

Frameworks – MinRProt

Übersicht über die Frameworks für den Oxygen XML Editor, die zur Bearbeitung der „Protokolle des Bayerischen Ministerrats 1945-1962“ benutzt werden

Inhalt

- [Allgemein](#)
- [Frameworkübergreifend](#)
- [TEI-Oxydation: MinRProt – Edition](#)
- [TEI-Oxydation: MinRProt – Register](#)
- [TEI-Oxydation: MinRProt – Gesamtdateien](#)

Allgemein

Dem Projekt *MinRProt* sind drei Frameworks zugeordnet. Sie sind jeweils auf einen anderen Teil der Editionsdateien ausgerichtet und bietet für diesen eine spezialisierte Arbeitsumgebung im Oxygen-Author-Modus mit angepassten Funktionalitäten. Die Zuordnung geschieht über Verknüpfungsregeln, die im Framework selbst konfiguriert werden. Bei *MinRProt* findet die Unterscheidung anhand der Dateinamen statt. Außerdem werden prinzipiell nur Dateien berücksichtigt, die den TEI-Namensraum benutzen.

Jedes Framework ist als eigener Ordner angelegt. Alle Komponenten des Frameworks sind darin in Unterordnern gespeichert, die Konfigurationen in der *.framework*-Datei. Diese können entweder im Punkt „Dokumententypzuordnung“ der Oxygen-Einstellung bearbeitet werden oder indem man die *.frameworks*-Datei, die auf XML basiert, direkt öffnet und ediert.

Frameworkübergreifend

Im Folgenden werden die Komponenten beschrieben, die in allen Frameworks vorkommen. Frameworkspezifische Änderungen und Ergänzungen werden in den jeweiligen Einzelbeschreibungen aufgeführt.

Icons

Teilordner: /icons

Um das Aussehen und die Benutzbarkeit der Arbeitsumgebung zu verbessern, wurden den Aktionen in der Menüdarstellung Icons zugeordnet. Dabei hat nicht jede Aktion ein solches Icon. Teilweise wurde bei Aktionen, die in einer zusammenhängenden Reihe stehen, nur der obersten von ihnen ein Icon beigelegt (als eine Art Überschrift).

Die Icons stammen von [FlatIcon](#). Es handelt sich ausschließlich um kostenlos angebotene Icons. Für den Nachweis der Urheber siehe [Lizenznachweis](#).

Operationen & Aktionen

Teilordner: /operations

Aktionen sind auslösbare Funktionen, die das im Author-Modus geöffnete XML-Dokument auslesen, durchsuchen und verändern können. Mit ihnen lassen sich bestimmte Arbeitsschritte unterstützen oder automatisieren. Die Benutzer*innen können sie aktivieren, indem sie einem der Menüs auf den entsprechenden Aktionsnamen klicken (Hauptmenü, Symbolleiste,

Kontextmenü). Innerhalb der Menüs sind Aktionen in mehrere Aktionsgruppen einsortiert. Einige, weniger häufig benutzte Aktionen können nur in der Symbolleiste angewählt werden. Für die meisten Aktionen steht außerdem eine Tastenkombination zur Verfügung um sie direkt auszulösen.

Aktionen werden eingerichtet, indem vorgegeben Kleinstprogramme, die **Operationen**, innerhalb des Frameworks für den konkreten Anwendungszweck konfiguriert werden. Einige der Aktionen in *MinRProt* basieren auf den in Oxygen enthaltenen Standardoperationen. Für die meisten aber wurde mittels der Java-Schnittstelle von Oxygen eine neue Operation programmiert. Diese Operationen werden als Paket „sdf.jar“ aus Java exportiert. Um sie in das Framework zu integrieren, muss dieses Paket in seinen Ordner kopiert und seinen Klassenpfad eingetragen werden.

Eine vollständige Dokumentation aller Aktionen findet sich [Funktionsbeschreibungen](#). Dort befindet sich auch eine Übersicht über die Tastenkombinationen.

Stile

Teilordner: /css

Im Oxygen-Author-Modus kann die Darstellung der XML-Dokumente auf vielfältige Weise angepasst werden. Dies geschieht mittels CSS-Stylesheets. Diese werden über **Stile** im Framework eingerichtet. Dabei ist es möglich in einem Framework mehrere verschiedene Stile anzulegen, die die Benutzer*innen dann nach Belieben auswählen und wechseln können.

Sofern nicht anders angegeben, verfügt ein Framework nur über einen Stil, der bei ihm immer aktiv ist.

Transformationen

Teilordner: /transformations

Die automatisierte Weiterverarbeitung von XML-Dateien geschieht über XSL-Stylesheets. Über **Transformationsszenarien**, die innerhalb der Frameworks angelegt sind, können sie aufgerufen und auf das aktuell geöffnete XML-Dokument angewandt werden. Die Transformationsszenarien können über die gleichnamigen Aktionen in den „Transformation“-Menüs ausgelöst werden. Alternativ stehen hierfür die Transformationsschaltflächen in der Symbolleiste zur Verfügung. Haupteinsatzzweck der Transformationen in *MinRProt* ist es mittels des Apache XSL-Formating-Objects Prozessors (Apache FOP) aus XML-Dateien PDFs zu erzeugen. Hierfür wird im [Oxygen-Projekt](#) ein externer Apache FOP 2.5 eingerichtet. Der Prozessor und weitere benötigte Bibliotheken befinden sich im Unterordner „Tools“ des Projektordners. Als Transformator ist „Saxon-HE 9.6.0.7“ eingestellt.

Für die PDF-Umwandlungen wird die (proprietäre) Schriftart „Adobe Garamond PHK“ benutzt, die auch für den finalen Druck zum Einsatz kommt. Ohne diese Schriftart kann das vorgesehene Druckbild nicht genau repliziert werden. Als Ersatz benutzt der Apache FOP standardmäßig die Schriftart „Times New Roman“. Die zu verwendenden Fonts kann in den verschiedenen .xsl/-Dateien angepasst werden unter dem Punkt „Schriftarten“ am Anfang der Stylesheets.

Die Ergebnisse der Transformationen werden in der Regel in Oxygen oder einem geeigneten PDF-Viewer geöffnet. Die erzeugte Dateien wird, falls nicht anders angegeben, am Ort der Ursprungsdatei im Unterordner „Transformationen“ gespeichert.

TEI-Oxydation: MinRProt – Edition (tei_oxydation_minrprot)

Beschreibung & Verknüpfungsregeln

Das Editionsframework findet Anwendung für alle Dateien, die einen Teil des Editionstextes darstellen. Dies sind die Protokolle und die Editionsabschnitte.

Das Framework wird auf alle Dateien angewandt, deren Namen mit einem bei MinRProt benutzten Bandkürzel beginnen („ehr“, „hoe“, „scf“).

Stile

Teilordner: /css

Das Framework bietet mehrere alternative Anzeigestile an, die jeweils einen bestimmten Arbeitsschritt unterstützen (\$Funktionsbeschreibung).

Jeder dieser Stile basiert auf einem eigenen CSS-Stylesheet, den *Modulen* (z.B. „MinRProt_Modul_Ansicht.css“). Alle *Modul*-Stylesheets importieren als gemeinsame Grundlage die Datei „MinRProt_Basis_TEI.css“. Diese beinhaltet alle Regeln, die immer gelten sollen, unabhängig vom gewählten Stil. Um die Übersicht zu verbessern wurden Regeln, die nur bestimmte Dokumententypen (Protokolle oder Editionsabschnitte) betreffen in getrennte *Basismodule* ausgelagert. Die *Basismodule* werden von „MinRProt_Basis_TEI.css“ importiert und an die *Module* weitergegeben.

Vorlagen

Teilordner: /templates

Vorlagedateien dienen als Grundlage zum Erstellen neuer Editionsdateien. Bei *MinRProt* werden sie nur für Protokolle benutzt. Bei Editionsabschnitten lohnt sich aufgrund der geringen Anzahl Dateien und deren heterogenen Aufbaus die Erstellung von Vorlagen nicht.

Protokollvorlagen sind bandspezifisch. Sie enthalten die Grundstruktur einer Protokolldatei und einen vorausgefüllten <teiHeader> mit allen Daten für den jeweiligen Editionsband.

Beim Anlegen einer neuen Protokolldatei kann die entsprechende Vorlage im „Neue Datei“-Dialogfenster unter dem Punkt „Frameworkvorlagen“ > „TEI-Oxydation: MinRProt – Edition“ ausgewählt werden.

Transformationen

Teilordner: /transformations

Das Editionsframework bietet drei Varianten an, um eine XML-Datei in ein PDF umzuwandeln. Die Szenarien „PDF - Ansicht“ (basiert auf „minrprot_ansicht_pdf.xml“), „PDF - Korrektur“ („minrprot_korrektur_pdf.xml“) und „PDF - Einzeldruckvorschau“ („mintprot_einzeldruck_pdf.xml“), stellen jeweils aus einer einzelnen Datei ein PDF her. Die *Einzeldruckvorschau* ist dabei dem späteren Druckbild nachempfunden. Das bedeutet unter anderem, dass die Seitengröße auf die Fläche der späteren Buchseite reduziert ist. *Ansicht* und *Korrektur* nutzen hingegen das volle A4-Blatt, wobei *Korrektur* größere Abstände am Seitenrand und zwischen den Zeilen lässt.

Das Szenario „XML – PTVZ – Zusammenstellung“ liest alle Protokolldateien aus und erzeugt daraus automatisch das „Verzeichnis der Protokolle und Tagesordnungspunkte“. Dafür kann die Transformation auf eine beliebige Protokolldatei eines Bandes angewandt werden. Die erzeugte Datei wird als „PTVZ“ im entsprechenden „Protokolle-Ordner“ gespeichert werden. Da die Umwandlung auch bandspezifische Metadaten erzeugt, beinhaltet sie eine Fallunterscheidung, die mit jedem neuen Band ergänzt werden muss.

Das Stylesheet „minrprot_druck_pdf.xsl im Unterordner „Druck“ ist keinem eigenen Transformationsszenario zugeordnet. Stattdessen erweiterte es das importierte „minrprot_ansicht_pdf.xsl“ um Anweisungen für die Umwandlung zur Druckversion, die sowohl von der *Einzeldruckvorschau*, als auch von dem Stylesheet „minrprot_gesamt_druck.xsl“ im Gesamtdatei-Framework benötigt und importiert werden (siehe unten).

TEI-Oxydation: MinRProt – Register (tei_oxydation_minrprot_register)

Beschreibung & Verknüpfung

Das Registerframework findet Anwendung für alle Dateien die Teil der Registerdatenbank (RDB) sind. Außerdem wird das Framework für das *Rohregister* angewandt, das im Zuge der [Drucklegung](#) entsteht.

Das Framework gilt für alle XML-Dateien, deren Namen mit „minrprotRDB“ beginnt oder die Zeichenfolge „rohregister“ oder enthalten.

Stile

Die Darstellung der RDB-Dateien wird von dem gemeinsamen Stil „Registerdatenbank“ (Basierend auf „MinRProt_Register.css“) gesteuert. Optional kann der Stile „Kompaktansicht“ hinzugeschaltet werden („MinRProt_Register_kompakt.css“) hinzugeschaltet werden. Dieser erzeugt eine zusammengeschrumpfte Darstellung, in der die Einträge enger zusammen stehen und alle Daten außer den Registerbegriffen selbst ausgeblendet sind.

Transformationen

Das Registerframework bietet ein einheitliches Transformationsszenario „PDF Ansicht“ für alle RDB-Varianten an. Die erzeugte PDF-Version ist auf die Seitengröße A4 angepasst und orientiert sich nicht am Drucklayout. Das *Ansichts*-PDF enthält keine Seitenzahlen.

Das Transformationsszenario „PDF - Rohregister“ ist dafür vorgesehen, eine *Rohregister* -Datei in ein PDF umzuwandeln, um sie so besser ausdruck- und korrigierbar zu machen.

TEI-Oxydation: MinRProt – Gesamtdateien (tei_oxydation_minrprot_gesamt)

Beschreibung & Verknüpfung

Das Gesamtframework findet Anwendung für die *Gesamtdatei*, die die Grundlage für die Drucklegung des Editionsbands ist. Mittels ihr werden die relevanten Einzeldateien des Bandes gesammelt über mehrere Schritte hinweg in ein fertiges *Druck-PDF* verwandelt (siehe [Drucklegung](#)).

Das Framework wird auf alle XML-Dateien angewandt, deren Namen ein Bandkürzel und den Textausschnitt „gesamt“ enthalten (z.B. „erh33_gesamt_druckregister.xml“).

Operationen & Aktionen

Das Framework enthält zwar das „sdf.jar“-Paket mit den selbst programmierten Operationen. Derzeit sind für es keine Aktionen eingerichtet, die auf diesen Operationen beruhen.

Transformationen

Das Gesamtframework enthält mehrere Transformationszenarien, die die Gesamtdatei jeweils auf unterschiedliche Weise weiterverarbeiten. Die Szenarien „Gesamtdatei – Druck – PDF“ und „Gesamtdatei – Druck – XSL“ beruhen beide auf „minrprot_gesamt_druck_pdf.xsl“. Ersteres erzeugt ein *Gesamt-PDF*, das sowohl zu Korrekturzwecken dient, als auch die finale Druckversion darstellen kann. Letzteres gibt das rohe XSL zurück, bevor die PDF-Umwandlung mittels Apache-FOP darauf ausgeführt wurde. Dieses Format kann zu Debuggingzwecken herangezogen werden.

Beide Szenarien importieren das Stylesheet „minrprot_druck_pdf.xsl“ aus dem *Editionsframework*. Die Umwandlung funktioniert deshalb nur, wenn auch dieses Framework in Oxygen installiert ist.

Außerdem wird die Datei „minrprot_gesamt_integration_durchfuehrung.xsl“ importiert. Dieses Stylesheet dient dazu vor Beginn der Umwandlung alle in der Gesamtdatei referenzierten Einzeldateien und RDB-Inhalte zusammen zu kopieren. Mit dem Szenario „Gesamtdatei – Integration – XML“ kann dieser Arbeitsschritt auch ausgeführt werden, ohne anschließend eine PDF-Umwandlung auszulösen. So entsteht eine XML-Datei, die den vollständigen Text der Edition enthält. In dieser können, z.B. zu Korrekturzwecken, bandübergreifende Such- und XPath-Anfragen ausgeführt werden. Das Ergebnis der Umwandlung wird als „BANDKÜRZEL_gesamt_integriert.xml“ gespeichert.

Maximilian Schrott [zuletzt geändert: 16.12.2020]