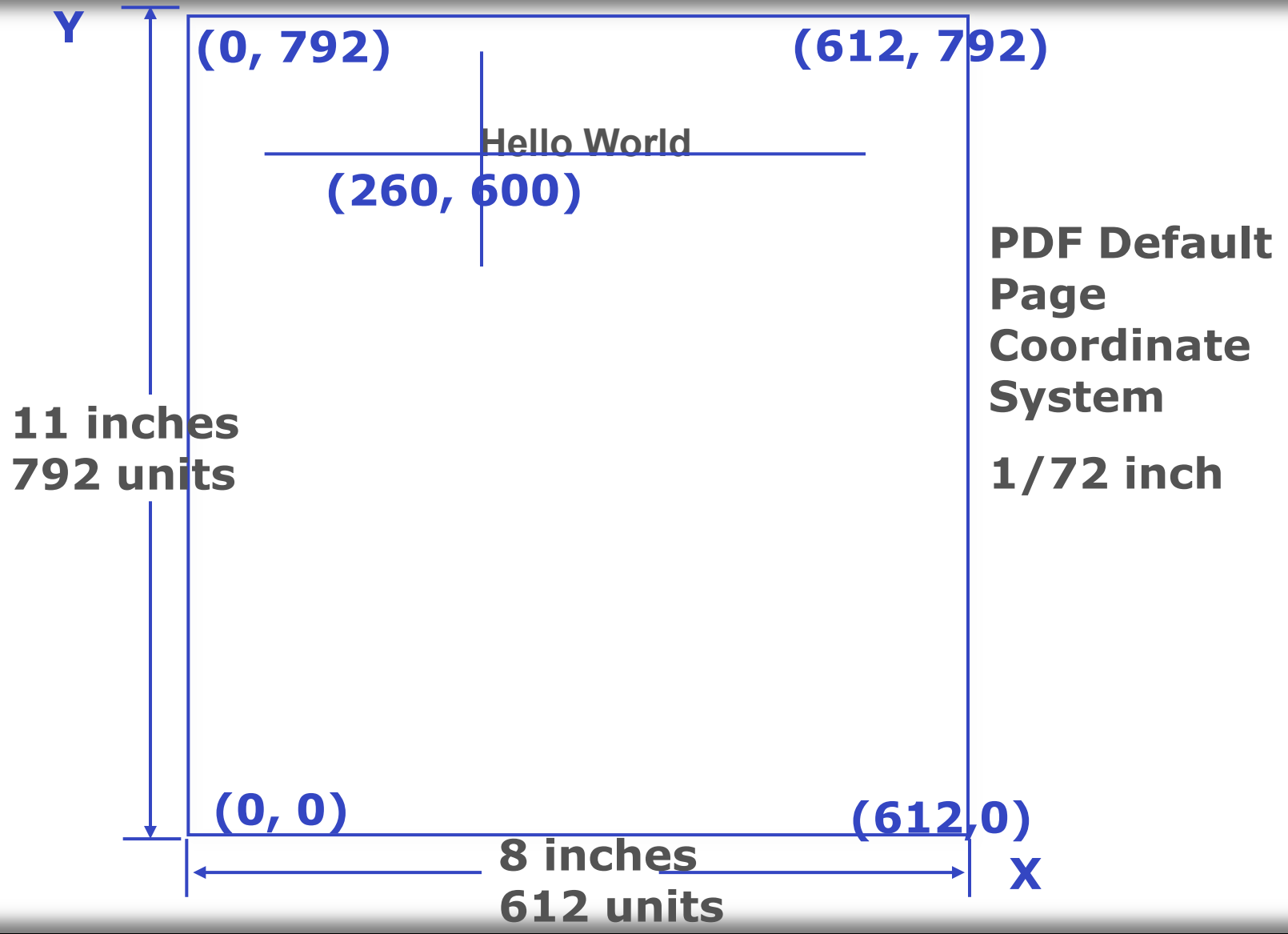
1. It is the belief of the Employer that the Employee possesses those skills, qualifications and abilities necessary to contribute to and further the aims of the Employer's business.

2. Both the Employee and Employer accept the terms and conditions set forth within this Agreement.

*IN CONSIDERATION OF the mutual benefits and responsibilities specified in this Agreement, the receipt and sufficiency of which consideration is hereby acknowledged, both parties to this Agreement agree as follows:*

Commencement Date and Term:

3. The Employee will commence full-time employment with the Employer on September 24th 2006 (the 'Commencement Date').



# Sample PDF File
## Composed from 6 objects

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

ex01.pdf

# Sample PDF File
## Composed from 6 objects

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

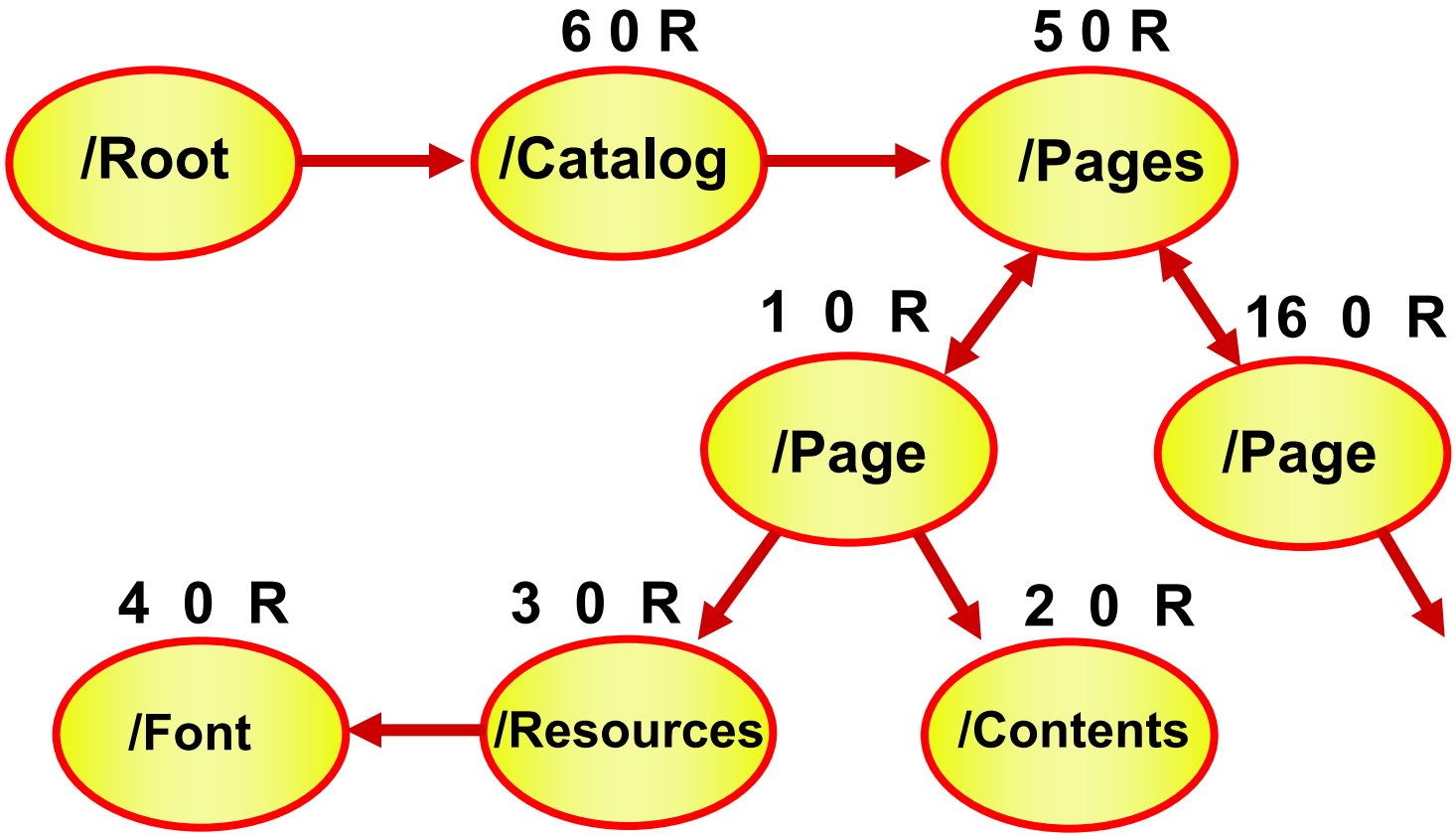
```
3 0 obj
<<
 /ProcSet[/PDF/Text]
 /Font <</F1 4 0 R >>
>>
endobj
xref
0 7
0000000000 65535 f
0000000060 00000 n
0000000228 00000 n
0000000424 00000 n
0000000145 00000 n
0000000333 00000 n
0000000009 00000 n
trailer
<<
  /Size 7
  /Root 6 0 R
>>
startxref
488
%%EOF
```

ex01.pdf

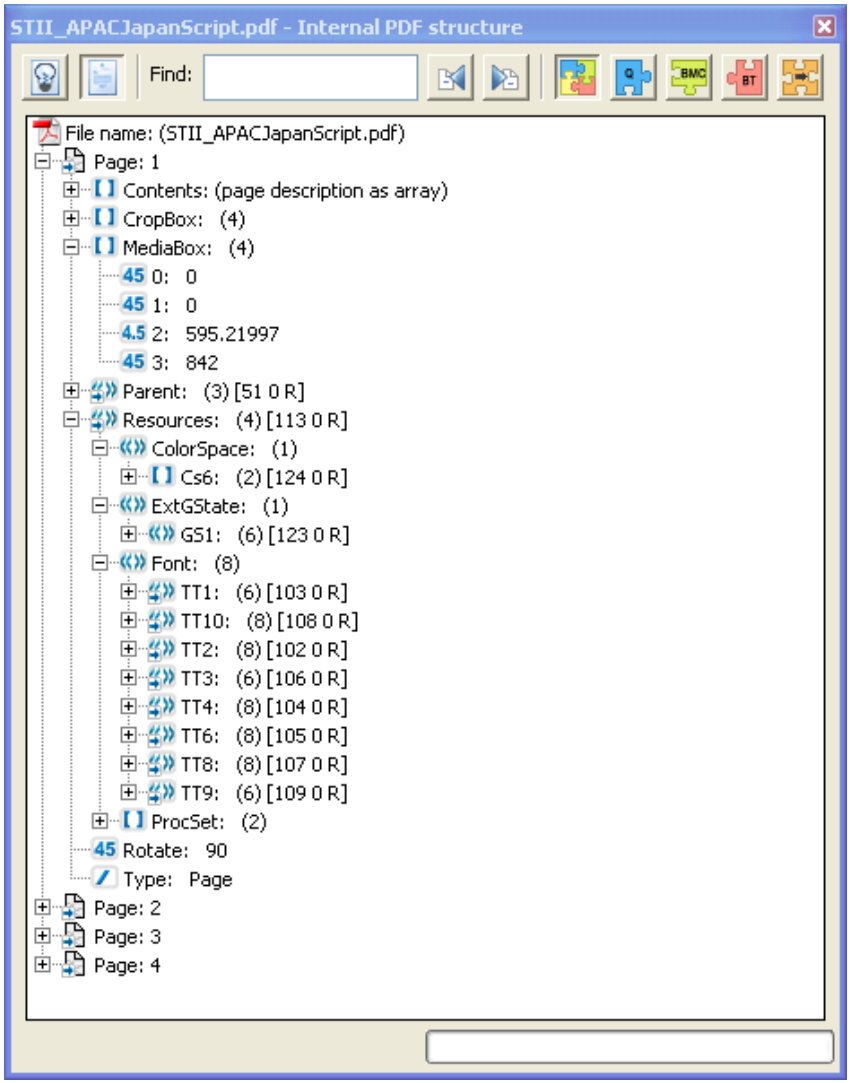
# How a conforming reader finds things

- Check the version
- Start at end of file
  - Look for the %EOF
- Walk backwards to locate "startxref"
  - Read the location
- Seek to that location
  - And read the xref
  - And the trailer!
- Trailer tells us where to find key objects
  - Info Dictionary
  - Root/Catalog Object

# Objects Form a Graph



# Detailed look at common PDF Objects



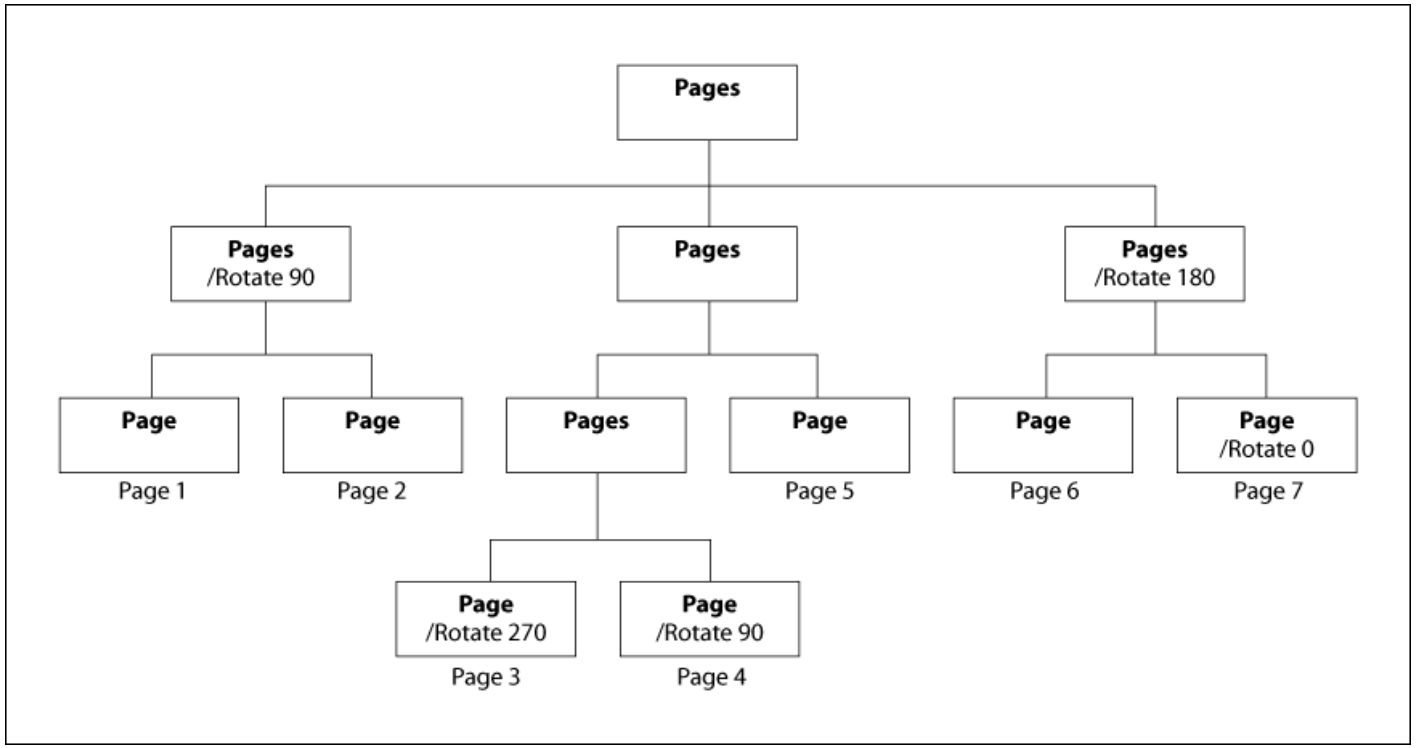
# Root

- Page tree
- XML-based Metadata
- Page labels
- AcroForms
- Document-level Actions
- Bookmark (outline) tree
- Structure Root
- Viewer preferences
  - show/hide toolbar
  - open mode
  - Display document title

```
<<
    /Type /Catalog
    /Pages 533 0 R
    /Metadata 537 0 R
    /PageLabels 531 0 R
    /PageLayout /SinglePage
    /OpenAction 540 0 R
    /AcroForm 541 0 R
    /Names 542 0 R
        /ViewerPreferences <<
        /DisplayDocTitle true >>
>>
```

# Page Tree

- Simple tree (supposedly balanced) with a node (array element) for Page object



# Page

- Includes references to all objects related to the page and required to display or process
  - "Boxes"
    - MediaBox, CropBox, etc.
    - Every page MUST have a MediaBox
      - This defines the physical size of the page (8.5x11, 11x17, etc.)
  - Contents
    - If you want something to appear on a page, you MUST have contents
  - Resources
    - If your contents refer to anything that is not in the stream itself, you need Resources to point to them.
  - Annotations
    - Standard (incl. AcroForm fields) and Custom
  - Page-level Actions
  - Slideshow-style transitions
  - Private application metadata
    - eg. Photoshop, Illustrator
  - And more…

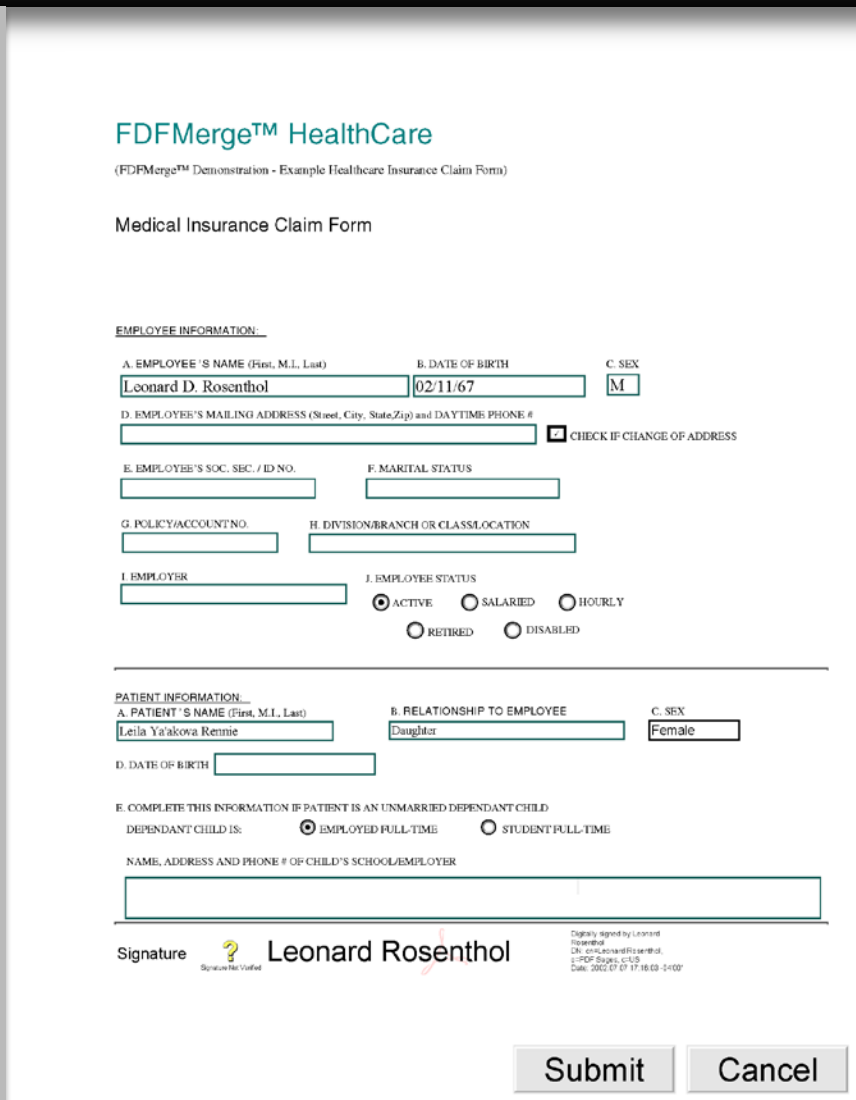
```
<<
    /Type /Page  /Parent 532 0 R
    /MediaBox [ 0 0 612 792 ]
    /CropBox [ 0 0 720 540 ]
    /Contents 564 0 R
    /Resources <<
        /ColorSpace << /CS0 562 0 R
        /CS1 563 0 R >>
        /Font << /T1_0 566 0 R >>
        /XObject << /Im0 577 0 R >>
        >>
    /Trans << /S /Dissolve >>
    /Rotate 0
    /Annots 549 0 R
    /AA << /C 578 0 R /O 579 0 R >>
>>
```

# Resources

- Any sort of object that is referenced by the content stream
  - Font
  - XObjects (Image, Form)
  - ColorSpaces (eg. Spots & Patterns)
  - Optional Content Properties
  - etc.
- Dictionary with keys for each type of resource
  - Values are always a dictionary mapping the names to the indirect objects
  - The name are totally arbitrary - just a mapping

```
/Resources <<
      /Font <<
            /F0 14 0 R
            /F1 17 0 R
            /F2 19 0 R >>
      /XObject <<
            /im0 8 0 R
            /im1 10 0 R
            /im2 11 0 R >>
      /Pattern <<
            /Pt0 7 0 R
            /Pt1 9 0 R >>
>>
```

# Example PDF Form



FDFMerge™ HealthCare

(FDFMerge™ Demonstration - Example Healthcare Insurance Claim Form)

Medical Insurance Claim Form

EMPLOYEE INFORMATION:

A. EMPLOYEE 'S NAME (First, M.I., Last): Leonard D. Rosenthol

B. DATE OF BIRTH: 02/11/67

C. SEX: M

D. EMPLOYEE'S MAILING ADDRESS (Street, City, State,Zip) and DAYTIME PHONE #

CHECK IF CHANGE OF ADDRESS

E. EMPLOYEE'S SOC. SEC. / ID NO.

F. MARITAL STATUS

G. POLICY/ACCOUNT NO.

H. DIVISION/BRANCH OR CLASS/LOCATION

I. EMPLOYER

J. EMPLOYEE STATUS

ACTIVE SALARIED HOURLY

RETIRED DISABLED

PATIENT INFORMATION:

A. PATIENT 'S NAME (First, M.I., Last): Leila Ya'akova Rennie

B. RELATIONSHIP TO EMPLOYEE: Daughter

C. SEX: Female

D. DATE OF BIRTH

E. COMPLETE THIS INFORMATION IF PATIENT IS AN UNMARRIED DEPENDANT CHILD

DEPENDANT CHILD IS: EMPLOYED FULL-TIME STUDENT FULL-TIME

NAME, ADDRESS AND PHONE # OF CHILD'S SCHOOL/EMPLOYER

Signature Leonard Rosenthol

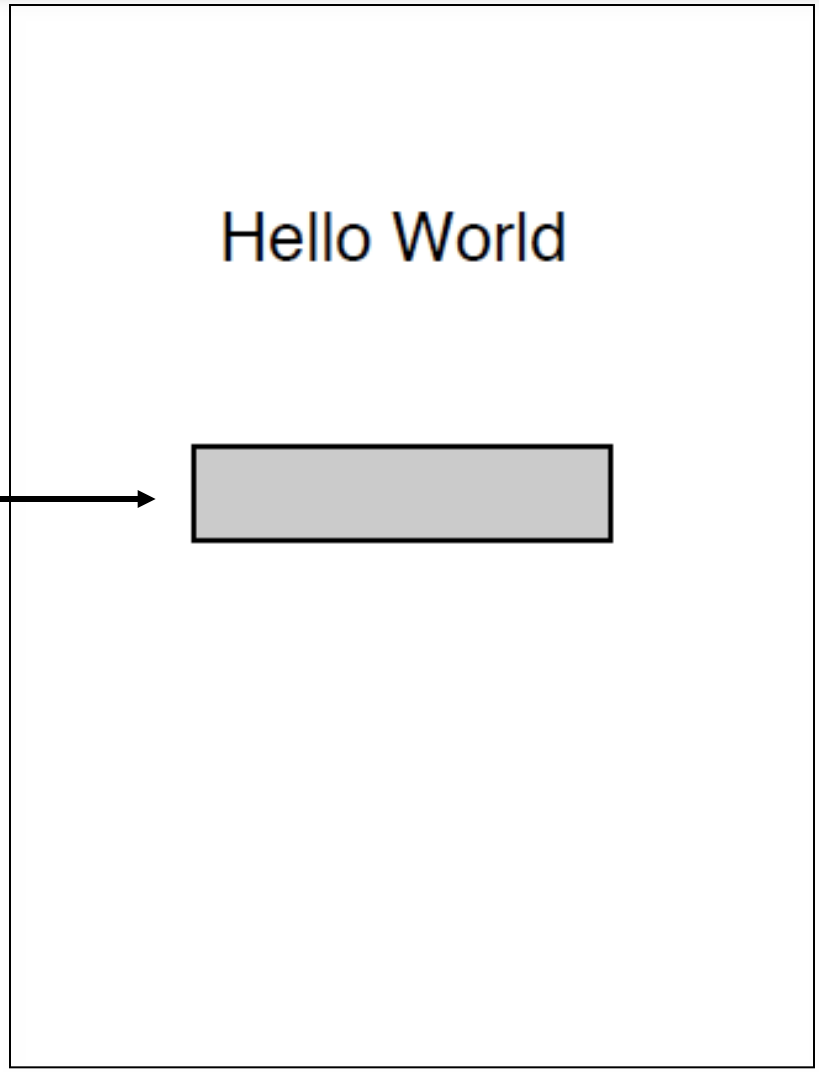
Submit Cancel





Adobe

# Example PDF Form File




ex02.pdf

# Sample PDF Form – One Text Field

```
%PDF-1.4
6 0 obj
<<
 /Type /Catalog
 /AcroForm
  <</Fields [8 0 R]>>
 /Pages 5 0 R
>>
endobj
5 0 obj
<<
 /Type/Pages /Count 1
 /Kids [ 1 0 R ]
 /MediaBox
  [ 0 0 612 792 ]
>>
endobj
3 0 obj
<<
 /ProcSet[/PDF/Text]
 /Font <</F1 4 0 R >>
>>
endobj
```

```
1 0 obj
<<
 /Type /Page
 /Parent 5 0 R
 /Resources 3 0 R
 /Contents 2 0 R
 /Annots [8 0 R]
>>
endobj

8 0 obj
<<
 /Type/Annot
 /Subtype/Widget
 /AP <</N 7 0 R>>
 /FT /Tx
 /Ff 2
 /Rect
  [250 500 400 535]
 /T (TextBox01)
 /V (<input here>)
>>
endobj
```

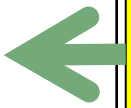
```
4 0 obj
<</Type /Font
 /Subtype /Type1
 /BaseFont/Helvetica
>>
endobj
2 0 obj
<</Length 50>>
stream
BT
 /F1 24 Tf
 1 0 0 1 260 600 Tm
 (Hello World)Tj
ET
endstream
endobj
```

An invisible form field might have a bounding box of [0 0 0 0] so that it wouldn't appear on the page. This is useful for hidden signatures!

ex02.pdf

# Sample PDF Form – One Text Field

```
7 0 obj
<<
 /Type /XObject
 /Subtype /Form
 /FormType 1
 /BBox
  [0.0 0.0
   150.0 32.0]
 /Matrix
  [1.0 0.0
   0.0 1.0
   0.0 0.0]
 /Length 61
>>
stream
 0.75 g
 0.0 0.0 150 32 re f
 0.00 G
 0.5 0.5 149 31 re s
endstream
endobj
```

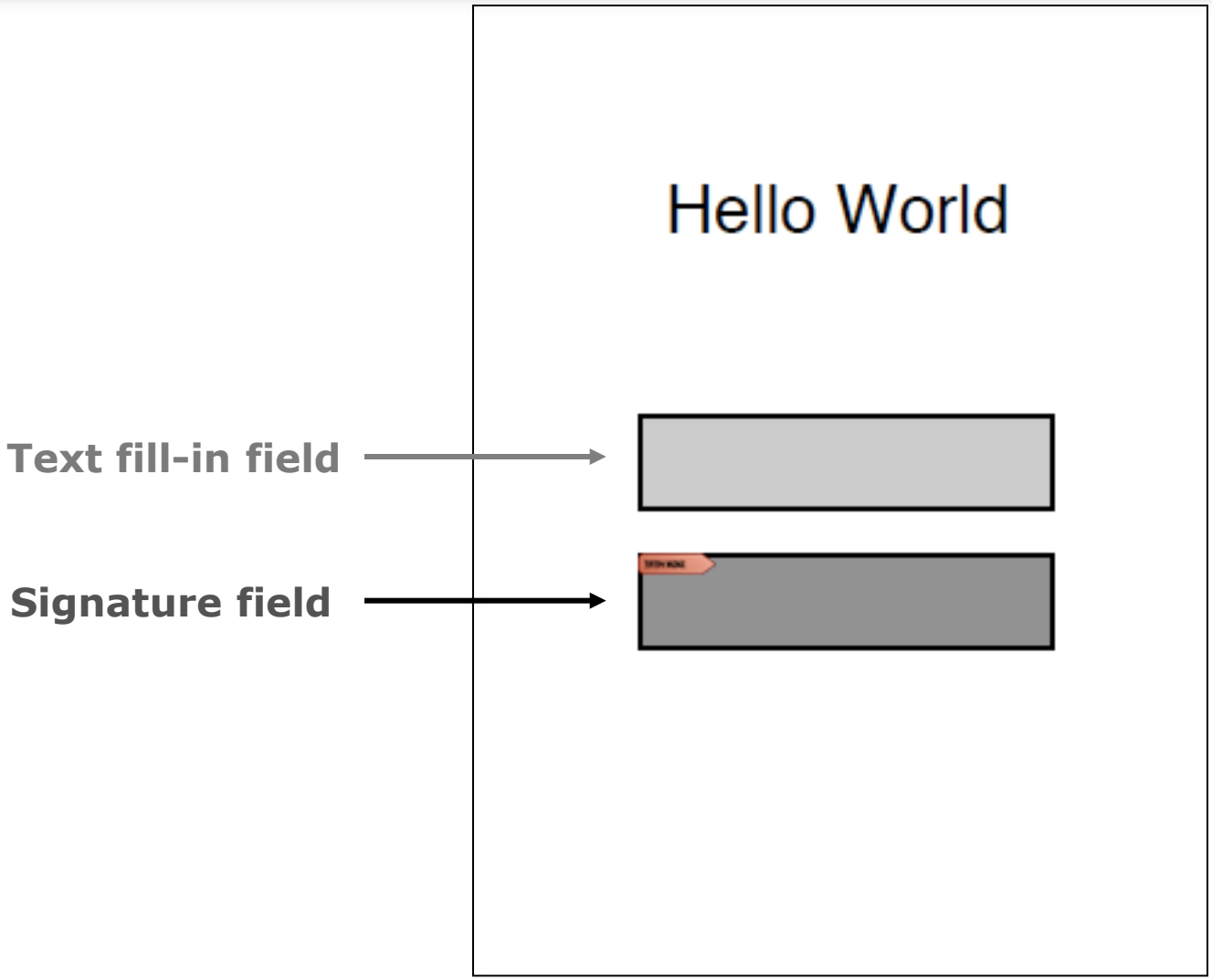


Every annotation (and therefore form field) should have an associated pre-rendered Appearance (/AP) – in fact, PDF/A requires them.

A conforming reader will use this appearance when it renders/draws the page – so that the presentation will always be the same.

```
xref
0 9
0000000000 65535 f
0000000091 00000 n
0000000425 00000 n
0000000617 00000 n
0000000202 00000 n
0000000526 00000 n
0000000009 00000 n
0000000688 00000 n
0000000288 00000 n
trailer
<<
  /Size 9
  /Root 6 0 R
>>
startxref
901
%%EOF
```

# PDF Form with Signature Field




ex03.pdf

# Sample PDF Form – One Text Field And One Signature Field

```
%PDF-1.4
6 0 obj
<<
 /Type /Catalog
 /AcroForm
  <</Fields [8 0 R
    10 0 R]>>
 /Pages 5 0 R
>>
endobj
1 0 obj
<<
 /Type /Page
 /Parent 5 0 R
 /Resources 3 0 R
 /Contents 2 0 R
 /Annots [8 0 R
    10 0 R]
>>
endobj
```

```
4 0 obj
<<
 /Type /Font
 /Subtype /Type1
 /Name /F1
 /BaseFont/Helvetica
>>
endobj
8 0 obj
<<
 /Type/Annot
 /Subtype/Widget
 /AP <</N 7 0 R>>
 /Ff 2
 /FT /Tx
 /Rect
  [250 500 400 535]
 /T (TextBox01)
 /V (<input here>)
>>
endobj
```

```
3 0 obj
<<
 /ProcSet[/PDF/Text]
 /Font <</F1 4 0 R >>
>>
endobj

10 0 obj
<<
 /Type/Annot
 /Subtype/Widget
 /AP <</N 9 0 R>>
 /Ff 2
 /FT /Sig
 /Rect
  [250 450 400 485]
 /T (SigField)
>>
endobj
```

ex03.pdf





Adobe

# Sample PDF Form – One Text Field And One Signature Field

```
2 0 obj
<<
 /Length 50
>>
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
 /MediaBox
  [ 0 0 612 792 ]
>>
endobj
```

```
7 0 obj
<<
 /Type /XObject
 /Subtype /Form
 /FormType 1
 /BBox [0 0 150 32]
 /Matrix[1 0 0 1 0 0]
 /Length 61
>>
stream
 0.75 g
 0.0 0.0 150 32 re f
 0.00 G
 0.5 0.5 149 31 re s
endstream
endobj
```

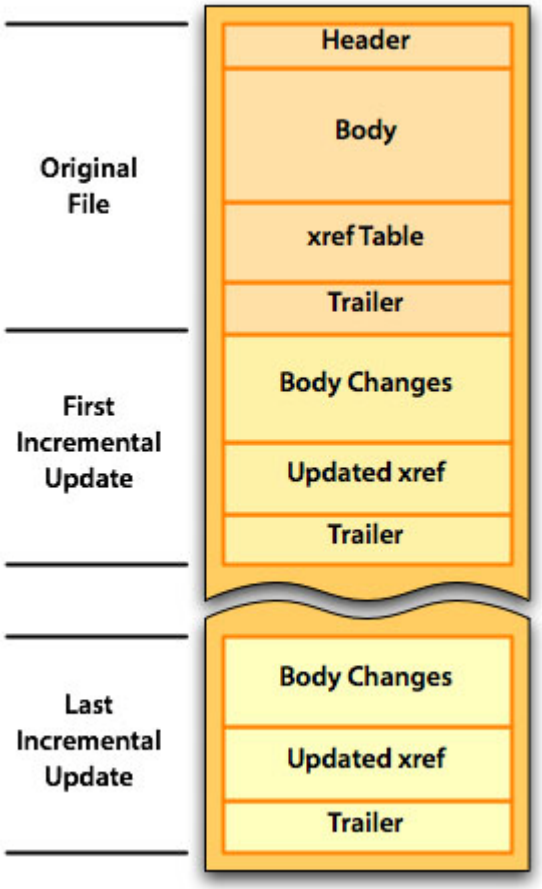
```
9 0 obj
<<
 /Type /XObject
 /Subtype /Form
 /FormType 1
 /BBox [0 0 150 32]
 /Matrix[1 0 0 1 0 0]
 /Length 61
>>
stream
 0.50 g
 0.0 0.0 150 32 re f
 0.00 G
 0.5 0.5 149 31 re s
endstream
endobj
```

# Sample PDF Form – One Text Field And One Signature Field

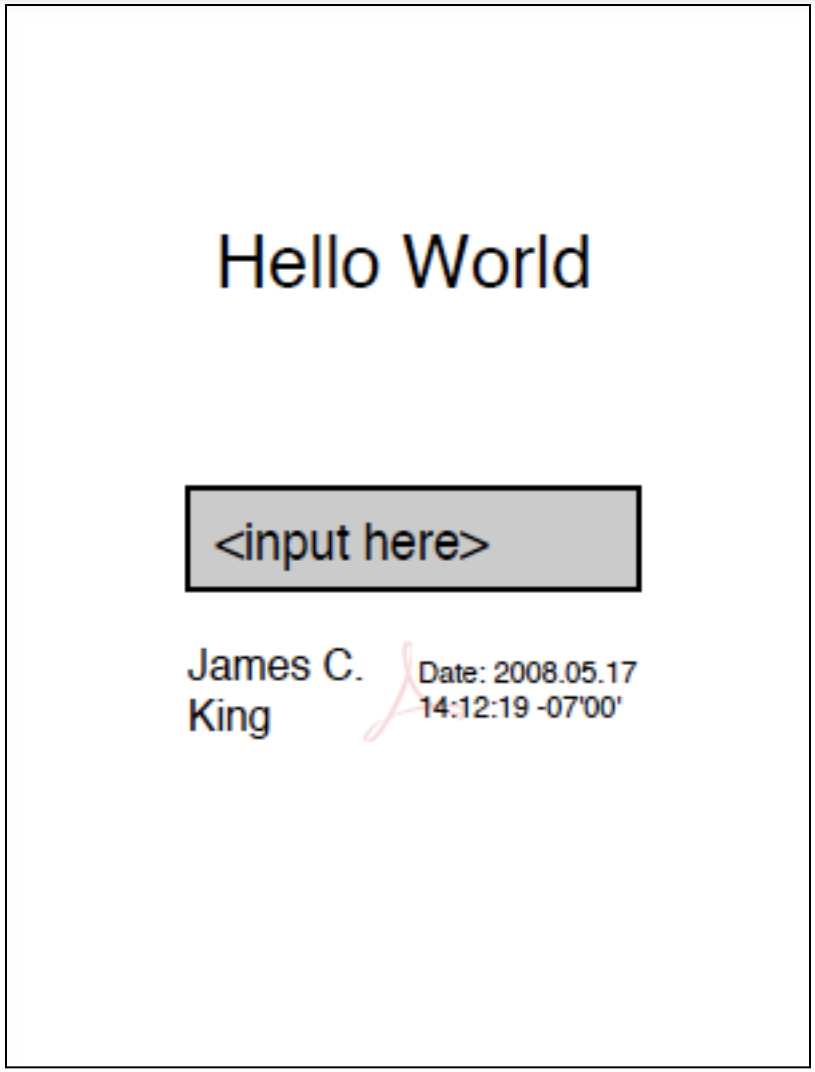
```
xref
0 11
0000000000 65535 f
0000000098 00000 n
0000000581 00000 n
0000000773 00000 n
0000000216 00000 n
0000000682 00000 n
0000000009 00000 n
0000000844 00000 n
0000000302 00000 n
0000001060 00000 n
0000000439 00000 n
trailer
<<
  /Size 11
  /Root 6 0 R
>>
startxref
1276
%%EOF
```

ex03.pdf

# Incremental Update of PDF files

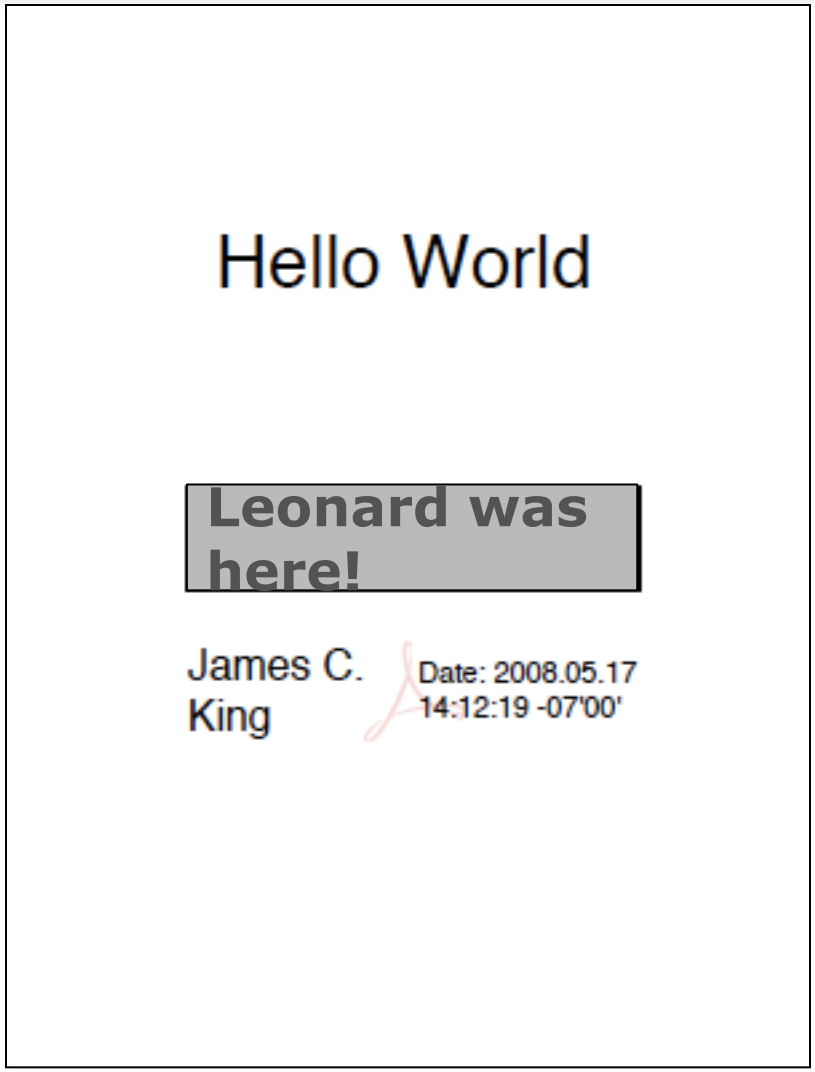


# Incremental Update




ex04.pdf

# For Example, Filling a form field




ex04.pdf

# Incremental Update

```
8 0 obj
<<
 /Type/Annot
 /Subtype/Widget
 /AP <</N 7 0 R>>
 /Ff 2
 /FT /Tx
 /Rect
  [250 500 400 535]
 /T (TextBox01)
 /V (<input here>)
>>
endobj
```

```
xref
0 12
0000000000 65535 f
0000000110 00000 n
0000000582 00000 n
0000000774 00000 n
0000000228 00000 n
0000000683 00000 n
0000000009 00000 n
0000006381 00000 n
0000000314 00000 n
0000006597 00000 n
0000000451 00000 n
0000000845 00000 n
trailer
<<
  /Size 12
  /Root 6 0 R
>>
startxref
6890
%%EOF
```

# Incremental Update

```
8 0 obj
<<
 /Type/Annot
 /Subtype/Widget
 /AP <</N 7 0 R>>
 /Ff 2
 /FT /Tx
 /Rect
  [250 500 400 535]
 /T (TextBox01)
 /V (<input here>)
>>
endobj
```

```
8 0 obj
<<
 /Type/Annot
 /Subtype/Widget
 /AP <</N 7 0 R>>
 /Ff 2
 /FT /Tx
 /Rect
  [250 500 400 535]
 /T (TextBox01)
 /V
  (Leonard was here)
>>
endobj
```

```
7 0 obj
<<
 /Type /XObject
 /Subtype /Form
 /FormType 1
 /BBox [0 0 150 32]
 /Matrix
       [1 0 0 1 0 0]
 /Length 61
>>
stream
 0.75 g
 0.0 0.0 150 32 re f
 0.00 G
 /F1 12 Tf
 (Leonard was
here)Tj
endstream
endobj
```

# Incremental Update

```
xref
0 12
0000000000 65535 f
0000000110 00000 n
0000000582 00000 n
0000000774 00000 n
0000000228 00000 n
0000000683 00000 n
0000000009 00000 n
0000006381 00000 n
0000000314 00000 n
0000006597 00000 n
0000000451 00000 n
0000000845 00000 n
trailer
<<
  /Size 12
  /Root 6 0 R
>>
startxref
6890
%%EOF
```

```
xref
0 1
0000000000 65535 f
7 1
0000007184 00000 n
0000007250 00000 n
trailer
<<
  /Size 12
  /Root 6 0 R
  /Prev 6886
>>
startxref
7474
%%EOF
```

# Wrap Up

- PDF is a rich container format
  - Static 2D vector, raster and text content
  - Multimedia & 3D
  - Metadata
  - Security, rights management and digital signatures
  - And more…


- It's also FULLY OPEN international standard (ISO 32000)
  - With various ISO subset standards for particular industries

- It's a structured binary format consisting of various types of objects that go together in a defined way but enable rich extensibility and version compatibility

# Questions