
PDFUnit-Monitor

Handbuch für Benutzer

Carsten Siedentop

Inhaltsverzeichnis

Vorwort	4
1. Über diese Dokumentation	5
2. Quickstart	6
3. Funktionalität	10
3.1. Strukturen hinter der Funktionalität	10
3.2. Darstellung - Überwachte Verzeichnisse	14
3.3. Darstellung - Liste der Prüfergebnisse, Filter	17
3.4. Darstellung - Details eines Validierungsergebnisses	20
3.5. DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente	20
3.6. Export und Import der Ergebnisse	22
4. Installation, Update, Deinstallation	24
4.1. Technische Voraussetzungen	24
4.2. Installation ohne vorhandenes iText	24
4.3. Installation bei existierendem iText	26
4.4. Pfade über Systemumgebungsvariablen setzen	26
4.5. Überprüfung der Konfiguration	26
4.6. Installation eines neuen Releases	27
4.7. Deinstallation	28
5. Starten der Anwendung	29
6. Konfiguration (pdfunit-monitor.config)	30
7. Typische Fehler	32
8. Logging mit Log4j2	33
Stichwortverzeichnis	35

Vorwort

Aktuelle Testsituation in Projekten

Telefonrechnungen, Versicherungspolicen, amtliche Bescheide, Verträge jeglicher Art werden heute häufig als PDF-Dokument elektronisch zugestellt. Die Erstellung erfolgt in vielen Programmiersprachen mit zahlreichen Bibliotheken. Je nach Komplexität der zu erstellenden Dokumente ist diese Programmierung nicht einfach. In jedem Prozessschritt auf dem Weg zum PDF können Fehler entstehen:

- Steht auf Seite 2 der richtige Text?
- Erscheint das neue Logo auf allen Dokumenten?
- Sind die Schriften eingebettet, wie beabsichtigt?
- Stimmt das Layout mit der Vorgabe überein?
- Enthält das Dokument die richtige Barcode-Graphik?
- Ist das PDF signiert?

Es sollte Entwickler, Projekt- und Unternehmensverantwortliche erschrecken, dass es bisher kaum Möglichkeiten gibt, PDF-Dokumente **automatisiert** zu testen. Und selbst diese Möglichkeiten werden im Projektalltag nicht genutzt. Manuelles Testen ist leider weit verbreitet. Es ist teuer und selber fehleranfällig.

Es war längst überfällig, ein einfach zu nutzendes Testsystem zu entwickeln.

Egal, ob PDF-Dokumente mit einem mächtigen Design-Werkzeug erstellt, aus MS-Word oder Libre-Office exportiert, über eine API erstellt wurden oder aus einem XSL-FO Workflow herausgefallen sind, mit PDFUnit kann jedes PDF-Dokument getestet werden.

Manuell oder automatisiert?

Das Konzept einer Anwendung, die durch einen Benutzer bedient wird, scheint ein Widerspruch gegen das automatisierte Testen zu sein. In der Praxis ist dieser Widerspruch aber gelöst, denn die Anwendung 'beobachtet' Dateiverzeichnisse und führt Testfälle aus, sobald in den Verzeichnissen neue PDF-Dokumente gespeichert werden. So kann die IT-Abteilung Dokumente erstellen - egal, auf welche Weise - und die Anwendung zeigt unmittelbar das Ergebnis der vorhandenen Tests an, ohne dass die Tests manuell angestoßen werden müssen.

Die Testanwendung ist leicht zu bedienen. Alle Funktionen sind über die rechte Maustaste zu erreichen (Kontextmenü). Zusätzlich sind zahlreiche Funktionen über die Tastatur erreichbar. Testfälle werden in Excel erfasst. Somit können auch Nicht-Entwickler Testfällen entwickeln und ausführen.

Eine ausführliche Beschreibung der verfügbaren Funktionen steht in Kapitel 3: „Funktionalität“ (S. 10).

Kapitel 1. Über diese Dokumentation

Wer sollte sie lesen

Die vorliegende Dokumentation richtet sich an Mitarbeiter der Qualitätssicherung oder anderer Abteilungen, die die Anwendung 'PDFUnit-Monitor' zur Überprüfung von PDF-Dokumenten nutzen.

Wenn es Probleme gibt

Haben Sie Schwierigkeiten, ein PDF zu testen? Recherchieren Sie zuerst im Internet, vielleicht ist dort ein ähnliches Problem schon beschrieben, eventuell mit einer Lösung. Sie können die Problembeschreibung auch per Mail an [problem\[at\]pdfunit.com](mailto:problem[at]pdfunit.com) schicken.

Neue Testfunktionen gewünscht?

Hätten Sie gerne neue Testfunktionen, wenden Sie sich per Mail an [request\[at\]pdfunit.com](mailto:request[at]pdfunit.com). Das Produkt befindet sich permanent in der Weiterentwicklung, die Sie durch Ihre Wünsche gerne beeinflussen dürfen.

Herstellung dieser Dokumentation

Die vorliegende Dokumentation wurde mit DocBook-XML erstellt. Sowohl PDF als auch HTML werden aus einer einzigen Textquelle erstellt. In beiden Zielformaten ist das Layout noch verbesserungswürdig, wie beispielsweise die Seitenumbrüche im PDF-Format. Die Verbesserung des Layouts steht schon auf der Aufgabenliste, jedoch gibt es noch Aufgaben mit höherer Priorität.

Feedback

Jegliche Art von Feedback ist willkommen, schreiben Sie einfach an [feedback\[at\]pdfunit.com](mailto:feedback[at]pdfunit.com).

Kapitel 2. Quickstart

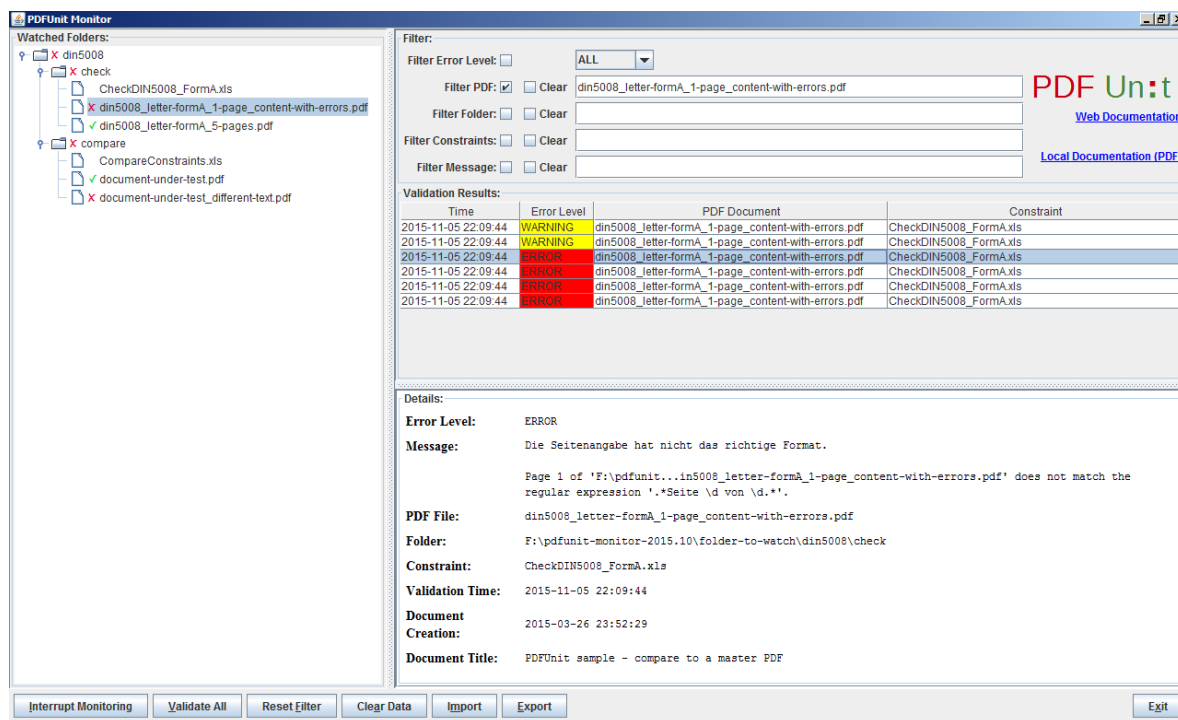
Der PDFUnit-Monitor ist eine graphische Anwendung, um Tests für PDF-Dokumente anzustoßen und das Ergebnis anzeigen zu lassen. Die Zielgruppe für die Anwendung sind Nicht-Programmierer.

Neben der vorliegenden Dokumentation gibt es noch ein erklärendes Video, das den PDFUnit-Monitor in Aktion zeigt. Alle Arten von Dokumentation können über diesen Link (Download) von den Webseiten von PDFUnit heruntergeladen werden.

Überwachte Verzeichnisse

Der PDFUnit-Monitor überwacht alle PDF-Dokumente unterhalb eines definierten Verzeichnisses und prüft die dortigen Dokumente gegen Regeln, die in Excel-Dateien hinterlegt sind. Die Excel-Dateien müssen ebenfalls in den überwachten Verzeichnissen liegen. Erfüllt ein PDF-Dokument alle Regeln, wird es in der Baumstruktur mit einem grünen Haken versehen. Verletzt ein PDF eine oder mehrere Regeln, werden alle Regelverletzungen in eine Übersichtsliste eingetragen. Zusätzlich erhält der Dateiname ein rotes Kreuz. Diese Statusanzeige geht auf die Verzeichnisnamen über. Enthält ein Verzeichnis und all seine Unterverzeichnisse ausschließlich gültige PDF-Dokumente, wird es mit einem grünen Haken dargestellt, andernfalls mit einem roten Kreuz.

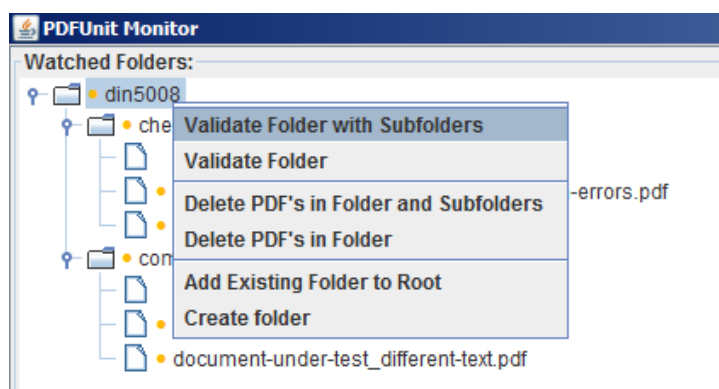
Das folgende Bild zeigt den PDFUnit-Monitor. Die linke Seite ist die Verzeichnisstruktur mit ihren PDF- und Excel-Dokumenten. Die rechte Seite zeigt oben die Fehlerliste mit Filtermöglichkeiten und unten die Details zu einem einzelnen Fehler.



Nachfolgende Bilder zeigen Teile dieser Oberfläche in vergrößerten Ausschnitten.

Ein Doppelklick auf ein PDF-Dokument in der Baumstruktur öffnet das Dokument mit der Standardanwendung des Betriebssystems. Gleiches gilt für einen Doppelklick auf eine Excel-Datei.

Die Verzeichnisstruktur ist nicht nur eine reine Darstellung. Wie unter Windows, Linux und MacOS üblich können verschiedene Funktionen über die rechte Maustaste ausgelöst werden. Das folgende Bild gibt einen kleinen Einblick in das Kontextmenü:



Fehlerübersicht mit Filtermöglichkeiten

Der Monitor zeigt in der oberen Hälfte der rechten Seite die Ergebnisse der Prüfungen aller PDF-Dokumente als Liste an. Für jede Regelverletzung existiert ein eigener Eintrag. Die Details einer Regelverletzung erscheinen in der unteren Hälfte sobald eine Zeile in der Liste selektiert wird.

Filter:

Filter Error Level: ☐ ALL

Filter PDF: ☐ clear

Filter Folder: ☒ clear din5008

Filter Constraints: ☒ clear FormA

Filter Message: ☐ clear

Validation Results:

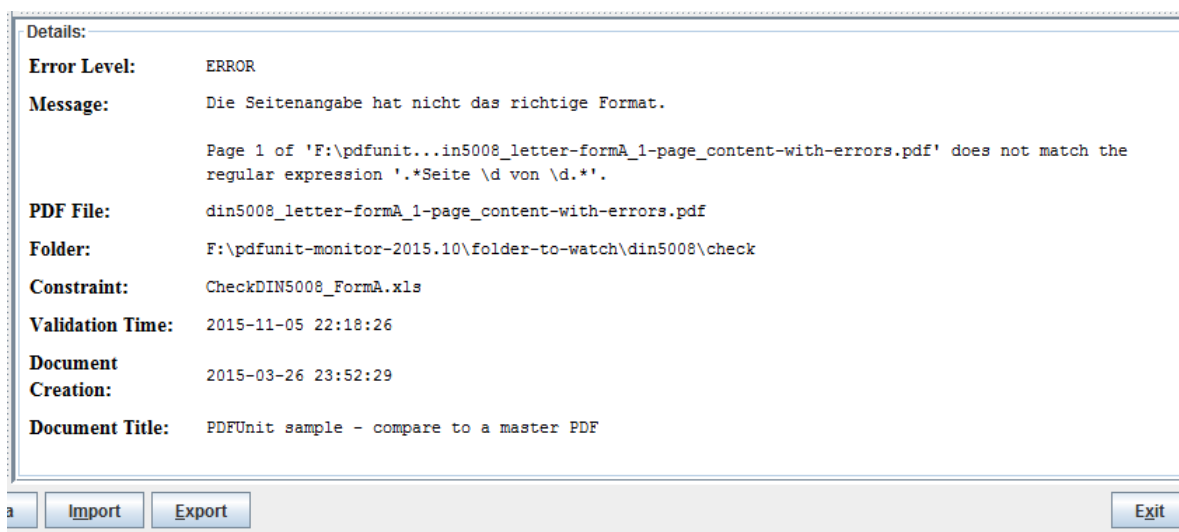
Time	Error Level	PDF Document	Constraint
2015-10-26 01:22:12	OK	din5008_letter-formA_5-pages.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	OK	din5008_letter-formA_5-pages.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:12	WARNING	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	WARNING	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	WARNING	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:12	WARNING	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_text-in-left-margin.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:12	WARNING	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	WARNING	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:12	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:11	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:11	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_individual.xls
2015-10-26 01:22:11	WARNING	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_general.xls
2015-10-26 01:22:11	WARNING	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA_general.xls

Die Liste der Fehler kann über Filter eingeschränkt werden. Die Filtermöglichkeiten stehen mit der Verzeichnisstruktur in Verbindungen. Wird eine Excel-Datei in der Struktur angeklickt, wird ein Filter mit diesem Namen aktiviert und es werden alle PDF-Dokumente, die mit dieser Excel-Datei geprüft wurden, angezeigt. Filter gibt es für PDF-Dokumente, Verzeichnisse, Excel-Dateien, und Fehlermeldungen.

Wird in der Liste ein PDF-Dokument oder eine Excel-Datei angeklickt, öffnet sich die Standardanwendung für diesen Dateityp.

Details zum Fehler

Wird ein Eintrag in der Fehlerliste selektiert, werden Details über den Fehler und über das fehlerhafte PDF in der unteren Hälfte der rechten Seite des Monitoroberfläche angezeigt.



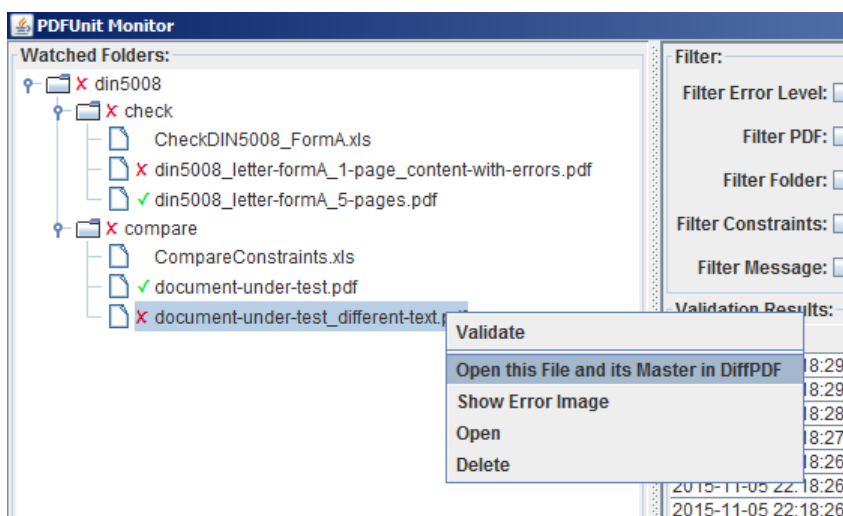
Der erste Teil der Fehlermeldung stammt aus der Excel-Datei und wird von der Person, die die Tests erstellt, geschrieben. Weitere Teile der Meldung stammen vom Testwerkzeug PDFUnit. Neben der eigentlichen Fehlermeldung werden weitere nützliche Informationen über das PDF-Dokument, die Regeldatei und den Testzeitraum angezeigt.

Die Fehlermeldungen von PDFUnit sind momentan auf Englisch verfasst, können aber mit wenig Aufwand auch auf Deutsch zur Verfügung gestellt werden.

Vergleich eines PDF-Dokumentes gegen eine Vorlage

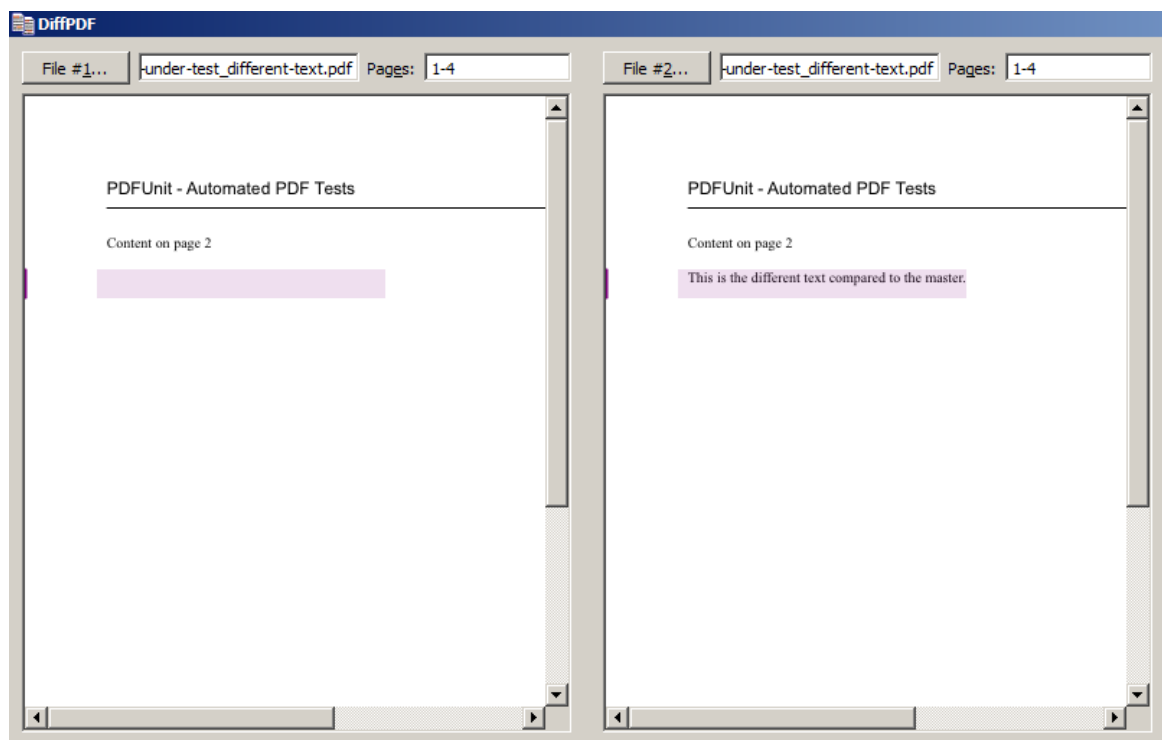
Ein PDF-Dokument kann auch gegen eine Vorlage verglichen werden. Die Regeln für einen solchen Vergleich werden ebenfalls in einer Excel-Datei abgelegt. Erkennt der PDFUnit-Monitor einen Unterschied zwischen dem Test-Dokument und dem Master-Dokument, wird der Name des Test-Dokuments in der Verzeichnisstruktur mit einem roten Kreuz markiert.

Über die rechte Maustaste kann anschließend das Programm 'DiffPDF 1.5.1, portable' gestartet werden, das den Unterschied gut darstellt.



Das Programm stammt von Mark Summerfield und steht als 'Portable App' über diesen Link (Download) zum Download zur Verfügung. DiffPDF kann auf Englisch, Deutsch, Französisch und Tschechisch benutzt werden. Herzlichen Dank an alle Beteiligten für Ihre Arbeit und das großartige Ergebnis.

Das nächste Bild zeigt die Anwendung DiffPDF unmittelbar, nachdem sie aus dem PDFUnit-Monitor heraus gestartet wurde. Auf der linken Seite wird die Vorlage dargestellt, auf der rechten das aktuelle Testdokument. Die Anwendung positioniert sich direkt auf dem ersten Fehler, hier auf Seite 2. Die Abweichungen werden farblich hinterlegt. Das Bild zeigt nicht die Buttons, mit denen von Fehler zu Fehler gesprungen werden kann.



Das Kapitel 3.5: „DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente“ (S. 20). gibt weitere Informationen über das Programm 'DiffPDFPortable'.

Export und Import der Ergebnisse

Die im PDFUnit-Monitor angezeigten Testergebnisse (ohne Filter) können über den Button 'Export' als XML-Datei exportiert werden. Sie stehen damit für einen dauerhaften Nachweis zur Verfügung. Mit XSLT-Stylesheets können die exportierten Dateien in HTML-Reports umgewandelt werden.

Über den Button 'Import' werden sie wieder importiert.

Mehrsprachigkeit

Der PDFUnit-Monitor steht momentan für die Sprache Englisch zur Verfügung. Eine Erweiterung auf andere Sprachen ist strukturell vorgesehen und kann auf Wunsch mit wenig Aufwand realisiert werden.

Kapitel 3. Funktionalität

3.1. Strukturen hinter der Funktionalität

Aufbau einer Excel-Datei

Die Regeln für eine PDF-Validierung werden in Excel-Dateien abgelegt. Eine solche Datei wird nach Tabellenblättern (Sheets) mit festgelegten Namen durchsucht:

Excel-Sheet Name	Bedeutung
regions	Definition von Seitenausschnitten.
check	Definition von Testfällen auf eines einzelnen PDF-Dokumentes.
compare	Definition von Testfällen für einen Vergleich eines PDF-Dokumentes mit einer geprüften Vorlage.

Alle drei Tabellenblätter haben einen festgelegten Aufbau, der in den nachfolgenden Abschnitten beschrieben wird. In allen drei Tabellenblättern bewirkt ein Stern '*' in der ersten Spalte, dass die Zeile als Kommentar gilt.

Die Reihenfolge der Spalten darf nicht geändert werden. Zusätzliche, individuelle Spalten führen dazu, dass Tests nicht funktionieren.

Struktur des Excel-Sheets 'region', Definition von Seitenausschnitten

Zusätzlich zu der Angabe von Seiten, ist es oftmals notwendig, Tests lediglich auf Ausschnitte einzelner Seiten zu beschränken. So macht ein Vergleich zweier Dokumente mit dem Text der vollen Seiten häufig keinen Sinn, weil sich die Dokumente in ihrem Datum unterscheiden. In den Excel-Dateien können deshalb Seitenbereiche ('regions') definiert werden. Um es genauer zu sagen: Solche Seitenbereich müssen definiert werden. Jeder Test benötigt immer einen Bereich.

Ein Seitenausschnitt wird durch 4 Werte beschrieben: die x/y-Koordinaten der linken oberen Ecke und durch Breite und Höhe des Ausschnitts. Alle Werte sind Millimeter-Angaben. Die Werte dürfen Nachkommastellen haben. Das nächste Bild zeigt ein paar Beispiele.

	A	B	C	D	E	F
1	*	Bereichsdefinition für DIN A4, DIN5008, Form-A				
2	*	Region definitions for letters according to DIN 5008 from 04/2011. Test cases are using the ID to reference the regions.				
3	*					
4	*	ID	x	y	width	height
5		Anschriftsbereich	25	32	80	40
6		Infoblock	125	32	75,0	40
7		Briefkopfinhalte	25	0	185	27
8		Textbereich Seite 1	25	80	165	165
9		Textbereich Seite 2ff	25	32	165	212
10		Seitenangabe jede Seite	25	245	165	11,5
11		Fußzeile	25	260	165	30
12		Linker Rand	0	0	25	297
13		Rechter Rand	190	70	20	227
14		Abstand hinter Kopfzeile	0	72	210	8,5

Wie zu erkennen ist, enthält das Tabellenblatt neben den 4 Werten für die Ausschnitte nur noch die Spalte mit der Überschrift ID. Jeder Ausschnitt muss eine eindeutige ID besitzen. Diese ID wird von den Testfalldeklarationen in den Tabellenblättern 'check' und 'compare' referenziert.

Struktur des Excel-Sheets 'check', Definition von Testfällen

Das Tabellenblatt 'check' dient für die Erfassung aller Informationen, die für Testfälle benötigt werden. Das sind:

Name der Spalte	Bedeutung
ID	Name (ID) des Testfalls.
pages	Definition der Seiten, auf die sich ein Testfall bezieht.
region	Referenz auf einen Seitenausschnitt, der im Tabellenblatt 'regions' definiert ist.
constraint type	Art der Überprüfung. Die erlaubten Werte sind weiter unten beschrieben.
constraint value	Der erwartete Wert, der in die Validierung einfließt.
whitespace	Angabe, wie mit Whitespaces (Leerzeichen, Zeilenumbrüchen, etc.) umgegangen werden soll. Die erlaubten Werte für diese Spalte werden weiter unten beschrieben.
message	Diese Spalte nimmt eine Fehlermeldung auf. Eine Fehlermeldung kann Platzhalter enthalten. Die erlaubten Platzhalter werden in einem der folgenden Kapitel beschrieben.

	A	B	C	D	E	F	G	H
1	*	Definition von Testfällen für die Verarbeitung durch den PDFUnit-Monitor						
2	*	Definition of test cases for validation of test PDF documents						
3	*							
4	*	ID	pages	region	constraint type	constraint value	whitespace	message
5		Anschriftsbereich_ID01	1	Anschriftsbereich	must contain	Anschrift	ignore	Der Text {constraint.value} fehlt im Bereich {region}.
6		Infoblock_ID01	1	Infoblock	must contain	Infoblock	normalize	Der Text {constraint.value} fehlt im Bereich {region}.
7		Briefkopfinhalte_OD01	all	Briefkopfinhalte	must contain	Automated PDF Tests	keep	Der Text {constraint.value} fehlt im Bereich {region}.
8		Textbereich Seite 1_ID01	1	Textbereich Seite 1	must contain	Lorem ipsum		Der Text {constraint.value} fehlt im Bereich {region}.
9	*	Textbereich Seite 2ff_ID	2..	Textbereich Seite 2ff	must contain	Lorem ipsum		Der Text {constraint.value} fehlt im Bereich {region}.
10		Seitenangabe jede Seite_ID01	all	Seitenangabe jede Seite	must match	*Seite 'id von 'id.*		Die Seitenangabe hat nicht das richtige Format.
11		Fußzeile	all	Fußzeile	must contain	Firmenangaben		Der Text {constraint.value} fehlt im Bereich {region}.
12	*							
13		Linker Rand	all	Linker Rand	must be empty			Der linke Rand darf keinen Text enthalten.
14		Rechter Rand	2..	Rechter Rand	must be empty			Der rechte Rand darf keinen Text enthalten.
15		Abstand hinter Kopfzeile	1	Abstand hinter Kopfzeile	must be empty			Zwischen Kopfzeile und Adresse muss genug Abstand sein.
16	*							
17		Keine leeren Seiten	2...3	Textbereich Seite 2ff	must not be empty			
18		Leerer Textbereich 4, 5	4, 5	Textbereich Seite 2ff	must be empty			
19								

Die Bedeutung der Spalten und die erlaubten Werte werden in den folgenden Abschnitten erklärt.

Benennung der Seiten, auf die sich ein Test bezieht

Ein Testfall bezieht sich nicht nur auf ein bestimmtes Dokument, sondern auch auf bestimmte Seiten eines Dokumentes. Die Excel-Dateien ermöglichen es, die gewünschten Seiten zu spezifizieren. Die folgende Liste zeigt alle verfügbaren Beispiele:

Seiten	Syntax in Excel
eine einzelne Seite	1
mehrere einzelne Seiten	1, 3, 5
alle Seiten	all
alle Seiten ab der angegebenen (einschließlich)	2...
alle Seiten vor der angegebenen (einschließlich)	...5
alle Seiten zwischen den angegebenen (einschließlich)	2...5

Zwei Seitenzahlen müssen durch ein Leerzeichen getrennt werden. Das Komma ist optional, es dient lediglich der besseren Lesbarkeit.

Spalte 'constraint type', verschiedene Arten von Textvergleichen

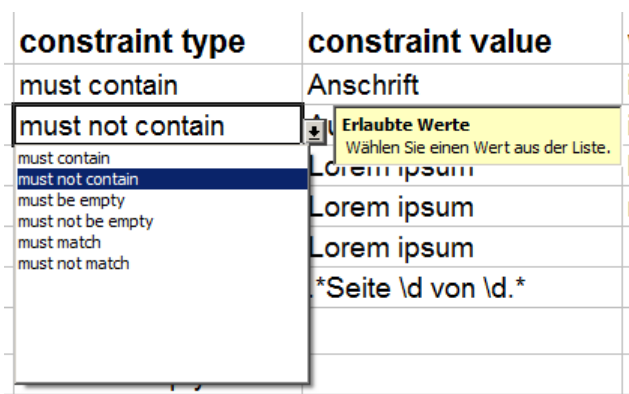
Die Spalte 'constraint type' dient dazu, festzulegen, wie Texte in einer Prüfung verglichen werden sollen, beispielsweise ob ein bestimmter Text in einem Bereich enthalten sein soll ('must contain') oder ob dieser Text gerade nicht in dem Bereich enthalten sein darf ('must not contain'). Die folgende Liste zeigt die erlaubten Werte für Prüfungen auf ein **einzelnes** Dokument:

Schlüsselwort	Verhalten
'must contain'	Der Text, der in der Spalte 'constraint value' steht, muss innerhalb des angegebenen Bereichs im PDF-Dokument vorkommen. Zusätzlich muss die interne Behandlung von Leerzeichen vorgegeben werden.
'must not contain'	Der Text, der in der Spalte 'constraint value' steht, darf innerhalb des angegebenen Bereichs im PDF-Dokument nicht vorkommen. Zusätzlich muss die interne Behandlung von Leerzeichen vorgegeben werden.
'must be empty'	Der angegebene Bereich darf keinen Text enthalten.
'must not be empty'	Der angegebene Bereich muss Text enthalten
'must match'	Der Text, der in der Spalte 'constraint value' angegeben ist, wird als regulärer Ausdruck auf den Text im angegebenen Bereich angewendet. Es muss mindestens einen Treffer geben.
'must not match'	Der reguläre Ausdruck in der Spalte 'constraint-value' darf für den Text innerhalb des angegebenen Bereichs keinen Treffer ergeben.

Für einen Vergleich **zweier** PDF-Dokumente sind folgende Werte in der Spalte 'constraint type' erlaubt:

Schlüsselwort	Verhalten
'visually identical'	Zwei PDF-Dokumente müssen im angegebenen Bereich als gerendertes Bild identisch sein.
'same text'	Zwei PDF-Dokumente müssen im angegebenen Bereich den gleichen Text enthalten. Zusätzlich muss noch die interne Behandlung von Leerzeichen vorgegeben werden.

Das Release des PDFUnit-Monitors wird mit einem Excel-Beispiel ausgeliefert, das die erlaubten Werte für die Spalte 'constraint type' als Auswahlliste enthält. Gleiches gilt für die Spalte 'whitespace' und für die Referenz-Spalte auf definierte Seitenausschnitte. Das Excel-Beispiel heißt 'PDFUnitConstraining_Vorlage.xls'. Hier eine Darstellung der Auswahlliste:



Wichtig: die Spalte 'constraint type' darf nicht leer sein. In einem solchen Fall bringt das System eine Fehlermeldung.

Spalte 'whitespace', Umgang mit Leerzeichen

Textvergleiche können an Leerzeichen und Zeilenumbrüchen scheitern. Schon bei einem Wechsel der Schriftart können andere Zeilenumbrüche entstehen. Um Tests davon unabhängig zu machen, kann PDFUnit auf dreierlei Arten mit Whitespaces umgehen:

Schlüsselwort	Verhalten
'ignore'	Text wird so komprimiert, dass er keine Whitespaces mehr enthält.
'keep'	Alle Whitespaces bleiben erhalten.
'normalize'	Whitespaces am Anfang und am Ende eines Textes werden gelöscht. Whitespaces innerhalb eines Textes werden auf ein Leerzeichen reduziert.

Wird die Spalte 'whitespace' leer gelassen, gilt 'normalize' als Voreinstellung.

Beim Vergleich zweier PDF-Dokumente spielen Whitespaces nicht immer eine Rolle. Deshalb wird die Angabe der Whitespace-Behandlung für den Constraint-Type 'visually identical' ignoriert:

Spalte 'constraint value', erwarteter Text

Wenn ein Testfall prüft, ob ein Bereich des Test-Dokumentes einen ganz bestimmten Text enthält, muss die Excel-Datei eine Spalte anbieten, die den erwarteten Text aufnimmt. Diese Spalte heißt 'constraint value'.

Der Text in dieser Spalte wird gegen den Text im Testdokument verglichen, wobei die Art des Vergleiches von der Angabe in der Spalte 'constraint type' abhängt.

Steht der Wert in der Spalte 'constraint type' auf 'must match' oder 'must not match', wird der Inhalt der Spalte 'constraint value' als Regulärer Ausdruck interpretiert. Informationen über reguläre Ausdrücke gibt es im Internet unter anderem hier.

Spalte 'message', Fehlermeldungen mit Platzhaltern

In den Excel-Dateien können auch Fehlermeldung hinterlegt werden. Eine solche Fehlermeldung kann Platzhalter für Laufzeitinformationen enthalten. Das folgende Bild zeigt mehrere Beispiele:

message
Der Text {constraint.value} fehlt im Bereich {region}.
Der Text {constraint.value} fehlt im Bereich {region}.
Der Text {constraint.value} fehlt im Bereich {region}.
Der Text {constraint.value} fehlt im Bereich {region}.
Der Text {constraint.value} fehlt im Bereich {region}.
Die Seitenangabe hat nicht das richtige Format. Testfall '{id}'.
Der Text {constraint.value} fehlt im Bereich {region}.

Wie aus dem Bild ersichtlich ist, werden Platzhalter im Text in geschweifte Klammern eingefasst. Folgende Platzhalter können benutzt werden:

Platzhalter, Schlüsselwort	Bedeutung
{constraint.value}	Der Wert der Spalte 'constraint value'
{constraint.type}	Der Wert der Spalte 'constraint type'
{id}	Die ID des aktuellen Testfalls

Platzhalter, Schlüsselwort Bedeutung

{pdf.page} Seitenzahl der Seite, auf der ein Fehler erkannt wurde

Die Platzhalter können an beliebiger Stelle im Text eingebaut werden.

Die Anwendung PDFUnit-Monitor ist darauf vorbereitet, interne Fehlermeldung in jeder beliebigen Sprache auszugeben. Zum Zeitpunkt der Auslieferung werden englische Meldungen ausgegeben. Um eine andere Sprache zu erhalten, schicken Sie eine kurze Anfrage an [license\[at\]pdfunit.com](mailto:license[at]pdfunit.com).

Struktur des Excel-Sheets 'compare', Definition von Testfällen für Vergleiche

Tests können so konzipiert sein, dass ein Test-Dokument gegen eine PDF-Vorlage verglichen wird. Ein solcher Test vergleicht entweder reinen Text oder gerenderte Seiten miteinander.

Dieses leicht abweichende Verhalten, verglichen mit Tests auf Einzeldokumente, hat zur Folge, dass in der Excel-Datei keine Angabe mehr über erwartete Texte gemacht werden muss. Somit verbleiben folgende Spalten:

	A	B	C	D	E	F	G
1	*	Definition von Testfällen für den Vergleich von Test-PDF-Dokumenten gegen eine geprüfte Vorlage					
2	*	Definition of test cases comparing a test PDF with a master PDF					
3	*						
4	*	ID	pages	region	constraint type	whitespace	message
5		Anschriftsbereich_ID01	1	Anschriftsbereich	same text	ignore	Textunterschiede im Anschriftsbereich.
6		Infoblock_ID01	1	Infoblock	same text	normalize	Textunterschiede im Infoblock.
7		Header_AlsText	all	Header	same text	keep	Unterschiede im Header beim Vergleich der Dokumente als Text.
8		Header_AlsBild	all	Header	visually identical		Unterschiede im Header beim Vergleich der Dokumente als Bild.
9		TexteSeite2ff_AlsText	2...	Textbereich Seite 2ff	same text	normalize	Unterschiede beim Vergleich der Dokumente als Text. ID: {id}.
10		TexteSeite2ff_AlsBild	2...	Textbereich Seite 2ff	visually identical		Unterschiede beim Vergleich der Dokumente als Bild. ID: {id}.
11		Fußzeile	2,3	Footer	same text	normalize	Unterschiede im Footer.
12							

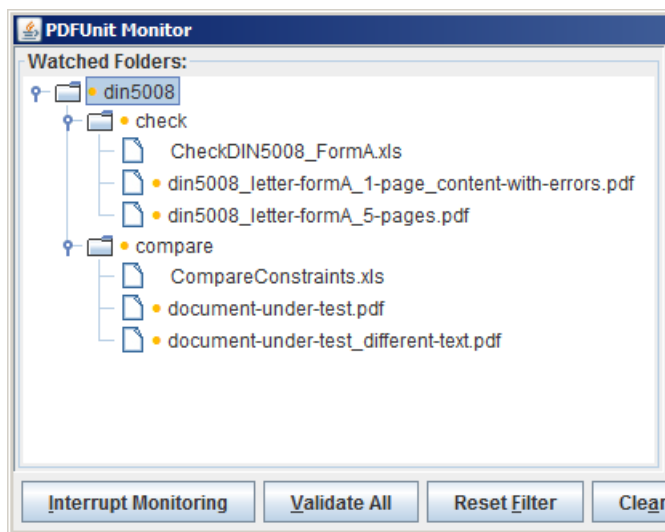
Die Bedeutungen der Spalten ändert sich nicht, sie sind in den vorhergehenden Abschnitten beschrieben. Lediglich die erlaubten Werte für die 'constraint types', also die Angabe, wie ein Test ausgeführt werden soll, hat sich geändert. Für Vergleiche sind folgende Constraint-Typen erlaubt:

- 'same text'
- 'visually identical'

Die Vergleichsvorlagen (PDF-Dokumente) werden unterhalb des in der Konfigurationsdatei 'pdfunit-monitor.config' für die Property 'FOLDER_WITH_MASTERDOCUMENTS' angegebenen Verzeichnisses gesucht. Dieses Verzeichnis muss mit dem überwachten Verzeichnis, das mit der Property 'WATCHED_FOLDER_FOR_COMPARISON' vorgegeben wird, in einer Beziehung stehen: Beide Verzeichnisse müssen untereinander durch einen relativen Pfad erreichbar sein.

3.2. Darstellung - Überwachte Verzeichnisse

Die Verzeichnisstruktur im linken Bereich des Monitors zeigt alle Dateien mit den Endungen PDF, XLS und XSLX in allen Verzeichnissen unterhalb des Verzeichnisses an, das in der Konfigurationsdatei 'pdfunit-monitor.config' mit der Property 'WATCHED_FOLDER' angegeben ist. Die Konfigurationsdatei ist in Kapitel 6: „Konfiguration (pdfunit-monitor.config)“ (S. 30). ausführlich beschrieben.

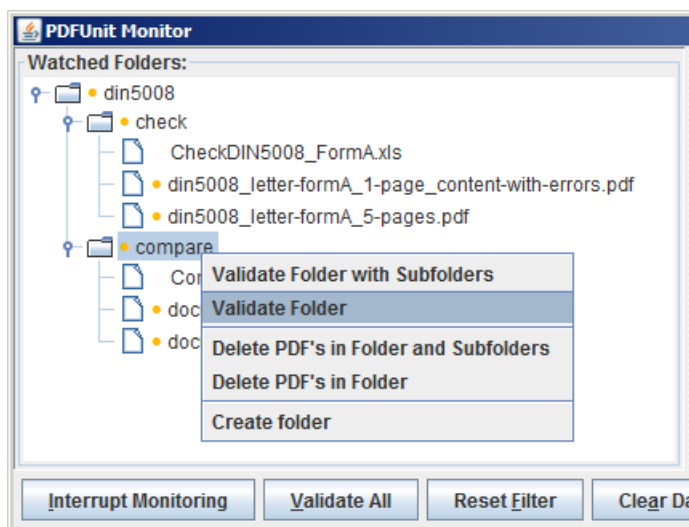


Der Monitor reagiert auf Veränderungen an den überwachten Dateien und Verzeichnissen. Das heißt, die Anzeige wird aktualisiert, wenn Dateien und Verzeichnisse neu hinzukommen, sich ändern oder gelöscht werden.

Jedes PDF-Dokument wird gegen alle Excel-Dateien geprüft, die sich im selben Verzeichnis befinden. Neue PDF-Dokumente werden automatisch gegen die vorhandenen Excel-Dateien überprüft. Das Ergebnis der Prüfungen landet in der Ergebnisliste.

Kontextmenü

Wenn der Monitor neu gestartet wird, kann die Validierung der vorhandenen PDF-Dokumente über das Kontextmenü oder über Schaltflächen (Buttons) im unteren Bereich des Monitors angestoßen werden.

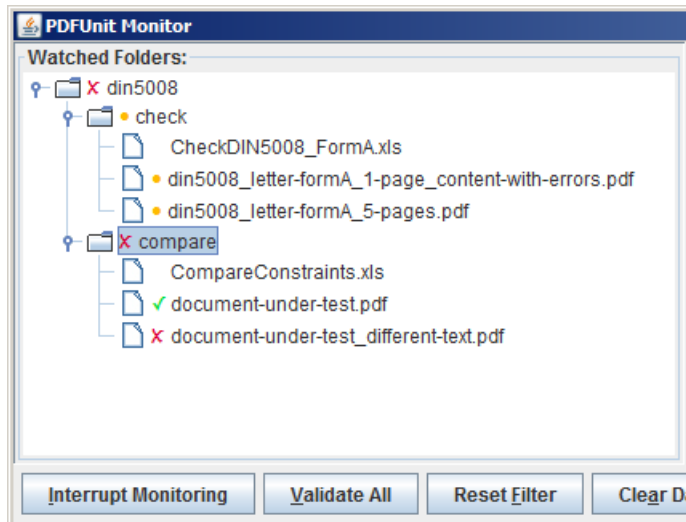


Selbstverständlich steht das Kontextmenü jederzeit zur Verfügung. Die Inhalte eines Kontextmenüs wechseln, abhängig davon, ob sich das Menü auf ein Verzeichnis, eine PDF- oder eine Excel-Datei bezieht.

Farbliche Darstellung des Validierungsergebnisses

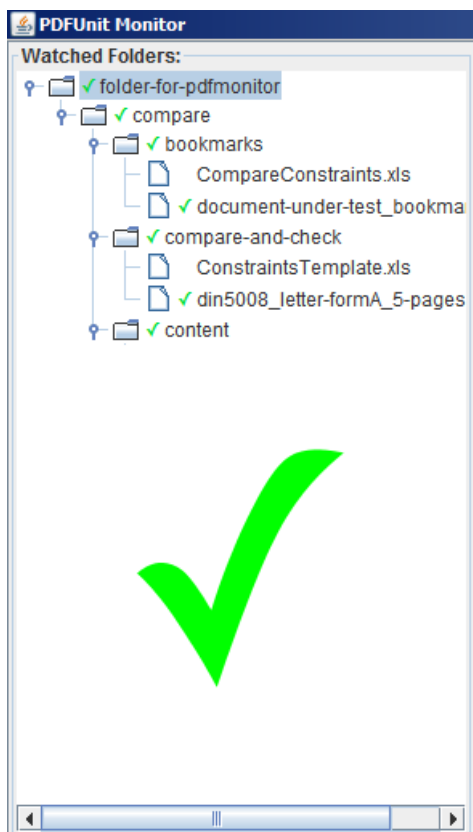
Das Validierungsergebnis eines PDF-Dokumentes wird durch ein farbiges Symbol vor seinem Namen dargestellt. Nur, wenn alle Regeln fehlerfrei durchlaufen wurden, wird der Name mit einem grünen Haken dargestellt. Sobald eine Regel verletzt wurde, wird der Name mit einem roten Kreuz versehen.

Die Validierungsergebnisse der PDF-Dokumente werden auf ihre Verzeichnisse übertragen. Ein Verzeichnis erhält einen grünen Haken, wenn sämtliche PDF-Dokumente in ihm selber und in seinen Unterverzeichnissen einen grünen Haken haben. Sobald nur ein PDF-Dokument fehlerhaft ist, erhalten alle Verzeichnisse oberhalb des Dokumentes ein rotes Kreuz. Das nächste Bild zeigt dieses Verhalten.



Ein gelber Punkt zeigt an, dass dieses Dokument noch nicht validiert wurde, bzw. dass ein Verzeichnis noch ungeprüfte Pdf-Dokumente enthält.

Wenn alle PDF-Dokumente fehlerfrei sind, zeigt der Monitor einen großen grünen Haken:



Drag and Drop

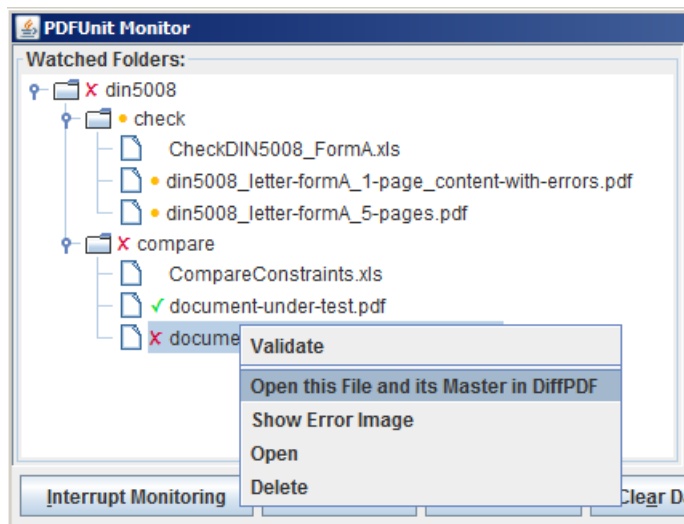
Der PDFUnit-Monitor unterstützt die 'Drop'-Funktion von 'Drag-and-Drop'. Das bedeutet, es können PDF-Dokumente per Maus an die angezeigten Verzeichnisse 'übergeben' werden. Die Dateien werden kopiert, nicht verschoben! Nach dem Kopieren startet automatisch die Validierung gegen die in dem Verzeichnis existierenden Excel-Dateien.

Anzeigen und Öffnen von Dateien

Die Dokumente in der Verzeichnisstruktur können geöffnet werden. Ein Doppelklick auf den Namen eines PDF- bzw. Excel-Dokumentes öffnet die Standardanwendung des Betriebssystems für den jeweiligen Dateityp.

Anzeige von Differenzen zwischen zwei PDF-Dateien

Ein Test-PDF-Dokument und die zu ihm gehörende Vergleichsvorlage können über das Kontextmenü vom Test-PDF mit einem speziellen Programm geöffnet werden, um die Differenzen anzuzeigen. Die nächste Abbildung zeigt das Kontextmenü.



Das spezielle Programm 'DiffPDFPortable' wird in Kapitel 3.5: „DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente“ (S. 20) genauer beschrieben.

3.3. Darstellung - Liste der Prüfergebnisse, Filter

Hat eine Validierung der PDF-Dokumente stattgefunden, werden alle Ergebnisse in einer Übersichtsliste dargestellt.

Filter:

Filter Error Level: ☐ ALL

Filter PDF: ☐ Clear

Filter Folder: ☒ Clear

Filter Constraints: ☐ Clear

Filter Message: ☐ Clear

PDF Un:t
[Web Documentation](#)
[Local Documentation \(PDF\)](#)

Validation Results:

Time	Error Level	PDF Document	Constraint
2015-11-05 22:48:47	ERROR	document-under-test_different-text.pdf	CompareConstraints.xls
2015-11-05 22:48:46	ERROR	document-under-test_different-text.pdf	CompareConstraints.xls
2015-11-05 22:48:46	OK	document-under-test.pdf	CompareConstraints.xls
2015-11-05 22:48:44	OK	din5008_letter-formA_5-pages.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	WARNING	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	WARNING	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:43	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls

Diese Liste kann über verschiedene Filter verkleinert werden. Die Filter können durch Eingaben in den Textfeldern neben den kleinen 'clear'-Checkboxes gesetzt werden, oder sie werden automatisch gesetzt, wenn in der Verzeichnisstruktur auf eine Datei oder ein Verzeichnis geklickt wird.

Die Filter wirken folgendermaßen:

Filter	Wirkung
Filter Error Level	Ein Validierungsergebnis landet in der Ergebnisliste, wenn es genau denselben Error-Level besitzt. Steht der Filter auf ALL, werden alle Ergebnisse angezeigt.
Filter PDF	Ein Validierungsergebnis landet in der Ergebnisliste, wenn der eingegebene Wert Teil des Dateinamens einer PDF-Datei ist. Die Klein-Großschreibung spielt für den Vergleich keine Rolle.
Filter Folder	Ein Validierungsergebnis landet in der Ergebnisliste, wenn sich ein PDF-Dokument unterhalb des Verzeichnisses befindet, dessen Name im Filter gesetzt wurde. Für diesen Filter ist die Klein-/Großschreibung relevant.
Filter Constraints	Ein Validierungsergebnis landet in der Ergebnisliste, wenn der eingegebene Wert Teil des Dateinamens einer Excel-Datei ist. Für den Vergleich ist die Klein-/Großschreibung irrelevant.
Filter Message	Ein Validierungsergebnis landet in der Ergebnisliste, wenn der eingegebene Wert Teil der Fehlermeldung ist. Klein-/Großschreibung ist irrelevant.

Das folgende Bild zeigt ein Beispiel mit mehreren gesetzten Filtern:

Filter:

Filter Error Level: ☒ ERROR

Filter PDF: ☒ Clear

Filter Folder: ☐ Clear

Filter Constraints: ☐ Clear

Filter Message: ☐ Clear

PDF Un:t
[Web Documentation](#)
[Local Documentation \(PDF\)](#)

Validation Results:

Time	Error Level	PDF Document	Constraint
2015-11-05 22:48:44	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:44	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2015-11-05 22:48:43	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls

Alle Filter können über den Button 'Reset Filter' und die Checkboxes 'Clear' zurückgesetzt werden.

Die Liste dient aber nicht nur zur Darstellung. Aus der Liste heraus können auch Dokumente geöffnet werden. Ein Klick auf einen PDF- oder Excel-Dateinamen bewirkt, dass die Standardanwendung des Betriebssystems für diesen Dateityp gestartet wird.

Hinweis: Wenn ein Element der Verzeichnisstruktur schon selektiert ist und es wird erneut auf dieses Element geklickt, wird der zweite Klick vom Betriebssystem nicht an den PDFUnit-Monitor weitergeleitet. In einer solchen Situation wird der Filter nicht aktualisiert.

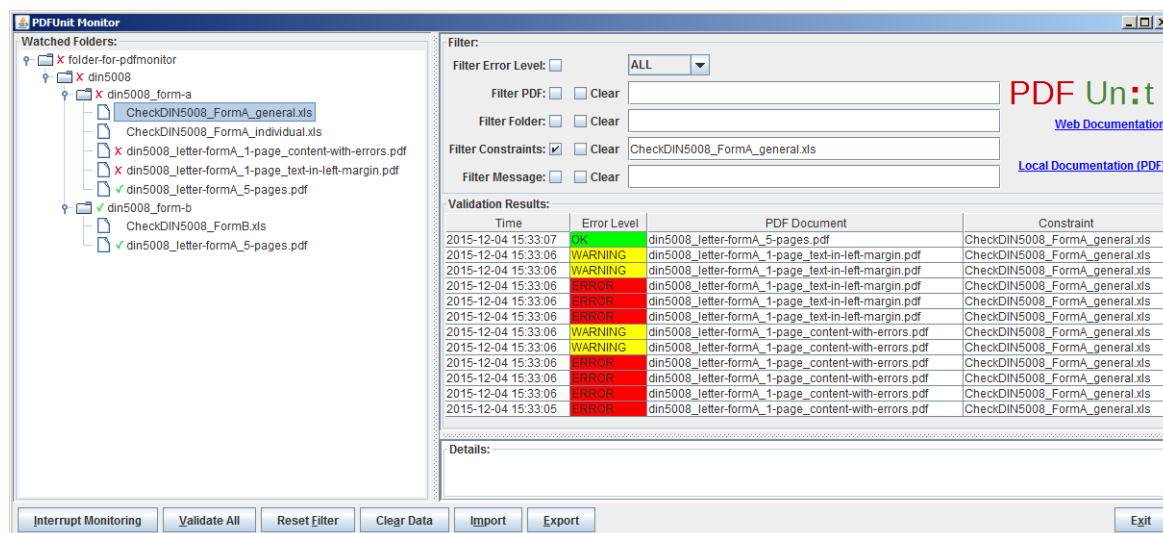
Bedeutung der Error-Level

Die im 'Filter Error Level' angezeigten Werte haben folgende Bedeutung:

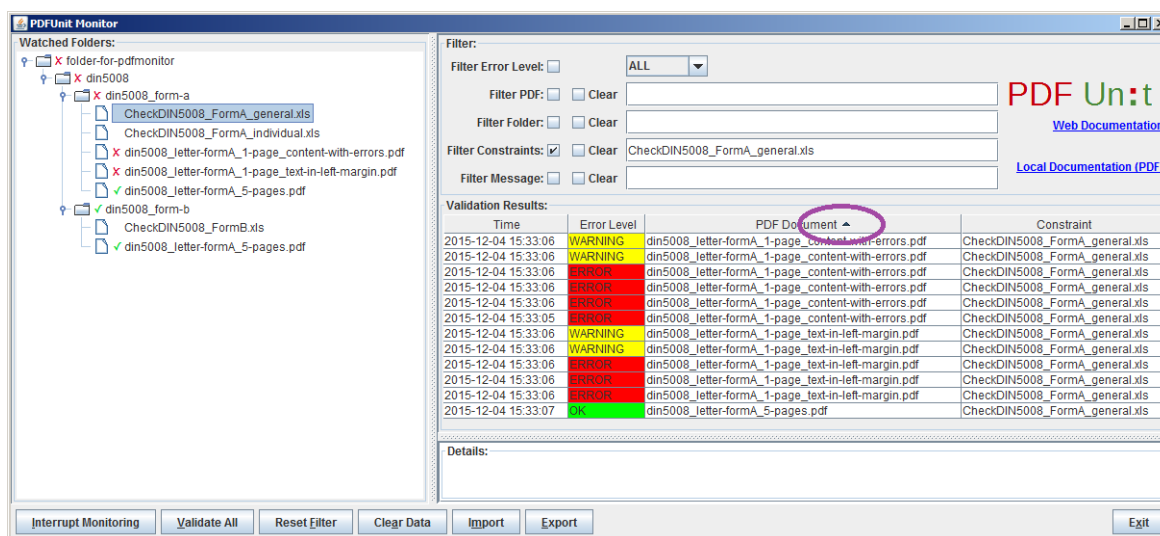
Error-Level	Bedeutung
INFO	Eine Excel-Datei wurde einem Verzeichnis hinzugefügt oder aus ihm entfernt.
WARNING	Eine Validierung konnte nicht durchgeführt werden. Das kann beispielsweise der Fall sein, wenn in dem Verzeichnis keine Excel-Datei existiert.
OK	Ein PDF-Dokument erfüllt alle Regeln aller Excel-Dateien im selben Verzeichnis.
ERROR	Ein PDF-Dokument verletzt eine oder mehrere Regeln in einer der Excel-Dateien desselben Verzeichnisses.
ALL	Alle Validierungsergebnisse werden angezeigt.

Sortieren der Fehlerliste

Die Einträge in der Fehlerliste können auf- oder absteigend sortiert werden, indem man mit der Maus auf den Titel einer Spalte klickt. Die folgenden zwei Bilder zeigen den Effekt.

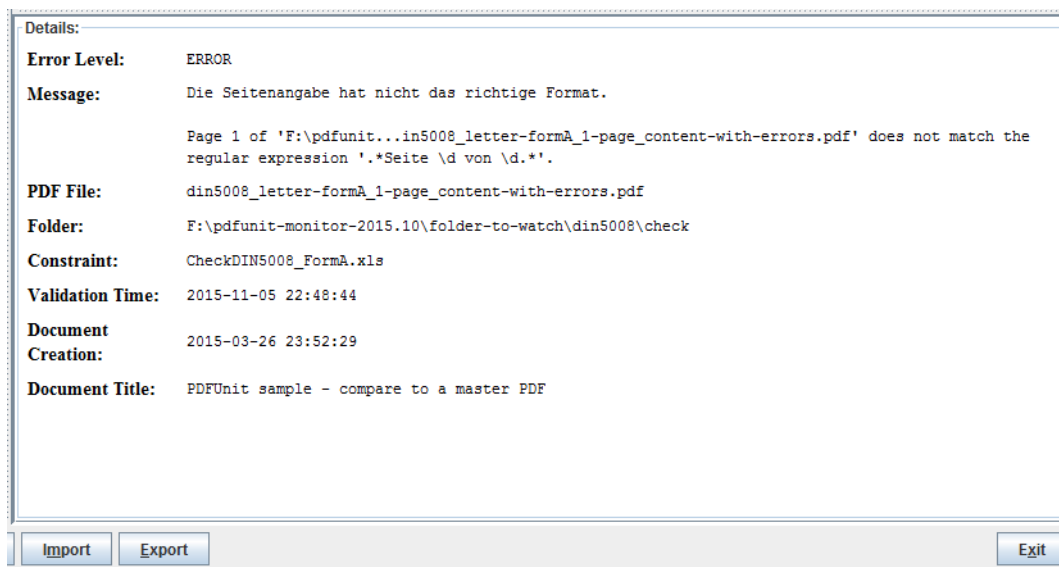


Das vorhergehende Bild zeigt die Standardsortierung, nämlich absteigend über die Spalte 'Time'. Nach einem Mausklick auf den Spaltentitel 'PDF Document' ändert sich die Sortierung, wie das folgende Bild zeigt.



3.4. Darstellung - Details eines Validierungsergebnisses

Wenn eine Zeile in der Liste der Validierungsergebnisse selektiert wird, erscheinen im Bereich darunter ausführliche Informationen zu diesem Ergebnis. Das nächste Bild zeigt ein Beispiel.

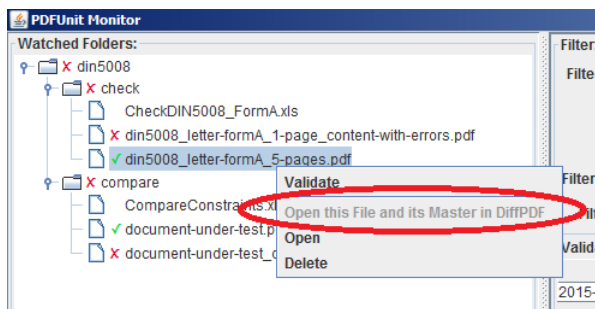


Die Titel vor jeder Zeile und die Fehlertexte werden zur Zeit in englischer Sprache ausgegeben. Die Anwendung ist aber strukturell darauf vorbereitet, diese Inhalte mit wenig Aufwand in anderen Sprachen zur Verfügung zu stellen.

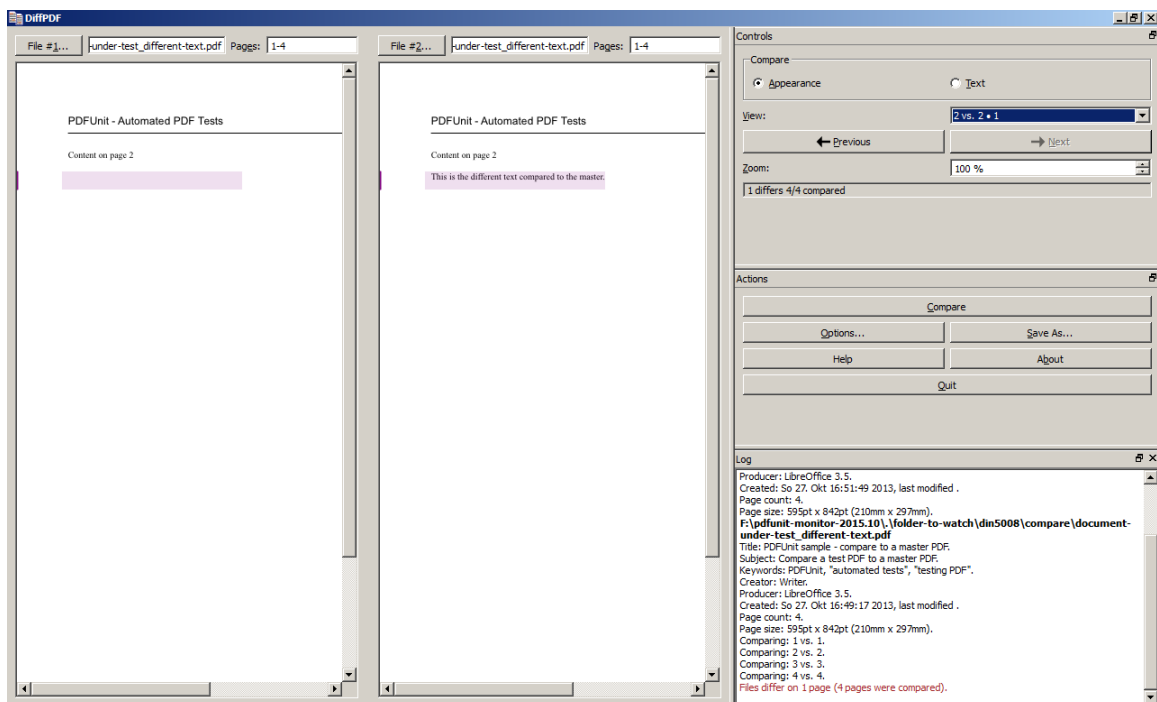
Die erste Zeile in der 'Message' stammt aus der Excel-Datei. Sie kann vom Entwickler des Testfalls beliebig verfasst werden, auch Platzhalter für variable Inhalte sind möglich. Die weiteren Angaben stammen aus dem Testwerkzeug selber.

3.5. DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente

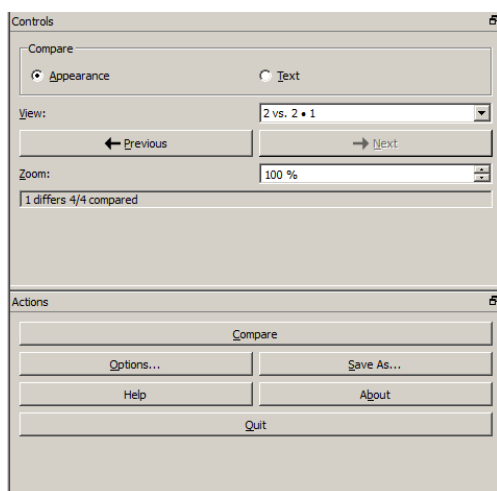
Wie schon zuvor im Kapitel 2: „Quickstart“ (S. 6) beschrieben, können Unterschiede zwischen einem Test-PDF und einer Vergleichsvorlage mit dem Programm DiffPDFPortable dargestellt werden. Das Programm wird über das Kontextmenü einer PDF-Datei gestartet. Ein entsprechender Menüpunkt ist aber nur aktiviert, wenn eine PDF-Datei mit gleichem Namen gefunden wird. Das folgende Bild zeigt ein Kontextmenü für den Fall, dass kein Vergleichs-PDF gefunden wird:



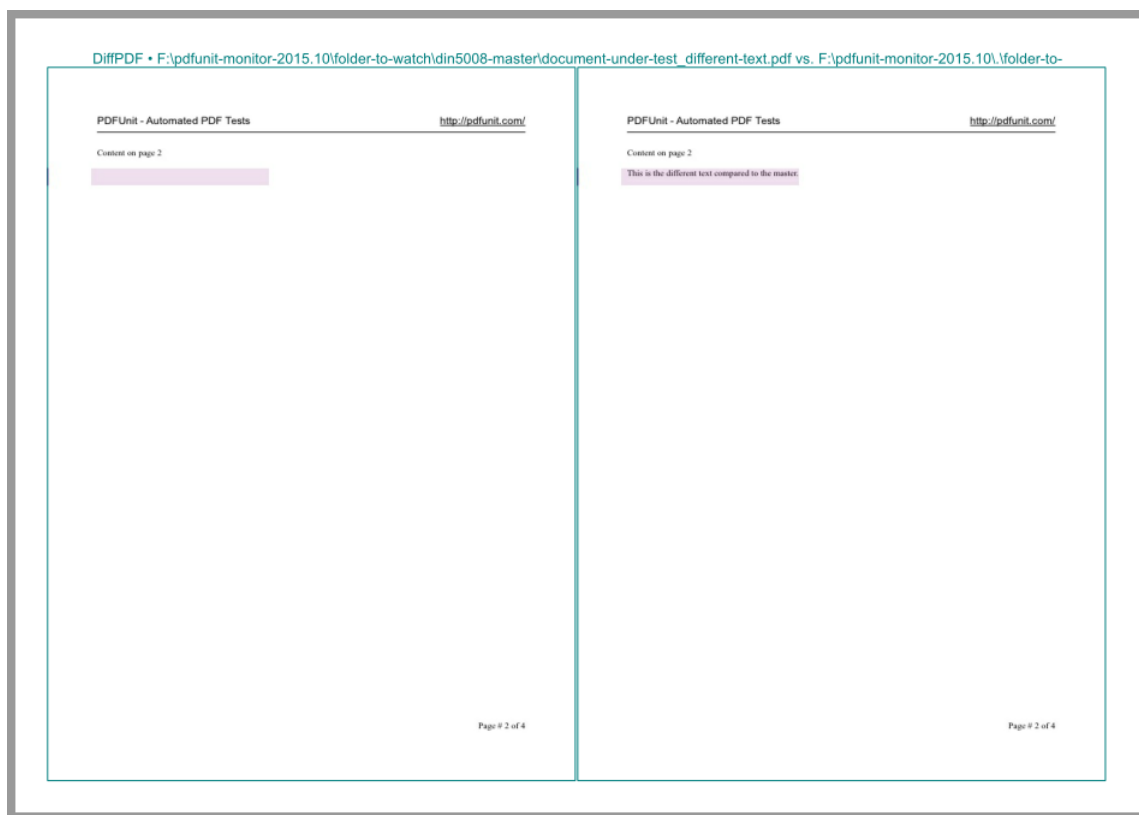
Wenn ein Vergleichs-PDF existiert, zeigt sich nach dem Start folgendes Bild:



Der große Hauptbereich zeigt beide Dokumente seitenweise nebeneinander, links das Master-Dokument (die Vergleichsvorlage) und rechts das aktuell getestete Dokument. Rechts unten gibt es einen kleinen Bereich mit textuellen Informationen über beide PDF-Dateien. Im rechten Bereich oberhalb der Mitte gibt es Buttons, um von Fehler zu Fehler zu navigieren.



Der Button 'Save As...' bietet die Möglichkeit, den angezeigten Vergleich in eine neue Datei zu exportieren. Das Ergebnis sieht dann beispielsweise so aus:



Die Anwendung 'DiffPDFPortable' kann auch unabhängig vom PDFUnit-Monitor genutzt werden.

Der PDFUnit-Monitor erstellt für jeden fehlgeschlagenen, vergleichenden Test ein Vergleichsbild als Report. Auf dessen linker Seite wird das Vergleichs-PDF dargestellt und auf dessen rechter Seite die Abweichungen des Test-PDF-Dokumentes. Diese Bilddateien liegen im Installationsverzeichnis und bekommen den Namen `<dokumenten-name>.<datum-uhrzeit>.out.png`. Das Verzeichnis für diese Dateien kann in der Datei 'config.properties' über die Property 'output.path.diffimages' geändert werden. Diese Datei liegt im Verzeichnis 'PDFUNIT_MONITOR_HOME/lib/PDFUnit-<version>'.

3.6. Export und Import der Ergebnisse

Der PDFUnit-Monitor bietet die Möglichkeit, aktuelle Validierungsergebnisse in eine XML-Datei zu exportieren, damit sie für einen dauerhaften Nachweis zur Verfügung stehen.

Der Export aller Testergebnisse, die der PDFUnit-Monitor aktuell im Hauptspeicher hält, wird über den Button 'Export' in der Button-Leiste am unteren Rand der Anwendung angestoßen. Anschließend erscheint ein übliches Auswahlfenster für das Verzeichnis und den Dateinamen, mit dessen Bestätigung der Export abgeschlossen wird. Ein erfolgreicher Export wird mit einer Erfolgsmeldung bestätigt.

Falls die ausgewählte Datei schon existiert, erscheint eine Sicherheitsabfrage und der Vorgang kann abgebrochen werden.

Der umgekehrte Weg ist auch möglich. Über den Button 'Import' kann eine zuvor exportierte Datei wieder geladen werden.

Wichtiger Hinweis. Es können nur die Validierungsergebnisse exportiert werden, die zum jeweiligen Zeitpunkt im Monitor enthalten sind. Da aber alle Tests auch in einer Logdatei gespeichert werden,

stehen Ergebnisse aus früheren Messungen auch nach dem Beenden der Anwendung zur Verfügung. Die Log-Datei kann aber nicht direkt importiert werden, da sie ein anderes Format hat, als die exportierte Datei.

Kapitel 4. Installation, Update, Deinstallation

4.1. Technische Voraussetzungen

PDFUnit benötigt mindestens Java-7 als Laufzeitumgebung.

Weiterhin werden die Bibliotheken von iText ab Version 5.3.3 benötigt, die aus Lizenzgründen von PDFUnit nicht mitgeliefert werden.

Getestete Umgebungen

Unter den folgenden Umgebungen wurde PDFUnit erfolgreich getestet:

Betriebssystem	Java Version
• Windows-XP, 32 Bit	• Oracle JDK-1.7.0, 32 + 64 Bit
• Windows-7, 64 Bit	• Oracle JDK-1.8.0, 64 Bit
• Kubuntu Linux 12/04, 32 Bit	• Oracle JDK-1.8.0 b100, Windows, 32 + 64 Bit
• Kubuntu Linux 12/04, 64 Bit	• IBM J9, R26_Java726_SR4, Windows 7, 64 Bit
• Mac OS X, 64 Bit	• OpenJDK-1.6.0, 64 Bit
	• Apple Inc. 1.6.0, 64 Bit

Weitere Java/Betriebssystem-Kombinationen werden ständig getestet.

Sollte es Probleme mit der Installation geben, schreiben Sie an [problem\[at\]pdfunit.com](mailto:problem[at]pdfunit.com).

4.2. Installation ohne vorhandenes iText

Download und Entpacken von PDFUnit-Monitor

Laden Sie die Datei `pdfunit-monitor-<VERSION>.zip` aus dem Internet: <http://www.pdfunit.com/de/download/index.html>.

Entpacken Sie die ZIP-Datei, z.B. in ein beliebiges Verzeichnis, nachfolgend `PDFUNIT_MONITOR_HOME` genannt.

Download und Entpacken von iText

Zur Ausführungszeit von PDFUnit-Java wird iText in einer Version 5.3.3 oder höher benötigt. Laden Sie sich iText (<http://sourceforge.net/projects/itext/files/>) herunter.

Entpacken Sie die ZIP-Datei von iText und kopieren Sie den entstandenen Ordner als Verzeichnis unterhalb von '`PDFUNIT_MONITOR_HOME/lib`'.

Beachten Sie, dass iText für den kommerziellen Einsatz lizenzpflichtig ist.

Startskript und Classpath

Im Verzeichnis `PDFUNIT_MONITOR_HOME` existiert ein Skript '`startPDFUnitMonitor.bat`', um den PDFUnit-Monitor zu starten. Tragen Sie in dieses Skript das Installationsverzeichnis von

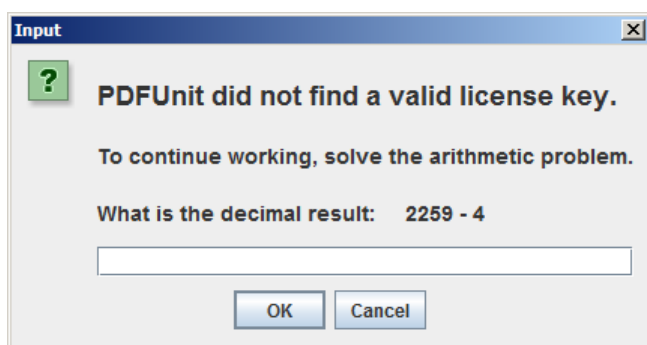
iText ein. Die relevante Skriptvariable heißt `ITEXT_HOME`. Für Unix existiert das Startskript `'startPDFUnitMonitor.sh'`.

Sollten Sie das benötigte PDFUNIT-Java in einem anderen Verzeichnis liegen haben, als in `PDFUNIT_MONITOR_HOME/lib`, tragen Sie den anderen Pfad in das vorhandene Startskript ein. Benutzen Sie dafür die Skriptvariable `PDFUNIT-JAVA_HOME`.

Sofern Sie ein lizenziertes PDFUnit verwenden, muss die Lizenzschlüsseldatei `license-key_pdfunit-java.lic` auch im Classpath liegen. Am einfachsten ist es, sie in das Home-Verzeichnis von PDFUnit-Java zu kopieren.

PDFUnit-Monitor ohne Lizenzschlüssel nutzen

Es ist erlaubt, den PDFUnit-Monitor zu Evaluationszwecken ohne Lizenz zu verwenden. Wenn Sie dann einen Test starten, erscheint ein kleines Fenster mit einer leichten Rechenaufgabe, die Sie lösen müssen. Mit der richtigen Lösung laufen die Tests durch, andernfalls nicht und Sie müssen den Monitor neu starten. Die Anzahl der ausgeführten Tests ist allerdings auf 50 beschränkt.



Das Fenster mit der Rechenaufgabe ist gelegentlich durch andere Anwendungen verdeckt. Dann „hängt“ der PDFUnit-Monitor. Sie finden das Dialogfenster, wenn Sie alle Anwendungsfenster minimieren.

Lizenzschlüssel beantragen

Der PDFUnit-Monitor und PDFUnit-Java sind im kommerziellen Umfeld lizenzpflichtig und benötigen eine Lizenz. Schreiben Sie ein Mail an [license\[at\]pdfunit.com](mailto:license[at]pdfunit.com), Sie erhalten umgehend eine Antwort.

Die Lizenzkosten werden individuell gestaltet. Ein kleines Unternehmen muss nicht genauso viel zahlen, wie ein großes Unternehmen. Und wer nur wenige PDF-Dokumente testet, zahlt selbstverständlich auch weniger. Sollten Sie in den Besitz einer kostenlosen Lizenz kommen wollen, lassen Sie sich Argumente einfallen - es ist möglich.

Lizenzschlüssel installieren

Wenn Sie eine Lizenz beantragt haben, erhalten Sie eine ZIP-Datei mit dem PDFUnit-Monitor und eine separate Datei `license-key_pdfunit-java.lic`. Die Lizenzdatei wird von der Anwendung im Classpath gesucht. Am einfachsten ist es, sie in das Home-Verzeichnis von PDFUnit-Java zu kopieren.

Jede Änderung an der Lizenzdatei macht diese unbrauchbar. Nehmen Sie in einem solchen Falle mit PDFUnit.com Verbindung auf und beantragen Sie eine neue Lizenzdatei.

Überprüfung der Installation

Wenn Sie Probleme mit der Konfiguration haben, starten Sie das Skript zur Überprüfung der Installation: `verifyInstallation.bat` oder `verifyInstallation.sh`. Es ist in Kapitel 4.5: „Überprüfung der Konfiguration“ (S. 26). ausführlich beschrieben.

4.3. Installation bei existierendem iText

PDFUnit mit vorhandener iText-Version nutzen

PDFUnit nutzt iText (<http://itextpdf.com>) und andere Bibliotheken. iText seinerseits benötigt auch weitere Bibliotheken. Aus Lizenzgründen wird PDFUnit ohne die iText-Bibliotheken ausgeliefert, aber mit allen sonstigen Zusatzbibliotheken.

Achten Sie bei der Konfiguration des Classpath darauf, dass Bibliotheken, die sowohl iText als auch PDFUnit nutzen, nur einmal registriert sind. Auch müssen die Versionen der Fremdbibliotheken sowohl für iText, als auch für PDFUnit passen.

Sollte es bei diesen Fremdbibliotheken zu Versionskonflikten kommen, müssen Sie über eine geeignete Classpath-Konfiguration dafür sorgen, dass PDFUnit die Fremdbibliotheken verwendet, die PDFUnit ausliefert, und Ihre PDF-erstellenden Programme die Fremdbibliotheken nutzen, die zu Ihrer iText-Version passen.

PDFUnit mit zusätzlicher iText-Version nutzen

Angenommen, Sie verwenden für die Erstellung ihrer PDF-Dokumente eine alte Version von iText und möchten oder können diese Version nicht aktualisieren. In diesem Falle benötigen Sie zwei getrennte Classpath-Konfigurationen. PDFUnit muss mit einer passenden neueren Version von iText benutzt werden. Beachten Sie aber, dass iText ab Version 5.3.3 für den kommerziellen Einsatz lizenzpflichtig ist.

4.4. Pfade über Systemumgebungsvariablen setzen

Die Dateien `pdfunit-monitor.config` und die Lizenzdatei können auch außerhalb des Classpath's liegen, wenn deren Orte über entsprechende Java-Runtime Umgebungsvariablen deklariert werden. Die Umgebungsvariablen lauten:

- `-Dpdfunit.configfile`
- `-Dpdfunit.licensekeyfile`

Diese Parameter können auf mehrere Weise gesetzt werden. Einerseits können Sie in das Startskript 'startPDFUnitMonitor.bat' (bzw. *.sh) eingetragen werden. Andererseits kann auch die Betriebssystem-Umgebungsvariable `_JAVA_OPTIONS` dafür verwendet werden:

```
set _JAVA_OPTIONS=-Dpdfunit.configfile=..\myfolder\pdfunit-monitor.config
```

Sollten Sie zu diesem Thema Fragen haben, schreiben Sie ein Mail an: [info\[at\]pdfunit.com](mailto:info[at]pdfunit.com).

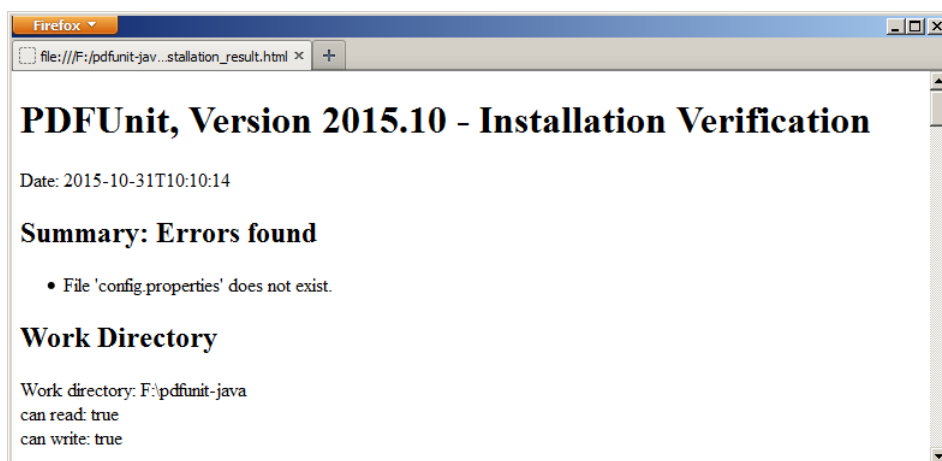
4.5. Überprüfung der Konfiguration

Die Installation vom PDFUnit-Monitor kann mit einem mitgelieferten Programm überprüft werden. Das Programm wird über das Skript `verifyInstallation.bat` bzw. `verifyInstallation.sh` gestartet:

Das Skript erzeugt folgende Ausgabe auf der Konsole:

```
Checking installation ...  
... finished. Report created, see 'verifyInstallation_result.html'.
```

Der Report listet einerseits eventuelle Fehler auf und andererseits protokolliert er allgemeine Laufzeitinformationen wie Classpath, Umgebungsvariablen und Dateien:



4.6. Installation eines neuen Releases

Die Installation eines neuen Releases von PDFUnit-Monitor verläuft genauso, wie die Erstinstallation, weil Releases immer vollständig zur Verfügung gestellt werden, nie als Differenz zum vorhergehenden Release.

Beschaffung des neuen Releases

Wenn Sie PDFUnit ohne Lizenzdatei einsetzen, laden Sie sich die neue ZIP-Datei aus dem Internet: <http://www.pdfunit.com/de/download/index.html>.

Wenn Sie PDFUnit mit Lizenzdatei einsetzen, erhalten Sie das neue Release per Mail mit der ZIP-Datei als Anhang sowie einer separaten Datei mit den Lizenzdaten.

Vorbereitende Schritte

Bevor Sie mit dem Releasewechsel beginnen, führen Sie alle vorhandenen Unittests mit dem alten Release durch. Diese sollten „grün“ sein.

Sichern Sie Ihr Projekt.

Durchführung des Updates

Entpacken Sie das neue Release in ein neues Verzeichnis. Schauen Sie in die Datei `release-notes.txt`.

Kopieren Sie die zuvor verwendete Datei `pdfunit-monitor.config` in das neue Verzeichnis, damit Ihre alten Einstellungen auch im neuen Release verfügbar sind. Falls sich die config-Datei für das neue Release geändert hat, steht ein entsprechender Hinweis in der Datei `release-notes.txt`. In dem Fall übertragen Sie die Inhalte der alten Datei in die entsprechenden Zeilen der neuen Datei.

Kopieren Sie die neue Lizenzdatei `license-key_pdfunit-java.lic` an den Ort, an dem sie beim alten Release lag.

Letzter Schritt

Führen Sie Ihre bestehenden Tests mit dem neuen Release durch. Sofern es keine dokumentierten Inkompatibilitäten zwischen dem alten und neuen PDFUnit-Release gibt, sollten Ihre Tests erfolgreich durchlaufen. Andernfalls lesen Sie die Release-Informationen.

4.7. Deinstallation

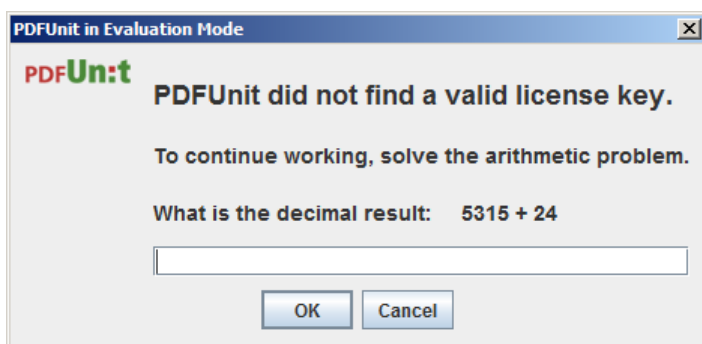
Analog zur Installation „per Copy“ wird PDFUnit durch das Löschen der Installationsverzeichnisse wieder sauber deinstalliert. Einträge in Systemverzeichnisse oder in die Registry können nicht zurückbleiben, weil solche nie erstellt wurden. Vergessen Sie nicht, in Ihren eigenen Skripten die Referenzen auf JAR-Dateien oder Verzeichnisse von PDFUnit zu entfernen.

Kapitel 5. Starten der Anwendung

Im Basisverzeichnis der Installation von PDFUnit-Monitor befinden sich die Startskripte 'startPDFUnit-Monitor.bat' bzw. 'startPDFUnitMonitor.sh'. Sofern sich auch die JAR-Dateien von 'iText' im Class-path befinden, startet der PDFUnit-Monitor, wenn die Skripte ausgeführt werden. Die gestartete Java-Klasse ist `com.pdfunit.monitor.PDFUnitMonitorMain`.

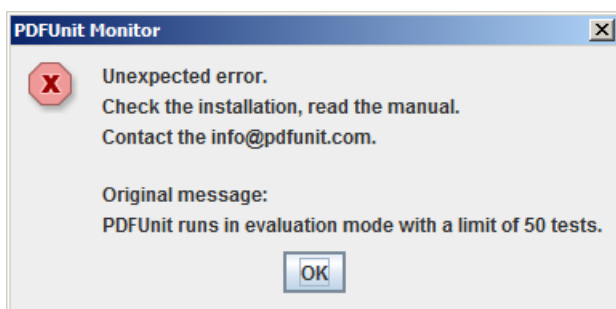
Nach dem Start im Evaluation-Modus

Der Start des Monitors mit einer Lizenz unterscheidet sich nicht vom Start im Evaluation-Modus. Wenn dann allerdings die Validierung angestoßen wird, erscheint ein kleines Fenster, in dem der Benutzer eine einfache Rechenaufgabe lösen muss. Diese Benutzerinteraktion verhindert, dass der Monitor und das darunter liegende PDFUnit-Java im produktiven Einsatz ohne Lizenz benutzt wird.



Wird die Rechenaufgabe richtig beantwortet, laufen bis zu 50 Prüfungen. Die Zählung beginnt mit dem Start des Monitors. Dieses Limit sollte reichen, die Nützlichkeit des Monitors für die eigenen Projektansprüche zu validieren. Nach einem Neustart beginnt die Zählung von vorne.

Wenn das Limit überschritten wird, erscheint folgende Meldung:



Wird die Rechenaufgabe falsch beantwortet, erscheint eine Meldung über die falsche Eingabe. Nach einer solchen falschen Eingabe muss der PDFUnit-Monitor erneut gestartet werden, da das Programm die Eingabe intern speichert und nicht erneut abfragt.

Bedenken Sie, dass diese Einschränkungen der Handhabung nur im Evaluation-Modus bestehen. Mit einer Lizenz gibt es kein Limit und keine Rechenaufgabe.

Kapitel 6. Konfiguration (pdfunit-monitor.config)

In der Datei `pdfunit-monitor.config` müssen die Namen der Verzeichnisse eingetragen werden, die vom PDFUnit-Monitor überwacht werden sollen. Darüber hinaus können weitere Parameter bei Bedarf eingestellt werden.

Überwachte Verzeichnisse

Die Verzeichnisse, die vom PDFUnit-Monitor überwacht werden sollen, müssen in der Konfiguration eingetragen werden. Dazu existieren drei Parameter.

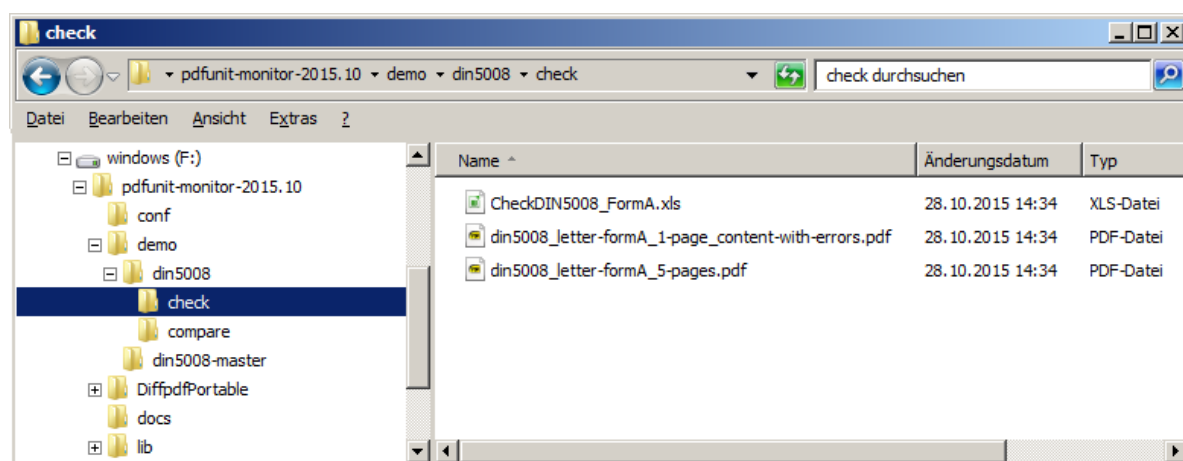
`WATCHED_FOLDER` ist das oberste Verzeichnis, auf das sich der PDFUnit-Monitor positioniert. Von hier aus werden alle Unterverzeichnisse überwacht.

Mit dem Parameter `WATCHED_FOLDER_FOR_COMPARISON` wird eines dieser Unterverzeichnisse für Vergleiche von Test-PDF-Dokumenten mit geprüften Master-Dokumenten deklariert. Auch PDF-Dokumente in den Unterverzeichnissen von `WATCHED_FOLDER_FOR_COMPARISON` werden mit Master-PDF-Dokumenten verglichen.

Das Verzeichnis, in denen der PDFUnit-Monitor die Master-PDF-Dokumente sucht, wird mit dem Parameter `FOLDER_WITH_MASTERDOCUMENTS` angegeben. Die Verzeichnisstrukturen unterhalb von `FOLDER_WITH_MASTERDOCUMENTS` und unterhalb von `WATCHED_FOLDER_FOR_COMPARISON` müssen identisch sein, damit der PDFUnit-Monitor die Dokumente findet. Ebenfalls müssen das Test-Dokument und ein Vergleichsdokument denselben Dateiname haben.

Nachfolgend die Konfiguration und die Verzeichnisstruktur für die im Release mitgelieferten Dokumente:

```
#
# Base folder for PDF document which are validated.
# All subfolders will be watched too.
#
WATCHED_FOLDER = ./folder-to-watch/din5008
WATCHED_FOLDER_FOR_COMPARISON = ./folder-to-watch/din5008/compare
FOLDER_WITH_MASTERDOCUMENTS = ./folder-to-watch/din5008-master
```



Logging-Port

Der PDFUnit-Monitor benutzt intern Log4j2 als Logging-Framework und verwendet dazu einen Port. Die Standardeinstellung ist Port 4445. Falls dieser Port belegt ist, muss ein anderer in der Datei `pdfunit-monitor.config` festgelegt werden. Wichtig: derselbe Port muss in der Datei `log4j2.xml` konfiguriert sein.

```
#
# The port defined here must be the same as in the configuration file
# for Log4J "log4j2.xml". Look for "SocketAppender" in that file.
#
PORT_TO_LISTEN = 4445
```

Neben der Konfigurationsdatei `pdfunit-monitor.config` gibt es für die Konfiguration des im Hintergrund laufenden Loggings noch die Konfigurationsdatei `log4j2.xml`. Weitere Informationen in Kapitel 8: „Logging mit Log4j2“ (S. 33).

Installationsverzeichnis von DiffPDF

Wenn zwischen zwei Dokumenten, die miteinander verglichen werden, ein Unterschied erkannt wird, ist es möglich, beide Dokumente mit einem Diff-Programm anzuzeigen. Der PDFUnit-Monitor benutzt dazu die Anwendung 'DiffpdfPortable', die in einem Verzeichnis unterhalb von `PDFUNIT_MONITOR_HOME` mitgeliefert wird. Sollten Sie diese Anwendung in ein anderes Verzeichnis legen wollen, müssen die nachfolgenden Parameter entsprechend angepasst werden. Wichtig: der Verzeichnisname muss mit einem 'forward slash' abgeschlossen werden.

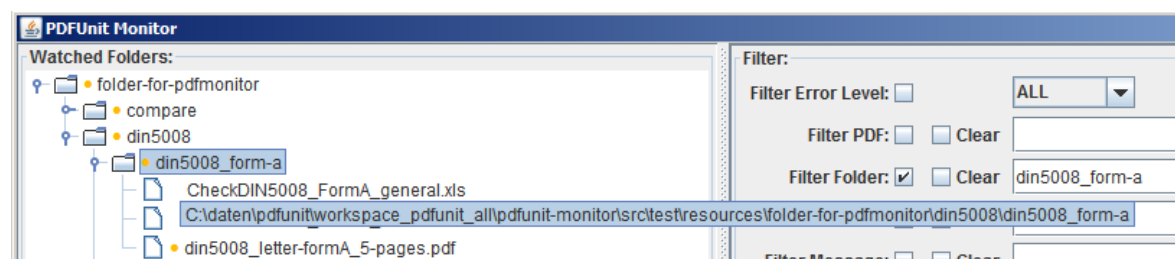
```
#
# Installation definitions for DiffPDF, the free Viewer.
# (Don't forget the trailing slash for the folder.)
# Information about DiffPDF are here: http://www.qtrac.eu/diffpdf.html
#
DIFFPDF_FOLDER      = DiffpdfPortable/
DIFFPDF_EXECUTABLE  = DiffpdfPortable.exe
```

Zeitverzögerung für die Tooltip-Anzeige

In der Ansicht der Verzeichnisstruktur wird der reale Dateiname über einen Tooltip angezeigt, wenn der Mauszeiger auf einer Datei oder einem Verzeichnis steht. Die Verzögerung, bis der Tooltip erscheint, kann über den Parameter `TOOLTIP_VISIBILITY_DURATION_MILLISECONDS` eingestellt werden.

```
#
# The following property sets the wait time,
# before a tooltip is visible.
#
TOOLTIP_VISIBILITY_DURATION_MILLISECONDS = 1500
```

Hier ein Beispiel für einen Tooltip:

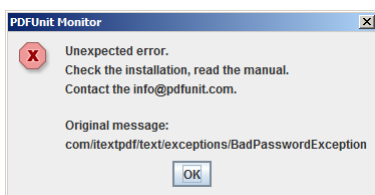


Kapitel 7. Typische Fehler

Zur Laufzeit kann es system- und situationsbedingt Fehlermeldung geben. Um diese besser zu verstehen und deren Ursachen zu beseitigen, werden nachfolgend typische Situationen und ihre Meldungen vorgestellt.

JAR-Dateien von iText nicht im Classpath

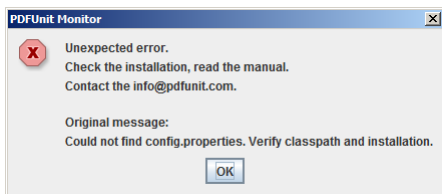
Die benötigten JAR-Dateien von iText werden aus Lizenzgründen nicht mit ausgeliefert. Wenn Sie den Monitor ohne diese Dateien starten, erscheint folgende Fehlermeldung.



Sie können iText zu Evaluationszwecken über Link. beziehen. Beachten Sie aber, dass iText im kommerziellen Umfeld lizenzkostenpflichtig ist.

Konfigurationsdatei 'pdfunit.config' nicht im Classpath

Aus welchem Grund auch immer eine Datei aus dem Installationsverzeichnis verschwindet - der PDFUnit-Monitor reagiert auf das Fehlen mit einer entsprechenden Meldung:



Die beste Art, zu kontrollieren, ob alle benötigten Dateien auch gefunden werden können, ist, das Skript `verifyInstallation.bat` (oder `*.sh`) zu starten. Die anschließend erzeugte HTML-Datei `verifyInstallation_result.html` muss den Text 'Summary: No Errors' enthalten.

Evaluation-Modus von PDFUnit-Monitor - Limit erreicht

Im Evaluierungs-Modus können mit dem PDFUnit-Monitor maximal 50 Testfälle durchgeführt werden. Wird diese Zahl überschritten, erscheint folgende Meldung:



Sie müssen dann die Anwendung neu starten. Selbstverständlich besteht die Möglichkeit, eine Testlizenz für ein ungestörtes Testen zu beantragen. Schreiben Sie dafür ein Mail an [license\[at\]pdfunit.com](mailto:license[at]pdfunit.com).

Kapitel 8. Logging mit Log4j2

Alle Validierungsergebnisse einer PDF-Validierung werden intern an das Logging-System 'Log4j Version 2' übergeben. Diese vielleicht aufwendig erscheinende Implementierung ermöglicht es, weitere Empfänger auf einfache Weise hinzuzufügen. Weitere Empfänger können nicht nur Dateien sein, sondern auch Log-Management-Systeme wie beispielsweise Graylog2 und LogStash. Geeignete Appender für diese Systeme sind im Internet verfügbar.

Die Konfigurationsdatei `log4j2.xml` befindet sich im Verzeichnis `./conf`

Aktuelle Konfiguration

Im Auslieferungszustand ist Log4j2 so konfiguriert, dass neben der Benutzung des internen Appenders 'PDFUnitMonitorAppender' auch ein File-Appender aktiviert ist. Dieser schreibt alle Validierungsergebnisse im XML-Format in die Datei `./log/pdfunit-monitor.log`.

```
<Configuration monitorInterval="10">
  <Appenders>
    <File name="XmlFileAppender"
          fileName="./log/pdfunit-monitor.log"
          append="true"
          bufferedIO="false"
          bufferSize="100"
          immediateFlush="true"
        >
      <XMLLayout complete="true" charset="UTF-8" compact="false"/>
    </File>
    <Console name="ConsoleAppender" target="SYSTEM_OUT">
      <PatternLayout pattern="%d %-5p [%t] %C{2} (%F:%L) - %m%n"/>
    </Console>
    <Socket name="PDFUnitMonitorAppender" host="localhost" port="4445">
      <SerializedLayout />
    </Socket>
  </Appenders>

  <Loggers>
    <Logger name="PDFConstraints" level="all" additivity="false">
      <AppenderRef ref="XmlFileAppender"/>
      <AppenderRef ref="PDFUnitMonitorAppender"/>
    <!-- <AppenderRef ref="ConsoleAppender"/> -->
    </Logger>
    <Root level="off">
      <AppenderRef ref="ConsoleAppender"/>
    </Root>
  </Loggers>
</Configuration>
```

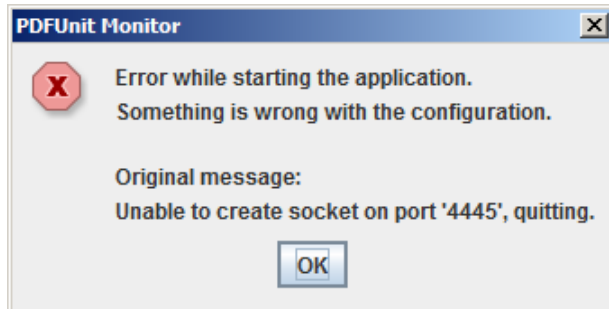
Ein dritter Appender, der Konsolen-Appender, ist auskommentiert. Wenn er aktiviert wird, werden alle Ergebnisse auch auf die Konsole geschrieben. Diese ist aber erst sichtbar, wenn die letzte Zeile in der Startdatei `startPDFUnitMonitor.bat` bzw. `*.sh` auf Kommentar gesetzt und die vorletzte Zeile vom Kommentar befreit wird.

```
:: Start the application:
start java com.pdfunit.monitor.PDFUnitMonitorMain
::start javaw com.pdfunit.monitor.PDFUnitMonitorMain
```

In der Log-Konfiguration kann der Error-Level als Filter angegeben werden. Mit der Auslieferung steht er für den Logger `PDFConstraints` auf `all`. Falls er auf `warning` gesetzt wird, erscheinen keine 'OK'-Meldungen mehr in der Übersichtsliste. Es ist aber nicht nötig, die 'OK'-Meldungen schon bei ihrer Entstehung herauszufiltern. Die Oberfläche bietet einen Error-Level-Filter an, um die Anzeige später einzuschränken.

Interne Nutzung, Port neu konfigurieren

Der Logging-Appender 'PDFUnitMonitorAppender' schickt seine Prüfergebnisse über Sockets an die Oberfläche. Dafür wird Port 4445 verwendet. Wenn dieser Port von anderen Anwendungen belegt ist, muss er für den Monitor in der Konfigurationsdatei `log4j2.xml` auf einen anderen Wert gesetzt werden. Falls die Anwendung gestartet wird und der Port schon belegt ist, erscheint folgende Fehlermeldung:



Diese Meldung erscheint auch, wenn der PDFUnit-Monitor ein zweites Mal gestartet wird.

Informationen zu Log4j2

Weitere Informationen zu Log4j-2 entnehmen Sie bitte Online-Dokumentationen, auf die die folgenden Links verweisen:

- Projekt-Home: <http://logging.apache.org/log4j/2.x/index.html>.
- Wiki: <https://wiki.apache.org/logging/log4j2>.
- Apache Log4j 2, User's Guide: [log4j-users-guide.pdf](#).

Stichwortverzeichnis

A

Anwendung starten, 29

C

Classpath, 24

D

Darstellung

 Fehlerdetails, 20

 Fehlerliste, 17

 überwachte Verzeichnisse, 14

Deinstallation, 28

DiffPDF, 20

 Export, 22

 Konfiguration, 31

E

Error-Level, 19

Erwarteter Text, 13

Evaluationsversion, 25

Excel-Datei

 Aufbau, 10

 Sheets, 10

Excel-Sheet

 check, 11

 compare, 14

 region, 10

Export, 22

F

Feedback, 5

Fehlerdetails, 20

Fehlerliste, 17

 Error-Level, 19

 Filter, 18

 sortieren, 19

Fehlermeldungen, 13

Filter, 18

Funktionalität, 10

I

Import, 22

Installation, 24

 bei existierendem iText, 26

 Classpath, 24

 iText, 24

 Lizenzschlüssel, 25

 neues Release, 27

 ohne vorhandenes iText, 24

 überprüfen, 26

iText, 24

 installieren, 24

K

Konfiguration, 30

 DiffPDF, 31

 Logging-Port, 30

 Tooltip, 31

 überwachte Verzeichnisse, 30

L

Leerzeichen, 13

Lizenzschlüssel

 beantragen, 25

 installieren, 25

Log4j2

 Logging, 33

 Port belegt, 34

Q

Quickstart, 6

 Export, 9

 Fehlerdetails, 7

 Filter, 7

 Import, 9

 überwachte Verzeichnisse, 6

S

Seitenbereiche, 10

Startskript, 24

Systemumgebungsvariablen, 26

T

Technische Voraussetzungen, 24

Testfall

 erwarteter Text, 13

 Fehlermeldungen, 13

 Leerzeichen, 13

Typische Fehler, 32

 kein iText, 32

 Konfigurationsdatei nicht gefunden, 32

 Limit erreicht, 32

U

Überwachte Verzeichnisse, 14

Umgebungsvariablen, 26

Update, 27

V

Vergleich gegen Vorlage, 8

W

Whitespaces, 13