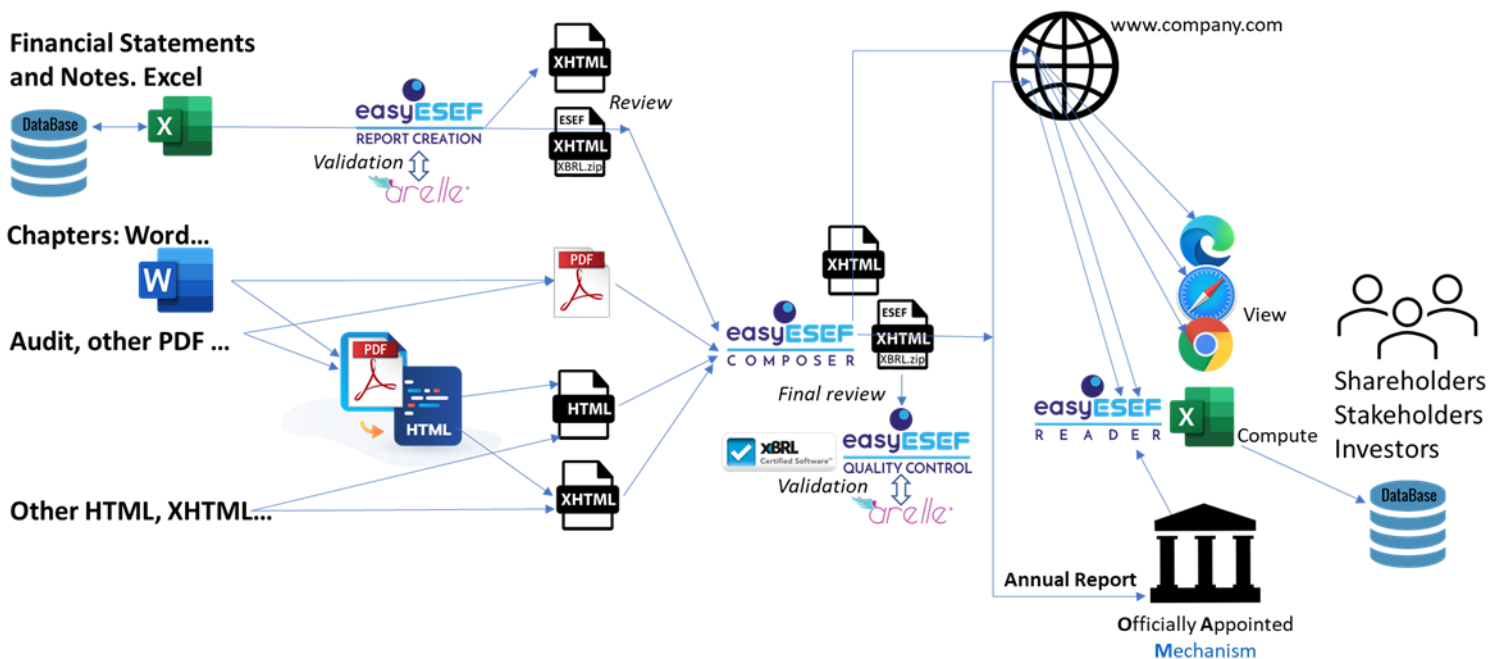


easyESEF COMPOSER

Technische Referenz

easyCOMPOSER: Veröffentlichen Sie den ESEF-Jahresbericht

easyCOMPOSER ist ein Tool zum Exportieren einer Datei mit dem Jahresbericht im ESEF-Format. Diese Akte ist an den *amtlich bestellten Mechanismus* zu senden und auf der Website des Unternehmens zu veröffentlichen. Die Financial Statements und Notes, die bereits im ESEF-Format als .zip-Datei getaggt wurden, sowie die Chapters und andere Komponenten im Format .pdf .html .xhtml ebenso.



easyCOMPOSER stellt den Jahresbericht im ESEF-Format als Datei zusammen, wobei der Geschäftsbericht einsehbar ist. xhtml und Dateien mit obligatorischen Meta-Informationen (XBRL-Linkbases), alle komprimiert nach dem standard *taxonomypackage.zip*

easyCOMPOSER exportiert Geschäftsbericht .zip im angegebenen Verzeichnis. Außerdem exportiert sie eine Kopie des Jahresberichts in .xhtml für die Bequemlichkeit der Verwendung.

easyCOMPOSER geht davon aus, dass die importierten Dateien korrekt sind. Es wird empfohlen, eine abschließende Überprüfung mit dem von XBRL International zertifizierten

easyQC Qualitätskontrolltool durchzuführen. Dieses Tool verwendet intern den Arelle Validator, der ebenfalls von XBRL International zertifiziert ist.

easyCOMPOSER fasst Jahresabschluss und Anhang zusammen.xhtml-Datei (die sich in der .zip-Datei befindet) mit den anderen .pdf oder .html oder .xhtml-Dateien in der angegebenen Reihenfolge. Sie können eine Datei höchstens .zip ablegen. Es funktioniert auch, wenn Sie nur .pdf oder .html oder .xhtml Dateien ablegen und die .zip-Datei weglassen.

easyCOMPOSER konvertiert Dateien im .pdf-Format und .html-Format intern in .xhtml-Format, wie es das ESEF-Format erfordert. Die Darstellung variiert nicht. Details finden Sie unter Chapter *Conversion .html to .xhtml*

easyCOMPOSER erkennt intern, ob eine .xhtml-Datei kann die Darstellung einer anderen Datei beeinflussen. Zum Beispiel, weil es einen Stil (<style>) definiert hat, der für alle Strukturen gilt.xhtml eines bestimmten Typs (z. B. <table> oder <td>). Oder weil mehrere Dateien .xhtml verwendet Stile (<style>) mit demselben Namen und unterschiedlichen Inhalten. Siehe Details in Chapter *Disambiguation of .xhtml Styles*

easyCOMPOSER exportiert die angepassten .pdf- und .html- und .xhtml-Dateien zur weiteren Kontrolle in das ausgewählte Verzeichnis.

easyCOMPOSER kombiniert die .xhtml-Quelldateien, die drei Teile unterscheiden: <style> <body> und "*other.xhtml*". Die Kombination <style> und <body> ist die unveränderte Verkettung jedes <Stils> und jedes <Körpers> jeder Datei.xhtml-Quelle zum Kombinieren. Es kann mit jedem Texteditor überprüft werden, ob sich die <style> und <body> der kombinierten Zieldatei nicht geändert hat, was genau der Reihenfolge der <style> und der <body> jeder kombinierten Quelldatei entspricht. Auf der anderen Seite folgt die Kombination von "*anderen.xhtml*" ihren spezifischen Regeln. Details finden Sie unter chapter *Combine .xhtml-Dateien*

Installationsanforderungen: Microsoft Excel, unter Windows- oder Mac-Versionen.

Composer-Blatt

Folder with files: Die Zelle unten gibt den *Pfad* eines Standardverzeichnisses an, in dem:

- Es gibt die zu importierenden Dateien, deren *Pfad* nicht angegeben ist
- Exportierte Dateien werden platziert
- Die notwendigen Arbeitsdateien werden erstellt und gelöscht.

Ordered import files (.zip .xhtml .html) to compose: In den Zellen unten werden die Dateien in der richtigen Reihenfolge angeordnet, .zip .html und .xhtml, die kombiniert werden sollen. Wenn der Pfad einer Datei nicht angegeben ist, wird das Standardverzeichnis verwendet.

Zum  Ausführen hier klicken

Composed file exported: Die Datei .xhtml und (falls zutreffend) der .zip zusammengesetzte Jahresbericht werden angezeigt. Beide Dateien befinden sich im Standardverzeichnis.

Files with .xhtml adjusts exported: Für jede Datei .xhtml, die eine Anpassung benötigt hat, wird diese Datei angezeigt, die sich im Standardverzeichnis befindet.

Converter from .pdf to .html: Absoluter Pfad oder relativ zum Standardverzeichnis zum *Konverter.exe* .pdf zu .html mit den entsprechenden Parametern (z.B. Zoom 1.1).

```
pdf2htmlEX-win32-0.14.6-upx-mit-poppler-daten-pdf2htmlEX.exe --zoom 1.1
```

Konverter verwende: soft.rubypdf.com/software/pdf2htmlEX-windows-version

Es ist Open Source, kostenloser Download, und in seinen verschiedenen Versionen das am weitesten verbreitete in ESEF.

Als kommerzieller Konverter wurde das [PDF2HTML Reflow Paragraph Module](#) mit guten Ergebnissen getestet. Es verfügt nicht über eine Zoom-Option. Die Demoversion kann heruntergeladen werden und funktioniert einwandfrei, konvertiert aber nur ungerade Seiten.

ListOfFiles-Blatt

Folder with files: Die Zelle unten gibt den *Pfad* eines Verzeichnisses an, aus dem alle darin enthaltenen Dateien und Unterverzeichnisse angezeigt werden. Es ist eine nützliche Hilfe.

Zum  **Ausführen hier klicken**

List of files and folders: Zeigt die Liste der Dateien und Unterverzeichnisse an, die im angegebenen Verzeichnis enthalten sind.

Mehr zum Thema

Nähere Informationen und Fragen zu **easyCOMPOSER** finden Sie unter info@easyESEF.eu

Sehen Sie sich unsere anderen Tools für ESEF in easyesef.es/resources an. Zusätzlich zu den Produkten verfügen wir über einen hochspezialisierten Support für ESEF. Dieser Service ist herstellerunabhängig und gilt für jede ESEF-Generierungssoftware.

HÄUFIG GESTELLTE FRAGEN:

Copyright © bei easyESEF Ltd. Irland. Alle Rechte vorbehalten. Hinterlegt in der WIPO. Reverse Reengineering ist verboten. Durch die [obfuscat.org-Lösung](https://obfuscat.org) geschützter Code in einigen Versionen.

Das Ausführen von Excel und/oder das Herunterladen von Arelle kann Sicherheitsautorisierungen in Ihrer Entität erfordern.

ENDE DES HAUPTTEXTES DER TECHNISCHEN REFERENZ

Anhang: Konvertierung .html in .xhtml

Das ESEF-Format erfordert, dass sich die Datei in befindet.xhtml Es gibt jedoch Zeiten, in denen sich die Datei in .html befindet und eine Konvertierung erforderlich ist. Die Konvertierung kann mit <https://validator.w3.org/> überprüft werden

easyCOMPOSER konvertiert eine .html Datei in eine .xhtml-Datei in den folgenden Schritten:

- a) Übertragen Bestimmter Tags¹ und Sonderzeichen² von .html auf .xhtml
- b) Fügen Sie in einem Kommentar die Anweisungen von <![if ! IE]> Geben Sie <!--<![if ! IE]>-->
- c) Entfernen Sie cr LF TAB, doppelte Leerzeichen auf Beschriftungen, https://www.w3.org/TR/xhtml1/guidelines.html_C5
- d) Tag-Namen und -Attribute in Tags , die vollständig aus Großbuchstaben bestehen, werden in Kleinbuchstaben geändert, z. B. COLSPAN=3 in colspan=3
- e) Attributwerte in Tags werden immer in Anführungszeichen gesetzt, z. B. colspan=3 bis colspan="3"
- f) Die Tags <img... > und <meta... >, die nicht mit oder </ meta > geschlossen sind, werden immer als <img... /> und <meta... /> geschlossen.
- g) Das tag muss das Attribut alt="...." haben. definiert. Andernfalls alt="" einschließen.
- h) Tags zum automatischen Schließen sollten kurz vor dem letzten /> ein Leerzeichen belassen, z. B.

- i) Auf dem Label <html... > wenn der Namespace xmlns="http://www.w3.org/1999/xhtml" fehlt, fügen Sie ihn als <html hinzu.... xmlns="http://www.w3.org/1999/xhtml" ... >

¹
</Br>|
| <hr></hr>| <hr>|<Gemischt|</embed>| <style type=""text/css"">| | | <u>| </u>| <nobr>| </nobr>

|
|<hr />|<hr />| <img || <style>| | | | | |

² |"|&|'|<|>|"|&|'|<|>

Anhang: Disambiguation of .xhtml-Stile

easyCOMPOSER überprüft jeweils .xhtml-Datei in der Reihenfolge, in der sie im *Composer-Blatt* deklariert wurden, um festzustellen, ob Sie Disambiguierungsanpassungen benötigen. Wenn eine Datei .zip vorhanden ist, wird diese Datei zuerst überprüft.

Die Stile des <stils> Bereich der . Die xhtml-Datei wird angepasst, wenn sie erkannt wird:

- Ein Stil, der direkt auf Tags anwendbar ist.xhtml als table{} td{}, was direkt auf alle Tags <table> <td> in der anwendbar wäre. kombinierte xhtml.
- Ein Stilname, der bereits in einer vorherigen .xhtml-Datei, aber mit anderem Inhalt (z. B. mit einer anderen Buchstabenschriftart).

In jeder anzupassenden Datei:

- Für jede anzupassende Datei wird ein eindeutiges Suffix (z. B. "___N") vorbereitet.
- Es wird disambiguiert, indem jeder ursprüngliche Stilname mit dem eindeutigen Suffix verkettet wird , um einen eindeutigen Stilnamen zu erstellen. Dadurch wird die Möglichkeit doppelter Stilnamen ausgeschlossen. Beispiel: .ft0 #page_1 in .ft0 ___N #page_1___N geändert
- Für Formatvorlagen, die direkt auf Tags .xhtml wie table{} oder td{} anwendbar sind, wird am Anfang des Formatnamens ein Punkt "." hinzugefügt. Beispiele: table{} td{} in geändert. Tabelle___N {} . td___N {}
- In der Zone <body> ändern sich die ursprünglichen Stilnamen in die neuen eindeutigen Stilnamen Beispiel: <p class="ft0 "> <div id="page_1"> in <p class="ft0___N"> <div id="page_1___N">
- In der Zone <body>, in Tags mit Stilen, die direkt auf .xhtml-Tags anwendbar sind, wie table{} oder td{}, wird der neue eindeutige Stilname als class=""-Attribut hinzugefügt. Beispiele: <table > <td > in <table class="" geändert. table___N" > <td class="" . td___N" >

Anhang: Kombinieren .xhtml Dateien

easyCOMPOSER nimmt jede .xhtml-Quelldatei , die kompindiert werden soll, in der Reihenfolge, in der sie im Composer-Blatt deklariert wurden . Aus jeder Datei extrahiert es den Inhalt von `<style>PAYLOADSTYLE</style>` und `<body>PAYLOADBODY</body>` sowie bestimmte Informationen aus "other .xhtml"-Tags wie `<title>... </title>`.

Die Zieldatei .xhtml ist das Ergebnis der Verkettung dieser Inhalte in einem einzigen `<style>... </style>` und in einem einzigen `<body>... </body>` sowie das Komponieren von "other .xhtml"-Tags.

Aus jeder zu kombinierenden Quelldatei "n" werden drei Arten von Informationen extrahiert:

1. Inhalt von (es kann keine, einen oder mehrere geben)
`<style>PAYLOADSTYLEn</style>`
2. Gehalt an `<body> NUTZLASTKÖRPER</body>`
3. Inhalt von "sonstigen.xhtml", zum Beispiel:
 - a) Startzeile `<?xml version="1.0" encoding="UTF-8"?>`
 - b) Markieren Sie `<!DOCTYPE >` (ignoriert)
 - c) Zusätzliche Namespaces in `<html... >`
 - d) Tag `<title>... </title>`
 - e) Tags, die nicht anders als die vorherigen wiederholt werden, wobei ihre relative Position in der zu beachten ist.xhtml-Struktur

easyCOMPOSER exportiert die .xhtml kombiniert, mit einer Struktur .xhtml-Typ:

```
<? XML... >
<html.... >
  <Kopf>
    <title>... </title>. Die Datei .zip hat hier Priorität
    </> und andere nicht duplizierte Tags an der Position, die sie hatten
    <style>PAYLOADSTYLE1PAYLOADSTYLE2PAYLOADSTYLE3</style>
  </Kopf>
<body>PAYLOADBODY1PAYLOADBODY2PAYLOADBODY3</body>
</html>
```

Anhang: SHA256-Hashcodes in .html

Praktisches Problem.

Ein Unternehmen bereitet seinen geprüften Jahresbericht vor. Jedes Kapitel wird hypothetisch von einer anderen Abteilung vorbereitet. Der Rechnungsprüfer erhält die verschiedenen Kapitel so, wie sie verfügbar sind, um angesichts der üblichen Fristenzurücken parallel voranzukommen. Es ist zu überprüfen, ob der Geschäftsbericht ausschließlich und ausschließlich aus der Integrität der geprüften Kapitel besteht.

Im Falle des einheitlichen europäischen einheitlichen Electronic Format ESEF ist der Jahresbericht eine Datei, die in jedem Internetbrowser (Format .html), sowie eine Reihe von nicht einsehbaren Dateien, die den Jahresabschluss ergänzen (iXBRL-Standard).

Aufgrund der Homogenität werden die Kapitel einzeln als .html einsehbare Dateien an den Auditor gesendet. Am Ende werden die Kapitel kombiniert, wodurch die .html Jahresbericht.

SHA256 Hash-Code als digitaler Fingerabdruck

Ein Hashcode ist der digitale Fingerabdruck des Inhalts einer Datei. Es gibt mehrere Algorithmen, um den Hash-Code einer Datei zu erhalten. Der Hash-Code wird mit dem Standardreferenzalgorithmus in der [SHA256-Kryptographie](#) abgerufen, der einen unwiederholbaren digitalen 256-Bit-Fingerabdruck erzeugt³. Jede Änderung des Inhalts der Datei ändert den SHA256-Hashcode. Der SHA256-Algorithmus ist von starker Sicherheit, daher ist die Methode zum Erstellen einer Datei, deren erhaltener SHA256 genau einen definierten Wert hat, unbekannt. Auf diese Weise ist die Integrität gewährleistet: Wenn ein bekannter SHA256 aus der Datei erhalten wird, ist es so, dass die Datei keiner Änderung unterzogen wurde.

Der SHA256-Hashcode hängt nicht vom Betriebssystem oder einer bestimmten Software ab. Der SHA256-Hashcode hängt von der Codierung der Zeichen ab, die utf8 gemäß dem ESEF-Format sein wird. Das Abrufen des SHA256 einer Datei oder Zeichenfolge ist mit Programmbibliotheken und Online-Lösungen einfach.

Wenn Sie also die SHA256-Hashcodes einer Reihe von Dateien kennen, können Sie jederzeit überprüfen, ob diese Dateien keine Änderungen erfahren haben.

Durch die Verkettung mehrerer Quelldateien in einer einzigen Zieldatei wird der SHA256-Hashcode verketteter Dateien nicht geändert. Wenn Sie die Größe jeder zu verkettenden Quelldatei und die Verkettungssequenz kennen, können Sie jede Quelldatei der Verkettung einzeln extrahieren ("deconcatenate") und ihren SHA256-Hashcode neu berechnen.

³ Diese 256-Bit Gewöhnlich darstellen oder Nicht schlecht wie 64 Zeichen Hexadezimal (0-9:A-F, Basis 16) oder Nicht schlecht wie 44 Zeichen A-Z:a-z:0-9:+/ in der Basis 64 und mit einem = am Ende.

Zusammensetzung der Dateien.xhtmll

Eine .xhtml-Datei kann grundsätzlich als drei Arten von XML-Strukturen beschrieben werden :

- `<body>`. `</body>` der die darzustellenden Texte und Bilder enthält, und die Struktur, wie er sie darstellt (Absätze, Tabellen, Listen...)
- `<style>.... </style>` der die Stile enthält, die für `<body>` anwendbar sind.... `</body>`, wie Schriftarten und Schriftgrößen, bestimmte sich wiederholende Bilder, Farben, Ausrichtungen ...
- "*another .xhtml*" als `<title>... </title>`, Namespaces und andere Meta-Informationen.

Beim Kombinieren mehrerer .xhtml-Quelldateien in einer .xhtml-Zieldatei, zum einen der Bereich `<body>`. ... `</body>` der einzelnen .xhtml-Quelldatei wird kombiniert, um eine einzelne `<body>` zu erzeugen.... `</body>` in der .xhtml-Zieldatei, da es nur eine einzige Struktur `<body>` geben kann.... `</body>` in einer .xhtml-Datei

Eine ähnliche Kombination wird für die zone `<style>` hergestellt. Und bestimmte doppelte Teile jeder Quelldatei (wie die tags `<body></body>` selbst) verschwinden einfach als redundant.

Jeweils .xhtml wurde in mehrere Zonen zerlegt, die unabhängig voneinander in der .xhtml-Ziel. Sie können den SHA256-Hashcode von jedem nicht mehr neu berechnen.xhtml-Quelldatei mit Blick auf .xhtml-Zieldatei .

PROBLEM: Wie kann man die Integrität jedes .xhtml oder .html Quelldatei sicherstellen, wenn es in einer einzigen .xhtml Zieldatei kombiniert wird?

Kombination von Quelldateien.xhtmll mit `<style>` und `<body>` Invariante

Ein einfaches Verfahren zum Abrufen einer .xhtml-Zieldatei soll den Inhalt der Struktur `<body>PAYLOADx</body>` jedes einzelnen der .xhtml-Quelldatei in einer einzigen Struktur `<body>PAYLOAD1PAYLOAD2PAYLOADn</body>` in der .xhtml-Zieldatei . Jeweils . Die xhtml-Datei hat nur eine `<body>x</body>`Struktur.

Die Größe und SHA256 der PAYLOADx von jedem .xhtml-Quelldateien werden gespeichert.

Wenn man die Größe jedes PAYLOADx kennt und die Reihenfolge kennt, in der sie verkettet sind (eine direkt nach anderen platziert), wird der SHA256 jedes PAYLOADx in der Struktur `<body>PAYLOAD1PAYLOAD2PAYLOADn</body>` des .xhtml-Zieldatei .

Stimmt der berechnete SHA256 mit dem gespeicherten SHA256 überein, dann wird nachgewiesen, dass sich die jeweiligen PAYLOADx nicht geändert haben.

Die gleiche Mechanik kann für `<style>PAYLOADx</style>`Strukturen befolgt werden. Jeweils . Die xhtml-Quelldatei hat null, eine oder mehrere Strukturen `<style>PAYLOADxy</style>`. Zuerst all die `<style>PAYLOADxy</style>` Strukturen von jedem .xhtml-Quelldateien werden so verkettet, als wären sie eine einzelne `<Style>PAYLOADx</style>`Struktur (die möglicherweise leer ist). Seine Größe und sein SHA256 werden gespeichert (wenn die Größe

größer als Null ist). Und es ist in der Struktur `<style>VERKETTET1PAYLOAD2PAYLOADn</style>` der .xhtml-Zieldatei . Wenn Sie die Größen kennen, können Sie den SHA256 neu berechnen und überprüfen, ob der Inhalt von .xhtml-Zieldatei invariant in Bezug auf die Verkettung von .xhtml-Quelldateien .

Als . Die xhtml-Präsentation besteht aus Texten und Bildern (`<body>` Zone) und anzuwendenden Stilen (`<style>` Zone), mit diesem SHA256-Hashcode-Mechanismus können Sie überprüfen, ob die .xhtml-Quelldateien werden in .xhtml-Zieldatei ohne Änderungen.

Zusammensetzung von .xhtml-Quelldateien mit `<style>` und `<body>` nicht-invariant: "destilliert"

Es kann Fälle geben, in denen es Abweichungen in den `<style>PAYLOADx</Style>` und/oder `<body>PAYLOADx</body>` Zonen eines gibt.xhtml-Quelle.

Beispielsweise ist es eine .html, die in transformiert werden soll.xhtml (siehe Kapitel *Konvertierung .html in .xhtml*). Oder dass bestimmte Definitionen von Stilen in der Datei andere . xhtml-Quelldateien (siehe Kapitel *Begriffsklärung von Styles .xhtml*).

In diesen Fällen kann es eine Vielzahl von Änderungen innerhalb von XML-Strukturen geben, z. B. wenn ein Stil umbenannt wird, um ihn zu verdeutlichen, oder wenn die Werte der Attribute gemäß .xhtml-Standardformat.

Wenn Zeichenfolgen in XML-Strukturen variieren, variieren die Größen und SHA256-Hashcodes.

Was nicht variiert, sind die Texte oder die Bilder. Transformationen in .xhtml meta-information (wie Texte und Bilder dargestellt werden sollen) verändern nicht, was die Information von Texten und Bildern selbst ist (deren Werte). Wo eine 3 ist, wird eine 4 nicht erscheinen. Und wo es ein Bild von einem Auto gibt, wird das Bild eines Flugzeugs nicht erscheinen.

Wenn Sie jedoch mit einem klassischen Zeichen-zu-Zeichen-Vergleich (z. B. dem Vergleichen von Dokumenten in Word) überprüfen, ob keine Änderungen an Texten und Bildern vorgenommen wurden, werden so viele Unterschiede festgestellt, dass dies nicht sinnvoll ist.

Die vorgeschlagene Lösung besteht darin, die .xhtml strukturiert und nimmt als Invarianten nur Text und Bilder.

Auch wenn die Anzeigestile oder Teile des Taggings in der .xhtml oder .html Datei, darf sie Texte oder Bilder nicht variieren. Und wenn Text und Bilder nicht variieren, können Text und Bilder aus der `<body>PAYLOAD</body>` Struktur eines .xhtml(oder .html) Datei in eine funktionierende "destillierte" Datei.

Die Größen und die SHA256-Hashcodes der funktionierenden "destillierten" Datei, die jeder .xhtml oder .html Quelldatei werden gespeichert.

Das. Die xhtml-Zieldatei wird durch Verkettung der `<body></body>` Strukturen der Quelldateien erstellt. Die resultierende "destillierte" Zieldatei ist wiederum die Verkettung

der "destillierten" Quelldateien , da weder Text noch Bilder im Kombinationsprozess variieren.

Wenn sie die Größe und SHA256 jeder "destillierten" Quelldatei kennen, kann überprüft werden, ob sie in der resultierenden "destillierten" Zieldatei unveränderlich sind.

Sobald der Jahresbericht erstellt ist, wird sein "destillierter" Inhalt nach der "destillierten" Größe jedes Kapitels, aus dem er besteht, aufgeteilt. Der SHA256 jedes Stücks muss mit dem SHA256 des "Destillats" des jeweiligen Kapitels übereinstimmen. Auf diese Weise wird sichergestellt, dass der .xhtml Jahresbericht genau die Verkettung der .xhtml oder .html Kapitel ist, aus denen er besteht, unabhängig von etwaigen Änderungen der XML-Strukturen . Jede Änderung führt dazu, dass der SHA256 nicht mehr übereinstimmt.

Theoretisch gibt es Möglichkeiten, die Visualisierung so zu verändern, dass es den Anschein hat, dass der Inhalt anders ist. Es mag ein interessantes Thema der akademischen Forschung sein, aber es ist unpraktisch, da die Manipulation von begrenzter Wirkung und zu berüchtigt wäre.

EINE PRAKTISCHE LÖSUNG:

easyCOMPOSER berechnet die SHA256-Größen und Hash-Codes der <style>, <body> und "distilled" (nur Text und Bilder) jeder zu kombinierenden Quelldatei sowie der .xhtml-Zieldatei kombiniert. Diese Informationen werden als nicht sichtbarer Kommentar am Ende der Zieldatei zusammengefasst platziert. Der SHA256-Hashcode der <style> und <body> von jedem . Die zu kombinierende .xhtml-Quelldatei ist identisch mit dem SHA256-Hashcode der entsprechenden Zone im <style> und <body> der Zieldatei kombiniert. Der SHA256-Hashcode des "Destillats" jeder Quelldatei .xhtml oder .html zu kombinieren ist identisch mit dem SHA256-Hashcode des "Destillats" der Kombinierten Zieldatei , obwohl beim Kombinieren bestimmte Tags oder Attribute in <style> und/oder <body> geändert worden sein können. Weitere Informationen finden Sie im Kapitel *SHA256-Hashcodes in .xhtml-Dateien*

easyCOMPOSER berechnet und zeigt die Größen und die SHA256-Hashcodes der <style>, <body> und "distilled" jeder im *HashCodes-Blatt angegebenen* Datei und prüft, ob es sich um eine kombinierte Datei handelt. Der Vorteil besteht darin, dass es parallel Kapitel für Kapitel in auditiert werden kann.xhtml .html mit Integritätssicherung. Die SHA256-Hashcodes jedes Kapitels werden gespeichert und es kann überprüft werden, ob sich ihre SHA256-Hash-Codes im Jahresbericht nicht geändert haben. Tatsächlich variiert der SHA256 des "Destillats" (nur Text und Bilder) nicht, selbst wenn **easyCOMPOSER** die .html Datei in konvertiert hat.xhtml oder musste Stile anpassen.

easyCOMPOSER berechnet jedes Mal jeden SHA256-Hashcode und prüft (ob genügend Informationen vorhanden sind), der mit den in den Metainformationen der zusammengesetzten Datei deklarierten .xhtml übereinstimmt, indem er rechts neben dem SHA256 ein OK setzt. Wenn es nicht übereinstimmt, zeigt es das abweichende SHA256 oder den erkannten Fehler an.

HashCodes-Blatt

Folder with files: Die Zelle unten gibt den *Pfad* eines Standardverzeichnisses an, in dem:

- Hier sind die Dateien und Unterverzeichnisse, deren *Pfad* nicht angegeben ist
- Die notwendigen Arbeitsdateien werden erstellt und gelöscht.

List of files (and folders) for hash codes: In den folgenden Zellen werden die Dateien platziert, von denen die SHA256-Hash-Codes angezeigt werden. Wenn ein Unterverzeichnis eingefügt wird, werden alle darin enthaltenen Dateien verarbeitet.

Zum  **Ausführen hier klicken**

File name: Name der verarbeiteten Datei oder des verarbeiteten Unterverzeichnisses.

- Wenn es sich um eine .zip datei oder ein Verzeichnis handelt, wird es erweitert und verarbeitet alle darin enthaltenen Dateien und Verzeichnisse.
- Der Einzug (der Dateiname wird nach rechts gesetzt) wird verwendet, um die hierarchische Reihenfolge innerhalb der Datei oder des Verzeichnisses anzuzeigen.
- Wenn es sich um eine zusammengesetzte .xhtml Datei handelt, haben die folgenden Zeilen den Namen in Klammern jeder Datei, die Teil der Komposition ist.
 - Die erste Datei ist die zusammengesetzte Datei selbst, jedoch ohne die Kompositions-Meta-Informationen (ca. 500 Bytes).
 - Im Folgenden sind die einzelnen Quelldateien aufgeführt , aus denen die zusammengesetzte Datei besteht.

File Size: Dateigröße in Byte.

File SHA256: SHA256-Hashcode der Menge aller Bytes der Datei.

Body Size: Größe der <body>Zone in Bytes.

Body SHA256: SHA256-Hashcode der Menge aller Bytes in der Zone <body>.

Style Size: Größe der zone <style> in Byte.

Style SHA256: SHA256-Hashcode der Menge aller Bytes in der Zone <style>.

Distilled Size: Größe des <body> Zone "*destilliert*" (nur Text und Bilder) in Bytes.

Distilled SHA256: SHA256-Hashcode aller Bytes im <body> Zone "*destilliert*" (nur Text und Bilder).

Check for export distilled as file .distilled.txt Wenn aktiviert, wird eine Datei exportiert, die die <body> Zone "*destilliert*" (nur Text und Bilder) der Datei enthält. Die Exportdatei wird im Standardverzeichnis erstellt und behält den gleichen Dateinamen bei, indem am Ende ".*distilled.txt*" hinzugefügt wird.

Destillationsalgorithmus (nur Texte und Bilder) einer .xhtml oder .html Datei:

1. Löschen Sie vom Anfang der Datei in den **<Body... >**-Tag
2. Aus dem **Tag </body>** bis zum Ende der Datei löschen
3. Alle Kommentare **löschen <!-- ... -->**
4. Ersetzen Sie jedes Bild **<img.... src="data:image/....." />** nur durch **data:image/.....**
5. Löschen Sie alle **<Tag... >** und **</Tag>** XML tags
6. Löschen Sie alle Zeichen-Escape-Sequenzen **&xxx;** wie ** ** oder ** **;
7. Löschen Sie Wagenrückläufe, Zeilenumbrüche, Tabstopps und Leerzeichen.

Am Ende gibt es einen invarianten "destillierten" Inhalt, etwa so:

Memoryofthe exercise2020data:image/OrK... 5CYII=AnnualAccountsoftheexercise.....

Der "Destillationsalgorithmus" kann in jeder Sprache oder mit regulären Ausdrücken programmiert werden. Die Zeichenkodierung ist UTF8 nach dem ESEF-Format.

Sha256-Größen und Hash-Codes speichern

Es wird empfohlen, die folgenden .xhtml-Kommentarstruktur zum Speichern von Größe und SHA256-Hashcodeinformationen. Diese Struktur wird für jede Quelldatei und für die Zieldatei selbst wiederholt. Diese Strukturen werden nach **</body>** in der Zieldatei platziert.

```
<!--hashfile order="" file=""
filesize="" filehash=""
bodysize="" bodyhash=""
stylesize="" stylehash=""
distilledsize="" distilledhash=""
-->
```

order="" Erforderlich. Ziffernfolge 0.. 9 nicht wiederholt. Gibt die Reihenfolge dieser Quelldatei innerhalb der Zieldatei an. **order="0"** gibt die Zieldatei an.

file="" Erforderlich. Dateiname ohne Pfad

filesize="" Optional. Dateigröße. Im Fall von **order="0"** entspricht die Dateigröße dem Tag **</body>** einschließlich.

filehash="" Optional. Nur wenn die Dateigröße > 0 ist. SHA256-Hashcode dieser Datei. Im Fall von **order="0"** ist die Dateigröße bis zum Tag **</body>** einschließlich.

bodysize="" Optional. Größe der PAYLOAD in der Dateistruktur **<Body... >PAYLOAD</body>**

bodyhash="" Optional. Nur wenn die Körpergröße > 0. SHA256-Hashcode von PAYLOAD in der Dateistruktur **<Body... >PAYLOAD</body>**

stylesize="" Optional. Größe der PAYLOAD in der Struktur **<Stil... >PAYLOAD</style>** der Datei, verkettet, wenn es mehrere gibt.

stylehash="" Optional. Nur wenn **stylesize > 0** ist. SHA256-Hashcode der PAYLOAD in der Dateistruktur **<stil... >PAYLOAD</style>**, verkettet, wenn es mehrere gibt.

distilledsize="" Optional. Größe des "Destillats" der PAYLOAD in der Dateistruktur <Körper... >PAYLOAD</body>.

distilledhash="" Optional. Nur wenn destilliert > 0. SHA256-Hashcode des PAYLOAD "Destillats" in der Dateistruktur <Body... >PAYLOAD</body>

Beispiel:

```
</body><!--hashfile order="0" file="company-2021-12-31.xhtml"
filesize="2051206" filehash="TockHgZ0BgSQCgLEx13ZsWdNLCqIDmFKC5rwR6SBVyQ="
bodysize="2042432" bodyhash="zxm9BacGlski1dbp+DfQ/zVuj2agm9rmaVIXu9dI36I="
stylesize="5580" stylehash="Bw+W0qjof9UuTfJasOUOpNV32d7ZBXRbotth+uqtTSA="
distilledsize="133796"
distilledhash="jmc62732j6gzjXcA8GvgEn7mTRghJmLMQYby2ZW8Q2U=" -->
```

Vorkonstituierte Beweismittel:

Eine einfache Möglichkeit, SHA256 unauslöschlich zu speichern, besteht darin, sie als Nachricht über Whatsapp, Instagram, Twitter oder einen anderen sicheren Messaging-Dienst zu senden. Sowohl der Absender als auch der Empfänger (oder jeder andere, der Zugriff hatte) konnten diese Nachricht als vorgefertigtes gerichtliches Beweismittel verwenden.⁴

⁴ Kann auch verwendet werden digital Unterschrift, oder fortgeschrittene solutions wie sigstore.dev (Software transparent Unterschrift) oder openfiling.info/blockchain (Signatur-Blockchain, desselben Autors dieses Dokuments)