

Dokumentation der Produktionsstrecke für JATS-XML auf InDesign-Basis

Stand: 02.12.2022, Version: 2.0

Dokumentation der Produktionsstrecke JATS-XML

Die vorliegende Dokumentation beschreibt Aufbau, Funktionen und Bedienung der Produktionsstrecke für JATS-XML auf Basis des InDesign-Exports, die im Auftrag des DAI im Jahr 2019 von *digital publishing competence* entwickelt wurde.

Die Dokumentation ist in zwei wesentliche Bereiche aufgeteilt:

- Die **Nutzer-Dokumentation** beschreibt die Bedienung der Produktionsstrecke im Rahmen eines dafür aufgesetzten Oxygen-Projektes.
- Die **Technische Dokumentation** beschreibt Projektaufbau, Code-Struktur und eingesetzte Werkzeuge und Software-Komponenten.

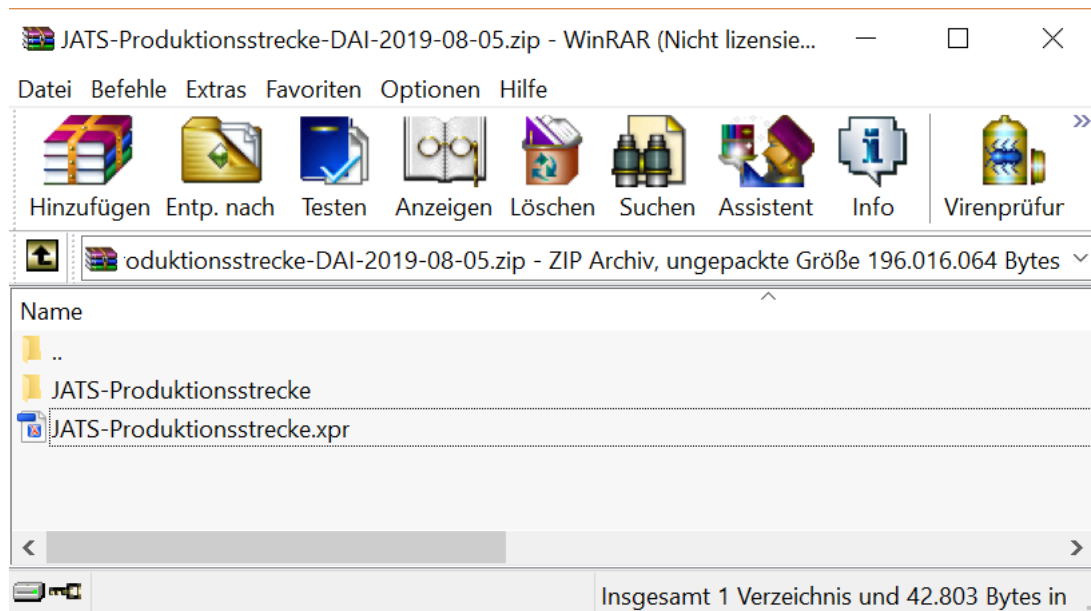
Inhalt

Dokumentation der Produktionsstrecke JATS-XML	1
Nutzer-Dokumentation	3
Übergabe und Entpacken des Oxygen-Projekts	3
Umgang mit neuen Versionen der Produktionsstrecke	3
Überblick über die enthaltenen Dateien	4
Wesentliche notwendige Bedienelemente von Oxygen	6
Überblick über die wesentlichen Bedienschritte	7
Schritt 1: InDesign-Export und exportierte Dateien	7
Schritt 2: InDesign-Preflight-Prüfung	8
Schritt 3: Arbeiten mit dem InDesign-Prüfprotokoll	9
Schritt 4: JATS-Konvertierung, Step 1	12
Schritt 5: JATS-Konvertierung, Step 2	12
Schritt 6: JATS-Konvertierung, Step 3	12
Schritt 7: Validierung des JATS-XML	14
Schritt 8: Erzeugung HTML-Preview für JATS-XML	15
Output der Konvertierung und Übergabe für die Lens-Viewer-Applikation	16
Arbeiten mit dem Menü „Transformations-Szenarios konfigurieren“	17
Technische Dokumentation	18
Im Projekt verwendeter JATS-Dialekt	18
Besonderheiten der für das DAI erzeugten JATS-Ausprägung	18
Eingesetzte Werkzeuge und Software-Komponenten	19
Aufbau und Arbeitsweise der InDesign-Preflight-Prüfung	20
Aufbau und Arbeitsweise von JATS-Konverter, Step 1	21
Aufbau und Arbeitsweise von JATS-Konverter, Step 2	22
Aufbau und Arbeitsweise von JATS-Konverter, Step 3	23
XSLT-Transformation für HTML-Preview	24
Weitere Dokumentationen	24

Nutzer-Dokumentation

Übergabe und Entpacken des Oxygen-Projekts

Die Produktionsstrecke wird Ihnen in Form einer ZIP-Datei übergeben, in dem das gesamte notwendige Dateisystem sowie eine Oxygen-Projekt-Datei enthalten ist:



Als ersten Bedienschritt entpacken Sie die ZIP-Datei bitte in einen beliebigen Ordner in Ihrem Dateisystem. Der eingesetzte PC sollte über eine lauffähige Oxygen-Installation verfügen.

Umgang mit neuen Versionen der Produktionsstrecke

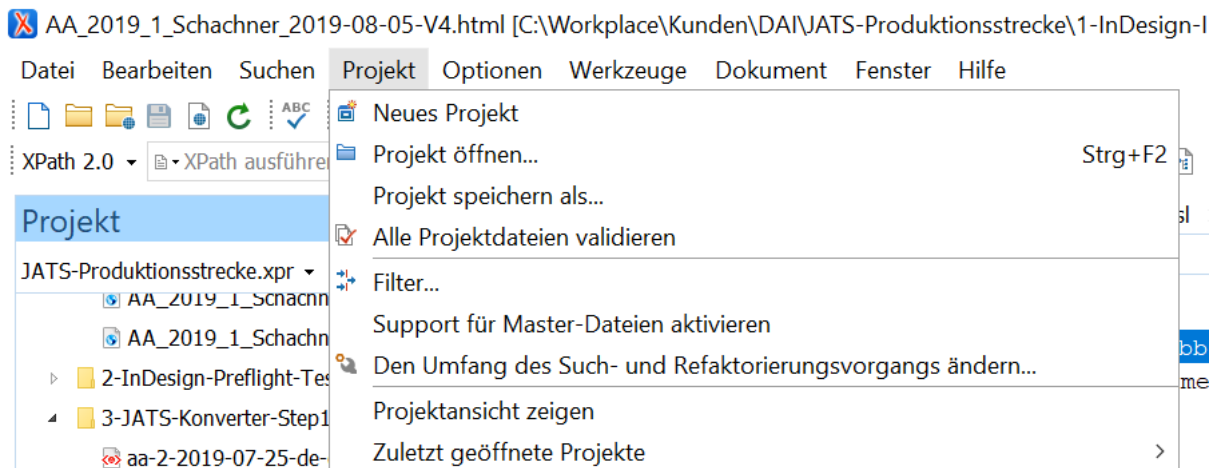
Sollten Sie auf einem Arbeitsplatz-PC bereits eine Version der Produktionsstrecke abgelegt haben und von digital publishing competence eine neue Version erhalten, dann gehen Sie bitte folgendermaßen vor:

1. Sichern Sie eventuell noch benötigte Produktionsdaten aus den Produktionsstrecken-Verzeichnissen in andere Ordner auf Ihrem Arbeitsplatz.
2. Löschen Sie das Verzeichnis „JATS-Produktionsstrecke“ und die gleichnamige .xpr-Datei komplett.
3. Entpacken Sie das ZIP-File mit der neuen Version in das Verzeichnis, in dem die Produktionsstrecke liegen soll.

Die Produktionsstrecke wird jedes Mal in komplett neuem Zustand übergeben, ein sauberes Dateisystem ist sinnvoll um eventuelle Versionskonflikte auszuschließen.

Überblick über die enthaltenen Dateien

Die im Paket enthaltene xpr-Datei enthält das Oxygen-Projekt: hier sind alle notwendigen Dateien, Ordner und Einstellungen für den reibungslosen Ablauf der Konvertierung hinterlegt. Zum Start öffnen Sie die Projekt-Datei entweder mit einem Doppelklick in Windows Explorer oder innerhalb von Oxygen über den Befehl „Projekt“ / „Projekt öffnen“:



Innerhalb des Ordners „JATS-Produktionsstrecke“ finden Sie folgendes Dateisystem vor:

- ✓ _Dokumentation
- ✓ _DTD
- ✓ _Produktionsdaten-AA
- ✓ _Referenz-Dateien
- ✓ _XSLT
- ✓ 1-InDesign-Input-Dateien
- ✓ 2-InDesign-Preflight-Test
- ✓ 3-JATS-Konverter-Step1
- ✓ 4-JATS-Konverter-Step2
- ✓ 5-JATS-Konverter-Step3
- ✓ 6-HTML-Preview

Die Ordner werden identisch auch in der Projektansicht von Oxygen angezeigt und haben folgende Bedeutung:

- **_Dokumentation:** Hier sind die vorliegende Dokumentation sowie alle weiteren notwendigen Dokumentationen abgelegt.
- **_DTD:** In diesem Ordner befindet sich die JATS-DTD, die zur Validierung der erzeugten Dateien verwendet wird. Bitte ändern oder löschen Sie hier keine Dateien.
- **_Produktionsdaten-AA:** In diesem Ordner befinden sich die Produktionsdaten der ersten AA-Ausgabe, d.h. sowohl InDesign-Export-Dateien als auch die daraus erzeugten JATS-XML-Dateien. Dieser Ordner dient insofern auch zur produktiven Datenübergabe an das DAI.
- **_Referenz-Dateien:** Der Ordner enthält die Referenz-Dateien, anhand derer der JATS-Konverter entwickelt und getestet wurde. Sie können für ihre ersten eigenen Schritte die Dateien nutzen, um den Umgang mit dem Konverter auszuprobieren.

- **_XSLT:** Der Ordner enthält die XSLT-Skripte für die einzelnen Transformations-Stufen des JATS-Konverters. Bitte ändern oder löschen Sie hier keine Dateien.
- **1-InDesign-Input-Dateien:** In diesem Ordner werden die XHTML-Dateien aus dem InDesign-Export abgelegt und von dort aus der weitere Konvertierungs-Prozess gestartet.
- **2-InDesign-Preflight-Test:** In diesem Ordner werden die Prüfprotokolle für den Test der InDesign-Export-Dateien abgelegt, die als HTML-Report ausgegeben werden.
- **3-JATS-Konverter-Step1:** In diesem Ordner legt der JATS-Konverter die Ergebnisse der ersten Transformations-Stufe ab; von hier aus wird die zweite Konvertierungs-Stufe gestartet.
- **4-JATS-Konverter-Step2:** In diesem Ordner legt der JATS-Konverter die Ergebnisse der zweiten Transformations-Stufe ab; von hier aus wird die dritte Konvertierungs-Stufe gestartet.
- **5-JATS-Konverter-Step3:** In diesem Ordner legt der JATS-Konverter die Ergebnisse der dritten Transformations-Stufe ab. Auf den Dateien in diesem Ordner wird die JATS-Validierung vorgenommen. Ist diese erfolgreich, können die hier erzeugten XML-Dateien für den Lens-Viewer-Import verwendet werden.
- **6-HTML-Preview:** In diesem Ordner werden die HTML-Previews auf Basis des JATS-XML generiert, die Sie für eine Voransicht der erzeugten Dateien vor Import in eine Lens-Viewer-Anwendung nutzen können.

Wichtig: Für die reibungslose Funktion der Produktionsstrecke ist es entscheidend, dass die Ordner-Struktur exakt wie übergeben bestehen bleibt. Bitte ändern Sie deswegen keine Ordernamen oder löschen Ordner.

Wesentliche notwendige Bedienelemente von Oxygen

Von den vielen Funktionen, die Oxygen als XML-Umgebung mitbringt, werden Sie beim Arbeiten mit der JATS-Produktionsstrecke nur wenige benötigen. Die für Ihr Arbeiten zentralen Elemente sind bereits über die Projekteinstellungen konfiguriert:

Die Projekt-Ansicht zeigt Ihnen alle Ordner und Dateien im Projekt

In der Symbolleiste sind einige wenige wichtige Funktionen untergebracht (s.u.)

Die Dokumentansicht zeigt Ihnen den Quellcode der HTML- oder XML-Datei an und gibt Ihnen die Möglichkeit zur Bearbeitung.

Die Gliederungsansicht zeigt Ihnen bei geöffneten HTML- oder XML-Dateien deren Struktur in einer kompakten, hierarchisch navigierbaren Form an.

Die Modellansicht zeigt Ihnen bei offenen HTML/XML-Dateien die Inhaltsmodelle der aktuell angewählten Elemente an.

Wichtige Funktionen in der Symbolleiste:

Aufruf der XML-Validierung

Aufruf der für die aktuelle Datei hinterlegten nächsten Transformation

Aufruf des Menüs, in dem alle im Projekt hinterlegten Transformationen zugreifbar sind

Überblick über die wesentlichen Bedienschritte

Für eine JATS-Produktion sind folgende wesentliche Bedienschritte notwendig:

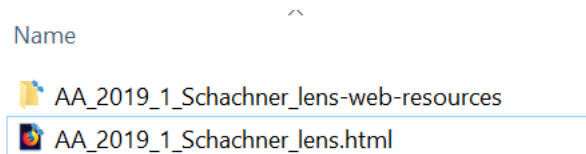
1. XHTML-Export aus InDesign und Kopieren der exportierten Dateien in das Dateisystem der Konvertierung
2. Aufruf der gewünschten Datei in Oxygen und Start des InDesign-Preflight-Test
3. Analyse des Prüfprotokolls

ggf. Rückkehr nach InDesign und Behebung von Tagging-Fehlern; dann erneutes Ausführen von Schritt 1-3, solange bis alle Fehler im Prüfprotokoll behoben sind.

4. Ausführen von Schritt 1 der JATS-Konvertierung
5. Ausführen von Schritt 2 der JATS-Konvertierung
6. Ausführung von Schritt 3 der JATS-Konvertierung
7. XML-Validierung der JATS-XML-Dateien
8. Optional: Generierung des HTML-Preview für die Betrachtung des Zwischenergebnis vor Import in Lens-Viewer-Applikation

Schritt 1: InDesign-Export und exportierte Dateien

Wenn Sie den XHTML-Export aus InDesign durchgeführt haben (siehe dazu die Dokumentation zum Arbeiten mit dem InDesign-Template), dann sollten Sie in ihrem Dateisystem danach folgende Dateien vorfinden:

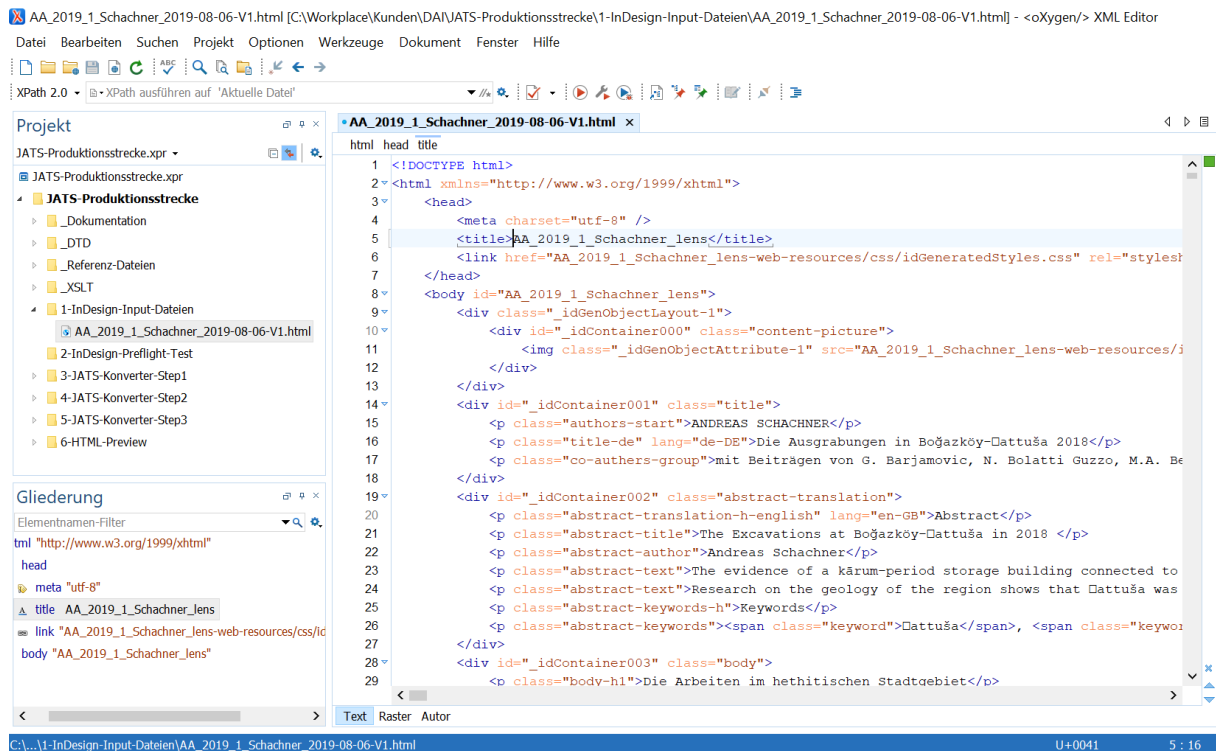


- Die .html-Datei enthält den Content des Artikels im InDesign-XHTML-Export-Format. Der Dateiname kann beliebig gewählt werden und sollte im Optimalfall dem entsprechen, der für die XML-Datei auch verwendet werden soll.
- Im parallel dazu abgelegten Ordner befinden sich alle im Artikel eingebundenen Bilder sowie ein von InDesign generiertes CSS zu den Daten. Die Bilder sind später in der Verarbeitung noch wichtig; das CSS wird nicht benötigt.

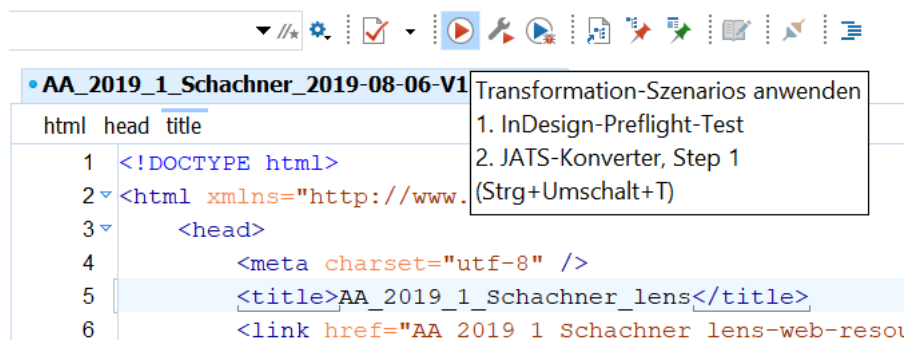
Wenn Sie eine JATS-Produktion beginnen möchten, legen Sie die HTML-Datei bitte zunächst im Ordner „1-InDesign-Input-Dateien“ ab. Nehmen Sie den Ordner mit den Bildern und kopieren Sie diesen ins Verzeichnis „5-JATS-Konverter-Step3“.

Schritt 2: InDesign-Preflight-Prüfung

Öffnen Sie zunächst die HTML-Datei in Oxygen per Doppelklick auf den Dateinamen im Ordner „1-InDesign-Input-Dateien“ in der Projekt-Ansicht. Danach sollten Sie folgendes Bild vorfinden:



Für alle Dateien im Ordner „1-InDesign-Input-Dateien“ ist im Projekt ein Transformations-Szenario hinterlegt, das sowohl das InDesign-Prüfskript als auch die erste Stufe der JATS-Konvertierung ausführt. Sie starten die Transformation über Klick auf den Button „Transformations-Szenarios anwenden“ in der Symbolleiste:



Dabei geschieht folgendes:

- Auf Basis der InDesign-Daten wird das Prüfprotokoll generiert, im Ordner „2-InDesign-Preflight-Test“ abgelegt und gleichzeitig im System-Browser geöffnet. Die HTML-Datei erhält den Namen der ursprünglichen Datei, ergänzt um „InDesign-Prüfprotokoll [DATUM]“.
- Die Ergebnisse von Step 1 der JATS-Konvertierung werden im Ordner „3-JATS-Konverter-Step1“ abgelegt und gleichzeitig als XML-Datei in Oxygen geöffnet. Die Datei behält ihren ursprünglichen Dateinamen, wird aber als XML-Datei gespeichert.

Schritt 3: Arbeiten mit dem InDesign-Prüfprotokoll

Wenn Sie das vom Prüf-Skript generierte Prüfprotokoll im Browser geöffnet haben, zeigt sich folgende Ansicht:

Prüfbericht für InDesign-Export zur Verarbeitung in JATS-XML

Artikel: Die Ausgrabungen in Boğazköy-Hattuša 2018

Autoren: ANDREAS SCHACHNER

Ausgabe: 1/2019

Konvertiert: 6.8.2019, 10:39:15

#	Prüfung	Ergebnis	Typ
1.	Dokument-Struktur	Allgemeine Prüfungen der InDesign-Dokument-Struktur	Bereich
1.1	Content-Image	Content-Image gefunden: OK	Info
1.2	Abstract	Abstract gefunden: OK	Info
1.3	Abstract-Translation	1 mal Abstract-Translation gefunden: OK	Info
1.4	Title-Container	Title-Container gefunden: OK	Info
1.5	Artikel-Body	Artikel-Body gefunden: OK	Info
1.6	Artikel-Bilder	80 Bildelemente mit dem Objektrahmen-Format 'picture' gefunden: OK	Info
1.7	Referenzen	Referenzen gefunden: OK	Info

- Im Kopf der Datei werden Ihnen zunächst die wesentlichen Metadaten zur Identifikation von Artikel und Konvertierungsdatum angegeben.
- Der Report besteht aus einer Tabelle mit insgesamt acht Sektionen, die den wesentlichen strukturellen Teilen einer JATS-Datei entsprechen.
- Jede Zeile entspricht einem Test, der bei der Prüfung ausgeführt wurde. Als Test realisiert haben wir jedes Struktur-Problem, das beim Aufsetzen und Testen der Produktionsstrecke schon einmal zu Fehlern geführt hat.

Für die Zeilen gibt es folgende Typen von Ausgabe, die wichtig zu beachten sind:

- **„Bereich“:** Die Zeile beinhaltet keinen Test, sondern dient nur der Gliederung des Report.
- **„Info“:** Alle Zeilen mit Infos beinhalten erfolgreiche Tests. Die Ausgabe dient nur der Information und muss nicht weiter bearbeitet werden. Info-Zeilen sind in der Regel sehr kurz und knapp gefasst, im Gegensatz zu den Fehlermeldungen der jeweiligen Tests.
- **„Warnung“:** Als Warnung bezeichnen wir Prüfungen, die auf eventuelle inhaltliche Probleme hinweisen können, die aber im weiteren Produktionsverlauf nicht zu Konvertierungsfehlern führen werden. Warnungen sollten inhaltlich und auf Plausibilität geprüft werden, erzeugen aber keine technischen Probleme.
- **„Fehler“:** Treten Fehler auf, dann sind Konventionen zu Aufbau und Struktur der InDesign-Export-Datei auf eine Weise verletzt worden, die im weiteren Verlauf zu Konvertierungs- bzw. Validierungs-Fehlern führen wird. Fehler müssen behoben werden, bevor weiter produziert werden kann. Im Fall von Fehler erzeugen wir eine ausführliche Beschreibung, die einem Bearbeiter in InDesign so viele Informationen wie möglich zur Fehlerstelle und dem zugrundeliegenden Problem mit an die Hand gibt.
- **„Schwerer Fehler“:** Schwere Fehler unterscheiden sich von Fehlern insofern, als hier das Problem nicht nur punktueller Natur ist: In der Regel ist hier zwar nur eine einzelne Stelle in der InDesign-

Export-Datei die Quelle des Problems, aber aufgrund ihres Charakters werden Folgefehler an mehreren anderen Stellen der Datei auftreten, d.h. es wird mehrere Tests geben, die aufgrund derselben Fehlerstelle fehlschlagen. Treten schwere Fehler auf, ist es sinnvoll, zunächst die Quelle dieses Fehlers zu beheben, erneut zu exportieren und die Prüfung laufen zu lassen – erst dann wird der Prüfbericht übersichtliche Mengen an Meldungen enthalten.

In der Praxis heißt dies:

- Beinhalten alle Zeilen ausschließlich die Typen Bereich, Info oder Warnung, so können Sie problemlos weiter produzieren.
- Gibt es Fehler oder schwere Fehler, so gehen Sie für die Bearbeitung zurück in InDesign, beheben Sie die Fehlerquelle, exportieren Sie erneut und führen die Prüfung nochmals durch.
- Wenn alle Fehler behoben sind, kann weiter produziert werden. **Auf Basis von Export-Dateien mit Fehlern oder schweren Fehlern wird eine erfolgreiche Produktion nicht möglich sein.**

Beispiel 1: Warnung

Ein typisches Beispiel für eine Warnung ist diese Meldung:

5.	Title	Prüfungen des Title-Textrahmens und der enthaltenen Absatz-Formate	Bereich
5.1	Absatz-Formate in Title	Der Textrahmen 'title' enthält nur die erwarteten Absatz-Formate: OK	Info
5.2	Titel-Element	Titel-Element/Absatz mit Absatz-Format 'title-[SPRACHE]' wurde genau einmal im Textrahmen 'title' gefunden: OK	Info
5.3	Subtitel-Element	Subtitel-Element/Absatz mit Absatz-Format 'subtitle' wurde im Textrahmen 'title' NICHT gefunden: Bitte überprüfen Sie die InDesign-Auszeichnung an dieser Stelle und verifizieren Sie, ob der Artikel tatsächlich aus inhaltlichen Gründen keinen Subtitel enthält. Wird hier eine falsche Auszeichnung verwendet, dann wird das Subtitel-Element in den JATS-Metadaten leer bleiben.	Warnung

In diesem Fall enthält ein Artikel aus inhaltlichen Gründen keinen Subtitel unterhalb des Artikel-Titels. Dies kann inhaltlich korrekt sein, aber auch ein möglicher Hinweis auf ein falsches Tagging eines durchaus existierenden Subtitels sein. Deswegen wird diese Struktur als Warnung ausgegeben. Ist die Struktur inhaltlich korrekt, können Sie die Warnung ignorieren.

Beispiel 2: Fehler

Ein typisches Beispiel für einen Fehler ist diese Meldung:

7.6	Referenz-Verweise ohne Ziel	2 Referenz-Verweise gefunden, die aufgrund ihres Textinhaltes nicht mit Literatur-Referenzen verknüpfbar sind: Element: span, Format: notes-reference-link, Text: 'Wild u. '; Element: span, Format: notes-reference-link, Text: 'a. 2008'; . Bitte prüfen Sie hier, ob die Schreibweise im Literatur-Verweis (Zeichenformat 'notes-reference-link') buchstabengenaue der Schreibweise in der entsprechenden Literatur-Referenz (Zeichenformat 'reference-label' in Absatzformat 'references') entspricht.	Fehler
-----	-----------------------------	--	--------

In diesem Fall wurden Verweise auf Literatur-Referenzen im Artikel-Text gefunden, die sich aufgrund ihres Text-Inhaltes nicht mit einer Referenz verbinden lassen. Die Quelle des Problems kann in diesem Fall sowohl beim Verweis als auch in der Literatur-Referenz selber befinden. Zur Fehlersuche ist hier das InDesign-Zeichen-Format und der gefundene Text im Format angegeben. Für die Behebung des Fehlers muss der Text im Referenz-Verweis mit dem Label in der Literatur-Referenz verglichen werden. Der Fehler ist dann behoben, wenn der Referenz-Verweis buchstabengenaue einem Label einer Literatur-Referenz entspricht.

Beispiel 3: Schwerer Fehler

Ein typisches Beispiel für einen schweren Fehler ist diese Meldung:

5.2	Titel-Element	Titel-Element/Absatz mit Absatz-Format 'title-[SPRACHE]' wurde 2 mal im Textrahmen 'title' gefunden: Bitte prüfen Sie hier die Auszeichnung, für eine korrekte Verarbeitung des title-Absatz darf diese Auszeichnung nur in genau einem Absatz verwendet werden. Wenn dieser Fehler auftritt, ist es sehr wahrscheinlich, dass im vorliegenden Prüf-Protokoll größere Mengen Folgefehler angezeigt werden, da von der eindeutigen Zuordnung des title-Formates auch die gesamte Sprach-Behandlung des JATS-Konverters abhängig ist. Fehler in den Abschnitten zu Abstracts und Bildern sind höchstwahrscheinlich Folgefehler. Bitte beheben Sie in diesem Fall zunächst den Auszeichnungsfehler 5.2, starten Sie dann nochmals einen InDesign-Export und lassen die Prüfung erneut laufen.	Schwerer Fehler
-----	---------------	--	-----------------

In diesem Fall wurde das Absatz-Format für den Artikel-Titel mehrfach gesetzt. Da an dieser Stelle auch die Dokument-Sprache gesetzt wird, erzeugt dieser Fehler vielfältige Folgefehler: Von der Sprachbehandlung hängt auch die Behandlung der Abstracts und der Bilder ab. Insofern werden in diesem Prüfprotokoll an mehreren Stellen Folgefehler ausgegeben werden.

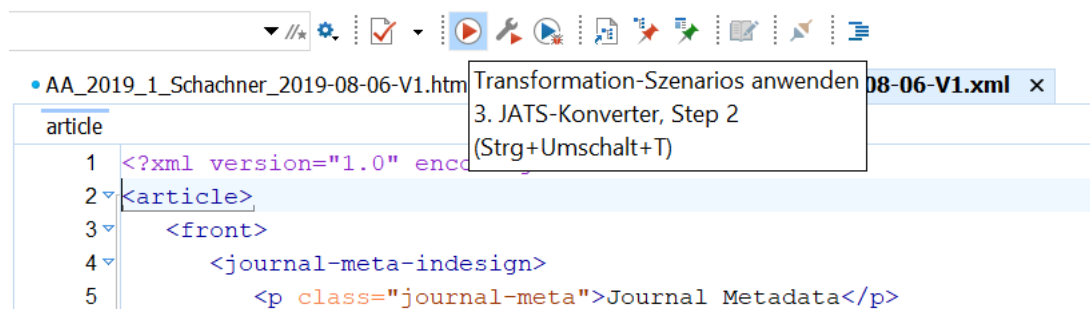
Sind Sie mit schweren Fehlern konfrontiert, ist es sinnvoll, zunächst die Quelle dieses Fehlers in InDesign zu beheben, daraus erneut eine XHTML-Datei zu exportieren und die Prüfung nochmals laufen zu lassen – erst dann wird der Prüfbericht übersichtliche Mengen an Meldungen enthalten.

Schritt 4: JATS-Konvertierung, Step 1

Der erste Schritt der JATS-Konvertierung wurde bereits über das Transformations-Szenario in Schritt 2 ausgelöst. Sie finden die Ergebnisse als XML-Datei in der XML-Ansicht von Oxygen.

Schritt 5: JATS-Konvertierung, Step 2

Starten Sie die nächste Transformation über Klick auf den Button „Transformations-Szenarios anwenden“ in der Symbolleiste:



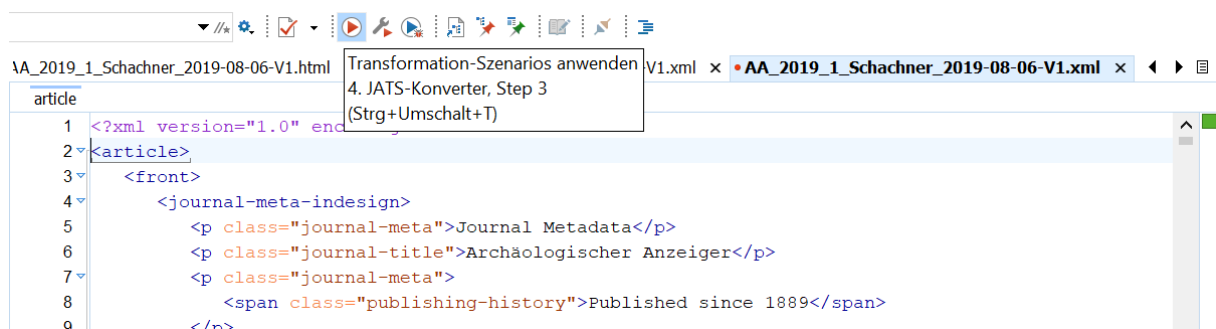
Dabei geschieht folgendes

- Die Ergebnisse von Step 2 der JATS-Konvertierung werden im Ordner „4-JATS-Konverter-Step2“ abgelegt und gleichzeitig als XML-Datei in Oxygen geöffnet. Die Datei behält ihren ursprünglichen Dateinamen.

Schritt 6: JATS-Konvertierung, Step 3

Sie finden die Ergebnisse von Step 2 als XML-Datei in der XML-Ansicht von Oxygen.

Starten Sie die nächste Transformation über Klick auf den Button „Transformations-Szenarios anwenden“ in der Symbolleiste:



Dabei geschieht folgendes:

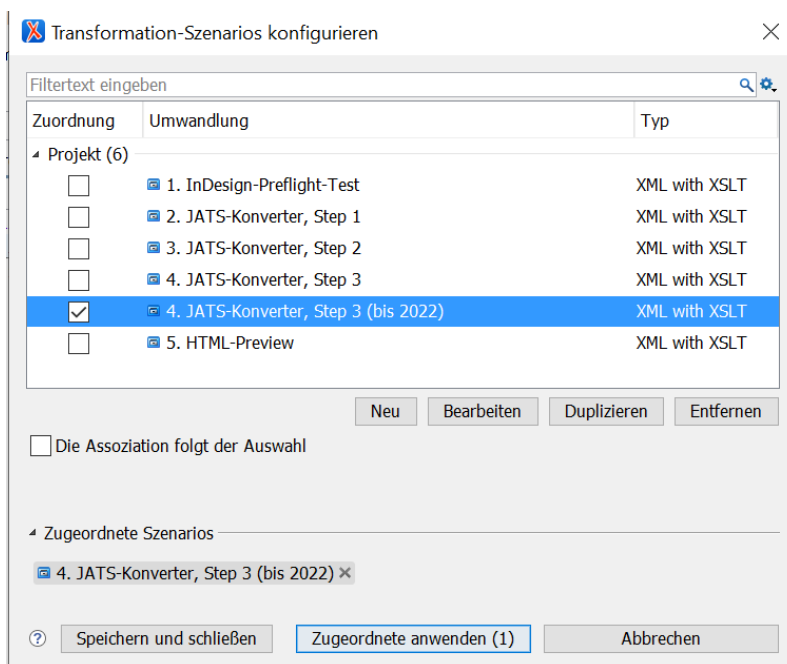
- Die Ergebnisse von Step 3 der JATS-Konvertierung werden im Ordner „5-JATS-Konverter-Step3“ abgelegt und gleichzeitig als XML-Datei in Oxygen geöffnet. Die Datei behält ihren ursprünglichen Dateinamen, ergänzt um das Tagesdatum der Konvertierung.

Ab der Version 2.0 werden in diesem Konvertierungsschritt die Journal-Metadaten nicht mehr aus einer statischen XML-Datei entnommen, sondern aus den InDesign-Daten konvertiert. Dieser Unterschied wird im XSLT-Code über einen Versions-Parameter gesteuert. Sollten Sie nach Dezember 2022 noch einmal InDesign-Daten im Tagging für die Version 1.0 konvertieren müssen, dann gehen Sie hier in Schritt 6 stattdessen über die Funktion „Transformations-Szenarios konfigurieren“ von Oxygen:



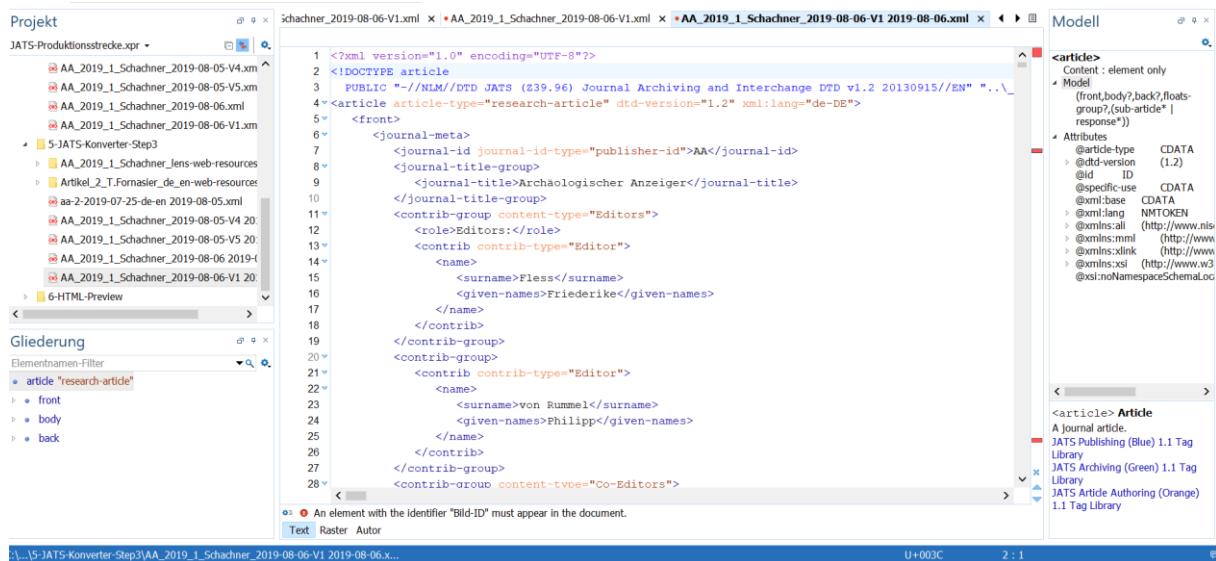
Transformation-Szenarios konfigurieren (Strg+Umschalt+C)

Wählen Sie dann den Eintrag „JATS-Konverter, Step 3 (bis 2022)“ und starten die Transformation über „Zugeordnete anwenden“:

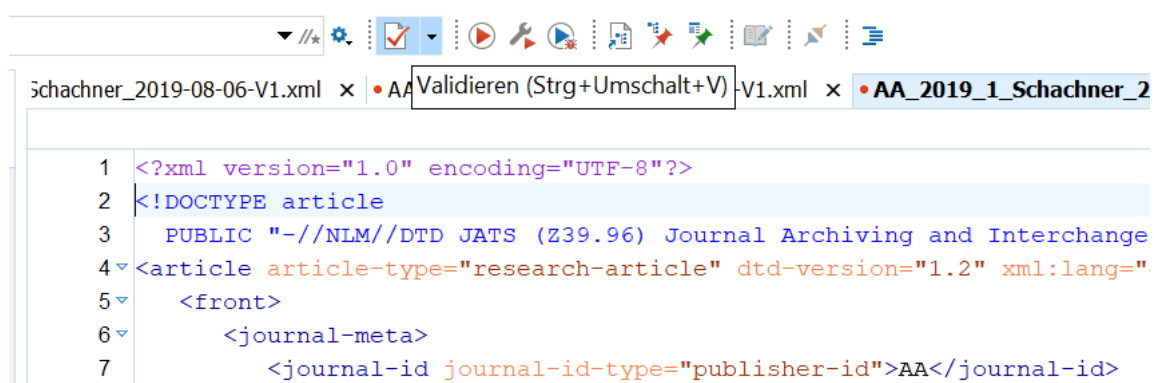


Schritt 7: Validierung des JATS-XML

Die erzeugte JATS-Datei finden Sie in der XML-Ansicht von Oxygen. Sie sollten folgendes Bild vorfinden:

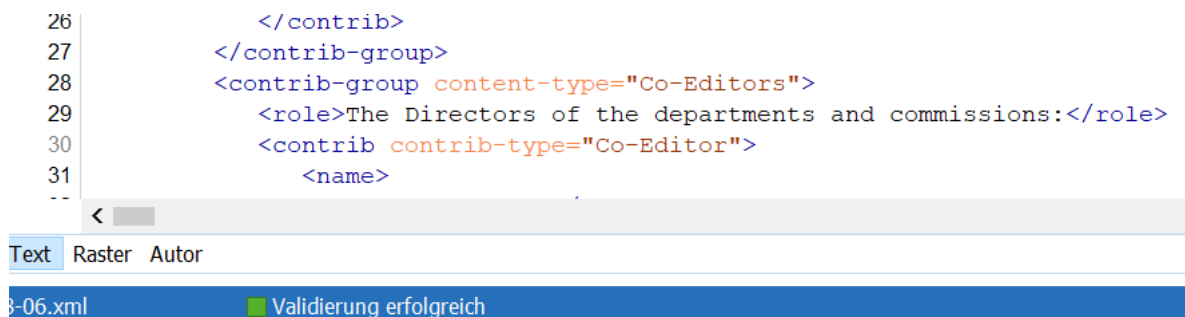


Stoßen Sie nun die Validierung der Datei nach der JATS-DTD über den Button „Validieren“ in der Symbolleiste an:

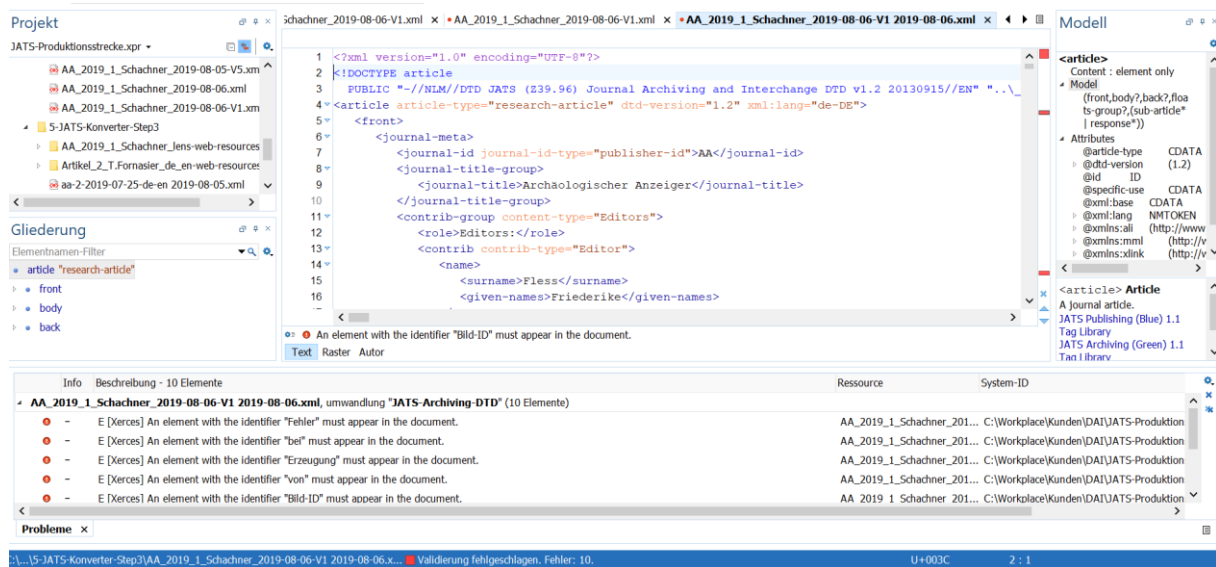


Im Projekt sind die Validierungs-Einstellungen für den entsprechenden Dateipfad so gesetzt, dass automatisch nach der korrekten DTD validiert wird.

Ist die Validierung erfolgreich, erhalten Sie eine entsprechende Meldung in der Statusleiste von Oxygen:



Ist die Validierung nicht erfolgreich, so gibt der XML-Parser eine Liste von Strukturfehlern aus:

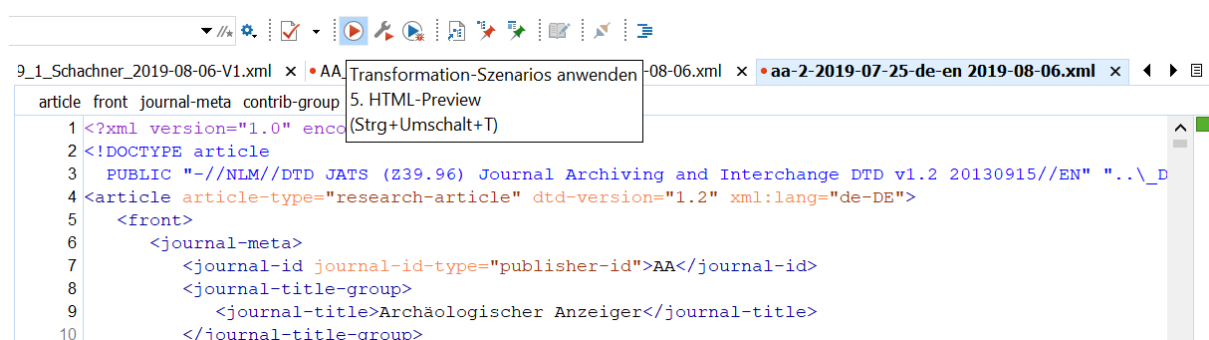


Sollte dies der Fall sein, obwohl alle Prüfungen des InDesign-Preflight-Test erfolgreich waren, wenden Sie sich bitte an Ihren Konvertierungs-Entwickler. Schicken Sie dazu unbedingt die ursprünglich verwendete InDesign-Export-HTML-Datei mit, denn nur auf dieser Basis kann eine Fehler-Analyse erfolgen.

Schritt 8: Erzeugung HTML-Preview für JATS-XML

Wenn Sie ein HTML-Preview für die konvertierte JATS-Datei erzeugen wollen, dann öffnen Sie die Datei aus Step3 der Konvertierung in der XML-Ansicht von Oxygen.

Sie starten die Transformation über Klick auf den Button „Transformations-Szenarios anwenden“ in der Symbolleiste:



Dabei geschieht folgendes:

- Auf Basis der JATS-Datei wird eine HTML-Vorschau geöffnet und direkt im System-Browser geöffnet. Die Datei wird gleichzeitig im Ordner „6-HTML-Preview“ abgelegt. Sie behält ihren ursprünglichen Dateinamen, ergänzt „HTML-Preview“ und als HTML-Datei gespeichert.

Im Browser sollten Sie folgende Ansicht vorfinden:

Vor den Toren der Stadt

Deutsch-ukrainische Forschungen in der Vorstadt von Olbia Pontike

Dr. Philip Brize
apl. Prof. Dr. Jochen Fornasier
Prof. ass. Dr. Eduard Shehi
Dr. Markus Helfert
M.A. Nathalie Kratzsch
PD Dr. Alla V. Bujskich

Journal Metadata

Journal Identifier: AA [journal-id-type=publisher-id]

Journal Title Group

Journal Title (Full): Archäologischer Anzeiger

Contributor Group [content-type=Editors]

Role or Function Title of Contributor: Editors:

Contributor [contrib-type=Editor]

Name of Person [name-style=western]

Surname: Fless

Given (First) Names: Friederike

Contributor Group

Contributor [contrib-type=Editor]

Name of Person [name-style=western]

Surname: von Rummel




Das HTML-Preview enthält eine einfache Vorschau der JATS-Inhalte, die insbesondere für die inhaltliche Prüfung vor dem Import in eine Lens-Viewer-Anwendung dienen kann.

Hinweis: Wenn Sie HTML-Previews erzeugen wollen, sollten Sie für eine komplette Ansicht der Inhalte auch den Bildordner des Artikels in den Ordner „6-HTML-Preview“. Nur dann werden auch die Bilder im Artikel im Browser angezeigt.

Output der Konvertierung und Übergabe für die Lens-Viewer-Applikation

Ist die JATS-Validierung erfolgreich und zeigt Ihre Prüfung des HTML-Preview keine Probleme auf, so können die erzeugten Daten in eine Lens-Viewer-Applikation importiert werden.

Nehmen Sie dazu aus dem Ordner „5-JATS-Konverter-Step3“ sowohl die erzeugte XML-Datei als auch den darin abgelegten Bild-Ordner und stellen Sie diesen dem Verantwortlichen für die Befüllung der Lens-Viewer-Applikation zur Verfügung:

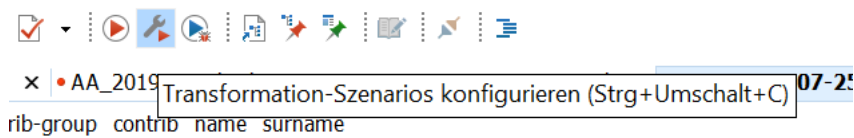
Name	Änderungsdatum
 AA_2019_1_Schachner_lens-web-resources	05.08.2019 09:38
 AA_2019_1_Schachner_2019-08-06-V1 2019-08-06.xml	06.08.2019 10:56
 aa-2-2019-07-25-de-en 2019-08-06.xml	06.08.2019 11:00

Bei Dateitransfer per Mail wird es sich wahrscheinlich anbieten, die Dateien vorher in eine ZIP-Datei zu verpacken und so zu übermitteln.

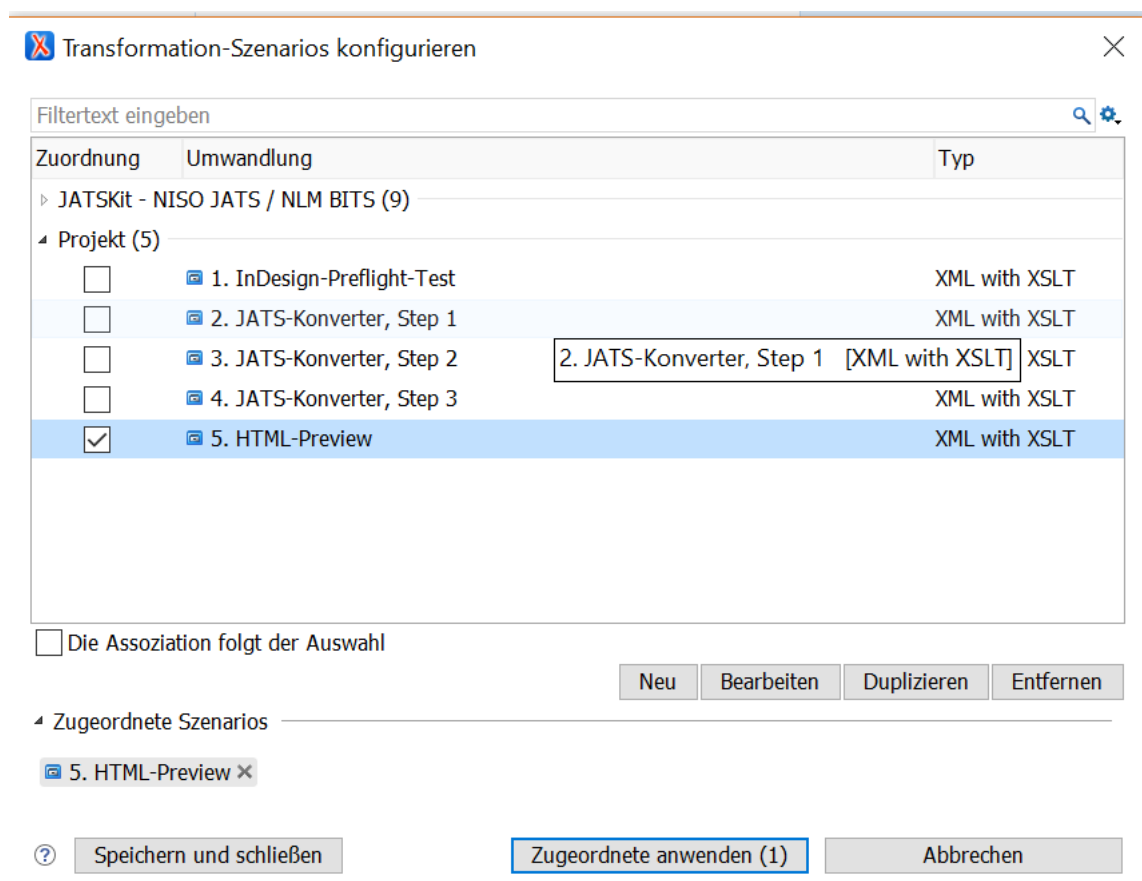
Arbeiten mit dem Menü „Transformations-Szenarios konfigurieren“

Alle Ordner im Projekt sind so konfiguriert, dass der jeweils nächste Transformations- bzw. Validierungs-Schritt bereits hinterlegt ist. Sie können die Produktionsstrecke insofern ausschließlich über die Buttons „Transformations-Szenario anwenden“ und „Validieren“ bedienen.

Sie haben aber über den Button „Transformations-Szenarios konfigurieren“ auch vollen Zugriff auf alle hinterlegten Transformationen:



Nach Klick auf den Button erhalten Sie folgenden Dialog:



Hier können alle hinterlegten Transformationen auch einzeln ausgewählt und manuell auf Dateien angewandt werden – auch wenn sich diese nicht in den Projekt-Ordern befinden sollten.

Technische Dokumentation

Im Projekt verwendeter JATS-Dialekt

Im vorliegenden Projekt wird die Journal Archiving and Interchange Tag Library NISO JATS mit dem Archiving Tagset („Green“) in der Version 1.2 verwendet. Die zur Validierung in dieser Ausprägung notwendigen Dateien sind im Projekt im Unterverzeichnis „_DTD“ hinterlegt und entsprechend in die Validierungs-Szenarien im Projekt eingebunden.

Die DTDs können bei Bedarf bzw. bei auch direkt bei der NISO unter folgender URL bezogen werden:

<https://jats.nlm.nih.gov/archiving/versions.html>

Das Projekt erhebt den Anspruch, stets vollständig valides JATS-XML zu erzeugen, wenn alle dazu notwendigen Konventionen bei Tagging und Export der zugrundeliegenden InDesign-Daten eingehalten wurden.

Besonderheiten der für das DAI erzeugten JATS-Ausprägung

Neben dem Anspruch vollständig valider XML-Daten war ein weiterer Anspruch im Projekt, die Daten soweit wie möglich nach den üblichen Konventionen für Standard-JATS-XML aufzubauen und im Detail danach zu strukturieren. Bei der Erzeugung der Daten haben wir uns soweit möglich an den in folgender Dokumentation niedergelegten Konventionen orientiert:

<https://jats.nlm.nih.gov/archiving/tag-library/1.2/>

Im Laufe des Projektes mussten jedoch für die Erfüllung der Anforderungen an die Darstellung und Funktionalität bzw. durch die Besonderheiten des Input-Formates folgende Besonderheiten eingeführt werden, die bei Verwendung des hier erzeugten JATS-XML in anderen Anwendungsfällen als der Erzeugung von Lens-Viewer-Applikationen wahrscheinlich beachtet werden müssen:

- **Tabellen und Formeln:** Im Projekt werden weder Tabellen noch mathematische Formeln verwendet. Der Konverter könnte grundsätzlich HTML-Tabellen verarbeiten, dies ist im Projekt jedoch nie praktisch getestet oder mit echten Daten durchgespielt worden. Auch wird im Projekt kein MathML verwendet.
- **Metadaten:** Nicht alle der im Projekt geforderten Journal- bzw. Artikel-Metadaten konnten mit dem Standard-JATS-Metadaten-Modell abgebildet werden. An einigen (wenigen) Stellen werden deswegen <custom-meta>-Elemente verwendet, die in je einer <custom-meta-group> in <journal-meta> und <article-meta> geklammert sind.
- **Absatz-Zahlen:** Für die geforderte Absatz-Zählung wurde das JATS-Element <named-content> verwendet, das für die Darstellung in Lens-Viewer-Anwendung gesondert angesprochen wird.
- **Styled-Content:** An einigen Stellen im Content bestanden Formatierungs-Anforderungen, die nicht alleine aufgrund von Standard-JATS-Elementen umsetzbar waren, z.B. in der Binnen-Struktur der Abstracts. An diesen Stellen verwenden wir <styled-content>-Elemente, die für das Layout im CSS der Lens-Viewer-Anwendung angesprochen werden.
- **Bilder im Artikel-Haupttext:** Aufgrund der Struktur des Indesign-Export-Formates ist die ursprüngliche Position der eingebundenen Bilder im originalen Satzbild nicht mehr erkennbar bzw. rekonstruierbar. Da die Bilder im Text zudem auch mehrfach referenzierbar sind, haben wir uns im Daten-Design entschieden, die Bilder nicht wie sonst üblich an der Stelle der Verwendung im Text zu platzieren. Die Bild-Container werden stattdessen in einem eigenen <sec>-Element

gesammelt, das über das @id-Attribut id="images-container" erkennbar ist und als letztes <sec>-Element im <body> der JATS-Datei steht. Für die Funktionalität der Lens-Viewer-Anwendung ist dies unerheblich. Sollten jedoch aus den JATS-Daten alternative Print-Layout erzeugt werden müssen, wäre die originale Positionierung der Bilder aus den Daten nicht mehr zu erkennen.

- **Referenzen:** Im Bereich der Literatur-Referenzen haben wir uns in Abstimmung mit dem DAI während dem Projekt für die Nutzung von <mixed-citation> entschieden. Die Verwendung von <element-citation> hätte zu erheblichen Mehraufwänden beim Tagging geführt, während die Vorteile dieses Taggings im konkreten Anwendungsfall kaum erkennbar waren. Innerhalb von <mixed-citation> werden keine semantischen Auszeichnungen verwendet, wir setzen nur in <ref> ein <label> mit der Kurzbezeichnung der zitierten Literatur, das gleichzeitig als Link-Anker für Literatur-Verweise dient.
- **Link-Typen:** Aufgrund der inhaltlichen Struktur der Daten müssen relativ viele verschiedene Link-Typen differenziert werden („normale“ Hyperlinks, aber auch Abbildungs-Verweise, Literatur-Referenzen, Supplements, Extra Features, etc.). Wir verwenden deswegen für Links durchgehend das JATS-Element <ext-link>, differenzieren dieses jedoch durch das @specific-use-Attribut, in dem der Link-Typ übergeben wird.

Eingesetzte Werkzeuge und Software-Komponenten

Die Produktionsstrecke besteht aus mehreren XSLT-Transformationen, die aus Gründen der besseren Verteilbarkeit im Rahmen eines Oxygen-Projektes übergeben werden. Für die Entwicklung bzw. die Ausführung aller Schritte sind folgende Komponenten notwendig:

- **XSLT:** Die XSLT-Skripte wurden mit XSLT 2 entwickelt; für einige Funktionen ist dies auch zwingend notwendig. Es werden jedoch keine eigenen Bibliotheken oder proprietäre Funktionen/Extensions verwendet.
- **Oxygen:** Als Entwicklungs-Umgebung für die Programmierung bzw. als Laufzeit-Umgebung für die Transformationen wird Oxygen XML Editor in der Version 20.1 verwendet.
- **XSLT-Prozessor:** Als XSLT-Prozessor wird Saxon in der Version Saxon HE 9.8 verwendet. Saxon kann auch unter folgender URL als lauffähige Java-Applikation bezogen werden: <https://www.saxonica.com/products/latest.xml>
- **XML-Parser:** Als XML-Parser wird Xerces in der Java-Version verwendet. Xerces kann auch unter folgender URL als lauffähige Java-Applikation bezogen werden: <https://xerces.apache.org/>

Sollte dies für weitere, zukünftige Anwendungs-Szenarien notwendig sein, können die XSLT-Skripte auch mit Java-/Kommandozeilen-Version von Saxon bzw. Xerces ausgeführt werden. So sind auch andere Produktionsstrecken-Funktionen, weitere Automatisierungen oder die Einbindung in CMS-Umgebungen möglich.

Aufbau und Arbeitsweise der InDesign-Preflight-Prüfung

Im Rahmen der InDesign-Preflight-Prüfung erfolgt eine Auswertung der Struktur der InDesign-Export-Datei. Dabei werden die bisher bekannten Fehlerquellen ausgewertet und in Form eines menschenlesbaren HTML-Reports ausgegeben.

XSLT-Skript: AA2JATS_CheckInDesignOutput.xsl

Input: InDesign-XHTML-Output

Output: Prüfbericht als menschenlesbarer HTML-Report mit eingebettetem Inline-CSS für das Browser-Layout

Arbeitsweise:

- In der Hauptfunktion des Skripts wird eine HTML-Tabellenstruktur geschrieben, die dann pro Test jeweils eine Zeile enthält
- Die Tabellen-Zeilen werden über die Dienstfunktion WriteReportRow geschrieben, der immer dieselben Parameter übergeben werden. Es gibt im Wesentlichen zwei Ausprägungen von Tests:
- Einfache Tests: Werden nur mit einer Funktion realisiert. Dabei wird mit einem Xpath-Ausdruck genau eine Zahl von Elementen mit einem bestimmten Kriterium ausgewertet, die über Bestehen/Nicht-Bestehen des Tests entscheidet. Der Auswertungs-Text wird direkt in der Funktion erzeugt.
- Komplexe Tests: Hier wird zunächst ein Paar von Dienstfunktionen aufgerufen, einmal für Count und einmal für Text des Tests. Die Count-Funktion wertet die Zahl problematischer Elemente aus, die Text-Funktion generiert den inhaltlichen Teil des Ausgabe-Textes. In der Regel werden komplexe Tests verwendet, wenn kombinierte Kriterien mit komplexem Xpath abgefragt werden müssen. Bei der Auswertung wird so viel Kontext wie möglich mit in die Rückgabe übernommen, um dem Bearbeiter bei der Fehlersuche zu helfen.
- Die Test-Zeilen sind für ihre Ergebnisse aufgeteilt in folgenden Typen: Bereich (bildet nur die Gliederung des Reports), Info (bestandene Tests oder reine Informationen statistischer Art), Warnung (Hinweise auf inhaltlich evtl. fehlerhafte Stellen oder Fehler, die nicht zu Konvertierungsfehlern führen sollten), Fehler (Alle fehlerhaften Stellen, die wahrscheinlich zu Fehlern in der Verarbeitung führen werden), Schwerer Fehler (Alle Fehler, die zu Folgefehlern an mehreren anderen Stellen des Reports führen). Die Fehlertypen werden im Layout visuell durch die Farben unterschieden.
- Grundsätzlich müssen alle Fehler und schweren Fehler beseitigt werden, bevor eine Chance auf valide Konvertierungsergebnisse besteht. Warnungen können ignoriert werden, wenn die zugrundeliegenden Strukturen inhaltlich richtig sind.

Das XSLT-Skript ist auch im Quellcode ausführlich mit Kommentaren versehen, insbesondere an funktional kritischen Stellen. Bitte konsultieren Sie dazu im Zweifelsfall auch den Quellcode der mitgelieferten XSLT-Datei.

Aufbau und Arbeitsweise von JATS-Konverter, Step 1

In Transformations-Stufe 1 erfolgt die Serialisierung des InDesign-Output in die notwendige Dokument-Reihenfolge für JATS

XSLT-Skript: AA2JATS_CreateDocumentOrder.xsl

Input: XHTML-Output von InDesign

Output: Zwischenformat mit JATS-Basis-Strukturen und Inhalten in korrekter Dokument-Reihenfolge

Arbeitsweise:

- In CreateDocumentOrder wird die Grundstruktur für die JATS-Datei erzeugt, insbesondere werden dabei die wichtigsten Abschnitte aus den verschiedenen Stellen der InDesign-Struktur in die richtige Reihenfolge gebracht und mit benannten Containern versehen, die eine Zuordnung in der Weiterverarbeitung erleichtern.
- Für den Frontmatter-Bereich werden ausgewertet und mit Element-Containern herausgeschrieben: Titel, Autoren, Abstract/Originalsprache, Abstract/Übersetzung, Keywords/Originalsprache, Keywords/Übersetzung
- Für den Bodymatter-Bereich wird der Haupttext-Fluss aus InDesign ausgewertet und zunächst unverändert übernommen. Allerdings werden bereits Überschriften ermittelt und als Header-Elemente erzeugt.
- Alle Bilder und ihre umgebenden div-Elemente werden am Ende des body gesammelt.
- Für den Backmatter-Bereich werden ausgewertet und mit Element-Containern herausgeschrieben: Fussnoten, Referenzen/Literaturverzeichnis, Abkürzungsverzeichnis/Glossar

Das XSLT-Skript ist auch im Quellcode ausführlich mit Kommentaren versehen, insbesondere an funktional kritischen Stellen. Bitte konsultieren Sie dazu im Zweifelsfall auch den Quellcode der mitgelieferten XSLT-Datei.

Aufbau und Arbeitsweise von JATS-Konverter, Step 2

In Transformations-Stufe 2 erfolgt die Erzeugung der Kapitel-Hierarchie und der Autoren-Container.

XSLT-Skript: AA2JATS_CreateDocumentStructures.xsl

Input: Zwischenformat mit JATS-Basis-Strukturen und Inhalten in korrekter Dokument-Reihenfolge

Output: Zwischenformat mit Kapitel-Hierarchien und Autoren-Containern

Arbeitsweise:

- An denjenigen Stellen, wo für die JATS-Struktur Hierarchien und Gruppierungen notwendig sind, die in den Ausgangsdaten nicht enthalten sind, erzeugen wir diese über for-each-groups und reichern die Daten so um Strukturen an. Das betrifft folgende Bereiche:
- Erzeugung von Sections für die JATS-Datei: Auf Basis der Elemente h1, h2, h3 werden die Überschriften ausgewertet, um per group-by die notwendigen verschachtelten sec-Elemente zu erzeugen.
- Erzeugung Container für Autoren/Contributors: Da in den Ausgangsdaten alle Metainfos zu allen Autoren in einer flachen Abfolge von p-Elementen stehen, verwenden wir den Absatz mit dem Autoren-Namen von author/co-author als Gruppierungs-Element, um einen contributor-Container je Person zu erzeugen.
- Neben diesen Struktur- und Hierarchie-Anpassungen bleibt die Datei ansonsten unverändert und wird über eine Identity-Transformation auf sich selbst abgebildet.

Das XSLT-Skript ist auch im Quellcode ausführlich mit Kommentaren versehen, insbesondere an funktional kritischen Stellen. Bitte konsultieren Sie dazu im Zweifelsfall auch den Quellcode der mitgelieferten XSLT-Datei.

Aufbau und Arbeitsweise von JATS-Konverter, Step 3

In Transformations-Stufe 3 erfolgt die Erzeugung des finalen JATS-Output

XSLT-Skript: AA2JATS_CreateJATS.xsl

Input: Zwischenformat mit JATS-Basis-Strukturen und Body-Text mit Kapitel-Hierarchien

Output: Endgültiger JATS-Output

Arbeitsweise:

- Auf Basis der in Step 1 und Step 2 erzeugten XML-Zwischenformate wird in der letzten Transformations-Stufe der finale JATS-XML-Output erzeugt.
- Im Dokument-Body sind alle Elemente und Strukturen bereits in der korrekten Reihenfolge. Für die Befüllung der JATS-Inhaltsmodelle werden im wesentlichen Element-Handler in Form von Match-Templates verwendet, die alle XHTML-Strukturen in ihre JATS-Äquivalente konvertieren. Lediglich zum Erzeugen von IDs und Verlinkungen werden in diesem Bereich Named-Templates gerufen, die als Dienst-Funktionen verwendet werden.
- Für die Erzeugung von Frontmatter und Backmatter werden die XML-Strukturen im Wesentlichen durch Named-Templates top/down erzeugt, denn hier spielen Inhaltsmodelle und Abfolge der Elemente eine besondere Rolle. Das gilt vor allem für die Erzeugung und Zuordnung des komplexen Metadaten-Modells für die Article-Metadaten, Autoren-Angaben, Abstracts und Keywords.
- Sprach-Steuerung: Eine besondere Rolle in Step 3 spielt die Sprach-Steuerung. Auf Basis der 'title-' Absatz-Formate wird die Dokumentsprache erkannt und mit den Sprachangaben von Abstract und Abstract-Übersetzungen verglichen, damit die korrekten Elemente und xml:lang-Attribute erzeugt werden können. Dies ist notwendig, weil auf xml:lang einzelne Funktionen der Lens-Applikation beruhen. Gleichzeitig wird auf Basis der Sprache entschieden, welches Prefix für die Abbildungs-Bezeichner verwendet wird ('Abb.', 'Fig.' o.ä.), denn auch hier wird an einigen Stellen des Konverters der Abbildungs-Bezeichner dynamisch abhängig von der Sprache gesetzt und verwendet.
- Verlinkung: An vielen Stellen werden IDs und Verweise für die Quer-Referenzierung von Datenstrukturen geschrieben, dies erfolgt komplett über dedizierte Named-Templates. Dies gilt insbesondere für: Bezüge Abbildungen/Abbildungsverweise/Quellen-Nachweise, Fussnoten/Fussnoten-Texte, Referenz-Verweise/Referenzen - aber auch für die verschiedenen Hyperlink-Typen (normale Hyperlinks, Zenon-Links, Supplements, Extra Features)
- Fehler-Toleranz: An vielen Stellen verlässt sich Step 3 darauf, dass inhaltlich problematische Datenstrukturen auf Basis der InDesign-Preflight-Checks behoben wurden (z.B. nicht verknüpfbare Referenz-Verweise) und versucht NICHT, Daten hier noch zu retten. Im Zweifelsfall lassen wir lieber Validierungsfehler als letzte Instanz geschehen, damit Fehler mit inhaltlichen Folgen nicht unbemerkt aus den Daten verschwinden.

Das XSLT-Skript ist auch im Quellcode ausführlich mit Kommentaren versehen, insbesondere an funktional kritischen Stellen. Bitte konsultieren Sie dazu im Zweifelsfall auch den Quellcode der mitgelieferten XSLT-Datei.

XSLT-Transformation für HTML-Preview

Die XSLT-Transformation für die Generierung des HTML-Preview stammt aus dem NISO JATSKit Projekt und wird nur aus Service-Gründen mitgeliefert. Für Funktionalität und Verwendung kann seitens digital publishing competence kein Support und keine Haftung übernommen werden. Sie finden das Projekt unter folgender URL: <https://github.com/wendellpiez/JATSKit> bzw. die Projektbeschreibung unter folgender URL: <https://www.ncbi.nlm.nih.gov/books/NBK350379/>

Für das Arbeiten mit JATS-Daten in Oxygen wird die Verwendung des JATSKit angeraten, denn hier sind über die Generierung von HTML-Previews hinaus noch wesentliche weitere Tools realisiert, die das Arbeiten mit JATS-XML deutlich erleichtern.

Weitere Dokumentationen

Neben dieser Dokumentation wird weiterhin verwiesen auf:

- **Change-Log der Produktionsstrecke JATS-XML:**
Hier werden die Änderungen der Versionen im Laufe der Entwicklung dokumentiert.
- **AA Tagging Bibliothek:**
Die Datei enthält eine Referenz aller Formate im verwendeten InDesign-Template, die vom JATS-Konverter unterstützt werden, sowie eine Dokumentation, welche Formate in welche JATS-Strukturen umgesetzt werden.

Beide Dokumentationen liegen der Produktionsstrecke ebenfalls bei. Für die Inline-Dokumentation der XSLT-Skripte konsultieren Sie bitte den Quell-Code der XSLT-Skripte im Ordner „_XSLT“.