
PDFUnit-Monitor

Handbuch für Benutzer

Carsten Siedentop

Inhaltsverzeichnis

Vorwort	4
1. Über diese Dokumentation	5
2. Quickstart	6
3. Funktionalität	10
3.1. Darstellung - Überwachte Verzeichnisse	10
3.2. Darstellung - Liste der Prüfergebnisse, Filter	12
3.3. Darstellung - Details eines Validierungsergebnisses	15
3.4. DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente	15
3.5. Export und Import der Ergebnisse	17
4. Validierungsregeln in Excel-Dateien	18
5. Installation, Update, Deinstallation	24
5.1. Technische Voraussetzungen	24
5.2. Installation	24
5.3. Pfade über Systemumgebungsvariablen setzen	25
5.4. Überprüfung der Konfiguration	26
5.5. Installation eines neuen Releases	26
5.6. Deinstallation	27
6. Starten der Anwendung	28
7. Konfiguration (pdfunit-monitor.config)	29
8. Typische Fehler	32
9. Logging mit Log4j2	34
Stichwortverzeichnis	36

Vorwort

Aktuelle Testsituation in Projekten

Telefonrechnungen, Versicherungspolicen, amtliche Bescheide, Verträge jeglicher Art werden heute als PDF-Dokumente elektronisch zugestellt. Ihre Erstellung erfolgt in vielen Programmiersprachen mit zahlreichen Bibliotheken. Je nach Komplexität der zu erstellenden Dokumente ist diese Programmierung nicht einfach und enthält wie jede Software auch Fehler, die eventuell zu fehlerhaften PDF Dokumenten führen. Deshalb sollte geprüft werden:

- Steht in einem bestimmten Bereich einer Seite der erwartete Text?
- Stimmt der Barcode (QR-Code) auf dem Dokument mit dem erwarteten Inhalt überein?
- Stimmt das Layout mit der Vorgabe überein?
- Stimmen die Werte der eingebetteten ZUGFeRD-Daten mit den erwarteten Daten überein?
- Stimmen die Werte der eingebetteten ZUGFeRD-Daten mit den sichtbaren Daten überein?
- Entspricht ein Dokument den Regeln von DIN 5008?
- Ist das PDF signiert? Wann und von wem?

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. Das ist teuer und fehleranfällig.

Egal, ob PDF-Dokumente mit einem mächtigen Design-Werkzeug, mit MS-Word/LibreOffice oder eigenen Programmen erstellt werden oder ob sie aus einem XSL-FO Workflow herausfallen, jedes PDF-Dokument kann mit PDFUnit getestet werden.

Ein User-Interface für automatisiertes Testen - ein Widerspruch?

Ein Programm, das von einem Benutzer bedient werden kann, scheint ein Widerspruch zum Ziel, automatisiert zu testen, zu sein. Dieser Widerspruch löst sich aber auf, denn der PDFUnit-Monitor 'beobachtet' Dateiverzeichnisse und führt automatisch Testfälle aus, sobald neue PDF-Dokumente in den Verzeichnissen gespeichert werden. So kann die IT-Abteilung Dokumente erstellen - egal, auf welche Weise - und der PDFUnit-Monitor zeigt unmittelbar das Ergebnis der vorhandenen Tests an, ohne dass Tests manuell angestoßen werden müssen.

Der PDFUnit-Monitor ist leicht zu bedienen. Testfälle werden in Excel erfasst und können deshalb auch von Nicht-Entwicklern entwickelt werden. Neben seinem autonomen Betrieb kann der PDFUnit-Monitor auch über die rechte Maustaste oder Tastatur bedient werden, sodass auch individuelle Tests durchgeführt werden können.

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 PDF-Dokumente automatisiert oder individuell überprüfen wollen.

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 info@pdfunit.com schicken.

Neue Testfunktionen gewünscht?

Hätten Sie gerne neue Testfunktionen, wenden Sie sich per Mail an info@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. Die PDF- und die HTML-Version stammen aus einer einzigen Textquelle und sind somit inhaltlich identisch. 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 andere Aufgaben mit höherer Priorität.

Feedback

Jegliche Art von Feedback ist willkommen, schreiben Sie einfach an info@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.

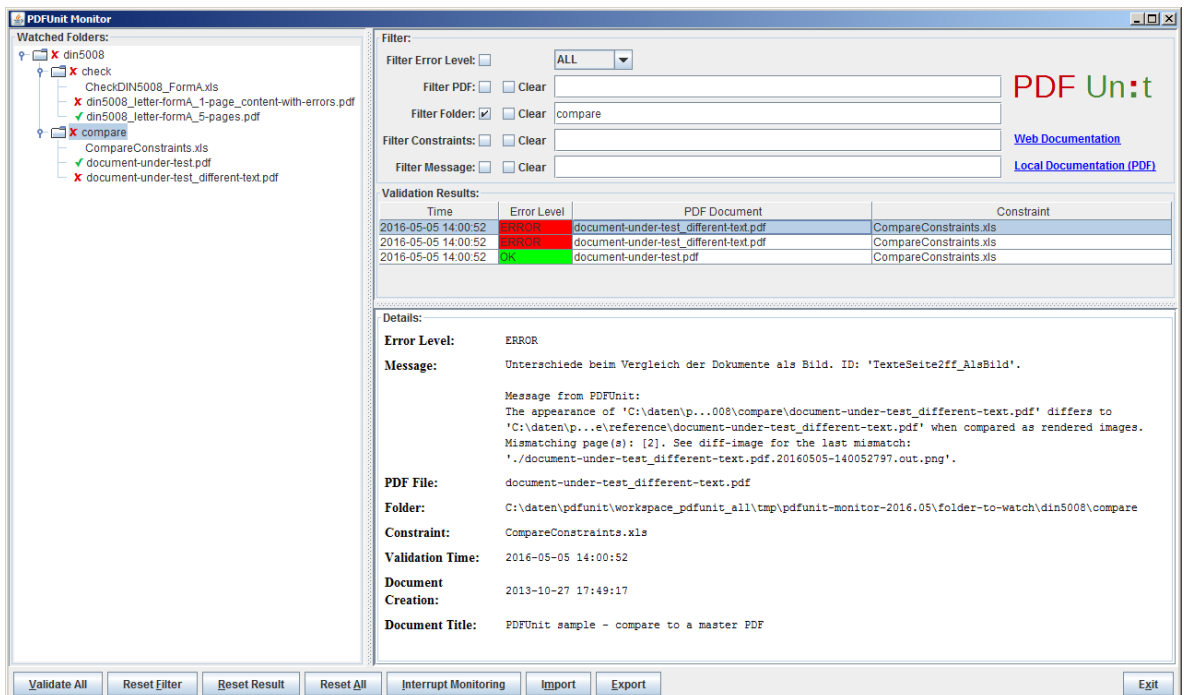
Der Funktionsumfang des PDFUnit-Monitors ist groß. Eine umfassende Beschreibung an dieser Stelle würde den Rahmen der vorliegenden Dokumentation sprengen. Deshalb existiert für ihn eine gesonderte Dokumentation und auch ein erklärendes Video. Beides kann über diesen Link ([Download](#)) von den Webseiten von PDFUnit heruntergeladen werden. Die separate Dokumentation beschreibt auch die Installation und Konfiguration des PDFUnit-Monitors. Die nachfolgenden Abschnitte beschreiben kurz die Hauptfunktionen.

Überwachte Verzeichnisse

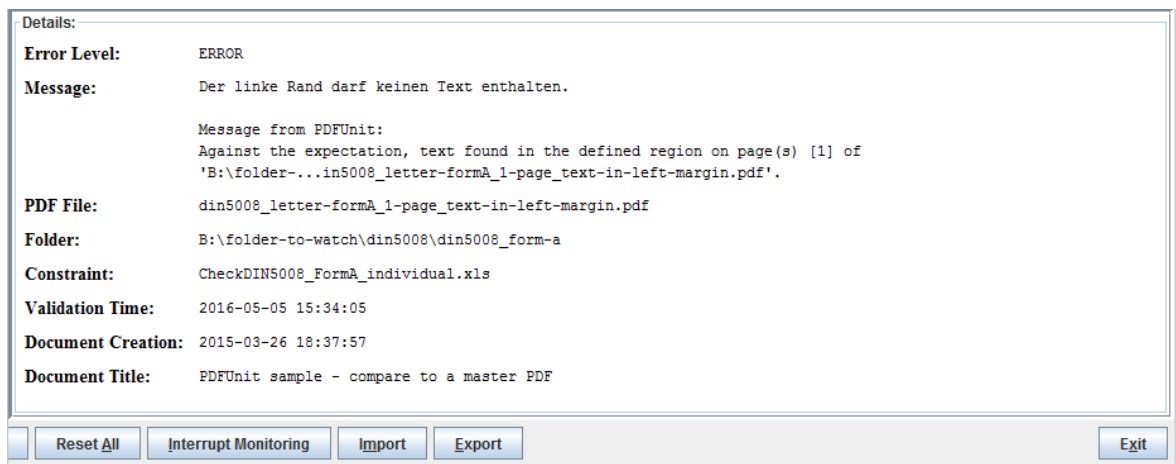
Der PDFUnit-Monitor überwacht alle PDF-Dokumente unterhalb eines definierten Verzeichnisses und prüft die dortigen Dokumente gegen Regeln, die in Excel-Dateien hinterlegt sind, die in den überwachten Verzeichnissen liegen müssen. Erfüllt ein PDF alle Regeln, wird es in der Baumstruktur mit einem grünen 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.

Verzeichnisse mit dem Namen 'reference' werden nicht überwacht. In ihnen müssen PDF-Dokumente abgelegt werden, die bei einer vergleichenden Prüfung als Referenz dienen.

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.



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.



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.

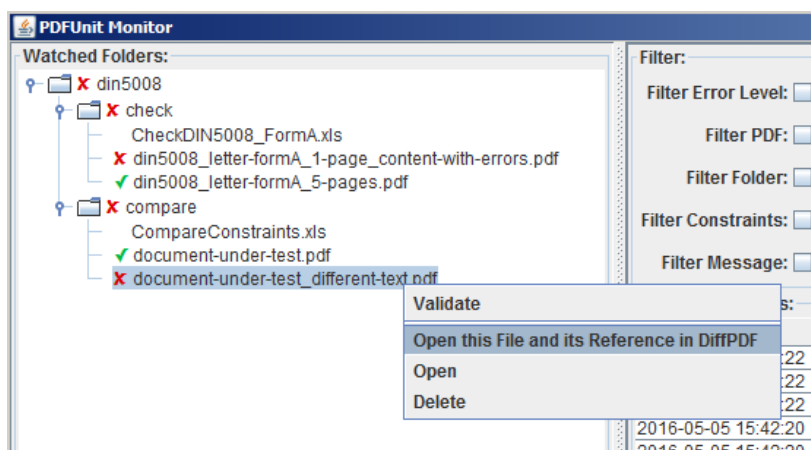
Die Fehlermeldungen von PDFUnit gibt es momentan in Deutsch und in Englisch. Weitere Sprachen können mit wenig Aufwand zur Verfügung gestellt werden.

Vergleich eines PDF-Dokumentes gegen eine Vorlage

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

Ein Referenz-Dokument muss den gleichen Namen haben, wie das 'PDF-Under-Test', und wird im Unterverzeichnis 'reference' des Ordners gesucht, in dem sich das 'PDF-Under-Test' befindet.

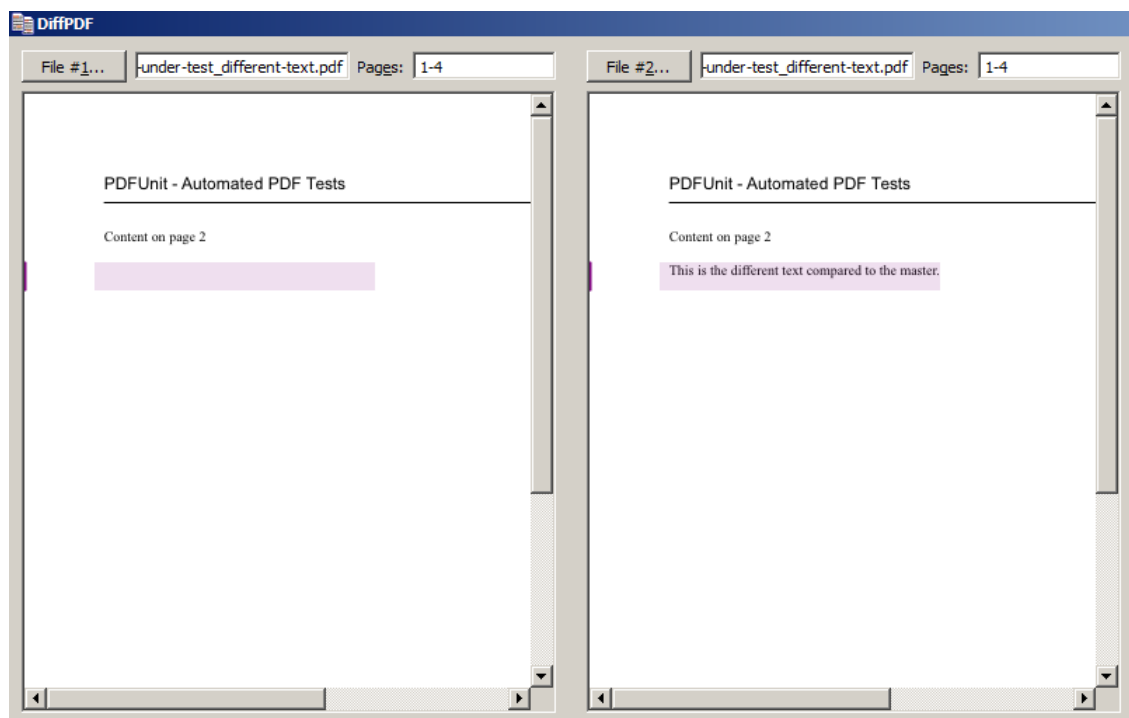
Wenn bei der Validierung ein Unterschied festgestellt wird, kann über die rechte Maustaste das Programm 'DiffPDF 1.5.1 portable' gestartet werden, das den Unterschied gut darstellt:



Das Programm stammt von Mark Summerfield und steht als 'Portable App' über diesen Link ([Download](#)) zum Download zur Verfügung. DiffPDF kann in 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.



Export und Import der Ergebnisse

Die Testergebnisse können über den Button 'Export' als XML-Datei exportiert werden und stehen damit auch 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 Sprachen Deutsch und 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. 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 [7: „Konfiguration \(pdfunit-monitor.config\)“ \(S. 29\)](#) ausführlich beschrieben.



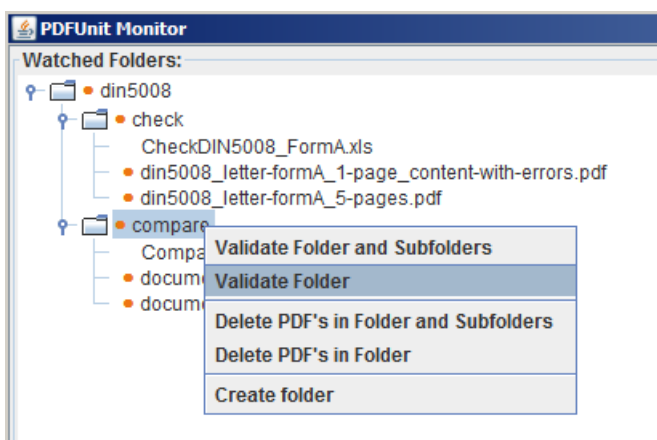
Der Monitor reagiert auf Veränderungen an den überwachten Dateien und Verzeichnissen. Das heißt, die Anzeige wird aktualisiert, wenn Dateien und Verzeichnisse neu hinzukommen, sich ändern oder gelöscht werden.

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

Hinweis: PDF-Dokumente, die mit einem Passwort geschützt sind, können mit dem PDFUnit-Monitor nicht geprüft werden. Über die Java-API von PDFUnit ist das aber möglich.

Kontextmenü

Wenn der Monitor neu gestartet wurde, kann die Validierung der vorhandenen PDF-Dokumente über den Button `Alles validieren` im unteren Bereich des Monitors angestoßen werden oder über das Kontextmenü:

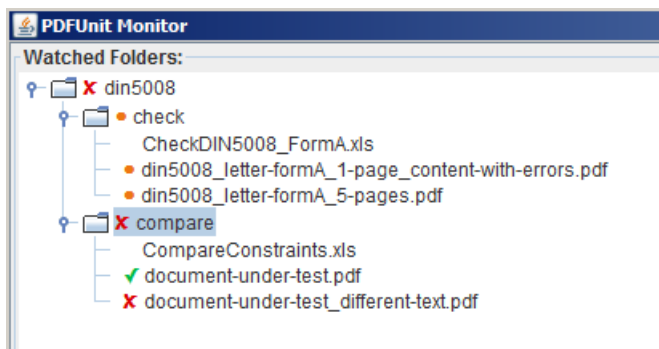


Selbstverständlich steht das Kontextmenü jederzeit zur Verfügung. Die Inhalte des Kontextmenüs wechseln, abhängig davon, ob sich das Menü auf ein Verzeichnis, eine PDF- oder eine Excel-Datei bezieht.

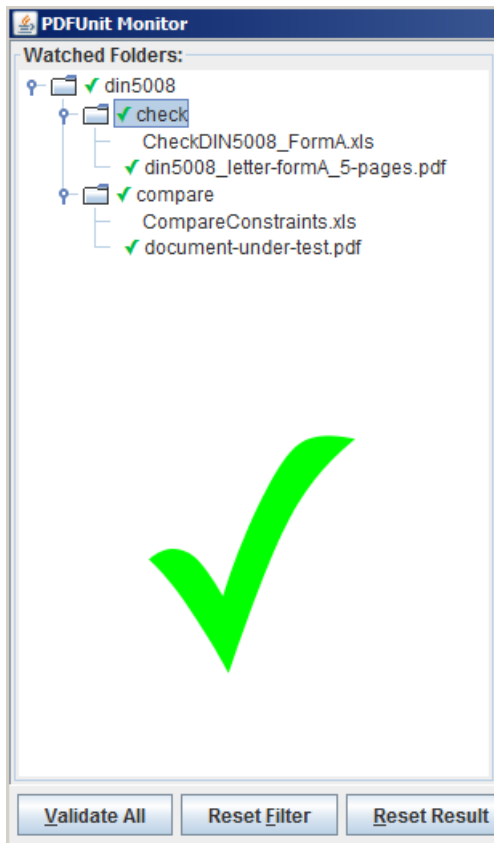
Farbliche Darstellung des Validierungsergebnisses

Das Validierungsergebnis eines PDF-Dokumentes wird durch ein farbiges Symbol vor seinem Namen dargestellt. Nur, wenn alle Regeln fehlerfrei durchlaufen wurden, wird der Name der PDF-Datei 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. Ein gelber Punkt zeigt an, dass dieses Dokument noch nicht validiert wurde, bzw. dass ein Verzeichnis noch ungeprüfte PDF-Dokumente enthält. Das nächste Bild zeigt dieses Verhalten.



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



Drag and Drop

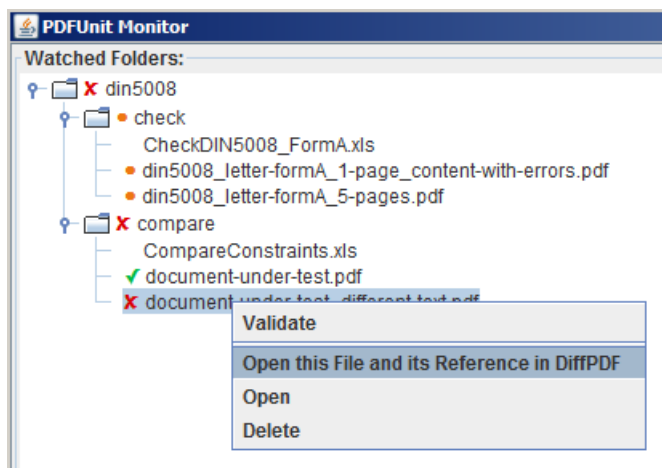
Der PDFUnit-Monitor unterstützt die 'Drop'-Funktion von 'Drag-and-Drop'. Das bedeutet, es können PDF-Dokumente per Maus an die angezeigten Verzeichnisse 'übergeben' werden. Die Dateien werden kopiert, nicht verschoben! Nach dem Kopieren startet automatisch die Validierung gegen die in dem Verzeichnis 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 'DiffPDF' wird in Kapitel [3.4: „DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente“ \(S. 15\)](#) genauer beschrieben.

3.2. 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 din5008

Filter Constraints: Clear

Filter Message: Clear

PDF Un:t

[Web Documentation](#)

[Local Documentation \(PDF\)](#)

Validation Results:

Time	Error Level	PDF Document	Constraint
2016-05-05 19:30:43	ERROR	document-under-test_different-text.pdf	CompareConstraints.xls
2016-05-05 19:30:43	ERROR	document-under-test_different-text.pdf	CompareConstraints.xls
2016-05-05 19:30:42	OK	document-under-test.pdf	CompareConstraints.xls
2016-05-05 19:30:41	OK	din5008_letter-formA_5-pages.pdf	CheckDIN5008_FormA.xls
2016-05-05 19:30:40	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2016-05-05 19:30:40	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2016-05-05 19:30:40	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2016-05-05 19:30:40	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2016-05-05 19:30:40	ERROR	din5008_letter-formA_1-page_content-with-errors.pdf	CheckDIN5008_FormA.xls
2016-05-05 19:30:40	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 die Textfelder rechts 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.
Filter Folder	Ein Validierungsergebnis landet in der Ergebnisliste, wenn sich ein PDF-Dokument unterhalb des Verzeichnisses befindet, dessen Name im Filter gesetzt wurde.
Filter Constraints	Ein Validierungsergebnis landet in der Ergebnisliste, wenn der eingegebene Wert Teil des Dateinamens einer Excel-Datei ist. irrelevant.
Filter Message	Ein Validierungsergebnis landet in der Ergebnisliste, wenn der eingegebene Wert Teil der Fehlermeldung ist.

Für alle Filter die Klein-/Großschreibung irrelevant.

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

Die Liste dient aber nicht nur zur Darstellung. Aus der Liste heraus können auch Dokumente geöffnet werden. Ein Doppel-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

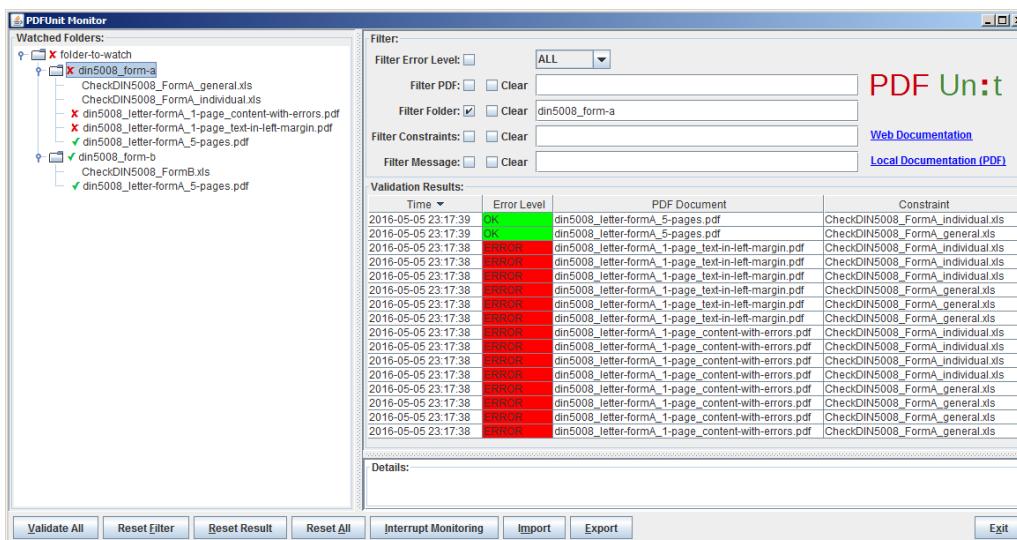
Die im 'Filter Error Level' angezeigten Werte haben folgende Bedeutung:

Error-Level	Bedeutung
OK	Ein PDF-Dokument erfüllt alle Regeln aller Excel-Dateien im selben Verzeichnis.

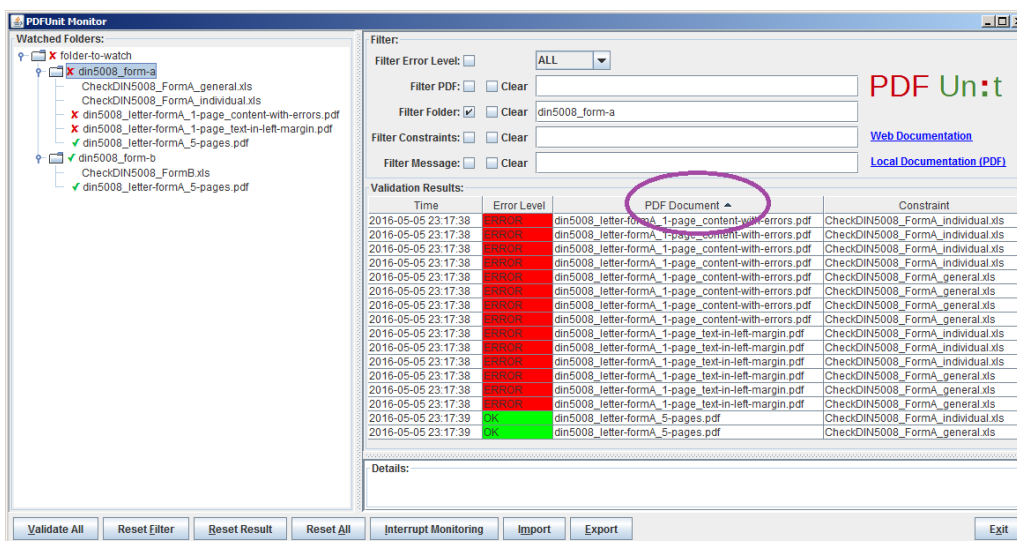
Error-Level	Bedeutung
ERROR	Ein PDF-Dokument verletzt eine oder mehrere Regeln in einer der Excel-Dateien desselben Verzeichnisses.
WARNING	Eine Validierung konnte nicht durchgeführt werden. Das kann beispielsweise der Fall sein, wenn in dem Verzeichnis keine Excel-Datei existiert, oder wenn eine Excel-Datei selber Fehler enthält.
INFO	Eine Excel-Datei wurde einem Verzeichnis hinzugefügt oder aus ihm entfernt.
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.

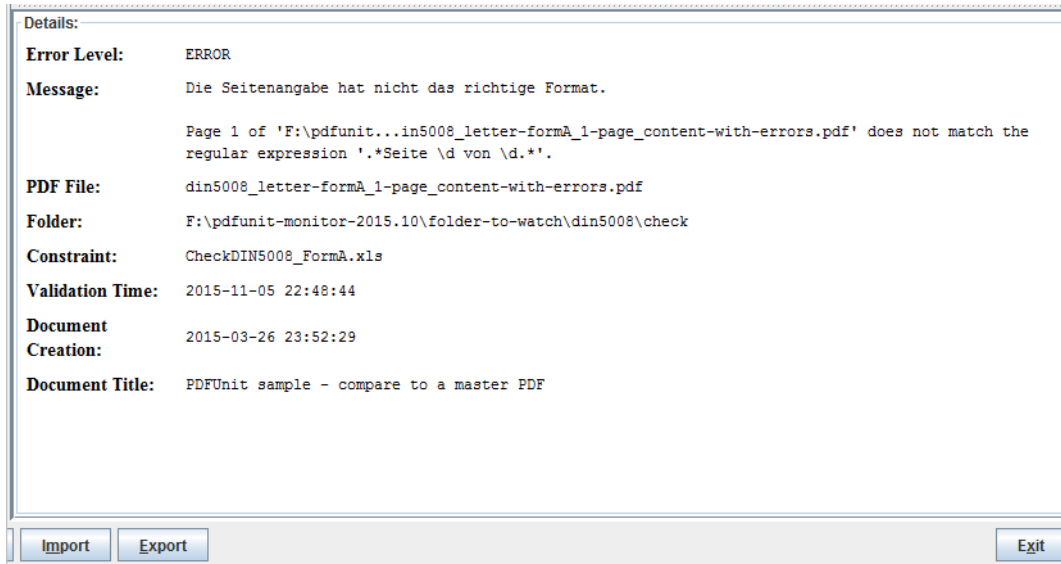


Das vorhergehende Bild zeigt die Standardsortierung, nämlich absteigend über die Spalte 'Time'. Nach einem Mausklick auf den Spaltentitel 'PDF Document' ändert sich die Sortierung, wie das folgende Bild zeigt.



3.3. 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.



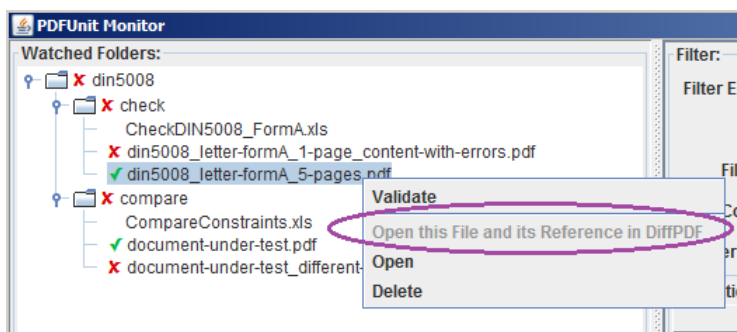
Die erste Zeile in der 'Message' stammt aus der Excel-Datei. Sie kann vom Entwickler des Testfalls beliebig verfasst werden, auch Platzhalter für variable Inhalte sind möglich. Die weiteren Zeilen in 'Message' sind das Ergebnis der Validierung mit PDFUnit.

Die Fehlermeldungen und auch die Bezeichnungen vor jeder Spalte können in Englisch oder Deutsch erscheinen. Die Sprache wird von Java automatisch aus der Betriebssystemumgebung abgeleitet. Soll der PDFUnit-Monitor in einer anderen Sprache als der des Betriebssystems verwendet werden, so kann die Sprache über die JVM-Properties gesetzt werden, beispielsweise auf Englisch: `-Duser.language=en -Duser.country=UK`.

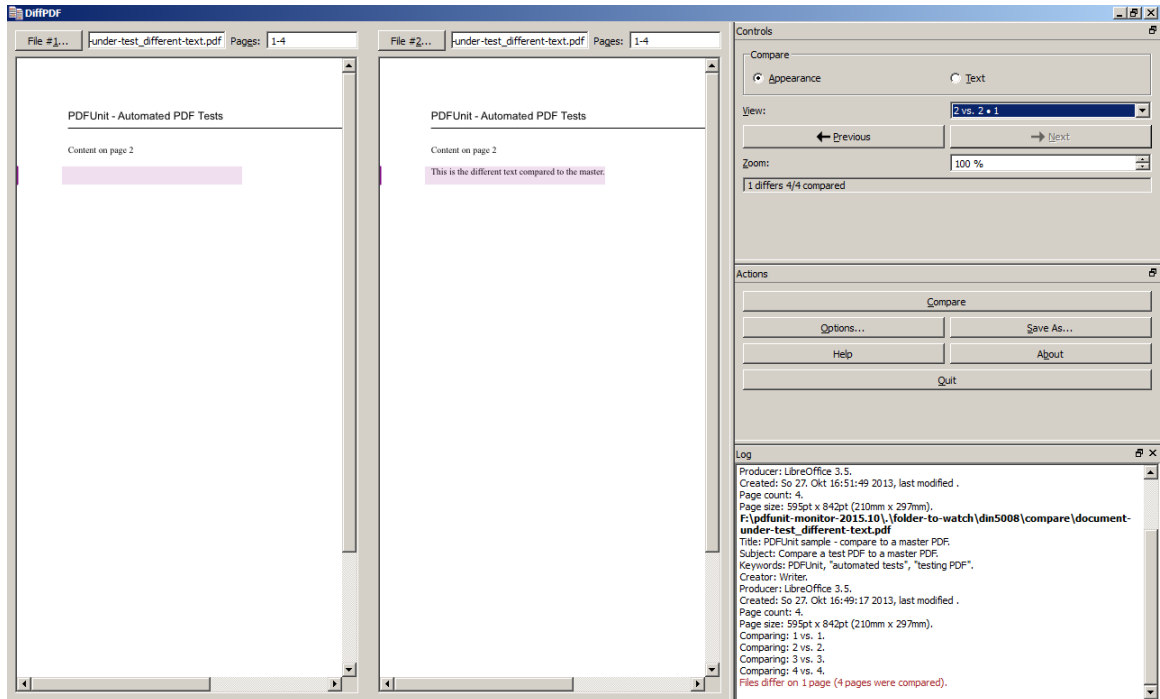
Die Anwendung ist aber strukturell darauf vorbereitet, diese Inhalte in anderen Sprachen zur Verfügung zu stellen. Schreiben Sie Ihren Wunsch an [info\[at\]pdfunit.com](mailto:info[at]pdfunit.com).

3.4. DiffPDF - Vergleichende Anzeige zweier PDF-Dokumente

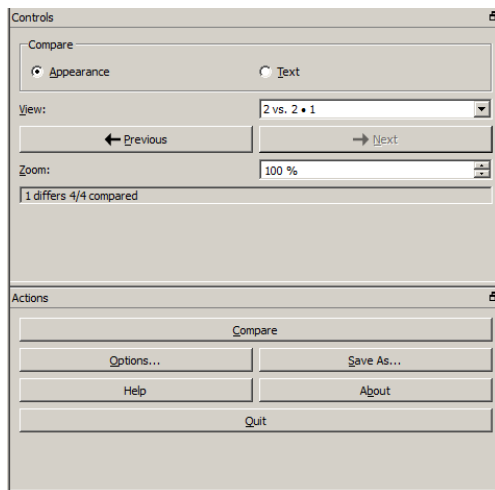
Wie schon zuvor im Kapitel 2: „Quickstart“ (S. 6) beschrieben, können Unterschiede zwischen einem Test-PDF und einer Vergleichsvorlage mit dem Programm 'DiffPDF' dargestellt werden. Das Programm wird über das Kontextmenü einer PDF-Datei gestartet. Ein entsprechender Menüpunkt ist aber nur aktiviert,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



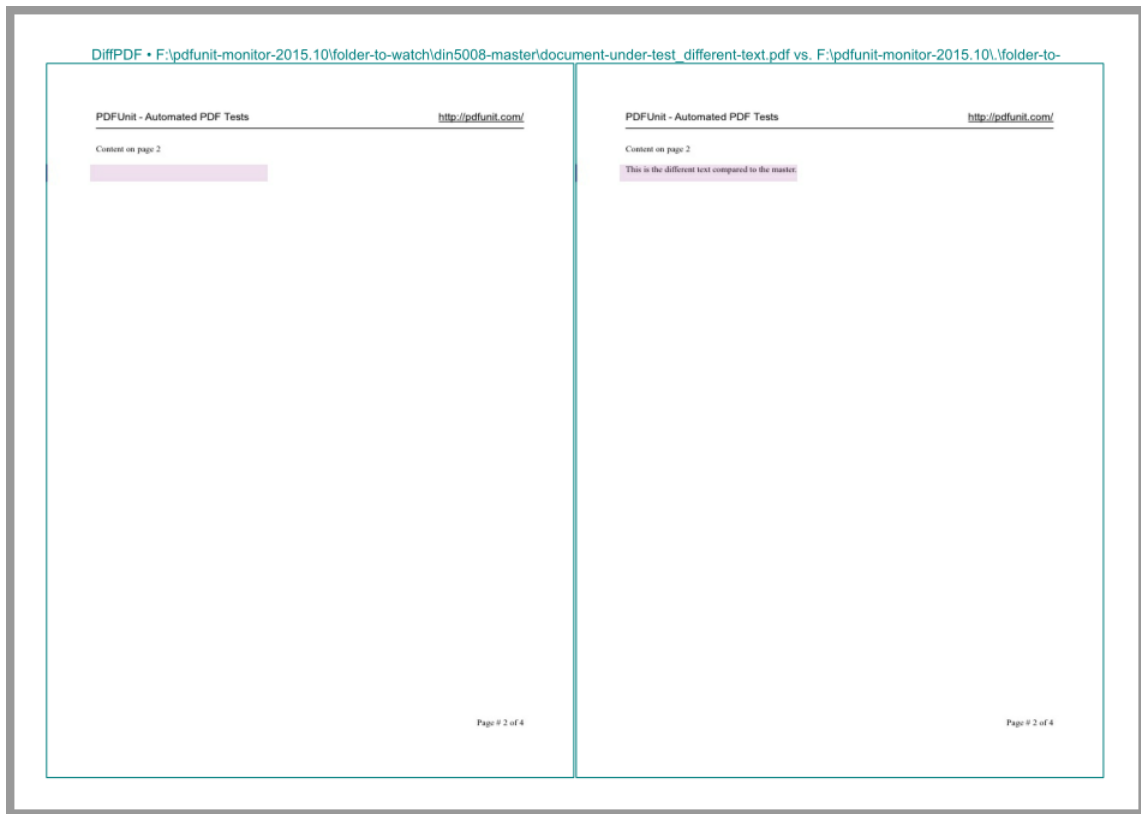
Wenn ein Vergleichs-PDF existiert, zeigt sich nach dem Start folgendes Bild:



Der große Hauptbereich zeigt beide Dokumente seitenweise nebeneinander, links das Referenz-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 'DiffPDF' kann auch unabhängig vom PDFUnit-Monitor genutzt werden.

Unabhängig von der Anwendung 'DiffPDF' erstellt der PDFUnit-Monitor 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 Bilddateien kann über die Property 'output.path.diffimages' in der Datei `PDFUNIT_MONITOR_HOME/lib/PDFUnit-<version>/pdfunit.config` geändert werden.

3.5. 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 in der Ergebnisliste angezeigt werden, 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. Validierungsergebnisse können über die Möglichkeiten von Log4j2 auch in eine Datei geschrieben oder an andere Logging-Monitore geschickt werden. Über die Konfigurationsmöglichkeiten von Log4j2 informiert Kapitel [9: „Logging mit Log4j2“ \(S. 34\)](#).

Kapitel 4. Validierungsregeln in Excel-Dateien

Regeln für eine Validierung von PDF-Dokumenten können in Excel-Dateien abgelegt werden. Deren Struktur sowie die an der Struktur hängende Funktionalität wird in den folgenden Abschnitten beschrieben.

Aufbau einer Excel-Datei

Eine Excel-Datei wird nach Tabellenblättern (Sheets) mit festgelegten Namen durchsucht:

Excel-Sheet	Bedeutung
regions	Definition von Seitenausschnitten
check	Definition von Testfällen für einzelne PDF-Dokumente
compare	Definition von Testfällen für den 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 hinter den festgelegten Spalten sind erlaubt. Zusätzliche Tabellenblätter in der Excel-Datei stören ebenfalls nicht.

Ein Tabellenblatt darf Leerzeilen enthalten. Falls ein Blatt aber zu viele leere Zeilen hat, kann es passieren, dass die letzten Zeilen mit Daten nicht mehr ausgelesen werden. Deshalb sollten leere Zeilen nicht verwendet werden. Setzen Sie in Spalte 1 einen Stern, dann gilt die Zeile als nicht-leer und bereitet keine technischen Probleme.

Definition von Seitenausschnitten

Zusätzlich zu der Angabe von Seiten, ist es fast immer 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. Deshalb erwartet PDFUnit, dass jeder Test in Excel einen Seitenausschnitt referenziert. Alle Seitenausschnitte werden in dem Sheet 'regions' definiert.

Ein Seitenausschnitt wird durch 4 Werte beschrieben: die x/y-Koordinaten der linken oberen Ecke und die Breite und Höhe des Ausschnitts. Alle Werte sind Millimeter-Angaben. Die Werte dürfen zwar Nachkommastellen haben, jedoch werden die Nachkommastellen auf die nächste Ganzzahl gerundet. Das nächste Bild zeigt ein Beispiele für Ausschnitte:

	A	B	C	D	E	F
1	*	Constraint file used for PDFUnit selftests.				
2	*	Definition of regions according to DIN5008, Form-A				
3	*					
4	*	id	x	y	width	height
5		address region	25	32	80	40
6		info block	125	32	75	40
7		header	25	0	185	27
8		text body page 1	25	80	165	165
9		text body page 2ff	25	32	165	212
10		page number region	25	245	165	12
11		footer	25	260	165	30
12		left margin	0	0	25	297
13		right margin	190	70	20	227
14		spacing below header	0	72	210	9

Wie zu erkennen ist, enthält das Tabellenblatt neben den 4 Werten für die Ausschnitte 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 zur Erfassung aller Informationen, die für Testfälle benötigt werden, die sich auf Einzeldokumente beziehen, also nicht für Testfälle, bei denen zwei Dokumente miteinander verglichen 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	Art der Überprüfung. Die erlaubten Werte sind weiter unten beschrieben.
expected 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 ebenfalls weiter unten beschrieben.

	A	B	C	D	E	F	G	H
1	*	Constraint file used for PDFUnit selftests.						
2	*	This sheet declares constraints to validate PDF documents according to rules defined by DIN 5008, Form-A.						
3	*							
4	*	id	pages	region	constraint	expected value	whitespace	message
5		address region page 1	1	address region	must contain	Anschrift	ignore	Text is missing. ID: {id}, region: {region}.
6		info block page 1	1	info block	must contain	Infoblock	normalize	Text is missing. ID: {id}, region: {region}.
7		header all pages	all	header	must contain	Automated PDF Tests	keep	Text is missing. ID: {id}, region: {region}.
8		text body page 1	1	text body page 1	must contain	Lorem ipsum		Text is missing. ID: {id}, region: {region}.
9	*	Textbereich Seite 2ff_ID	2..	text body page 2ff	must contain	Lorem ipsum		Text is missing. ID: {id}, region: {region}.
10		page number region all pages	all	page number region	must match	*Seite id von id.*		The page number is missing. ID: {id}, region: {region}.
11		footer all pages	all	footer	must contain	Firmenangaben		Text is missing. ID: {id}, region: {region}.
12	*							
13		left margin all pages	all	left margin	must be empty			Region is not empty. ID: {id}, region: {region}.
14		right margin pages 2ff	2..	right margin	must be empty			Region is not empty. ID: {id}, region: {region}.
15		spacing below header page 1	1	spacing below header	must be empty			Region is not empty. ID: {id}, region: {region}.
16	*							
17		empty pages pages 1, 2, 3	...3	text body page 2ff	must not be empty			Region is empty. ID: {id}, region: {region}.
18		empty pages pages 4, 5	4..5	text body page 2ff	must be empty			Region should not contain text. ID: {id}, region: {region}.

Benennung der Seiten, auf die sich ein Test bezieht

Ein Testfall bezieht sich häufig auf bestimmte Seiten eines Dokumentes. Deshalb können die gewünschten Seiten spezifiziert werden. 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', verschiedene Arten von Textvergleichen

Die Spalte 'constraint' dient dazu, festzulegen, welche Prüfungen durchgeführt werden sollen, ob beispielsweise 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 mögliche Werte der Spalte 'constraint' im Tabellenblatt 'check' für Textvergleiche:

Schlüsselwort	Verhalten
'must contain'	Der Text, der in der Spalte 'expected 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 'expected 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 'expected 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 'expected value' darf für den Text innerhalb des angegebenen Bereichs keinen Treffer ergeben.

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

Im Tabellenblatt 'compare' sind in der Spalte 'constraint' andere Werte erlaubt. Sie werden weiter unten erläutert.

Spalte 'constraint', Signaturen und Bilder testen

Weiterhin kann die Spalte 'constraint' im Tabellenblatt 'check' auch Schlüsselwörter für Tests auf Unterschriften und Bilder enthalten:

Schlüsselwort	Verhalten
'is signed'	Ein PDF-Dokument muss unterschrieben sein.

Schlüsselwort	Verhalten
'is signed by'	Ein PDF-Dokument muss von der Person unterschrieben sein, deren Name in der Spalte 'expected value' steht.
'has number of images'	Auf den ausgewählten Seiten und Seitenausschnitten wird die Anzahl der sichtbaren Bilder überprüft. Die erwartete Anzahl muss in der Spalte 'expected value' angegeben werden.

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. Mehrfache Whitespaces innerhalb eines Textes werden auf ein Leerzeichen reduziert.

Fehlerhafte Whitespace-Werte führen zu einer Fehlermeldung. Die Spalte 'whitespace' darf aber leer gelassen werden. In dem Falle gilt 'normalize' als Voreinstellung.

Beim Vergleich zweier PDF-Dokumente spielen Whitespaces nicht immer eine Rolle, beispielsweise beim Vergleich von Lesezeichen (Bookmarks). In den Fälle wird eine vorhandene Angabe der Whitespace-Behandlung ignoriert.

Spalte 'expected value', erwarteter Text

Wenn eine Prüfung gegen einen Erwartungswert stattfindet, muss es eine Spalte geben, die diesen Wert aufnimmt. Diese Spalte heißt 'expected value'.

Der Inhalt der Spalte 'expected value' wird als Regulärer Ausdruck interpretiert, wenn die Spalte 'constraint' die Werte 'must match' oder 'must not match' enthält. Informationen über reguläre Ausdrücke gibt es im Internet unter anderem bei [selfhtml](#).

Der Inhalt der Spalte 'expected value' wird in eine Ganzzahl umgewandelt, wenn die Spalte 'constraint' den Text 'has number of images' enthält.

Spalte 'message', Fehlermeldungen mit Platzhaltern

In den Excel-Dateien können auch individuelle Fehlermeldung hinterlegt werden, die im Fehlerfalle zusätzlich zu den Standardmeldungen von PDFUnit ausgegeben werden. Eine solche Fehlermeldung kann Platzhalter für Laufzeitinformationen enthalten. Das folgende Bild zeigt mehrere Beispiele:

message
Text is missing. ID: {id}, region: {region}.
Text is missing. ID: {id}, region: {region}.
Text is missing. ID: {id}, region: {region}.
Text is missing. ID: {id}, region: {region}.
Text is missing. ID: {id}, region: {region}.
The page number is missing. ID: {id}, region: {region}.
Text is missing. ID: {id}, region: {region}.

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

Platzhalter, Schlüsselwort Bedeutung

{id}	Die ID des aktuellen Testfalls
{pages}	Seitenzahl der Seite, auf der ein Fehler erkannt wurde
{region}	Der Wert der Spalte 'region'
{constraint}	Der Wert der Spalte 'constraint'

Die Platzhalter können an beliebiger Stelle im Text eingebaut werden. Die Werte des aktuellen Tests für die Platzhalter werden zur Laufzeit in Hochkommata eingefasst. Insofern müssen solche Hochkommata im Meldungstext nicht berücksichtigt werden.

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

Mit Excel-Validierungsregeln können auch vergleichende Tests durchgeführt werden. Dabei wird das aktuelle PDF-Dokument gegen eine Vorlage verglichen.

Vergleichendes Testen hat zur Folge, dass in der Excel-Datei keine Angabe mehr über einen erwarteten Text gemacht werden muss. Somit entfällt im Tabellenblatt 'compare' die Spalte 'expected value'. Es verbleiben diese:

	A	B	C	D	E	F	G
1	*	Constraint file used for PDFUnit selftests.					
2	*	Definition of test cases.					
3	*						
4	*	id	pages	region	constraint	whitespace	message
5		info block page 1	1	info block	same text	normalize	Different text found. ID: {id}, region: {region}.
6		header as text	all	header	same text	keep	Different text found. ID: {id}, region: {region}.
7		header as image	all	header	same appearance		Different appearance between test PDF and the reference PDF. ID: {id}, region: {region}.
8		pages 1, 3, 4 as text	1, 3, 4	body pages 2ff	same text	normalize	Different text found. ID: {id}, region: {region}.
9		pages 1, 3, 4 as image	1, 3, 4	body pages 2ff	same appearance		Different appearance between test PDF and the reference PDF. ID: {id}, region: {region}.
10		footer from page 2	2...	footer	same text	normalize	Different text found. ID: {id}, region: {region}.

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

Schlüsselwort

Verhalten

'same text'	Zwei PDF-Dokumente müssen im angegebenen Bereich den gleichen Text enthalten. Zusätzlich muss noch die interne Behandlung von Leerzeichen vorgegeben werden.
'same appearance'	Zwei PDF-Dokumente müssen im angegebenen Bereich als gerendertes Bild identisch sein.
'same bookmarks'	Zwei PDF-Dokumente müssen die gleichen Lesezeichen haben. Aus technischen Gründen darf der Wert in der Spalte 'region' nicht leer sein. Da sich Lesezeichen aber nicht auf einen Seitenausschnitt beziehen, muss die Spalte 'region' den Wert 'NO_REGION' enthalten.

Ein Test, Lesezeichen eines PDF-Dokumentes mit einer Vorlage zu vergleichen, sieht folgendermaßen aus:

	A	B	C	D	E	F	G
1	*	Constraint file used for PDFUnit selftests.					
2	*	Comparing bookmarks.					
3	*						
4	*	id	pages	region	constraint	whitespace	message
5	*						
6		bookmarks	all	NO_REGION	same bookmarks		Different bookmarks found. ID: {id}.

PDFUnit sucht die PDF-Dokumente für einen Vergleichs im Unterverzeichnis mit dem Namen 'reference' direkt unterhalb des Verzeichnisses der gerade im Test befindlichen PDF-Datei. Dort wird eine Datei mit dem gleichen Namen, wie die aktuelle Test-PDF-Datei geladen.

Fehlermeldungen zur Laufzeit

Eine Validierung eines PDF-Dokumentes gegen einer Excel-Datei endet nicht mit dem ersten erkannten Fehler. Alle Regeln werden durchlaufen. Für jede Regelverletzung wird eine Fehlermeldung erzeugt.

Kapitel 5. Installation, Update, Deinstallation

5.1. Technische Voraussetzungen

Der PDFUnit-Monitor benötigt als Laufzeitumgebung Java ab Version 7 sowie PDFUnit-Java mit seinen mitgelieferten Bibliotheken.

Getestete Umgebungen

Mit den folgenden Systemen wurde PDFUnit erfolgreich getestet:

Betriebssystem	Java Version
• Windows-7, 32 + 64 Bit	• Oracle JDK-1.7, 32 + 64 Bit
• Kubuntu Linux 12/04, 32 + 64 Bit	• Oracle JDK-1.8, Windows, 32 + 64 Bit
• Mac OS X, 64 Bit	• IBM J9, R26_Java726_SR4, Windows 7, 64 Bit

Weitere Java/Betriebssystem-Kombinationen werden ständig getestet.

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

5.2. Installation

Download und Entpacken von PDFUnit-Monitor

Laden Sie die Datei `pdfunit-monitor-<VERSION>.zip` aus dem Internet: .

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

Startskript und Classpath

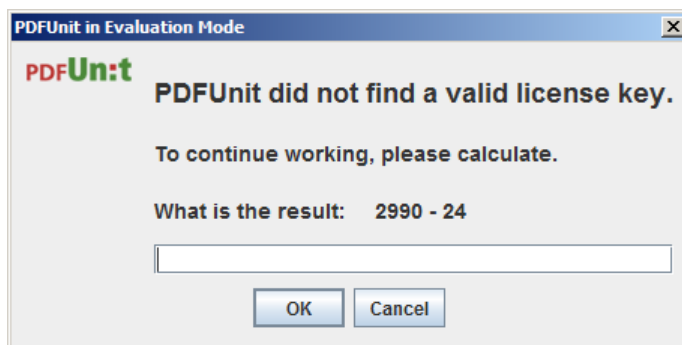
Im Verzeichnis `PDFUNIT_MONITOR_HOME` existiert ein Skript `startPDFUnitMonitor.bat`, um den PDFUnit-Monitor zu starten. Für Unix existiert das Startskript `startPDFUnitMonitor.sh`.

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. Schreiben Sie an [info\[at\]pdfunit.com](mailto:info[at]pdfunit.com) und fragen Sie nach einer Testlizenz oder einer Produktionslizenz.

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 PDFUnit-Java sowie eine separate Datei `license-key_pdfunit-java.lic`. Diese 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 ein Skript zur Überprüfung der Installation: `verifyInstallation.bat` oder `verifyInstallation.sh`. Es ist in Kapitel [5.4: „Überprüfung der Konfiguration“ \(S. 26\)](#) ausführlich beschrieben.

5.3. 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).

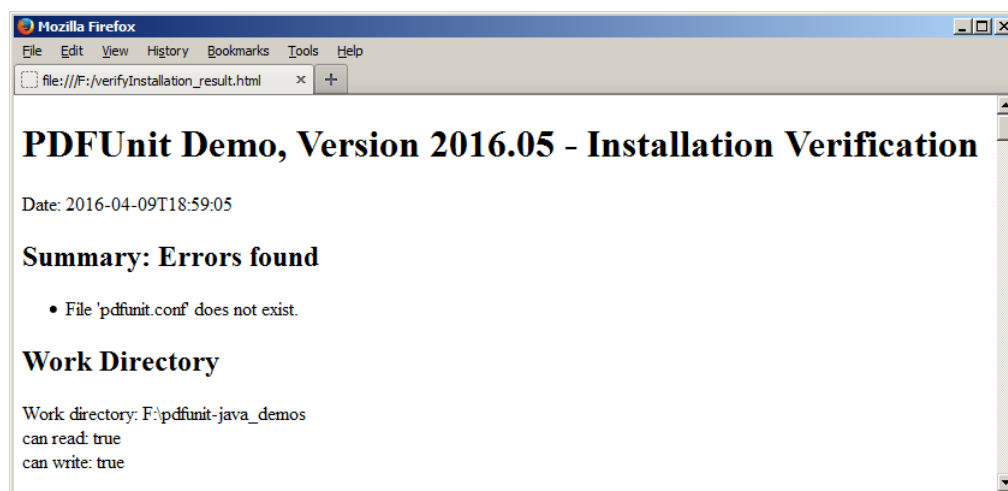
5.4. Überprüfung der Konfiguration

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:



5.5. 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: .

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 aktuelle 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` in den Classpath des neuen Releases.

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.

5.6. Deinstallation

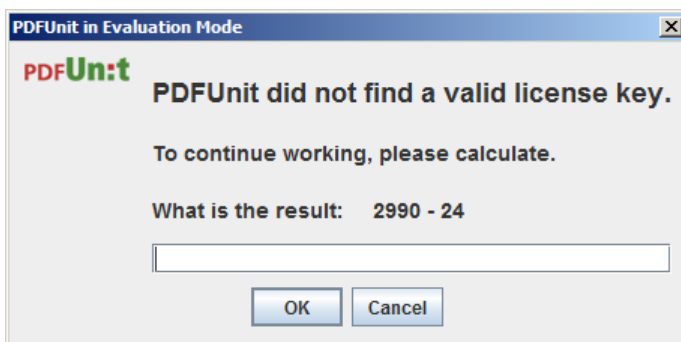
Analog zur Installation „per Copy“ wird PDFUnit durch das Löschen der Installationsverzeichnisse wieder sauber deinstalliert. Einträge in Systemverzeichnisse oder in die Registry können nicht zurückbleiben, weil solche nie erstellt wurden. Vergessen Sie nicht, in Ihren eigenen Skripten die Referenzen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

Kapitel 6. Starten der Anwendung

Im Basisverzeichnis der Installation von PDFUnit-Monitor befinden sich die Startskripte `startPDFUnit-Monitor.bat` bzw. `startPDFUnitMonitor.sh`. Sofern die JAR-Dateien alle dort liegen, wo das Skript sie erwartet, startet der PDFUnit-Monitor.

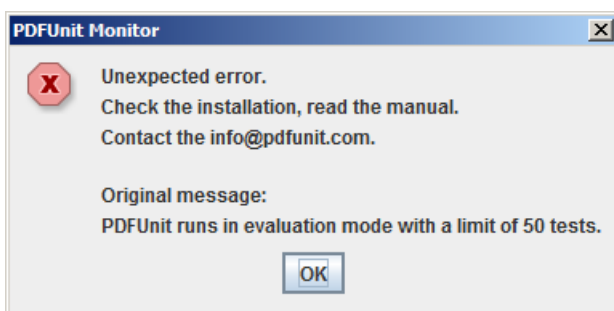
Nach dem Start im Evaluation-Modus

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. Wird die Rechenaufgabe falsch beantwortet, erscheint eine Meldung über die falsche Eingabe und der PDFUnit-Monitor erneut gestartet werden. Bedenken Sie, dass diese Einschränkungen der Handhabung nur im Evaluation-Modus bestehen. Mit einer Lizenz gibt es kein Limit und keine Rechenaufgabe.

Wenn das Limit überschritten wird, erscheint folgende Meldung:



Wenn das Limit erreicht wurde, muss der PDFUnit-Monitor neu gestartet werden. Die Zählung beginnt dann von vorne.

Kapitel 7. Konfiguration (pdfunit-monitor.config)

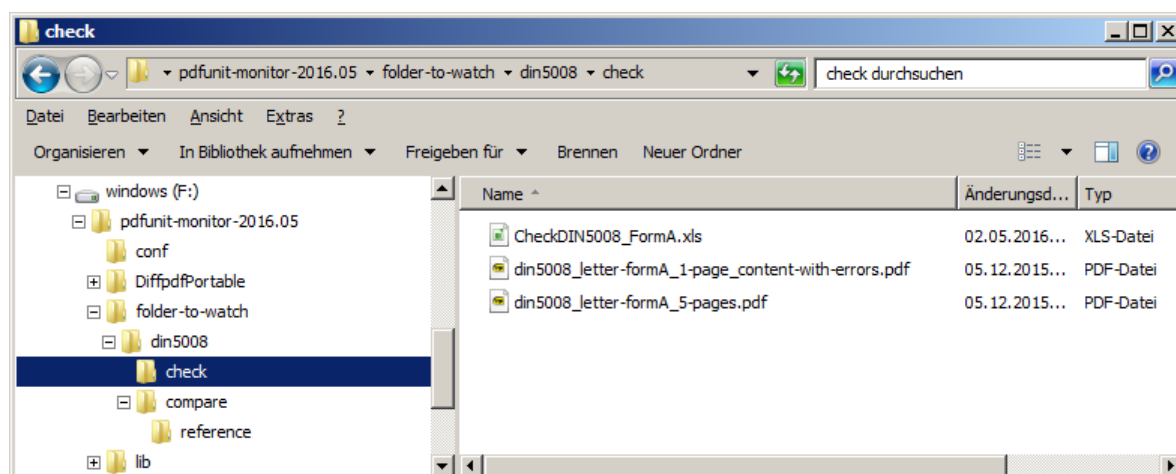
Überwachte Verzeichnisse

In der Datei `pdfunit-monitor.config` muss der Name des Wurzelverzeichnisses eingetragen werden, das vom PDFUnit-Monitor mit all seinen Unterverzeichnissen überwacht werden sollen. `WATCHED_FOLDER` ist das oberste Verzeichnis, auf das sich der PDFUnit-Monitor positioniert. Von hier aus werden alle Unterverzeichnisse überwacht.

Referenzdokumente für vergleichende Tests sucht der PDFUnit-Monitor immer in einem Unterverzeichnis `reference/` unterhalb des Verzeichnis, in dem sich das zu testende PDF-Dokument befindet. Das Referenzdokument muss den gleichen Dateinamen haben, wie das zu testende PDF-Dokument.

Nachfolgend Beispiele für die Konfiguration und die Verzeichnisstruktur:

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



Logging-Port

Der PDFUnit-Monitor benutzt intern Log4j2 als Logging-Framework und verwendet dazu in der Standardeinstellung Port 4445. Falls dieser Port belegt ist, muss ein anderer Port in den Dateien `pdfunit-monitor.config` und `log4j2.xml` festgelegt werden.

```
#
# 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
```

Für die Konfiguration des im Hintergrund laufenden Loggings gibt es noch die Konfigurationsdatei `log4j2.xml`. Weitere Informationen dazu stehen in Kapitel 9: „Logging mit Log4j2“ (S. 34).

Look and Feel

Das Aussehen der Anwendung kann angepasst werden. Im Auslieferungszustand hat sie den typischen Java-Look, über einen kleinen Eintrag in der Datei `pdfunit-monitor.config` kann aber auch das Aussehen des aktuellen Betriebssystems erreicht werden:

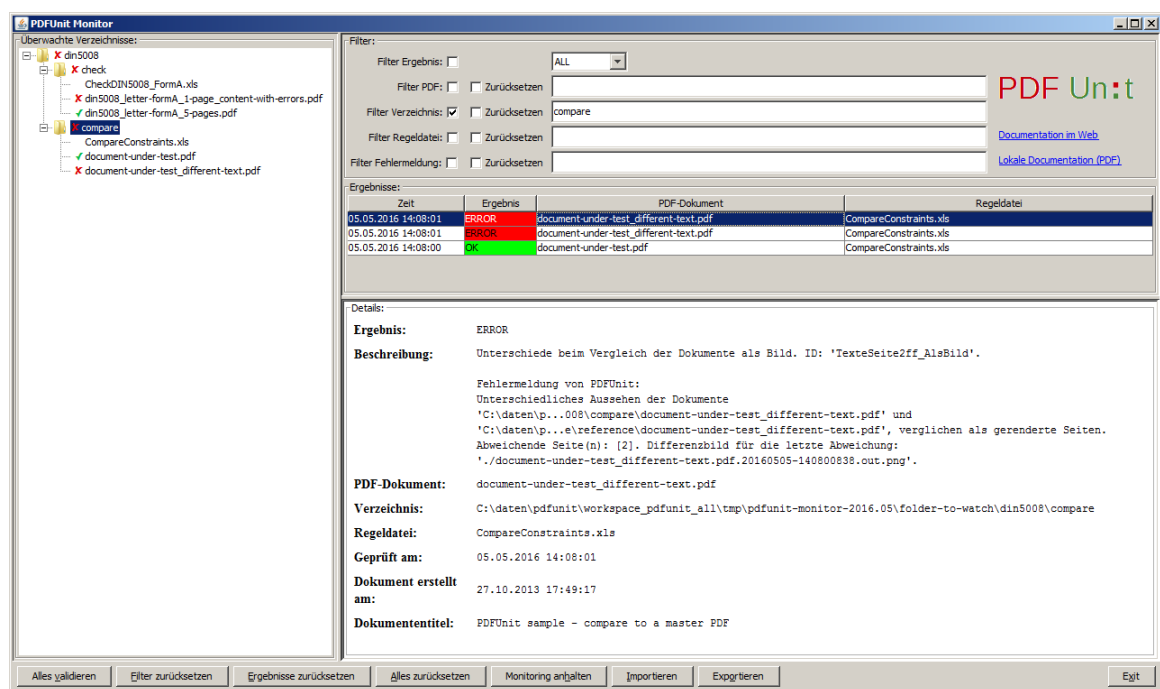
```
#
# Uncomment the next line to set the general look and feel of the operating system
#LOOK_AND_FEEL = System default
```

Sprache umstellen

Der PDFUnit-Monitor wird in den Sprachen Deutsch und Englisch ausgeliefert. Diese Mehrsprachigkeit bezieht sich sowohl auf die Oberfläche, als auch auf Fehlermeldungen. Ohne zusätzliche Konfiguration wird die Sprache des Betriebssystems verwendet. Damit der Monitor aber beispielsweise auf einem deutschen Betriebssystem in englischer Sprache läuft, müssen vor dem Start folgende Java-Systemproperties gesetzt werden:

```
-Duser.language=en -Duser.country=UK
```

Das folgende Bild zeigt den PDFUnit-Monitor in deutscher Sprache und im Aussehen von Windows-7:



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 'DiffPDF', die in einem Verzeichnis unterhalb von PDFUNIT_MONITOR_HOME mitgeliefert wird. Sollten Sie diese Anwendung in ein anderes Verzeichnis legen wollen, müssen Sie die nachfolgenden Parameter entsprechend anpassen. Wichtig: der Verzeichnisname muss mit einem 'forward slash' abgeschlossen werden.

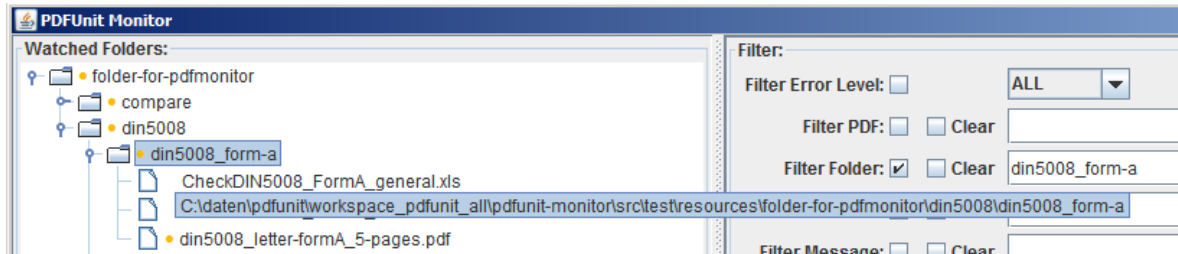
```
#
# Installation definitions for DiffPDF, the free Viewer.
# (Don't forget the trailing slash for the folder.)
# Information about DiffPDF are here: http://www.qtrac.eu/diffpdf.html
#
DIFFPDF_FOLDER = DiffpdfPortable/
DIFFPDF_EXECUTABLE = DiffpdfPortable.exe
```

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 Schlüssel `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:

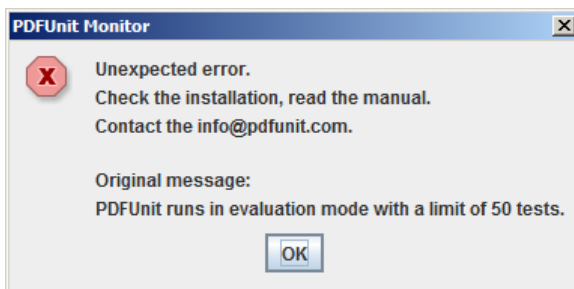


Kapitel 8. Typische Fehler

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.

Evaluation-Modus von PDFUnit-Monitor - Limit erreicht

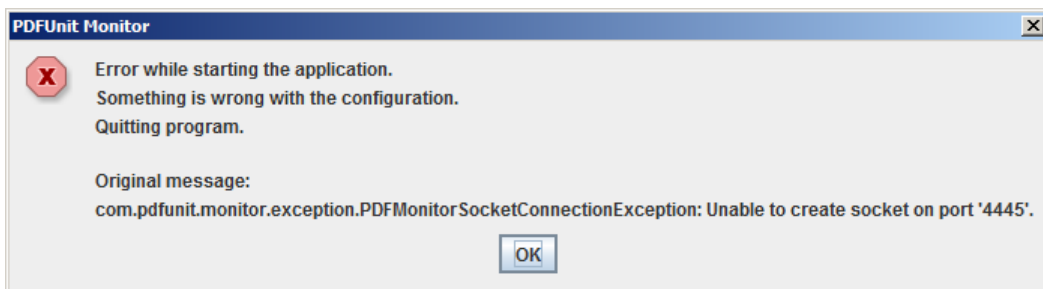
Im Evaluierungs-Modus können mit dem PDFUnit-Monitor maximal 50 Testfälle durchgeführt werden. Wird diese Zahl überschritten, erscheint folgende Meldung:



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 [info\[at\]pdfunit.com](mailto:info[at]pdfunit.com).

Anwendung zweimal gestartet - Logging-Port belegt

Der interne Logging-Appender 'PDFUnitMonitorAppender' (siehe log4j2.xml) schickt seine Prüfergebnisse über Sockets an die Oberfläche. Dafür wird Port 4445 verwendet. Falls die Anwendung ein zweites Mal gestartet wird, ist dieser Port schon belegt und es erscheint folgende Fehlermeldung:



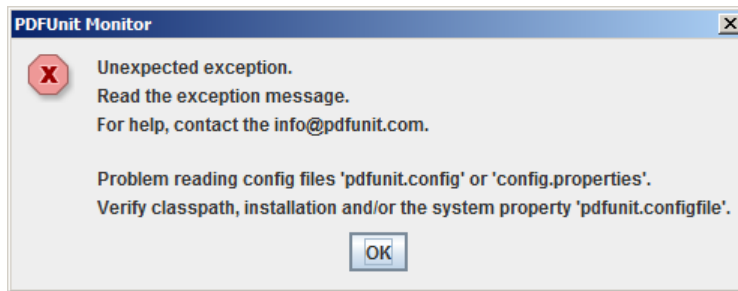
Wenn dieser Port schon verwendet wird, muss er für den Monitor in den Dateien `pdfunit-monitor.config` und `log4j2.xml` auf einen anderen Wert gesetzt werden.

Anzeige im Verzeichnisbaum unvollständig

Gelegentlich werden die Dateinamen in der Verzeichnisstruktur nicht vollständig dargestellt. In solchen oder ähnlichen Fällen kann der Button 'Reset All' gedrückt werden. Dann werden alle Verzeichnisse frisch eingelesen. Die bis dahin erhaltenen Prüfergebnisse werden dabei verworfen. Falls die Prüfergebnisse erhalten bleiben sollen, müssen sie zuvor exportiert werden.

Konfigurationsdatei `pdfunit.config` nicht im Classpath

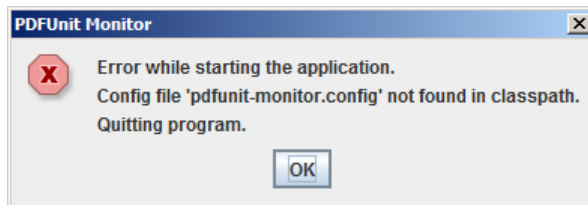
Aus welchem Grund auch immer eine Datei aus dem Installationsverzeichnis verschwindet - der PDFUnit-Monitor reagiert auf das Fehlen mit einer entsprechenden Meldung:



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.

Konfigurationsdatei `pdfunit-monitor.config` nicht im Classpath

Auch die Konfigurationsdatei `pdfunit-monitor.config` könnte aus dem Classpath verschwinden. Dann erscheint beim Starten folgende Meldung:



Kapitel 9. 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](#).

Aktuelle Konfiguration

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

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 `PDFUNIT_MONITOR_HOME/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>
```

In der Log-Konfiguration kann ein Error-Level als Filter angegeben werden. Mit der Auslieferung steht er für den Logger `PDFConstraints` auf `all`. Es ist nicht nötig, die 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 den Konfigurationsdateien `pdfunit-monitor.config` und `log4j2.xml` auf einen anderen Wert gesetzt werden.

Informationen zu Log4j2

Weitere Informationen zu Log4j2 entnehmen Sie bitte den Online-Dokumentationen:

- 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, 28

C

Classpath, 24

D

Darstellung

 Fehlerdetails, 15

 Fehlerliste, 12

 überwachte Verzeichnisse, 10

Deinstallation, 27

DiffPDF, 8, 15

 Export, 16

 Konfiguration, 30

E

Error-Level, 13

Erwarteter Text, 21

Evaluationsversion, 24

Excel-Datei

 Fehlermeldungen, 23

 Sheets, 18

Excel-Sheet

 check, 19

 compare, 22

 region, 18

Export, 17

F

Feedback, 5

Fehlerdetails, 15

Fehlerliste

 Error-Level, 13

 Filter, 12

 sortieren, 14

Fehlermeldungen in Excel, 21

Filter, 12

Funktionalität, 10

I

Import, 17

Installation, 24, 24

 Classpath, 24

 Lizenzschlüssel, 25

 neues Release, 26

 überprüfen, 26

K

Konfiguration, 29

 DiffPDF, 30

 Logging-Port, 29

 Look and Feel, 29

 Sprache, 30

 Tooltip, 30

 überwachte Verzeichnisse, 29

L

Leerzeichen, 21

Lizenzschlüssel

 beantragen, 25

 installieren, 25

Log4j2

 Logging, 34

 Port belegt, 32, 34

Logging-Port, 29

Look and Feel, 29

P

PDFUnit-Monitor

 Export, 9

 Fehlerdetails, 7

 Filter, 7

 Import, 9

 Vergleich gegen Vorlage, 8

Q

Quickstart, 6

S

Seitenbereiche, 18

Sprache einstellen, 30

Startskript, 24

Systemumgebungsvariablen, 25

T

Technische Voraussetzungen, 24

Testfall

 erwarteter Text, 21

 Fehlermeldungen, 21

 Leerzeichen, 21

Typische Fehler, 32

 Anzeige unvollständig, 32

 Limit erreicht, 32

 pdfunit.config nicht gefunden, 32

 pdfunit-monitor.config nicht gefunden, 33

U

Überwachte Verzeichnisse, 10

Umgebungsvariablen, 25

Update, 26

W

Whitespaces, 21