



Εργασία στα πλαίσια του μαθήματος επιλογής
του 8^{ου} εξαμήνου

**ΣΥΝΤΗΡΗΣΗ ΚΑΙ ΕΠΑΝΥΛΟΠΟΙΗΣΗ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

Καθηγητής:
Σπινέλλης Διομήδης

ΘΕΜΑ:

**ΚΑΤΑΝΟΗΣΗ – ΠΡΟΣΘΗΚΗ – ΟΛΟΚΛΗΡΩΣΗ
ΚΑΙ ΤΕΚΜΗΡΙΩΣΗ ΝΕΩΝ ΔΥΝΑΤΟΤΗΤΩΝ ΣΕ
ΛΟΓΙΣΜΙΚΟ ΑΝΟΙΧΤΟΥ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ**

ΕΡΓΟ ΑΝΟΙΧΤΟΥ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ:

PDFBox

(<http://sourceforge.net/projects/pdfbox>)



ΦΟΙΤΗΤΗΣ:

Τουμάσης Παναγιώτης (8000131)
(Phaethon)

ΟΚΤΩΒΡΙΟΣ 2004

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	2
1. ΕΙΣΑΓΩΓΗ	3
2. ΠΕΡΙΓΡΑΦΗ ΕΡΓΟΥ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ	3
2.1 ΓΕΝΙΚΑ	3
2.2 ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ ΕΡΓΟΥ	4
2.3 ΕΚΔΟΣΕΙΣ ΕΡΓΟΥ	5
2.4 ΠΑΚΕΤΑ ΤΕΛΕΥΤΑΙΑΣ ΕΚΔΟΣΗΣ	5
2.5 ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ JAVA ΤΕΛΕΥΤΑΙΑΣ ΕΚΔΟΣΗΣ	7
2.6 ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΕΛΕΥΤΑΙΑΣ ΕΚΔΟΣΗΣ	11
2.7 BUGS – PATCHES – FEATURE/SUPPORT REQUESTS	12
2.8 ΠΕΡΙΓΡΑΦΗ ΠΑΚΕΤΟΥ ORG.PDFBOX.PDMODEL	15
3. ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ	16
3.1 ΑΡΧΙΚΗ ΠΡΟΣΠΑΘΕΙΑ	16
3.2 ΤΕΛΙΚΗ ΑΠΟΦΑΣΗ	17
3.3 ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΤΡΟΠΟΠΟΙΗΣΗ ΚΛΑΣΕΩΝ	17
3.4 ΣΥΝΕΡΓΑΣΙΑ ΜΕ ΤΗΝ ΟΜΑΔΑ ΥΛΟΠΟΙΗΣΗΣ	19
4. ΣΥΜΠΕΡΑΣΜΑΤΑ	19
5. ΠΑΡΑΡΤΗΜΑ	21
5.1 ΑΔΕΙΑ ΧΡΗΣΗΣ PDFBOX: BSD LICENCE	21
5.2 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟ ΔΙΑΧΕΙΡΙΣΤΗ ΤΟΥ ΕΡΓΟΥ	22
5.3 ΤΡΟΠΟΠΟΙΗΜΕΝΕΣ ΚΑΙ ΝΕΕΣ ΚΛΑΣΕΙΣ JAVA	31
5.3.1 PDAction.java Class	31
5.3.2 PDAnnotationAdditionalActions.java Class	35
5.3.3 PDFFormFieldAdditionalActions.java Class	43
5.3.4 PDPAGEAdditionalActions.java Class	48
5.3.5 PDDocumentCatalogAdditionalActions.java Class	51
5.3.6 PDGoToAction.java Class	56
5.3.7 PDRemoteGoToAction.java Class	59
5.3.8 PDLaunchAction.java Class	64
5.3.9 PDThreadAction.java Class	70
5.3.10 PDURIAction.java Class	76
5.4 ΠΑΡΑΔΟΤΕΑ ΚΑΙ ΠΑΡΟΥΣΙΑΣΕΙΣ ΚΑΤΑ ΤΗ ΔΙΑΡΚΕΙΑ ΤΟΥ ΕΞΑΜΗΝΟΥ	80
5.4.1 Team Form	80
5.4.2 Project Form	81
5.4.3 Presentation 1: Project (21/4/2004)	82
5.4.4 Presentation 2: Design (10/5/2004)	86
5.4.5 Presentation 3: Implementation (8/10/2004)	92

1. ΕΙΣΑΓΩΓΗ

Η εργασία αυτή έχει ως στόχο την παρουσίαση του αρχιτεκτονικού ανασχεδιασμού μιας εφαρμογής λογισμικού ανοιχτού πηγαίου κώδικα. Συγκεκριμένα, περιλαμβάνονται η αναλυτική περιγραφή του επιλεγμένου έργου ανοιχτού πηγαίου κώδικα, ο τρόπος υλοποίησης, προσθήκης και ολοκλήρωσης των αλλαγών στον ήδη υπάρχοντα κώδικα και τα τελικά συμπεράσματα από την προσωπική εμπλοκή του γράφοντα στην προαναφερθείσα διαδικασία.

Επιπλέον, σε παράρτημα περιλαμβάνονται:

- το σύνολο τροποποιημένου και νέου κώδικα,
- η άδεια χρήσης του ανοιχτού πηγαίου κώδικα,
- η επικοινωνία –μέσω ηλεκτρονικού ταχυδρομείου– με τα υπόλοιπα μέλη της ομάδας υλοποίησης του έργου,
- οι φόρμες που έχουν παραδοθεί στο διδάσκοντα και
- οι παρουσιάσεις που έγιναν κατά τη διάρκεια του εξαμήνου.

Το κύριο πρόβλημα που ως φοιτητής κλήθηκα να αντιμετωπίσω στην παρούσα εργασία ήταν η πρωτότυπη, για τα δεδομένα που έως τότε είχα αντιμετωπίσει, μορφή της. Έπρεπε να επιλέξω ένα ήδη υπάρχον έργο και να συνεργαστώ μέσω του διαδικτύου με άτομα που μάλλον δε θα γνωρίσω ποτέ από κοντά. Επιπλέον, το ίδιο το ζητούμενο της εργασίας αποτελούσε μια πρόκληση, πολύ διαφορετική από το πνεύμα των υπόλοιπων - μάλλον πολυάριθμων και απαιτητικών - εργασιών που μέχρι εκείνη τη στιγμή είχα φέρει σε πέρας.

Συμπερασματικά όμως πιστεύω ότι το μάθημα έχει πετύχει το σκοπό του, όπως φαίνεται και από το αποτέλεσμα, αλλά και ότι επιτυγχάνει να κεντρίσει το ενδιαφέρον όσων φοιτητών το παρακολουθούν για την ανάπτυξη λογισμικού ανοιχτού πηγαίου κώδικα.

2. ΠΕΡΙΓΡΑΦΗ ΕΡΓΟΥ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ

2.1 ΓΕΝΙΚΑ

Το έργο ανοιχτού πηγαίου κώδικα στο οποίο θα αναφερθούμε έχει επιλεγεί από τον ιστότοπο <http://sourceforge.net>, ο οποίος αποτελεί μια διεθνή «πύλη» συγκέντρωσης και ανάπτυξης έργων τέτοιας μορφής, αλλά και συνεργασίας και ανταλλαγής απόψεων όσων ασχολούνται με την ανάπτυξη λογισμικού ανοιχτού πηγαίου κώδικα.

Η ονομασία του έργου που επιλέχτηκε είναι **PDFBox** και μπορεί κανείς να έχει άμεση πρόσβαση σε αυτό μέσω του διαδικτύου, πληκτρολογώντας την ακόλουθη διεύθυνση:

<http://sourceforge.net/projects/pdfbox>

Επιπλέον, το έργο έχει και δική του ανεξάρτητη ιστοσελίδα, την:

www.pdfbox.org

Το **PDFBox** είναι στην ουσία μια **ολοκληρωμένη βιβλιοθήκη σε γλώσσα Java για την επεξεργασία αρχείων μορφής PDF**. Παρέχει έτσι τη δυνατότητα σε όσους το χρησιμοποιούν να έχουν πρόσβαση σε όλα τα συστατικά μέρη ενός εγγράφου μορφής PDF. Περισσότερα στοιχεία χειρισμού θα προστεθούν σε επόμενες εκδόσεις του έργου, ενώ περιλαμβάνεται ήδη η δυνατότητα μετατροπής ενός εγγράφου PDF σε αρχείο κειμένου.

Ανήκει στην κατηγορία δυναμικού περιεχομένου (dynamic content), ευρετηρίων – αναζήτησης (indexing – search) και επεξεργασίας κειμένου (text processing). Το στάδιο ανάπτυξης του είναι το 4^ο (Beta), σύμφωνα με την κατηγοριοποίηση του Sourceforge.net. Η γλώσσα προγραμματισμού που έχει χρησιμοποιηθεί για την ανάπτυξή του είναι η Java, ενώ το έργο «τρέχει» ανεξάρτητα από την πλατφόρμα και το λειτουργικό σύστημα που χρησιμοποιούμε. Απευθύνεται σε όσους ασχολούνται με την ανάπτυξη λογισμικού (software developers) και η άδεια χρήσης στην οποία υπόκειται είναι η BSD License, μια άδεια χρήσης παρόμοια με τη γνωστότερη MIT License. Η άρτια και ολοκληρωμένη τεκμηρίωση του έργου έχει γίνει με το εργαλείο Javadoc.

Η ομάδα ανάπτυξης του έργου αποτελείται από δύο άτομα, τον Ben Litchfield (<http://sourceforge.net/users/benlitchfield/>) και τον Bob Dickinson (<http://sourceforge.net/users/leftbrute/>), οι οποίοι είναι και οι διαχειριστές του έργου, ενώ διάφορα άλλα μέλη της διαδικτυακής «πύλης» έχουν συμβάλει στην προσθήκη δυνατοτήτων και την αποσφαλμάτωση του κώδικα.

2.2 ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ ΕΡΓΟΥ

Οι **λόγοι** για τους οποίους επιλέχτηκε η ενασχόληση με το συγκεκριμένο έργο μεταξύ χιλιάδων άλλων που υπάρχουν στο διαδίκτυο είναι οι εξής:

- Μεγαλύτερη εξοικείωση με την ευρέως διαδεδομένη γλώσσα προγραμματισμού Java στην οποία είναι εξολοκλήρου γραμμένο.
- Σχετικά υψηλό στάδιο ανάπτυξης και ύπαρξη δοκιμασμένου και διορθωμένου κώδικα, ο οποίος χρησιμοποιείται ήδη, ώστε να μπορεί να μετασχηματιστεί εύκολα και να προστεθούν νέες λειτουργίες και δυνατότητες.
- Δυνατότητα χρήσης του συγκεκριμένου έργου σε πολλές διαφορετικές περιστάσεις και από πολλούς διαφορετικούς επιστημονικούς τομείς, εφόσον το αντικείμενο που πραγματεύεται (διαχείριση αρχείων PDF και επεξεργασία κειμένου) έχει εφαρμογές σε όλους τους χρήστες ηλεκτρονικών υπολογιστών και δεν αποτελεί κάποιο εξειδικευμένο εργαλείο ενός συγκεκριμένου επιστημονικού κλάδου.
- Δυνατότητα χρήσης του έργου σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα.
- Μικρός αριθμός των μελών της ομάδας ανάπτυξης λογισμικού, ώστε να επιτυγχάνεται καλύτερη συνεργασία.

- Γρήγορος ρυθμός ανάπτυξης του έργου και διαρκής ενασχόληση με αυτό από τα μέλη της ομάδας, όπως φαίνεται και από το στάδιο ανάπτυξής του αλλά και από τη συχνότητα των νέων εκδόσεων.
- Σχετικά μεγάλο μέγεθος του έργου ώστε να υπάρχει δυνατότητα αλλαγών, διορθώσεων και προσθήκης νέων λειτουργιών.
- Πολύ καλή τεκμηρίωση όλων των πακέτων, των κλάσεων και των μεθόδων που χρησιμοποιούνται (εργαλείο Javadoc)

2.3 ΕΚΔΟΣΕΙΣ ΕΡΓΟΥ

Το έργο καταχωρήθηκε στις 7/4/2003 και από τότε έως σήμερα έχουν δημοσιευτεί 6 εκδόσεις, οι οποίες παρουσιάζονται στον ακόλουθο πίνακα.

Όνομα έκδοσης	Ημερομηνία	Ώρα	Υπεύθυνος
PDFBox – 0.6.1	11/4/2003	7:19	Ben Litchfield
PDFBox – 0.6.2	18/4/2003	21:40	Bob Dickinson
PDFBox – 0.6.3	14/9/2003	6:57	Ben Litchfield
PDFBox – 0.6.4	1/11/2003	16:52	Ben Litchfield
PDFBox – 0.6.5	8/3/2004	19:28	Ben Litchfield
PDFBox – 0.6.6	20/7/2004	17:19	Ben Litchfield

Το **PDFBox** περιλαμβάνει 34 πακέτα της γλώσσας Java, 153 κλάσεις και περισσότερες από 10000 γραμμές κώδικα. Το σύνολο των πακέτων του έργου με σύντομη περιγραφή της λειτουργίας του παρατίθεται στον επόμενο πίνακα, ενώ μπορούμε να δούμε την ιεραρχία των κλάσεων στο δέντρο που υπάρχει στη συνέχεια.

2.4 ΠΑΚΕΤΑ ΤΕΛΕΥΤΑΙΑΣ ΕΚΔΟΣΗΣ

org.pdfbox	This package holds executable classes that interact with the PDFBox application.
org.pdfbox.afmparser	This package holds classes used to parse AFM(Adobe Font Metrics) files.
org.pdfbox.afmtypes	This package contains all the base types that can be found in an AFM file.
org.pdfbox.ant	ANT tasks that utilize PDFBox features can be found in this package.
org.pdfbox.cmapparser	This package holds classes that are necessary to parse cmap files.

<u>org.pdfbox.cmaptypes</u>	This package holds classes that are used to represent cmap files as java objects.
<u>org.pdfbox.cos</u>	These are the low level objects that make up a PDF document.
<u>org.pdfbox.encoding</u>	This package contains the implementations for all of the encodings that are used in PDF documents.
<u>org.pdfbox.encryption</u>	These classes deal with encryption algorithms that are used in the PDF Document.
<u>org.pdfbox.examples</u>	The packages in this package will show how to use the PDFBox API.
<u>org.pdfbox.examples.encryption</u>	These examples will show how to use the encryption features of PDFBox.
<u>org.pdfbox.examples.fdf</u>	These are examples that use the FDF features of a PDF document.
<u>org.pdfbox.examples.pdmodel</u>	These examples show how to use the classes in the PDModel package.
<u>org.pdfbox.examples.persistence</u>	These examples will show how to use the persistence features of the PDFBox.
<u>org.pdfbox.examples.signature</u>	These examples will show how to gain access to the PDF signature.
<u>org.pdfbox.exceptions</u>	This package is a place holder for exceptions that are used in the PDFBox project.
<u>org.pdfbox.filter</u>	This package will hold the PDFBox implementations of the filters that are used in PDF documents.
<u>org.pdfbox.io</u>	This package contains IO streams.
<u>org.pdfbox.pdfparser</u>	The pdfparser package contains classes to parse PDF documents and objects within the document.
<u>org.pdfbox.pdfviewer</u>	The pdfviewer package contains classes to graphically display information about a PDF document.
<u>org.pdfbox.pdfwriter</u>	This is the persistence layer used to write the PDFBox documents to a stream.
<u>org.pdfbox.pdmodel</u>	The PDModel package represents a high level API for creating and manipulating PDF documents.
<u>org.pdfbox.pdmodel.common</u>	High level PD classes that are used throughout several packages are placed in the PDModel common package.
<u>org.pdfbox.pdmodel.encryption</u>	The encryption package will handle the PDF document security handlers and the functionality of pluggable security handlers.
<u>org.pdfbox.pdmodel.font</u>	Classes to deal with font functionality in a PDF Document.

<u>org.pdfbox.pdmodel.graphics</u>	The PModel graphics package deals with graphics states, operations, and parameters within the PDF document.
<u>org.pdfbox.pdmodel.graphics.color</u>	This package deals with colors that are stored in a PDF document.
<u>org.pdfbox.pdmodel.graphics.image</u>	This package deals with images that are stored in a PDF document.
<u>org.pdfbox.pdmodel.interactive.annotation</u>	The annotation package contains classes that work with PDF annotation elements.
<u>org.pdfbox.pdmodel.interactive.form</u>	interactive annotations such as textfields and buttons.
<u>org.pdfbox.pdmodel.text</u>	The PModel text package deals with text states, operations, and parameters within the PDF document.
<u>org.pdfbox.persistence.util</u>	These are utilities used by the persistence layer.
<u>org.pdfbox.searchengine.lucene</u>	This package holds classes that are used to integrate the PDFBox project with lucene.
<u>org.pdfbox.util</u>	This package contains utility classes that are used by the PDFBox project.

2.5 ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ JAVA ΤΕΛΕΥΤΑΙΑΣ ΕΚΔΟΣΗΣ

- class java.lang.[Object](#)
 - class org.pdfbox.examples.[AbstractExample](#)
 - class org.pdfbox.examples.fdf.[ImportFDF](#)
 - class org.pdfbox.examples.fdf.[SetField](#)
 - class org.pdfbox.afmparser.[AFMParser](#)
 - class org.pdfbox.examples.persistence.[AppendAndFillDoc](#)
 - class org.pdfbox.examples.persistence.[AppendDoc](#)
 - class org.pdfbox.encryption.[ARCFour](#)
 - class org.pdfbox.pdfviewer.[ArrayEntry](#)
 - class org.pdfbox.filter.[ASCII85Filter](#) (implements org.pdfbox.filter.[Filter](#))
 - class org.pdfbox.filter.[ASCIHexFilter](#) (implements org.pdfbox.filter.[Filter](#))
 - class org.pdfbox.pdfparser.[BaseParser](#)
 - class org.pdfbox.pdfparser.[PDFParser](#)
 - class org.pdfbox.pdfparser.[PDFStreamParser](#)
 - class org.pdfbox.util.[BoundingBox](#)
 - class org.pdfbox.filter.[CCITTFaxDecodeFilter](#) (implements org.pdfbox.filter.[Filter](#))
 - class org.pdfbox.afmtypes.[CharMetric](#)
 - class org.pdfbox.cmaptypes.[CMap](#)
 - class org.pdfbox.cmapparser.[CMapParser](#)
 - class org.pdfbox.cmaptypes.[CodespaceRange](#)

- class java.awt.[Component](#) (implements java.awt.image.[ImageObserver](#), java.awt.[MenuContainer](#), java.io.[Serializable](#))
 - class java.awt.[Container](#)
 - class javax.swing.[JComponent](#) (implements java.io.[Serializable](#))
 - class javax.swing.[JLabel](#) (implements javax.accessibility.[Accessible](#), javax.swing.[SwingConstants](#))
 - class javax.swing.tree.[DefaultTreeCellRenderer](#) (implements javax.swing.tree.[TreeCellRenderer](#))
 - class org.pdfbox.pdfviewer.[PDFTreeCellRenderer](#)
 - class javax.swing.[JPanel](#) (implements javax.accessibility.[Accessible](#))
 - class org.pdfbox.pdfviewer.[PDFPagePanel](#)
 - class java.awt.[Window](#) (implements javax.accessibility.[Accessible](#))
 - class java.awt.[Frame](#) (implements java.awt.[MenuContainer](#))
 - class javax.swing.[JFrame](#) (implements javax.accessibility.[Accessible](#), javax.swing.[RootPaneContainer](#), javax.swing.[WindowConstants](#))
 - class org.pdfbox.[PDFViewer](#)
- class org.pdfbox.afmtypes.[Composite](#)
- class org.pdfbox.afmtypes.[CompositePart](#)
- class org.pdfbox.examples.persistence.[CopyDoc](#)
- class org.pdfbox.pdmodel.common.[COSArrayList](#) (implements java.util.[List](#))
- class org.pdfbox.cos.[COSBase](#)
 - class org.pdfbox.cos.[COSArray](#)
 - class org.pdfbox.cos.[COSBoolean](#)
 - class org.pdfbox.cos.[COSDictionary](#)
 - class org.pdfbox.cos.[COSDocument](#)
 - class org.pdfbox.cos.[COSName](#)
 - class org.pdfbox.cos.[COSNull](#)
 - class org.pdfbox.cos.[COSNumber](#)
 - class org.pdfbox.cos.[COSFloat](#)
 - class org.pdfbox.cos.[COSInteger](#)
 - class org.pdfbox.cos.[COSObject](#)
 - class org.pdfbox.cos.[COSStream](#)
 - class org.pdfbox.pdmodel.common.[COSStreamArray](#)
 - class org.pdfbox.cos.[COSString](#)
- class org.pdfbox.pdmodel.common.[COSDictionaryMap](#) (implements java.util.[Map](#))

- class org.pdfbox.persistence.util.[COSObjectKey](#)
- class org.pdfbox.pdfwriter.[COSWriter](#) (implements org.pdfbox.cos.[ICOSVisitor](#))
- class org.pdfbox.pdfwriter.[COSWriterXRefEntry](#) (implements java.lang.[Comparable](#))
- class org.pdfbox.examples.pdmodel.[CreateBlankPDF](#)
- class org.pdfbox.util.[DateConverter](#)
- class org.pdfbox.filter.[DCTFilter](#) (implements org.pdfbox.filter.[Filter](#))
- class org.pdfbox.encryption.[DecryptDocument](#)
- class org.pdfbox.examples.encryption.[DecryptPDFDocument](#)
- class org.pdfbox.encoding.[Encoding](#)
 - class org.pdfbox.encoding.[AFMEncoding](#)
 - class org.pdfbox.encoding.[DictionaryEncoding](#)
 - class org.pdfbox.encoding.[MacRomanEncoding](#)
 - class org.pdfbox.encoding.[PdfDocEncoding](#)
 - class org.pdfbox.encoding.[StandardEncoding](#)
 - class org.pdfbox.encoding.[WinAnsiEncoding](#)
- class org.pdfbox.encoding.[EncodingManager](#)
- class org.pdfbox.[ExtractText](#)
- class org.pdfbox.examples.persistence.[FieldsDoc](#)
- class javax.swing.filechooser.[FileFilter](#)
 - class org.pdfbox.util.[DefaultFileFilter](#)
- class org.pdfbox.filter.[FilterManager](#)
- class org.pdfbox.filter.[FlateFilter](#) (implements org.pdfbox.filter.[Filter](#))
- class org.pdfbox.afmtypes.[FontMetric](#)
- class org.pdfbox.examples.persistence.[HelloWorldDoc](#)
- class org.pdfbox.searchengine.lucene.[IndexFiles](#)
- class java.io.[InputStream](#)
 - class java.io.[FilterInputStream](#)
 - class org.pdfbox.io.[ASCII85InputStream](#)
 - class java.io.[PushbackInputStream](#)
 - class org.pdfbox.io.[PushBackInputStream](#)
 - class org.pdfbox.io.[RandomAccessFileInputStream](#)
- class org.pdfbox.afmtypes.[KernPair](#)
- class org.pdfbox.afmtypes.[Ligature](#)
- class org.pdfbox.searchengine.lucene.[LucenePDFDocument](#)
- class org.pdfbox.filter.[LZWFilter](#) (implements org.pdfbox.filter.[Filter](#))
- class org.pdfbox.pdfviewer.[MapEntry](#)
- class org.pdfbox.util.[Matrix](#) (implements java.lang.[Cloneable](#))
- class org.pdfbox.io.[NBitInputStream](#)
- class org.pdfbox.io.[NBitOutputStream](#)
- class java.io.[OutputStream](#)
 - class java.io.[ByteArrayOutputStream](#)
 - class org.pdfbox.io.[FastByteArrayOutputStream](#)
 - class java.io.[FilterOutputStream](#)
 - class org.pdfbox.io.[ASCII85OutputStream](#)
 - class org.pdfbox.pdfwriter.[COSEscapedOutputStream](#)

- class
 - org.pdfbox.pdfwriter.[COSStandardOutputStream](#)
 - class org.pdfbox.io.[RandomAccessFileOutputStream](#)
- class org.pdfbox.[Overlay](#)
- class org.pdfbox.pdfviewer.[PageWrapper](#)
- class org.pdfbox.pdmodel.interactive.form.[PDAcroForm](#)
- class org.pdfbox.pdmodel.interactive.annotation.[PDAnnotation](#)
(implements org.pdfbox.pdmodel.common.[COSObjectable](#))
 - class org.pdfbox.pdmodel.interactive.annotation.[PDWidget](#)
- class org.pdfbox.pdmodel.interactive.form.[PDAppearance](#)
- class
 - org.pdfbox.pdmodel.interactive.annotation.[PDAppearanceDictionary](#)
(implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class
 - org.pdfbox.pdmodel.interactive.annotation.[PDAppearanceStream](#)
(implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class org.pdfbox.pdmodel.graphics.color.[PDCColorSpace](#)
- class org.pdfbox.pdmodel.[PDDocument](#)
- class org.pdfbox.pdmodel.[PDDocumentCatalog](#)
- class org.pdfbox.pdmodel.[PDDocumentInformation](#)
- class org.pdfbox.pdmodel.encryption.[PDEncryptionDictionary](#)
 - class org.pdfbox.pdmodel.encryption.[PDStandardEncryption](#)
- class org.pdfbox.pdmodel.encryption.[PDEncryptionManager](#)
- class org.pdfbox.pdmodel.graphics.[PDExtendedGraphicsState](#)
(implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class org.pdfbox.encryption.[PDFEncryption](#)
- class org.pdfbox.pdmodel.interactive.form.[PDFField](#) (implements org.pdfbox.pdmodel.common.[COSObjectable](#))
 - class org.pdfbox.pdmodel.interactive.form.[PDChoiceButton](#)
 - class
 - org.pdfbox.pdmodel.interactive.form.[PDCheckbox](#)
 - class
 - org.pdfbox.pdmodel.interactive.form.[PDRadioButton](#)
 - class org.pdfbox.pdmodel.interactive.form.[PDChoiceField](#)
 - class org.pdfbox.pdmodel.interactive.form.[PDPushButton](#)
 - class org.pdfbox.pdmodel.interactive.form.[PDSignature](#)
 - class org.pdfbox.pdmodel.interactive.form.[PDTextbox](#)
- class org.pdfbox.pdmodel.interactive.form.[PDFFieldFactory](#)
- class org.pdfbox.pdmodel.font.[PDFont](#) (implements org.pdfbox.pdmodel.common.[COSObjectable](#))
 - class org.pdfbox.pdmodel.font.[PDCIDFont](#)
 - class org.pdfbox.pdmodel.font.[PDCIDFontType0Font](#)
 - class org.pdfbox.pdmodel.font.[PDCIDFontType2Font](#)
 - class org.pdfbox.pdmodel.font.[PDSimpleFont](#)
 - class org.pdfbox.pdmodel.font.[PDMMType1Font](#)
 - class org.pdfbox.pdmodel.font.[PDTrueTypeFont](#)
 - class org.pdfbox.pdmodel.font.[PDType1Font](#)
 - class org.pdfbox.pdmodel.font.[PDType3Font](#)
 - class org.pdfbox.pdmodel.font.[PDType0Font](#)

- class org.pdfbox.pdmodel.font.[PDFFontFactory](#)
- class org.pdfbox.pdmodel.graphics.[PDFFontSetting](#) (implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class org.pdfbox.util.[PDFOperator](#)
- class org.pdfbox.[PDFSplit](#)
- class org.pdfbox.util.[PDFStreamEngine](#)
 - class org.pdfbox.pdfviewer.[PageDrawer](#)
 - class org.pdfbox.util.[PDFTextStripper](#)
 - class org.pdfbox.pdmodel.font.[Type3StreamParser](#)
- class org.pdfbox.pdfviewer.[PDFTreeModel](#) (implements javax.swing.tree.[TreeModel](#))
- class org.pdfbox.pdfparser.[PDFXref](#)
- class org.pdfbox.pdmodel.graphics.[PDGraphicsState](#) (implements java.lang.[Cloneable](#))
- class org.pdfbox.pdmodel.graphics.image.[PDInlinedImage](#)
- class org.pdfbox.pdmodel.graphics.[PDLineDashPattern](#) (implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class org.pdfbox.pdmodel.[PDPage](#) (implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class org.pdfbox.pdmodel.[PDPageNode](#) (implements org.pdfbox.pdmodel.common.[COSObjectable](#))
- class org.pdfbox.pdmodel.interactive.form.[PDRadioButton](#)
- class org.pdfbox.pdmodel.common.[PDRectangle](#)
- class org.pdfbox.pdmodel.[PDResources](#)
- class org.pdfbox.pdmodel.text.[PDTextState](#) (implements java.lang.[Cloneable](#))
- class org.pdfbox.examples.pdmodel.[PrintDocumentMetaData](#)
- class org.pdfbox.examples.fdf.[PrintFields](#)
- class org.apache.tools.ant.ProjectComponent
 - class org.apache.tools.ant.Task
 - class org.pdfbox.ant.[PDFToTextTask](#)
- class org.pdfbox.util.[ResourceLoader](#)
- class org.pdfbox.examples.signature.[ShowSignature](#)
- class org.pdfbox.util.[SimpleConfigurator](#) (implements org.apache.log4j.spi.Configurator)
- class org.pdfbox.util.[Splitter](#)
- class org.pdfbox.util.[TextPosition](#)
- class java.lang.[Throwable](#) (implements java.io.[Serializable](#))
 - class java.lang.[Exception](#)
 - class org.pdfbox.exceptions.[COSVisitorException](#)
 - class org.pdfbox.exceptions.[CryptographyException](#)
 - class org.pdfbox.exceptions.[InvalidPasswordException](#)
- class org.pdfbox.afmtypes.[TrackKern](#)
- class org.pdfbox.examples.persistence.[WriteDecodedDoc](#)

2.6 ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΕΛΕΥΤΑΙΑΣ ΕΚΔΟΣΗΣ

Στην τελευταία έκδοση του έργου οι κύριες λειτουργίες που επιτελούνται είναι:

- Η μετατροπή ενός εγγράφου μορφής PDF σε κείμενο μορφής .txt.
- Η συγχώνευση εγγράφων μορφής PDF.
- Η αποκρυπτογράφηση εγγράφων μορφής PDF.
- Η κρυπτογράφηση εγγράφων μορφής PDF.
- Η ολοκλήρωση με τη μηχανή αναζήτησης Lucene.
- Δεδομένα συμπλήρωσης της μορφής εγγράφων PDF.

Μέχρι στιγμής έχουν τεθεί σε εφαρμογή τα ακόλουθα φίλτρα, με τη χρήση των οποίων τα έγγραφα PDF μπορούν να αποκρυπτογραφηθούν:

- FlateDecode
- ASCIIHexDecode
- ASCII85Decode
- LZWDecode

Στο μέλλον πρόκειται επίσης να τεθούν σε εφαρμογή τα φίλτρα που αναφέρονται στη συνέχεια:

- RunLengthDecode
- CCITTFaxDecode
- JBIG2Decode
- DCTDecode

Στο σημείο αυτό πρέπει να σημειωθεί ότι δεν υπάρχει κάποιο δομημένο περιβάλλον εργασίας, δηλαδή κάποιο συγκεκριμένο παράθυρο στο οποίο να ενσωματώνεται το σύνολο των δυνατοτήτων και των λειτουργιών που παρέχει το έργο με τη μορφή έτοιμων εντολών, σε μορφή κουμπιών για παράδειγμα. Αντίθετα, το σύνολο των λειτουργιών απαιτεί τη χρήση της γραμμής εντολών για να υλοποιηθεί, γεγονός που αποτελεί και ένα αρκετά σημαντικό πρόβλημα για κάποιον ελάχιστα εξοικειωμένο με παρόμοια έργα.

2.7 BUGS – PATCHES – FEATURE/SUPPORT REQUESTS

Αυτή τη στιγμή έχουν εντοπιστεί συνολικά 104 σφάλματα (bugs), από τα οποία τα 29 είναι ακόμα ανοιχτά. Επίσης έχουν γίνει 4 συνολικά επιδιορθώσεις (patches) στον κώδικα, ενώ από τις 35 νέες δυνατότητες (feature requests) που έχουν ζητηθεί, μόνο οι 6 έχουν υλοποιηθεί. Τέλος έχει ζητηθεί συνολικά 6 φορές υποστήριξη (support requests) σχετικά με συγκεκριμένες λειτουργίες της εφαρμογής, ενώ μέχρι στιγμής εκκρεμούν 3 τέτοια αιτήματα. Όσα προαναφέρθηκαν παρουσιάζονται στους πίνακες που ακολουθούν.

A/A	Συνοπτική περιγραφή σφάλματος	Ημ/νια υποβολής	Ανάθεση σε	Υποβολή από
1033455	Wrong text formatting after parsing Landscape-formated .pdf	2004-09-23 08:24	benlitchfield	maxim72
1029873	Parse error, xref where startxref expected	2004-09-17 06:35	benlitchfield	thansson
1026366	pdfbox stops extracting text when encoutering a bullet (149)	2004-09-11 08:29	benlitchfield	nobody
1025060	PDFTextStripper extracting garbage	2004-09-09 05:13	benlitchfield	ralphscheuer
1024174	java.io.IOException: Error: Expected an integer type, actual	2004-09-07 22:53	benlitchfield	nobody
1023460	when I parse I am not getting the correct content	2004-09-06 23:03	benlitchfield	nobody
1022127	Problem decrypting documents with v6 security	2004-09-03 15:19	benlitchfield	ng_aldrige
1021691	IOException	2004-09-03 02:09	benlitchfield	nobody
1020203	Possible to Extractact Just Table From PDF	2004-08-31 23:23	benlitchfield	nobody
1017697	Problem with reading pdf	2004-08-27 09:01	benlitchfield	martinbucht
1017175	Large number converted when re-written	2004-08-26 13:44	benlitchfield	nobody
1010849	SetField always throws NullPointerException	2004-08-17 09:23	benlitchfield	steveholt
1006832	PDFParser invoking COSObject.setObject with a COSStream	2004-08-10 13:03	benlitchfield	nobody
1003188	Extracting text from pdf containg text under the images	2004-08-04 04:33	benlitchfield	nobody
1003149	Extracting images places when extracting the text	2004-08-04 03:02	benlitchfield	nobody
1000165	German umlauts are not extracted correctly	2004-07-29 08:28	benlitchfield	b4market
999071	ClassCastEx PDAppearanceDictionary::getNormalAppearance	2004-07-27 16:26	benlitchfield	snsnor
996050	Class Cast exception when importing	2004-07-22 10:24	benlitchfield	nobody
995914	NPE is isButton for org.pdfbox.examples.fdf.PrintFields	2004-07-22 06:20	benlitchfield	benlitchfield
977267	Unable to extract OpenOffice pdf export	2004-06-22 00:35	benlitchfield	nobody
974661	getKids() Null Pointer Exception when parsing pdf	2004-06-17 04:04	benlitchfield	sbelt
955043	Extracting from pdf with Chinese characters	2004-05-16 18:14	benlitchfield	nobody
919215	IOException in LucenePDFDocument.getDocument	2004-03-18 17:40	benlitchfield	nobody

907432	Exception While trying to extract pdf with Japanese fonts	2004-03-01 02:21	benlitchfield ✍	nobody
878903	Extracting special non-english-alphabet characters	2004-01-17 08:56	benlitchfield ✍	nobody
875685	Type 3 Font Extracts As Garbage	2004-01-12 13:24	benlitchfield ✍	tadwoods
840204	TextPosition data wrong when orientation is landscape	2003-11-11 11:34	benlitchfield ✍	tadwoods
835290	EOFException reading InflaterInputStream	2003-11-03 13:04	benlitchfield ✍	tadwoods
830512	Poor text extraction for excel converter PDF	2003-10-26 06:42	benlitchfield ✍	benlitchfield ✍

A/A	Συνοπτική περιγραφή νέας δυνατότητας	Ημ/νια υποβολής	Ανάθεση σε	Υποβολή από
1033054	Linearize command line tool	2004-09-22 17:42	nobody	benlitchfield ✍
1023133	Support PDF Functions	2004-09-06 08:24	benlitchfield ✍	benlitchfield ✍
1021241	Text extraction should follow PDF article divisions	2004-09-02 08:19	benlitchfield ✍	jkronegg
1019898	Convert FDF to XFDF and vice versa	2004-08-31 11:15	benlitchfield ✍	benlitchfield ✍
1016101	Replace existing font with new font	2004-08-25 09:16	benlitchfield ✍	benlitchfield ✍
1016097	Font Embedding	2004-08-25 09:13	benlitchfield ✍	benlitchfield ✍
1010721	Add ability to extract comments	2004-08-17 06:24	benlitchfield ✍	benlitchfield ✍
1000668	Add ability to set document information using PDFViewer	2004-07-30 05:22	benlitchfield ✍	benlitchfield ✍
1000109	Add ability to digitally sign a PDF	2004-07-29 06:27	benlitchfield ✍	benlitchfield ✍
939868	text from box	2004-04-22 01:19	benlitchfield ✍	ramyamd
931628	CID to Unicode mapping	2004-04-08 02:45	benlitchfield ✍	vadimbit
912459	Set label on PDPushButton	2004-03-08 19:25	benlitchfield ✍	benlitchfield ✍
865758	Support for Internet Printing Protocol?	2003-12-25 12:25	benlitchfield ✍	elfring
841169	HTML -> PDF	2003-11-12 20:03	benlitchfield ✍	nobody
838444	Color Space support	2003-11-08 07:34	benlitchfield ✍	benlitchfield ✍
830521	alternate header form	2003-10-26 07:13	benlitchfield ✍	benlitchfield ✍
830508	Handle new PDF 1.5 cross reference	2003-10-26	benlitchfield	benlitchfield

	streams	06:35		
830505	Handle new PDF 1.5 object streams	2003-10-26 06:34	benlitchfield 	benlitchfield
807179	XFDF Import/Export	2003-09-16 07:57	benlitchfield 	benlitchfield
805623	extract information from tagged PDF	2003-09-13 07:38	benlitchfield 	benlitchfield
805134	extract TOC/bookmarks	2003-09-12 08:00	benlitchfield 	benlitchfield
804658	Extract Images	2003-09-11 12:18	benlitchfield 	benlitchfield
804468	FDF export	2003-09-11 07:38	benlitchfield 	benlitchfield
802407	PDF to HTML conversion	2003-09-08 04:42	benlitchfield 	winstanley_john \$
794289	Notification about output progress	2003-08-24 12:50	benlitchfield 	elfring
771559	Support for PDF/X-3	2003-07-15 05:23	benlitchfield 	elfring
770905	cooperation with other PDF-APIs and applications	2003-07-14 05:16	benlitchfield 	elfring
765686	CJK decoding	2003-07-03 17:57	benlitchfield 	bguan
765682	Font Info and X, Y positioning	2003-07-03 17:52	benlitchfield 	bguan

A/A	Συνοπτική περιγραφή αιτήματος υποστήριξης	Ημ/νια υποβολής	Ανάθεση σε	Υποβολή από
1033457	Make Splitter private attribute protected	2004-09-23 08:26	nobody	pczeus
1033282	When can we expect new stable version 0.6.7?	2004-09-23 03:48	nobody	nobody
868449	loss of access to fields after merging forms	2003-12-31 06:33	nobody	danlee6

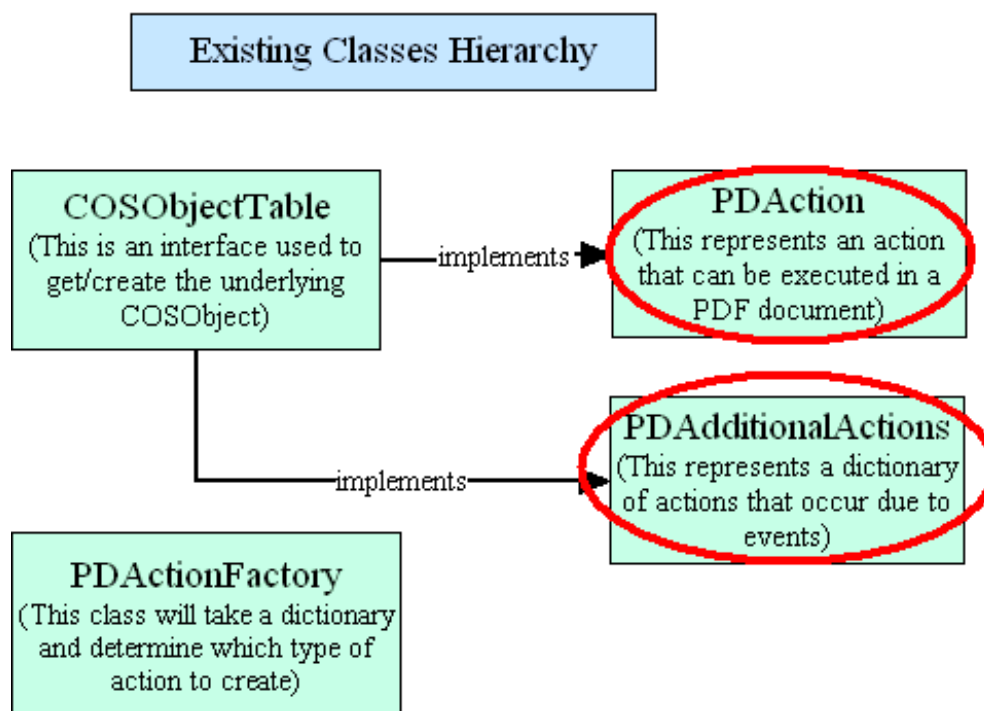
2.8 ΠΕΡΙΓΡΑΦΗ ΠΑΚΕΤΟΥ ORG.PDFBOX.PDMODEL

Το πακέτο org.pdfbox.pdmodel είναι το πακέτο στο οποίο ανήκουν οι κλάσεις που θα τροποποιήσουμε και στο οποίο θα προσθέσουμε τις νέες κλάσεις που θα κατασκευάσουμε. Επίσης, στο πακέτο αυτό υπάρχουν υποδείγματα κλάσεων, τα οποία θα μας βοηθήσουν στην τροποποίηση των υπαρχόντων και στη δημιουργία των νέων κλάσεων.

Στην έκδοση που θα ακολουθήσει σύντομα, θα συμπεριληφθούν πολλές νέες κλάσεις στο συγκεκριμένο πακέτο. Έτσι, σύμφωνα με το καθημερινά τροποποιημένο CVS, στις 02/10/2004 το πακέτο αυτό περιλάμβανε 105 κλάσεις, δηλαδή ήδη 56 κλάσεις περισσότερες από αυτές που έχουν δημοσιευτεί στην τελευταία έκδοση. Όπως σημειώνεται, οι

περισσότερες από τις κλάσεις που υπήρχαν στην τελευταία έκδοση είχαν δηλωθεί και περιγραφεί αλλά δεν είχαν υλοποιηθεί.

Συγκεκριμένα, το `org.pdfbox.pdmodel` αποτελεί ένα API υψηλού επιπέδου, για τη δημιουργία και το χειρισμό εγγράφων της μορφής PDF. Το υποπακέτο που θα μας απασχολήσει είναι το `org.pdfbox.pdmodel.interactive.action`, του οποίου η ιεραρχία φαίνεται στο επόμενο διάγραμμα.



3. ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

3.1 ΑΡΧΙΚΗ ΠΡΟΣΠΑΘΕΙΑ

Ύστερα από αρκετές ώρες σκέψης και αναζήτησης πιθανών αλλαγών που θα μπορούσαν να υλοποιηθούν στον ήδη υπάρχοντα κώδικα και να αποτελέσουν τμήμα της εργασίας κατέληξα στο συμπέρασμα ότι περισσότερο χρήσιμο για το έργο θα είναι η προσθήκη κάποιας νέας δυνατότητας που έχει ζητηθεί, παρά η αποσφαλμάτωση ήδη γραμμένου κώδικα.

Μία αρκετά χρήσιμη προσθήκη αποτελεί ο κώδικας για μετατροπή ενός εγγράφου PDF σε έγγραφο XML, αφού πρώτα αυτό μετατραπεί σε αρχείο της μορφής `.txt`. Κατόπιν, κάποιος θα μπορούσε σχετικά εύκολα μέσω XSLT να μετατρέψει το έγγραφο XML σε HTML ή κάποια άλλη χρήσιμη για αυτόν μορφή. Η ιδέα αυτή αν και αρκετά ενδιαφέρουσα για το έργο, απορρίφθηκε, μετά από μακρό χρονικό διάστημα, εφόσον κρίθηκε ιδιαίτερα φιλόδοξη και απαιτητική για το επίπεδο των γνώσεων του γράφοντος.

3.2 ΤΕΛΙΚΗ ΑΠΟΦΑΣΗ

Ύστερα λοιπόν από υποδείξεις του κου Ben Litchfield, αποφάσισα να ασχοληθώ με ένα θέμα επίκαιρο και για τον ίδιο, αλλά και μέσα στα πλαίσια των δυνατοτήτων μου. Έτσι εστίασα τις προσπάθειές μου στον τομέα “[Support for Actions](#)”, δηλαδή στη δημιουργία κώδικα για την υλοποίηση από το PDFBox συγκεκριμένων «ενεργειών» που μπορούν να γίνουν σε ένα έγγραφο τέτοιας μορφής.

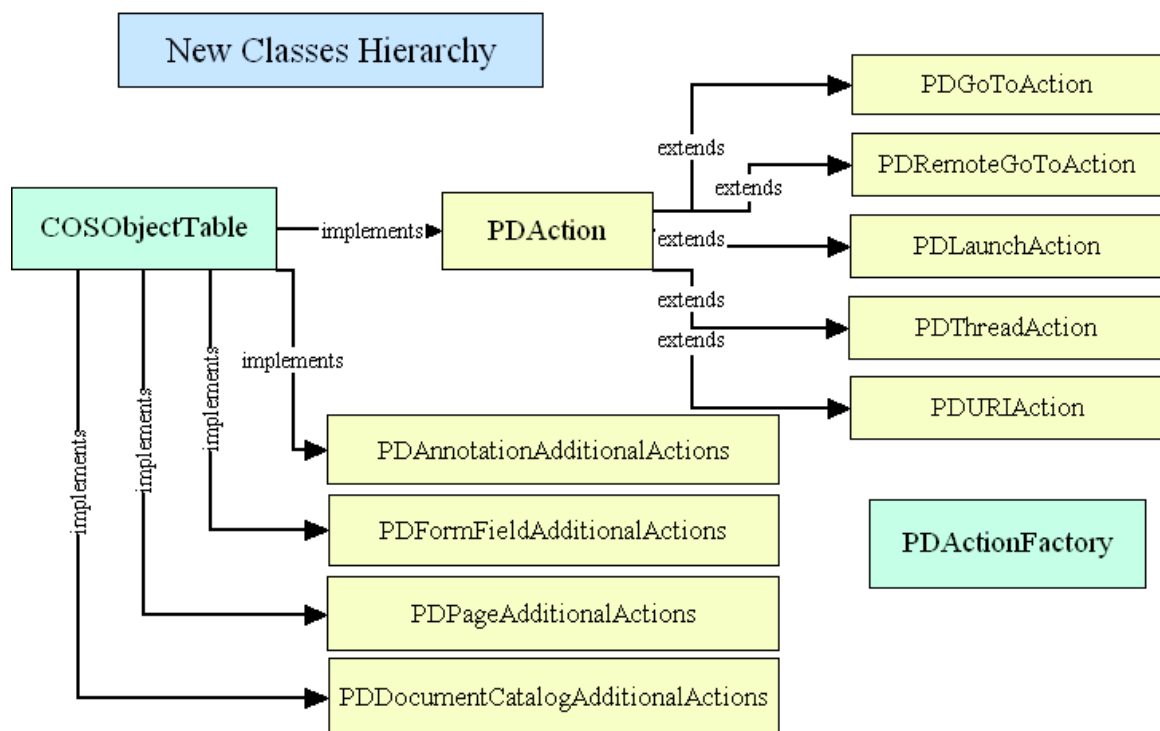
Υλοποιήθηκαν έτσι 10 κλάσεις γλώσσας Java, από τις οποίες οι δύο είχαν αρχικά οριστεί (η μία λανθασμένα), ενώ οι υπόλοιπες είναι εντελώς καινούριες. Για την τροποποίηση και δημιουργία των κλάσεων αυτών χρειάστηκε να ανατρέξω στην τέταρτη έκδοση του PDF Reference (version 1.5, section 8.5), η οποία είναι διαθέσιμη μέσω διαδικτύου και καθορίζει τα standards της μορφής και του τρόπου χειρισμού των εγγράφων PDF (Portable Document Format) της εταιρείας Adobe. Επίσης, στηρίχτηκα σε υποδείγματα αντίστοιχων κλάσεων του ίδιου πακέτου που θα συμπεριληφθούν στην επόμενη έκδοση του έργου.

3.3 ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΤΡΟΠΟΠΟΙΗΣΗ ΚΛΑΣΕΩΝ

Συγκεκριμένα, η συνεισφορά μου στο έργο ανοικτού πηγαίου κώδικα PDFBox περιλαμβάνει τις κλάσεις που παρατίθενται στον πίνακα που ακολουθεί με μία σύντομη περιγραφή της λειτουργίας τους. Το σύνολο του κώδικα που δημιουργήθηκε παρατίθεται στο παράρτημα. Στη συνέχεια ακολουθεί η ιεραρχία των κλάσεων του πακέτου `org.pdfbox.pdmodel.interactive.action`, στο οποίο ανήκουν όλες.

Όνομασία κλάσης Java	Συνοπτική περιγραφή λειτουργίας	Γραμμές κώδικα	Τροποποιημένη / νέα
PDAction	Αναπαριστά μία ενέργεια (action) η οποία μπορεί να εκτελεστεί σε ένα έγγραφο PDF.	184	Τροποποιημένη
PDAnnotationAdditionalActions	Αναπαριστά ένα λεξικό ενεργειών για υπομνήματα που μπορεί να εκτελεστεί εξαιτίας γεγονότων σε ένα έγγραφο PDF.	379	Τροποποιημένη
PDFFormFieldAdditionalActions	Αναπαριστά ένα λεξικό ενεργειών για τον καθορισμό της μορφής πεδίων που μπορεί να εκτελεστεί εξαιτίας γεγονότων σε ένα έγγραφο PDF.	215	Νέα
PDPageAdditionalActions	Αναπαριστά ένα λεξικό ενεργειών για τα αντικείμενα μιας σελίδας που μπορεί να εκτελεστεί	149	Νέα

	εξαιτίας γεγονότων σε ένα έγγραφο PDF.		
PDDocumentCatalogAdditionalActions	Αναπαριστά ένα λεξικό ενεργειών για τον κατάλογο του εγγράφου που μπορεί να εκτελεστεί εξαιτίας γεγονότων σε ένα έγγραφο PDF.	237	Νέα
PDGoToAction	Αναπαριστά μία ενέργεια GoTo που μπορεί να εκτελεστεί σε ένα έγγραφο PDF.	163	Νέα
PDRemoteGoToAction	Αναπαριστά μία ενέργεια RemoteGoTo που μπορεί να εκτελεστεί σε ένα έγγραφο PDF.	214	Νέα
PDLaunchAction	Αναπαριστά μία ενέργεια Launch που μπορεί να εκτελεστεί σε ένα έγγραφο PDF.	283	Νέα
PDThreadAction	Αναπαριστά μία ενέργεια Thread που μπορεί να εκτελεστεί σε ένα έγγραφο PDF.	271	Νέα
PDURIAction	Αναπαριστά μία ενέργεια URI που μπορεί να εκτελεστεί σε ένα έγγραφο PDF.	182	Νέα
Σύνολο	-	2277	-



3.4 ΣΥΝΕΡΓΑΣΙΑ ΜΕ ΤΗΝ ΟΜΑΔΑ ΥΛΟΠΟΙΗΣΗΣ

Η συνεργασία μου με τον κο Litchfield, ο οποίος αποτελεί και την «ψυχή» του συγκεκριμένου έργου ήταν άριστη. Όπως φαίνεται και από την ηλεκτρονική αλληλογραφία που ανταλλάξαμε και παρατίθεται στο παράρτημα ο κος Litchfield απαντούσε άμεσα και με σαφήνεια στις όποιες απορίες μου, αν και αρκετές φορές με λακωνικό τρόπο.

Από την πλευρά μου θα έπρεπε ίσως να είχα ρωτήσει νωρίτερα για κάποια ιδέα διαφορετική από τον αρχικό μου στόχο (τη δημιουργία κλάσης που να μετατρέπει ένα .txt αρχείο σε XML), ώστε να είχα εξοικονομήσει πολύτιμο χρόνο. Σε γενικές γραμμές πάντως η συνεργασία μας ήταν άψογη και θα συνεχίσει να είναι εφόσον σκοπεύω να συμμετάσχω και στη συνέχεια στην προσπάθεια για ανάπτυξη ανοιχτού πηγαίου κώδικα.

Το αρνητικό στοιχείο είναι πως προς το παρόν δεν έχει υλοποιηθεί το σύνολο της εργασίας μου, καθώς το πακέτο στο οποίο ανήκουν οι κλάσεις που δημιούργησα ανανεώνεται και επεκτείνεται διαρκώς.

Πάντως το σύνολο της εργασίας μου υιοθετήθηκε από τον κο Litchfield, ο οποίος αναγνώρισε τη συνεισφορά μου, και θα συμπεριληφθεί μαζί με αρκετές άλλες αλλαγές στη νέα έκδοση που θα δημοσιευτεί σύντομα.

4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Για την υλοποίηση του έργου κρίθηκε αναγκαία η πολύωρη μελέτη και εμπλοκή με τα παρακάτω θέματα, τα οποία συμπληρώνουν το υπόβαθρο και την εμπειρία του γράφοντα:

- Τρόπος χειρισμού αρχείων τύπου .java (compile, run),
- Επαφή με τη γραμμή εντολών και χρήση ορισμένων βασικών εντολών,
- Γνωριμία αρχείων τύπου .jar και τρόπων χειρισμού τους,
- Γνώση νομικών θεμάτων σχετικά με τις άδειες χρήσης λογισμικού,
- Μελέτη και εφαρμογή του τρόπου καθορισμού του classpath,
- Εξοικείωση με τον τρόπο εργασίας σε έργα λογισμικού ανοιχτού πηγαίου κώδικα μέσω διαδικτύου (CVS, bugs, patches, documentation, κ.ά),
- Δημιουργία και χρήση αρχείων της μορφής autoexec.bat,
- Μελέτη τρόπων μετατροπής διαφόρων τύπων αρχείων σε μορφή XML με τη χρήση της Java, αλλά και επαφή με παρόμοια έργα λογισμικού (jasperreports, jdom), γνώση η οποία δεν εφαρμόστηκε βέβαια στην παρούσα εργασία,
- Γνωριμία και χρήση του εργαλείου προγραμματισμού Eclipse έκδοση 2.1.

Επίσης, οι διαλέξεις που έγιναν κατά τη διάρκεια του εξαμήνου κατόρθωσαν να δώσουν στο φοιτητή μια συνολική εικόνα για τον τρόπο μελέτης, κατανόησης και επαναχρησιμοποίησης του κώδικα που περιλαμβάνεται σε έργα λογισμικού ανοιχτού πηγαίου κώδικα, αλλά και να τον ωθήσουν στην εμπλοκή του σε παρόμοια έργα, γεγονός σπάνιο για τα ελληνικά πάντα δεδομένα. Τέλος, οι συχνές παρουσιάσεις κράτησαν αμείωτο το ενδιαφέρον των φοιτητών και τους βοήθησαν να πετύχουν τον τελικό τους στόχο.

Συμπερασματικά, πιστεύω πως η πρωτότυπη για τα δεδομένα ελληνικού πανεπιστημίου μορφή της εργασίας λειτούργησε εποικοδομητικά και πρόσθεσε πολλές νέες εμπειρίες σε όσους ενεπλάκησαν ενεργά. Η επιλογή ενός πραγματικού και εφαρμοσμένου έργου, η συνεργασία με άγνωστα άτομα μέσω του διαδικτύου, αλλά και το ίδιο το ζητούμενο της εργασίας αποτέλεσε όπως έχει προαναφερθεί και αποτελεί μια πρόκληση για κάθε τελειόφοιτο φοιτητή.



5. ΠΑΡΑΡΤΗΜΑ

5.1 ΑΔΕΙΑ ΧΡΗΣΗΣ PDFBOX: BSD LICENCE

The following is a BSD license template. To generate your own license, change the values of OWNER, ORGANIZATION and YEAR from their original values as given here, and substitute your own.

Note: The advertising clause in the license appearing on BSD Unix files was officially [rescinded](#) by the Director of the Office of Technology Licensing of the University of California on July 22 1999. He states that clause 3 is "hereby deleted in its entirety."

Note the new BSD license is thus equivalent to the [MIT License](#), except for the no-endorsement final clause.

<OWNER> = Regents of the University of California
<ORGANIZATION> = University of California, Berkeley
<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5.2 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟ ΔΙΑΧΕΙΡΙΣΤΗ ΤΟΥ ΕΡΓΟΥ

Mail 1a

To: benlitchfield@users.sourceforge.net
Subject: Undergraduate student from Greece
From: [ptoum <ptoum@users.sourceforge.net>](mailto:ptoum@users.sourceforge.net)
Date: Wed, 05 May 2004 06:23:36 -0700

Mr. Ben Litchfield,

I am a new member of the community and I am writing to you asking for your advice and help. I am an undergraduate student in Athens University of Economics and Business, in the Department of Management Science and Technology (dmst.aueb.gr). In order to accomplish successfully the course "Maintenance and Refactoring of Information Systems" I have to make some changes and add new functionality to a project of the "Sourceforge.net". As you can understand I have chosen your project, written in pure Java, because I am more familiar with this language, and for other reasons like the development status and the topic. Our professor, Mr. Diomidis Spinellis (author of the book "Code Reading", if you know), would be very satisfied if I could commit some useful changes to your project. Have you any ideas? During this period what new elements do you want to add to PDFBox? I would appreciate if you could help me!

PS: I am a new member of the "software community" so, you might assume that I am not an experienced programmer or a well-informed member of sourceforge.net. I would also like to continue my studies in Computer Science and to help the efforts of open source software development in my country, as my professor already does.

Yours sincerely,
Panagiotis Toumasis

Mail 1b

Date: Wed, 5 May 2004 19:49:25 -0400 (EDT)
From: [Ben Litchfield <ben@csh.rit.edu>](mailto:ben@csh.rit.edu)
To: [ptoum <ptoum@users.sourceforge.net>](mailto:ptoum@users.sourceforge.net)
Subject: Re: Undergraduate student from Greece

I would love some help, if this is for a class maybe you guys could do a team effort and do something big, or there are enough smaller things that a couple people could work different pieces.

You could start by looking at the PDFBox RFE page at:
http://sourceforge.net/tracker/?group_id=78314&atid=552835

A couple that would be pretty interesting would be

- Extract Images
- support color space
- PDF->HTML conversion
- HTML->PDF conversion
- adding support for PDF 1.5 object streams
- FDF export/import

I have started some work on a PDFViewer(org.pdfbox.PDFViewer), which allows a developer to view the internals of the PDF document structure. It basically sucks right now, I would like to have it be able to change values and write the PDF back to disk.

I would also be interested in creating an Acrobat Reader like application so people can view PDF documents in Java. This is no way you could complete this but if you started it maybe it would spark some interest and other people could work on it as well.

I don't know how much time or effort you have but if something interests you let me know and I can point you in the right direction.

Ben

Mail 2a

To: ben@csh.rit.edu
From: Panagiotis Toumasis <ptoumasis@mail.gr>
Subject: PDFBox - greek student
Date: Thu, 06 May 2004 12:09:02 EET

Mr. Ben Litchfield,
First of all I'd like to thank you very much for accepting me to your project! My fellows have problems with the projects that have chosen. I have not much time (just a month) to complete the changes for my course, but I would like to continue with further work and after this period.

I think that PDF->HTML conversion and HTML->PDF conversion or extracting images is closer to my abilities. I will present my ideas to my professor on Monday, 10/5, so I would appreciate if you could tell me which files should I examine and if you could "point me to the right direction" in order to become efficient and effective!
Thank you very much for your time!

Panagiotis Toumasis

Mail 2b

Date: Thu, 6 May 2004 07:31:45 -0400 (EDT)
From: Ben Litchfield <ben@csh.rit.edu>
To: Panagiotis Toumasis <ptoumasis@mail.gr>

Subject: Re: PDFBox - greek student

The `org.pdfbox.util.PDFTextStripper` is the class that extracts the text out of the PDF. This class will give you the text and font information, there is no image extraction right now so you will have to ignore images. Look at the `showString` method in that class.

Ben

Mail 3a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: PDFBox - greek student
Date: Thu, 06 May 2004 14:20:55

Ok, I'll start from there! Thank you!

Panagiotis Toumasis

Mail 3b

Date: Thu, 6 May 2004 08:33:24 -0400 (EDT)
From: [Ben Litchfield <ben@csh.rit.edu>](mailto:ben@csh.rit.edu)
To: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: PDFBox - greek student

There are some people asking for a PDF->XML as well, that might be a little easier to implement. It would need to contain font/positioning information.

Ben

Mail 4a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: PDFBox - greek student
Date: Thu, 06 May 2004 23:12:20 EET

It is useful too, but why do you think it is easier? As I wrote in a previous mail, I'll ask my professor first and then decide. I'll inform you soon!

Panagiotis

Mail 5a

To: ben@cs.h.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: PDFBox - greek student
Date: Tue, 11 May 2004 10:50:00 EET

Hello Mr. Litchfield,

I discussed my ideas with my professor and he thinks that it is a good task! I will try to create a new class, subclass of the class PDFStreamEngine, which will convert a PDF Document to XML. After that, if possible, we could convert XML to HTML (through XSLT). If you already have any comments or ideas on this implementation plan, please let me know!

Panagiotis Toumasis

Mail 6a

To: ben@cs.h.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Student from Athens
Date: Thu, 30 Sep 2004 21:05:51 EET

Mr. Ben Litchfield,

I hope you remember me! I have sent you some mails during the last May, informing you that I have chosen the PDFBox for my project in the course "Maintenance and Refactoring of Information Systems". I am now in the difficult position to ask for your help again. I try to write a program that takes as input the output of your class PDFStreamEngine (or the class PDFTextStripper) and gives the output of a PDF document in XML format.

During the summer I have read the book "Processing XML with Java" but I am still not able to write the code! I am sure I haven't the experience that is needed for such an enterprise. I am very disappointed and I ask for your help. Could you please show me another direction of work? What else do you want to add to your project?

Yours sincerely,
Panagiotis Toumasis

Mail 6b

Date: Thu, 30 Sep 2004 15:23:13 -0400 (EDT)
From: [Ben Litchfield <ben@cs.h.rit.edu>](mailto:ben@cs.h.rit.edu)
To: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: Student from Athens

There are several new RFE available on the website that might be easier.

For example

Support for actions

This is really just adding support for PDF actions, all you would need to do it read a dozen or so pages in the PDF Reference (1.5 Section 8.5) and follow the pattern of the rest of the pdmodel. It is really easy, basically just wrapping some other code. For an example of what you would need to do you could look at the newly added org.pdfbox.pdmodel.fdf package and how it relates to section 8.6.6 of the PDF Reference.

http://sourceforge.net/tracker/index.php?func=detail&aid=1037395&group_id=78314&atid=552835

Take a look at some other RFE on the site and see if one would be easier.

The other thing that needs to be done is simply to create examples that use the library. There are a ton of things that can be done with a PDF document and I only have a handful of examples. These would go in the org.pdfbox.examples package.

Your schools must be different in Greece. What kind of class goes from May until November?

Good luck,
Ben

Mail 7a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:Panagiotis.Toumasis@ptoumasis@mail.gr)
Subject: Re: Student from Athens
Date: Fri, 01 Oct 2004 19:47:43 EET

Hello Mr. Litchfield,

First of all thank you for your time! Secondly, I have searched all the folders of the zip file PDFBox0.6.6 for the PDF Reference, but I have found nothing! What type of file is it and where can I find it? Where are the ".fdf" files (I can't find the org.pdfbox.pdmodel.fdf file)?

The last thing I'd like to ask is to explain me what these examples do and give me some examples of what "examples" you want to add. I concede I haven't understand the role of this package.

Sorry for my questions, but as you understand I am a student and not a professional developer.

Thank you in advance,
Panagiotis Toumasis

PS: Here in Greece we are able to be examined during June and September - October for a second time at the same lessons.

Mail 7b

Date: Fri, 1 Oct 2004 14:23:18 -0400 (EDT)
From: [Ben Litchfield <ben@csh.rit.edu>](mailto:ben@csh.rit.edu)
To: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: Student from Athens

The PDF Reference is published by Adobe. It is a PDF document that can be found at:

<http://partners.adobe.com/asn/tech/pdf/specifications.jsp>

This is a new package that has not been released yet, it is part of CVS (which you should learn how to use) or you can get a nightly release from <http://www.pdfbox.org> You should not be looking at the released version, it is outdated, CVS is what you should be looking at!!

I didn't have any specific examples in mind. What would you like to be able to do with a pdf document? Maybe list out all fonts that are being used, print out the bookmarks of a document.

Do you know Java Swing? In the nightly release is a program called org.pdfbox.PDFViewer that shows the internal structure of a pdf document.

You could update that to allow a document to be edited, because right now it only shows a a read-only view.

Ben

Mail 8a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: Student from Athens
Date: Fri, 01 Oct 2004 21:55:34 EET

Thanks a lot! I found everything! I'll take a look and I'll send you the results soon!

Panagiotis Toumasis

Mail 9a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: Student from Athens
Date: Tue, 05 Oct 2004 9:29:50 EET

Good morning Mr. Litchfield,

I've studied the section 8.5 from the PDF Reference, but I haven't understand exactly what to do. Could you be a little more specific? For example you want to add a new .java file in the package org.pdfbox.pdmodel.interactive.actions that implements a particular action (go to, launch, etc) or something else? And where is the code that I could "wrap"? I would be grateful if you gave me an example.

Thank you for your time!

Panagiotis Toumasis

Mail 9b

Date: Tue, 5 Oct 2004 08:31:45 -0400 (EDT)
From: [Ben Litchfield <ben@csh.rit.edu>](mailto:ben@csh.rit.edu)
To: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: Student from Athens

That section lists attributes that are part of an Action, if you look at the Action class those attributes are not there. Add them to that class.

Look at other pdmodel classes to see how I have done attributes.

Ben

Mail 10a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: Student from Athens
Date: Tue, 05 Oct 2004 19:34:32 EET
Επισυναπτόμενα: [PDAction_1.2.java](#)

Mr. Lithchfield,

I send you the PDAction version 1.2. Please check my attempt and tell me if it is right or ...nonsense! If you can correct any mistakes or write something better please do it. It is based to the table 8.37 (pg 592) of the PDF Reference. May I continue with the class PDAdditionalActions and which table should I look?

Panagiotis Toumasis

Mail 10b

Date: Tue, 5 Oct 2004 20:33:21 -0400 (EDT)
From: [Ben Litchfield <ben@csh.rit.edu>](mailto:ben@csh.rit.edu)
To: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)

Subject: Re: Student from Athens

Yes, what you have looks pretty good. I think there is a mistake with my current code.

I have a class called PAdditionalActions, I created that as a placeholder for another class that had additional actions, it was a mistake.

There really should be the following classes

```
PDAnnotationAdditionalActions Table 8.38
PDPageAdditionalActions Table 8.39
PDFormFieldAdditionalActions Table 8.40
PDDocumentCatalogAdditionalActions Table 8.41
```

If you still have some time after finishing those classes you could move onto the Action type classes defined in section 8.5.3

The action types will extend/inherit from PDAction and add some additional entries.

So for example

```
public class PDGoToAction extends PDAction
{
    //add all attributes defined in table 8.43
}
```

```
PDRemoteGoToAction Table 8.44
PDLaunchAction Table 8.45
...
PDTransitionAction Table 8.57
```

If you got all of that done that would be very awesome. If you still have time you could create an example application that read in a PDF and set an action. So you could create a PDDocumentCatalogAdditionActions object set the WS(document save) action to be Launch and launch an application like notepad or something.

Ben

Mail 11a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:Panagiotis.Toumasis@mail.gr)
Subject: New classes
Date: Wed, 06 Oct 2004 17:56:45 EET

Επισυναπτόμενα: [PDAction.java](#)
[PDAnnotationAdditionalActions.java](#)
[PDDocumentCatalogAdditionalActions.java](#)
[PDFormFieldAdditionalActions.java](#)
[PDPageAdditionalActions.java](#)

Hello Mr. Litchfield

I send you the classes:

PDAction (with a miscellaneous correction)
PDAnnotationAdditionalActions
PDPageAdditionalActions
PDFormFieldAdditionalActions
PDDocumentCatalogAdditionalActions

I will work on the action type classes and I will send you them tomorrow, but I would appreciate some help with the example. is there any example to help me?

Thanks a lot!
Panagiotis Toumasis

PS:Will you publish these new classes through the cvs? I need to write it at the documentation of my project.

Mail 11b

Date: Wed, 6 Oct 2004 13:30:48 -0400 (EDT)
From: [Ben Litchfield <ben@csh.rit.edu>](mailto:ben@csh.rit.edu)
To: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)
Subject: Re: New classes

I have not looked at the classes in detail but you are on the right path. I will publish these to CVS, I would prefer to do it all at once when you are finished with the ActionTypes, is that alright for your project?

There is not really an example to help you look at for creating an Action example.

Here is some pseudo code for what I think you need to do.

1)In PDDocumentCatalog you need to add the "AA" attribute, described in table 3.25 in section 3.6.1, this will get/set the PDDocumentAdditionalAction class that you wrote.

2)In your example create a PDDocumentAdditionalAction and set it on the document using the setter create from step one

3)In your example create a PDLaunchAction and set it on one of the properties in the PDDocumentAdditionalAction

You should take a look at the other command line/examples to see how to start out.

Ben

Mail 12a

To: ben@csh.rit.edu
From: [Panagiotis Toumasis <ptoumasis@mail.gr>](mailto:ptoumasis@mail.gr)

Subject: Re: New classes

Date: Thu, 07 Oct 2004 21:29:58 EET

[PDGoToAction.java](#)

[PDLaunchAction.java](#)

Επισυναπτόμενα: [PDRemoteGoToAction.java](#)

[PDThreadAction.java](#)

[PDURIAction.java](#)

Hello Mr. Litchfield,

I send you the action type classes:

```
PDGoToAction
PDRemoteGoToAction
PDLaunchAction
PDThreadAction
PDURIAction
```

I have to prepare the documentation of my project now, so I do not have enough time to make the rest classes, but I will work on these soon. There is no problem with the publishing to the CVS. My professor will understand. I'll show him the e-mails.

Thank you again for your time!
Panagiotis Toumasis

5.3 ΤΡΟΠΟΠΟΙΗΜΕΝΕΣ ΚΑΙ ΝΕΕΣ ΚΛΑΣΕΙΣ JAVA

5.3.1 PDAction.java Class

```
/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 * contributors may be used to endorse or promote products derived from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
 * CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE
```

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

*
* http://www.pdfbox.org
*
*/
package org.pdfbox.pdmodel.interactive.action;

```

```

import java.util.ArrayList;
import java.util.List;

```

```

import org.pdfbox.cos.COSArray;
import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;

```

```

import org.pdfbox.pdmodel.common.COSObjectable;
import org.pdfbox.pdmodel.common.COSArrayList;

```

```

/**
 * This represents an action that can be executed in a PDF document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.2 $
 */

```

```

public class PDAction implements COSObjectable
{
    private COSDictionary action;

```

```

/**
 * Default constructor.
 */
public PDAction()
{
    action = new COSDictionary();

```



```

}

/**
 * Constructor.
 *
 * @param a The action dictionary.
 */
public PDAction( COSDictionary a )
{
    action = a;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSBase getCOSObject()
{
    return action;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSDictionary getCOSDictionary()
{
    return action;
}

/**
 * This will get the type of PDF object that the actions dictionary describes.
 * If present must be Action for an action dictionary.
 *
 * @return The Type of PDF object.
 */
public String getType()
{
    return action.getNameAsString( "Type" );
}

/**
 * This will set the type of PDF object that the actions dictionary describes.
 * If present must be Action for an action dictionary.
 *
 * @param Type The new Type for the PDF object.
 */
public void setType( String type )

```

```

{
    action.setName( "Type", type );
}

/**
 * This will get the type of action that the actions dictionary describes.
 * If present, must be Action for an action dictionary.
 *
 * @return The S entry of actions dictionary.
 */
public String getS()
{
    return action.getNameAsString( "S" );
}

/**
 * This will set the type of action that the actions dictionary describes.
 * If present, must be Action for an action dictionary.
 *
 * @param S The new type of action.
 */
public void setS( String S )
{
    action.setName( "S", S );
}

/**
 * This will get the next action, or sequence of actions, to be performed after this
one.
 * The value is either a single action dictionary or an array of action dictionaries
 * to be performed in order.
 *
 * @return The Next action or sequence of actions.
 */

// In case the value is a single action dictionary.

public ActionDictionary getNext()
{
    COSDictionary next = (COSDictionary)action.getDictionaryObject( "Next" );
    ActionDictionary retval = null;
    if( Next != null )
    {
        retval = new ActionDictionary( next );
    }
    return retval;
}

// In case the value is an array of action dictionaries.

public COSArray getNext()

```

```

    {
        return (COSArray)action.getDictionaryObject( "Next" );
    }

/**
 * This will set the next action, or sequence of actions, to be performed after this
one.
 * The value is either a single action dictionary or an array of action dictionaries
 * to be performed in order.
 *
 * @param next The Next action or sequence of actions.
 */

// In case the value is a single action dictionary.

public void setNext( ActionDictionary next )
    {
        action.setItem( "Next", next );
    }
// In case the value is an array of action dictionaries.

    public void setNext( COSArray next )
    {
        action.setItem( "Next", next );
    }
}

```

5.3.2 PDAnnotationAdditionalActions.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 * contributors may be used to endorse or promote products derived from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE

```

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*
 * <http://www.pdfbox.org>
 *
 */
 package org.pdfbox.pdmodel.interactive.action;

import org.pdfbox.cos.COSBase;
 import org.pdfbox.cos.COSDictionary;

import org.pdfbox.pdmodel.common.COSObjectable;

```

/**
 * This class represents an annotation's dictionary of actions
 * that occur due to events.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */
public class PDAnnotationAdditionalActions implements COSObjectable
{
    private COSDictionary actions;

    /**
     * Default constructor.
     */
    public PDAnnotationAdditionalActions()
    {
        actions = new COSDictionary();
    }

    /**
     * Constructor.
     */
  
```

```

* @param a The action dictionary.
*/
public PDAnnotationAdditionalActions( COSDictionary a )
{
    actions = a;
}

/**
* Convert this standard java object to a COS object.
*
* @return The cos object that matches this Java object.
*/
public COSBase getCOSObject()
{
    return actions;
}

/**
* Convert this standard java object to a COS object.
*
* @return The cos object that matches this Java object.
*/
public COSDictionary getCOSDictionary()
{
    return actions;
}

/**
* This will get an action to be performed when the cursor
* enters the annotation's active area.
*
* @return The E entry of annotation's additional actions dictionary.
*/
public AnnotationAdditionalActionsDictionary getE()
{
    COSDictionary E = (COSDictionary)actions.getDictionaryObject( "E" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( E != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( E );
    }
    return retval;
}

/**
* This will set an action to be performed when the cursor
* enters the annotation's active area.
*
* @param E The action to be performed.
*/

```

```

public void setE( AnnotationAdditionalActionsDictionary E )
    {
        actions.setItem( "E", E );
    }

/**
 * This will get an action to be performed when the cursor
 * exits the annotation's active area.
 *
 * @return The X entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getX()
{
    COSDictionary X = (COSDictionary)actions.getDictionaryObject( "X" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( X != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( X );
    }
    return retval;
}

/**
 * This will set an action to be performed when the cursor
 * exits the annotation's active area.
 *
 * @param E The action to be performed.
 */
public void setX( AnnotationAdditionalActionsDictionary X )
    {
        actions.setItem( "X", X );
    }

/**
 * This will get an action to be performed when the mouse button
 * is pressed inside the annotation's active area.
 * The name D stands for "down".
 *
 * @return The D entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getD()
{
    COSDictionary D = (COSDictionary)actions.getDictionaryObject( "D" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( D != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( D );
    }
    return retval;
}

```

```

/**
 * This will set an action to be performed when the mouse button
 * is pressed inside the annotation's active area.
 * The name D stands for "down".
 *
 * @param D The action to be performed.
 */
public void setD( AnnotationAdditionalActionsDictionary D )
    {
        actions.setItem( "D", D );
    }

/**
 * This will get an action to be performed when the mouse button
 * is released inside the annotation's active area.
 * The name U stands for "up".
 *
 * @return The U entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getU()
    {
        COSDictionary U = (COSDictionary)actions.getDictionaryObject( "U" );
        AnnotationAdditionalActionsDictionary retval = null;
        if( U != null )
            {
                retval = new AnnotationAdditionalActionsDictionary( U );
            }
        return retval;
    }

/**
 * This will set an action to be performed when the mouse button
 * is released inside the annotation's active area.
 * The name U stands for "up".
 *
 * @param U The action to be performed.
 */
public void setU( AnnotationAdditionalActionsDictionary U )
    {
        actions.setItem( "U", U );
    }

/**
 * This will get an action to be performed when the annotation
 * receives the input focus.
 *
 * @return The Fo entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getFo()

```

```

{
    COSDictionary Fo = (COSDictionary)actions.getDictionaryObject( "Fo" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( Fo != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( Fo );
    }
    return retval;
}

/**
 * This will set an action to be performed when the annotation
 * receives the input focus.
 *
 * @param Fo The action to be performed.
 */
public void setFo( AnnotationAdditionalActionsDictionary Fo )
{
    actions.setItem( "Fo", Fo );
}

/**
 * This will get an action to be performed when the annotation
 * loses the input focus.
 * The name Bl stands for "blurred".
 *
 * @return The Bl entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getBl()
{
    COSDictionary Bl = (COSDictionary)actions.getDictionaryObject( "Bl" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( Bl != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( Bl );
    }
    return retval;
}

/**
 * This will set an action to be performed when the annotation
 * loses the input focus.
 * The name Bl stands for "blurred".
 *
 * @param Bl The action to be performed.
 */
public void setBl( AnnotationAdditionalActionsDictionary Bl )
{
    actions.setItem( "Bl", Bl );
}

```



```

/**
 * This will get an action to be performed when the page containing
 * the annotation is opened. The action is executed after the O action
 * in the page's additional actions dictionary and the OpenAction entry
 * in the document catalog, if such actions are present.
 *
 * @return The PO entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getPO()
{
    COSDictionary PO = (COSDictionary)actions.getDictionaryObject( "PO" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( PO != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( PO );
    }
    return retval;
}

/**
 * This will set an action to be performed when the page containing
 * the annotation is opened. The action is executed after the O action
 * in the page's additional actions dictionary and the OpenAction entry
 * in the document catalog, if such actions are present.
 *
 * @param PO The action to be performed.
 */
public void setPO( AnnotationAdditionalActionsDictionary PO )
{
    actions.setItem( "PO", PO );
}

/**
 * This will get an action to be performed when the page containing
 * the annotation is closed. The action is executed before the C action
 * in the page's additional actions dictionary, if present.
 *
 * @return The PC entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getPC()
{
    COSDictionary PC = (COSDictionary)actions.getDictionaryObject( "PC" );
    AnnotationAdditionalActionsDictionary retval = null;
    if( PC != null )
    {
        retval = new AnnotationAdditionalActionsDictionary( PC );
    }
    return retval;
}

```

```

/**
 * This will set an action to be performed when the page containing
 * the annotation is closed. The action is executed before the C action
 * in the page's additional actions dictionary, if present.
 *
 * @param PC The action to be performed.
 */
public void setPC( AnnotationAdditionalActionsDictionary PC )
    {
        actions.setItem( "PC", PC );
    }

/**
 * This will get an action to be performed when the page containing
 * the annotation becomes visible in the viewer application's user interface.
 *
 * @return The PV entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getPV()
    {
        COSDictionary PV = (COSDictionary)actions.getDictionaryObject( "PV" );
        AnnotationAdditionalActionsDictionary retval = null;
        if( PV != null )
            {
                retval = new AnnotationAdditionalActionsDictionary( PV );
            }
        return retval;
    }

/**
 * This will set an action to be performed when the page containing
 * the annotation becomes visible in the viewer application's user interface.
 *
 * @param PV The action to be performed.
 */
public void setPV( AnnotationAdditionalActionsDictionary PV )
    {
        actions.setItem( "PV", PV );
    }

/**
 * This will get an action to be performed when the page containing the annotation
 * is no longer visible in the viewer application's user interface.
 *
 * @return The PI entry of annotation's additional actions dictionary.
 */
public AnnotationAdditionalActionsDictionary getPI()
    {
        COSDictionary PI = (COSDictionary)actions.getDictionaryObject( "PI" );

```

```

AnnotationAdditionalActionsDictionary retval = null;
if( PI != null )
{
    retval = new AnnotationAdditionalActionsDictionary( PI );
}
return retval;
}

/**
 * This will set an action to be performed when the page containing the annotation
 * is no longer visible in the viewer application's user interface.
 *
 * @param PI The action to be performed.
 */
public void setPI( AnnotationAdditionalActionsDictionary PI )
{
    actions.setItem( "PI", PI );
}
}

```

5.3.3 PDFFormFieldAdditionalActions.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice,
 * this list of conditions and the following disclaimer in the
 * documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 * contributors may be used to endorse or promote products derived
 * from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
 * CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 * PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
 * BE LIABLE FOR ANY

```

* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
 GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* <http://www.pdfbox.org>

```

*/
package org.pdfbox.pdmodel.interactive.action;

```

```

import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;

```

```

import org.pdfbox.pdmodel.common.COSObjectable;

```

```

/**

```

```

 * This class represents a form field's dictionary of actions
 * that occur due to events.

```

```

 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */

```

```

public class PDFFormFieldAdditionalActions implements COSObjectable
{

```

```

    private COSDictionary actions;

```

```

    /**

```

```

     * Default constructor.

```

```

    */

```

```

    public PDFFormFieldAdditionalActions()

```

```

    {
        actions = new COSDictionary();
    }

```

```

    /**

```

```

     * Constructor.

```

```

     *

```

```

     * @param a The action dictionary.

```

```

    */

```

```

    public PDFFormFieldAdditionalActions( COSDictionary a )

```

```

    {
        actions = a;
    }

```

```

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSBase getCOSObject()
{
    return actions;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSDictionary getCOSDictionary()
{
    return actions;
}

/**
 * This will get a JavaScript action to be performed when the user
 * types a keystroke into a text field or combo box or modifies the
 * selection in a scrollable list box. This allows the keystroke to
 * be checked for validity and rejected or modified.
 *
 * @return The K entry of form field's additional actions dictionary.
 */
public FormFieldAdditionalActionsDictionary getK()
{
    COSDictionary K = (COSDictionary)actions.getDictionaryObject( "K"
);
    FormFieldAdditionalActionsDictionary retval = null;
    if( K != null )
    {
        retval = new FormFieldAdditionalActionsDictionary( K );
    }
    return retval;
}

/**
 * This will set a JavaScript action to be performed when the user
 * types a keystroke into a text field or combo box or modifies the
 * selection in a scrollable list box. This allows the keystroke to
 * be checked for validity and rejected or modified.
 *
 * @param K The action to be performed.
 */
public void setK( FormFieldAdditionalActionsDictionary K )
{
    actions.setItem( "K", K );
}

```

```

/**
 * This will get a JavaScript action to be performed before
 * the field is formatted to display its current value. This
 * allows the field's value to be modified before formatting.
 *
 * @return The F entry of form field's additional actions dictionary.
 */
public FormFieldAdditionalActionsDictionary getF()
{
    COSDictionary F = (COSDictionary)actions.getDictionaryObject( "F"
);
    FormFieldAdditionalActionsDictionary retval = null;
    if( F != null )
    {
        retval = new FormFieldAdditionalActionsDictionary( F );
    }
    return retval;
}

/**
 * This will set a JavaScript action to be performed before
 * the field is formatted to display its current value. This
 * allows the field's value to be modified before formatting.
 *
 * @param F The action to be performed.
 */
public void setF( FormFieldAdditionalActionsDictionary F )
{
    actions.setItem( "F", F );
}

/**
 * This will get a JavaScript action to be performed
 * when the field's value is changed. This allows the
 * new value to be checked for validity.
 * The name V stands for "validate".
 *
 * @return The V entry of form field's additional actions dictionary.
 */
public FormFieldAdditionalActionsDictionary getV()
{
    COSDictionary V = (COSDictionary)actions.getDictionaryObject( "V"
);
    FormFieldAdditionalActionsDictionary retval = null;
    if( V != null )
    {
        retval = new FormFieldAdditionalActionsDictionary( V );
    }
    return retval;
}

```

```

/**
 * This will set a JavaScript action to be performed
 *   * when the field's value is changed. This allows the
 *   * new value to be checked for validity.
 * The name V stands for "validate".
 *
 * @param V The action to be performed.
 */
public void setV( FormFieldAdditionalActionsDictionary V )
    {
        actions.setItem( "V", V );
    }

/**
 * This will get a JavaScript action to be performed in order to
recalculate
 * the value of this field when that of another field changes. The order
in which
 * the document's fields are recalculated is defined by the CO entry in
the
 * interactive form dictionary.
 * The name C stands for "calculate".
 *
 * @return The C entry of form field's additional actions dictionary.
 */
public FormFieldAdditionalActionsDictionary getC()
    {
        COSDictionary C = (COSDictionary)actions.getDictionaryObject( "C"
);
        FormFieldAdditionalActionsDictionary retval = null;
        if( C != null )
            {
                retval = new FormFieldAdditionalActionsDictionary( C );
            }
        return retval;
    }

/**
 * This will set a JavaScript action to be performed in order to
recalculate
 * the value of this field when that of another field changes. The
order in which
 * the document's fields are recalculated is defined by the CO entry
in the
 * interactive form dictionary.
 * The name C stands for "calculate".
 *
 * @param C The action to be performed.
 */
public void setC( FormFieldAdditionalActionsDictionary C )
    {
        actions.setItem( "C", C );
    }

```

```
}  
}
```

5.3.4 PDPageAdditionalActions.java Class

```
/**  
 * Copyright (c) 2004, www.pdfbox.org  
 * All rights reserved.  
 *  
 * Redistribution and use in source and binary forms, with or without  
 * modification, are permitted provided that the following conditions are  
 met:  
 *  
 * 1. Redistributions of source code must retain the above copyright  
 notice,  
 * this list of conditions and the following disclaimer.  
 * 2. Redistributions in binary form must reproduce the above copyright  
 notice,  
 * this list of conditions and the following disclaimer in the  
 documentation  
 * and/or other materials provided with the distribution.  
 * 3. Neither the name of pdfbox; nor the names of its  
 * contributors may be used to endorse or promote products derived  
 from this  
 * software without specific prior written permission.  
 *  
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND  
 CONTRIBUTORS "AS IS"  
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
 LIMITED TO, THE  
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A  
 PARTICULAR PURPOSE ARE  
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS  
 BE LIABLE FOR ANY  
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
 CONSEQUENTIAL DAMAGES  
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE  
 GOODS OR SERVICES;  
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
 HOWEVER CAUSED AND ON  
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
 LIABILITY, OR TORT  
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT  
 OF THE USE OF THIS  
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
 *  
 * http://www.pdfbox.org  
 *  
 */  
package org.pdfbox.pdmodel.interactive.action;  
  
import org.pdfbox.cos.COSBase;
```



```

import org.pdfbox.cos.COSDictionary;

import org.pdfbox.pdmodel.common.COSObjectable;

/**
 * This class represents a page object's dictionary of actions
 * that occur due to events.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */
public class PDPageAdditionalActions implements COSObjectable
{
    private COSDictionary actions;

    /**
     * Default constructor.
     */
    public PDPageAdditionalActions()
    {
        actions = new COSDictionary();
    }

    /**
     * Constructor.
     *
     * @param a The action dictionary.
     */
    public PDPageAdditionalActions( COSDictionary a )
    {
        actions = a;
    }

    /**
     * Convert this standard java object to a COS object.
     *
     * @return The cos object that matches this Java object.
     */
    public COSBase getCOSObject()
    {
        return actions;
    }

    /**
     * Convert this standard java object to a COS object.
     *
     * @return The cos object that matches this Java object.
     */
    public COSDictionary getCOSDictionary()
    {
        return actions;
    }
}

```

```

}

/**
 * This will get an action to be performed when the page
 * is opened. This action is independent of any that may be
 * defined by the OpenAction entry in the document catalog,
 * and is executed after such an action.
 *
 * @return The O entry of page object's additional actions dictionary.
 */
public PageAdditionalActionsDictionary getO()
{
    COSDictionary O = (COSDictionary)actions.getDictionaryObject( "O"
);
    PageAdditionalActionsDictionary retval = null;
    if( O != null )
    {
        retval = new PageAdditionalActionsDictionary( O );
    }
    return retval;
}

/**
 * This will set an action to be performed when the page
 * is opened. This action is independent of any that may be
 * defined by the OpenAction entry in the document catalog,
 * and is executed after such an action.
 *
 * @param O The action to be performed.
 */
public void setO( PageAdditionalActionsDictionary O )
{
    actions.setItem( "O", O );
}

/**
 * This will get an action to be performed when the page
 * is closed. This action applies to the page being closed,
 * and is executed before any other page opened.
 *
 * @return The C entry of page object's additional actions dictionary.
 */
public PageAdditionalActionsDictionary getC()
{
    COSDictionary C = (COSDictionary)actions.getDictionaryObject( "C"
);
    PageAdditionalActionsDictionary retval = null;
    if( C != null )
    {
        retval = new PageAdditionalActionsDictionary( C );
    }
    return retval;
}

```

```

}

/**
 * This will set an action to be performed when the page
 *   * is closed. This action applies to the page being closed,
 * and is executed before any other page opened.
 *
 * @param C The action to be performed.
 */
public void setC( PageAdditionalActionsDictionary C )
    {
        actions.setItem( "C", C );
    }
}

```

5.3.5 PDDocumentCatalogAdditionalActions.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
met:
 *
 * 1. Redistributions of source code must retain the above copyright
notice,
 *   this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
notice,
 *   this list of conditions and the following disclaimer in the
documentation
 *   and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 *   contributors may be used to endorse or promote products derived
from this
 *   software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
BE LIABLE FOR ANY
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON

```

```
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
*
* http://www.pdfbox.org
```

```
*
*/
```

```
package org.pdfbox.pdmodel.interactive.action;
```

```
import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
```

```
import org.pdfbox.pdmodel.common.COSObjectable;
```

```
/**
```

```
* This class represents a document catalog's dictionary of actions
* that occur due to events.
```

```
*
* @authors Ben Litchfield (ben@csh.rit.edu)
*          Panagiotis Toumasis (ptoumasis@mail.gr)
```

```
* @version $Revision: 1.1 $
```

```
*/
```

```
public class PDDocumentCatalogAdditionalActions implements
COSObjectable
```

```
{
```

```
    private COSDictionary actions;
```

```
    /**
```

```
     * Default constructor.
```

```
    */
```

```
    public PDDocumentCatalogAdditionalActions()
```

```
    {
```

```
        actions = new COSDictionary();
```

```
    }
```

```
    /**
```

```
     * Constructor.
```

```
     *
```

```
     * @param a The action dictionary.
```

```
    */
```

```
    public PDDocumentCatalogAdditionalActions( COSDictionary a )
```

```
    {
```

```
        actions = a;
```

```
    }
```

```
    /**
```

```
     * Convert this standard java object to a COS object.
```

```
     *
```

```
     * @return The cos object that matches this Java object.
```

```
    */
```

```

public COSBase getCOSObject()
{
    return actions;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSDictionary getCOSDictionary()
{
    return actions;
}

/**
 * This will get a JavaScript action to be performed
 * before closing a document.
 * The name DC stands for "document close".
 *
 * @return The DC entry of document catalog's additional actions
dictionary.
 */
public DocumentCatalogAdditionalActionsDictionary getDC()
{
    COSDictionary DC = (COSDictionary)actions.getDictionaryObject(
"DC" );
    DocumentCatalogAdditionalActionsDictionary retval = null;
    if( DC != null )
    {
        retval = new DocumentCatalogAdditionalActionsDictionary( DC );
    }
    return retval;
}

/**
 * This will set a JavaScript action to be performed
 * before closing a document.
 * The name DC stands for "document close".
 *
 * @param DC The action to be performed.
 */
public void setDC( DocumentCatalogAdditionalActionsDictionary DC )
{
    actions.setItem( "DC", DC );
}

/**
 * This will get a JavaScript action to be performed
 * before saving a document.
 * The name WS stands for "will save".
 *

```

```

    * @return The WS entry of document catalog's additional actions
dictionary.
    */
    public DocumentCatalogAdditionalActionsDictionary getWS()
    {
        COSDictionary WS = (COSDictionary)actions.getDictionaryObject(
"WS" );
        DocumentCatalogAdditionalActionsDictionary retval = null;
        if( WS != null )
        {
            retval = new DocumentCatalogAdditionalActionsDictionary( WS );
        }
        return retval;
    }

/**
 * This will set a JavaScript action to be performed
 * before saving a document.
 * The name WS stands for "will save".
 *
 * @param WS The action to be performed.
 */
    public void setWS( DocumentCatalogAdditionalActionsDictionary WS )
    {
        actions.setItem( "WS", WS );
    }

/**
 * This will get a JavaScript action to be performed
 * after saving a document.
 * The name DS stands for "did save".
 *
 * @return The DS entry of document catalog's additional actions
dictionary.
    */
    public DocumentCatalogAdditionalActionsDictionary getDS()
    {
        COSDictionary DS = (COSDictionary)actions.getDictionaryObject(
"DS" );
        DocumentCatalogAdditionalActionsDictionary retval = null;
        if( DS != null )
        {
            retval = new DocumentCatalogAdditionalActionsDictionary( DS );
        }
        return retval;
    }

/**
 * This will set a JavaScript action to be performed
 * after saving a document.
 * The name DS stands for "did save".
 *

```

```

    * @param DS The action to be performed.
    */
    public void setDS( DocumentCatalogAdditionalActionsDictionary DS )
    {
        actions.setItem( "DS", DS );
    }

    /**
     * This will get a JavaScript action to be performed
     * before printing a document.
     * The name WP stands for "will print".
     *
     * @return The WP entry of document catalog's additional actions
    dictionary.
    */
    public DocumentCatalogAdditionalActionsDictionary getWP()
    {
        COSDictionary WP = (COSDictionary)actions.getDictionaryObject(
"WP" );
        DocumentCatalogAdditionalActionsDictionary retval = null;
        if( WP != null )
        {
            retval = new DocumentCatalogAdditionalActionsDictionary( WP );
        }
        return retval;
    }

    /**
     * This will set a JavaScript action to be performed
     * before printing a document.
     * The name WP stands for "will print".
     *
     * @param WP The action to be performed.
     */
    public void setWP( DocumentCatalogAdditionalActionsDictionary WP )
    {
        actions.setItem( "WP", WP );
    }

    /**
     * This will get a JavaScript action to be performed
     * after printing a document.
     * The name DP stands for "did print".
     *
     * @return The DP entry of document catalog's additional actions
    dictionary.
    */
    public DocumentCatalogAdditionalActionsDictionary getDP()
    {
        COSDictionary DP = (COSDictionary)actions.getDictionaryObject(
"DP" );
        DocumentCatalogAdditionalActionsDictionary retval = null;

```

```

        if( DP != null )
        {
            retval = new DocumentCatalogAdditionalActionsDictionary( DP );
        }
        return retval;
    }

    /**
     * This will set a JavaScript action to be performed
     * after printing a document.
     * The name DP stands for "did print".
     *
     * @param DP The action to be performed.
     */
    public void setDP( DocumentCatalogAdditionalActionsDictionary DP )
    {
        actions.setItem( "DP", DP );
    }
}

```

5.3.6 PDGoToAction.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice,
 * this list of conditions and the following disclaimer in the
 * documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 * contributors may be used to endorse or promote products derived
 * from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
 * CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 * PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
 * BE LIABLE FOR ANY

```



```

* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*

```

```

* http://www.pdfbox.org
*
*/

```

```

package org.pdfbox.pdmodel.interactive.action;

```

```

import java.util.ArrayList;
import java.util.List;

```

```

import org.pdfbox.cos.COSArray;
import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;

```

```

import org.pdfbox.pdmodel.common.COSObjectable;
import org.pdfbox.pdmodel.common.COSArrayList;

```

```

/**
 * This represents a go-to action that can be executed in a PDF document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */

```

```

public class PDGoToAction extends PDAction

```

```

{
    private COSDictionary action;

```

```

    /**
     * Default constructor.
     */
    public PDGoToAction()
    {
        action = new COSDictionary();
    }

```

```

    /**
     * Constructor.
     *
     * @param a The action dictionary.
     */

```

```

public PDGoToAction( COSDictionary a )
{
    action = a;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSBase getCOSObject()
{
    return action;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSDictionary getCOSDictionary()
{
    return action;
}

/**
 * This will get the type of action that the actions dictionary
describes.
 * It must be GoTo for a go-to action.
 *
 * @return The S entry of the specific go-to action dictionary.
 */
public String getS()
{
    return action.getNameAsString( "S" );
}

/**
 * This will set the type of action that the actions dictionary
describes.
 * It must be GoTo for a go-to action.
 *
 * @param S The go-to action.
 */
public void setS( String S )
{
    action.setName( "S", S );
}

/**
 * This will get the destination to jump to.
 *

```

```

    * @return The D entry of the specific go-to action dictionary.
    */

    // In case the value is a name.
    public String getD()
    {
        return action.getNameAsString( "D" );
    }

    // In case the value is a string.
    public String getD()
    {
        return action.getString( "D" );
    }

    // In case the value is an array.
    public COSArray getD()
    {
        return (COSArray)action.getDictionaryObject( "D" );
    }

    /**
     * This will set the destination to jump to.
     *
     * @param D The destination.
     */

    // In case the value is a name.
    public void setD( String D )
    {
        action.setName( "D", D );
    }

    // In case the value is a string.
    public void setD( String D )
    {
        action.setString( "D", D );
    }

    // In case the value is an array.
    public void setD( COSArray D )
    {
        action.setItem( "D", D );
    }
}

```

5.3.7 PDRemoteGoToAction.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.

```

*
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 met:
 *
 * 1. Redistributions of source code must retain the above copyright
 notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 notice,
 * this list of conditions and the following disclaimer in the
 documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 * contributors may be used to endorse or promote products derived
 from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
 CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
 BE LIABLE FOR ANY
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
 GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 * <http://www.pdfbox.org>
 *
 */
 package org.pdfbox.pdmodel.interactive.action;

```
import java.util.ArrayList;
import java.util.List;
```

```
import org.pdfbox.cos.COSArray;
import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;
```

```
import org.pdfbox.pdmodel.common.COSObjectable;
```

```

import org.pdfbox.pdmodel.common.COSArrayList;

/**
 * This represents a remote go-to action that can be executed in a PDF
 document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */
public class PDRemoteGoToAction extends PDAction
{
    private COSDictionary action;

    /**
     * Default constructor.
     */
    public PDRemoteGoToAction()
    {
        action = new COSDictionary();
    }

    /**
     * Constructor.
     *
     * @param a The action dictionary.
     */
    public PDRemoteGoToAction( COSDictionary a )
    {
        action = a;
    }

    /**
     * Convert this standard java object to a COS object.
     *
     * @return The cos object that matches this Java object.
     */
    public COSBase getCOSObject()
    {
        return action;
    }

    /**
     * Convert this standard java object to a COS object.
     *
     * @return The cos object that matches this Java object.
     */
    public COSDictionary getCOSDictionary()
    {
        return action;
    }
}

```

```

/**
 * This will get the type of action that the actions dictionary
describes.
 * It must be GoToR for a remote go-to action.
 *
 * @return The S entry of the specific remote go-to action
dictionary.
 */
public String getS()
{
    return action.getNameAsString( "S" );
}

/**
 * This will set the type of action that the actions dictionary
describes.
 * It must be GoToR for a remote go-to action.
 *
 * @param S The remote go-to action.
 */
public void setS( String S )
{
    action.setName( "S", S );
}

/**
 * This will get the file in which the destination is located.
 *
 * @return The F entry of the specific remote go-to action dictionary.
 */
public PDFFileSpecification getFile()
{
    return PDFFileSpecification.createFS( action.getDictionaryObject( "F" )
);
}

/**
 * This will set the file in which the destination is located.
 *
 * @param fs The file specification.
 */
public void setFile( PDFFileSpecification fs )
{
    action.setItem( "F", fs );
}

/**
 * This will get the destination to jump to.
 * If the value is an array defining an explicit destination,
 * its first element must be a page number within the remote
 * document rather than an indirect reference to a page object
 * in the current document. The first page is numbered 0.

```

```

*
* @return The D entry of the specific remote go-to action dictionary.
*/

// In case the value is a name.
public String getD()
{
    return action.getNameAsString( "D" );
}

// In case the value is a string.
public String getD()
{
    return action.getString( "D" );
}

// In case the value is an array.
public COSArray getD()
{
    return (COSArray)action.getDictionaryObject( "D" );
}

/**
 * This will set the destination to jump to.
 * If the value is an array defining an explicit destination,
 * its first element must be a page number within the remote
 * document rather than an indirect reference to a page object
 * in the current document. The first page is numbered 0.
 *
 * @param D The destination.
 */

    // In case the value is a name.
public void setD( String D )
{
    action.setName( "D", D );
}

// In case the value is a string.
public void setD( String D )
{
    action.setString( "D", D );
}

// In case the value is an array.
public void setD( COSArray D )
{
    action.setItem( "D", D );
}

/**

```

```

    * This will specify whether to open the destination document in a new
window.
    * If this flag is false, the destination document will replace the current
    * document in the same window. If this entry is absent, the viewer
application
    * should behave in accordance with the current user preference.
    *
    * @return A flag specifying whether to open the destination document
in a new window.
    */
    public boolean shouldOpenInNewWindow()
    {
        return action.getBoolean( "NewWindow", true );
    }

/**
 * This will specify the destination document to open in a new window.
 *
 * @param value The flag value.
 */
    public void OpenInNewWindow( boolean value )
    {
        action.setItem( "NewWindow", value );
    }
}

```

5.3.8 PDLaunchAction.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
met:
 *
 * 1. Redistributions of source code must retain the above copyright
notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
notice,
 * this list of conditions and the following disclaimer in the
documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of pdfbox; nor the names of its
 * contributors may be used to endorse or promote products derived
from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"

```


* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

*
* http://www.pdfbox.org
*
*/
package org.pdfbox.pdmodel.interactive.action;

```

```

import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;

import org.pdfbox.pdmodel.common.COSObjectable;

```

```

/**
 * This represents a launch action that can be executed in a PDF document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */
public class PDLaunchAction extends PDAction
{
    private COSDictionary action;

    /**
     * Default constructor.
     */
    public PDLaunchAction()
    {
        action = new COSDictionary();
    }

    /**
     * Constructor.
     */

```

```

    * @param a The action dictionary.
    */
public PDLaunchAction( COSDictionary a )
{
    action = a;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSBase getCOSObject()
{
    return action;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSDictionary getCOSDictionary()
{
    return action;
}

/**
 * This will get the type of action that the actions dictionary
describes.
 * It must be Launch for a launch action.
 *
 * @return The S entry of the specific launch action dictionary.
 */
public String getS()
{
    return action.getNameAsString( "S" );
}

/**
 * This will set the type of action that the actions dictionary
describes.
 * It must be Launch for a launch action.
 *
 * @param S The launch action.
 */
public void setS( String S )
{
    action.setName( "S", S );
}

/**

```

```

* This will get the application to be launched or the document
* to be opened or printed. It is required if none of the entries
* Win, Mac or Unix is present. If this entry is absent and the
* viewer application does not understand any of the alternative
* entries it should do nothing.
*
* @return The F entry of the specific launch action dictionary.
*/
public PDFFileSpecification getFile()
{
    return PDFFileSpecification.createFS( action.getDictionaryObject( "F" )
);
}

/**
* This will set the application to be launched or the document
* to be opened or printed. It is required if none of the entries
* Win, Mac or Unix is present. If this entry is absent and the
* viewer application does not understand any of the alternative
* entries it should do nothing.
*
* @param fs The file specification.
*/
public void setFile( PDFFileSpecification fs )
{
    action.setItem( "F", fs );
}

/**
* This will get a dictionary containing Windows-specific launch
parameters.
*
* @return The Win entry of of the specific launch action dictionary.
*/
public AdditionalLaunchActionDictionary getWin()
{
    COSDictionary Win = (COSDictionary)action.getDictionaryObject(
"Win" );
    AdditionalLaunchActionDictionary retval = null;
    if( Win != null )
    {
        retval = new AdditionalLaunchActionDictionary( Win );
    }
    return retval;
}

/**
* This will set a dictionary containing Windows-specific launch
parameters.
*
* @param Win The action to be performed.
*/

```

```

public void setWin( AdditionalLaunchActionDictionary Win )
    {
        action.setItem( "Win", Win );
    }

/**
 * This will get the file name to be launched or the document to be
opened
 * or printed, in standard Windows pathname format. If the name
string includes
 * a backslash character (\), the backslash must itself be preceded by a
backslash.
 * This value must be a single string; it is not a file specification.
 *
 * @return The F entry of the specific Windows launch parameter
dictionary.
 */
public String getF()
    {
        return action.getString( "F" );
    }

/**
 * This will set the file name to be launched or the document to be
opened
 * or printed, in standard Windows pathname format. If the name
string includes
 * a backslash character (\), the backslash must itself be preceded
by a backslash.
 * This value must be a single string; it is not a file specification.
 *
 * @param F The file name to be launched.
 */
public void setF( String F )
    {
        action.setString( "F", F );
    }

/**
 * This will get the string specifying the default directory in standard
DOS syntax.
 *
 * @return The D entry of the specific Windows launch parameter
dictionary.
 */
public String getD()
    {
        return action.getString( "D" );
    }

/**

```

```

    * This will set the string specifying the default directory in standard
DOS syntax.
    *
    * @param D The default directory.
    */
public void setD( String D )
    {
        action.setString( "D", D );
    }

/**
 * This will get the string specifying the operation to perform:
 * open to open a document
 * print to print a document
 * If the F entry designates an application instead of a document, this
entry
 * is ignored and the application is launched. Default value: open.
 *
 * @return The O entry of the specific Windows launch parameter
dictionary.
 */
public String getO()
    {
        return action.getString( "O" );
    }

/**
 * This will set the string specifying the operation to perform:
 * open to open a document
 * print to print a document
 * If the F entry designates an application instead of a document,
this entry
 * is ignored and the application is launched. Default value: open.
 *
 * @param O The operation to perform.
 */
public void setO( String O )
    {
        action.setString( "O", O );
    }

/**
 * This will get a parameter string to be passed to the application
designated by the F entry.
 * This entry should be omitted if F designates a document.
 *
 * @return The P entry of the specific Windows launch parameter
dictionary.
 */
public String getP()
    {
        return action.getString( "P" );
    }

```

```

    }

    /**
     * This will set a parameter string to be passed to the application
    designated by the F entry.
     * This entry should be omitted if F designates a document.
     *
     * @param P The parameter string.
     */
    public void setP( String P )
    {
        action.setString( "P", P );
    }

    /**
     * This will specify whether to open the destination document in a new
    window.
     * If this flag is false, the destination document will replace the current
     * document in the same window. If this entry is absent, the viewer
    application
     * should behave in accordance with the current user preference. This
    entry is
     * ignored if the file designated by the F entry is not a PDF document.
     *
     * @return A flag specifying whether to open the destination document
    in a new window.
     */
    public boolean shouldOpenInNewWindow()
    {
        return action.getBoolean( "NewWindow", true );
    }

    /**
     * This will specify the destination document to open in a new window.
     *
     * @param value The flag value.
     */
    public void OpenInNewWindow( boolean value )
    {
        action.setItem( "NewWindow", value );
    }
}

```

5.3.9 PDThreadAction.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
    met:

```

```

*
* 1. Redistributions of source code must retain the above copyright
notice,
*   this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
notice,
*   this list of conditions and the following disclaimer in the
documentation
*   and/or other materials provided with the distribution.
* 3. Neither the name of pdfbox; nor the names of its
*   contributors may be used to endorse or promote products derived
from this
*   software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* http://www.pdfbox.org
*
*/
package org.pdfbox.pdmodel.interactive.action;

import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;

import org.pdfbox.pdmodel.common.COSObjectable;

/**
 * This represents a thread action that can be executed in a PDF
document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $

```

```

*/
public class PDThreadAction extends PDAction
{
    private COSDictionary action;

    /**
     * Default constructor.
     */
    public PDThreadAction()
    {
        action = new COSDictionary();
    }

    /**
     * Constructor.
     *
     * @param a The action dictionary.
     */
    public PDThreadAction( COSDictionary a )
    {
        action = a;
    }

    /**
     * Convert this standard java object to a COS object.
     *
     * @return The cos object that matches this Java object.
     */
    public COSBase getCOSObject()
    {
        return action;
    }

    /**
     * Convert this standard java object to a COS object.
     *
     * @return The cos object that matches this Java object.
     */
    public COSDictionary getCOSDictionary()
    {
        return action;
    }

    /**
     * This will get the type of action that the actions dictionary
describes.
     * It must be Thread for a thread action.
     *
     * @return The S entry of the specific thread action dictionary.
     */
    public String getS()
    {

```



```

        return action.getNameAsString( "S" );
    }

    /**
     * This will set the type of action that the actions dictionary
describes.
     * It must be Thread for a thread action.
     *
     * @param S The thread action.
     */
    public void setS( String S )
    {
        action.setName( "S", S );
    }

    /**
     * This will get the file containing the desired thread.
     * If this entry is absent, the thread is in the current file.
     *
     * @return The F entry of the specific thread action dictionary.
     */
    public PDFFileSpecification getFile()
    {
        return PDFFileSpecification.createFS( action.getDictionaryObject( "F" )
);
    }

    /**
     * This will set the file containing the desired thread.
     * If this entry is absent, the thread is in the current file.
     *
     * @param fs The file specification.
     */
    public void setFile( PDFFileSpecification fs )
    {
        action.setItem( "F", fs );
    }

    /**
     * This will get the desired destination thread, specified in one of the
following forms:
     * 1) An indirect reference to a thread dictionary.
     * In this case, the thread must be in the current file.
     * 2) The index of the thread within the Threads array of its document's
catalog.
     * The first thread in the array has index 0.
     * 3) The title of the thread, as specified in its thread information
dictionary.
     * If two or more threads have the same title, the one appearing first in
the document
     * catalog's Threads array will be used.
     *

```

```

    * @return The D entry of of the specific thread action dictionary.
    */

    // In case the value is a dictionary.
    public AdditionalThreadActionDictionary getD()
    {
        COSDictionary D =
(COSDictionary)action.getDictionaryObject( "D" );
        AdditionalThreadActionDictionary retval = null;
        if( D != null )
        {
            retval = new AdditionalThreadActionDictionary( D );
        }
        return retval;
    }

    //In case the value is an integer.
    public Integer getD()
    {
        Integer retval = null;
        COSNumber D = (COSNumber)action.getDictionaryObject( "D" );
        if( D != null )
        {
            retval = new Integer( D.intValue() );
        }
        return retval;
    }

    //In case the value is a text string.
    public String getThreadDestination()
    {
        return action.getString( "D" );
    }

    /**
    * This will set the desired destination thread, specified in one of the
    following forms:
    * 1) An indirect reference to a thread dictionary.
    * In this case, the thread must be in the current file.
    * 2) The index of the thread within the Threads array of its document's
    catalog.
    * The first thread in the array has index 0.
    * 3) The title of the thread, as specified in its thread information
    dictionary.
    * If two or more threads have the same title, the one appearing first in
    the document
    * catalog's Threads array will be used.
    *
    * @param D The desired destination thread.
    */

    // In case the value is a dictionary.

```

```

public void setD( AdditionalThreadActionDictionary D )
    {
        action.setItem( "D", D );
    }

//In case the value is an integer.
public void setD( Integer D )
    {
        COSInteger value = null;
        if( D != null )
            {
                value = new COSInteger( D.intValue() );
            }
        action.setItem( "D", value );
    }

//In case the value is a text string.
public void setThreadDestination( String destination )
    {
        action.setString( "D", destination );
    }

/**
 * This will get the desired bead in the destination thread, specified in
one of the following forms:
 * 1) An indirect reference to a bead dictionary.
 * In this case, the thread must be in the current file.
 * 2) The index of the bead within its thread.
 * The first thread in the array has index 0.
 *
 * @return The B entry of of the specific thread action dictionary.
 */

// In case the value is a dictionary.
public AdditionalThreadActionDictionary getB()
    {
        COSDictionary B = (COSDictionary)action.getDictionaryObject(
"B" );
        AdditionalThreadActionDictionary retval = null;
        if( B != null )
            {
                retval = new AdditionalThreadActionDictionary( B );
            }
        return retval;
    }

//In case the value is an integer.
public Integer getB()
    {
        Integer retval = null;
        COSNumber B = (COSNumber)action.getDictionaryObject( "B" );
        if( B != null )

```

```

        {
            retval = new Integer( B.intValue() );
        }
        return retval;
    }

    /**
     * This will set the desired bead in the destination thread, specified in
     one of the following forms:
     * 1) An indirect reference to a bead dictionary.
     * In this case, the thread must be in the current file.
     * 2) The index of the bead within its thread.
     * The first thread in the array has index 0.
     *
     * @param B The desired bead thread.
     */

    // In case the value is a dictionary.
    public void setB( AdditionalThreadActionDictionary B )
        {
            action.setItem( "B", B );
        }

    //In case the value is an integer.
    public void setB( Integer B )
        {
            COSInteger value = null;
            if( B != null )
                {
                    value = new COSInteger( B.intValue() );
                }
            action.setItem( "B", value );
        }
    }
}

```

5.3.10 PDURIAction.java Class

```

/**
 * Copyright (c) 2004, www.pdfbox.org
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 met:
 *
 * 1. Redistributions of source code must retain the above copyright
 notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 notice,
 * this list of conditions and the following disclaimer in the
 documentation

```

```

* and/or other materials provided with the distribution.
* 3. Neither the name of pdfbox; nor the names of its
* contributors may be used to endorse or promote products derived
from this
* software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* http://www.pdfbox.org
*
*/
package org.pdfbox.pdmodel.interactive.action;

import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;

import org.pdfbox.pdmodel.common.COSObjectable;

/**
 * This represents a URI action that can be executed in a PDF document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.1 $
 */
public class PDURIAction extends PDAction
{
    private COSDictionary action;

    /**
     * Default constructor.
     */
    public PDURIAction()

```

```

    {
        action = new COSDictionary();
    }

/**
 * Constructor.
 *
 * @param a The action dictionary.
 */
public PDURIAction( COSDictionary a )
{
    action = a;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSBase getCOSObject()
{
    return action;
}

/**
 * Convert this standard java object to a COS object.
 *
 * @return The cos object that matches this Java object.
 */
public COSDictionary getCOSDictionary()
{
    return action;
}

/**
 * This will get the type of action that the actions dictionary
describes.
 * It must be URI for a URI action.
 *
 * @return The S entry of the specific URI action dictionary.
 */
public String getS()
{
    return action.getNameAsString( "S" );
}

/**
 * This will set the type of action that the actions dictionary
describes.
 * It must be URI for a URI action.
 *
 * @param S The URI action.

```

```

        */
        public void setS( String S )
    {
        action.setName( "S", S );
    }

    /**
     * This will get the uniform resource identifier to resolve, encoded in 7-
    bit ASCII.
     *
     * @return The URI entry of the specific URI action dictionary.
     */
    public String getURI()
    {
        return action.getString( "URI" );
    }

    /**
     * This will set the uniform resource identifier to resolve, encoded in 7-
    bit ASCII.
     *
     * @param URI The uniform resource identifier.
     */
    public void setURI( String URI )
    {
        action.setString( "URI", URI );
    }

    /**
     * This will specify whether to track the mouse position when the URI is
    resolved.
     * Default value: false.
     * This entry applies only to actions triggered by the user's clicking an
    annotation;
     * it is ignored for actions associated with outline items or with a
    document's OpenAction entry.
     *
     * @return A flag specifying whether to track the mouse position when
    the URI is resolved.
     */
    public boolean shouldTrackMousePosition()
    {
        return action.getBoolean( "MousePosition", true );
    }

    /**
     * This will specify whether to track the mouse position when the URI is
    resolved.
     *
     * @param value The flag value.
     */
    public void TrackMousePosition( boolean value )

```

```

    {
        action.setItem( "MousePosition", value );
    }

/**
 * This will get the base URI to be used in resolving relative URI
references.
 * URI actions within the document may specify URIs in partial form, to
be interpreted
 * relative to this base address. If no base URI is specified, such partial
URIs
 * will be interpreted relative to the location of the document itself.
 * The use of this entry is parallel to that of the body element <BASE>,
as described
 * in the HTML 4.01 Specification.
 *
 * @return The URI entry of the specific URI dictionary.
 */
public String getBase()
    {
        return action.getString( "Base" );
    }

/**
 * This will set the base URI to be used in resolving relative URI
references.
 * URI actions within the document may specify URIs in partial
form, to be interpreted
 * relative to this base address. If no base URI is specified, such
partial URIs
 * will be interpreted relative to the location of the document itself.
 * The use of this entry is parallel to that of the body element
<BASE>, as described
 * in the HTML 4.01 Specification.
 *
 * @param URI The the base URI to be used.
 */
public void setBase( String Base )
    {
        action.setString( "Base", Base );
    }
}

```

5.4 ΠΑΡΑΔΟΤΕΑ ΚΑΙ ΠΑΡΟΥΣΙΑΣΕΙΣ ΚΑΤΑ ΤΗ ΔΙΑΡΚΕΙΑ ΤΟΥ ΕΞΑΜΗΝΟΥ

5.4.1 Team Form

Team name: [Phaethon](#)

Team web site URL: _____
(optional)

Team Member 1

(Contact person)

Name: [Panagiotis](#)

Surname: [Toumasis](#)

ID (AM): [8000131](#)

email: ptoumasis@mail.gr

Date: [31/3/2004](#) Signature:



Team Member 2

(Optional)

Name: _____

Surname: _____

ID (AM): _____

email: _____

Date: _____

Signature: _____

5.4.2 Project Form

Team name: [Phaethon](#)

Project name: [PDFBox](#)

Project web site URL: <http://sourceforge.net/projects/pdfbox>

Implementation language: [Java](#)

Platform: [OS Independent](#)

License: [BSD Licence](#)

Development team size: [2](#)
(Estimate)

Approximate source code size:
(Use a measure you consider appropriate)

30 packages, 139 classes, more than 10000 lines of java code

Short description of the project:

PDFBox is a Java PDF Library. This project will allow access to all of the components in a PDF document. More PDF manipulation features will be added as the project matures. This ships with a utility to take a PDF document and output a text file.

Short description of your planned contribution (non-binding):

Debugging and including some viewing capabilities (which now can only be used for debug purposes) in the PDF Viewer (a GUI application) that show the internal structure of a PDF document.

Date: 19/4/2004

Signature:



5.4.3 Presentation 1: Project (21/4/2004)



ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY



Maintenance and Reengineering of Information Systems

Presentation of the Project

21/4/2004

Maintenance and Reengineering
of Information Systems

 Undergraduate Student P. Tourmasis 

1

Team



Team name: [Phaethon](#)

Team Members: 1

Name: [Panagiotis](#)

Surname: [Toumasis](#)

ID (AM): [8000131](#)

email: ptoumasis@mail.gr

21/4/2004

Maintenance and Reengineering
of Information Systems

 Undergraduate Student P. Toumasis 

2

Project Details (1)



Project name: [PDFBox](#)

Project web site URL: <http://sourceforge.net/projects/pdfbox>,
www.pdfbox.org

Topic: [Dynamic Content](#), [Indexing/Search](#), [Text Processing](#)

Development Status: [4 – Beta](#)

Implementation - Programming language: [Java](#)

Platform: [OS Independent](#)

21/4/2004

Maintenance and Reengineering
of Information Systems

 Undergraduate Student P. Toumasis 

3

Project Details (2)



Intended Audience: [Developers](#)

License: [BSD Licence](#) (similar to [MIT Licence](#))

Development team size: [2](#) ([Ben Litchfield](#), [Bob Dickinson](#))

Approximate source code size:

[30 packages](#), [139 classes](#),
[more than 10000 lines of java code](#)

Registration Date: [7/4/2003](#)

21/4/2004

Maintenance and Reengineering
of Information Systems

 [Undergraduate Student P. Tourmasis](#) 

4

Project Details (3)



Short description of the project:

- Ⓒ [PDFBox](#) is a [Java PDF Library](#).
- Ⓒ This project will allow access to all of the components in a PDF document.
- Ⓒ More PDF manipulation features will be added as the project matures.
- Ⓒ It ships with a utility to take a PDF document and output a text file.

21/4/2004

Maintenance and Reengineering
of Information Systems

 [Undergraduate Student P. Tourmasis](#) 

5

Project Versions



©PDFBox – 0.6.1

Released 11/4/2003, 7:19, Ben Litchfield

©PDFBox – 0.6.2

Released 18/4/2003, 21:40, Bob Dickinson

©PDFBox – 0.6.3

Released 14/9/2003, 6:57, Ben Litchfield

©PDFBox – 0.6.5

Released 8/3/2004, 19:28, Ben Litchfield

21/4/2004

Maintenance and Reengineering
of Information Systems

Undergraduate Student P. Toumasis

6

Some Bugs



Request ID	Summary	Date	Assigned to	Submitted by
927793	org.pdfbox.encryption.DecryptDocument	1/4/2004, 12:15	nobody	Pcwizard_01
926998	Class Cast Exception During appendDocument	31/3/2004, 13:07	berlitchfield	valnifj
921378	IOException: expected='endobj'	22/3/2004, 13:18	nobody	nobody
921373	No 'ToUnicode' and no 'Encoding' for Font	22/3/2004, 13:11	nobody	nobody
920900	PDFTextStripper hangs on extraction	22/3/2004, 2:48	nobody	Hal_9000
919215	IOException in LucenePDFDocument.getDocument	18/3/2004, 17:40	nobody	nobody

21/4/2004

Maintenance and Reengineering
of Information Systems

Undergraduate Student P. Toumasis

7

My contribution (non-binding)



Short description of my planned contribution (non-binding):

- ⊙ Debugging (some of the 21 bugs, if possible!).
- ⊙ Including some viewing capabilities in the PDF Viewer (a GUI application) that show the internal structure of a PDF document.

[This is a task that the members of the team want to add to one of the next versions of the PDFbox.]

21/4/2004

Maintenance and Reengineering
of Information Systems

Undergraduate Student P. Tournasis

8

5.4.4 Presentation 2: Design (10/5/2004)



ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY



Maintenance and Refactoring of Information Systems

Design of my planned contribution to the PDFBox Project

10/5/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Tournasis

1

Communication with Mr. Litchfield (1)



Me: ...I am writing to you asking for your advice and help...if I could commit some useful changes to your project... Have you any ideas?... what new elements do you want to add?...

Mr. Litchfield: ...I would love some help...A couple that would be pretty interesting would be:

- @Extract Images
- @support color space
- @PDF->HTML conversion
- @HTML->PDF conversion
- @adding support for PDF 1.5 object streams
- @FDF export/import

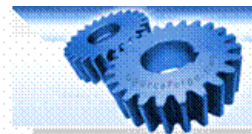
10/5/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

2

Communication with Mr. Litchfield (2)



Mr. Litchfield:

...I have started some work on a PDFViewer.... I would like to have it be able to change values and write the PDF back to disk...

...I would also be interested in creating an Acrobat Reader like application in Java...**This is no way you could complete this** but if you started it maybe other people could work on it as well...

...There are some people asking for a PDF->XML as well, **that might be a little easier** than PDF-> HTML to implement. It would need to contain font/positioning information...

10/5/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

3

My Final Decision



Conversion of a PDF document to XML or HTML

10/5/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

4

Class PDFTextStripper (1)



Class PDFTextStripper

org.pdfbox.util

Hierarchy:

Java.lang.Object

|

+--org.pdfbox.util.PDFStreamEngine

|

+--org.pdfbox.util.PDFTextStripper

@public class PDFTextStripper extends PDFStreamEngine

@Version: \$Revision: 1.33 \$

@Author: Ben Litchfield (ben@csh.rit.edu)

@PDFTextStripper: 555 lines of code

@PDFStreamEngine: 864 lines of code

10/5/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

5

Class PDFTextStripper (2)



©The `org.pdfbox.util.PDFTextStripper` is the class that extracts the text out of the PDF.

©This class takes a PDF document, strips out all of the text and ignores the formatting.

©This class give us only the text and font information.

©There is no image extraction right now so we have to ignore images.

Official documentation:

This class runs through a PDF content stream, executes certain operations and provides a callback interface for clients that want to do things with the stream.

10/5/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Tournas

6

Method Summary (1)



©**Methods inherited from class `org.pdfbox.util.PDFStreamEngine`:**

protected void `processOperator`(PDFOperator operator, List arguments)

This is used to handle an operation.

void `processStream`(COSStream cosStream, Map fontMap)

This will process the contents of the stream.

©**Methods inherited from class `java.lang.Object`:**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

©protected void `flushText`()

This will print the text to the output stream.

©int `getEndPage`()

This will get the last page that will be extracted.

10/5/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Tournas

7

Method Summary (2)



©String `getLineSeparator()`

This will get the line separator.

©String `getPageSeparator()`

This will get the page separator.

©int `getStartPage()`

This is the page that the text extraction will start on.

©String `getText(COSDocument doc)`

Deprecated.

©String `getText(PDDocument doc)`

This will return the text of a document.

©void `writeText(COSDocument doc, Writer outputStream)`

Deprecated.

©void `writeText(PDDocument doc, Writer outputStream)`

This will take a PDDocument and write the text of that document to the print writer.

10/5/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Toumasis

8

Method Summary (3)



©protected void `processPage(PDPage page, COSStream content)`

This will process the contents of a page.

©protected void `processPages(List pages)`

This will process all of the pages and the text that is in them.

©void `setEndPage(int endPage)`

This will set the last page to be extracted by this class.

©void `setLineSeparator(String separator)`

Set the desired line separator for output text.

©void `setPageSeparator(String separator)`

Set the desired page separator for output text.

©void `setStartPage(int startPage)`

This will set the first page to be extracted by this class.

©protected TextPosition `showString(byte[] string)`

This will show a string.

←←← (This method will be useful!)

10/5/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Toumasis

9

Method showString



showString

protected `TextPosition` **showString**(byte[] string) throws `IOException`

This will show a string.

Overrides:

`showString` in class `PDFStreamEngine`

Parameters:

string - The string to show.

Returns:

A description of the text being shown.

Throws:

`IOException` - If there is an error showing the string.

10/5/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tournasis 

10

Implementation Problems



Ⓢ I was not able to set my classpath yet, so I have not run the application.

(This is the first problem to solve!)

Ⓢ I know just the basics of XML.

Ⓢ Do I have to create a new class or just to extend the existing `PDFTextStripper`?

Ⓢ Is this work enough for our course?

Ⓢ I don't know how much time I really need!

10/5/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tournasis 

11

5.4.5 Presentation 3: Implementation (8/10/2004)



ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY



Maintenance and Refactoring of Information Systems

Implementation of my planned contribution to the PDFBox Project

8/10/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Tournas

1

Short Description of the PDFBox (1)



Project name: **PDFBox**

Project web site URL:

<http://sourceforge.net/projects/pdfbox>,
www.pdfbox.org

Topic: **Dynamic Content, Indexing/Search, Text
Processing**

Development Status: **4 – Beta**

Implementation - Programming language: **Java**

Platform: **OS Independent**

8/10/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Tournas

2

Short Description of the PDFBox (2)



Administrator: **Ben Litchfield**

Approximate source code size: **30 packages, 139 classes, more than 10000 lines of java code**

Registration Date: **7/4/2003**

Last version: **PDFBox – 0.6.6, Released 20/7/2004**

Short description of the project:

PDFBox is a Java PDF Library.

This project allows access to all of the components in a PDF document.

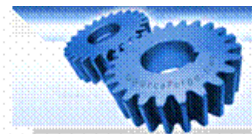
8/10/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

3

My First Decision



Conversion of a PDF document
to XML format

Rejected

8/10/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

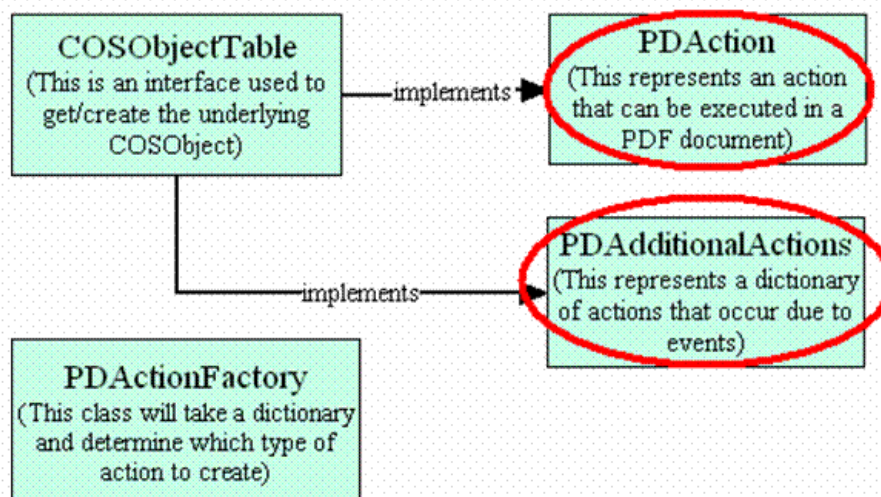
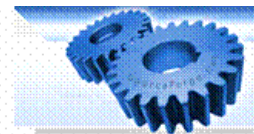
4



Adding Support for PDF Actions

Carried out

Existing classes Hierarchy



New Java Classes



Java Class Name	Summary of Class Function	Code lines	Changed / New
PDAction	This represents an action that can be executed in a PDF document.	184	Changed
PDAnnotationAdditionalActions	This class represents an annotation's dictionary of actions that occur due to events.	379	Changed
PDFFormFieldAdditionalActions	This class represents a form field's dictionary of actions that occur due to events.	215	New
PDPageAdditionalActions	This class represents a page object's dictionary of actions that occur due to events.	149	New
PDDocumentCatalogAdditionalActions	This class represents a document catalog's dictionary of actions that occur due to events.	237	New

8/10/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis

7

New Java Classes



Java Class Name	Summary of Class Function	Code lines	Changed / New
PDGoToAction	This represents a go-to action that can be executed in a PDF document.	163	New
PDRemoteGoToAction	This represents a remote go-to action that can be executed in a PDF document.	214	New
PDLaunchAction	This represents a launch action that can be executed in a PDF document.	283	New
PDThreadAction	This represents a thread action that can be executed in a PDF document.	271	New
PDURIAction	This represents a URI action that can be executed in a PDF document.	182	New
Sum	-	2277	-

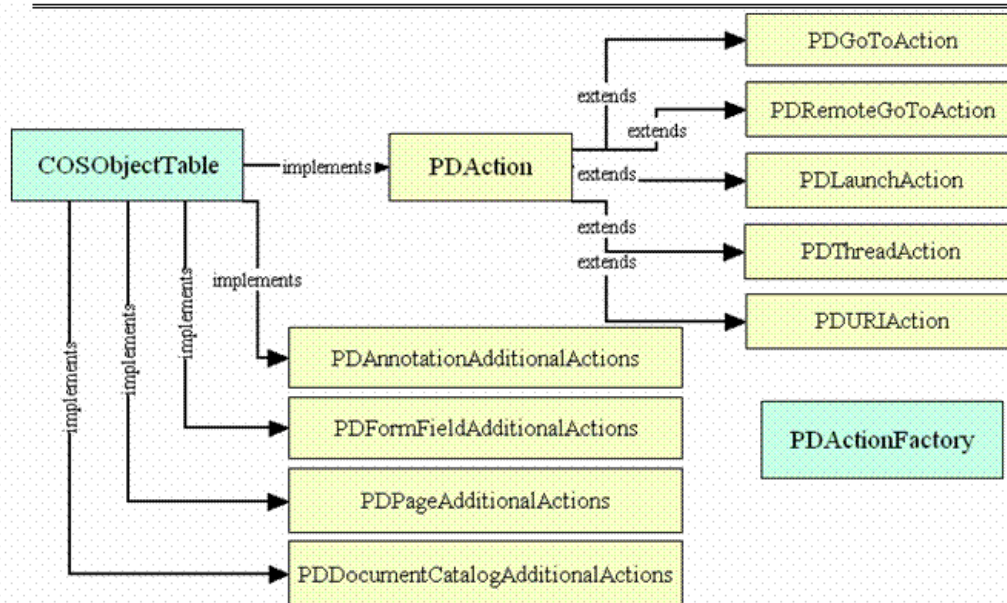
8/10/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis

8

New Classes Hierarchy



8/10/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Toumasis

9

A Java Code Example (1)



```
package org.pdfbox.pdmodel.interactive.action;

import java.util.ArrayList;
import java.util.List;

import org.pdfbox.cos.COSArray;
import org.pdfbox.cos.COSBase;
import org.pdfbox.cos.COSDictionary;
import org.pdfbox.cos.COSStream;

import org.pdfbox.pdmodel.common.COSObjectable;
import org.pdfbox.pdmodel.common.COSArrayList;

/**
 * This represents an action that can be executed in a PDF document.
 *
 * @authors Ben Litchfield (ben@csh.rit.edu)
 *          Panagiotis Toumasis (ptoumasis@mail.gr)
 * @version $Revision: 1.2 $
 */
public class PDACTION implements COSObjectable
```

8/10/2004

Maintenance and Refactoring
of Information Systems

Undergraduate Student P. Toumasis

10

A Java Code Example (2)



```
/**
 * This will get the type of action that the actions dictionary describes.
 * If present, must be Action for an action dictionary.
 *
 * @return The S entry of actions dictionary.
 */
public String getS()
{
    return action.getNameAsString( "S" );
}

/**
 * This will set the type of action that the actions dictionary describes.
 * If present, must be Action for an action dictionary.
 *
 * @param S The new type of action.
 */
public void setS( String S )
{
    action.setName( "S", S );
}
```

8/10/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

11

Cooperation with Mr. Litchfield



- ⊙ Exchange of about a dozen e-mails.
- ⊙ Instant response.
- ⊙ Always eager to help.
- ⊙ Excellent communication.

8/10/2004

Maintenance and Refactoring
of Information Systems

 Undergraduate Student P. Tourmasis 

12

Conclusion



- © One of the most demanding projects I've ever done.

- © I gain a lot of experience which will be useful for the continuation of my studies.

- © I hope this type of project to be implemented in many other lessons in Greece.