

Fachhochschule Potsdam
Fachbereich Informationswissenschaften

Entwicklung einer
Open-Source-Archivsoftware
für kleine und mittlere Archive



Diplomarbeit
zur Erlangung des Titels Diplom-Archivar (FH)

Gutachter:

1. Prof. Dr. Hartwig Walberg
2. Prof. Dr. Günther Neher

vorgelegt von:

Stefan Jentsch
Brassenweg 9
26127 Oldenburg
stefanjentsch@gmx.de

27. Juni 2006

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt habe. Andere als die angegebenen Hilfsmittel und Quellen wurden nicht genutzt und wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Oldenburg, 27. Juni 2006

Stefan Jentsch

Inhaltsverzeichnis

Vorwort.....	09
Danksagung.....	10
1. Einleitung.....	11
2. Voraussetzungen für die Programmierung einer Archivsoftware.....	12
2.1. Empfehlungen der Fachgremien für die EDV-gestützte Erschließung in Archiven	12
2.2. Archivsoftware.....	14
2.2.1. Ariadne.....	14
2.2.2. AUGIAS-Express 3.....	16
2.2.3. FAUST Entry Archiv	18
2.2.4. MidosaxML.....	19
2.2.5. Fazit.....	21
2.3. Freie Software vs. Proprietäre Software.....	22
2.3.1. Proprietäre Software.....	22
2.3.2. Freie Software und Open Source.....	23
2.3.3. Fazit.....	24
2.4. Datenorganisation.....	25
2.4.1. Datenstruktur.....	25
2.4.1.1. XML.....	26
2.4.1.2. Verarbeitung von XML-Dokumenten.....	28
2.4.2. Datenspeicherung.....	38
2.5. Auswahl und Vorstellung der Programmiersprache.....	30
2.5.1. Auswahl der Programmiersprache.....	30
2.5.2. PHP 5.....	32
2.5.2.1. Entwicklung, Einsatzmöglichkeiten und Vorteile von PHP.....	32
2.5.2.2. PhpGtk.....	33
2.6. Fazit.....	34
3. Programmierung einer Open-Source-Archivsoftware.....	36
3.1. Aufbau des grafischen Systems und Abfragen der Zugangsberechtigungen.....	37
3.2. Aufbau des Tektonikbaumes	41
3.3. Arbeiten in der Tektonik.....	45
3.4. Die Findmittel.....	48
3.5. Die Menüleiste.....	51

4. Resümee und weitere Pläne.....	53
5. Quellen.....	54
5.1. Gedruckte Quellen.....	54
5.2. Online-Quellen.....	55
Anhänge.....	56
A – Installation und Benutzung	56
A.1. CD-Inhalt	56
A.2. Installationsanweisungen	56
A.3. Bedienungsanleitung.....	57
B – Vergleich der DTDs.....	60
B.1. ArchivDTD.dtd vs. AsterArchiv.dtd.....	60
B.2. AF-Findmittel.dtd vs. AsterFind.dtd.....	65
C – Quellcode.....	75
C.1. AsterVE.php.....	75
C.2. attributes.inc.php.....	76
C.2.1. attributgrundbuch.....	76
C.3. elementsarchiv.inc.php.....	77
C.3.1. archivelementaltuebform.....	77
C.3.2. archivelementanschrift.....	78
C.3.3. archivelementarchivname.....	78
C.3.4. archivelementarchref.....	78
C.3.5. archivelementaufloesung.....	80
C.3.6. archivelementbem.....	80
C.3.7. archivelementbibliographie.....	81
C.3.8. archivelementbibref.....	82
C.3.9. archivelementemail.....	83
C.3.10. archivelementgruendung.....	83
C.3.11. archivelementfax.....	83
C.3.12. archivelementfindmitteldatei.....	83
C.3.13. archivelementfmref.....	84
C.3.14. archivelementhomepage.....	85
C.3.15. archivelementkompetenz.....	85
C.3.16. archivelementland.....	85

C.3.17. archivelementtektbeschreibung.....	86
C.3.18. archivelementtektnr.....	86
C.3.19. archivelementtekttitel.....	86
C.3.20. archivelementtelefon.....	87
C.3.21. archivelementverweise.....	87
C.4. elementsfm.inc.php.....	87
C.4.1. elementabgstelle.....	88
C.4.2. elementabkuerzungen.....	89
C.4.3. elementakzession.....	90
C.4.4. elementaltsig.....	92
C.4.5. elementaltuebform.....	92
C.4.6. elementanlass.....	94
C.4.7. elementanschrift.....	94
C.4.8. elementarchref.....	95
C.4.9. elementauflage.....	97
C.4.10. elementausfuehrung.....	97
C.4.11. elementaz.....	97
C.4.12. elementband.....	98
C.4.13. elementbearbeiter.....	98
C.4.14. elementbem.....	99
C.4.15. elementbeschreibung.....	100
C.4.16. elementbestandabgstelle.....	101
C.4.17. elementbestandakzession.....	102
C.4.18. elementbestandkurz.....	103
C.4.19. elementbestandprovenienz.....	103
C.4.20. elementbestandsig.....	104
C.4.21. elementbestandsname.....	104
C.4.22. elementbestellsig.....	105
C.4.23. elementbibliographie.....	105
C.4.24. elementbibref.....	106
C.4.25. elementblatt.....	107
C.4.26. elementeigentuemer.....	107
C.4.27. elementeinzeichnung.....	108

C.4.28. elementthaelt.....	109
C.4.29. elementstehungsstufe.....	111
C.4.30. elementfarbe.....	111
C.4.31. elementfmname.....	111
C.4.32. elementfmref.....	112
C.4.33. elementfmsig.....	112
C.4.34. elementformat.....	113
C.4.35. elementgemarkung.....	113
C.4.36. elementhilfsfeld.....	113
C.4.37. elementinstitution.....	114
C.4.38. elementkartentyp.....	116
C.4.39. elementklassnr.....	117
C.4.40. elementklasstitel.....	117
C.4.41. elementkonkordanz.....	117
C.4.42. elementlagerung.....	119
C.4.43. elementlaufzeit.....	120
C.4.44. elementlzanfang.....	121
C.4.45. elementlrende.....	121
C.4.46. elementlztext.....	121
C.4.47. elementmassstab.....	122
C.4.48. elementmaterial.....	122
C.4.49. elementnebenkarten.....	123
C.4.50. elementort.....	123
C.4.51. elementperson.....	125
C.4.52. elementprovenienz.....	127
C.4.53. elementrechtsstatus.....	128
C.4.54. elementseite.....	128
C.4.55. elementserientitel.....	128
C.4.56. elementsignatur.....	129
C.4.57. elementsperrvermerk.....	129
C.4.58. elementtext.....	131
C.4.59. elementtitel.....	132
C.4.60. elementtopogrdaten.....	135

C.4.61. elementumfang.....	137
C.4.62. elementverweise.....	137
C.4.63. elementvorgaenge.....	137
C.4.64. elementvorgfoto.....	138
C.4.65. elementvorgkarte.....	142
C.4.66. elementvorgsachakte.....	147
C.4.67. elementvorgurkunde.....	149
C.4.68. elementvorprov.....	150
C.4.69. elementzustand.....	151
C.5. functions.inc.php.....	152
C.5.1. addind.....	152
C.5.2. calend.....	152
C.5.3. cell_edited.....	153
C.5.4. cleartree.....	153
C.5.5. clickedTree.....	153
C.5.6. createframe.....	153
C.5.7. doPopup.....	154
C.5.8. errorhandle.....	154
C.5.9. get_tags.....	154
C.5.10. getChild.....	154
C.5.11. getids.....	155
C.5.12. hasChild.....	160
C.5.13. idattr.....	160
C.5.14. load_xml.....	161
C.5.15. login.....	161
C.5.16. model_overview.....	162
C.5.17. model_ve.....	162
C.5.18. new_konkord.....	163
C.5.19. onHelp.....	169
C.5.20. onQuit.....	170
C.5.21. saveFile.....	170
C.5.22. selectOVTreeRow.....	170
C.5.23. selectVETreeRow.....	171

C.5.24. set_model_iterbest.....	171
C.5.25. set_model_iterve.....	174
C.5.26. tag_exists.....	179
C.6. masksarchiv.inc.php.....	179
C.6.1. mask_archivinfo.....	179
C.6.2. mask_bestand.....	181
C.6.3. mask_findmittel.....	181
C.6.4. mask_tektonik.....	181
C.6.5. mask_used.....	182
C.7. masksfm.inc.php.....	183
C.7.1. mask_band.....	183
C.7.2. mask_bestandsinfo.....	183
C.7.3. mask_einleitung.....	185
C.7.4. mask_fall.....	186
C.7.5. mask_film.....	186
C.7.6. mask_findinfo.....	186
C.7.7. mask_foto.....	187
C.7.8. mask_grundbuch.....	190
C.7.9. mask_indizes.....	192
C.7.10. mask_index.....	192
C.7.11. mask_karte.....	192
C.7.12. mask_klassi.....	195
C.7.13. mask_konkordanz.....	196
C.7.14. mask_sach.....	196
C.7.15. mask_siegel.....	198
C.7.16. mask_urkunde.....	198

Vorwort

Die Idee zur Entwicklung einer eigenen Archivsoftware kam mir erstmalig während meiner Praktika im Rahmen meines Studiums an der Fachhochschule Potsdam.

Keines der Programme, mit denen ich dabei arbeitete, konnte meinen Ansprüchen genügen, oder mich vollständig zufrieden stellen.

Verstärkt wurde dieses Empfinden durch ein Projekt, das ich im Stadtarchiv Westerstede leitete. Dort wird das Archiv von drei Verwaltungsfachkräften und den ehrenamtlichen Helfern des Fördervereins betreut, von denen keiner fundiertes Wissen über das Archivwesen aufweisen kann.

Während des Projekts stand es auch an, eine funktionstüchtige Archivsoftware einzuführen und die Mitarbeiter mit den Arbeitsweisen und Aufgaben in einem Archiv vertraut zu machen. Dabei stellte es sich schnell heraus, dass die meisten EDV-Programme, die für einen Einsatz im Archiv in Frage kommen, entweder zu unübersichtlich oder zu teuer sind.¹

So kristallisierte sich immer mehr der Gedanke heraus, das es doch möglich sein müsste, eine Archivsoftware zu programmieren, die den vollen benötigten Funktionsumfang bietet, benutzerfreundlich ist,² auf einem offenen Quelltext basiert, damit es auf die einzelnen Gegebenheiten in einem Archiv schnell und einfach angepasst werden kann, und dazu möglichst kostenlos oder zumindest für wenig Geld erhältlich ist.³

Durch diese ganze Geschichte ist das Projekt natürlich auch ein sehr persönliches Projekt! Dies ist aber sehr wichtig, da damit gewährleistet ist, dass die vorliegende Diplomarbeit nicht das Ende bedeutet, sondern den Anfang!

¹ Die Wahl fiel letztendlich auf MidosaXML, da es kostengünstig und weitestgehend benutzerfreundlich ist. Vom Funktionsumfang her genügt es zumindest den Grundanforderungen in einem kleinen Archiv.

² Dieser Punkt ist sehr wichtig, da das Programm vor allem in kleinen und mittleren Archiven eingesetzt werden soll, wo durchaus auch davon auszugehen ist, dass das Archiv, wie in Westerstede, nur von Laien geführt wird.

³ Natürlich gibt es schon eine Open-Source-Archivsoftware, die vom Archivverbund Mecklenburg-Vorpommern entwickelt wurde und die quelltextoffen und kostenlos ist. Das war es dann aber auch schon mit den Vorteilen. Siehe auch: Kapitel 1.2.1.

Danksagung

An dieser Stelle möchte ich mich auch bei

- Dr. Peter Worm vom Westfälischen Archivamt in Münster,
- Dr. Ulrich Fischer vom Stadtarchiv Köln und
- Dr. Stefan Przigoda vom Deutschen Bergbauarchiv in Bochum

für ihre freundliche Unterstützung bei technischen Fragen, der Bereitstellung der Beispielfindmittel und ihrem großen Interesse an diesem Projekt herzlichst bedanken.

Stefan Jentsch

1. Einleitung

Wie bereits erwähnt, ist es Ziel, eine Software zu entwickeln, die von ihrem Funktionsumfang her mit kommerziellen Angeboten Schritt halten kann, benutzerfreundlich, flexibel erweiterbar und quelltextoffen ist. Damit soll den Archiven die Möglichkeit gegeben werden, selbst bei der Entstehung von Software für ihren Arbeitsbereich tätig zu werden, indem sie Gedanken, Anregungen und Vorschläge oder sogar Programmteile beisteuern.

Um die erwähnte Benutzerfreundlichkeit zu gewährleisten, wurde beschlossen, dass es nicht ein Programm mit allen Funktionen geben soll, sondern mehrere kleine Spezialanwendungen, die als Module kombiniert werden können. Daher leitet sich auch der Name ab, unter dem das Projekt läuft: Archivische Software-Tools für Erschließung und Recherche, kurz: **ASTER**.

Im zeitlichen Rahmen einer Diplomarbeit ist es nicht möglich, das Konzept für eine komplette Software entwickeln, geschweige denn, diese zu programmieren. In diesem Sinne stellt die vorliegende Arbeit einen ersten Arbeitsschritt dar, indem zuerst die ganze Thematik von der fachlichen und technischen theoretischen Seite betrachtet werden soll, und dann anhand des Verzeichnungsmoduls erste Versuche in Richtung praktische Umsetzung durchgeführt werden.

Zu den im 4. Kapitel aufgeführten Quellen sei gesagt, dass sie nur eine Querschnittsmenge darstellen und sich hauptsächlich um spezielle technische oder fachliche Aspekte im Zusammenhang mit dieser Arbeit drehen, da das Angebot an Literatur – sowohl als Buch oder auch im Internet – zu den meisten hier behandelten Themen schier unüberschaubar ist, und es somit besser dem Leser überlassen bleibt, ein Buch nach seinen Wünschen zur weiteren Vertiefung heranzuziehen.⁴

⁴ Z.B. gibt es für PHP Tausende Bücher, die sich entweder an Einsteiger, Semiprofessionals oder professionelle Programmierer richten. Hier ist es besser, der Leser wählt selber ein Buch, das seinen eigenen Kenntnissen am besten entspricht, als dass eines empfohlen wird, das für die einen „zu leicht“ und für die anderen „zu schwer“ ist.

2. Voraussetzungen für die Programmierung einer Archivsoftware

2.1. Empfehlungen der Fachgremien für die EDV-gestützte Erschließung in Archiven

Wenn man sich ein wenig in der Fachpresse umschaute, erhält man den Eindruck, dass das Thema „EDV in Archiven“ kaum von Belang ist.

1993 wurden die „Empfehlungen für den EDV-Einsatz in Kommunalarchiven“ von der Bundeskonferenz der Kommunalarchive (BKK) im „Archivar“ veröffentlicht,⁵ und ein Jahr darauf ein Kriterienkatalog des EDV-Ausschusses der Archivreferentenkonferenz (ARK) des Bundes und der Länder für EDV-Programme zur Archivalienerschließung⁶. Abgesehen von diesen beiden Texten finden sich lediglich vereinzelte Vorstellungen von in der Praxis eingesetzten Programmen und sehr viele Artikel zum Thema „Archive und Internet“. Für die vorliegende Arbeit sind aber lediglich die zwei oben genannten Texte von Interesse. Beide wurden zwar unabhängig von einander entwickelt, stimmen aber in wesentlichen Grundaussagen überein. Aufgrund des „Alters“⁷ beider Texte, sind die darin gemachten Angaben teilweise veraltet und entsprechen nicht mehr dem heutigen Stand der Technik.

Während die BKK ihre Empfehlungen nur für den Anwender – sprich den Archivar – gemacht hat, spricht die ARK mit ihrem Kriterienkatalog auch direkt die Programmierer an und gibt Hinweise zur möglichen Umsetzung der geforderten Kriterien. Aufgrund der unterschiedlichen Zielgruppen sind die Texte auch in ihren Details verschieden. Die ARK geht sehr detailliert auf die allgemeinen Anforderungen an die Software, die Sortierung der Titelaufnahmen und die Indizierung ein, während die BKK sich mehr auf die möglichen Funktionen der Software konzentriert.

Generell lässt sich sagen, dass sowohl die BKK als auch die ARK von einer für den Einsatz in Archiven gedachten Software vor allem auf die Automatisierung von Abläufen Wert legen. So sollen vor allem Plausibilitätskontrollen z.B. für Sperrfristen oder Datumsangaben und die automatische Generierung von Konkordanzlisten, Indizes, Findmitteln etc. angeboten werden. Heute eingesetzte Erschließungsprogramme unterstützen zumindest die Minimalforderungen der beiden Gremien, teilweise auch mehr.

⁵ Der Archivar 46 (1993), Sp. 431-442

⁶ Der Archivar 47 (1994), Sp. 283-288

⁷ 12 bzw. 13 Jahre sind in der IT-Branche nunmal sehr lange, da die Innovationszyklen und die Zeitspanne, nach der eine Software als „veraltet“ gilt, sehr kurz sind.

Eine aktuellere Übersicht, die aber auch auf die Empfehlungen und Richtlinien von ARK und BKK aufbaut, findet sich in dem Buch „Praktische Archivkunde“ von Norbert Reimann. Darin werden folgende Standardanforderungen für Archivsoftware genannt:

1. die Möglichkeit der Zwischenarchivverwaltung (Fristenüberwachung)
2. die Unterstützung der Verzeichnungsarbeiten von Archivalien unter der Verwendung unterschiedlicher Masken für die verschiedenen Archivgutarten. Dazu gehört auch:
 - a. die Beachtung der verschiedenen Verzeichnungsangaben bei der Erschließung von Urkunden, Akten, Amtsbüchern, Karten, Fotos usw.,
 - b. die Möglichkeit der Erstellung von Spezialinventaren
 - c. die Einbindung von Klassifikationen, Importieren von Aktenplänen,
 - d. die Indizierung (Geographicum, Sachbetreffe, Personen),
 - e. die Möglichkeit der Erstellung archivinterner Findbücher und Benutzerfindbücher; das meint die Möglichkeit des Sperrens einzelner Archivalieneinheiten,
 - f. Eingabe von Schadensdaten bei verzeichneten Archivalien, um zu einem späteren Zeitpunkt beständeübergreifende Schadenslisten ausdrucken zu können.
3. komfortable Recherchefunktionen
4. Möglichkeiten der Beständeverwaltung
5. Import- und Export der Bestandsdaten
6. Benutzerverwaltung mit Nachweis der bearbeiteten Themen und der benutzten Archivalien
7. differenzierte Rechtevergabe, z.B. die Vergabe von Zugriffsberechtigungen für einzelne Bestände für einen oder mehrere Bearbeiter
8. Statistik, Benutzerstatistik, Verzeichnungsstatistik⁸

Diese Anforderungen muss Aster natürlich auch erfüllen.

⁸ Nimz: EDV und Archive, in: Praktische Archivkunde, S. 204

2.2. Archivsoftware

Im Folgenden sollen vier Archivanwendungen vorgestellt werden.

Ziel ist es dabei nicht, die Anwendungen direkt miteinander zu vergleichen⁹ oder eine vollständige Analyse nach softwareergonomischen Gesichtspunkten oder DIN-Konformität durchzuführen.¹⁰

Vielmehr sollen diese Anwendungen als bereits heute in kleinen und mittleren Archiven eingesetzte Lösungen mit ihren Vor- und Nachteilen vorgestellt werden, um eine Art Pflichtenheft für die zu erstellende freie Archivsoftware zu realisieren. Wichtige Aspekte, die beachtet wurden, waren unter anderem das Grundkonzept der Software, die Möglichkeit zur Abbildung der Archivtektonik, Import- und Exportmöglichkeiten und der Ablauf der Verzeichnungsarbeiten.

2.2.1. Ariadne

Ariadne ist eine kostenlose Open-Source-Archivsoftware, die im Rahmen des DFG-Projektes „Archivverbund in Mecklenburg-Vorpommern“ erstellt wurde. Ziel soll es dabei sein, eine virtuelle Darstellung der Archivlandschaft und ein breites Spektrum an Informationsmöglichkeiten wie z.B. detaillierte Informationen über die Archivbestände bis hin zur archiv- und bestandsübergreifenden Recherche zu erreichen. Aufgrund dieser Konzeption liegt der Schwerpunkt auf der Erstellung von onlinefähigen Findmitteln und weniger auf der von Printversionen. Druckbare Findmittel werden im PDF-Format generiert. Leider fehlen jegliche Möglichkeiten zur Indizierung, da man diesen Aspekt zugunsten von Volltextrecherche und feldbezogener Suche vernachlässigt hat.

„Die Idee zum ARIADNE-Projekt wurde 1999 im Archiv der Ernst-Moritz-Arndt-Universität Greifswald entwickelt. Ausgangspunkt waren Überlegungen, ein bedienungsfreundliches Verzeichnungsprogramm für alle Arten von Aktenbeständen und Sammlungsgut in Archiven kleiner und mittlerer Größe zu entwickeln, das auf herkömmlicher, handelsüblicher Software basiert (MS Access), die gemeinsamen archivischen Anforderungen (Standards) an die Ordnung und Verzeichnung befriedigt und ohne größeren organisatorischen oder technischen

⁹ Ein direkter Vergleich würde auch mehr als schwer fallen, da sich diese Anwendungen in Hinsicht auf Konzeption und Umsetzung sehr unterscheiden.

¹⁰ Dafür sei an dieser Stelle auf eine Arbeit von Andreas Berger verwiesen:
www.archive.nrw.de/dok/Transferarbeiten/Berger/Transferarbeit%20web.pdf

Aufwand in der praktischen Archivarbeit eingesetzt werden könnte. Im Laufe eines Jahres wurde in enger Kooperation von Universitätsarchiv und Universitätsrechenzentrum eine Programm-Architektur entwickelt, die diesen Anforderungen entspricht. Einige bereits weit entwickelte Module von ARIADNE wurden am 8.11.2000 auf der 2. Konferenz der vorpommerschen Archive in Greifswald einer größeren Anzahl von Vertretern der vorpommerschen Archive vorgestellt und sind sehr positiv aufgenommen worden.

In diesem Zusammenhang entstand die Idee, mehr daraus zu machen. Was Bibliotheken schon lange kennen, nämlich einen virtuellen Verbund im Internet, der es Benutzern und Mitarbeitern ermöglicht quer durch die verschiedenen Kataloge aller beteiligten Häuser, vom eigenen Schreibtisch aus zu recherchieren, dass ist für Archive noch immer Zukunftsmusik, auch wenn es erste Ansätze gibt.“¹¹

Seit 1999 wurde die Software von der Universität Greifswald kontinuierlich weiterentwickelt, und seit 2002 erfolgt eine Förderung von Seiten der Deutschen Forschungsgemeinschaft.

Ariadne ist sowohl unter Windows als auch unter Linux lauffähig und arbeitet im Hintergrund mit einer MySQL-Datenbank.

Für die Erschließung gibt es unterschiedliche Erfassungsmasken für Sachakten, Personalakten, Urkunden, Karten/Risse/Pläne und Fotos. Eine Möglichkeit der Veränderung einer Maske oder gar eine Neuanlage ist nicht vorgesehen. Für die Verzeichnung gibt es weiterhin drei verschiedene Ansichtsmodi: die Maskenansicht, die Listenansicht und eine Tabellenansicht.

Die Maskenansicht ist für die schnelle Verzeichnung und Korrektur von Verzeichnungsangaben gedacht. Mit dem Listenmodus kann man in einigen Feldern nach bestimmten Begriffen suchen lassen und die Treffermenge in einer Liste anzeigen lassen. Diese können dann in der Maskenansicht weiter bearbeitet werden. In der Tabellenansicht werden die Verzeichnungseinheiten und deren Datenfelder in einer Tabelle dargestellt, in welcher auch einzelne Felder bearbeitet werden können.

Bei der Foto- und Kartenverzeichnung ist das Einbinden von Bildern möglich. Die Verzeichnung an sich erfolgt nach den OVG. Serienbildung und Sperrfristenkontrolle werden leider nicht unterstützt. Innerhalb der Verzeichnungsmaske ist es möglich, eine Verzeichnungseinheit für die Gastrecherche freizugeben oder zu sperren.

Überhaupt sind Automationen und logische Kontrollen kaum in das Programm integriert.

Die Software enthält eine Beständeübersicht in Form einer flachen Liste. Klassifikationen werden separat angelegt und verwaltet.

¹¹ http://ariadne.uni-greifswald.de/images/Ariadne/de_history.html

Ein wichtiger Schwerpunkt liegt auf der Benutzerverwaltung, so dass diese sehr ausgereift ist. So kann jedes Archiv beliebig viele Benutzergruppen besitzen und jeder Benutzer kann wiederum beliebig vielen Benutzergruppen zugeordnet werden.

„Fazit: Diese Lösung genügt nur archivfachlichen Grundanforderungen und lässt im Detail Wünsche offen. Flexibilität und Anpassungsfähigkeit sind kaum vorhanden. Vom Archivar wird ein hohes Maß an Disziplin gefordert, die Verzeichnungen gleichmäßig zu erstellen, da jegliche Kontrolle oder Strukturierung seitens des Programms fehlen. Die Beurteilung muss jedoch vor dem Hintergrund, dass es sich um Freeware handelt, die durchaus auf niedrigem Niveau gebrauchstauglich ist, etwas milder ausfallen.“¹²

2.2.2. AUGIAS-Express 3

Seit 15 Jahren gibt es mittlerweile die Firma AUGIAS-Data. AUGIAS-Data ist vor allem auf die Erstellung von Anwendungen im Museums-, Bibliotheks- und Archivbereich spezialisiert. Für archivarische Belange gibt es verschiedene Anwendungen: AUGIAS-ZwA für die Verwaltung von Zwischenarchiven und Altregistraturen, AUGIAS-Archiv für umfangreiche Archive und Archivverwaltungen und dessen „kleiner Bruder“ AUGIAS-Express. AUGIAS-Express ist das Einsteigerpaket von AUGIAS und ist vom Funktionsumfang für kleine und mittlere Archive völlig ausreichend.

Mit AUGIAS-Express lassen sich Bestände archivgutartenspezifisch verzeichnen. Das große Manko des Vorgängers, der nur eine Archivgutart pro Bestand bei der Verzeichnung zuließ, ist mittlerweile beseitigt und so können nun verschiedene Erfassungsmasken bei der Erschließung eines Bestandes verwendet werden. Die Formulare sind sogar veränderbar, d.h. man kann die Feldgrößen den Bedürfnissen anpassen, einzelne Felder komplett weglassen, neue hinzufügen oder Standardwerte vorgeben, die beim Erstellen eines neuen Datensatzes bereits eingetragen sind.

AUGIAS-Express speichert seine Daten in einer einzigen Access-Datenbankdatei, in der jedoch beliebig viele Bestände verzeichnet werden können. Eine Abbildung der Archivtektonik ist leider nicht möglich. Zu jedem Bestand lassen sich bis zu zwei Klassifikationen zuordnen, die jedoch an anderer Stelle im Programm verwaltet werden. Um also während der Verzeichnung die Klassifikation zu bearbeiten, ist ein „Wandern“ durch diverse Masken und Menüs nicht zu vermeiden. Dies ist die größte Schwachstelle des

¹² www.archive.nrw.de/dok/Transferarbeiten/Berger/Transferarbeit%20web.pdf

Programms: Es ist in seinem Aufbau und seiner Strukturierung sehr unübersichtlich, so dass es vorkommen kann, dass sich ein unerfahrener oder wenig erfahrener Benutzer auf der Suche nach der richtigen Funktion durchaus „verlaufen“ kann. Dieser Effekt wird dadurch noch verstärkt, dass es nirgendwo eine Navigationsleiste (ähnlich denen auf Internetseiten) gibt, wo grob dargestellt wird, wo man sich zurzeit innerhalb der Programmstruktur befindet.

Ein anderes Problem ist, dass es keine vollständige Anzeige des gesamten Findmittels gibt, sondern immer nur Teilansichten wie die Klassifikation oder die Anzeige aller Verzeichnungseinheiten, die einem ausgewählten Klassifikationspunkt zugeordnet wurden. Überhaupt sind Listen ein integraler Bestandteil des Programms. Es gibt zum Beispiel so genannte „Lookup-Listen“, die das Auffinden von Archivalien dank diverser Sortierfunktionen schnell und komfortabel machen sollen. In vielen Fällen sind diese jedoch nicht immer die beste Wahl.

Das Programm unterstützt die automatische Vergabe von Signaturen, wobei man sogar zwischen dem Bär'schen Prinzip und der Signierung nach dem Auftreten im Findbuch wählen kann.

Weiterhin gibt es ein vollständig integriertes Fotoarchiv, das alle Standardgrafikformate unterstützt. Zu jeder Verzeichnungseinheit lassen sich beliebig viele Bilder hinterlegen.

Die Recherche ist als Volltextrecherche, als feldbezogene Suche mit Booleschen Operatoren oder als Indexrecherche durchführbar. Es gibt sowohl einen Bestands- als auch einen Generalindex. Verweise innerhalb der Indizes sind ebenfalls möglich. Die Recherchen können bestandsübergreifend, in ausgewählten Beständen oder in einem einzelnen Bestand durchgeführt werden.

Die Import- und Exportfunktionen von AUGIAS-Data sind leider nicht zufrieden stellend. Man kann lediglich einen Datenaustausch zwischen verschiedenen AUGIAS-Anwendungen vornehmen. Der Import und Export in andere Formate wird nur gegen Entgelt von AUGIAS-Data selbst durchgeführt. Da AUGIAS-Express jedoch auf MS Access aufbaut, ist es durchaus möglich, eine erweiterte Import/Exportschnittstelle selbst zu programmieren.¹³

Die Findbücher können ohne Probleme ausgedruckt werden, man kann sogar zwischen verschiedenen Layouts wählen oder neue erstellen. Spezialfindbücher wie sachthematische Inventare sind ebenso realisierbar. Für den Findbuchausdruck wird MS Word benötigt. Ebenso können die Findbücher unter www.findbuch.net im Internet veröffentlicht werden.

¹³ Erste Tests bezüglich eines Exports von AUGIAS-Express-Daten wurden im Rahmen dieses Projekts bereits zufriedenstellend durchgeführt. Da im Moment aber die Erstellung des Erfassungsprogramms Vorrang hat, wurde der Datenimport und -export anderer Formate erstmal zurückgestellt.

Natürlich besitzt AUGIAS-Express auch eine gut durchdachte Benutzerverwaltung und es ist sogar möglich, einzelne Bestände für die Recherche zu sperren.

Abschließend bleibt zu sagen, das AUGIAS-Archiv ein leistungsfähiges Archivprogramm ist, mit dem nahezu alle Ansprüche eines kleinen oder mittleren Archivs erfüllt werden, das jedoch beim Programmablauf und beim Datenimport und –export einige gravierende Mängel aufweist.

2.2.3. FAUST Entry Archiv

FAUST Entry Archiv ist eine Entwicklung der Firma Doris Land Software-Entwicklung. FAUST ist ein Datenbanksystem, das ursprünglich für den Einsatz in Museen und Sammlungen konzipiert wurde und die dokumentarische Bearbeitung von Objekten aller Art anstrebt.

Mit FAUST Entry Archiv liegt nun ein Datenbanksystem vor, das hauptsächlich für die Arbeit in einem Archiv konzipiert ist.

FAUST Entry Archiv bietet eigentlich alle benötigten Funktionen zur Arbeit in einem Archiv. Man kann damit Archivbestände aufbauen, erschließen, verwalten und veröffentlichen. Es lassen sich beliebig viele Datenbanken mit beliebig vielen Beständen in einer Datenbank anlegen und verwalten. Es gibt auch die Möglichkeit, eine Registratur oder ein Zwischenarchiv zu verwalten und ausgewählte Daten daraus in den Archivbestand zu übernehmen.

Bei den meisten Feldern können bereits vorhandene Daten übernommen werden und der Arbeitsablauf gestaltet sich somit fehlerfreier und effizienter. So können z.B. Akzessionsdaten in der Bestandsbeschreibung und der Verzeichnung direkt übernommen werden. Es gibt verschiedene Erfassungsmasken, die auch noch weiter angepasst werden können und so die Verzeichnung unterschiedlichen Archivguts ermöglichen. Ebenso ist es möglich, die Archivbibliothek und neue Medien wie CDs und DVDs zu verzeichnen.

Mittels Mausclick auf ein Feldlabel ist es auch möglich, sich die Feldbeschreibung anzeigen zu lassen. Alle Eingaben werden automatisch geprüft, damit zum Beispiel jede Signatur nur einmal vorhanden ist. Eine automatische Signaturenvergabe ist ebenfalls möglich.

Die Klassifikationen lassen sich tief schachteln und können in das System importiert, exportiert und jederzeit – auch rückwirkend – verändert werden. Es ist auch möglich, eine Verzeichnungseinheit zu mehreren Klassifikationspunkten hinzufügen.

Daten können im CSV- oder XML-Format importiert und exportiert werden, dies erfolgt über eine komfortable Dialogsteuerung. Damit ist es auch möglich, unbekannte XML-Daten den FAUST-Feldern zuzuordnen.

Die Bestände können in verschiedenen Formaten veröffentlicht werden. So gibt es die Möglichkeiten einer Ausgabe als PDF, als RTF, als HTML, nach Word, als Email oder als Textdatei. Auch das direkte Ausdrucken ist möglich.

Mit FAUST Entry Archiv zu arbeiten ist etwas gewöhnungsbedürftig, aber wenn man sich einmal eingearbeitet hat, stehen einem eine Vielzahl an Funktionen und Möglichkeiten offen.

Das große Manko ist, dass es sehr unübersichtlich und dadurch teilweise recht verwirrend ist.

2.2.4. MidosaXML

MidosaXML wurde zwischen 2000 und 2003 von einer Arbeitsgemeinschaft der Entwicklungspartnerschaft PARSIFAL erstellt. Die Arbeitsgemeinschaft setzte sich aus dem Bundesarchiv, der Archivschule Marburg und der Firma Startext zusammen. Ziel war es, die Software Midosa, die bereits seit über 10 Jahren in Baden-Württembergischen Staatsarchiven im Einsatz war, zu modernisieren.

PARSIFAL wiederum ist ein Zusammenschluss des Bundesarchivs, der Archivschule Marburg und der Landesarchivdirektion Baden-Württemberg mit der Zielsetzung, non-proprietäre, nachnutzbare und zukunfts offene Erschließungssoftware für Archive unter Einsatz moderner Technologien zu fördern.¹⁴

MidosaXML wurde für die schnelle Verzeichnung und anschließende Veröffentlichung von Beständen im Internet oder als gedrucktes Findmittel entwickelt. Die Daten werden dabei innerhalb einer Datenbank¹⁵ im XML-Format gespeichert. Das Programm zeichnet sich durch breite Import- und Export-Funktionalitäten aus.

Aufgrund seiner Konzeption ist MidosaXML eine reine Erschließungssoftware, d.h. es gibt keine Funktionen zu Archivverwaltungsaufgaben wie Benutzer-, Magazin- und Bestandsverwaltung. Da die Daten im XML-Format gespeichert werden, spiegelt sich auch in der Programmstruktur wieder. Der Programmaufbau und –ablauf ist streng hierarchisch, d.h. man folgt bei der Verzeichnung einem genau festgelegten Ablauf, indem man zuerst ein

¹⁴ MidosaXML-Handbuch, S. 5. (zu finden unter www.archivschule.de/uploads/midosa/HandbuchMidosaXM.doc)

¹⁵ in der Einzelplatzversion: MS Access, im Netzwerk: MSSQL oder Oracle.

Findmittel anlegt¹⁶, dann in dem Findmittel das Element „Angaben zum Bestand“ erzeugt und darunter mit der eigentlichen Verzeichnung beginnt. Ein Findmittel wiederum kann mehrere Bestände enthalten. Dieser strenge Ablauf bietet jedoch den Vorteil, dass das Programm – wenn man sich einmal darauf eingelassen hat – zum einen nahezu selbsterklärend ist, und zum anderen, dass man Schritt für Schritt alle wichtigen Informationen erfasst und sich somit die Zeit für die Nachbearbeitung zur Veröffentlichung drastisch reduziert. Die Programmoberfläche ist ebenso einfach wie gut durchdacht. Es gibt nur eine Oberfläche, in der gearbeitet wird. Auf der linken Seite befindet sich ein Fenster mit einer Baumansicht, in der mittels entsprechenden Buttons zwischen der Anzeige der vorhandenen Findmittel, der Inhalte der einzelnen Findmittel und dem Index eines Findmittels hin und her geschaltet werden kann. In diesem Fenster können auch direkte Manipulationen wie das Löschen und Verschieben von Verzeichnungseinheiten¹⁷ innerhalb des Bestands vorgenommen werden.

Durch dieses Fenster ist es auch jederzeit auf einen Blick möglich festzustellen, wo man sich momentan befindet und wie der Bestand als solcher gegliedert ist.

Im mittleren Fenster erfolgt die eigentliche Erfassung. Es gibt zwei Arbeitsmodi: „Redaktion“ und „Erfassung“. Im Redaktionsmodus wird das gesamte Findmittel in einer Listenansicht dargestellt. Diese Ansicht ist in erster Linie für die abschließende Überarbeitung des Findmittels vor der Veröffentlichung gedacht, hat aber den Vorteil, dass hier weitere Elemente eingefügt werden können, die im Erfassungsmodus nicht angeboten werden.¹⁸ Ansonsten ist die Arbeit in diesem Modus relativ unkomfortabel.

Für die eigentliche Verzeichnung ist der Erfassungsmodus gedacht. Dazu gibt es für jede Verzeichnungseinheit drei Reiter, die zu den eigentlichen Elementen führen. Der erste Reiter nimmt die wichtigsten Informationen wie die Signatur, den Titel, die Enthält-Vermerke und die Laufzeit auf. Der zweite Reiter enthält u.a. die Elemente „Schäden“, „Provenienz“, „Vorprovenienzen“ und „Altsignaturen“. Der dritte Reiter steht für die Indexbegriffe – sortiert nach Beruf, Kompetenzen, Ort, Sachen, Personen etc. – zur Verfügung. Die einzelnen Indizes sind – wie die Datenstruktur überhaupt – eng an EAD angelehnt.

Das rechte Fenster zeigt an, welche neuen Elemente – ausgehend von dem Punkt, an dem man sich momentan im Findmittel befindet – auf derselben Ebene oder unterhalb hinzugefügt werden können.

¹⁶ In einer Datenbank können beliebig viele Findmittel gespeichert werden.

¹⁷ Jedoch immer nur eine!

¹⁸ z. Bsp. die Elemente „Digitale Objekte“ und „Bearbeiter“.

Für die Veröffentlichung der Findmittel gibt es zwei Möglichkeiten: Export der Daten in ein Textverarbeitungsprogramm wie MS Word oder OpenOffice zum Ausdrucken oder nach HTML zur Präsentation der Findmittel im Internet. Dabei wird ein Ordner erstellt, den man direkt auf einen Webserver kopieren kann und sofort einsatzbereit ist. Für diese Zwecke ist es bereits in MidosaXML möglich, rudimentäre Textverarbeitungsfunktionen wie Fettschrift und Unter- und Durchstreichung von Textteilen zu nutzen.

Die großen Nachteile von MidosaXML sind, dass es – bedingt durch die Konzeption – keine Beständeübersicht gibt, logische Prüfungen zu selten eingesetzt werden und dass es keine unterschiedlichen Erfassungsmasken für verschiedenartiges Archivgut wie Fotos, Urkunden, Sachakten etc. gibt.¹⁹

2.2.5. Fazit

Aufgrund der hier vorgestellten Softwarelösungen sollte die zu entwickelnde Software folgende Aspekte berücksichtigen: Die Software soll nicht wie MidosaXML nur zur Verzeichnung und Recherche von Beständen einsetzbar sein, sondern auch Features wie Benutzerverwaltung, Beständeübersichten und –verwaltung, Verwaltung der Zugänge von Archivgut und ähnliches beinhalten. Auf eine Ausleihfunktion kann in Hinblick auf den anvisierten Nutzerkreis (kleine und mittlere Archive) vorerst verzichtet werden. Da alle drei Programme, die solche Funktionen bieten (Faust Entry Archiv, Augias-Express 3 und Ariadne) unter einer gewissen Unübersichtlichkeit leiden, sollten die einzelnen Teile in verschiedene Module ausgelagert werden, wobei man natürlich benötigte Informationen auch aus den anderen Modulen von einem Modul aus einsehen und verwenden können sollte. Es soll also ein Programmpaket entwickelt werden, das die benötigten Funktionen bereitstellt und auch untereinander kommunizieren kann.

Das Verzeichnungsmodul sollte in seinem Aufbau ähnlich dem von MidosaXML sein, da dies in seiner Einfachheit und Intuitivität nicht zu übertreffen ist und trotzdem eine Vielzahl an Möglichkeiten und Funktionalitäten in sich vereint. Auf jeden Fall muss hier aber noch eine Beständeübersicht integriert werden, um einen einfachen und schnellen Arbeitsablauf zu gewährleisten.

Die Benutzerverwaltung sollte analog zu Augias-Express 3 und Ariadne die Bildung von Gruppen und Rollen ermöglichen um so auch eine komplexe Benutzerverwaltung zu

¹⁹ MidosaXML ist nur für die Erschließung von Sachakten ausgelegt.

realisieren, da der Datenschutz immer wichtiger wird und in einer elektronisch vernetzten Welt nicht mehr wegzudenken ist. Dabei ist es auch wichtig, dass die Daten im Hintergrund verschlüsselt und geschützt werden und es keine so genannten „Backdoors“ gibt, durch die man ohne Probleme an die Daten herankommt.²⁰

Das Recherchemodul sollte sehr umfangreich und variabel sein, so soll es ähnlich den Recherchemodulen von Online-Bibliotheksverbänden mehrere Suchfunktionen geben, z. B. eine einfache Suche, in der nur nach Schlagworten gesucht wird, eine erweiterte Suche in der man z.B. die zu durchsuchenden Felder beschränken kann oder auch bestimmte Schlagworte ausschließen kann und eine Expertensuche in der z.B. Trunkierungen und die Nutzung von Abstandsoperatoren und Klammerungen möglich sind, Die Recherchen sollen natürlich sowohl bestandsübergreifend, als auch nach einer bestimmten Auswahl durchgeführt werden können. Langfristiges Ziel bei der Recherche soll sein, dass man Funktionalitäten wie bei den großen Dankbankanbietern zur Verfügung hat, z. B. Durchführen mehrerer Recherchen, deren Ergebnisse dann verknüpft werden können, trunkierte Suchen, Nutzung der Booleschen Operatoren, Bereitstellen der Ergebnismengen in verschiedenen Ausgabeformaten und ähnliches.

2.3. Freie Software vs. Proprietäre Software

2.3.1. Proprietäre Software²¹

Nach den Regeln der Free Software Foundation²² sind proprietäre Programme, bei der eine Firma oder ein Entwickler alle Rechte an einer Software hält, und anderen gleichzeitig den Zugang zum Quelltext und das Recht die Software zu kopieren, verändern oder zu studieren verbietet. Dies trifft mehr oder weniger auf sämtliche Software zu, die es zu kaufen gibt.

Das Problem proprietärer Software stellt sich aus den Augen der FSF wie folgt dar: Proprietäre Software verhindert ein Studium und Verstehen des Programms, der Nutzer ist in Bezug auf die Möglichkeiten der Programms vollständig von dem Entwickler abhängig, was

²⁰ Ein schlechtes Beispiel hierfür ist Augias-Express, das auf eine Access-Datenbank aufsetzt und eine gute Benutzerverwaltung hat, jedoch kann man die Datenbank auch in Access direkt öffnen, und dort jegliche Daten, d.h. sowohl Benutzerdaten, Passwörter und Zugriffsberechtigungen, als auch Erschließungsdaten, ohne Weiteres manipulieren.

²¹ http://de.wikipedia.org/wiki/Propriet%C3%A4re_Software

²² Die Free Software Foundation (FSF) wurde 1985 als gemeinnützige Organisation gegründet, um freie Software zu fördern und Kapital für diese Arbeit zusammenzutragen. Seit Mitte der 1990er Jahre konzentriert sich die Arbeit der FSF hauptsächlich auf rechtliche und strukturelle Belange, da seit dieser Zeit viele Firmen und Privatpersonen begonnen haben, freie Software zu schreiben. Homepage der FSF: <http://www.fsf.org>

die unterstützten Standards, Datenformate und ähnliches angeht. Im schlimmsten Fall bedeutet dies, dass man eine sehr leistungsfähige Software hat, aber keinen Datenaustausch mit anderen Anwendungen vornehmen kann, da die Software nur ihr eigenes Format unterstützt.

Ebenso ist eine Anpassung der Software auf die gewünschten Erfordernisse nicht möglich, bzw. nur über den Entwickler, der dies gewiss nicht umsonst tun wird.

Natürlich hat proprietäre Software auch einige Vorteile, so ist die Weiterentwicklung des Programms für eine Weile gewährleistet, man muss aber damit rechnen, dass diese Entwicklung irgendwann aus Kostengründen eingestellt werden kann. In diesem Fall bleibt nur noch zu hoffen, dass der Entwickler den Programmcode spätestens zu diesem Zeitpunkt freigibt, damit man mittels freier Software eine Schnittstelle zum Programm schaffen kann, ansonsten droht unter Umständen ein Datenverlust!

Ein weiterer Vorteil ist die Bessere Nutzung der Datenformate. Im Gegensatz zu einem offenen Standard wird die komplette Software zu einem einzigen bestimmten Zweck programmiert und alle Teile des Programms sind im Optimalfall darauf abgestimmt, was eine höhere Leistungsfähigkeit mit sich bringen kann.

2.3.2. Freie Software und Open Source²³

Von freier Software dagegen spricht man, wenn diese mit dem Quelltext vertrieben wird bzw. der Quellcode auf Anfrage einsehbar ist, und dem Nutzer das Recht einräumt wird, diesen zu studieren, das Programm zu kopieren, es zu verändern und die veränderte Version kostenlos oder gegen Entgelt weiter zu geben.

Vorteile sind hierbei natürlich die Möglichkeit, das Programm den eigenen Bedürfnissen entsprechend weiter anzupassen und zu verändern und es weiterzugeben. Allgemein muss hier nochmals ausdrücklich erwähnt werden, dass Freie Software nicht gleichbedeutend mit kostenloser Software sein muss. In den meisten Fällen ist dem so, aber es nicht zwingend erforderlich.

Des Weiteren ergibt sich durch die Quelloffenheit ein besserer Umgang mit Programmfehlern und Erweiterungen. So sind bei einer aktiven Gemeinschaft mehr und bessere Erweiterungen der Software zu erwarten, als bei einer proprietären Software, bei der ein Handvoll von

²³ http://de.wikipedia.org/wiki/Open_Source, http://de.wikipedia.org/wiki/Freie_Software

Entwicklern das leistet, was bei freier Software im Idealfall Hunderte oder Tausende von Leuten machen.²⁴ Sicherheit ist bei Freier Software natürlich ein sehr heikles Thema, da offener Quellcode nach Meinung von Kritikern zu einem einfacheren Hacken der Software führen kann, und nach Meinung der Befürworter Fehler einfacher zu finden aber auch zu korrigieren sind, was letztendlich zu einem robusteren und stabileren System führt²⁵.

1998 wurde aus Marketing-Gründen der Begriff „Open Source“ von Eric S. Raymond²⁶, Bruce Perens und Tim O’Reilly²⁷ eingeführt. Diese drei gründeten auch die Open Source Initiative (OSI). Obwohl Freie Software und Open Source somit die gleiche Bedeutung haben, hat sich mittlerweile ein Spannungsfeld zwischen der OSI und der FSF bezüglich der Lizenzierungen aufgebaut, welches hier aber nicht weiter erläutert werden soll, da dies nicht Thema dieser Arbeit ist, und eindeutig zu weit führen würde²⁸.

Ein gravierender Nachteil von freier Software soll an dieser Stelle nicht unerwähnt bleiben: Durch die Offenlegung des Quellcodes und der erlaubten Weitergabe und Änderung des Programms verliert man sehr schnell die Kontrolle über die Software, so dass man durchaus mit Derivaten rechnen muss, die im Extremfall zu Systeminstabilität oder Datenverlust führen können. Eine Möglichkeit um dies zu verhindern wäre die Einrichtung einer Internetseite für die Software, auf der die Möglichkeit besteht, Codeänderungen von Dritten vorzustellen, zu prüfen und dann in die eigentliche Software zu implementieren, damit diese wieder der breiten Nutzermasse zu Verfügung stehen.

2.3.3. Fazit

Die Entscheidung, ob man eine proprietäre oder eine freie Software entwickeln möchte, sollte wohl überlegt sein. Beide Varianten haben eine Menge an Vor- und Nachteilen, die gewissenhaft mit einander abgewogen werden sollten. Letztendlich kann – für die Software, die hier entwickelt werden soll – die Entscheidung nur auf eine freie Software fallen.

²⁴ bestes Beispiel hierfür ist der Vergleich zwischen dem Internet Explorer von Microsoft und Mozillas Firefox

²⁵ <http://www.zdnet.de/itmanager/tech/0,39023442,2130055,00.htm>

²⁶ Autor des Essays „Die Kathedrale und der Bazaar“, der zur Freigabe des Quellcodes des Netscape Navigators führte.

²⁷ Gründer und Vorstand des O’Reilly-Verlags, welcher sich ausschließlich mit der Verlegung von Büchern betreffend Computer und Programmierung befasst (www.oreilly.com). O’Reilly war z.B. maßgeblich an der Entwicklung der Programmiersprache Perl beteiligt.

²⁸ Für einen kurzen aber verständlichen Einstieg sei hier auf das Glossar des GNU-Projekts der FSF verwiesen: <http://www.gnu.org/philosophy/categories.de.html>

Der Grund dafür liegt zum Einen darin, dass auf dem Nischenmarkt für Archivsoftware fast ausschließlich proprietäre Anwendungen verfügbar sind, und – da es ein so kleiner Markt ist – sich nicht viel bewegt, was z.B. Innovationen angeht. Das liegt sicherlich auch daran, dass der Markt für reine Erschließungsprogramme mit der zunehmenden digitalen Verwaltung von Schriftgut zugunsten integrierter modularer Lösungen wie z.B. Consul/Star der Schweizer Firma CM Informatik AG²⁹ gänzlich wegbrechen wird.

Zum anderen soll die Software „von Archivaren für Archivare“ sein, um ein System zu schaffen, das ganz auf die Erfordernisse eines Archivs abgestimmt ist. Da es in der aktuellen Berufsbild-Diskussion auch darum geht, ob ein Archivar auch Informatiker sein sollte, so ist das ganz richtig, wenn diese Frage verneint wird. Jedoch sind Archivare bereits heute dazu aufgefordert, sich zumindest grundlegende Informatik-Kenntnisse anzueignen, um für die Zukunft gewappnet zu sein. Eine freie Archivsoftware kann von dieser Entwicklung nur profitieren, da die Archivare nun die Möglichkeit haben, sich nicht nur bei den Produzenten proprietärer Software über fehlende Features zu beschweren, oder sich neue zu wünschen³⁰, sondern diese für den Eigengebrauch nun selber erstellen³¹ und sogar anderen Archiven zu Verfügung stellen können, wodurch die Innovationsgeschwindigkeit extrem erhöht werden kann.

2.4. Datenorganisation

2.4.1. Datenstruktur

Um Daten besser nutzbar zu machen, werden sie strukturiert. Für die Strukturierung gibt es mehrere Möglichkeiten:

Eine wäre es, die Daten als CSV-Datei³² abzuspeichern, was bedeutet, dass die Daten in einer einfachen Textdatei in Tabellenform gespeichert werden. Da jedoch archivische Findmittel in ihrer Form einem Buch ähneln, eignen sie sich nicht unbedingt für diese Art von

²⁹ www.cmiag.ch

³⁰ Wobei sich eine Umsetzung der Wünsche auf jeden Fall im monetären Aspekt niederschlagen wird, und Archive sind je bekanntlich finanziell nicht unbedingt hervorragend ausgestattet.

³¹ Da es ja Programmierer wie Sand am Meer gibt, kann man sich auch einen solchen ins Boot holen, das ist wahrscheinlich immer noch billiger, als bei den Archivsoftware-Herstellern eine Einzellösung in Auftrag zu geben.

³² Comma Separated Value

Speicherung. Natürlich ist es möglich, die Daten in diese Form zu bringen, aber das Format bietet sich wirklich nur für den Datenaustausch, nicht für die Datenvorhaltung an.³³

Eine weitere Möglichkeit wäre die Nutzung von Datenbanken. Der Vorteil ist eine bessere Strukturierung und funktionstüchtige und mächtige Retrieval-Tools mittels SQL. Datenbanken unterscheiden sich in relationale und objektorientierte Datenbanken.

Relationale Datenbanken sind der heutige Standard und zeichnen sich dadurch aus, dass die Daten mittels Relationen verknüpft sind. Leider besteht bei relationalen Datenbanken das Problem, das man letztendlich gezwungen ist, seine Daten an die Datenbankstruktur anzupassen und nicht umgekehrt. Dies wird auch dadurch verstärkt, dass die Findbücher verschiedener Archive in ihrer Strukturierung extrem unterschiedlich sein können. Objektorientierte Datenbanken hingegen sind in dieser Hinsicht sehr viel flexibler, allerdings haben sie sich zurzeit noch nicht durchgesetzt bzw. sind noch nicht voll einsatzfähig. Abgesehen davon gibt es keine freie objektorientierte Datenbank, so dass diese Lösung für dieses Projekt hinfällig ist.

Eine weitere Möglichkeit ist die Nutzung von Beschreibungs- oder Markup-Sprachen.

Beschreibungssprachen haben den enormen Vorteil, dass sie extrem flexibel sind. Je nach Bedürfnis kann man damit Daten kaum, etwas oder sehr stark strukturieren. Je nach Strukturierungsgrad bleibt die Lesbarkeit des Dokuments auch für den Menschen erhalten. Natürlich können die Strukturierungsgrade auch innerhalb eines Dokuments variieren. So könnte man in einem Buchkapitel alle vorkommenden Personennamen markieren und mit Zusatzinformationen versehen und beim nächsten genau dieses unterlassen. Somit lässt sich die Struktur eines Dokuments mittels Beschreibungssprachen in bester Weise kenntlich machen, die Daten für beliebige Filterungen bereitstellen und dass alles ohne Leereinträge, wie sie in Datenbanken oft vorkommen.

2.4.1.1. XML

Durch die weltweite Vernetzung von Rechnersystemen kam das Problem auf, dass man zwar Daten auf elektronischem Wege bis ans andere Ende der Welt übermitteln konnte, aber die verschiedensten Softwareapplikationen nicht unbedingt mit den unterschiedlichen Datenstrukturen zurecht kamen. Aus diesem Grund wurden so genannte Austauschformate

³³ Wenn man sich einmal die Import- und Export-Schnittstellen anschaut, so wird man feststellen, dass CSV fast immer unterstützt wird, da es in den meisten Fällen den kleinsten gemeinsamen Nenner darstellt.

entwickelt, die die Daten in einer festgelegten Struktur vorhalten, die mit Hilfe von Importfiltern in diverse Anwendungen eingelesen werden kann.

Ein solches Austauschformat ist XML. XML-Dokumente sind selbstbeschreibend, einfach und extrem flexibel strukturierbar, plattformunabhängig und non-proprietary. XML hat sich außerdem auch in der Archivwelt bereits bewährt, zu erwähnen seien hier nur die archivischen Austauschformate EAD und die Findmittel-DTD des DFG-Projekts „Entwicklung von Werkzeugen zur Retrokonversion archivischer Findmittel“³⁴. Die Programme MidosXML und V.E.R.A sind als Archivanwendungen ganz auf die Verarbeitung von XML-Dokumenten ausgerichtet.

XML steht für eXtensible Markup Language und ist eine vereinfachte Form von SGML. Im Gegensatz zu HTML, welches ebenfalls eine auf SGML beruhende Beschreibungssprache ist, dient XML zur Beschreibung und Identifikation von Daten und ermöglicht die Definition eigener Markupsprachen. Die mit XML definierten Markup-Sprachen werden als XML-Anwendungen bezeichnet.

Wofür kann man nun solche "XML-Anwendungen", verwenden? Welchen Zweck kann es haben, mit XML solche Sprachen festzulegen und zu normieren? Welche Vorteile haben solche XML-Anwendungen gegenüber HTML oder anderen Datei-Formaten?

- Mit XML kann man die logische Bedeutung von Daten, Informationen und Texten definieren - ähnlich wie die Tabellen- und Spalten-Bezeichnungen in Datenbanken und Tabellenkalkulationen.
- XML ermöglicht im Gegensatz zu HTML die Definition eigener oder zusätzlicher "Befehle" (Tags) - ähnlich wie bei der Definition von Makros in der Textverarbeitung
- XML-Applikationen eignen sich als Plattform- und Software-unabhängiges Austausch-Format für Daten zwischen verschiedenen Programmen und Rechnern - ähnlich wie RTF für Texte, CVS für Tabellen, EDI für kommerzielle Anwendungen - aber in einem einheitlichen, allgemein verwendbaren, Hersteller-unabhängigen Format.

Außerdem ist die Syntax von XML so streng festgelegt, dass XML-Anwendungen wesentlich einfacher, bequemer und effizienter von Programmen weiter verarbeitet werden können als HTML-Files.

³⁴ <http://www.archive.nrw.de/findbuch-digital/index.htm>

XML-Anwendungen können mittels XML-Parsern mit DOM (Document Object Model) oder SAX (Simple API for XML) verarbeitet werden. Ebenso lassen sie sich auch mit Hilfe von XSL (Extensible Style Language) in andere Formate umformen (z. B. CSV oder HTML) oder durch CSS (Cascading Style Sheets) grafisch darstellen. Des Weiteren gibt es mit XPath und XQuery die Möglichkeit XML-Dokumente zu durchsuchen und die Ergebnisse weiterzuverarbeiten. Daneben gibt es noch etliche andere Erweiterungen, die hier aber nicht weiter erwähnt werden sollen.³⁵

2.4.1.2. Verarbeitung von XML-Dokumenten

Um XML-Dokumente verarbeiten zu können, müssen diese erst einmal eingelesen werden. Dies erfolgt mit einem so genannten Parser. Der Parser gibt das Dokument in der Form von DOM oder SAX wieder.

SAX steht für „Simple API for XML“ und ist eine Programmschnittstelle (Application Programmers Interface - API) für die Weiterverarbeitung der Daten mit einer Programmiersprache. SAX liefert ein XML-Element nach dem anderen in einem Eingabestrom und eignet sich daher auch für sehr große XML-Dateien.

DOM (Document Object Model) ist, wie der Name schon sagt, ein Objektmodell, es beschreibt die in einem XML-Dokument enthaltenen Elemente als Objekte. DOM liefert eine komplette Baumstruktur aller Objekte eines XML-Dokuments und eignet sich daher nicht unbedingt für extrem große XML-Dateien.

2.4.2. Datenspeicherung

Zu Beginn des Projektes war es angedacht, die Findmittel-DTD des DFG-Projekts zur „Entwicklung von Werkzeugen zur Retrokonversion archivischer Findmittel“ zur Datenspeicherung zu verwenden. Es wurde jedoch schnell deutlich, dass die DTD zu umfangreich und zu flexibel ist, um in ihrer eigentlichen Form zur Datenhaltung eingesetzt zu werden. So ist es zum Beispiel möglich, mehrere „Provenienz“- oder „Umfang“-Elemente innerhalb einer Verzeichnungseinheit zu erstellen, was aus archivarischer Sicht nicht sein darf

³⁵ Für ein tieferes Eindringen in die Welt von XML sei an dieser Stelle die Internetseite des W3Consortiums (www.w3.org) verwiesen, das die Entwicklung von XML und dessen Zusätze leitet.

und eigentlich auch nicht passieren sollte, aber zumindest die DTD erlaubt solche Fälle und toleriert diese.

Des Weiteren gab es etliche Elemente wie das „Person“-Element, das sowohl reine Textdaten, Unterelemente wie „Pers_Name“ oder beliebige Kombinationen aus Textdaten und Unterelementen enthalten kann. Diese Freiheit war bei der Erstellung der Findmittel-DTD nötig, da die zu konvertierenden Findbücher in meist sehr unterschiedlichen Strukturierungen vorlagen. Da Aster nicht zur Retrokonversion eingesetzt wird, muss auf solche Belange keine Rücksicht genommen werden. Aus diesen Gründen war es nötig, die Findmittel-DTD zu modifizieren.

In der vorliegenden vorläufigen DTD für Aster wurden in diesen Fällen meist alle Kind-Elemente entfernt und es blieb ein einfacher PCDATA-Eintrag. Das wird sich mittelfristig wieder zugunsten einer stärkeren Strukturierung ändern! Einige Elemente, wie z. B. solche, die nur für retrokonvertierte Findmittel gebraucht wurden, wie das „Inhaltsverzeichnis“-Element, wurden ganz entfernt.

Trotz der Modifikation erlaubt die DTD z.B. immer noch multiple „Provenienz“-Elemente, aber da dies ein laufender Arbeitsprozess ist, sind die Änderungen noch nicht vollständig abgeschlossen.

Es werden auf jeden Fall noch Elemente für die Erschließung von Druckschriften hinzukommen, damit auch die Archivbibliothek in diesem Programm erfasst werden kann. Dies bietet den Vorteil, dass der Nutzer bei einer Recherche nicht nur die Archivalien findet, sondern auch passende Buchtitel aus der Archivbibliothek³⁶. Weiterhin soll die DTD noch um Elemente für die Anzeige von Bildern ergänzt werden, so dass z.B. zu jeder Urkunde auch gleich ein Bild derselben und der angehängten Siegel einsehbar ist.

Für einen Vergleich der Findmittel-DTD und der vorläufigen DTD für Aster siehe Anhang B. Für Aster gibt es zwei DTDs, eine für die interne Verarbeitung der Dokumente und eine als Speicherformat. Der Unterschied liegt darin, dass in der internen DTD alle ID-Attribute in PCDATA-Attribute umgewandelt wurden und stattdessen jedes Element ein ID-Attribut „idstef“ erhalten hat. Damit ist es möglich, intern jedes Element mittels dieses Attributes direkt anzusteuern und auszulesen.

Für die Bestandsverwaltung wird eine modifizierte Version der ArchivDTD.dtd von Dr. Przigoda vom Deutschen Bergbau-Museum Bochum verwendet. Dr. Przigoda ist auch einer der „Väter“ der Findmittel-DTD.

³⁶ Vgl. Der Archivar 59 (2006), Sp. 172-176

Die ArchivDTD.dtd wird eigentlich über die Findmittel-DTD „gestülpt“ und erlaubt damit die Abbildung der Tektonik eines Archivs und enthält Angaben zum Archiv. Die wichtigsten Änderungen für Aster sind hierbei die Einführung von ID-Attributen für wichtige Elemente und die Trennung von Tektonik und Findmittel³⁷.

2.5. Auswahl und Vorstellung der Programmiersprache

2.5.1. Auswahl der Programmiersprache

Um eine Software zu erstellen muss natürlich zu allererst eine geeignete Programmiersprache ausgewählt werden. Davon gibt es ja nun mehr als genug, so dass eine Auswahl erstmal schwer fällt. Die Programmiersprachen, die für unseren Zweck in Frage kommen, werden im Allgemeinen „höhere Programmiersprachen“ genannt. Diese werden nochmals in Compiler- und Scriptsprachen unterteilt.

Compilersprachen, zu denen z.B. C zählt, werden so genannt, weil der Quellcode des Programms zuerst mittels eines Compilers in Maschinencode gewandelt wird, bevor das Programm benutzt werden kann. Diese kompilierten Programme haben den Vorteil, dass sie eigenständig sind³⁸, aufgrund der Kompilation in Maschinencode eine hohe Arbeitsgeschwindigkeit aufweisen und wenige Ressourcen verbrauchen.

Bei Scriptsprachen wie Perl und PHP dagegen wird der Quellcode erst während der Ausführung des Programms (parallel zur Laufzeit) mittels eines Interpreters in Maschinencode gewandelt. Dies verlangsamt natürlich die Arbeitsgeschwindigkeit und erhöht den Ressourcenverbrauch, da mehr Code im Speicher abgelegt wird. Da die Leistungsfähigkeit heutiger PCs aber bereits dermaßen enorm ist, und bei der Erschließung von Archivalien hauptsächlich reine Textdaten entstehen (die je nach Bestandsumfang zugegebenermaßen beachtliche Größen erreichen können), sind diese Kriterien für die Auswahl einer Programmiersprache in diesem Fall zu vernachlässigen.

Scriptsprachen sind gegenüber den Compilersprachen auch leichter zu erlernen und – vor allem Perl und PHP – hervorragend für die Verarbeitung von Textdaten geeignet. Weitere

³⁷ In der ArchivDTD.dtd wird das Findmittel direkt eingehängt. Das heißt, das gesamte (!) Archiv mit allen Beständen und Findmitteln befindet sich in einer einzelnen XML-Datei. Für den Datenaustausch mag das praktisch sein, für Arbeitszwecke ist es zuviel des guten, da diese Datei bei entsprechend großen oder vielen Beständen sehr groß werden kann.

³⁸ Bei der Kompilation wird eine ausführbare Datei (*.exe) generiert, die prinzipiell ohne weitere Abhängigkeiten von anderer Software lauffähig ist. Es ist natürlich trotzdem möglich, dass das Programm zur ordnungsgemäßen Erfüllung seiner Aufgabe von anderer Software abhängig ist.

Vorteile sind unter anderem die plattformübergreifende Benutzbarkeit der Scripte, die Möglichkeit sowohl Internet- als auch Desktop-Anwendungen zu programmieren, und große Anhängerschaften sowohl im professionellen wie auch im privaten Einsatzbereich.

Für Aster fiel die Entscheidung, eine Desktop-Anwendung zu programmieren, aus folgenden Gründen: Desktop-Anwendungen sind erheblich flexibler und bieten mehr Funktionalitäten als Web-Anwendungen. Außerdem kann - zumindest bei den kleinen Archiven, die ja zudem meist „Ein-Mann-Archive“ sind – nicht davon ausgegangen werden, dass diese überhaupt an ein Netzwerk in irgendeiner Form angeschlossen sind. Bei diesen müsste dann z.B. noch auf dem Rechner ein Webserver installiert werden etc. Alles in allem rechtfertigt der Nutzen den Aufwand dafür nicht. Natürlich kann eine Web-Oberfläche für Aster – aufgrund der gewählten Programmiersprache – bei Bedarf ohne weiteres erstellt werden.

Der größte Unterschied zwischen Desktop-Anwendungen und Web-Anwendungen ist die Verarbeitung der Benutzereingaben. Web-Anwendungen sind linear aufgebaut, d.h. sie folgen einem bestimmten Ablaufplan, dem sich der Benutzer nicht entziehen kann und es muss bei jeder Änderung der angezeigten Daten immer die gesamte Seite neu aufgebaut werden.³⁹

Desktop-Anwendungen basieren im Gegensatz dazu auf Signalen („Events“) und führen je nach gesendetem Signal unterschiedliche Aktionen aus. Das heißt, der Benutzer ist nicht an einen starren Ablauf gebunden, sondern entscheidet selbst, was er gerade machen möchte. Außerdem wird das grafische System nur einmal aufgebaut und dann immer nur – entsprechend den Benutzereingaben – der betreffende Teil der Anwendung aktualisiert.

Letztendlich fällt die Wahl auf PHP, da es im Vergleich zu Perl einfacher zu erlernen und programmieren, aber genauso leistungsfähig ist. Scripte und Codebeispiele sind in unzähligen Mengen und zu allen möglichen Problemen im Netz frei erhältlich und bei Problemen bekommt man in den Internetforen schnell und unkompliziert Hilfe. Dies alles sollte man bei der Programmierung von Open-Source-Software durchaus beachten, sonst könnte es passieren, dass man eine Software entwickelt, die gerne heruntergeladen und benutzt wird, aber keine aktive Anhängerschaft hervorbringt, die neue Ideen oder sogar Programmfunktionen beisteuert und somit irgendwann zum Scheitern verurteilt ist⁴⁰.

³⁹ Durch die Entwicklung von AJAX (Asynchronous JavaScript and XML) ist es mittlerweile möglich, ganz gezielt nur bestimmte Teile von Web-Seiten zu aktualisieren, wodurch das Internet zwar benutzerfreundlicher wird, aber auch der Datendurchsatz und somit die Netzlast erheblich steigen.

⁴⁰ z.B. wenn der Entwickler noch anderweitige Verpflichtungen hat.

2.5.2. PHP 5

2.5.2.1. Entwicklung, Einsatzmöglichkeiten und Vorteile von PHP⁴¹

PHP ist eine Open-Source-Scriptsprache, die in erster Linie auf einem Webserver eingesetzt wird, um dynamische und interaktive Webseiten zu erstellen. PHP wird derzeit auf ca. 19 Millionen Webdomänen eingesetzt⁴². PHP steht für das rekursive Akronym **PHP: Hypertext Preprozessor** und wurde eigentlich nur für das Web konzipiert.

Die erste Fassung stammt aus dem Jahr 1994 von Rasmus Lerdorf und hieß noch „**Personal Home Page Tools**“. 1995 wurde der Interpreter neu geschrieben und unter dem Namen PHP/FI veröffentlicht. FI stand hierbei für Form Interpreter und bedeutete, dass nun auch HTML-Formulare ausgewertet werden konnten. Ab 1997 arbeitete Lerdorf nicht mehr allein an PHP, sondern mit ihm eine Gruppe von Entwicklern, so dass immer schneller immer neue Funktionen zur Verfügung standen. 1998 wurde Version 3 und 2000 Version 4 freigegeben.

Mit PHP 4 kam auch der Begriff ZEND auf. ZEND ist ein Kunstwort, das sich aus den Namen zweier PHP-Entwickler, nämlich **Zeev** Suraski und **Andi** Gutmans, zusammensetzt.

2004 wurde PHP 5 veröffentlicht und um einige wichtige Funktionalitäten erweitert. So ist es nun möglich, mit PHP auch objektorientiert zu programmieren und die Verarbeitung von XML-Dokumenten via DOM, XSLT, XPATH, etc. wurde entsprechend den W3C-Standards implementiert.⁴³

Die Vorzüge von PHP liegen auf der Hand: PHP ist schnell, stabil, gut ausgestattet und einfach zu erlernen. Für eine komplette Übersicht über die Ausstattung von PHP sei hier auf das PHP-Handbuch verwiesen, das zyklisch erweitert wird.⁴⁴ Weiterhin ist PHP plattformübergreifend einsetzbar und es ist frei erhältlich. Außerdem gibt es im Internet unzählige Seiten, auf denen Erweiterungen, Tutorials, Berichte und Codebeispiele downloadbar sind.

⁴¹ Dittrich, Helmut: PHP 4 – Workshop, S. 11-22

⁴² www.dynamicwebpages.de/60.php-statistiken.php

⁴³ Dies ist besonders für diese Arbeit von Interesse..

⁴⁴ <http://www.php.net/manual/de/>

2.5.2.2. PhpGtk⁴⁵

PHPGtk+ stellt eine Erweiterung von PHP dar. Damit ist es möglich Desktop-Anwendungen (GUI-Anwendungen) mit PHP zu realisieren. Da PHPGtk+ clientseitig arbeitet, wird dafür kein Webserver oder –Browser benötigt. Jedoch muss auf dem entsprechenden Rechner PHP eingerichtet sein. PHPGtk+ liegt Gtk+-Bibliothek zugrunde. Gtk+ (GIMP-Toolkit) ist eine Bibliothek, die für das Programm The GIMP (GNU Image Manipulation Programm) entwickelt wurde⁴⁶. Die Gtk+-Bibliothek ist mittlerweile für alle Betriebssysteme verfügbar, womit eine plattformübergreifende Programmierung möglich ist. Eigentlich wurde Gtk+ für C entwickelt, aber seit einiger Zeit liegen auch Anbindungen für weitere Programmiersprachen wie Python, Perl und PHP vor.

Da von Gtk+ zwei verschiedene Versionen, Gtk+ und Gtk2+, erhältlich sind, gibt es auch für PHP zwei Versionen: PHPGtk+⁴⁷ und PHPGtk2+⁴⁸. PHPGtk+ besteht aus PHP 4 und Gtk+. Es wird durch die Mängel des PHP 4-Objektmodells geplagt und wird mittlerweile kaum noch erweitert. PHPGtk2+ besteht aus Gtk2+ und PHP 5. Es ist zwar derzeit noch in der Entwicklung, d.h., es gibt momentan nur Alpha-Releases⁴⁹, in denen noch nicht alle Gtk2+-Funktionen abgebildet sind und die Dokumentation ist noch im Aufbaustadium.⁵⁰

Die Entscheidung fiel trotzdem zugunsten von PHPGtk2+ aus, da es zum einen mit PHP 5 einsetzbar ist und zum anderen bereits jetzt einen sehr guten Eindruck in Hinsicht auf seine Möglichkeiten macht⁵¹. Außerdem gibt es bereits jetzt sehr aktive Mailinglisten und Foren, in denen sehr fleißig diskutiert wird, und in denen man auf jeden Fall Hilfe bekommt, wenn die Dokumentation versagt.⁵²

⁴⁵ <http://www.php-center.de/artikel/php-gtk-primer.php3>, Mattocks: Pro PHP-GTK

⁴⁶ www.gtk.org

⁴⁷ <http://www.phpgtk.de/>, <http://gtk.php.net/>

⁴⁸ <http://php-gtk2.de/>, <http://www.gnope.org>

⁴⁹ Bis zur Fertigstellung der Software wird es auch einen stabilen Beta-Release geben, schlimmstenfalls muss der Programmcode nochmals ein wenig angepasst werden.

⁵⁰ <http://php-gtk2.de/manual/en/html/>

⁵¹ Zum Beispiel das Anzeigen von Grafiken im JPG-, PNG- oder TIF-Format, und in Bezug auf Gtk+ erheblich verbesserte Features.

⁵² Maillisten: <http://marc.theaimsgroup.com/> (unter GUI befinden sich dann 4 PHPGtk-Maillisten), <http://news.php.net/> (dort lassen sich in einer Liste 6 Maillisten zu PHPGTK finden); Die Maillisten befassen sich sowohl mit PHPGtk+ als auch mit PHPGtk2+.

Foren: www.gnope.org/forum

2.6. Fazit

Bevor es im zweiten Teil der Arbeit um eine erste praktische Umsetzung der Software geht, soll noch einmal zusammengetragen werden, welche Funktionen die Software unterstützen sollte, und wie der Ablauf sein sollte.

Als Programmiersprache wurde PHP 5 ausgewählt, da dies eine einfach zu erlernende Sprache ist, welche bereits in sehr vielen Projekten – sowohl im kommerziellen als auch im Open-Source-Bereich – eingesetzt wird. Weiterhin wurde entschieden, dass die Software als Desktop-Programm und nicht als Web-Anwendung eingesetzt werden soll. Einer späteren Umsetzung als Web-Anwendung steht aufgrund der gewählten Sprache natürlich nichts entgegen. Um die Benutzeroberfläche zu erstellen, wird PhpGtk2+ verwendet. Die Datenhaltung erfolgt in XML, wobei als Grundlage modifizierte Versionen der Findmittel-DTD des Retrokonversionsprojekts der DFG und der Archiv-DTD von Dr. Przigoda vom Deutschen Bergbaumuseum Bochum verwendet werden. Die dabei entstandenen DTDs sind zur Zeit von vorläufiger Art und werden noch ausführlich überarbeitet.

Der Aufbau der Software soll modular erfolgen. Das heißt, es gibt mehrere Programme, die eigenständig lauffähig sind, und bei Bedarf Daten miteinander austauschen können. Um die Kommunikation zwischen den Anwendungen zu ermöglichen, muss die Software objekt-orientiert programmiert werden. Für die praktische Umsetzung im Rahmen dieser Arbeit wird darauf erst einmal aus Zeitgründen verzichtet. Ebenso wird in Hinsicht auf die Zielgruppe (kleine und mittlere Archive) und der Einfachheit halber erst einmal eine Einzelplatzversion umgesetzt, eine Netzwerkversion soll es aber später auch geben.

Folgende Module sind derzeit geplant:

1. Erschließungsmodul
2. Recherchemodul
3. Zwischenarchiv
4. Archivverwaltung

Da das Erschließungsmodul das „Herzstück“ der Software ist und zusammen mit dem Recherchemodul auch das wichtigste, sollen diese beiden zuerst fertig gestellt werden.

Das Erschließungsmodul, welches in Ansätzen im zweiten Teil der Arbeit umgesetzt wird, soll dabei die Verwaltung der Archivtektonik, der Bestände und der dazugehörigen Findmittel

bereitstellen. Dazu gehören fundamentale Aufgaben wie das Hinzufügen, Ändern und Löschen von Informationen, aber auch der Datenimport und -export, das Suchen und Ersetzen von Zeichenketten, das Erstellen von Indizes und Konkordanzen, das Generieren von Findbüchern und das Sperren von Beständen, Findmitteln und einzelnen Verzeichnungseinheiten für verschiedene Benutzergruppen oder Berechtigungsstufen.

Für die Umsetzung der Benutzerverwaltung wird wahrscheinlich das Pear-Paket „LiveUser“⁵³ eingesetzt werden. LiveUser stellt einige Klassen für PHP zur Verfügung, mit deren Hilfe sich die Autorisierung und Authentifizierung von Benutzern realisieren lassen. LiveUser unterstützt dabei auch die Bildung von Gruppen und Untergruppen, sodass sich damit auch ein sehr komplexes Rechtemanagement realisieren lässt.

Das Recherchemodul soll je nach gerade angemeldetem Benutzer und dessen Berechtigungsstufen nur in den erlaubten Daten suchen und Fundstellen ausgeben. Ebenso muss der Benutzer persönliche Auswahllisten (z.B. Bestelllisten) erstellen können, die in verschiedenen Formaten (z.B. in XML, HTML, als Text oder als Excelliste) speicherbar sein müssen. Es wäre zu überlegen, dass Recherchemodul von vornherein als reine Webanwendung zu konzipieren, da die Internetrecherche auch bei den Archiven immer mehr an Gewicht zunimmt.

Ebenso sollte das Recherchemodul verschiedene Suchmodi anbieten, um dem Benutzer genauere Spezifikationen der Suche zu erlauben.

Zum Modul Zwischenarchiv muss wohl nicht viel gesagt werden, hier ist vor allem die automatische Fristenüberwachung wichtig, und die Festlegung, welche Daten endgültig in das Archiv übernommen werden und welche nicht. Dazu gehört auch die Generierung von Kassationslisten und ähnlichem. Ansonsten ist auch hier auf eine Möglichkeit zum Datenimport- und -export zu achten. Möglicherweise sollte, da sowohl das Erschließungsmodul als das Zwischenarchivmodul Daten importieren und exportieren können müssen, ein eigenständiges Import/Export-Modul programmiert werden.

In das Modul Archivverwaltung gehören letztendlich allgemeine Aufgaben wie die Adressverwaltung, Erstellung von Schadenslisten, die schriftliche Auskunftserteilung und die Verwaltung der Akzessionen.

⁵³ <http://oss.backendmedia.com/LiveUser/>

3. Programmierung einer Open-Source-Archivsoftware

Nach dem Abschluss der Vorbereitungen (Datenhaltung, Auswahl der Programmiersprache etc.) folgt nun der praktische Teil der Arbeit. Ziel ist es hierbei, das grafische System des Verzeichnungsmoduls weitestgehend aufzubauen und vorhandene Beispieldaten anzeigen zu lassen. Im Folgenden geht es hier primär um die technischen Aspekte, für eine einfache Bedienungsanleitung schauen Sie bitte in den Anhang A.

Eine Übersicht über den Programmablauf können Sie dem folgenden Programmablaufplan entnehmen (Abb.1). Dies ist natürlich nur eine stilisierte Ansicht des Programms und zur besseren Verständlichkeit habe ich den Ablauf in 5 Teilschritte unterteilt.

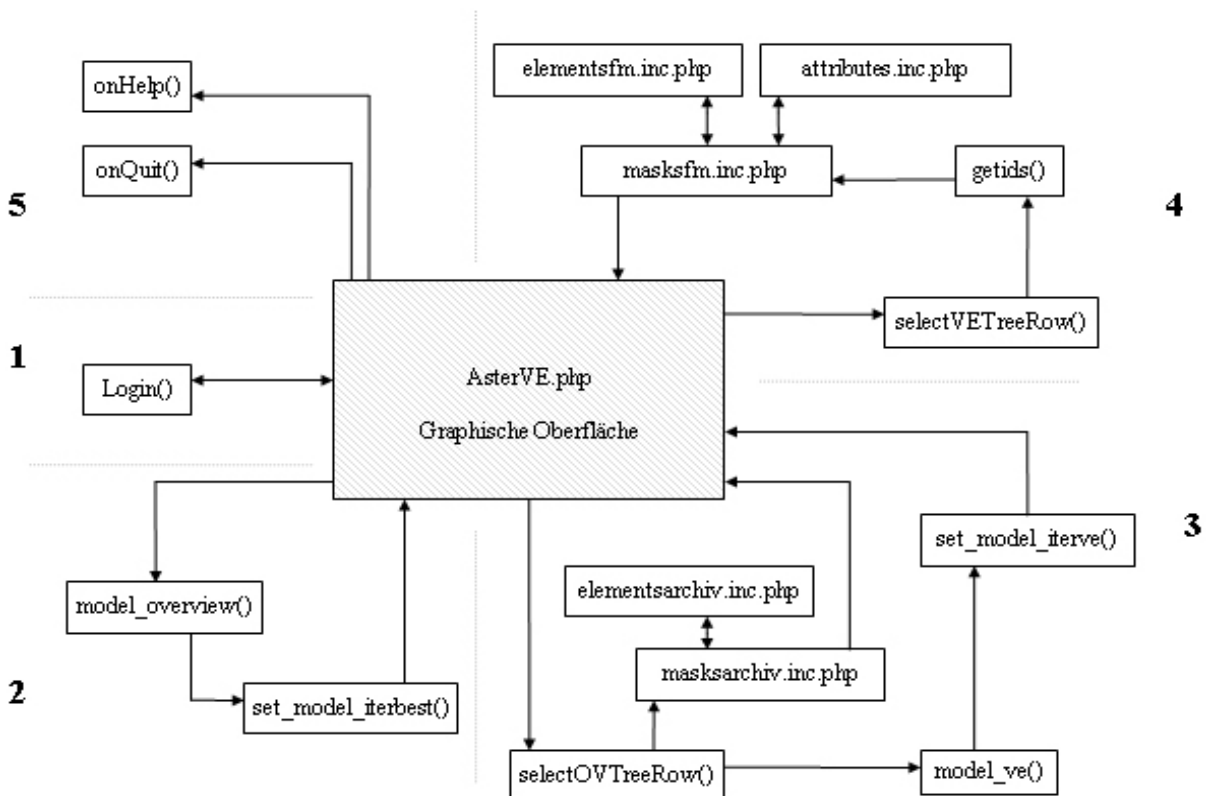


Abbildung 1: Programmablaufplan

Im ersten Schritt wird das grafische System aufgebaut und die Zugangsdaten abgefragt.

Im zweiten Schritt werden die Archivtektonik-Daten eingelesen, um im dritten angezeigt zu werden. Im dritten erfolgt auch das Einlesen der Findmittel.

Der vierte Schritt dient der Anzeige von Findmittel-Daten und der fünfte gibt die Menü-Möglichkeiten wieder.

3.1. Aufbau des grafischen Systems und Abfragen der Zugangsberechtigungen

Wenn das Programm gestartet wird, wird zuerst das grafische System aufgebaut. Im Folgenden sehen Sie den dafür relevanten Quellcode. Dieser liegt in der Datei AsterVE.php.

```
include("functions.inc.php");
...
$userid="";
$state="";
$vedom;
$vefile;
$overviewdom;
$overviewfile;

$window = new GtkWindow();
$pixbuf = GdkPixbuf::new_from_file('img/astericon.png');
$window->set_default_icon($pixbuf);
$window->set_title("Aster");
$window->set_position('GTK_WIN_POS_CENTER');
$window->set_default_size(800,600);
$window->connect_simple("destroy", "onQuit");
$window->show();

$vbox = new GtkVBox(false, 5);

$MenuBar = new GtkMenuBar;
$menuFile = new GtkMenuItem('_Datei');
$subMenuFile = new GtkMenu;
$itemFile1 = new GtkImageMenuItem(GTK::STOCK_SAVE);
$itemFile1->connect('activate','savefile');
...
$subMenuFile->append($itemFile1);
...
$menuFile->set_submenu($subMenuFile);
$MenuBar->append($menuFile);
...
$vbox->pack_start($MenuBar, false, false, 0);
...

$shpane = new GtkHPaned();
$shpane->set_position(250);
$spane = new GtkVPaned();
$spane->set_position(250);
$vboxoverview = new GtkVBox();

$bstree = new GtkTreeView;
$bstmodel = new GtkTreeStore(Gtk::TYPE_OBJECT, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
$bstcolumn1 = new GtkTreeViewColumn();
$bstcolumn1->set_title('Tektonik');
$bstcell_renderer1 = new GtkCellRendererPixBuf();
$bstcell_renderer2 = new GtkCellRendererText();
$bstcolumn1->pack_start($bstcell_renderer1, false);
$bstcolumn1->pack_start($bstcell_renderer2, true);
$bstcell_renderer2->set_property('width', 200);
```

```

$bestcolumn1->set_attributes($bestcell_renderer1, 'pixbuf', 0);
$bestcolumn1->set_attributes($bestcell_renderer2, 'text', 1);
$besttree->append_column($bestcolumn1);
$bestcolumn2 = new GtkTreeViewColumn();
$bestcolumn2->set_title('Ids');
$bestcell_renderer2 = new GtkCellRendererText();
$bestcolumn2->pack_start($bestcell_renderer2, true);
$bestcell_renderer2->set_property('width', 0);
$bestcolumn2->set_attributes($bestcell_renderer2, 'text', 2);
//$besttree->append_column($bestcolumn2);
$besttree->set_model($bestmodel);
$selection = $besttree->get_selection();
$selection->connect('changed', 'selectOVTreeRow');
$besttree->connect_after('event', 'clickedTree');
$besttree->set_events(Gdk::_2BUTTON_PRESS | Gdk::BUTTON_RELEASE);

model_overview();

$bestscroll = new GtkScrolledWindow();
$bestscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bestscroll->add($besttree);
$bestscroll->show_all();

$ vboxoverview->pack_start($bestscroll, true, true, 0);
$ vpane->add1($ vboxoverview);
$ vboxve = new GtkVBox();
$ vtree = new GtkTreeView;
$ vmodel = new GtkTreeStore(Gtk::TYPE_OBJECT, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
$ vcolumn1 = new GtkTreeViewColumn();
$ vcolumn1->set_title('Findmittel');
$ vcell_renderer1 = new GtkCellRendererPixBuf();
$ vcell_renderer2 = new GtkCellRendererText();
$ vcolumn1->pack_start($ vcell_renderer1, false);
$ vcolumn1->pack_start($ vcell_renderer2, true);
$ vcell_renderer2->set_property('width', 200);
$ vcolumn1->set_attributes($ vcell_renderer1, 'pixbuf', 0);
$ vcolumn1->set_attributes($ vcell_renderer2, 'text', 1);
$ vtree->append_column($ vcolumn1);
$ vcolumn2 = new GtkTreeViewColumn();
$ vcolumn2->set_title('Ids');
$ vcell_renderer2 = new GtkCellRendererText();
$ vcolumn2->pack_start($ vcell_renderer2, true);
$ vcell_renderer2->set_property('width', 0);
$ vcolumn2->set_attributes($ vcell_renderer2, 'text', 2);
//$ vtree->append_column($ vcolumn2);
$ vtree->set_model($ vmodel);
$ selection = $ vtree->get_selection();
$ selection->connect('changed', 'selectVETreeRow');
$ vtree->connect_after('event', 'clickedTree');
$ vtree->set_events(Gdk::_2BUTTON_PRESS | Gdk::BUTTON_RELEASE);
$ vscroll = new GtkScrolledWindow();
$ vscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$ vscroll->add($ vtree);
$ vscroll->show_all();
$ vboxve->pack_start($ vscroll, true, true, 0);
$ vpane->add2($ vboxve);
$ hpane->add1($ vpane);

$ vboxmask = new GtkVBox();

```

```
$note = new GtkNotebook();
$ vboxmask->pack_start($note, true, true, 0);
$ hpane->add2($ vboxmask);
$ vbox->pack_start($ hpane, true, true, 0);

$ window->add($ vbox);
$ window->show_all();
login();
gtk::main();
```

Im ersten Block werden die weiteren Dateien mittels `include()` eingebunden, und dann einige globale Variablen definiert. `$userid` enthält den Namen des angemeldeten Benutzers, `$state` enthält die ID des Elements, das sich der Benutzer gerade ansieht. `$vfile` enthält den Pfad zur aktuellen Findmitteldatei, `$vedom` den geparsten XML-Baum. Ebenso verhält es sich mit `$overviewdom` und `$overviewfile`.

Danach geht es mit dem eigentlichen Aufbau des Systems los.⁵⁴ Es wird ein neues `GtkWindow` (`$window`) erstellt. `GtkWindow` ist das Hauptfenster einer jeden `Gtk`-Anwendung. Da es bei `Gtk`-Anwendungen einige Widgets⁵⁵ gibt, die nur ein Kind-Widget enthalten können (z.B. das `GtkWindow`), wird nun zunächst eine `GtkVBox` (`$vbox`) erstellt, welche sämtliche weiteren Widgets aufnehmen wird und dem Hauptfenster mittels „`$window->add($vbox);`“ hinzugefügt.⁵⁶

Dann wird nun die Menüleiste (`$MenuBar`) erstellt und die Menüs (`$SubMenuFile`) die Menüitems (`$MenuFile`) eingehängt.

Dann erfolgt das Erstellen der Navigationsbäume und des Maskenfensters. Dazu wird zuerst ein `GtkHPaned` erstellt, welcher zwei Kindelemente aufnehmen kann und dazwischen einen verschiebbaren Trenner setzt. Auf der linken Seite, wo sich die beiden Bäume befinden, wird parallel dazu ein `GtkVPaned` eingehängt, welcher die beiden Bäume aufnimmt. Damit sind die Fenstergrößen dieser drei Elemente nach Belieben skalierbar.

Die Navigationsbäume (`$besttree` und `$vetree`) basieren auf eine `GtkTreeView`-Komponente. `GtkTreeView` kann sowohl Baumstrukturen (`GtkTreeStore`) als auch einfache Listen (`GtkListStore`) darstellen. `GtkTreeView` ermöglicht eine Abgrenzung von Daten-

⁵⁴ Für die genaue Verwendung von Widgets und deren Eigenschaften siehe `PhpGtk-Handbuch` unter <http://php-gtk2.de/manual/en/html/>

⁵⁵ Widget ist ein Begriff, der in der X-Window-Welt für jegliche Benutzerschnittstelle gebraucht wird. Ein Widget kann eine Textbox sein, eine Beschriftung, ein Rahmen, ein Fenster oder irgendein anderer GUI-Bestandteil.

⁵⁶ Man hätte an Stelle der `VBox` auch eine `HBox`, ein `GtkTable` oder `GtkLayout` verwenden können. Eine `VBox` packt alle Kindelemente vertikal untereinander, `HBox` packt alle analog dazu vertikal nebeneinander, `GtkTable` packt alle in eine Tabelle und `GtkLayout` positioniert alle Elemente fest an die vorgegebene Stelle. Im Gegensatz zu `GtkLayout` und `GtkTable` ist die Arbeit mit vertikalen und horizontalen Boxen erheblich flexibler, weshalb ich hauptsächlich diese beiden verwende.

Präsentations- und Steuerungsschicht nach dem MVC-Muster (Model-View-Controller). Das heißt, das Datenmodell (GtkTreeStore oder GtkListstore) ist von der Darstellung auf dem Bildschirm getrennt. Damit ist es zum Beispiel möglich, ein und dieselben Daten an verschiedenen Stellen der Anwendung unterschiedlich darzustellen.

Die beiden hier benötigten Modelle (\$bestmodel und \$vemodel) enthalten jeweils drei Felder. Das Erste ist vom Typ GtkObjekt, d.h. dieses Feld kann nur Objekte – wie in unserem Fall Bilder (die Icons) aufnehmen, das Zweite und das Dritte nehmen nur Strings auf. Die Modelle werden dann dem GtkTreeView entsprechenden TreeView hinzugefügt. Das Einfügen der Daten erfolgt später mittels der Funktion `model_overview()`⁵⁷. Als nächstes werden jeweils zwei CellRenderer (\$bestcell_renderer1, \$bestcell_renderer2, \$vecell_renderer1 und \$vecell_renderer2,) instanziiert. CellRenderer sind für das korrekte Darstellen der Daten in den Reihen des TreeView zuständig. Der erste CellRenderer (\$...cell_renderer1) enthält Bilder und der zweite reinen Text. Danach werden die CellRenderer zu den Spalten des TreeView hinzugefügt:

```
$vecolumn1->set_attributes($vecell_renderer1, 'pixbuf', 0);  
$vecolumn1->set_attributes($vecell_renderer2, 'text', 1);
```

Spalte 1 (\$bestcolumn1 und \$vecolumn1) erhält dabei beide CellRenderer, die Reihenfolge der Datenpräsentation legt der letzte Parameter fest.

Ebenso wird der zweite CellRenderer zur zweiten Spalte (\$...column2) hinzugefügt. Diese Spalte enthält später die ID-Nummern der Elemente und wird nicht angezeigt. Die TreeViews werden jeweils mit den Funktionen „`clickedTree()`“ und „`selectOVTreeRow()`“ bzw. „`selectVETreeRow()`“ verbunden.⁵⁸ Die Funktion „`clickedTree()`“ sorgt dafür, das Mausklicks in diesem Widget abgefangen und ausgewertet werden.⁵⁹ Ein Linksklick zeigt die darunterliegende Hierarchieebene an, ein Rechtsklick schließt den gesamten Knoten wieder und ein Doppelklick oder (falls vorhanden) ein Klick mit dem mittleren Mausbutton öffnet alle tieferen Hierarchieebenen des Knotens.

Die Funktionen „`selectOVTreeRow()`“ und „`selectVETreeRow()`“ werden in Abschnitt 2.3. bzw. 2.4. genauer behandelt.

⁵⁷ Diese Funktion wird hier bereits aufgerufen und in Abschnitt 2.2. erläutert.

⁵⁸ Diese drei Funktionen wurden dem Programm „`Var_Dump`“ von Christian Weiske entnommen und – wo nötig – modifiziert. Das Programm erlaubt die Analyse von Gtk2-Objekten parallel zur Laufzeit und ist unter http://cweiske.de/phpgtk2_var_dump.htm zu finden.

⁵⁹ In diese Funktion wird später auch das Popupmenü eingehängt, mit dessen Hilfe neue Elemente dem Baum hinzugefügt werden sollen, bzw. vorhandene verschoben oder gelöscht werden können.

Zum Schluss werden die Bäume einem GtkScrolledWindow hinzugefügt (\$bestscroll und \$vescroll). Dieses ScrolledWindow sorgt dafür, dass Scrollbalken angezeigt werden, wenn die Daten mehr Platz brauchen, als der sichtbare Bereich des TreeView zur Verfügung stellt.

Auf der rechten Seite des GtkHPaned wird noch ein GtkNoteBook (\$note) instanziiert, in welchem später die Erfassungsmasken angezeigt werden.

“\$window->show_all();” sorgt abschließend dafür, dass alle Kindelemente von \$window auf dem Monitor dargestellt werden. „gtk::main();“ beendet die Hauptschleife des Programms.

Nachdem das grafische System aufgebaut wurde wird die login()-Funktion aufgerufen. In dieser wird ein GtkDialog instanziiert, der zwei GtkEntry-Widgets enthält, in die der Benutzername und das Passwort eingegeben werden können. Diese Funktion ist noch fehlerbehaftet, aber da eine ordentliche Benutzerverwaltung noch aussteht, ist sie im Moment völlig ausreichend. Aufgrund der fehlenden Benutzerverwaltung ist auch die Eingabe eines Passworts nicht nötig, nur das Benutzer-ID-Feld erwartet einen Eintrag, der aber auch noch nicht geprüft wird! Diese Funktion wurde erstmal nur eingebaut, um den Betrieb im Netzwerk zu testen.⁶⁰

3.2. Aufbau des Tektonikbaumes

Nach dem Aufbau des grafischen Systems müssen nun die Tektonik-Daten ausgelesen und dargestellt werden. Dazu wird von der Datei „AsterVE.php“ die Funktion „model_overview()“ aufgerufen, welche sich in der Datei „functions.inc.php“ befindet.

In dieser Funktion wird zuerst die globale Variable \$overviewfile „eingeladen“, und sie erhält den Pfad der XML-Datei, in der die Tektonik aufbewahrt wird (archiv.xml). Dann wird die Datei geparkt⁶¹ und der so entstandenen Dokumentenbaum in der globalen Variable \$overviewdom gespeichert. Danach wird das

```
function model_overview(){
    global $overviewfile;
    $overviewfile="data/archiv.xml";
    global $overviewdom;
    $overviewdom=load_xml($overviewfile);
    $root=$overviewdom->documentElement;
    set_model_iterbest($root, null);
    global $besttree;
    $besttree->show_all();
}
```

⁶⁰ So kann die Software auf einem Netzwerkrechner installiert werden, und mehrere Benutzer aus dem Netzwerk diese starten. Wenn ein Nutzer einen Eintrag wählt, den ein anderer gerade benutzt, so wird ein Sperrvermerk ausgegeben. Dies funktioniert natürlich auch bei zwei laufenden Instanzen des Programms auf einem System. Wenn zwei Instanzen mit demselben Nutzernamen laufen, können beide parallel auf dieselbe Maske zugreifen, aber eigentlich sollte dieser Fall ja nicht eintreten. Zur Funktionsweise dieser Sperrung siehe auch Abschnitt 2.2. Diese Lösung ist aber nur eine Zwischenlösung, bis die Benutzerverwaltung erstellt ist.

⁶¹ Die Funktionen load_xml(), tag_exists() und get_tags() wurden dem Buch „PHP & XML für Web Developer“ von M. Seeboerger-Weichselbaum entnommen.

Wurzelement (\$root) ermittelt und die Funktion „set_model_iterbest()“ aufgerufen, welche dem Modell (\$bestmodel) Daten hinzufügt. Die übergebenen Werte sind das Wurzelement (und alle darunterliegenden Knoten – sprich der Dokumentenbaum⁶²) und der Wert „null“ – dieser gibt die Hierarchieebene des GtkTreeView wieder. Es ist leider nicht möglich, \$overviewdom direkt zu übergeben. Zum Schluss wird der TreeView, der die Tektonik präsentiert aktualisiert.

Die Funktion „set_model_iterbest()“ ist eine rekursive Funktion, d.h. sie wandert vom „Startpunkt“⁶³ durch die erste Hierarchieebenen des Dokuments, führt ihre Anweisungen aus und ruft sich – wenn Kindelemente vorhanden sind – erneut auf, bis sie durch das gesamte Dokument gewandert ist.

```
function set_model_iterbest($n, $iter) {
    global $bestmodel;
    global $vmodel;
    global $window;
    $home=$window->render_icon(Gtk::STOCK_HOME, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $directory=$window->render_icon(Gtk::STOCK_DIRECTORY, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $file = $window->render_icon(Gtk::STOCK_FILE, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $multi=$window->render_icon(Gtk::STOCK_DND_MULTIPLE, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $it="";
    $children=$n->childNodes;
    $anz=$children->length;
    for ($a=0; $a<$anz;$a++) {
        $c=$children->item($a);
        if ($c->nodeName == "Archivinfo") {
            $id=$c->getAttributeNode("id")->nodeValue;
            if (hasChild($c, 'Archivname')) {
                $child=getChild($c, 'Archivname');
                $value=$child->nodeValue;
                if (empty($value)){ $value='Archivinfo';}
            } else { $value='Archivinfo';}
            $it=$bestmodel->append($iter, array($home, utf8_decode($value), $id));
        }
        if ($c->nodeName == "Tektonik") {
            $id=$c->getAttributeNode("id")->nodeValue;
            if (hasChild($c, 'Tekt_Nr')) {
                $nrs=getChild($c, 'Tekt_Nr');
                $nr=$nrs->nodeValue;
                if (!empty($nr)){ $nr=$nr.' ';}
            } else { $nr="";}
            if (hasChild($c, 'Tekt_Titel')) {
                $titels=getChild($c, 'Tekt_Titel');
                $titel=$titels->nodeValue;
                if (empty($titel)){ $titel='Tektonik';}
            } else { $titel='Tektonik';}
        }
    }
}
```

⁶² Es ist leider nicht möglich, den Dokumentenbaum (\$overviewdom) direkt zu übergeben, das dieser zur Klasse DomDocument gehört und somit nicht die geeigneten Methoden zur Verfügung stellt, um durch den Baum direkt zu wandern. Die Methoden stehen aber der Klasse DomElement zur Seite, von welchem Typ \$root ist.

Vgl. Seeboerger-Wechselbaum, PHP und XML, S.125 - 160

⁶³ In diesem Fall das Wurzelement

```

        $value=$nr.$titel;
        $sit=$bestmodel->append($iter, array($directory, utf8_decode($value), $id));
        if (hasChild($c, 'Findmitteldatei')) {
            $dat=$c->childNodes;
            for ($b=0; $b<$dat->length;$b++) {
                $d=$dat->item($b);
                if ($d->nodeName=='Findmitteldatei'){
                    $idf=$d->getAttributeNode("id")->nodeValue;
                    $files=$d->nodeValue;
                    if (!empty($files)){
                        $siders= $bestmodel->append($sit, array($multi,
'Findmittel', $idf));

                        $dom=load_xml($files);
                        $infos=get_tags($dom, 'Bestand_Info');
                        $infozahl=$infos->length;
                        $count=1;
                        if ($infozahl>0){
                            for ($e=0; $e<$infozahl;$e++) {
                                $bestinfo=$infos->item($e);
                                if (hasChild($bestinfo, 'Bestand_Sig'))

                                $nrs=getChild($bestinfo, 'Bestand_Sig');
                                $nr=$nrs->nodeValue;
                                if (!empty($nr)){ $nr=$nr.' ';}

                                } else { $nr="";}
                                if (hasChild($bestinfo, 'Bestandsname'))

                                $titels=getChild($bestinfo, 'Bestandsname');
                                $titel=$titels->nodeValue;
                                if (empty($titel)){ $titel='Bestand';}

                                } else { $titel='Bestand';}
                                $value=$nr.$titel;
                                $sids=$idf.'!'.$count;
                                $count++;
                                $bestmodel->append($siders, array($file, $value, $sids));
                                }
                            } else {

                                $sids=$idf.'!'.$count;
                                $bestmodel->append($siders, array($file, 'Bestand', $sids));
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
if ($c->nodeName == "Bestand") {
    $id=$c->getAttributeNode("id")->nodeValue;
    if (hasChild($c, 'Bestand_Sig')) {
        $nrs=getChild($c, 'Bestand_Sig');
        $nr=$nrs->nodeValue;
        if (!empty($nr)){ $nr=$nr.' ';}
    } else { $nr="";}
    if (hasChild($c, 'Bestandsname')) {
        $titels=getChild($c, 'Bestandsname');
        $titel=$titels->nodeValue;
        if (empty($titel)){ $titel='Bestand';}
    } else { $titel='Bestand';}
}

```

```

        $value=$nr.$titel;
        $it=$bestmodel->append($iter, array($file, utf8_decode($value), $id));
        if (hasChild($c, 'Findmitteldatei')) {
            $dat=$c->childNodes;
            $zahl=1;
            for ($b=0; $b<$dat->length;$b++) {
                $d=$dat->item($b);
                if ($d->nodeName=='Findmitteldatei'){
                    $idf=$d->getAttributeNode("id")->nodeValue;
                    $files=$d->nodeValue;
                    if (!empty($files)){
                        $value='Findmittel'.$zahl;
                        $zahl++;
                        $bestmodel->append($it, array($multi,
utf8_decode($value), $idf));
                    }
                }
            }
        }
    }
    if ($c->hasChildNodes()) {set_model_iterbest($c, $it);}
}
}

```

Zuerst werden wieder die benötigten globalen Variablen „eingeladen“. Dann werden nötigen Icons gerendert.

Wenn die Funktion auf ein Archivinfo-Element stößt, liest sie das ID-Attribut und das Archivname-Element aus. Wenn es kein Archivname-Element gibt, oder dieses leer ist, wird stattdessen der String „Archivinfo“ in das Feld eingefügt. Mittels „append“ werden die Daten in der richtigen Reihenfolge in das Modell eingehängt.

Wenn die Funktion auf ein Tektonik-Element trifft, werden die Tektonik-Nummer und der Tektonik-Titel ausgelesen. Falls das Element Kindelemente mit dem Namen „Findmitteldatei“ hat, wird für jede Datei ein Punkt „Findmittel“ in das Modell eingehängt, dann wird jede geparkt, und jeweils eventuelle Bestandssignaturen und Bestandsnamen ausgelesen und die Bestände eingehängt unterhalb des Findmittel-Punkts dem Modell hinzugefügt.

Die ID dieser Bestandsinformationen setzt sich aus der ID des Findmitteldatei-Elements und einer laufenden Nummer zusammen.⁶⁴

Sollte die Funktion ein Bestand-Element finden, so werden wieder ID-Attribut, Signatur- und Namens-Elemente ausgelesen und dem Modell hinzugefügt. Sollte das Bestand-Element noch Findmitteldatei-Elemente haben, wird für jedes ein Punkt „Findmittel“ mit einer zusätzlichen laufenden Nummer eingehängt. Auf das Auslesen der Bestandsinformationen in der

⁶⁴ z.B. „id26-2“

Findmitteldatei kann verzichtet werden, da in einer Findmitteldatei, die eindeutig einem Bestand zugeordnet ist, eigentlich keine weiteren Bestände zu finden sein sollten.

Dieses tiefe Verschachteln der Anzeige ist leider nötig, da die Archiv-DTD als auch die Findmittel-DTD sehr flexibel mit ihren Strukturen umgehen. So kann ein Findmittel mehrere Bestände enthalten, ein Tektonik-Punkt mehrere Findmittel und Bestände und ein Bestand in der Archiv-DTD wiederum beliebig viele Findmittel. Das mag auf den ersten Blick etwas eigentümlich anmuten, ist aber sehr praktisch, da man auf diese Weise viele kleinere und kleinste Bestände zu einem Tektonik-Punkt in einer Findmitteldatei aufbewahren kann oder auch einen großen Bestand auf mehrere Findmittel aufteilen kann. Wie das in der Praxis letztendlich gehandhabt wird, hängt einzig und allein von den Anwendern ab. Für die Recherche ist dieses Unterteilen in einerseits Findmittel und deren enthaltene Bestände, und Bestände und deren enthaltene Findmittel nicht nötig, da den Benutzer nur das schnelle Finden von Informationen in den Beständen interessiert und nicht deren Speicherung im System. Insofern wird das Rechercheprogramm intuitiver sein, aber bei der Verzeichnung spielt auch die Transparenz der Datenhaltung eine Rolle, so dass dieser Weg hier nötig wurde.

3.3. Arbeiten in der Tektonik

Nachdem die vorangegangenen Abschnitte beim Start der Anwendung automatisch abliefen, wird nun eine Benutzeraktion erwartet. Der Benutzer kann zu diesem Zeitpunkt entweder ein Item aus dem Menü wählen (siehe Abschnitt 2.5.) oder mittels des oberen Navigationsbaums durch die Archivtektonik wandern. Wenn der Benutzer einen Eintrag im Baum anklickt, wird zuerst die Funktion „clickedTree()“ ausgeführt⁶⁵ und dann die Funktion „selectOVTreeRow()“. Der Übergabeparameter ist die gewählte Reihe.

```
function selectOVTreeRow($selection) {
    global $overviewfile;
    global $overviewdom;
    global $vemodel;
    global $state;
    global $userid;
    $overviewdom=load_xml($overviewfile);
    if(!$overviewdom->schemaValidate("transform/AsterArchiv.xsd")){
        errorhandle('warning', 'Die Datei "'.$overviewfile.'"entspricht nicht dem Schema
("transform/AsterArchiv.xsd"));
    } else {
        if(!empty($state)== true){
```

⁶⁵ siehe Abschnitt 2.1.

```

        $levels=$overviewdom->getElementById($state);
        $states=$levels->getAttributeNode('state');
            $states->nodeValue="open";
            $user=$levels->getAttributeNode('user');
            $user->nodeValue="";
            $state="";
            $overviewdom->save($overviewfile);
        }
    }
list($model, $iter) = $selection->get_selected();
if ($iter === null) {
    return;
}
$id = $model->get_value($iter, 2);
$count=substr_count($id, '-');
if ($count>0){
    $pos=strrpos($id, '-');
    $posa=$pos+1;
    $beid=substr($id, $posa);
    $id=substr($id, 0, $pos);
} else {$beid=";}
$level=$overviewdom->getElementById($id);
cleartree($vmodel);
if ($level->getAttribute('state') == "locked" AND $level->getAttribute('user') != $userid) {
    $user=$level->getAttribute('user'); mask_used($user);
} else {
    $states=$level->getAttributeNode('state');
    $states->nodeValue="locked";
    $user=$level->getAttributeNode('user');
    $user->nodeValue=$userid;
    $overviewdom->save($overviewfile);
    $state = $id;
    $name=$level->nodeName;
    if ($name== 'Archivinfo'){mask_archivinfo($id);}
    if ($name== 'Tektonik'){mask_tektonik($id);}
    if ($name== 'Bestand'){mask_bestand($id);}
    if ($name== 'Findmitteldatei'){
        $file=$level->nodeValue;
        model_ve($file);
        mask_findmittel($id, $beid);
    }
}
}
}

```

Zu Beginn werden wieder die globalen Variablen initialisiert, dann wird die XML-Datei, die die Tektonik enthält erneut geladen. Dies ist nötig, da die die Information, welcher Benutzer gerade in welchem Element arbeitet direkt in dieser Datei gespeichert wird.⁶⁶ Damit immer der aktuelle Stand verarbeitet wird, muss die Datei deshalb neu geparkt werden.

Dann wird der Dokumentenbaum gegen ein XML-Schema validiert.⁶⁷ Die Datei „AsterArchiv.xsd“ wurde aus der „AsterArchiv.dtd“ generiert, so dass diese beiden Dateien

⁶⁶ genauer: in den Attributen „state“ und „user“.

⁶⁷ PHP unterstützt eigentlich auch die Validierung gegen DTDs, aber aus irgendeinem unerfindlichen Grund, funktionierte diese Funktion bei mehreren Tests nicht.

unterschiedliche Formate derselben Information darstellen. Sollte die Validierung einen Fehler ergeben, wird ein Fehlerdialog ausgegeben (Funktion „errorhandle()“). Wenn die Validierung erfolgreich war, wird geprüft, ob die globale Variable \$state, welche die ID des bisher angezeigten Elements enthält, leer ist⁶⁸. Falls \$state nicht leer ist, wird das entsprechende Element im Dokumentbaum gesucht („\$overviewdom->getElementById(\$state);“) und die benötigten Attribute „state“ und „user“ auf „open“ gesetzt bzw. geleert. Anschließend wird auch \$state geleert.

Anschließend werden aus dem Übergabeparameter das Datenmodell und der Zeiger („Iter“) ausgelesen. Aus dem Datenmodell wird das dritte Feld, welches die IDs enthält, der ausgewählten Zeile ausgelesen („\$id = \$model->get_value(\$iter, 2);“). Dann wird geprüft, ob die ID einen oder mehrere Bindestriche enthält.⁶⁹ Wenn dem so ist, wird diese ID geteilt, so dass in der Variable \$id die eigentliche ID enthalten ist und in der Variable \$beid die Nummer des gewählten Bestandes.⁷⁰

Danach wird das aktuelle Element gesucht und in der Variable \$level gespeichert. Das Modell für den Findmittelbaum wird in der Funktion „cleartree()“ geleert.

Wenn das geschehen ist, wird überprüft, ob die gewählten Informationen bereits von einem anderen Benutzer eingesehen werden. In diesem Fall wird aus der Datei „masksarchiv.inc.php“ die Funktion „mask_used()“ mit dem Namen des anderen Benutzers als Parameter geladen.

Wenn die Informationen frei sind, werden sie zuerst in der „Archiv.xml“ für andere gesperrt (Attribut „state“ wird auf „locked“ gesetzt und Attribut „user“ erhält den Benutzernamen des aktuellen Benutzers), dann wird der Name des Elements ausgelesen und die entsprechende Erfassungsmaske wird in die rechte Bildschirmseite geladen.

Dies geschieht, indem auf eine bestimmte Funktion in der Datei „masksarchiv.inc.php“ weitergeleitet wird. In dieser Funktion werden dann die Unterelemente bestimmt, diese wiederum werden an eine passende Funktion in der Datei „elementsarchiv.inc.php“ weitergeleitet, wo die Knoten ausgelesen werden und die benötigten Widgets erstellt werden. Diese Widgets werden an die Maskenfunktion zurückgeliefert, wo die Masken dann mittels GktVBox- und GtkHBox-Objekten zusammengesetzt werden.

⁶⁸ Eigentlich tritt dieser Fall nur beim Programmstart ein.

⁶⁹ Dies ist der Fall, wenn ein Bestandselement ausgewählt wurde, welches sich in einer Findmitteldatei befindet, die direkt in einen Tektonikpunkt eingehängt wurde. Siehe auch Abschnitt 2.2.

⁷⁰ Damit weiß das Programm später, dass es z.B. bei der übergebenen ID „id26-2“ die Findmitteldatei mit der ID „id26“ parsen muss und die Bestandsangabenmaske mit den Daten aus dem zweiten Bestand_Info-Element aus der Datei anzeigen muss.

Sollte ein Findmitteldatei-Element ausgewählt worden sein, wird zusätzlich zur Maske noch die Findmittelstruktur in den unteren Navigationsbaum eingesetzt. Dazu werden der Dateiname und –pfad (in der Variable \$file) an die Funktion „model_ve()“ übergeben. Dort wird der übergebene String aus von Unicode nach ISO-8859-1 decodiert. Dies ist wichtig für eventuell vorhandene Umlaute.

```
function model_ve($file){
    global $vedom;
    global $vefile;
    $vefile=utf8_decode($file);
    $vedom=load_xml($vefile);
    idattrs($vedom, 'veidadd');
    $root=$vedom->documentElement;
    set_model_iterve($root, null);
    global $vetree;
    $vetree->show_all();
}
```

Anschließend wird die Datei geparkt und der DOM-Baum an die Funktion „idattrs()“ übergeben. Dort wird jedem Element der Datei ein ID-Attribut „idstef“ hinzugefügt, womit alle Elemente einzeln ansprechbar werden.⁷¹ Der Rest des Ablaufs ist ähnlich der Funktion „model_overview()“ aus Abschnitt 2.1.: der DOM-Baum wird an die Funktion „set_model_iterve()“⁷² übergeben, welche das Modell des unteren GtkTreeView-Objekts mit Inhalten füllt, und abschließend wird das Widget aktualisiert, so dass der DOM-Baum des Findmittels angezeigt wird.

3.4. Die Findmittel

Falls der Benutzer ein Findmittel ausgewählt hat, wird dieses Findmittel – wie in Abschnitt 2.3. beschrieben – geparkt und im unteren Navigationsbaum angezeigt. Mittels dieses Navigationsbaums kann der Benutzer durch die Struktur des Findmittels „wandern“ und sich einzelne Verzeichnungseinheiten oder Klassifikationsstufen und ähnliches anschauen und bearbeiten.

Zunächst wird dafür die ausgewählte Reihe an die Funktion „selectVETreeRow()“ übergeben. Diese Funktion liest lediglich die Element-ID aus und prüft, ob die Dateistruktur der AsterFind.dtd entspricht. Wenn dem so ist, wird die Funktion „getids()“ aufgerufen und die Element-ID übergeben.

Die Funktion „getids()“ wiederum liest den Elementnamen und die IDs aller Kindelemente des gewählten Elements aus und gibt diese an die passende Maskenfunktion.

⁷¹ Je nach übergebenem Parameter löscht diese Funktion die „idstef“-Attribute auch wieder und ist ebenso dafür zuständig, durch den Benutzer zeitweise gesperrte Bestände beim Beenden des Programms wieder zur Bearbeitung freizugeben. Siehe auch Abschnitt 2.5.

⁷² Vgl. auch die Funktion „set_model_iterbest()“

Die Funktion „hasChild()“ prüft, ob ein Element ein bestimmtes Kindelement hat und gibt entsprechend „true“ oder „false“ zurück und die Funktion „getChild()“ gibt den entsprechenden Kindknoten zurück. Sollte es mehrere Kindknoten mit diesem Namen geben, wird nur der erste zurückgegeben. Bei allen Kindelementen, die laut DTD mehrfach vorkommen können⁷³, wird die ID des Vaterknotens (also der, der von der Funktion „selectVETreeRow()“ übergeben wurde) als Element-ID übergeben, und die Auswertung erfolgt dann in den Funktionen, die die Elemente letztendlich graphisch darstellen⁷⁴.

```
function getids($id){
    global $vedom;
    $ve=$vedom->getElementById($id);
    $tag=$ve->nodeName;
    if ($ve->hasAttribute('Gba_Art') or $ve->hasAttribute('Art')) {
        $idattgba=$id;
    } else {$idattgba=";}
    if (hasChild($ve,'Abg_Stelle')==true){
        if ($tag=='Foto_Plakat' or $tag=='Karte' or $tag=='Sachakte' or $tag=='Grundbuch_akte'){
            $schild=getChild($ve,'Abg_Stelle');
            $idabgstelle=$schild->getAttribute('idstef');
        } else {$idabgstelle=$id;}
    } else {$idabgstelle=";}
    if (hasChild($ve,'Abkuerzungen')==true){
        $sabk=getChild($ve,'Abkuerzungen');
        $idabk=$sabk->getAttribute('idstef');
    } else {$idabk=";}
    if (hasChild($ve,'Akzession')==true){
        if ($tag=='Foto_Plakat' or $tag=='Karte' or $tag=='Sachakte' or $tag=='Grundbuch_akte'){
            $schild=getChild($ve,'Akzession');
            $idakzession=$schild->getAttribute('idstef');
        } else {$idakzession=$id;}
    } else {$idakzession=";}
    ...
    if ($tag == "Findmittel_Info"){
        mask_findinfo($id, $idfmname, $idfmsig, $idlz, $idbearb, $idbem, $idsperrvermerk, $idhif,
        $idverweise);
    }
    if ($tag == "Klassifikation"){
        mask_klassi($id, $idsperrvermerk, $idklassnr, $idklasstitel, $idbem, $idverweise);
    }
    if ($tag == "Konkordanz"){
        mask_konkordanz($id, $idbem);
    }
    if ($tag == "Sachakte"){
        mask_sach($id, $idsig, $idlz, $idtitel, $identhaelt, $idumfang, $idsperrvermerk, $idbestkurz,
        $idbestellsig, $idlager, $idzustand, $idaltsig, $idvorprov, $svorgid, $idaz, $idbem, $idabgstelle, $idakzession,
        $idprov, $idverweise);
    }
    ...
}
```

⁷³ z.B. Vorprovenienzen

⁷⁴ Dies sind alle Funktionen, die sich in der Datei „elementsfm.inc.php“ befinden.

Anschließend werden die benötigten Element-IDs an die passende Maskenfunktion übergeben.⁷⁵

Beispielhaft ist an dieser Stelle die Funktion dargestellt, die die Maske für das Konkordanzelement bereitstellt. Übergabeparameter sind die Element-ID und die der Kinderelemente.⁷⁶

Zunächst wird ein GtkHBox-Objekt (\$p1hbox1) erstellt, dann wird \$idbem an die Funktion

„elementbem()“⁷⁷

übergeben, welche einen GtkTreeView⁷⁸ erstellt, mit dem Inhalt eines jeden „Bem“-Elements als

```
function mask_konkordanz($id, $idbem) {
    $p1hbox1= new GtkHBox(false, 0);
    $bem=elementbem($idbem);
    $framebem = createframe('Bemerkungen', $bem, 'HV', true);
    $p1hbox1->pack_start($framebem, true, true, 5);
    $p1hbox2= new GtkHBox(false, 0);
    $konkord=elementkonkordanz($id);
    $framekonkord = createframe('Konkordanzen', $konkord, 'HV', true);
    $p1hbox2->pack_start($framekonkord, true, true, 5);
    $vbox = new GtkVBox(false, 8);
    $vbox->pack_start($p1hbox1, false, false, 5);
    $vbox->pack_start($p1hbox2, true, true, 5);

    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages(>0);
    $note->append_page($vbox, new GtkLabel('Konkordanzen'));
    $note->show_all();
}
```

einzelne Reihe. Das zurückgegebene Widget wird an die Funktion „createframe()“ übergeben, wo es noch einen Rahmen mit der Bezeichnung „Bemerkungen“ erhält. Anschließend wird das fertige Widget in das GtkHBox-Objekt eingehängt.

Dann wird eine zweite horizontale Box erstellt und die Funktion „elementkonkordanz()“ aufgerufen. In dieser Funktion werden alle „Kon_Eintrag“-Elemente ausgelesen und als Liste dargestellt. Das einzige Manko an dieser Funktion

```
function elementkonkordanz($id){
    global $vedom;
    $root=$vedom->documentElement;
    ...
    $menu = new GtkMenu();
    $echo1= new GtkImageMenuItem(GTK::STOCK_REFRESH);
    $echo1->connect_simple('activate','cleartree', $skonmodel);
    $echo1->connect_simple('activate', 'new_konkord', $root,
$skonmodel);
    $menu->append($echo1);
    $menu->show_all();
    $konlist->set_events($konlist->get_events() |
Gdk::BUTTON_PRESS_MASK);
    $konlist->connect('button-press-event', 'doPopup', $menu);
    return $konscroll;
}
```

⁷⁵ Die Masken liegen in der Datei „masksarchiv.inc.php“.

⁷⁶ In diesem Fall gibt es nur „Bem“-Elemente. Da diese mehrfach vorkommen können, sind die Inhalte von \$id und \$idbem identisch.

⁷⁷ Alle Funktionen, die ein Element grafisch darstellen, sind in der Datei „elements.inc.php“ gespeichert. Alle Funktionen, die – falls nötig – ein Attribut grafisch darstellen, liegen in der Datei „attributes.inc.php“.

Momentan befindet sich darin nur eine Funktion für das Attribut „Grundbuch_Art“.

⁷⁸ Diesmal ist das Modell vom Typ GtkListStore und stellt eine einfache Liste und keinen Baum dar.

ist, dass zur Zeit nur das erste von eventuell mehreren vorhandenen „Altsig“-Elementen ausgelesen und dargestellt wird. Nachdem die Liste fertig ist, wird zusätzlich noch ein Rechtsklickmenü eingehängt.⁷⁹

Dieses bietet lediglich die Option „Aktualisieren“ an. Wenn das Item aktiviert wird, wird zuerst das Modell an die Funktion „cleartree()“ übergeben und die Anzeige gelöscht. Anschließend wird die Funktion „new_konkord()“ aufgerufen, welche den gesamten Findmittel-Baum durchläuft und vollautomatisch eine neue Konkordanzliste erstellt. Auch hier ist momentan noch das Problem, das nur das erste Element ausgelesen wird. Die neue Konkordanzliste wird auch noch nicht in den DOM-Baum geschrieben, sondern nur das Modell, da dies momentan eher Demonstrationszwecken dient.

Anschließend werden die beiden horizontalen Boxen in eine vertikale Box gepackt, bereits angezeigte Masken werden gelöscht und die neue Maske mittels des Befehls „append“ eingehängt und angezeigt.

3.5. Die Menüleiste

Von der Menüleiste sind zurzeit nur das Item „Beenden“ im Menü „Datei“ und der „About“-Dialog im Hilfe-Menü aktiviert.

Wenn man auf „Speichern“ klickt, öffnet sich zwar ein Dialog-Fenster, aber alle hier eingegebenen Werte werden nicht ausgewertet!

Beim Beenden wird die Funktion „onQuit()“ aufgerufen.

Diese lädt zuerst erneut die „Archiv.xml“ und übergibt den DOM-Baum an die Funktion „idattrs()“, welche die gesperrten Informationen

```
function onQuit(){
    global $overviewdom;
    global $overviewfile;
    global $vedom;
    global $vefile;
    $overviewdom=load_xml($overviewfile);
    idattrs($overviewdom, 'ovidclear');
    $overviewdom->save($overviewfile);
    if (!empty($vedom)){
        idattrs($vedom, 'veidclear');
        $vedom->save($vefile);
    }
    Gtk::main_quit();
}
```

wieder freigibt und dann wird die Datei gespeichert. Als nächstes wird überprüft, ob in der globalen Variable \$vedom, welche die DOM-Bäume für die Findmittel enthält Daten gespeichert sind. Sollte dies der Fall sein, dann kommt es zu einem ähnlichen Ablauf wie bei der Tektonik. Zuerst werden die „idstef“-Attribute, welche für die Verarbeitung der

⁷⁹ Die Auswertung, ob der rechte Mausbutton gedrückt wurde, erfolgt in der Funktion „doPopup“. Solche Popup-Menüs sollen später auch in den Navigationsbäumen das Löschen, Erstellen und Verschieben von Elementen ermöglichen.

Findmittel in Aster wichtig sind, komplett entfernt, dann wird das Findmittel in die Datei gespeichert. Abschließend wird die Hauptschleife mit dem Befehl „Gtk::main_quit()“ beendet.

Da der „About“-Dialog sich weitestgehend selbst erklärt, wird hier nur kurz darauf eingegangen: Zuerst wird ein neuer GtkAboutDialog instanziiert und die Position auf die

Mitte des Bildschirms festgelegt, anschließend werden die üblichen Informationen (Name des Programms, Version, Copyright, Homepage etc.) eingehängt. Mit dem Befehl „set_license“ kann man dem GtkAboutDialog ein „End

```
function onHelp() {
    $dlg = new GtkAboutDialog();
    $dlg->set_position('GTK_WIN_POS_CENTER');
    $dlg->set_name('ASTER');
    $dlg->set_version('0.0.1');
    $dlg->set_comments('Verzeichnungstool');
    $dlg->set_copyright('No Copyright');
    $dlg->set_license("Hier kann das EULA eingesehen werden.\n"
        . "MfG S.J.");
    $pixmap = GdkPixbuf::new_from_file('img/aster.png');
    $dlg->set_logo($pixmap);
    $dlg->set_website('http://example.org');
    $dlg->run();
}
```

User License Agreement (EULA)“ hinzufügen, welches über einen Button aufgerufen wird.

4. Resümee und weitere Pläne

Da die Entwicklung einer Open-Source-Software für Archive im Rahmen dieser Arbeit vom Umfang her lediglich einer Art Machbarkeitsstudie entspricht und sowohl in Theorie als auch in der praktischen Umsetzung das ganze Konstrukt nur grob umrissen werden konnte, ist es zunächst nötig, das gesamte Konzept detaillierter auszuarbeiten.

Dazu gehört ein durchdachter Gesamtplan, wie die einzelnen Teile der Software miteinander arbeiten sollen und die DTDs müssen überarbeitet werden. Beim Verzeichnungstool müssen die Masken fertig gestellt werden, d.h. noch fehlende Masken werden erstellt, bei den existierenden muss die Anordnung der Felder verbessert werden und noch teilweise fehlende einzelne Elemente ergänzt werden. Die gesamte Problematik des Indexierens muss durchdacht und realisiert werden.

Ebenso müssen noch vorhandene Redundanzen in der Programmierung beseitigt bzw. reduziert werden, schätzungsweise sind momentan 35 – 45% des vorhandenen Quellcodes redundant und lassen sich somit auch entfernen. Als Beispiel seien hier die Funktionen angeführt, die in der Datei „elementsfm.inc.php“ liegen. Diese stellen die einzelnen Elemente grafisch dar

und lesen die vorhandenen Daten aus. Dabei sind zum Beispiel die Funktionen „elementauflage()“, „elementausfuehrung()“ und „elementsignatur()“ identisch.

Parallel dazu muss eine funktionale Benutzerverwaltung aufgebaut und integriert werden, da die momentan verwendete Lösung nur zu Testzwecken dient.

Alles in allem konnte aber gezeigt werden, dass es durchaus möglich ist, eine quelloffene Software für Archive zu schreiben, die letztendlich auch mit kommerziellen Produkten Schritt halten können sollte und aufgrund der gewählten Programmiersprache auch für Einsteiger und Hobbyprogrammierer im Archivbereich viele Möglichkeiten bietet, direkt an der Entwicklung teilzuhaben.

Aufgrund des regen Interesses, das mein Projekt an diversen Stellen hervorgerufen hat, hoffe ich bis Herbst diesen Jahres die Entwicklung soweit vorangetrieben zu haben, dass eine erste Veröffentlichung des Projekts denkbar ist.

5. Quellen

5.1. Gedruckte Quellen

Dittrich, Helmut; Mellenthin, Jürgen: PHP 4 – Workshop, Bonn 2002

Enseleit, Damir: PHP 4/5. Befehlsreferenz, 2004 Poing

Mattocks, Scott: Pro PHP-Gtk, Berkeley 2006

Kannengiesser, Matthias: PHP 5. Das Praxisbuch, 2004 Poing

Myer, Thomas: No Nonsense XML Web Development with PHP, SitePoint Pty. Ltd. 2005

Nimz, Brigitta: EDV und Archive, in: Praktische Archivkunde. Ein Leitfaden für Medien- und Informationsdienste – Fachrichtung Archiv, hrsg. Von Norbert Reimann, Münster 2004, S. 201 - 225

Reimann, Norbert: Empfehlungen für den EDV-Einsatz in Kommunalarchiven. Einführung und Textabdruck, in: Der Archivar 46 (1993), Sp. 431 - 442

Reimann, Norbert: Praktische Archivkunde. Ein Leitfaden für Fachangestellte für Medien- und Informationsdienste – Fachrichtung Archiv, Münster 2004

Richter, Sabine: Vergleich der Formalkatalogisierung in Bibliotheken mit der Verzeichnung in Archiven für ein gemeinsames Datenangebot beider Sparten, in: Der Archivar 59 (2006), Sp. 172-176

Seeboerger-Weichselbaum, Michael: PHP & XML für Web Developer. Praxis und Referenz, Frankfurt 2005

Trugenberger, Volkert: EDV-Programme zur Archivalienerschließung. Kriterienkatalog des EDV-Ausschusses der Archivreferentenkonferenz des Bundes und der Länder - Einführung und Textabdruck, in: Der Archivar 47 (1994), Sp. 283 - 288

5.2. Online-Quellen

<http://ariadne.uni-greifswald.de/> (vom 17.06.2006)
http://cweiske.de/phpgtk2_var_dump.htm (vom 17.06.2006)
http://de.wikipedia.org/wiki/Freie_Software (vom 17.06.2006)
http://de.wikipedia.org/wiki/Open_Source (vom 17.06.2006)
http://de.wikipedia.org/wiki/Propriet%C3%A4re_Software (vom 17.06.2006)
<http://gtk.php.net/> (vom 17.06.2006)
<http://marc.theaimsgroup.com/> (vom 17.06.2006)
<http://news.php.net/> (vom 17.06.2006)
<http://oss.backendmedia.com/LiveUser/> (vom 17.06.2006)
<http://php-gtk2.de/> (vom 17.06.2006)
www.archive.nrw.de/dok/Transferarbeiten/Berger/Transferarbeit%20web.pdf (vom 17.06.2006)
www.archive.nrw.de/findbuch-digital/index.htm (vom 17.06.2006)
www.archivschule.de/content/108.html (vom 17.06.2006)
www.cmiag.ch (vom 17.06.2006)
www.dynamicwebpages.de/60.php-statistiken.php (vom 17.06.2006)
www.fsf.org (vom 17.06.2006)
www.gnope.org (vom 17.06.2006)
www.gnu.org/philosophy/categories.de.html (vom 17.06.2006)
www.gtk.org (vom 17.06.2006)
www.php.net/manual/de/ (vom 17.06.2006)
www.php-center.de/artikel/php-gtk-primer.php3 (vom 17.06.2006)
www.phpgtk.de/ (vom 17.06.2006)
www.w3.org (vom 17.06.2006)
www.zdnet.de/itmanager/tech/0,39023442,2130055,00.htm (vom 17.06.2006)

Anhänge

A – Installation und Benutzung

A.1. CD-Inhalt

Im Hauptverzeichnis der CD befinden zum einen der Ordner "Aster_NoInstall" und zum anderen der Ordner "DFG".

Der Ordner "Aster_NoInstall" enthält das Archivprogramm und der Ordner "DFG" die Findmittel-DTD des Retrokonversionsprojektes der DFG zusammen mit der dazugehörigen Tag Library (AF-FindmittelTagLibrary.doc) und der Archiv-DTD von Dr. Przigoda.

Weiterhin befindet sich im Hauptverzeichnis die vorliegende Arbeit im PDF-Format und eine selbstextrahierende ZIP-Datei (AsterSetup.exe), die ebenfalls Aster installiert.

A.2. Installationsanweisungen

Im Hauptverzeichnis der CD

Diese Version von Aster ist unter den Windows-Betriebssystemen lauffähig.

Für den Einsatz auf anderen Systemen laden Sie sich bitte aus dem Internet die entsprechenden Distributionen von PHP5 und PHPGtk2+ herunter und kompilieren diese. Eine Anleitung dazu für *nix-Systeme findet sich im Internet unter <http://gtk.php.net/wiki/PhpGtk2/InstallingOnUnix> und für Mac OS X unter <http://gtk.php.net/wiki/PhpGtk2/InstallingOnMacOSX>.

Bei Aster handelt es sich um eine so genannte „portable Applikation“, das heißt, es werden keine Einträge in die Windows-Registrierung geschrieben.

Sie können entweder die AsterSetup.exe verwenden, um Aster zu installieren, oder Sie kopieren die Daten aus dem Verzeichnis „Aster_NoInstall“ in ein beliebiges Verzeichnis auf ihrer Festplatte.

Nach der Installation befinden sich in ihrem Zielordner vier Verzeichnisse und einige PHP-Scripte. Die PHP-Scripte sind die eigentlichen Programmdateien, siehe dazu auch Anhang C und Kapitel 2.3.

Im Ordner „System“ befindet sich eine CLI-Version von PHP5 und die benötigten PHPGtk2+-Bibliotheken.

Der Ordner „img“ beinhaltet einige Grafiken, die für das graphische System nötig sind. Im Ordner „data“ liegen die archiv.xml, welche die Archivinformationen und die Tektonik beinhaltet und die einzelnen Findmitteldateien. Im Ordner „transform“ befinden sich die benötigten DTD und XML-Schemata. Hier werden später auch die XSL-Scripte für die Überführung von Findmittelformaten in andere Formate aufbewahrt.

A.3. Bedienungsanleitung

Das Programm wird mit der AsterVE.exe gestartet. Zuerst erfolgt eine Abfrage der Benutzerdaten (Abb.2). Da eine Benutzerverwaltung zurzeit noch nicht eingerichtet ist, erfolgt auch keine Überprüfung der eingegebenen Daten, es wird aber trotzdem mindestens ein Benutzername erwartet. Sollten Sie das Programm in einem Netzwerk oder auf einem Server installieren und es somit möglich ist, dass mehrere Benutzer gleichzeitig mit dem Programm arbeiten, so werden die Masken, die gerade von einem anderen Benutzer bearbeitet werden, nicht angezeigt. (Abb. 3).

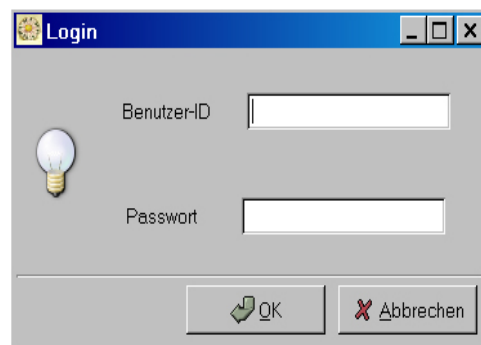


Abbildung 2: Login-Dialog

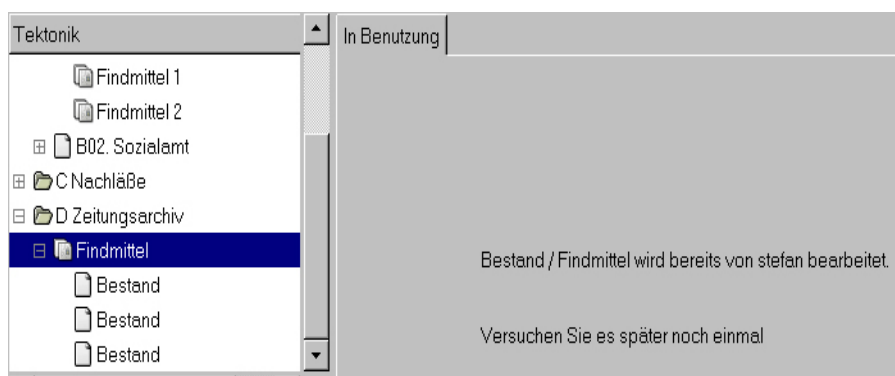


Abbildung 3: Bestand in Benutzung

Nach dem Login hat man Zugriff auf das Programm.

Im Menü sind momentan nicht alle Punkte aktiviert. Im Menüpunkt „Datei“ ist nur die Option „Beenden“ nutzbar, bei der Option „Speichern“ wird zwar ein Dialog gestartet, die

übergebenen Werte jedoch nicht ausgewertet. Der Menüpunkt „Einstellungen“ ist ebenfalls nicht aktiviert. Im „Hilfe“-Menü können Sie sich die Informationen zur Software anzeigen lassen.

Der Hauptteil der Programmoberfläche wird von den zwei Navigationsbäumen auf der linken Seite und dem Erfassungsfenster auf der rechten Seite in Anspruch genommen.

Der obere Navigationsbaum gibt die Archivtektonik wieder und wenn Sie dort ein Findmittel auswählen, dann wird das Findmittel in den unteren Baum geladen und Sie können dort mit der Verzeichnung beginnen. Parallel dazu werden im Erfassungsfenster entsprechend der Auswahl unterschiedliche Erfassungsmasken angezeigt (Abb. 4).

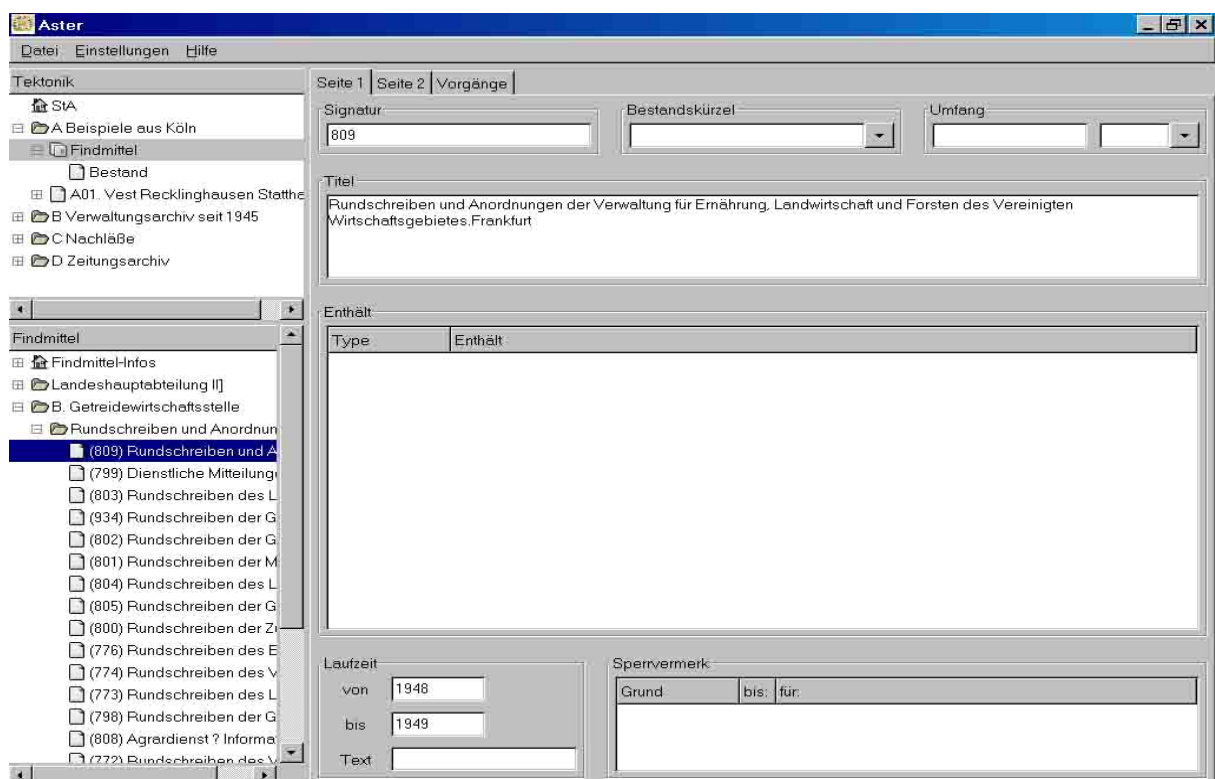


Abbildung 4: Hauptfenster

In der Archivtektonik ist die Erfassungsmaske für Bestandsinformationen und in der Findmittelansicht die Masken für Fallakten, Urkunden, Siegel und die einzelnen Indizes noch nicht erstellt!

Zur Zeit befindet sich die Software im „Nur-Lesen“-Modus, das heißt, Sie können sich die Testdaten, die zu Präsentationszwecken eingebunden wurden, anschauen und auch in den einzelnen Masken ändern, aber diese Änderungen werden nicht gespeichert!

Später sollen Elemente in den Navigationsbäumen mittels eines Popup-Menüs (rechter Mausbutton) und Drag'n'Drop hinzugefügt, gelöscht oder verschoben werden können⁸⁰.

Die Erfassungsmasken sind ebenfalls noch nicht vollständig! Manchmal tragen die Beschreibungen und Auswahlelemente den Namen des XML-Elements, das die Werte aufnimmt, sind aber für Personen, die die DTD nicht kennen, relativ nichtssagend.

Einige Elemente wie zum Beispiel „Bestandskürzel“, „Provenienz“ und „Abg. Stelle“ bieten bei den Erfassungsmasken für Verzeichnungseinheiten so genannte Drop-Down-Listen an. Hier werden - falls vorhanden – die entsprechenden Elemente aus den Bestandsinformationen ausgelesen und zur Auswahl zur Verfügung gestellt. Das heißt, falls im Element „Bestand_Info“ diverse Vorprovenienzen aufgenommen wurden, können diese mit einem Klick der Verzeichnungseinheit zugeordnet werden. Dadurch reduziert sich zum einen der Schreibaufwand erheblich und auch Schreibfehler werden verhindert.

Zu den Beispielbeständen ist zu sagen, dass die beiden Findbücher in „A – Beispiele aus Köln“ mir freundlicherweise vom Stadtarchiv Köln zur Verfügung gestellt wurden. Der Rest der Tektonik verweist immer auf die Datei „1.xml“, deren Einträge zugegebenermaßen sinnlos sind, aber dieses Findbuch dient in erster Linie zur Prüfung der Funktionalität der Software.

Die AsterArchiv.dtd ist so flexibel aufgebaut, dass es möglich ist, ein oder mehrere Findbücher (die wiederum ein oder mehrere Bestände enthalten können) direkt in die Tektonik einzuhängen, wobei die enthaltenen Bestände anhand des „Bestand_Info“-Elements ausgelesen und angezeigt werden. Ebenso können in der Tektonik auch ein oder mehrere Bestände aufgenommen werden, die wiederum ein oder mehrere Findmittel enthalten (wobei dann in den Findmitteln nur ein Bestand oder Teilbestand enthalten sein sollte).

Mehrere dieser Möglichkeiten sind beispielhaft in der Archiv.xml dargestellt (vgl. v.a. Tektonik-Nr. A und D).

Momentan sind auch alle Index-Aspekte noch nicht realisiert.

⁸⁰ Als Beispiel sei hier auf die Konkordanzen in 1.xml hingewiesen. Dort wird anfangs die in der XML-Datei gespeicherte Liste angezeigt, bei einem Rechtsklick erscheint ein Popupmenü mit der Option „Aktualisieren“. Wenn man dies wählt, wird das gesamte Dokument nach den entsprechenden Elementen durchsucht und die Ansicht aktualisiert. Die Liste ist leider noch nicht sortierbar, bei mehreren Altsignaturen wird nur die erste ausgelesen und die Änderungen werden noch nicht in die Datei geschrieben

B – Vergleich der DTDs

Um die bisher gemachten Änderungen zwischen den diversen DTDs sichtbar zu machen, werden hier die relevanten DTDs gegenübergestellt.

Es muss an dieser Stelle noch einmal darauf hingewiesen werden, dass die Aster-DTDs einen vorläufigen Charakter haben und aus Zeitgründen bei den meisten Elementen die Minimal-Lösung gewählt wurde. Dies wird sich in der weiteren Entwicklung definitiv wieder ändern, sodass die Elemente so originalgetreu wie möglich nach der AF-Findmittel.dtd und der ArchivDTD.dtd wiedergegeben werden.

B.1. ArchivDTD.dtd vs. AsterArchiv.dtd

Die ArchivDTD.dtd wurde von Dr. Stefan Przigoda vom Deutschen Bergbaumuseum Bochum entwickelt. Ziel war es dabei, eine Instanz zu schaffen, die die Beschränkung der AF-Findmittel.dtd aufheben und die Darstellung der Archivtektonik in XML ermöglichen sollte. Dazu wird die AF-Findmittel.dtd in die ArchivDTD.dtd eingehängt und es können bereits definierte Elemente sogar weiterverwendet werden. Diese Strukturierung bedeutet aber letztendlich, dass das gesamte Archiv in einer XML-Datei dargestellt wird.

Für eine Verzeichnungssoftware ist dies natürlich sehr umständlich, aus diesem Grund wurden die beiden DTDs von einander gelöst (das Element „Findmittel“ wurde in das Element „Findmitteldatei“ geändert und nimmt nun den Pfad der Datei auf!) und sämtliche Elemente in der AsterArchiv.dtd erneut definiert.

Die kursiv gedruckten Elemente in der ArchivDTD.dtd stellen all diejenigen Elemente dar, welche in der AF-Findmittel.dtd definiert wurden und nicht in der ArchivDTD.dtd.

Sie wurden nur zur besseren Übersicht und auch nur in der ersten Ebene hier eingefügt!

ArchivDTD.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.1 (http://www.xmlspy.com) by
Someone at ... (Nordrhein-Westfälisches Hauptstaatsarchiv) -->
<!--
*****
*****-->
<!--Autor: Dr. Stefan Przigoda, c/o Deutsches Bergbau-Museum
Bochum, Montanhistorisches Dokumentationszentrum, Am
Bergbaumuseum 28, 44791 Bochum, Tel. (0234) 58 77 -118, Mail
stefan.przigoda@bergbaumuseum.de-->
<!--
*****
```

AsterArchiv.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.1 (http://www.xmlspy.com) by
Someone at ... (Nordrhein-Westfälisches Hauptstaatsarchiv) -->
<!--
*****
*****-->
<!-- DTD für die Datenspeicherung im Erschließungsprogramm
Aster von S. Jentsch, stefanjentsch@gmx.de
Dies ist eine modifizierte Version der ArchivDTD.dtd von Dr. Stefan
Przigoda, c/o Deutsches Bergbau-Museum Bochum,
Montanhistorisches Dokumentationszentrum, Am Bergbaumuseum
28, 44791 Bochum, Tel. (0234) 58 77 -118, Mail
```

```

*****-->
<!--Stand: 21.10.2003-->
<!--
*****
*****-->
<!--Standardmäßig ist das im Rahmen des DFG-Projektes
"Retrokonversion archivischer Findmittel" entwickelte
Austauschformat für die differenzierte Abbildung von
Erschließungsinformationen in die Archiv-DTD eingebunden.
Die entsprechende DTD muss also ebenfalls vorhanden und am
besten im selben Verzeichnis abgelegt sein!
Sollen jedoch Metadaten zu Images von handschriftlichen
Findmitteln abgebildet werden, so wäre eine entsprechende DTD
einzubinden indem der Wert für die Entity IMAGES aus
"INCLUDE" und entsprechend der Wert für die Entity
Austauschformat auf "IGNORE" eingestellt wird.-->
<!--
*****
*****-->
<!ENTITY % Austauschformat "INCLUDE">

<!ENTITY % Images "IGNORE">

<![ %Austauschformat; [
<!ENTITY % AF-Findmittel_Betaversion.dtd SYSTEM "AF-
Findmittel_Betaversion.dtd">
%AF-Findmittel_Betaversion.dtd;
]]>

<![ %Images; [
<!ENTITY % AF_HS-Meta.dtd SYSTEM "AF_HS-Meta.dtd">
%AF_HS-Meta.dtd;
]]>

<!ENTITY % a.bezug 'Bezug (Gesamtbestand | Teilbestand |
Findmittel | Teil-Findmittel | Klass_Stufe | Bandserie | VE |
Vorgang) #IMPLIED'>

<!ENTITY % e.Verweis 'archref | bibref | FM_ref | altübform |
Register'>

<!ENTITY % e.link 'Vwint | Vwext'>

<!ENTITY % a.level 'level (01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
10) #IMPLIED
levelflex CDATA #IMPLIED'>

<!ENTITY % e.VE_Basis 'Signatur, Laufzeit?, Titel'>

<!ENTITY % e.VE_Option 'Bestellsig | Altsig | Provenienz |
Vor_Prov | Abg_Stelle | Akzession | Sperrvermerk | Umfang |
Lagerung | Zustand | Bestand_Kurz | %e.Verweis; | Bem'>

<!ENTITY % e.Klass_Info 'Klass_Nr?, Klass_Titel?,
(%e.Verweis;)*, Sperrvermerk?, Bem*'>

<!ENTITY % e.Serien_Info 'Serientitel, Enthael*, Laufzeit?,
(%e.Verweis;)*, Sperrvermerk?, Bem*'>

<!ELEMENT Archiv (Datei_Info, Archivinfo?, (Tektonik? |
Bestand*)+)>

<!ELEMENT Archivinfo (#PCDATA | Archivname | Anschrift |
Telefon | Fax | Email | Homepage | Land | Kompetenz | Gruendung |
Aufloesung | Vorgaenger | Nachfolger | Bibliographie | %e.Verweis;
| p)*>

<!ELEMENT Archivname (#PCDATA | Lang | Kurz)*>

<!ELEMENT Anschrift (#PCDATA | PLZ | Ort | Strasse | Nr)*>

```

```

stefan.przigoda@bergbaumuseum.de-->
<!--
*****
*****-->
<!--Stand: 01.06.2006-->
<!--
*****
*****-->

<!ENTITY % a.bezug 'Bezug (Gesamtbestand | Teilbestand |
Findmittel | Teil-Findmittel | Klass_Stufe | Bandserie | VE |
Vorgang) #IMPLIED'>

<!ENTITY % e.Verweis 'archref | bibref | FM_ref | altübform |
Register'>

<!ENTITY % e.link 'Vwint | Vwext'>

<!ENTITY % a.level 'level (01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
10) #IMPLIED
levelflex CDATA #IMPLIED'>

<!ELEMENT Archiv (Archivinfo?, (Tektonik? | Bestand*)+)>

<!ELEMENT Archivinfo (Archivname | Anschrift | Telefon | Fax |
Email | Homepage | Land | Kompetenz | Gruendung | Aufloesung |
Vorgaenger | Nachfolger | Bibliographie | %e.Verweis;)*>
<ATTLIST Archivinfo
state (open | locked) #REQUIRED
user CDATA #REQUIRED
id ID#REQUIRED
>

<!ELEMENT Archivname (#PCDATA)*>

<!ELEMENT Anschrift (#PCDATA)>

```

```

<!ELEMENT Telefon (#PCDATA)>

<!ELEMENT Fax (#PCDATA)>

<!ELEMENT Email (#PCDATA)>
<!ATTLIST Email
    Home_Mail CDATA #IMPLIED
>

<!ELEMENT Homepage (#PCDATA)>
<!ATTLIST Homepage
    Home_URL CDATA #IMPLIED
>

<!ELEMENT Land (#PCDATA)>

<!ELEMENT Bibliographie (#PCDATA | p | bibref)*>
<!ATTLIST Bibliographie
    Publ_Art (Literatur | Amtsdrucksachen | Editionen)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT archref (#PCDATA | %e.link;)*>
<!ATTLIST archref
    archref_Fkt (siehe_auch | Abschrift | Original |
Entnommen_aus | Separiertes_Material) #IMPLIED
    Fkt CDATA #IMPLIED
    archref_Art (Findmittel | Bestand | Akte | Drucksache |
Urkunde | Siegel | Karte | Plakat | Foto | Film) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT bibref (#PCDATA | %e.link;)*>
<!ATTLIST bibref
    bibref_Art (Literatur | Amtsdrucksache | Edition)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT FM_ref (#PCDATA | %e.link;)*>

<!ELEMENT altübform (#PCDATA | %e.link;)*>
<!ATTLIST altübform
    altübform_Art (Lesefilm | Sicherungsfilm | Microfiche |
Datei) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Register (#PCDATA | %e.link;)*>
<!ATTLIST Register
    Register_Art (Person | Sache | Ort) #IMPLIED
    Art CDATA #IMPLIED
    Thesaurusreferenz CDATA #IMPLIED
>

<!ELEMENT Vwint (#PCDATA)>
<!ATTLIST Vwint
    Vwint_Art (VE | Index | FM_Seite) #IMPLIED
    idref IDREFS #IMPLIED
>

<!ELEMENT Vwext (#PCDATA)>
<!ATTLIST Vwext
    Vwext_Art (analog | digital) #IMPLIED
    dateiref CDATA #IMPLIED
    href CDATA #IMPLIED
>

<!ELEMENT Tektonik (Tekt_Nr | Tekt_Titel | Tekt_Beschreibung |
(Bestand | Findmittel)* | Indizes | Konkordanz | %e.Verweis; |
Hilfsfeld | Bem | Tektonik)+>
<!ATTLIST Tektonik

```

```

<!ELEMENT Telefon (#PCDATA)>

<!ELEMENT Fax (#PCDATA)>

<!ELEMENT Email (#PCDATA)>
<!ATTLIST Email
    Home_Mail CDATA #IMPLIED
>

<!ELEMENT Homepage (#PCDATA)>
<!ATTLIST Homepage
    Home_URL CDATA #IMPLIED
>

<!ELEMENT Land (#PCDATA)>

<!ELEMENT Bibliographie (bibref)*>
<!ATTLIST Bibliographie
    Publ_Art (Literatur | Amtsdrucksachen | Editionen)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT archref (#PCDATA)>
<!ATTLIST archref
    archref_Fkt (siehe_auch | Abschrift | Original |
Entnommen_aus | Separiertes_Material) #IMPLIED
    Fkt CDATA #IMPLIED
    archref_Art (Findmittel | Bestand | Akte | Drucksache |
Urkunde | Siegel | Karte | Plakat | Foto | Film) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT bibref (#PCDATA)>
<!ATTLIST bibref
    bibref_Art (Literatur | Amtsdrucksache | Edition)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT FM_ref (#PCDATA)>

<!ELEMENT altübform (#PCDATA)>
<!ATTLIST altübform
    altübform_Art (Lesefilm | Sicherungsfilm | Microfiche |
Datei) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Register (%e.link;)*>
<!ATTLIST Register
    Register_Art (Person | Sache | Ort) #IMPLIED
    Art CDATA #IMPLIED
    Thesaurusreferenz CDATA #IMPLIED
>

<!ELEMENT Vwint (#PCDATA)>
<!ATTLIST Vwint
    Vwint_Art (VE | Index) #IMPLIED
    idref IDREFS #IMPLIED
>

<!ELEMENT Vwext (#PCDATA)>
<!ATTLIST Vwext
    Vwext_Art (analog | digital) #IMPLIED
    dateiref CDATA #IMPLIED
    href CDATA #IMPLIED
>

<!ELEMENT Tektonik (Tekt_Nr | Tekt_Titel | Tekt_Beschreibung |
(Bestand | Findmitteldatei)* | %e.Verweis; | Bem | Tektonik)+>
<!ATTLIST Tektonik

```

```

    %a.level;
>

<!ELEMENT Tekt_Nr (#PCDATA)>

<!ELEMENT Tekt_Titel (#PCDATA)>

<!ELEMENT Tekt_Beschreibung (#PCDATA | p)*>

<!ELEMENT Bestand (Bestandsname, Bestand_Kurz?,
Bestand_Sig?, Laufzeit?, (Bestandsgeschichte | Bestand_Inhalt |
Provenienz | Prov_Info | Vor_Prov | Abg_Stelle | Bewertung |
Erschliessung | Rechtsstatus | Akzession | Sperrvermerk | Umfang |
Bearbeiter | Zustand | Lagerung | Nr | %e.Verweis; | Bem | Hilfsfeld
| Findmittel | Bestand)*>
<!ATTLIST Bestand
    Best_Art (Hauptbestand | Teilbestand | Nebenbestand)
#IMPLIED
    Art CDATA #IMPLIED
    %a.bezug;
>

<!ELEMENT Rechtsstatus (#PCDATA)>
<!ATTLIST Rechtsstatus
    Depositum (ja | nein) #IMPLIED
>

<!ELEMENT Findmittel (Datei_Info, Findmittel_Info,
Klassifikation*, Indizes?, Konkordanz?, Bem*, Hilfsfeld*)>
<!ATTLIST Findmittel
    %a.bezug;
    Art CDATA #IMPLIED
>

<!ELEMENT Bestandsgeschichte (#PCDATA | p | %e.Verweis;)*>

<!ELEMENT Bestand_Inhalt (#PCDATA | p)*>

<!ELEMENT Bewertung (#PCDATA | p)*>

<!ELEMENT Erschliessung (#PCDATA | p)*>

<!ELEMENT Prov_Info (#PCDATA | Person | Institution |
Kompetenz | Gruendung | Aufloesung | Vorgaenger | Nachfolger |
Bibliographie | %e.Verweis; | p | Prov_Info)*>

<!ELEMENT Kompetenz (#PCDATA)>

<!ELEMENT Gruendung (#PCDATA | Datum)*>

<!ELEMENT Aufloesung (#PCDATA | Datum)*>

<!ELEMENT Vorgaenger (#PCDATA | Person | Institution |
Kompetenz | Gruendung | Aufloesung | Vorgaenger | Nachfolger |
Bibliographie | %e.Verweis; | p)*>

<!ELEMENT Nachfolger (#PCDATA | Person | Institution |
Kompetenz | Gruendung | Aufloesung | Vorgaenger | Nachfolger |
Bibliographie | %e.Verweis; | p)*>

<!ELEMENT Hilfsfeld (#PCDATA | p)*>

```

```

    %a.level;
state (open | locked) #REQUIRED
user CDATA #REQUIRED
id ID#REQUIRED
>

<!ELEMENT Tekt_Nr (#PCDATA)>

<!ELEMENT Tekt_Titel (#PCDATA)>

<!ELEMENT Tekt_Beschreibung (#PCDATA)>

<!ELEMENT Bestand (Bestandsname, Bestand_Kurz?,
Bestand_Sig?, Laufzeit?, (Bestandsgeschichte | Bestand_Inhalt |
Provenienz | Prov_Info | Vor_Prov | Abg_Stelle | Bewertung |
Erschliessung | Rechtsstatus | Akzession | Sperrvermerk | Umfang |
Bearbeiter | Zustand | Lagerung | %e.Verweis; | Bem | Hilfsfeld |
Findmitteldatei | Bestand)*>
<!ATTLIST Bestand
    Best_Art (Hauptbestand | Teilbestand | Nebenbestand)
#IMPLIED
    Art CDATA #IMPLIED
    %a.bezug;
state (open | locked) #REQUIRED
user CDATA #REQUIRED
id ID#REQUIRED
>

<!ELEMENT Rechtsstatus (#PCDATA)>
<!ATTLIST Rechtsstatus
    Depositum (ja | nein) #IMPLIED
>

<ELEMENT Findmitteldatei (#PCDATA)>
<ATTLIST Findmitteldatei
    state (open | locked) #REQUIRED
    user CDATA #REQUIRED
    id ID#REQUIRED
>

<ELEMENT Bestandsgeschichte (#PCDATA | %e.Verweis;)*>

<!ELEMENT Bestand_Inhalt (#PCDATA)>

<!ELEMENT Bewertung (#PCDATA)>

<!ELEMENT Erschliessung (#PCDATA)>

<!ELEMENT Prov_Info (Person | Institution | Kompetenz |
Gruendung | Aufloesung | Vorgaenger | Nachfolger | Bibliographie |
%e.Verweis; | Prov_Info)*>

<!ELEMENT Kompetenz (#PCDATA)>

<!ELEMENT Gruendung (#PCDATA)>

<!ELEMENT Aufloesung (#PCDATA)>

<!ELEMENT Vorgaenger (Person | Institution | Kompetenz |
Gruendung | Aufloesung | Vorgaenger | Nachfolger | Bibliographie |
%e.Verweis;)*>

<!ELEMENT Nachfolger (Person | Institution | Kompetenz |
Gruendung | Aufloesung | Vorgaenger | Nachfolger | Bibliographie |
%e.Verweis;)*>

<ELEMENT Hilfsfeld (#PCDATA)>

```

```

<!ATTLIST Hilfsfeld
    %a.bezug;
    Fkt CDATA #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Bem (#PCDATA | p)*>

<!ELEMENT Laufzeit (LZ_Text* | (LZ_Anfang, LZ_Ende)* |
    Datierung* | Bem* | Hilfsfeld*)+>
<!ATTLIST Laufzeit
    %a.bezug;
    LZ_Art (Gegeben | Erschlossen | Geschaetzt) #IMPLIED
    Art CDATA #IMPLIED
    Dati_Fkt (Produktion | Aufnahme | Herstellung |
    Premiere | Zensur | Druck | Sachverhalt) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT LZ_Text (#PCDATA)>
<!ELEMENT LZ_Anfang (#PCDATA)>
<!ELEMENT LZ_Ende (#PCDATA)>
<!ELEMENT Bestandsname (#PCDATA)>
<!ELEMENT Bestand_Sig (#PCDATA | Sig_nr | Sig_alphanum |
    Hilfsfeld)*>
<!ELEMENT Bestand_Kurz (#PCDATA)>
<!ELEMENT Provenienz (#PCDATA | Person | Institution | Az |
    %e.Verweis; | Bem | Hilfsfeld)*>
<!ELEMENT Vor_Prov (#PCDATA | Person | Institution | Az |
    %e.Verweis; | Bem | Hilfsfeld)*>
<!ELEMENT Abg_Stelle (#PCDATA | Person | Institution | Az |
    %e.Verweis; | Bem | Hilfsfeld)*>
<!ELEMENT Akzession (#PCDATA | Nr | Datum)*>
<!ELEMENT Sperrvermerk (#PCDATA | Datum)*>
<!ATTLIST Sperrvermerk
    Sperr_Art (Konservatorisch | Bundesrecht | Landesrecht
    | Sonstige) #IMPLIED
    Art CDATA #IMPLIED
>
<!ELEMENT Umfang (#PCDATA)>
<!ATTLIST Umfang
    Mass (Blatt | Seiten | Folio | cm | VE | lfd.m. | Anzahl |
    Kartons) #IMPLIED
    Art CDATA #IMPLIED
>
<!ELEMENT Lagerung (#PCDATA)>
<!ATTLIST Lagerung
    Art CDATA #IMPLIED
>
<!ELEMENT Zustand (#PCDATA)>
<!ELEMENT Bearbeiter (#PCDATA)>
<!ELEMENT Person (#PCDATA | Pers_Name | Rang_Titel |
    Beruf_Funktion | Institution | Datum | Ort | Nationalitaet |
    Geschlecht | Konfession | Familienstand | Anschrift | %e.Verweis; |
    Bem | Hilfsfeld)*>
<!ATTLIST Person
    Pers_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
    Patient | Techniker | Herausgeber | Klaeger | Beklagter |
    Prokurator_Kl | Prokurator_Bekl | Urheber | Auftraggeber |

```

```

<!ELEMENT Bem (#PCDATA)>
<!ELEMENT Laufzeit (LZ_Text | (LZ_Anfang, LZ_Ende))+>
<!ATTLIST Laufzeit
    %a.bezug;
    LZ_Art (Gegeben | Erschlossen | Geschaetzt) #IMPLIED
    Art CDATA #IMPLIED
>
<!ELEMENT LZ_Text (#PCDATA)>
<!ELEMENT LZ_Anfang (#PCDATA)>
<!ELEMENT LZ_Ende (#PCDATA)>
<!ELEMENT Bestandsname (#PCDATA)>
<!ELEMENT Bestand_Sig (#PCDATA)>
<!ELEMENT Bestand_Kurz (#PCDATA)>
<!ELEMENT Provenienz (#PCDATA)>
<!ELEMENT Vor_Prov (#PCDATA)>
<!ELEMENT Abg_Stelle (#PCDATA)>
<!ELEMENT Akzession (#PCDATA)>
<!ELEMENT Sperrvermerk (Datum?, User?)>
<!ATTLIST Sperrvermerk
    Sperr_Art (Konservatorisch | Bundesrecht | Landesrecht |
    Sonstige) #IMPLIED
    Art CDATA #IMPLIED
>
<!ELEMENT Umfang (#PCDATA)>
<!ATTLIST Umfang
    Mass (Blatt | Seiten | Folio | cm | VE | lfd.m. | Anzahl |
    Kartons) #IMPLIED
>
<!ELEMENT Lagerung (#PCDATA)>
<!ATTLIST Lagerung
    Art CDATA #IMPLIED
>
<!ELEMENT Zustand (#PCDATA)>
<!ELEMENT Bearbeiter (#PCDATA)>
<!ELEMENT Person (#PCDATA)>
<!ATTLIST Person
    Pers_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
    Patient | Techniker | Herausgeber | Klaeger | Beklagter |
    Prokurator_Kl | Prokurator_Bekl | Urheber | Auftraggeber |

```



```

Rechteinhaber | Produzent | Regie | Kamera | Drehbuch | Vorlage |
Schnitt | Darsteller | Sprecher | Musik | Trick) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Institution (#PCDATA | Inst_Name | Org_Einheit |
Anschrift | %e.Verweis; | Bem | Hilfsfeld)*>
<!ATTLIST Institution
    Inst_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
Klaeger | Beklagter | Auftragnehmer | Herausgeber | Druckerei |
Rechteinhaber | Auftraggeber | Rechteinhaber | Produzent | Musik |
Trick) #IMPLIED
    Fkt CDATA #IMPLIED
    Inst_Art (Behoerde | Betrieb | Unternehmen) #IMPLIED
    Art CDATA #IMPLIED
>

```

```

Rechteinhaber | Produzent | Regie | Kamera | Drehbuch | Vorlage |
Schnitt | Darsteller | Sprecher | Musik | Trick) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Institution (#PCDATA)>
<!ATTLIST Institution
    Inst_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
Klaeger | Beklagter | Auftragnehmer | Herausgeber | Druckerei |
Rechteinhaber | Auftraggeber | Rechteinhaber | Produzent | Musik |
Trick) #IMPLIED
    Fkt CDATA #IMPLIED
    Inst_Art (Behoerde | Betrieb | Unternehmen) #IMPLIED
    Art CDATA #IMPLIED
>

```

B.2. AF-Findmittel.dtd vs. AsterFind.dtd

Hier erfolgt nun die Gegenüberstellung der Findmittel-DTDs. Sämtliche Kommentare wurden aus Platzgründen weggelassen! Bei Unverständlichkeiten betreffend einige Abkürzungen sei an dieser Stelle ein Blick auf die entsprechende DTD im Transform-Verzeichnis von Aster empfohlen, dort sind weiterhin alle Kommentare enthalten.

AF-Findmittel.dtd

```

<?xml version="1.0" encoding="UTF-8"?>

<!ENTITY % a.bezug 'Bezug (Gesamtbestand | Teilbestand |
Findmittel | Teil-Findmittel | Klass_Stufe | Bandserie | VE |
Vorgang) #IMPLIED'>

<!ENTITY % a.level 'level (01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
10) #IMPLIED
levelflex CDATA #IMPLIED'>

<!ENTITY % e.Verweis 'archref | bibref | FM_ref | altübform |
Register'>

<!ENTITY % e.link 'Vwint | Vwext'>

<!ENTITY % e.VE_Basis 'Signatur, Laufzeit?, Titel'>

<!ENTITY % e.VE_Option 'Bestellsig | Altsig | Provenienz |
Vor_Prov | Abg_Stelle | Akzession | Sperrvermerk | Umfang |
Lagerung | Zustand | FM_Seite | Bestand_Kurz | %e.Verweis; | Bem
| Hilfsfeld'>

<!ENTITY % e.Klass_Info 'Klass_Nr?, Klass_Titel?,
(%e.Verweis;)*, Sperrvermerk?, Bem*, Hilfsfeld*'>

<!ENTITY % e.Serien_Info 'Serientitel, Enthaeft*, Laufzeit?,
(%e.Verweis;)*, Sperrvermerk?, Bem*, Hilfsfeld*'>

<!ELEMENT Findmittel (Datei_Info, Findmittel_Info,
Klassifikation*, Indizes?, Konkordanz?, Bem*, Hilfsfeld*)>
<!ATTLIST Findmittel
    %a.bezug;
    Art CDATA #IMPLIED
>

<!ELEMENT Band (#PCDATA | Nr)*>

```

AsterFind.dtd

```

<?xml version="1.0" encoding="UTF-8"?>

<!ENTITY % a.bezug 'Bezug (Gesamtbestand | Teilbestand |
Findmittel | Teil-Findmittel | Klass_Stufe | Bandserie | VE |
Vorgang) #IMPLIED'>

<!ENTITY % a.level 'level (01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
10) #IMPLIED
levelflex CDATA #IMPLIED'>

<!ENTITY % e.Verweis 'archref | bibref | FM_ref | altübform |
Register'>

<!ENTITY % e.link 'Vwint | Vwext'>

<!ENTITY % e.VE_Basis 'Signatur, Laufzeit?, Titel'>

<!ENTITY % e.VE_Option 'Bestellsig | Altsig | Provenienz |
Vor_Prov | Abg_Stelle | Akzession | Sperrvermerk | Umfang |
Lagerung | Zustand | Bestand_Kurz | %e.Verweis; | Bem'>

<!ENTITY % e.Klass_Info 'Klass_Nr?, Klass_Titel?,
(%e.Verweis;)*, Sperrvermerk?, Bem*'>

<!ENTITY % e.Serien_Info 'Serientitel, Enthaeft*, Laufzeit?,
(%e.Verweis;)*, Sperrvermerk?, Bem*'>

<!ELEMENT Findmittel (Findmittel_Info, Klassifikation*,
Indizes?, Konkordanz?, Bem*)>
<!ATTLIST Findmittel
    %a.bezug;
    Art CDATA #IMPLIED
>

<!ELEMENT Band (#PCDATA)>

```

```

<!ELEMENT Nr (#PCDATA | Nr_Anfang | Nr_Ende | Zusatz)*>
<!ATTLIST Nr
    Nr_Fkt (Signatur | Akzession | Lfdnr | Siegelnr |
    Personalnr | Handelsregnr | Hausnr | Blatt | Band | Repositur |
    Artikel | Vol | Folio) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Nr_Anfang (#PCDATA | Zusatz)*>

<!ELEMENT Nr_Ende (#PCDATA | Zusatz)*>

<!ELEMENT Zusatz (#PCDATA)>

<!ELEMENT Datum (#PCDATA | Jahr | Monat | Tag)*>
<!ATTLIST Datum
    Dat_Fkt (Geburt | Tod | Akzession | FM_Erstellung |
    Sperrfrist | Herstellung | Druck) #IMPLIED
    %a.bezug;
    Fkt CDATA #IMPLIED
>

<!ELEMENT Jahr (#PCDATA)>

<!ELEMENT Monat (#PCDATA)>

<!ELEMENT Tag (#PCDATA)>

<!ELEMENT Hilfsfeld (#PCDATA | p)*>
<!ATTLIST Hilfsfeld
    %a.bezug;
    Fkt CDATA #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Bem (#PCDATA | p)*>

<!ELEMENT Person (#PCDATA | Pers_Name | Rang_Titel |
    Beruf_Funktion | Institution | Datum | Ort | Nationalitaet |
    Geschlecht | Konfession | Familienstand | Anschrift | %e.Verweis; |
    Bem | Hilfsfeld)*>
<!ATTLIST Person
    Pers_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
    Patient | Techniker | Herausgeber | Klaeger | Beklagter |
    Prokurator_Kl | Prokurator_Bekl | Urheber | Auftraggeber |
    Rechteinhaber | Produzent | Regie | Kamera | Drehbuch | Vorlage |
    Schnitt | Darsteller | Sprecher | Musik | Trick) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Pers_Name (#PCDATA | Vorname | Nachname)*>
<!ATTLIST Pers_Name
    PN_Art (Geburtsname | Kuenstlername | Alias)
    #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Vorname (#PCDATA)>

<!ELEMENT Nachname (#PCDATA)>

<!ELEMENT Beruf_Funktion (#PCDATA)>

<!ELEMENT Rang_Titel (#PCDATA)>

<!ELEMENT Nationalitaet (#PCDATA)>

<!ELEMENT Geschlecht (#PCDATA)>

<!ELEMENT Konfession (#PCDATA)>

<!ELEMENT Familienstand (#PCDATA)>

```

```

<!ELEMENT Nr (#PCDATA)>
<!ATTLIST Nr
    Nr_Fkt (Signatur | Akzession | Lfdnr | Siegelnr |
    Personalnr | Handelsregnr | Hausnr | Blatt | Band | Repositur |
    Artikel | Vol | Folio) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Datum (#PCDATA)>
<!ATTLIST Datum
    Dat_Fkt (Geburt | Tod | Akzession | FM_Erstellung |
    Sperrfrist | Herstellung | Druck) #IMPLIED
    %a.bezug;
    Fkt CDATA #IMPLIED
>

<!ELEMENT User (#PCDATA)>

<!ELEMENT Hilfsfeld (#PCDATA)>
<!ATTLIST Hilfsfeld
    %a.bezug;
>

<!ELEMENT Bem (#PCDATA)>

<!ELEMENT Person (#PCDATA)>

<!ATTLIST Person
    Pers_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
    Patient | Techniker | Herausgeber | Klaeger | Beklagter |
    Prokurator_Kl | Prokurator_Bekl | Urheber | Auftraggeber |
    Rechteinhaber | Produzent | Regie | Kamera | Drehbuch | Vorlage |
    Schnitt | Darsteller | Sprecher | Musik | Trick) #IMPLIED
    Fkt CDATA #IMPLIED
>

```

```

<!ELEMENT Ort (#PCDATA)>
<!ATTLIST Ort
    Ort_Fkt (Geburtsort | Sterbeort | Wohnort | Dienstort |
Lagerung | Region | Land | Druckort | Gemeinde | Gemarkung |
Ausstellungsort) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Institution (#PCDATA | Inst_Name | Org_Einheit |
Anschrift | %e.Verweis; | Bem | Hilfsfeld)*>
<!ATTLIST Institution
    Inst_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
Klaeger | Beklagter | Auftragnehmer | Herausgeber | Druckerei |
Rechteinhaber | Auftraggeber | Rechteinhaber | Produzent | Musik |
Trick) #IMPLIED
    Fkt CDATA #IMPLIED
    Inst_Art (Behoerde | Betrieb | Unternehmen) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Inst_Name (#PCDATA | Lang | Kurz)*>

<!ELEMENT Org_Einheit (#PCDATA)>
<!ATTLIST Org_Einheit
    OE_Art (Abt | Gruppe | Referat) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Anschrift (#PCDATA | PLZ | Ort | Strasse | Nr)*>

<!ELEMENT PLZ (#PCDATA)>

<!ELEMENT Strasse (#PCDATA)>

<!ELEMENT p (#PCDATA)>

<!ELEMENT Signatur (#PCDATA | Sig_nr | Sig_alphanum | Band |
Hilfsfeld)*>

<!ELEMENT Sig_nr (#PCDATA)>

<!ELEMENT Sig_alphanum (#PCDATA)>

<!ELEMENT Laufzeit (LZ_Text* | (LZ_Anfang, LZ_Ende)* |
Datierung* | Bem* | Hilfsfeld*)+>
<!ATTLIST Laufzeit
    %a.bezug;
    LZ_Art (Gegeben | Erschlossen | Geschaetzt) #IMPLIED
    Art CDATA #IMPLIED
    Dati_Fkt (Produktion | Aufnahme | Herstellung |
Premiere | Zensur | Druck | Sachverhalt) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT LZ_Text (#PCDATA)>

<!ELEMENT LZ_Anfang (#PCDATA)>

<!ELEMENT LZ_Ende (#PCDATA)>

<!ELEMENT Datierung (#PCDATA | Jahr | Monat | Tag)*>
<!ATTLIST Datierung
    Urk_Dat (Actum | Datum_et_Actum) #IMPLIED
>

<!ELEMENT Titel (#PCDATA | Band | Bem)*>
<!ATTLIST Titel
    Titel_Art (Haupttitel | Untertitel | Originaltitel | Kurztitel
| Arbeitstitel | Archivtitel) #IMPLIED
    Art CDATA #IMPLIED
    Titel_ID CDATA #IMPLIED
>

```

```

<!ELEMENT Ort (#PCDATA)>
<!ATTLIST Ort
    Ort_Fkt (Geburtsort | Sterbeort | Wohnort | Dienstort |
Lagerung | Region | Land | Druckort | Gemeinde | Gemarkung |
Ausstellungsort) #IMPLIED
    Fkt CDATA #IMPLIED
>

<!ELEMENT Institution (#PCDATA)>
<!ATTLIST Institution
    Inst_Fkt (Aussteller | Empfaenger | Siegler | Zeuge |
Klaeger | Beklagter | Auftragnehmer | Herausgeber | Druckerei |
Rechteinhaber | Auftraggeber | Rechteinhaber | Produzent | Musik |
Trick) #IMPLIED
    Fkt CDATA #IMPLIED
    Inst_Art (Behoerde | Betrieb | Unternehmen) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Anschrift (#PCDATA)>

<!ELEMENT Signatur (#PCDATA)>

<!ELEMENT Laufzeit (LZ_Text | (LZ_Anfang, LZ_Ende))+>
<!ATTLIST Laufzeit
    %a.bezug;
    LZ_Art (Gegeben | Erschlossen | Geschaetzt) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT LZ_Text (#PCDATA)>

<!ELEMENT LZ_Anfang (#PCDATA)>

<!ELEMENT LZ_Ende (#PCDATA)>

<!ELEMENT Titel (#PCDATA)>
<!ATTLIST Titel
    Titel_Art (Haupttitel | Untertitel | Originaltitel | Kurztitel
| Arbeitstitel | Archivtitel) #IMPLIED
    Art CDATA #IMPLIED
    Titel_ID CDATA #IMPLIED
>

```

```

<!ELEMENT Bestellsig (#PCDATA)>

<!ELEMENT Altsig (#PCDATA)>

<!ELEMENT Provenienz (#PCDATA | Person | Institution | Az |
%e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT Vor_Prov (#PCDATA | Person | Institution | Az |
%e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT Abg_Stelle (#PCDATA | Person | Institution | Az |
%e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT Akzession (#PCDATA | Nr | Datum)*>

<!ELEMENT Sperrvermerk (#PCDATA | Datum)*>
<!ATTLIST Sperrvermerk
    Sperr_Art (Konservatorisch | Bundesrecht | Landesrecht |
Sonstige) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Umfang (#PCDATA)>
<!ATTLIST Umfang
    Mass (Blatt | Seiten | Folio | cm | VE | lfd.m. | Anzahl |
Kartons) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Lagerung (#PCDATA)>
<!ATTLIST Lagerung
    Art CDATA #IMPLIED
>

<!ELEMENT Zustand (#PCDATA)>

<!ELEMENT FM_Seite (#PCDATA | Nr)*>

<!ELEMENT Bestand_Kurz (#PCDATA)>

<!ELEMENT archref (#PCDATA | %e.link;)*>
<!ATTLIST archref
    archref_Fkt (siehe_auch | Abschrift | Original |
Entnommen_aus | Separiertes_Material) #IMPLIED
    Fkt CDATA #IMPLIED
    archref_Art (Findmittel | Bestand | Akte | Drucksache |
Urkunde | Siegel | Karte | Plakat | Foto | Film) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT bibref (#PCDATA | %e.link;)*>
<!ATTLIST bibref
    bibref_Art (Literatur | Amtsdrucksache | Edition)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT FM_ref (#PCDATA | %e.link;)*>

<!ELEMENT altübform (#PCDATA | %e.link;)*>
<!ATTLIST altübform
    altübform_Art (Lesefilm | Sicherungsfilm | Microfiche |
Datei) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Register (#PCDATA | %e.link;)*>
<!ATTLIST Register
    Register_Art (Person | Sache | Ort) #IMPLIED
    Art CDATA #IMPLIED
    Thesaurusreferenz CDATA #IMPLIED
>

```

```

<!ELEMENT Bestellsig (#PCDATA)>

<!ELEMENT Altsig (#PCDATA)>

<!ELEMENT Provenienz (#PCDATA)>

<!ELEMENT Vor_Prov (#PCDATA)>

<!ELEMENT Abg_Stelle (#PCDATA)>

<!ELEMENT Akzession (#PCDATA)>

<!ELEMENT Sperrvermerk (Datum?, User?)>
<!ATTLIST Sperrvermerk
    Sperr_Art (Konservatorisch | Bundesrecht | Landesrecht |
Sonstige) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Umfang (#PCDATA)>
<!ATTLIST Umfang
    Mass (Blatt | Seiten | Folio | cm | VE | lfd.m. | Anzahl |
Kartons) #IMPLIED
>

<!ELEMENT Lagerung (#PCDATA)>
<!ATTLIST Lagerung
    Art CDATA #IMPLIED
>

<!ELEMENT Zustand (#PCDATA)>

<!ELEMENT Bestand_Kurz (#PCDATA)>

<!ELEMENT archref (#PCDATA)>
<!ATTLIST archref
    archref_Fkt (siehe_auch | Abschrift | Original |
Entnommen_aus | Separiertes_Material) #IMPLIED
    Fkt CDATA #IMPLIED
    archref_Art (Findmittel | Bestand | Akte | Drucksache |
Urkunde | Siegel | Karte | Plakat | Foto | Film) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT bibref (#PCDATA)>
<!ATTLIST bibref
    bibref_Art (Literatur | Amtsdrucksache | Edition)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT FM_ref (#PCDATA)>

<!ELEMENT altübform (#PCDATA)>
<!ATTLIST altübform
    altübform_Art (Lesefilm | Sicherungsfilm | Microfiche |
Datei) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Register (%e.link;)*>
<!ATTLIST Register
    Register_Art (Person | Sache | Ort) #IMPLIED
    Art CDATA #IMPLIED
    Thesaurusreferenz CDATA #IMPLIED
>

```

```

<!ELEMENT Vwint (#PCDATA)>
<!ATTLIST Vwint
    Vwint_Art (VE | Index | FM_Seite) #IMPLIED
    idref IDREFS #IMPLIED
>

<!ELEMENT Vwext (#PCDATA)>
<!ATTLIST Vwext
    Vwext_Art (analog | digital) #IMPLIED
    dateiref CDATA #IMPLIED
    href CDATA #IMPLIED
>

<!ELEMENT Klass_Nr (#PCDATA)>

<!ELEMENT Klass_Titel (#PCDATA)>

<!ELEMENT Serientitel (#PCDATA)>

<!ELEMENT Enthaelte (#PCDATA | p)*>
<!ATTLIST Enthaelte
    Enth_Art (Darin | Enthaelte | Enthaelte_auch | Enthaelte_nur
| Enthaelte_ua | Enthaelte_va | Hierin | Intus) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Dateiname (Dateiname?, Vwext*, Erstellung?,
Ueberarbeitung*, Bem*, Hilfsfeld*)>

<!ELEMENT Dateiname (#PCDATA)>

<!ELEMENT Erstellung (Datum, Bearbeiter, Institution?, Bem*)>

<!ELEMENT Bearbeiter (#PCDATA)>

<!ELEMENT Ueberarbeitung (Datum, Bearbeiter, Institution?,
Bem*)+>

<!ELEMENT Findmittel_Info (FM_Name, FM_Sig?, Laufzeit?,
Inhaltsverzeichnis*, Einleitung?, (Sperrvermerk | Bearbeiter |
Datum | Bestand_Info | %e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT FM_Name (#PCDATA | Band)*>

<!ELEMENT FM_Sig (#PCDATA)>

<!ELEMENT Inhaltsverzeichnis (#PCDATA | p)*>

<!ELEMENT Einleitung (Text | Bibliographie | Abkuerzungen | p |
%e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT Text (#PCDATA | FM_Seite | Fussnote | p |
%e.Verweis; | %e.link;)*>
<!ATTLIST Text
    Text_Art (Bestandsbeschreibung | Bestandsinhalt |
Bewertung | Erschließung | Prov_Geschichte) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Fussnote (#PCDATA | %e.Verweis; | %e.link;)*>
<!ATTLIST Fussnote
    ID ID #IMPLIED
>

<!ELEMENT Bibliographie (#PCDATA | p | bibref)*>
<!ATTLIST Bibliographie
    Publ_Art (Literatur | Amtsdrucksachen | Editionen)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Abkuerzungen (Text | Abk_Eintrag)>

<!ELEMENT Abk_Eintrag (Kurz, Lang, Bem*, Hilfsfeld)*>

```

```

<!ELEMENT Vwint (#PCDATA)>
<!ATTLIST Vwint
    Vwint_Art (VE | Index) #IMPLIED
    idref IDREFS #IMPLIED
>

<!ELEMENT Vwext (#PCDATA)>
<!ATTLIST Vwext
    Vwext_Art (analog | digital) #IMPLIED
    dateiref CDATA #IMPLIED
    href CDATA #IMPLIED
>

<!ELEMENT Klass_Nr (#PCDATA)>

<!ELEMENT Klass_Titel (#PCDATA)>

<!ELEMENT Serientitel (#PCDATA)>

<!ELEMENT Enthaelte (#PCDATA)>
<!ATTLIST Enthaelte
    Enth_Art (Darin | Enthaelte | Enthaelte_auch | Enthaelte_nur
| Enthaelte_ua | Enthaelte_va | Hierin | Intus) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Bearbeiter (#PCDATA)>

<!ELEMENT Findmittel_Info (FM_Name, FM_Sig?, Laufzeit?,
Einleitung?, (Sperrvermerk | Bearbeiter | Bestand_Info |
%e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT FM_Name (#PCDATA)>

<!ELEMENT FM_Sig (#PCDATA)>

<!ELEMENT Einleitung (Text | Bibliographie | Abkuerzungen |
Bem)*>

<!ELEMENT Text (#PCDATA)>
<!ATTLIST Text
    Text_Art (Bestandsbeschreibung | Bestandsinhalt |
Bewertung | Erschließung | Prov_Geschichte) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Bibliographie (bibref)*>
<!ATTLIST Bibliographie
    Publ_Art (Literatur | Amtsdrucksachen | Editionen)
#IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Abkuerzungen (Abk_Eintrag)*>

<!ELEMENT Abk_Eintrag (Kurz, Lang, Bem)>

```

```

<!ELEMENT Kurz (#PCDATA)>

<!ELEMENT Lang (#PCDATA)>

<!ELEMENT Bestand_Info (Bestandsname?, Bestand_Sig?,
Bestand_Kurz?, Laufzeit?, Text*, Provenienz*, Vor_Prov*,
Abg_Stelle*, Akzession*, Sperrvermerk*, Rechtsstatus?, Umfang*,
Nr*, Bearbeiter*, Lagerung?, (%e.Verweis;)*, Bem*, Hilfsfeld*)>
<!ATTLIST Bestand_Info
    %a.bezug;
    Art CDATA #IMPLIED
>

<!ELEMENT Bestandsname (#PCDATA)>

<!ELEMENT Bestand_Sig (#PCDATA | Sig_nr | Sig_alphanum |
Hilfsfeld)*>

<!ELEMENT Klassifikation (%e.Klass_Info;,
Verzeichnungseinheiten*, Klassifikation*)>
<!ATTLIST Klassifikation
    %a.level;
>

<!ELEMENT Verzeichnungseinheiten (Sachakte | Fallakte |
Bandserie | Grundbuch_akte | Urkunde | Siegel | Karte | Foto_Plakat
| Film | Bem)+>

<!ELEMENT Sachakte (%e.VE_Basis;, (Enthaeht | %e.VE_Option;
| Nr | Az | Vorgang_Sachakte | Vorgang_Urkunde | Vorgang_Karte |
Vorgang_Foto)*)>
<!ATTLIST Sachakte
    ID ID #IMPLIED
    Vorgang (nein | ja) #IMPLIED
>

<!ELEMENT Az (#PCDATA)>

<!ELEMENT Rechtsstatus (#PCDATA)>

<!ELEMENT Vorgang_Sachakte (Laufzeit*, Titel, (Enthaeht |
%e.VE_Option; | Az)*)>

<!ELEMENT Vorgang_Urkunde (Laufzeit*, Beschreibung+,
Formalbeschreibung?, (%e.VE_Option; | Person | Ort |
Beglaubigungsart | Siegel)*)>

<!ELEMENT Vorgang_Karte (Laufzeit*, Titel+, Beschreibung?,
(Enthaeht | %e.VE_Option; | Kartentyp | Einzeichnung | Az |
Massstab | Topogr_Daten | Person | Institution | Ort | Ausfuehrung |
Material | Entstehungsstufe | Auflage | Format | Nebenkarten)*)>
<!ELEMENT Vorgang_Foto (Datierung*, Titel+, (Enthaeht |
Beschreibung | %e.VE_Option; | Anlass | Person | Institution | Ort |
Format | Farbe | Material | Entstehungsstufe)*)>

<!ELEMENT Fallakte (Signatur, Laufzeit?, Titel?, (Enthaeht |
%e.VE_Option; | Person | Institution | Sachverhalt | Datum | Nr | Ort
| Anschrift | Az | Prozessart | Instanz | Beweismittel |
Formalbeschreibung)*)>
<!ATTLIST Fallakte
    ID ID #IMPLIED
    FA_Art (Justiz | RKG-Prozess | Patient | Personal |
Firmen | Hauptakte | Beiakte | Entschaedigung | Entnazifizierung |
Einbuengerung | Soziales | Polizei) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Sachverhalt (#PCDATA)>
<!ATTLIST Sachverhalt
    Sachv_Art (Diagnose | Delikt | Streitgegenstand |
Verwaltungsverfahren) #IMPLIED
    Art CDATA #IMPLIED
>

```

```

<!ELEMENT Kurz (#PCDATA)>

<!ELEMENT Lang (#PCDATA)>

<!ELEMENT Bestand_Info (Bestandsname?, Bestand_Sig?,
Bestand_Kurz?, Laufzeit?, Text*, Provenienz*, Vor_Prov*,
Abg_Stelle*, Akzession*, Sperrvermerk*, Rechtsstatus?, Umfang*,
Bearbeiter*, Lagerung?, (%e.Verweis;)*, Bem*)>
<!ATTLIST Bestand_Info
    %a.bezug;
    Art CDATA #IMPLIED
>

<!ELEMENT Bestandsname (#PCDATA)>

<!ELEMENT Bestand_Sig (#PCDATA)>

<!ELEMENT Klassifikation (%e.Klass_Info;,
Verzeichnungseinheiten*, Klassifikation*)>
<!ATTLIST Klassifikation
    %a.level;
>

<!ELEMENT Verzeichnungseinheiten (Sachakte | Fallakte |
Bandserie | Grundbuch_akte | Urkunde | Siegel | Karte | Foto_Plakat
| Film | Bem)+>

<!ELEMENT Sachakte (%e.VE_Basis;, (Enthaeht | %e.VE_Option;
| Az | Vorgang_Sachakte | Vorgang_Urkunde | Vorgang_Karte |
Vorgang_Foto)*)>
<!ATTLIST Sachakte
    ID ID #IMPLIED
    Vorgang (nein | ja) #IMPLIED
>

<!ELEMENT Az (#PCDATA)>

<!ELEMENT Rechtsstatus (#PCDATA)>

<!ELEMENT Vorgang_Sachakte (Laufzeit?, Titel, (Enthaeht |
%e.VE_Option; | Az)*)>

<!ELEMENT Vorgang_Urkunde (Laufzeit?, Beschreibung+,
Formalbeschreibung?, (%e.VE_Option; | Person | Ort |
Beglaubigungsart | Siegel)*)>

<!ELEMENT Vorgang_Karte (Laufzeit?, Titel+, Beschreibung?,
(Enthaeht | %e.VE_Option; | Kartentyp | Einzeichnung | Az |
Massstab | Topogr_Daten | Person | Institution | Ort | Ausfuehrung |
Material | Entstehungsstufe | Auflage | Format | Nebenkarten)*)>
<!ELEMENT Vorgang_Foto (Laufzeit?, Titel+, (Enthaeht |
Beschreibung | %e.VE_Option; | Anlass | Person | Institution | Ort |
Format | Farbe | Material | Entstehungsstufe)*)>

<!ELEMENT Fallakte (Signatur, Laufzeit?, Titel?, (Enthaeht |
%e.VE_Option; | Person | Institution | Sachverhalt | Datum | Nr | Ort
| Anschrift | Az | Prozessart | Instanz | Beweismittel |
Formalbeschreibung)*)>
<!ATTLIST Fallakte
    ID ID #IMPLIED
    FA_Art (Justiz | RKG-Prozess | Patient | Personal |
Firmen | Hauptakte | Beiakte | Entschaedigung | Entnazifizierung |
Einbuengerung | Soziales | Polizei) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Sachverhalt (#PCDATA)>
<!ATTLIST Sachverhalt
    Sachv_Art (Diagnose | Delikt | Streitgegenstand |
Verwaltungsverfahren) #IMPLIED
    Art CDATA #IMPLIED
>

```

```

<!ELEMENT Prozessart (#PCDATA)>

<!ELEMENT Instanz (#PCDATA | Instanzbez | Laufzeit | Az)*>
<!ATTLIST Instanz
  Stufe (01 | 02 | 03 | 04 | 05) #IMPLIED
  Art CDATA #IMPLIED
>

<!ELEMENT Instanzbez (#PCDATA)>

<!ELEMENT Beweismittel (#PCDATA)>

<!ELEMENT Bandserie (%e.Serien_Info;, (Sachakte | Fallakte |
Bandserie)*)*>
<!ATTLIST Bandserie
  %a.level;
>

<!ELEMENT Grundbuch_akte (Gemarkung, Band?, Blatt?, Nr*,
Signatur?, Laufzeit?, (%e.VE_Option; | Seite | Eigentuermer |
Anschrift)*)>
<!ATTLIST Grundbuch_akte
  Gba_Art (GB | HYB | EH | LG | GA | HYA | EB)
#IMPLIED
  Art CDATA #IMPLIED
>

<!ELEMENT Gemarkung (#PCDATA | Ort)*>

<!ELEMENT Blatt (#PCDATA | Nr)*>

<!ELEMENT Seite (#PCDATA | Nr)*>

<!ELEMENT Eigentuermer (#PCDATA | Person | Institution)*>

<!ELEMENT Urkunde (Signatur, Nr?, Laufzeit?, Beschreibung+,
Formalbeschreibung?, (%e.VE_Option; | Person | Institution | Ort |
Beglaubigungsart | Siegel)*)>
<!ATTLIST Urkunde
  ID ID #IMPLIED
>

<!ELEMENT Beschreibung (#PCDATA | p | Fussnote)*>
<!ATTLIST Beschreibung
  Besch_Art (Regest | Kopfrege | Kurzregest | Vollregest
| Urkundentext | Plakattext | Begleittext) #IMPLIED
  Art CDATA #IMPLIED
>

<!ELEMENT Formalbeschreibung (#PCDATA | Material | Sprache
| Ueberlieferung | Zustand | Format | Fussnote | Vermerke |
Siegel_ang | Siegel_vorh | p | %e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT Material (#PCDATA)>

<!ELEMENT Sprache (#PCDATA)>

<!ELEMENT Ueberlieferung (#PCDATA)>

<!ELEMENT Format (#PCDATA | Hoehe | Breite |
Durchmesser)*>
<!ATTLIST Format
  M_Mass (mm | cm | m) #IMPLIED
>

<!ELEMENT Hoehe (#PCDATA)>

<!ELEMENT Breite (#PCDATA)>

<!ELEMENT Durchmesser (#PCDATA)>

<!ELEMENT Vermerke (#PCDATA | p)*>
<!ATTLIST Vermerke

```

```

<!ELEMENT Prozessart (#PCDATA)>

<!ELEMENT Instanz (#PCDATA)>
<!ATTLIST Instanz
  Stufe (01 | 02 | 03 | 04 | 05) #IMPLIED
  Art CDATA #IMPLIED
>

<!ELEMENT Instanzbez (#PCDATA)>

<!ELEMENT Beweismittel (#PCDATA)>

<!ELEMENT Bandserie (%e.Serien_Info;, (Sachakte | Fallakte |
Bandserie)*)*>
<!ATTLIST Bandserie
  %a.level;
>

<!ELEMENT Grundbuch_akte (Gemarkung, Band?, Blatt?,
Signatur?, Laufzeit?, (%e.VE_Option; | Seite | Eigentuermer |
Anschrift)*)>
<!ATTLIST Grundbuch_akte
  Gba_Art (GB | HYB | EH | LG | GA | HYA | EB)
#IMPLIED
  Art CDATA #IMPLIED
>

<!ELEMENT Gemarkung (#PCDATA)>

<!ELEMENT Blatt (#PCDATA)>

<!ELEMENT Seite (#PCDATA)>

<!ELEMENT Eigentuermer (Person | Institution)*>

<!ELEMENT Urkunde (Signatur, Nr?, Laufzeit?, Beschreibung+,
Formalbeschreibung?, (%e.VE_Option; | Person | Institution | Ort |
Beglaubigungsart | Siegel)*)>
<!ATTLIST Urkunde
  ID ID #IMPLIED
>

<!ELEMENT Beschreibung (#PCDATA)>
<!ATTLIST Beschreibung
  Besch_Art (Regest | Kopfrege | Kurzregest | Vollregest
| Urkundentext | Plakattext | Begleittext) #IMPLIED
  Art CDATA #IMPLIED
>

<!ELEMENT Formalbeschreibung (Material | Sprache |
Ueberlieferung | Zustand | Format | Vermerke | Siegel_ang |
Siegel_vorh | %e.Verweis; | Bem | Hilfsfeld)*>

<!ELEMENT Material (#PCDATA)>

<!ELEMENT Sprache (#PCDATA)>

<!ELEMENT Ueberlieferung (#PCDATA)>

<!ELEMENT Format (#PCDATA)>
<!ATTLIST Format
  M_Mass (mm | cm | m) #IMPLIED
>

<!ELEMENT Vermerke (#PCDATA)>
<!ATTLIST Vermerke

```

```

Ver_Art (RÄ¼ck | Kanzlei | Registratur | Tax)
#IMPLIED
Art CDATA #IMPLIED
>
<!ELEMENT Siegel_ang (#PCDATA)>
<!ELEMENT Siegel_vorh (#PCDATA)>
<!ELEMENT Beglaubigungsart (#PCDATA)>
<!ELEMENT Siegel (#PCDATA | Signatur | Person | Institution |
Nr | Laufzeit | Siegeltyp | Form | Anbringung | Format | Farbe |
Material | Siegelbild | Umschrift | Sperrvermerk | Zustand | p |
%e.Verweis; | Bem | Hilfsfeld)*>
<!ATTLIST Siegel
ID ID #IMPLIED
>
<!ELEMENT Siegeltyp (#PCDATA)>
<!ELEMENT Form (#PCDATA)>
<!ELEMENT Anbringung (#PCDATA)>
<!ELEMENT Farbe (#PCDATA)>
<!ELEMENT Siegelbild (#PCDATA)>
<!ELEMENT Umschrift (#PCDATA)>
<!ELEMENT Karte (Signatur, Laufzeit?, Titel+, Beschreibung?,
(%e.VE_Option; | Ort | Enthalt | Kartentyp | Einzeichnung | Az |
Massstab | Topogr_Daten | Person | Institution | Ausfuehrung |
Material | Entstehungsstufe | Auflage | Format | Nebenkarten)*>
<!ATTLIST Karte
ID ID #IMPLIED
>
<!ELEMENT Kartentyp (#PCDATA)>
<!ELEMENT Einzeichnung (#PCDATA)>
<!ELEMENT Massstab (#PCDATA)>
<!ATTLIST Massstab
Massstab_Art (Original | Umrechnung) #IMPLIED
>
<!ELEMENT Topogr_Daten (#PCDATA | TK | Nr | GK_hoch |
GK_rechts | GK_Identifikation | Breitengrad | Laengengrad | Bem |
Hilfsfeld)*>
<!ELEMENT TK (#PCDATA)>
<!ELEMENT GK_hoch (#PCDATA)>
<!ELEMENT GK_rechts (#PCDATA)>
<!ELEMENT GK_Identifikation (#PCDATA)>
<!ELEMENT Breitengrad (#PCDATA)>
<!ELEMENT Laengengrad (#PCDATA)>
<!ATTLIST Laengengrad
Meridian_Art (Greenwich | Ferro) #IMPLIED
Art CDATA #IMPLIED
>
<!ELEMENT Ausfuehrung (#PCDATA)>
<!ELEMENT Entstehungsstufe (#PCDATA)>
<!ELEMENT Auflage (#PCDATA)>

```

```

Ver_Art (RÄ¼ck | Kanzlei | Registratur | Tax)
#IMPLIED
Art CDATA #IMPLIED
>
<!ELEMENT Siegel_ang (#PCDATA)>
<!ELEMENT Siegel_vorh (#PCDATA)>
<!ELEMENT Beglaubigungsart (#PCDATA)>
<!ELEMENT Siegel (Beschreibung | Signatur | Person | Institution |
Nr | Laufzeit | Siegeltyp | Form | Anbringung | Format | Farbe |
Material | Siegelbild | Umschrift | Sperrvermerk | Zustand |
%e.Verweis; | Bem)*>
<!ATTLIST Siegel
ID ID #IMPLIED
>
<!ELEMENT Siegeltyp (#PCDATA)>
<!ELEMENT Form (#PCDATA)>
<!ELEMENT Anbringung (#PCDATA)>
<!ELEMENT Farbe (#PCDATA)>
<!ELEMENT Siegelbild (#PCDATA)>
<!ELEMENT Umschrift (#PCDATA)>
<!ELEMENT Karte (Signatur, Laufzeit?, Titel+, Beschreibung?,
(%e.VE_Option; | Ort | Enthalt | Kartentyp | Einzeichnung | Az |
Massstab | Topogr_Daten | Person | Institution | Ausfuehrung |
Material | Entstehungsstufe | Auflage | Format | Nebenkarten)*>
<!ATTLIST Karte
ID ID #IMPLIED
>
<!ELEMENT Kartentyp (#PCDATA)>
<!ELEMENT Einzeichnung (#PCDATA)>
<!ELEMENT Massstab (#PCDATA)>
<!ATTLIST Massstab
Massstab_Art (Original | Umrechnung) #IMPLIED
>
<!ELEMENT Topogr_Daten (TK | GK_hoch | GK_rechts |
GK_Identifikation | Breitengrad | Laengengrad | Bem | Hilfsfeld)*>
<!ELEMENT TK (#PCDATA)>
<!ELEMENT GK_hoch (#PCDATA)>
<!ELEMENT GK_rechts (#PCDATA)>
<!ELEMENT GK_Identifikation (#PCDATA)>
<!ELEMENT Breitengrad (#PCDATA)>
<!ELEMENT Laengengrad (#PCDATA)>
<!ATTLIST Laengengrad
Meridian_Art (Greenwich | Ferro) #IMPLIED
Art CDATA #IMPLIED
>
<!ELEMENT Ausfuehrung (#PCDATA)>
<!ELEMENT Entstehungsstufe (#PCDATA)>
<!ELEMENT Auflage (#PCDATA)>

```



```

<!ELEMENT Nebenkarten (#PCDATA | NK_Bez | Anzahl | Bem |
Hilfsfeld)*>

<!ELEMENT NK_Bez (#PCDATA)>

<!ELEMENT Anzahl (#PCDATA)>

<!ELEMENT Foto_Plakat (Signatur, Laufzeit?, Titel+, (Enthaeft |
%e.VE_Option; | Anlass | Beschreibung | Person | Institution | Ort |
Format | Farbe | Material | Entstehungsstufe)*>
<!ATTLIST Foto_Plakat
    ID ID #IMPLIED
>

<!ELEMENT Anlass (#PCDATA)>

<!ELEMENT Film (Titel+, Signatur?, Laufzeit?, Inhalt?, (Person |
Institution | Sprache | Praedikate | %e.VE_Option; | Datum | Form |
Format | Farbe | Ton | Dauer | Laenge | Entstehungsstufe | Material |
Film_Einheit)*>
<!ATTLIST Film
    ID ID #IMPLIED
>

<!ELEMENT Inhalt (#PCDATA | Szene_Anf | Szene_End | p)*>

<!ELEMENT Szene_Anf (#PCDATA)>

<!ELEMENT Szene_End (#PCDATA)>

<!ELEMENT Praedikate (#PCDATA)>

<!ELEMENT Ton (#PCDATA)>

<!ELEMENT Dauer (#PCDATA)>
<!ATTLIST Dauer
    Dauer_Art (Referenz | Gemessen) #IMPLIED
    Vorfuehrgeschwindigkeit CDATA #IMPLIED
>

<!ELEMENT Laenge (#PCDATA)>
<!ATTLIST Laenge
    Laenge_Art (Referenz | Gemessen) #IMPLIED
>

<!ELEMENT Film_Einheit (Titel*, Signatur, Laufzeit?,
(%e.VE_Option; | Datum | Sprache | Form | Format | Farbe | Ton |
Dauer | Laenge | Entstehungsstufe | Material)*>
<!ATTLIST Film_Einheit
    ID ID #IMPLIED
>

<!ELEMENT Indizes (Bem*, Index+)>

<!ELEMENT Index (Bem*, Indexeintrag+)>
<!ATTLIST Index
    Index_Art (Sache | Ort | Person) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Indexeintrag (Indexbegriff, FM_Seite*, Signatur*,
Bestellsig*, Nr*, (%e.Verweis;)*, Indexeintrag*, Hilfsfeld*)>
<!ATTLIST Indexeintrag
    %a.level;
>

<!ELEMENT Indexbegriff (#PCDATA)>
<!ATTLIST Indexbegriff
    ID ID #IMPLIED
>

<!ELEMENT Konkordanz (Bem*, Kon_Eintrag+)>

```

```

<!ELEMENT Nebenkarten (#PCDATA)>

<!ELEMENT Foto_Plakat (Signatur, Laufzeit?, Titel+, (Enthaeft |
%e.VE_Option; | Anlass | Beschreibung | Person | Institution | Ort |
Format | Farbe | Material | Entstehungsstufe)*>
<!ATTLIST Foto_Plakat
    ID ID #IMPLIED
>

<!ELEMENT Anlass (#PCDATA)>

<!ELEMENT Film (Titel+, Signatur?, Laufzeit?, Inhalt?, (Person |
Institution | Sprache | Praedikate | %e.VE_Option; | Datum | Form |
Format | Farbe | Ton | Dauer | Laenge | Entstehungsstufe | Material |
Film_Einheit)*>
<!ATTLIST Film
    ID ID #IMPLIED
>

<!ELEMENT Inhalt (#PCDATA)>

<!ELEMENT Praedikate (#PCDATA)>

<!ELEMENT Ton (#PCDATA)>

<!ELEMENT Dauer (#PCDATA)>
<!ATTLIST Dauer
    Dauer_Art (Referenz | Gemessen) #IMPLIED
    Vorfuehrgeschwindigkeit CDATA #IMPLIED
>

<!ELEMENT Laenge (#PCDATA)>
<!ATTLIST Laenge
    Laenge_Art (Referenz | Gemessen) #IMPLIED
>

<!ELEMENT Film_Einheit (Titel*, Signatur, Laufzeit?,
(%e.VE_Option; | Datum | Sprache | Form | Format | Farbe | Ton |
Dauer | Laenge | Entstehungsstufe | Material)*>
<!ATTLIST Film_Einheit
    ID ID #IMPLIED
>

<!ELEMENT Indizes (Bem*, Index+)>

<!ELEMENT Index (Bem*, Indexeintrag+)>
<!ATTLIST Index
    Index_Art (Sache | Ort | Person) #IMPLIED
    Art CDATA #IMPLIED
>

<!ELEMENT Indexeintrag (Indexbegriff, Signatur*, Bestellsig*,
Nr*, (%e.Verweis;)*, Indexeintrag*)>
<!ATTLIST Indexeintrag
    %a.level;
>

<!ELEMENT Indexbegriff (#PCDATA)>
<!ATTLIST Indexbegriff
    ID ID #IMPLIED
>

<!ELEMENT Konkordanz (Bem*, Kon_Eintrag+)>

```

<!ELEMENT Kon_Eintrag (Signatur | Altsig | FM_Seite | Bestellsig | Lagerung | Hilfsfeld)+> | <!ELEMENT Kon_Eintrag (Signatur | Altsig | Bestellsig | Lagerung)+>

C – Quellcode

C.1. AsterVE.php

```
<?php
include("functions.inc.php");
include("masksfm.inc.php");
include("masksarchiv.inc.php");
include("elementsfm.inc.php");
include("elementsarchiv.inc.php");
include("attributes.inc.php");
$userid='';
$state='';
$vedom;
$vefile;
$overviewdom;
$overviewfile;
$window = new GtkWindow();
$pixbuf = GdkPixbuf::new_from_file('img/astericon.png');
$window->set_default_icon($pixbuf);
$window->set_title("Aster");
$window->set_position('GTK_WIN_POS_CENTER');
$window->set_default_size(800,600);
$window->connect_simple("destroy", "onQuit");
$window->show();
$vbox = new GtkVBox(false, 5);
$menuBar = new GtkMenuBar;
$menuFile = new GtkMenuItem('_Datei');
$subMenuFile = new GtkMenu;
$itemFile1= new GtkImageMenuItem(GTK::STOCK_SAVE);
$itemFile1->connect('activate', 'savefile');
$itemFile2= new GtkImageMenuItem(GTK::STOCK_CONVERT);
$itemFile2->connect('activate', 'openfile');
$itemFile4= new GtkMenuItem;
$itemFile4->set_sensitive(false);
$itemFile5= new GtkImageMenuItem(GTK::STOCK_QUIT);
$itemFile5->connect_simple("activate", 'onQuit');
$subMenuFile->append($itemFile1);
$subMenuFile->append($itemFile2);
$subMenuFile->append($itemFile4);
$subMenuFile->append($itemFile5);
$menuFile->set_submenu($subMenuFile);
$menuEdit = new GtkMenuItem('_Einstellungen');
$subMenuEdit= new GtkMenu;
$itemEdit1= new GtkImageMenuItem(GTK::STOCK_EDIT);
$itemEdit1->connect('activate', 'onEdit');
$subMenuEdit->append($itemEdit1);
$menuEdit->set_submenu($subMenuEdit);
$menuHelp = new GtkMenuItem('_Hilfe');
$subMenuHelp= new GtkMenu;
$itemHelp1= new GtkImageMenuItem(GTK::STOCK_ABOUT);
$itemHelp1->connect('activate', 'onHelp');
$subMenuHelp->append($itemHelp1);
$menuHelp->set_submenu($subMenuHelp);
$menuBar->append($menuFile);
$menuBar->append($menuEdit);
$menuBar->append($menuHelp);
$vbox->pack_start($menuBar, false, false, 0);
$hpane = new GtkHPaned();
$hpane->set_position(250);
$vpane = new GtkVPaned();
$vpane->set_position(250);
$vboxoverview = new GtkVBox();
$besttree = new GtkTreeView;
$bestmodel = new GtkTreeStore(Gtk::TYPE_OBJECT, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
$bestcolumn1 = new GtkTreeViewColumn();
$bestcolumn1->set_title('Tektonik');
$bestcell_renderer1 = new GtkCellRendererPixBuf();
$bestcell_renderer2 = new GtkCellRendererText();
$bestcolumn1->pack_start($bestcell_renderer1, false);
$bestcolumn1->pack_start($bestcell_renderer2, true);
$bestcell_renderer2->set_property('width', 200);
$bestcolumn1->set_attributes($bestcell_renderer1, 'pixbuf', 0);
```

```

$bestcolumn1->set_attributes($bestcell_renderer2, 'text', 1);
$besttree->append_column($bestcolumn1);
$bestcolumn2 = new GtkTreeViewColumn();
$bestcolumn2->set_title('Ids');
$bestcell_renderer2 = new GtkCellRendererText();
$bestcolumn2->pack_start($bestcell_renderer2, true);
$bestcell_renderer2->set_property('width', 0);
$bestcolumn2->set_attributes($bestcell_renderer2, 'text', 2);
//$besttree->append_column($bestcolumn2);
$besttree->set_model($bestmodel);
$selection = $besttree->get_selection();
$selection->connect('changed', 'selectOVTreeRow');
$besttree->connect_after('event', 'clickedTree');
$besttree->set_events(Gdk::_2BUTTON_PRESS | Gdk::BUTTON_RELEASE);
model_overview();
$bestscroll = new GtkScrolledWindow();
$bestscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bestscroll->add($besttree);
$bestscroll->show_all();
$vboxoverview->pack_start($bestscroll, true, true, 0);
$vpave->add1($vboxoverview);
$vboxve = new GtkVBox();
$vetree = new GtkTreeView;
$vemodel = new GtkTreeStore(Gtk::TYPE_OBJECT, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
$vecolumn1 = new GtkTreeViewColumn();
$vecolumn1->set_title('Findmittel');
$vecell_renderer1 = new GtkCellRendererPixBuf();
$vecell_renderer2 = new GtkCellRendererText();
$vecolumn1->pack_start($vecell_renderer1, false);
$vecolumn1->pack_start($vecell_renderer2, true);
$vecell_renderer2->set_property('width', 200);
$vecolumn1->set_attributes($vecell_renderer1, 'pixbuf', 0);
$vecolumn1->set_attributes($vecell_renderer2, 'text', 1);
$vetree->append_column($vecolumn1);
$vecolumn2 = new GtkTreeViewColumn();
$vecolumn2->set_title('Ids');
$vecell_renderer2 = new GtkCellRendererText();
$vecolumn2->pack_start($vecell_renderer2, true);
$vecell_renderer2->set_property('width', 0);
$vecolumn2->set_attributes($vecell_renderer2, 'text', 2);
//$vetree->append_column($vecolumn2);
$vetree->set_model($vemodel);
$selection = $vetree->get_selection();
$selection->connect('changed', 'selectVETreeRow');
$vetree->connect_after('event', 'clickedTree');
$vetree->set_events(Gdk::_2BUTTON_PRESS | Gdk::BUTTON_RELEASE);
$vescroll = new GtkScrolledWindow();
$vescroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$vescroll->add($vetree);
$vescroll->show_all();
$vboxve->pack_start($vescroll, true, true, 0);
$vpave->add2($vboxve);
$hpane->add1($vpave);
$vboxmask = new GtkVBox();
$note = new GtkNotebook();
$vboxmask->pack_start($note, true, true, 0);
$hpane->add2($vboxmask);
$vbox->pack_start($hpane, true, true, 0);
$window->add($vbox);
$window->show_all();
login();
gtk::main();
?>

```

C.2. attributes.inc.php

```

<?php
function attributegrundbuch($id) {
    if(!empty($id)) {
        global $vedom;

```

```

        $element=$vedom->getElementById($id);
        $titart=$element->getAttribute("Gba_Art");
        if ($titart == 'B') {$titart='Grundbuch';}
        if ($titart == 'HYB') {$titart='Hypothekenbuch';}
        if ($titart == 'EH') {$titart='Erbhofrolle';}
        if ($titart == 'LG') {$titart='Landgüter';}
        if ($titart == 'GA') {$titart='Grundakte';}
        if ($titart == 'HYA') {$titart='Hypothekenakte';}
        if ($titart == 'EB') {$titart='Erbhofakte';}
        $art=$element->getAttribute("Art");
        if (empty($titart)){$attr=$art;}
        else {$attr=$titart;}
    } else {$attr='';}
    $hbox= new GtkHBox(false, 0);
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Grundbuch', 'Hypothekenbuch', 'Erbhofrolle', 'Landgüter', 'Grundakte',
'Hypothekenakte', 'Erbhofakte', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $comboentry->get_child()->set_text($attr);
    $hbox->pack_start($comboentry, true, true, 5);
    return $hbox;
}
?>

```

C.3. elementsarchiv.inc.php

```

<?php
function archivelementaltuebform($altueb) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkListStore(Gtk::TYPE_STRING);
    $attrs->append(array(utf8_decode("Lesefilm")));
    $attrs->append(array(utf8_decode('Sicherungsfilm')));
    $attrs->append(array(utf8_decode('Microfiche')));
    $attrs->append(array(utf8_decode('Datei')));
    if(!empty($altueb)) {
        for($a=0;$a<$altueb->length;$a++) {
            $child=$altueb->item($a);
            $enthart=$child->getAttribute("altübform_Art");
            $art=$child->getAttribute("Art");
            if (empty($enthart)) {$attr=$art;}
            else {$attr=$enthart;}
            if (empty($attr)) {$attr="";}
            $value=$child->nodeValue;
            $enthmodel->append(array(utf8_decode($attr), utf8_decode($value)));
        }
    } else {$enthmodel->append(array('', ''));}
    $enthlist = new GtkTreeView;
    $enthattrcol = new GtkTreeViewColumn();
    $enthcol = new GtkTreeViewColumn();
    $enthattrcol->set_title('Type');

```

```

$enthcol->set_title('Referenz');
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable' , true);
//$enthcell_renderer2->set_property('has-entry' , false);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 1);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 0);
$enthcol->set_attributes($enthcell_renderer1, 'text', 1);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function archivelementanschrift($nr) {
    if (!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementarchivname($nr) {
    if (!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementarchref($archs) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);

```

```

$attrfkt= new GtkListStore(Gtk::TYPE_STRING);
$attrfkt->append(array(utf8_decode("siehe auch")));
$attrfkt->append(array(utf8_decode('Abschrift')));
$attrfkt->append(array(utf8_decode('Original')));
$attrfkt->append(array(utf8_decode('Entnommen aus')));
$attrfkt->append(array(utf8_decode('Separiertes Material')));
$attrart= new GtkListStore(Gtk::TYPE_STRING);
$attrart->append(array(utf8_decode("Findmittel")));
$attrart->append(array(utf8_decode('Bestand')));
$attrart->append(array(utf8_decode('Akte')));
$attrart->append(array(utf8_decode('Drucksache')));
$attrart->append(array(utf8_decode("Urkunde")));
$attrart->append(array(utf8_decode('Siegel')));
$attrart->append(array(utf8_decode('Karte')));
$attrart->append(array(utf8_decode('Plakat')));
$attrart->append(array(utf8_decode("Foto")));
$attrart->append(array(utf8_decode('Film')));
if(!empty($archs)) {
    for($a=0;$a<$archs->length;$a++) {
        $child=$archs->item($a);
        $enthfkt=$child->getAttribute("archref_Fkt");
        $fkt=$child->getAttribute("Fkt");
        if (empty($enthfkt)) {$attrf=$fkt;}
        else {$attrf=$enthfkt;}
        if (empty($attrf)) {$attrf="";}
        $enthart=$child->getAttribute("archref_Art");
        $art=$child->getAttribute("Art");
        if (empty($enthart)) {$attra=$art;}
        else {$attra=$enthart;}
        if (empty($attra)) {$attra="";}
        $value=$child->nodeValue;
        $enthmodel->append(array(utf8_decode($attrf), utf8_decode($attra),
utf8_decode($value)));
    }
}
} else {$enthmodel->append(array(' ', ' ', ' '));}
$enthlist = new GtkTreeView;
$enthattrfktcol = new GtkTreeViewColumn();
$enthattrartcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$enthattrfktcol->set_title('Type');
$enthattrartcol->set_title('Ref. auf');
$enthcol->set_title('Referenz');
$enthattrfktcol->set_resizable(true);
$enthattrartcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$enthcell_renderer3 = new GtkCellRendererCombo();
$enthattrfktcol->pack_start($enthcell_renderer2, true);
$enthattrartcol->pack_start($enthcell_renderer3, true);

```

```

$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer2->set_property('model', $attrfkt);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable' , true);
//$enthcell_renderer2->set_property('has-entry' , false);
$enthcell_renderer3->set_property('width', 100);
$enthcell_renderer3->set_property('model', $attrart);
$enthcell_renderer3->set_property('text-column', 0);
$enthcell_renderer3->set_property('editable' , true);
//$enthcell_renderer3->set_property('has-entry' , false);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 0);
$enthcell_renderer3->connect('edited', 'cell_edited', $enthmodel, 1);
$enthattrfktcol->set_attributes($enthcell_renderer2, 'text', 0);
$enthattrartcol->set_attributes($enthcell_renderer3, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
$enthlist->append_column($enthattrfktcol);
$enthlist->append_column($enthattrartcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function archivelementaufloesung($nr) {
    if(!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $datebutton=new GtkButton('');
    $datebutton->set_size_request(33, 25);
    $datebutton->connect_simple('clicked', 'calend');
    $image = new GtkImage();
    $image->set_from_file('img/datepicker.png');
    $datebutton->set_image($image);
    $hbox->pack_start($entry, true, true, 5);
    $hbox->pack_start($datebutton, false, false, 0);
    return $hbox;
}

function archivelementbem($bems) {
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING);
    if(!empty($bems)) {
        for($a=0;$a<$bems->length;$a++) {

```



```

        $schild=$bems->item($a);
        $value=$schild->nodeValue;
        $bemmodel->append(array(utf8_decode($value)));
    }
} else{$bemmodel->append(array(''));}
    $bemlist = new GtkTreeView;
    $bemcol = new GtkTreeViewColumn();
    $bemcol->set_title('Bemerkungen');
    $bemcol->set_resizable(true);
    $bemcell_renderer2= new GtkCellRendererText();
    $bemcol->pack_start($bemcell_renderer2, true);
    $bemcell_renderer2->set_property('width', 50);
    $bemcell_renderer2->set_property('editable', true);
    $bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 0);
    $bemcol->set_attributes($bemcell_renderer2, 'text', 0);
    $bemlist->append_column($bemcol);
    $bemlist->set_model($bemmodel);
    $bemscroll = new GtkScrolledWindow();
    $bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $bemscroll->add_with_viewport($bemlist);
    $bemscroll->show_all();
    return $bemscroll;
}

function archivelementbibliographie($biblio){
    $note= new GtkNotebook();
    $attr='';
    $bibrefs='';
    if (!empty($biblio)){
        for($a=0;$a<$biblio->length;$a++){
            $schild=$elements->item($a);
            $titart=$schild->getAttribute("Publ_Art");
            $art=$schild->getAttribute("Art");
            if (empty($titart)){ $attr=$art;}
            else {$attr=$titart;}
            if (hasChild($schild, 'bibref')==true){$bibrefs=get_tags($schild, 'bibref');}
            $bib=archivelementbibref($bibrefs);
            $vbox= new GtkVBox(false, 0);
            $comboentry = GtkComboBoxEntry::new_text();
            $data = array('Literatur', 'Amtdrucksachen', 'Editionen', '');
            foreach ($data as $str) {$comboentry->append_text($str);}
            $comboentry->get_child()->set_text($attr);
            if (empty($attr)){ $attr="Bibliographie";}
            $vbox->pack_start($comboentry, false, false, 5);
            $vbox->pack_start($bib, true, true, 0);
            $note->append_page($vbox, new GtkLabel($attr));
        }
    } else{$vbox= new GtkVBox(false, 0);
        $bib=archivelementbibref($bibrefs);
        $comboentry = GtkComboBoxEntry::new_text();
    }
}

```

```

    $data = array('Literatur', 'Amtsdrucksachen', 'Editionen', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $comboentry->get_child()->set_text($attr);
    $vbox->pack_start($comboentry, false, false, 5);
    $vbox->pack_start($bib, true, true, 0);
    $note->append_page($vbox, new GtkLabel('Bibliographie'));
}
return $note;
}

function archivelementbibref($bibs) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkListStore(Gtk::TYPE_STRING);
    $attrs->append(array(utf8_decode("Literatur")));
    $attrs->append(array(utf8_decode('Amtsdrucksache')));
    $attrs->append(array(utf8_decode('Edition')));
    $attrs->append(array(utf8_decode('')));
    if(!empty($bibs)) {
        for($a=0;$a<$bibs->length;$a++){
            $schild=$bibs->item($a);
            $enthart=$schild->getAttribute("bibref_Art");
            $art=$schild->getAttribute("Art");
            if (empty($enthart)) {$attr=$art;}
            else {$attr=$enthart;}
            if (empty($attr)) {$attr="";}
            $value=$schild->nodeValue;
            $enthmodel->append(array(utf8_decode($attr), utf8_decode($value)));
        }
    }else{$enthmodel->append(array('', ''));}
    $enthlist = new GtkTreeView;
    $enthattrcol = new GtkTreeViewColumn();
    $enthcol = new GtkTreeViewColumn();
    $enthattrcol->set_title('Type');
    $enthcol->set_title('Referenz');
    $enthattrcol->set_resizable(true);
    $enthcol->set_resizable(true);
    $enthcell_renderer1 = new GtkCellRendererText();
    $enthcell_renderer2 = new GtkCellRendererCombo();
    $enthattrcol->pack_start($enthcell_renderer2, true);
    $enthcol->pack_start($enthcell_renderer1, true);
    $enthcell_renderer2->set_property('width', 100);
    $enthcell_renderer1->set_property('editable', true);
    $enthcell_renderer2->set_property('model', $attrs);
    $enthcell_renderer2->set_property('text-column', 0);
    $enthcell_renderer2->set_property('editable' , true);
    // $enthcell_renderer2->set_property('has-entry' , false);
    $enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 1);
    $enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 0);
    $enthattrcol->set_attributes($enthcell_renderer2, 'text', 0);
    $enthcol->set_attributes($enthcell_renderer1, 'text', 1);
}

```

```

$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function archivelementemail($nr) {
    if (!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementgruendung($nr) {
    if (!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $datebutton=new GtkButton('');
    $datebutton->set_size_request(33, 25);
    $datebutton->connect_simple('clicked', 'calend');
    $image = new GtkImage();
    $image->set_from_file('img/datepicker.png');
    $datebutton->set_image($image);
    $hbox->pack_start($entry, true, true, 5);
    $hbox->pack_start($datebutton, false, false, 0);
    return $hbox;
}

function archivelementfax($nr) {
    if (!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementfindmitteldatei($dats) {

```

```

$bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
if(!empty($dats)) {
    for($a=0;$a<$dats->length;$a++){
        $child=$dats->item($a);
        $id=$child->getAttribute("id");
        $value=$child->nodeValue;
        $bemmodel->append(array($id, utf8_decode($value)));
    }
} else{$bemmodel->append(array('', ''));}

$bemlist = new GtkTreeView;
$idbemcol = new GtkTreeViewColumn();
$bemcol = new GtkTreeViewColumn();
$idbemcol->set_title('ID-Nummer');
$bemcol->set_title('Findmitteldateien');
$idbemcol->set_resizable(true);
$bemcol->set_resizable(true);
$bemcell_renderer1 = new GtkCellRendererText();
$bemcell_renderer2= new GtkCellRendererText();
$idbemcol->pack_start($bemcell_renderer1, true);
$bemcol->pack_start($bemcell_renderer2, true);
$bemcell_renderer2->set_property('width', 50);
$bemcell_renderer2->set_property('editable', true);
$bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
$idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
$bemcol->set_attributes($bemcell_renderer2, 'text', 1);
//$bemlist->append_column($idbemcol);
$bemlist->append_column($bemcol);
$bemlist->set_model($bemmodel);
$bemscroll = new GtkScrolledWindow();
$bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bemscroll->add_with_viewport($bemlist);
$bemscroll->show_all();
return $bemscroll;
}

function archivelementfmref($fms) {
    $saltmodel = new GtkListStore(Gtk::TYPE_STRING);
    if(!empty($fms)) {
        for($a=0;$a<$fms->length;$a++){
            $child=$fms->item($a);
            $value=$child->nodeValue;
            $saltmodel->append(array(utf8_decode($value)));
        }
    } else{$saltmodel->append(array(''));}

    $saltlist = new GtkTreeView;
    $saltcol = new GtkTreeViewColumn();
    $saltcol->set_title('Referenz');
    $saltcol->set_resizable(true);
    $saltcell_renderer1 = new GtkCellRendererText();

```

```

    $saltcol->pack_start($saltcell_renderer1, true);
    $saltcell_renderer1->set_property('width', 50);
    $saltcell_renderer1->set_property('editable', true);
    $saltcell_renderer1->connect('edited', 'cell_edited', $saltmodel, 0);
    $saltcol->set_attributes($saltcell_renderer1, 'text', 0);
    $saltlist->append_column($saltcol);
    $saltlist->set_model($saltmodel);
    $saltscroll = new GtkScrolledWindow();
    $saltscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $saltscroll->add_with_viewport($saltlist);
    $saltscroll->show_all();
    return $saltscroll;
}

function archivelementhomepage($nr) {
    if(!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementkompetenz($titel){
    if(!empty($titel)) {
        $text=utf8_decode($titel->nodeValue);
    } else {$text='';}

    $hbox= new GtkHBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $descbuffer->set_text($text);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $hbox->pack_start($entryscroll, true, true, 5);
    return $hbox;
}

function archivelementland($nr) {
    if(!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();

```

```

$entry->set_text($text);
$hbox->pack_start($entry, true, true, 5);
return $hbox;
}

function archivelementtektbeschreibung($besch) {
    if(!empty($besch)) {
        $text=utf8_decode($besch->nodeValue);
    } else {$text='';}

    $hbox= new GtkHBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $descbuffer->set_text($text);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $hbox->pack_start($entryscroll, true, true, 5);
    return $hbox;
}

function archivelementtektnr($nr) {
    if(!empty($nr)) {
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementtekttitel($titel){
    if(!empty($titel)) {
        $text=utf8_decode($titel->nodeValue);
    } else {$text='';}

    $hbox= new GtkHBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $descbuffer->set_text($text);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);

```

```

    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $hbox->pack_start($entryscroll, true, true, 5);
    return $hbox;
}

function archivelementtelefon($nr){
    if(!empty($nr)){
        $text=utf8_decode($nr->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function archivelementverweise($id){
    if (empty($id)){$sarch=''; $bib=''; $fm=''; $altueb='';}
    } else{
        global $overviewdom;
        $element=$overviewdom->getElementById($id);
        if (hasChild($element,'archref')==true){$sarch=get_tags($element,'archref');}
        else {$sarch='';}
        if (hasChild($element,'bibref')==true){$bib=get_tags($element,'bibref');}
        else {$bib='';}
        if (hasChild($element,'fmref')==true){$fm=get_tags($element,'fmref');}
        else {$fm='';}
        if (hasChild($element,'altuebform')==true){$altueb=get_tags($element,'altuebform');}
        else {$altueb='';}
    }
    $sarchref=archivelementarchref($sarch);
    $bibref=archivelementbibref($bib);
    $fmref=archivelementfmref($fm);
    $altuebform=archivelementaltuebform($altueb);

    $hbox= new GtkHBox(false, 5);
    $verweisnote= new GtkNoteBook();
    $verweisnote->append_page($sarchref, new GtkLabel('Archivalien'));
    $verweisnote->append_page($bibref, new GtkLabel('Literatur'));
    $verweisnote->append_page($fmref, new GtkLabel('Findmittel'));
    $verweisnote->append_page($altuebform, new GtkLabel('Alt. Überlieferung'));
    $hbox->pack_start($verweisnote, true, true, 5);
    return $hbox;
}
?>

```

C.4. elementsfm.inc.php

```
<?php
```

```

function elementabgstelle($id) {
    global $vedom;
    $list = new GtkTreeStore (Gtk::TYPE_STRING);
    $bestinfo=get_tags($vedom, "Bestand_Info");
    $anz=$bestinfo->length;
    for ($a=0;$a<$anz;$a++){
        $bestandinfo = $bestinfo->item($a);
        $kurz=''; $name=''; $sig='';
        if(hasChild($bestandinfo, 'Bestandsname')){
            $name=getChild($bestandinfo, 'Bestandsname');
            $name=$name->nodeValue;
        }
        if(hasChild($bestandinfo, 'Bestand_Kurz')){
            $kurz=getChild($bestandinfo, 'Bestand_Kurz');
            $kurz=$kurz->nodeValue;
        }
        if(hasChild($bestandinfo, 'Bestand_Sig')){
            $sig=getChild($bestandinfo, 'Bestand_Sig');
            $sig=$sig->nodeValue;
        }
        }
        if($anz == 1){$iter=null;
    } else {
        if(!empty($kurz)){
            $iter=$list->append(null, array(utf8_decode($kurz)));
        }
        elseif(!empty($sig)){
            $iter=$list->append(null, array(utf8_decode($sig)));
        }
        elseif(!empty($name)){
            $iter=$list->append(null, array(utf8_decode($name)));
        }
        else{$iter=$list->append(null, array('*'));}
    }
    if(tag_exists($bestandinfo, "Abg_Stelle")){
        $abgs=get_tags($bestandinfo, "Abg_Stelle");
        $anzahl=$abgs->length;
        for ($b=0;$b<$anzahl;$b++){
            $abg = $abgs->item($b);
            $list->append($iter, array(utf8_decode($abg->nodeValue)));
        }
    } else{$list->append($iter, array('*'));}
    if ($anz == 0){$list->append(null, array('*'));}
}
if(empty($id)){ $text='';}
else {$element=$vedom->getElementById($id);
    $text=utf8_decode($element->nodeValue);}
if ($anz == 0){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('abgstelle');
}

```



```

    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}
if ($anz >= 1){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('abgstelle');
    $comboentry = GtkComboBoxEntry::new_text();
    $comboentry->get_child()->set_text($text);
    $comboentry->set_model($list);
    $hbox->pack_start($comboentry, true, true, 5);
    return $hbox;
}
}

function elementabkuerzungen($id){
    $model = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_S
TRING);
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element,'Abk_Eintrag')==true){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Abk_Eintrag"){
                    $id=$child->getAttribute("idstef");
                    if (hasChild($child,'Kurz')==true){
                        $kurz=getChild($child,'Kurz');
                        $kurzval=$kurz->nodeValue;
                    } else {$kurzval='';}
                    if (hasChild($child,'Lang')==true){
                        $lang=getChild($child,'Lang');
                        $langval=$lang->nodeValue;
                    } else {$langval='';}
                    if (hasChild($child,'Bem')==true){
                        $bem=getChild($child,'Bem');
                        $bemval=$bem->nodeValue;
                    } else {$bemval='';}
                    $model->append(array($id, utf8_decode($kurzval), utf8_decode($langval),
utf8_decode($bemval)));
                }
            }
        } else{$model->append(array('',' ',' ',' '));}
    } else{$model->append(array('',' ',' ',' '));}
    $list = new GtkTreeView;
    $idcol = new GtkTreeViewColumn();
    $col1 = new GtkTreeViewColumn();
    $col2 = new GtkTreeViewColumn();
    $col3 = new GtkTreeViewColumn();

```

```

$idcol->set_title('ID-Nummer');
$col1->set_title('Kurz');
$col2->set_title('Lang');
$col3->set_title('Bemerkung');
$idcol->set_resizable(true);
$col1->set_resizable(true);
$col2->set_resizable(true);
$col3->set_resizable(true);
$cell_renderer1= new GtkCellRendererText();
$cell_renderer2= new GtkCellRendererText();
$cell_renderer3= new GtkCellRendererText();
$cell_renderer4= new GtkCellRendererText();
$idcol->pack_start($cell_renderer1, true);
$col1->pack_start($cell_renderer2, true);
$col2->pack_start($cell_renderer3, true);
$col3->pack_start($cell_renderer4, true);
$cell_renderer2->set_property('width', 50);
$cell_renderer2->set_property('editable', true);
$cell_renderer2->connect('edited', 'cell_edited', $model, 1);
$cell_renderer3->set_property('width', 100);
$cell_renderer3->set_property('editable', true);
$cell_renderer3->connect('edited', 'cell_edited', $model, 2);
$cell_renderer4->set_property('width', 100);
$cell_renderer4->set_property('editable', true);
$cell_renderer4->connect('edited', 'cell_edited', $model, 3);
$idcol->set_attributes($cell_renderer1, 'text', 0);
$col1->set_attributes($cell_renderer2, 'text', 1);
$col2->set_attributes($cell_renderer3, 'text', 2);
$col3->set_attributes($cell_renderer4, 'text', 3);
//$list->append_column($idcol);
$list->append_column($col1);
$list->append_column($col2);
$list->append_column($col3);
$list->set_model($model);
$scroll = new GtkScrolledWindow();
$scroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$scroll->add_with_viewport($list);
$scroll->show_all();
return $scroll;
}

function elementakzession($id) {
    global $vedom;
    $list = new GtkTreeStore(Gtk::TYPE_STRING);
    $bestinfo=get_tags($vedom, "Bestand_Info");
    $anz=$bestinfo->length;
    for ($a=0;$a<$anz;$a++) {
        $bestandinfo = $bestinfo->item($a);
        $kurz=''; $name=''; $sig='';
        if(hasChild($bestandinfo, 'Bestandsname')){

```

```

        $name=getChild($bestandinfo, 'Bestandsname');
        $name=$name->nodeValue;
    }
    if(hasChild($bestandinfo, 'Bestand_Kurz')){
        $kurz=getChild($bestandinfo, 'Bestand_Kurz');
        $kurz=$kurz->nodeValue;
    }
    if(hasChild($bestandinfo, 'Bestand_Sig')){
        $sig=getChild($bestandinfo, 'Bestand_Sig');
        $sig=$sig->nodeValue;
    }
    if($anz == 1){$iter=null;
    } else {
        if(!empty($kurz)){ $iter=$list->append(null, array(utf8_decode($kurz)));}
        elseif(!empty($sig)){ $iter=$list->append(null, array(utf8_decode($sig)));}
        elseif(!empty($name)){ $iter=$list->append(null, array(utf8_decode($name)));}
        else{ $iter=$list->append(null, array('*'));}
    }
    if(tag_exists($bestandinfo, "Akzession"){
        $abgs=get_tags($bestandinfo, "Akzession");
        $anzahl=$abgs->length;
        for ($b=0;$b<$anzahl;$b++){
            $abg = $abgs->item($b);
            $list->append($iter, array(utf8_decode($abg->nodeValue)));
        }
    } else{ $list->append($iter, array('*'));}
    if ($anz == 0){ $list->append(null, array('*'));}
}
if(empty($id)){ $text='';}
else { $element=$vedom->getElementById($id);
    $text=utf8_decode($element->nodeValue);}
if ($anz == 0){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('akzession');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}
if ($anz >= 1){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('akzession');
    $comboentry = GtkComboBoxEntry::new_text();
    $comboentry->set_model($list);
    $comboentry->get_child()->set_text($text);
    $hbox->pack_start($comboentry, true, true, 5);
    return $hbox;
}
}
}

```

```

function elementaltsig($id) {
    $saltmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {
        global $vedom;
        $selement=$vedom->getElementById($id);
        if (hasChild($selement, 'Altsig')==true) {
            $selements=$selement->childNodes;
            for($a=0;$a<$selements->length;$a++){
                $schild=$selements->item($a);
                if ($schild->nodeName == "Altsig"){
                    $sid=$schild->getAttribute("idstef");
                    $value=$schild->nodeValue;
                    $saltmodel->append(array($sid, utf8_decode($value)));
                }
            }
        }
    } else{$saltmodel->append(array('', ''));}
    $saltlist = new GtkTreeView;
    $idaltcol = new GtkTreeViewColumn();
    $saltcol = new GtkTreeViewColumn();
    $idaltcol->set_title('ID-Nummer');
    $saltcol->set_title('Altsignatur');
    $idaltcol->set_resizable(true);
    $saltcol->set_resizable(true);
    $saltcell_renderer1 = new GtkCellRendererText();
    $idaltcol->pack_start($saltcell_renderer1, true);
    $saltcol->pack_start($saltcell_renderer1, true);
    $saltcell_renderer1->set_property('width', 50);
    $saltcell_renderer1->set_property('editable', true);
    $saltcell_renderer1->connect('edited', 'cell_edited', $saltmodel, 1);
    $idaltcol->set_attributes($saltcell_renderer1, 'text', 0);
    $saltcol->set_attributes($saltcell_renderer1, 'text', 1);
    // $saltlist->append_column($idaltcol);
    $saltlist->append_column($saltcol);
    $saltlist->set_model($saltmodel);
    $saltscroll = new GtkScrolledWindow();
    $saltscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $saltscroll->add_with_viewport($saltlist);
    $saltscroll->show_all();
    return $saltscroll;
}

```

```

function elementaltuebform($id) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $satrs= new GtkListStore(Gtk::TYPE_STRING);
    $satrs->append(array(utf8_decode("Lesefilm")));
    $satrs->append(array(utf8_decode('Sicherungsfilm')));
    $satrs->append(array(utf8_decode('Microfiche')));
}

```

```

$attrs->append(array(utf8_decode('Datei')));
if(!empty($id)){
    global $vedom;
    $element=$vedom->getElementById($id);
    if (hasChild($element,'altübform')==true){
        $elements=$element->childNodes;
        for($a=0;$a<$elements->length;$a++){
            $child=$elements->item($a);
            if ($child->nodeName == "altübform"){
                $id=$child->getAttribute("idstef");
                $enthart=$child->getAttribute("altübform_Art");
                $art=$child->getAttribute("Art");
                if (empty($enthart)){ $attr=$art;}
                else { $attr=$enthart;}
                if (empty($attr)) { $attr="";}
                $value=$child->nodeValue;
                $enthmodel->append(array($id, utf8_decode($attr), utf8_decode($value)));
            }
        }
    }else{$enthmodel->append(array(' ', ' ', ' '));}
}else{$enthmodel->append(array(' ', ' ', ' '));}
$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrcol->set_title('Type');
$enthcol->set_title('Referenz');
$identhcol->set_resizable(true);
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable' , true);
// $enthcell_renderer2->set_property('has-entry' , false);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
// $bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrcol);

```

```

$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

```

```

function elementanlass($id) {
    $text='';
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

```

```

function elementanschrift($id) {
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Anschrift')==true) {
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++) {
                $schild=$elements->item($a);
                if ($schild->nodeName == "Anschrift") {
                    $id=$schild->getAttribute("idstef");
                    $value=$schild->nodeValue;
                    $bemmodel->append(array($id, utf8_decode($value)));
                }
            }
        } else{$bemmodel->append(array('', ''));}
    } else{$bemmodel->append(array('', ''));}
    $bemlist = new GtkTreeView;
    $idbemcol = new GtkTreeViewColumn();
    $bemcol = new GtkTreeViewColumn();
    $idbemcol->set_title('ID-Nummer');
    $bemcol->set_title('Anschriften');
    $idbemcol->set_resizable(true);
    $bemcol->set_resizable(true);
    $bemcell_renderer1 = new GtkCellRendererText();
    $bemcell_renderer2= new GtkCellRendererText();
    $idbemcol->pack_start($bemcell_renderer1, true);
    $bemcol->pack_start($bemcell_renderer2, true);
}

```

```

$bemcell_renderer2->set_property('width', 50);
$bemcell_renderer2->set_property('editable', true);
$bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
$idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
$bemcol->set_attributes($bemcell_renderer2, 'text', 1);
//$bemlist->append_column($idbemcol);
$bemlist->append_column($bemcol);
$bemlist->set_model($bemmodel);
$bemscroll = new GtkScrolledWindow();
$bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bemscroll->add_with_viewport($bemlist);
$bemscroll->show_all();
return $bemscroll;
}

function elementarchref($id) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING,
    Gtk::TYPE_STRING);
    $attrfkt= new GtkListStore(Gtk::TYPE_STRING);
    $attrfkt->append(array(utf8_decode("siehe auch")));
    $attrfkt->append(array(utf8_decode('Abschrift')));
    $attrfkt->append(array(utf8_decode('Original')));
    $attrfkt->append(array(utf8_decode('Entnommen aus')));
    $attrfkt->append(array(utf8_decode('Separiertes Material')));
    $attrart= new GtkListStore(Gtk::TYPE_STRING);
    $attrart->append(array(utf8_decode("Findmittel")));
    $attrart->append(array(utf8_decode('Bestand')));
    $attrart->append(array(utf8_decode('Akte')));
    $attrart->append(array(utf8_decode('Drucksache')));
    $attrart->append(array(utf8_decode("Urkunde")));
    $attrart->append(array(utf8_decode('Siegel')));
    $attrart->append(array(utf8_decode('Karte')));
    $attrart->append(array(utf8_decode('Plakat')));
    $attrart->append(array(utf8_decode("Foto")));
    $attrart->append(array(utf8_decode('Film')));
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'archref')==true) {
            $elements=$element->childNodes;
            for ($a=0; $a<$elements->length; $a++) {
                $child=$elements->item($a);
                if ($child->nodeName == "archref") {
                    $id=$child->getAttribute("idstef");
                    $enthfkt=$child->getAttribute("archref_Fkt");
                    $fkt=$child->getAttribute("Fkt");
                    if (empty($enthfkt)) {$attrf=$fkt;}
                    else {$attrf=$enthfkt;}
                    if (empty($attrf)) {$attrf="";}
                    $enthart=$child->getAttribute("archref_Art");

```

```

        $art=$schild->getAttribute("Art");
        if (empty($enthart)){ $attra=$art;}
        else { $attra=$enthart;}
        if (empty($attra)){ $attra="";}
        $value=$schild->nodeValue;
        $enthmodel->append(array($id, utf8_decode($attrf), utf8_decode($attra),
utf8_decode($value)));
    }
}
} else { $enthmodel->append(array('', '', '', '')); }
} else { $enthmodel->append(array('', '', '', '')); }
$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrfktcol = new GtkTreeViewColumn();
$enthattrartcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrfktcol->set_title('Type');
$enthattrartcol->set_title('Ref. auf');
$enthcol->set_title('Referenz');
$identhcol->set_resizable(true);
$enthattrfktcol->set_resizable(true);
$enthattrartcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$enthcell_renderer3 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrfktcol->pack_start($enthcell_renderer2, true);
$enthattrartcol->pack_start($enthcell_renderer3, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer2->set_property('model', $attrfkt);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable', true);
$enthcell_renderer3->set_property('width', 100);
$enthcell_renderer3->set_property('model', $attrart);
$enthcell_renderer3->set_property('text-column', 0);
$enthcell_renderer3->set_property('editable', true);
        $enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 3);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$enthcell_renderer3->connect('edited', 'cell_edited', $enthmodel, 2);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrfktcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthattrartcol->set_attributes($enthcell_renderer3, 'text', 2);
$enthcol->set_attributes($enthcell_renderer1, 'text', 3);
// $bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrfktcol);
$enthlist->append_column($enthattrartcol);

```



```

    $enthlist->append_column($enthcol);
    $enthlist->set_model($enthmodel);
    $enthscroll = new GtkScrolledWindow();
    $enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $enthscroll->add_with_viewport($enthlist);
    $enthscroll->show_all();
    return $enthscroll;
}

function elementauflage($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementausfuehrung($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementaz($id){
    $altmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Az')==true){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Az"){
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $altmodel->append(array($id, utf8_decode($value)));
                }
            }
        }
    }
}

```

```

    }
    } else{$saltmodel->append(array('', ''));}
} else{$saltmodel->append(array('', ''));}
$altlist = new GtkTreeView;
$idaltcol = new GtkTreeViewColumn();
$saltcol = new GtkTreeViewColumn();
$idaltcol->set_title('ID-Nummer');
$saltcol->set_title('Aktenzeichen');
$idaltcol->set_resizable(true);
$saltcol->set_resizable(true);
$saltcell_renderer1 = new GtkCellRendererText();
$idaltcol->pack_start($saltcell_renderer1, true);
$saltcol->pack_start($saltcell_renderer1, true);
$saltcell_renderer1->set_property('width', 50);
$saltcell_renderer1->set_property('editable', true);
$saltcell_renderer1->connect('edited', 'cell_edited', $saltmodel, 1);
$idaltcol->set_attributes($saltcell_renderer1, 'text', 0);
$saltcol->set_attributes($saltcell_renderer1, 'text', 1);
//$altlist->append_column($idaltcol);
$altlist->append_column($saltcol);
$altlist->set_model($saltmodel);
$altscroll = new GtkScrolledWindow();
$altscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$altscroll->add_with_viewport($altlist);
$altscroll->show_all();
return $altscroll;
}

function elementband($id) {
    $text='';
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementbearbeiter($id) {
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Bearbeiter')==true) {
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++) {
                $child=$elements->item($a);
            }
        }
    }
}

```

```

        if ($child->nodeName == "Bearbeiter"){
            $id=$child->getAttribute("idstef");
            $value=$child->nodeValue;
            $bemmodel->append(array($id, utf8_decode($value)));
        }
    }
} else{$bemmodel->append(array('', ''));}
} else{$bemmodel->append(array('', ''));}
$bemlist = new GtkTreeView;
$idbemcol = new GtkTreeViewColumn();
$bemcol = new GtkTreeViewColumn();
$idbemcol->set_title('ID-Nummer');
$bemcol->set_title('Bearbeiter');
$idbemcol->set_resizable(true);
$bemcol->set_resizable(true);
$bemcell_renderer1 = new GtkCellRendererText();
$bemcell_renderer2= new GtkCellRendererText();
$idbemcol->pack_start($bemcell_renderer1, true);
$bemcol->pack_start($bemcell_renderer2, true);
$bemcell_renderer2->set_property('width', 50);
$bemcell_renderer2->set_property('editable', true);
$bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
$idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
$bemcol->set_attributes($bemcell_renderer2, 'text', 1);
//$bemlist->append_column($idbemcol);
$bemlist->append_column($bemcol);
$bemlist->set_model($bemmodel);
$bemscroll = new GtkScrolledWindow();
$bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bemscroll->add_with_viewport($bemlist);
$bemscroll->show_all();
return $bemscroll;
}

function elementbem($id){
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Bem')==true){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Bem"){
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $bemmodel->append(array($id, utf8_decode($value)));
                }
            }
        }
    } else{$bemmodel->append(array('', ''));}
}

```

```

} else{$bemmodel->append(array('', ''));}
$bemlist = new GtkTreeView;
$idbemcol = new GtkTreeViewColumn();
$bemcol = new GtkTreeViewColumn();
$idbemcol->set_title('ID-Nummer');
$bemcol->set_title('Bemerkungen');
$idbemcol->set_resizable(true);
$bemcol->set_resizable(true);
$bemcell_renderer1 = new GtkCellRendererText();
$bemcell_renderer2= new GtkCellRendererText();
$idbemcol->pack_start($bemcell_renderer1, true);
$bemcol->pack_start($bemcell_renderer2, true);
$bemcell_renderer2->set_property('width', 50);
$bemcell_renderer2->set_property('editable', true);
$bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
$idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
$bemcol->set_attributes($bemcell_renderer2, 'text', 1);
//$bemlist->append_column($idbemcol);
$bemlist->append_column($bemcol);
$bemlist->set_model($bemmodel);
$bemscroll = new GtkScrolledWindow();
$bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bemscroll->add_with_viewport($bemlist);
$bemscroll->show_all();
return $bemscroll;
}

function elementbeschreibung($id) {
    $attr='';
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
        $besart=$child->getAttribute("Beschr_Art");
        $art=$child->getAttribute("Art");
        if (!empty($besart)){$attr=$besart;}
        elseif (!empty($art)){$attr=$art;}
        print "hoho $attr;";
    }
    $vbox= new GtkVBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $descbuffer->set_text($text);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);

```

```

$entryscroll->add_with_viewport($entry);
$entryscroll->show_all();
$comboentry = GtkComboBoxEntry::new_text();
$data = array('Regest', 'Kopfregist', 'Kurzregist', 'Vollregist', 'Urkundentext', 'Plakatte
xt', 'Begleittext', '');
foreach ($data as $str) {$comboentry->append_text($str);}
$comboentry->get_child()->set_text($attr);
$vbox->pack_start($comboentry, true, true, 5);
$vbox->pack_start($entryscroll, true, true, 5);
return $vbox;
}

```

```

function elementbestandabgstelle($id){
    global $vedom;
    $vorprovmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)){
        $element=$vedom->getElementById($id);
        if (hasChild($element,'Abg_Stelle')==true){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Abg_Stelle"){
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $vorprovmodel->append(array($id, utf8_decode($value)));
                }
            }
        } else{$vorprovmodel->append(array('', ''));}
    } else{$vorprovmodel->append(array('', ''));}
    $vorprovlist = new GtkTreeView;
    $idvorprovcol = new GtkTreeViewColumn();
    $vorprovcol = new GtkTreeViewColumn();
    $idvorprovcol->set_title('ID-Nummer');
    $vorprovcol->set_title('Abgebende Stellen');
    $idvorprovcol->set_resizable(true);
    $vorprovcol->set_resizable(true);
    $vorprovcell_renderer1 = new GtkCellRendererText();
    $vorprovcell_renderer2 = new GtkCellRendererText();
    $idvorprovcol->pack_start($vorprovcell_renderer1, true);
    $vorprovcol->pack_start($vorprovcell_renderer2, true);
    $vorprovcell_renderer1->set_property('width', 50);
    $vorprovcell_renderer2->set_property('editable', true);
    $vorprovcell_renderer2->set_property('width', 50);
    $vorprovcell_renderer2->connect('edited', 'cell_edited', $vorprovmodel, 1);
    $idvorprovcol->set_attributes($vorprovcell_renderer1, 'text', 0);
    $vorprovcol->set_attributes($vorprovcell_renderer2, 'text', 1);
    // $vorprovlist->append_column($idvorprovcol);
    $vorprovlist->append_column($vorprovcol);
}

```

```

$vorprovlist->set_model($vorprovmodel);
$vorprovscroll = new GtkScrolledWindow();
$vorprovscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$vorprovscroll->add_with_viewport($vorprovlist);
$vorprovscroll->show_all();
return $vorprovscroll;
}

function elementbestandakzession($id){
    global $vedom;
    $vorprovmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)){
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Akzession')==true){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Akzession"){
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $vorprovmodel->append(array($id, utf8_decode($value)));
                }
            }
        } else{$vorprovmodel->append(array('', ''));}
    } else{$vorprovmodel->append(array('', ''));}
    $vorprovlist = new GtkTreeView;
    $idvorprovcol = new GtkTreeViewColumn();
    $vorprovcol = new GtkTreeViewColumn();
    $idvorprovcol->set_title('ID-Nummer');
    $vorprovcol->set_title('Akzessionen');
    $idvorprovcol->set_resizable(true);
    $vorprovcol->set_resizable(true);
    $vorprovcell_renderer1 = new GtkCellRendererText();
    $vorprovcell_renderer2 = new GtkCellRendererText();
    $idvorprovcol->pack_start($vorprovcell_renderer1, true);
    $vorprovcol->pack_start($vorprovcell_renderer2, true);
    $vorprovcell_renderer1->set_property('width', 50);
    $vorprovcell_renderer2->set_property('editable', true);
    $vorprovcell_renderer2->set_property('width', 50);
    $vorprovcell_renderer2->connect('edited', 'cell_edited', $vorprovmodel, 1);
    $idvorprovcol->set_attributes($vorprovcell_renderer1, 'text', 0);
    $vorprovcol->set_attributes($vorprovcell_renderer2, 'text', 1);
    // $vorprovlist->append_column($idvorprovcol);
    $vorprovlist->append_column($vorprovcol);
    $vorprovlist->set_model($vorprovmodel);
    $vorprovscroll = new GtkScrolledWindow();
    $vorprovscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $vorprovscroll->add_with_viewport($vorprovlist);
    $vorprovscroll->show_all();
    return $vorprovscroll;
}

```

```

}

function elementbestandkurz ($id) {
    global $vedom;
    $list = new GtkTreeStore (Gtk::TYPE_STRING);
    $bestinfo=get_tags ($vedom, "Bestand_Info");
    $anz=$bestinfo->length;
    for ($a=0;$a<$anz;$a++){
        $bestandinfo = $bestinfo->item($a);
        if (hasChild($bestandinfo, "Bestand_Kurz")){
            $bestandkurzallg=getChild($bestandinfo, "Bestand_Kurz");
            $list->append(null, array(utf8_decode($bestandkurzallg->nodeValue)));
        } else {$list->append(null, array('*'));}
    }
    if(empty($id)){$text='';}
    else {$element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    if ($anz == 0){
        $hbox=new GtkHbox(false,0);
        $hbox->set_name('bestandkurz');
        $entry=new GtkEntry();
        $entry->set_text($text);
        $hbox->pack_start($entry, true, true, 5);
        return $hbox;
    }
    if ($anz >= 1){
        $hbox=new GtkHbox(false,0);
        $hbox->set_name('bestandkurz');
        $comboentry = GtkComboBoxEntry::new_text();
        $comboentry->set_model($list);
        $comboentry->get_child()->set_text($text);
        $hbox->pack_start($comboentry, true, true, 5);
        return $hbox;
    }
}

function elementbestandprovenienz ($id) {
    global $vedom;
    $vorprovmodel = new GtkListStore (Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Provenienz')==true) {
            $elements=$element->childNodes;
            for ($a=0;$a<$elements->length;$a++) {
                $child=$elements->item($a);
                if ($child->nodeName == "Provenienz"){
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $vorprovmodel->append(array($id, utf8_decode($value)));
                }
            }
        }
    }
}

```

```

    }
    } else{$vorprovmodel->append(array('', ''));}
} else{$vorprovmodel->append(array('', ''));}
$vorprovlist = new GtkTreeView;
$idvorprovcol = new GtkTreeViewColumn();
$vorprovcol = new GtkTreeViewColumn();
$idvorprovcol->set_title('ID-Nummer');
$vorprovcol->set_title('Provenienzen');
$idvorprovcol->set_resizable(true);
$vorprovcol->set_resizable(true);
$vorprovcell_renderer1 = new GtkCellRendererText();
$vorprovcell_renderer2 = new GtkCellRendererText();
$idvorprovcol->pack_start($vorprovcell_renderer1, true);
$vorprovcol->pack_start($vorprovcell_renderer2, true);
$vorprovcell_renderer1->set_property('width', 50);
$vorprovcell_renderer2->set_property('editable', true);
$vorprovcell_renderer2->set_property('width', 50);
$vorprovcell_renderer2->connect('edited', 'cell_edited', $vorprovmodel, 1);
$idvorprovcol->set_attributes($vorprovcell_renderer1, 'text', 0);
$vorprovcol->set_attributes($vorprovcell_renderer2, 'text', 1);
//$vorprovlist->append_column($idvorprovcol);
$vorprovlist->append_column($vorprovcol);
$vorprovlist->set_model($vorprovmodel);
$vorprovscroll = new GtkScrolledWindow();
$vorprovscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$vorprovscroll->add_with_viewport($vorprovlist);
$vorprovscroll->show_all();
return $vorprovscroll;
}

function elementbestandsig($id) {
    $text='';
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('signaturpc');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementbestandsname($id) {
    $text='';
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
}

```



```

}
$ hbox=new GtkHbox(false,0);
$ entry=new GtkEntry();
$ entry->set_text($text);
$ hbox->pack_start($entry, true, true, 5);
return $hbox;
}

function elementbestellsig($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $ hbox=new GtkHbox(false,0);
    $ entry=new GtkEntry();
    $ entry->set_text($text);
    $ hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementbibliographie($id){
    $note= new GtkNotebook();
    $attr='';
    $bibrefids='';
    global $vedom;
    if (!empty($id)){
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Bibliographie')){
            $elements=$element->childNodes;
            for ($a=0; $a<$elements->length; $a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Bibliographie"){
                    $ids=$child->getAttribute("idstef");
                    $titart=$child->getAttribute("Publ_Art");
                    $art=$child->getAttribute("Art");
                    if (empty($titart)){$attr=$art;}
                    else {$attr=$titart;}
                    if (hasChild($child, 'bibref')==true){$bibrefids=$ids;}
                    $bib=elementbibref($bibrefids);
                    $vbox= new GtkVBox(false, 0);
                    $comboentry = GtkComboBoxEntry::new_text();
                    $data = array('Literatur', 'Amtsdrucksachen', 'Editionen', '');
                    foreach ($data as $str) {$comboentry->append_text($str);}
                    $comboentry->get_child()->set_text($attr);
                    if (empty($attr)){$attr="Bibliographie";}
                    $vbox->pack_start($comboentry, false, false, 5);
                    $vbox->pack_start($bib, true, true, 0);
                    $note->append_page($vbox, new GtkLabel($attr));
                }
            }
        }
    }
}

```

```

    }
}
} else{$vbox= new GtkVBox(false, 0);
    $bib=elementbibref($bibrefids);
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Literatur', 'Amtsdrucksachen', 'Editionen', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $comboentry->get_child()->set_text($attr);
    $vbox->pack_start($comboentry, false, false, 5);
    $vbox->pack_start($bib, true, true, 0);
    $note->append_page($vbox, new GtkLabel('Bibliographie'));
}
return $note;
}

function elementbibref($id) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attr = new GtkListStore(Gtk::TYPE_STRING);
    $attr->append(array(utf8_decode("Literatur")));
    $attr->append(array(utf8_decode('Amtsdrucksache')));
    $attr->append(array(utf8_decode('Edition')));
    $attr->append(array(utf8_decode('')));
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'bibref')==true) {
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "bibref"){
                    $id=$child->getAttribute("idstef");
                    $enthart=$child->getAttribute("bibref_Art");
                    $art=$child->getAttribute("Art");
                    if (empty($enthart)){$attr=$art;}
                    else {$attr=$enthart;}
                    if (empty($attr)){$attr="";}
                    $value=$child->nodeValue;
                    $enthmodel->append(array($id, utf8_decode($attr), utf8_decode($value)));
                }
            }
        }
        } else{$enthmodel->append(array('', '', ''));}
    } else{$enthmodel->append(array('', '', ''));}
    $enthlist = new GtkTreeView;
    $identhcol = new GtkTreeViewColumn();
    $enthattrcol = new GtkTreeViewColumn();
    $enthcol = new GtkTreeViewColumn();
    $identhcol->set_title('ID-Nummer');
    $enthattrcol->set_title('Type');
    $enthcol->set_title('Referenz');
    $identhcol->set_resizable(true);

```

```

$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable' , true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
//$bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function elementblatt($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementeigentuemer($id){
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkListStore(Gtk::TYPE_STRING);
    $attrs->append(array("Person"));
    $attrs->append(array('Institution'));
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element,'Person')==true or hasChild($element,'Institution')==true){

```

```

$elements=$element->childNodes;
for($a=0;$a<$elements->length;$a++){
    $child=$elements->item($a);
    $id=$child->getAttribute("idstef");
    $value=$child->nodeValue;
    if ($child->nodeName == "Person"){ $attr='Person';}
    if ($child->nodeName == "Institution"){ $attr='Institution';}
    $enthmodel->append(array($id, utf8_decode($attr), utf8_decode($value)));
}
} else{$enthmodel->append(array('', '', ''));}
} else{$enthmodel->append(array('', '', ''));}
$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrcol->set_title('Type');
$enthcol->set_title('Name');
$identhcol->set_resizable(true);
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable', true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
// $bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function elementzeichnung($id) {
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {

```

```

global $vedom;
$element=$vedom->getElementById($id);
if (hasChild($element,'Einzeichnung')==true){
    $elements=$element->childNodes;
    for($a=0;$a<$elements->length;$a++){
        $child=$elements->item($a);
        if ($child->nodeName == "Einzeichnung"){
            $id=$child->getAttribute("idstef");
            $value=$child->nodeValue;
            $bemmodel->append(array($id, utf8_decode($value)));
        }
    }
} else{$bemmodel->append(array('',''));}
} else{$bemmodel->append(array('',''));}
$bemlist = new GtkTreeView;
$idbemcol = new GtkTreeViewColumn();
$bemcol = new GtkTreeViewColumn();
$idbemcol->set_title('ID-Nummer');
$bemcol->set_title('Einzeichnung');
$idbemcol->set_resizable(true);
$bemcol->set_resizable(true);
$bemcell_renderer1 = new GtkCellRendererText();
$bemcell_renderer2= new GtkCellRendererText();
$idbemcol->pack_start($bemcell_renderer1, true);
$bemcol->pack_start($bemcell_renderer2, true);
$bemcell_renderer2->set_property('width', 50);
$bemcell_renderer2->set_property('editable', true);
$bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
$idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
$bemcol->set_attributes($bemcell_renderer2, 'text', 1);
//$bemlist->append_column($idbemcol);
$bemlist->append_column($bemcol);
$bemlist->set_model($bemmodel);
$bemscroll = new GtkScrolledWindow();
$bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bemscroll->add_with_viewport($bemlist);
$bemscroll->show_all();
return $bemscroll;
}

function elemententhaelt($id){
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkListStore(Gtk::TYPE_STRING);
    $attrs->append(array("Enthält"));
    $attrs->append(array('Enth. auch'));
    $attrs->append(array('Enth. nur'));
    $attrs->append(array('Enth. ua'));
    $attrs->append(array('Enth. va'));
    $attrs->append(array('Darin'));
    $attrs->append(array('Hierin'));
}

```



```

    $enthlist->set_model($enthmodel);
    $enthscroll = new GtkScrolledWindow();
    $enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $enthscroll->add_with_viewport($enthlist);
    $enthscroll->show_all();
    return $enthscroll;
}

```

```

function elemententstehungsstufe($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

```

```

function elementfarbe($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

```

```

function elementfmname($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

```

```

function elementfmref($id) {
    $saltmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'FM_ref')==true) {
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "FM_ref"){
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $saltmodel->append(array($id, utf8_decode($value)));
                }
            }
        } else{$saltmodel->append(array('', ''));}
    } else{$saltmodel->append(array('', ''));}
    $saltlist = new GtkTreeView;
    $idaltcol = new GtkTreeViewColumn();
    $saltcol = new GtkTreeViewColumn();
    $idaltcol->set_title('ID-Nummer');
    $saltcol->set_title('Referenz');
    $idaltcol->set_resizable(true);
    $saltcol->set_resizable(true);
    $saltcell_renderer1 = new GtkCellRendererText();
    $idaltcol->pack_start($saltcell_renderer1, true);
    $saltcol->pack_start($saltcell_renderer1, true);
    $saltcell_renderer1->set_property('width', 50);
    $saltcell_renderer1->set_property('editable', true);
    $saltcell_renderer1->connect('edited', 'cell_edited', $saltmodel, 1);
    $idaltcol->set_attributes($saltcell_renderer1, 'text', 0);
    $saltcol->set_attributes($saltcell_renderer1, 'text', 1);
    // $saltlist->append_column($idaltcol);
    $saltlist->append_column($saltcol);
    $saltlist->set_model($saltmodel);
    $saltscroll = new GtkScrolledWindow();
    $saltscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $saltscroll->add_with_viewport($saltlist);
    $saltscroll->show_all();
    return $saltscroll;
}

function elementfmsig($id) {
    $text='';
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false, 0);
}

```



```

    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementformat($id){
    $attr='';
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $attr=$element->getAttribute("M_Mass");
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $entry->set_size_request(45 ,25);
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('mm', 'cm', 'm');
    foreach ($data as $string) {
        $comboentry->append_text($string);
    }
    $comboentry->get_child()->set_text($attr);
    $comboentry->set_size_request(85 ,25);
    $hbox->pack_start($entry, true, true, 5);
    $hbox->pack_start($comboentry, false, false, 5);
    return $hbox;
}

function elementgemarkung($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementhilfsfeld($id){
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Hilfsfeld')==true){

```

```

    $elements=$element->childNodes;
    for($a=0;$a<$elements->length;$a++){
        $child=$elements->item($a);
        if ($child->nodeName == "Hilfsfeld"){
            $id=$child->getAttribute("idstef");
            $value=$child->nodeValue;
            $bemmodel->append(array($id, utf8_decode($value)));
        }
    }
} else{$bemmodel->append(array('', ''));}
} else{$bemmodel->append(array('', ''));}
$bemlist = new GtkTreeView;
$idbemcol = new GtkTreeViewColumn();
$bemcol = new GtkTreeViewColumn();
$idbemcol->set_title('ID-Nummer');
$bemcol->set_title('Ergänzung');
$idbemcol->set_resizable(true);
$bemcol->set_resizable(true);
$bemcell_renderer1 = new GtkCellRendererText();
$bemcell_renderer2= new GtkCellRendererText();
$idbemcol->pack_start($bemcell_renderer1, true);
$bemcol->pack_start($bemcell_renderer2, true);
$bemcell_renderer2->set_property('width', 50);
$bemcell_renderer2->set_property('editable', true);
$bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
$idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
$bemcol->set_attributes($bemcell_renderer2, 'text', 1);
//$bemlist->append_column($idbemcol);
$bemlist->append_column($bemcol);
$bemlist->set_model($bemmodel);
$bemscroll = new GtkScrolledWindow();
$bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$bemscroll->add_with_viewport($bemlist);
$bemscroll->show_all();
return $bemscroll;
}

function elementinstitution($id){
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrfkt= new GtkTreeStore(Gtk::TYPE_STRING);
    $iter=$attrfkt->append(null, array("Urkunde"));
    $attrfkt->append($iter, array("Aussteller"));
    $attrfkt->append($iter, array('Empfaenger'));
    $attrfkt->append($iter, array('Siegler'));
    $attrfkt->append($iter, array('Zeuge'));
    $iter=$attrfkt->append(null, array("Recht"));
    $attrfkt->append($iter, array('Zeuge'));
    $attrfkt->append($iter, array('Klaeger'));
    $attrfkt->append($iter, array("Beklagter"));
}

```

```

$attrfkt->append($iter, array('Zeuge'));
$iter=$attrfkt->append(null, array("Film"));
$attrfkt->append($iter, array('Auftragnehmer'));
$attrfkt->append($iter, array('Herausgeber'));
$attrfkt->append($iter, array('Druckerei'));
$attrfkt->append($iter, array('Rechteinhaber'));
$attrfkt->append($iter, array("Auftraggeber"));
$attrfkt->append($iter, array('Rechteinhaber'));
$attrfkt->append($iter, array('Produzent'));
$attrfkt->append($iter, array('Musik'));
$attrfkt->append($iter, array('Trick'));
$attrfkt->append(null, array(''));
$attrart= new GtkListStore(Gtk::TYPE_STRING);
$attrart->append(array("Behörde"));
$attrart->append(array('Betrieb'));
$attrart->append(array('Unternehmen'));
$attrart->append(array(''));
if(!empty($id)){
    global $vedom;
    $element=$vedom->getElementById($id);
    if (hasChild($element, 'Institution')==true){
        $elements=$element->childNodes;
        for($a=0;$a<$elements->length;$a++){
            $child=$elements->item($a);
            if ($child->nodeName == "Institution"){
                $id=$child->getAttribute("idstef");
                $enthfkt=$child->getAttribute("Inst_Fkt");
                $fkt=$child->getAttribute("Fkt");
                if (empty($enthfkt)){ $attrf=$fkt;}
                else { $attrf=$enthfkt;}
                if (empty($attrf)){ $attrf="";}
                $enthart=$child->getAttribute("Inst_Art");
                $art=$child->getAttribute("Art");
                if (empty($enthart)){ $attra=$art;}
                else { $attra=$enthart;}
                if (empty($attra)){ $attra="";}
                $value=$child->nodeValue;
                $enthmodel->append(array($id, utf8_decode($attrf), utf8_decode($attra), utf8_dec
ode($value)));
            }
        }
    }
    }else{$enthmodel->append(array(' ', ' ', ' ', ' '));}
}else{$enthmodel->append(array(' ', ' ', ' ', ' '));}
$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrfktcol = new GtkTreeViewColumn();
$enthattrartcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrfktcol->set_title('Funktion');

```

```

$enthattrartcol->set_title('Art');
$enthcol->set_title('Name');
$identhcol->set_resizable(true);
$enthattrfktcol->set_resizable(true);
$enthattrartcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$enthcell_renderer3 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrfktcol->pack_start($enthcell_renderer2, true);
$enthattrartcol->pack_start($enthcell_renderer3, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer2->set_property('model', $attrfkt);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable', true);
$enthcell_renderer3->set_property('width', 100);
$enthcell_renderer3->set_property('model', $attrart);
$enthcell_renderer3->set_property('text-column', 0);
$enthcell_renderer3->set_property('editable', true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 3);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$enthcell_renderer3->connect('edited', 'cell_edited', $enthmodel, 2);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrfktcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthattrartcol->set_attributes($enthcell_renderer3, 'text', 2);
$enthcol->set_attributes($enthcell_renderer1, 'text', 3);
// $bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrfktcol);
$enthlist->append_column($enthattrartcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function elementkartentyp($id) {
    $text='';
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();

```

```

    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementklassnr($id){
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    } else {$text='';}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementklasstitel($id){
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    } else {$text='';}
    $hbox= new GtkHBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $descbuffer->set_text($text);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $hbox->pack_start($entryscroll, true, true, 5);
    return $hbox;
}

function elementkonkordanz($id){
    global $vedom;
    $root=$vedom->documentElement;
    $konmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)){
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Kon_Eintrag')==true){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Kon_Eintrag"){

```

```

    $sig=""; $bestsig=""; $altsig=""; $lager="";
    $id=$child->getAttribute("idstef");
    if (haschild($child, "Signatur")== true){
        $sigs=getchild($child, "Signatur");
        $sig=$sigs->nodeValue;
    }
    if (haschild($child, "Bestellsig")== true){
        $bestsigs=getchild($child, "Bestellsig");
        $bestsig=$bestsigs->nodeValue;
    }
    if (haschild($child, "Altsig")== true){
        $altsigs=getchild($child, "Altsig");
        $altsig=$altsigs->nodeValue;
    }
    if (haschild($child, "Lagerung")== true){
        $lagers=getchild($child, "Lagerung");
        $lager=$lagers->nodeValue;
    }
    $konmodel->append(array($id, $sig, $bestsig, $altsig, utf8_decode($lager)));
}
}
} else{$konmodel->append(array(' ', ' ', ' ', ' ', ' '));}
} else{$konmodel->append(array(' ', ' ', ' ', ' ', ' '));}
$konlist = new GtkTreeView;
$idcol = new GtkTreeViewColumn();
$sigcol = new GtkTreeViewColumn();
$bestsigcol = new GtkTreeViewColumn();
$altsigcol = new GtkTreeViewColumn();
$lagercol = new GtkTreeViewColumn();
$idcol->set_title('ID-Nummer');
$sigcol->set_title('Signatur');
$bestsigcol->set_title('Bestellnummer');
$altsigcol->set_title('Altsignatur');
$lagercol->set_title('Lagerung');
$idcol->set_resizable(true);
$sigcol->set_resizable(true);
$bestsigcol->set_resizable(true);
$altsigcol->set_resizable(true);
$lagercol->set_resizable(true);
$cell_renderer1 = new GtkCellRendererText();
$idcol->pack_start($cell_renderer1, true);
$sigcol->pack_start($cell_renderer1, true);
$bestsigcol->pack_start($cell_renderer1, true);
$altsigcol->pack_start($cell_renderer1, true);
$lagercol->pack_start($cell_renderer1, true);
$cell_renderer1->set_property('width', 50);
$idcol->set_attributes($cell_renderer1, 'text', 0);
$sigcol->set_attributes($cell_renderer1, 'text', 1);
$bestsigcol->set_attributes($cell_renderer1, 'text', 2);

```

```

$altsigcol->set_attributes($cell_renderer1, 'text', 3);
$lagercol->set_attributes($cell_renderer1, 'text', 4);
$konlist->append_column($sigcol);
$konlist->append_column($bestsigcol);
$konlist->append_column($altsigcol);
$konlist->append_column($lagercol);
$konlist->set_model($konmodel);
$konscroll = new GtkScrolledWindow();
$konscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$konscroll->add_with_viewport($konlist);
$konscroll->show_all();
$menu = new GtkMenu();
$echol = new GtkImageMenuItem(GTK::STOCK_REFRESH);
$echol->connect_simple('activate', 'cleartree', $konmodel);
$echol->connect_simple('activate', 'new_konkord', $root, $konmodel);
$menu->append($echol);
$menu->show_all();
$konlist->set_events($konlist->get_events() | Gdk::BUTTON_PRESS_MASK);
$konlist->connect('button-press-event', 'doPopup', $menu);
return $konscroll;
}

```

```

function elementlagerung($id){
    global $vedom;
    $list = new GtkTreeStore(Gtk::TYPE_STRING);
    $bestinfo=get_tags($vedom, "Bestand_Info");
    $anz=$bestinfo->length;
    for ($a=0;$a<$anz;$a++){
        $bestandinfo = $bestinfo->item($a);
        $kurz=''; $name=''; $sig='';
        if(hasChild($bestandinfo, 'Bestandsname')){
            $name=getChild($bestandinfo, 'Bestandsname');
            $name=$name->nodeValue;
        }
        if(hasChild($bestandinfo, 'Bestand_Kurz')){
            $kurz=getChild($bestandinfo, 'Bestand_Kurz');
            $kurz=$kurz->nodeValue;
        }
        if(hasChild($bestandinfo, 'Bestand_Sig')){
            $sig=getChild($bestandinfo, 'Bestand_Sig');
            $sig=$sig->nodeValue;
        }
        if($anz == 1){$iter=null;}
        else {
            if(!empty($kurz)){
                $iter=$list->append(null, array(utf8_decode($kurz)));
            }
            elseif(!empty($sig)){
                $iter=$list->append(null, array(utf8_decode($sig)));
            }
        }
    }
}

```

```

        elseif(!empty($name)){
            $iter=$list->append(null, array(utf8_decode($name)));
        }
        else{$iter=$list->append(null, array('*'));}
    }
    if(hasChild($bestandinfo, "Lagerung"){
        $lag=getChild($bestandinfo, "Lagerung");
        $list->append($iter, array(utf8_decode($lag->nodeValue)));
    } else{$list->append($iter, array('*'));}
}
if(empty($id)){$text='';}
else {$selement=$vedom->getElementById($id);
    $text=utf8_decode($selement->nodeValue);}
if ($anz == 0){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('lagerung');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}
if ($anz >= 1){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('lagerung');
    $comboentry = GtkComboBoxEntry::new_text();
    $comboentry->set_model($list);
    $comboentry->get_child()->set_text($text);
    $hbox->pack_start($comboentry, true, true, 5);
    return $hbox;
}
}
}

```

```

function elementlaufzeit($id){
    $idanfang='';
    $idende='';
    $idtext='';
    if(!empty($id)){
        global $vedom;
        $selement=$vedom->getElementById($id);
        if (hasChild($selement, 'LZ_Anfang')==true){
            $schild=getChild($selement, 'LZ_Anfang');$idanfang=$schild->getAttribute('idstef');}
        if (hasChild($selement, 'LZ_Ende')==true){
            $schild=getChild($selement, 'LZ_Ende');$idende=$schild->getAttribute('idstef');}
        if (hasChild($selement, 'LZ_Text')==true){
            $schild=getChild($selement, 'LZ_Text');$idtext=$schild->getAttribute('idstef');}
    }
    $vboxlz=new GtkVbox(false,0);
    $vboxlz->set_name('laufzeitlz');
}

```



```

$anfang=elementlzanfang($idanfang);
$sende=elementlzende($idende);
$text=elementlztext($idtext);
$vboxlz->pack_start($anfang, false, false, 5);
$vboxlz->pack_start($sende, false, false, 5);
$vboxlz->pack_start($text, false, false, 5);
return $vboxlz;
}

```

```

function elementlzanfang($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 5);
    $hbox->set_name('lzanfang');
    $label=new GtkLabel('von');
    $label->set_size_request(50,25);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $entry->set_size_request(80,25);
    $hbox->pack_start($label, false, false, 0);
    $hbox->pack_start($entry, false, false, 0);
    return $hbox;
}

```

```

function elementlzende($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 5);
    $hbox->set_name('lzende');
    $label=new GtkLabel('bis');
    $label->set_size_request(50,25);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $entry->set_size_request(80,25);
    $hbox->pack_start($label, false, false, 0);
    $hbox->pack_start($entry, false, false, 0);
    return $hbox;
}

```

```

function elementlztext($id){
    $text='';
    if(!empty($id)){
        global $vedom;

```

```

        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 5);
    $label=new GtkLabel('Text');
    $label->set_size_request(50,25);
    $hbox->set_name('lztext');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($label, false, false, 0);
    $hbox->pack_start($entry, false, false, 0);
    return $hbox;
}

function elementmassstab($id){
    $attr='';
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $attr=$element->getAttribute("Massstab_Art");
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $entry->set_size_request(60,25);
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Original', 'Umrechnung');
    foreach ($data as $string) {
        $comboentry->append_text($string);
    }
    $comboentry->get_child()->set_text($attr);
    $comboentry->set_size_request(85,25);
    $hbox->pack_start($entry, true, true, 5);
    $hbox->pack_start($comboentry, false, false, 5);
    return $hbox;
}

function elementmaterial($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

```

```

}

function elementnebenkarten($id) {
    $bemmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Nebenkarten')==true) {
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++) {
                $child=$elements->item($a);
                if ($child->nodeName == "Nebenkarten") {
                    $id=$child->getAttribute("idstef");
                    $value=$child->nodeValue;
                    $bemmodel->append(array($id, utf8_decode($value)));
                }
            }
        } else{$bemmodel->append(array('', ''));}
    } else{$bemmodel->append(array('', ''));}
    $bemlist = new GtkTreeView;
    $idbemcol = new GtkTreeViewColumn();
    $bemcol = new GtkTreeViewColumn();
    $idbemcol->set_title('ID-Nummer');
    $bemcol->set_title('Nebenkarten');
    $idbemcol->set_resizable(true);
    $bemcol->set_resizable(true);
    $bemcell_renderer1 = new GtkCellRendererText();
    $bemcell_renderer2= new GtkCellRendererText();
    $idbemcol->pack_start($bemcell_renderer1, true);
    $bemcol->pack_start($bemcell_renderer2, true);
    $bemcell_renderer2->set_property('width', 50);
    $bemcell_renderer2->set_property('editable', true);
    $bemcell_renderer2->connect('edited', 'cell_edited', $bemmodel, 1);
    $idbemcol->set_attributes($bemcell_renderer1, 'text', 0);
    $bemcol->set_attributes($bemcell_renderer2, 'text', 1);
    // $bemlist->append_column($idbemcol);
    $bemlist->append_column($bemcol);
    $bemlist->set_model($bemmodel);
    $bemscroll = new GtkScrolledWindow();
    $bemscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $bemscroll->add_with_viewport($bemlist);
    $bemscroll->show_all();
    return $bemscroll;
}

function elementort($id) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkListStore(Gtk::TYPE_STRING);
    $attrs->append(array("Geburtsort"));
    $attrs->append(array('Sterbeort'));
    $attrs->append(array('Wohnort'));
}

```

```

$attrs->append(array('Dienstort'));
$attrs->append(array('Lagerung'));
$attrs->append(array('Region'));
$attrs->append(array('Land'));
$attrs->append(array('Druckort'));
$attrs->append(array('Gemeinde'));
$attrs->append(array('Gemarkung'));
$attrs->append(array('Ausstellungsort'));
if(!empty($id)){
    global $vedom;
    $element=$vedom->getElementById($id);
    if (hasChild($element, 'Ort')==true){
        $elements=$element->childNodes;
        for($a=0;$a<$elements->length;$a++){
            $child=$elements->item($a);
            if ($child->nodeName == "Ort"){
                $id=$child->getAttribute("idstef");
                $enthart=$child->getAttribute("Ort_Fkt");
                $art=$child->getAttribute("Fkt");
                if (empty($enthart)){$attr=$art;}
                else {$attr=$enthart;}
                if (empty($attr)){$attr="";}
                $value=$child->nodeValue;
                $enthmodel->append(array($id, utf8_decode($attr), utf8_decode($value)));
            }
        }
    }
}
}
}

}else{$enthmodel->append(array('', '', ''));}

$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrcol->set_title('Type');
$enthcol->set_title('Name');
$identhcol->set_resizable(true);
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 120);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable', true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);

```

```

$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
//$bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function elementperson($id) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkTreeStore(Gtk::TYPE_STRING);
    $iter=$attrs->append(null, array("Urkunde"));
    $attrs->append($iter, array("Aussteller"));
    $attrs->append($iter, array('Empfaenger'));
    $attrs->append($iter, array('Sieglер'));
    $attrs->append($iter, array('Zeuge'));
    $iter=$attrs->append(null, array("Recht"));
    $attrs->append($iter, array('Zeuge'));
    $attrs->append($iter, array('Klaeger'));
    $attrs->append($iter, array('Beklagter'));
    $attrs->append($iter, array('Prokurator_Kl'));
    $attrs->append($iter, array('Prokurator_Bekl'));
    $attrs->append($iter, array('Urheber'));
    $attrs->append($iter, array('Auftraggeber'));
    $attrs->append($iter, array('Rechteinhaber'));
    $iter=$attrs->append(null, array("Film"));
    $attrs->append($iter, array('Produzent'));
    $attrs->append($iter, array('Regie'));
    $attrs->append($iter, array('Kamera'));
    $attrs->append($iter, array('Drehbuch'));
    $attrs->append($iter, array('Vorlage'));
    $attrs->append($iter, array('Schnitt'));
    $attrs->append($iter, array('Darsteller'));
    $attrs->append($iter, array('Sprecher'));
    $attrs->append($iter, array('Musik'));
    $attrs->append($iter, array('Trick'));
    $iter=$attrs->append(null, array("Sonstiges"));
    $attrs->append($iter, array('Patient'));
    $attrs->append($iter, array('Techniker'));
    $attrs->append($iter, array('Herausgeber'));
    $iter=$attrs->append(null, array(""));
    if(!empty($id)) {
        global $vedom;

```

```

$element=$vedom->getElementById($id);
if (hasChild($element, 'Person')==true) {
    $elements=$element->childNodes;
    for ($a=0; $a<$elements->length; $a++) {
        $child=$elements->item($a);
        if ($child->nodeName == "Person"){
            $id=$child->getAttribute("idstef");
            $enthart=$child->getAttribute("Pers_Fkt");
            $art=$child->getAttribute("Fkt");
            if (empty($enthart)) {$attr=$art;}
            else {$attr=$enthart;}
            if (empty($attr)) {$attr="";}
            $value=$child->nodeValue;
            $enthmodel->append(array($id, utf8_decode($attr), utf8_decode($value)));
        }
    }
}
} else {$enthmodel->append(array('', '', ''));}
$enthlist = new GtkTreeView();
$identhcol = new GtkTreeViewColumn();
$enthattrcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrcol->set_title('Type');
$enthcol->set_title('Name');
$identhcol->set_resizable(true);
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 150);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable' , true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
//$bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);

```

```

$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

function elementprovenienz($id) {
    global $vedom;
    $list = new GtkTreeStore(Gtk::TYPE_STRING);
    $bestinfo=get_tags($vedom, "Bestand_Info");
    $anz=$bestinfo->length;
    for ($a=0;$a<$anz;$a++){
        $bestandinfo = $bestinfo->item($a);
        $kurz=''; $name=''; $sig='';
        if(hasChild($bestandinfo, 'Bestandsname')){
            $name=getChild($bestandinfo, 'Bestandsname');
            $name=$name->nodeValue;
        }
        if(hasChild($bestandinfo, 'Bestand_Kurz')){
            $kurz=getChild($bestandinfo, 'Bestand_Kurz');
            $kurz=$kurz->nodeValue;
        }
        if(hasChild($bestandinfo, 'Bestand_Sig')){
            $sig=getChild($bestandinfo, 'Bestand_Sig');
            $sig=$sig->nodeValue;
        }
        if($anz == 1){$iter=null;
        } else {
            if(!empty($kurz)){
                $iter=$list->append(null, array(utf8_decode($kurz)));
            }
            elseif(!empty($sig)){
                $iter=$list->append(null, array(utf8_decode($sig)));
            }
            elseif(!empty($name)){
                $iter=$list->append(null, array(utf8_decode($name)));
            }
            else{$iter=$list->append(null, array('*'));}
        }
        if(tag_exists($bestandinfo, "Provenienz")){
            $provs=get_tags($bestandinfo, "Provenienz");
            $anzahl=$provs->length;
            for ($b=0;$b<$anzahl;$b++){
                $prov = $provs->item($b);
                $list->append($iter, array(utf8_decode($prov->nodeValue)));
            }
        } else{$list->append($iter, array('*'));}
    }
    if(empty($id)){$text='';}
    else {$element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
}

```

```

}
if ($anz == 0){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('provenienz');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}
if ($anz >= 1){
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('provenienz');
    $comboentry = GtkComboBoxEntry::new_text();
    $comboentry->set_model($list);
    $comboentry->get_child()->set_text($text);
    $hbox->pack_start($comboentry, true, true, 5);
    return $hbox;
}
}

function elementrechtsstatus($id){
    global $vedom;
    if(empty($id)){$text='';}
    else {$element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);}
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementseite($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementserientitel($id){
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
}

```



```

} else {$text='';}
$hbox= new GtkHBox(false, 0);
$entry = new GtkTextView();
$entry->set_wrap_mode(Gtk::WRAP_WORD);
$descbuffer = new GtkTextBuffer();
$entry->set_buffer($descbuffer);
$descbuffer->set_text($text);
$entry->set_editable(true);
$entry->set_cursor_visible(true);
$entryscroll = new GtkScrolledWindow();
$entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$entryscroll->add_with_viewport($entry);
$entryscroll->show_all();
$hbox->pack_start($entryscroll, true, true, 5);
return $hbox;
}

function elementsignatur($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('signaturpc');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}

function elementsperrvermerk($id){
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING , Gtk::T
YPE_STRING);
    $attrs= new GtkListStore(Gtk::TYPE_STRING);
    $attrs->append(array("konservatorisch"));
    $attrs->append(array('Landesrecht'));
    $attrs->append(array('Bundesrecht'));
    $attrs->append(array('Sonstige'));
    $attrs->append(array(''));
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Sperrvermerk')==true){
            $elements=$element->childNodes;
            for ($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "Sperrvermerk"){
                    $id=$child->getAttribute("idstef");
                    $enthart=$child->getAttribute("Sperr_Art");
                }
            }
        }
    }
}

```

```

        $art=$schild->getAttribute("Art");
        if (empty($enthart)){ $attr=$art;}
        else { $attr=$enthart;}
        if (empty($attr)) { $attr="";}
        if (hasChild($schild, "Datum")==true){
            $dates=getChild($schild, "Datum");
            $date=$dates->nodeValue;
        }
        else{ $date='';}
        if (hasChild($schild, "User")==true){
            $whos=getChild($schild, "User");
            $who=$whos->nodeValue;
        }
        else{ $who='';}
        $enthmodel->append(array($id, utf8_decode($attr), utf8_decode($date),
utf8_decode($who)));
    }
}
} else{ $enthmodel->append(array(' ', ' ', ' ', ' '));}
$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$enthcoll = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrcol->set_title('Grund');
$enthcol->set_title('bis:');
$enthcoll->set_title('für:');
$identhcol->set_resizable(true);
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcoll->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer3 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcoll->pack_start($enthcell_renderer3, true);
$enthcell_renderer2->set_property('width', 100);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable', true);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer3->set_property('editable', true);
$enthcell_renderer3->connect('edited', 'cell_edited', $enthmodel, 3);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);

```

```

$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
$enthcoll->set_attributes($enthcell_renderer3, 'text', 3);
// $enthlist->append_column($identhcol);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->append_column($enthcoll);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();
return $enthscroll;
}

```

```

function elementtext($id) {
    $note= new GtkNotebook();
    global $vedom;
    if (!empty($id)) {
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Text')) {
            $elements=$element->childNodes;
            for ($a=0; $a<$elements->length; $a++) {
                $child=$elements->item($a);
                if ($child->nodeName == "Text") {
                    $ids=$child->getAttribute("idstef");
                    $titart=$child->getAttribute("Text_Art");
                    $art=$child->getAttribute("Art");
                    if (empty($titart)) {$attr=$art;}
                    else {$attr=$titart;}
                    $vbox= new GtkVBox(false, 0);
                    $entry = new GtkTextView();
                    $entry->set_wrap_mode(Gtk::WRAP_WORD);
                    $descbuffer = new GtkTextBuffer();
                    $entry->set_buffer($descbuffer);
                    $entry->set_editable(true);
                    $entry->set_cursor_visible(true);
                    $entryscroll = new GtkScrolledWindow();
                    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
                    $entryscroll->add_with_viewport($entry);
                    $entryscroll->show_all();
                    $comboentry = GtkComboBoxEntry::new_text();
                    $data = array('Bestandsbeschreibung', 'Bestandsinhalt', 'Bewertung',
'Erschließung', 'Prov_Geschichte', 'Vorwort', '');
                    foreach ($data as $str) {$comboentry->append_text($str);}
                    $text=utf8_decode($child->nodeValue);
                    $descbuffer->set_text($text);
                    $comboentry->get_child()->set_text($attr);
                }
            }
        }
    }
}

```

```

        if (empty($attr)){ $attr="Titel";}
        $vbox->pack_start($comboentry, false, false, 5);
        $vbox->pack_start($entryscroll, true, true, 0);
        $note->append_page($vbox, new GtkLabel($attr));
    }
}
} else{
    $vbox= new GtkVBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Bestandsbeschreibung', 'Bestandsinhalt', 'Bewertung',
'Erschließung', 'Prov_Geschichte', 'Vorwort', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $vbox->pack_start($comboentry, false, false, 5);
    $vbox->pack_start($entryscroll, true, true, 0); $note->append_page($vbox, new GtkLabel
('Text'));}
} else{
    $vbox= new GtkVBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Bestandsbeschreibung', 'Bestandsinhalt', 'Bewertung', 'Erschließung',
'Prov_Geschichte', 'Vorwort', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $vbox->pack_start($comboentry, false, false, 5);
    $vbox->pack_start($entryscroll, true, true, 0);
    $note->append_page($vbox, new GtkLabel('Text'));
}
return $note;
}

function elementtitel($id, $mode){
    if ($mode== 'single') {
        $text='';

```

```

    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 0);
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $descbuffer->set_text($text);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $hbox->pack_start($entryscroll, true, true, 5);
    return $hbox;
}
if ($mode== 'multi') {
    global $vedom;
    $note= new GtkNotebook();
    if (!empty($id)) {
        $element=$vedom->getElementById($id);
        if (hasChild($element, 'Titel')) {
            $elements=$element->childNodes;
            for ($a=0; $a<$elements->length; $a++) {
                $child=$elements->item($a);
                if ($child->nodeName == "Titel") {
                    $ids=$child->getAttribute("idstef");
                    $titart=$child->getAttribute("Titel_Art");
                    $art=$child->getAttribute("Art");
                    if (empty($titart)) {$attr=$art;}
                    else {$attr=$titart;}
                    $text=utf8_decode($child->nodeValue);
                    $entry = new GtkTextView();
                    $entry->set_wrap_mode(Gtk::WRAP_WORD);
                    $descbuffer = new GtkTextBuffer();
                    $entry->set_buffer($descbuffer);
                    $entry->set_editable(true);
                    $entry->set_cursor_visible(true);
                    $entryscroll = new GtkScrolledWindow();
                    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
                    $entryscroll->add_with_viewport($entry);
                    $entryscroll->show_all();
                    $comboentry = GtkComboBoxEntry::new_text();
                    $data = array('Haupttitel', 'Untertitel', 'Originaltitel',
'Kurztitel', 'Arbeitstitel', 'Archivtitel', '');
                    foreach ($data as $str) {$comboentry->append_text($str);}
                }
            }
        }
    }
}

```

```

        $vbox= new GtkVBox(false, 0);
        $vbox->pack_start($comboentry, false, false, 5);
        $vbox->pack_start($entryscroll, true, true, 0);
        $descbuffer->set_text($text);
        $comboentry->get_child()->set_text($attr);
        if (empty($attr)){ $attr="Titel";}
        $note->append_page($vbox, new GtkLabel($attr));
    }
}
} else{
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Haupttitel', 'Untertitel', 'Originaltitel', 'Kurztitel',
'Arbeitstitel', 'Archivtitel', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $vbox= new GtkVBox(false, 0);
    $vbox->pack_start($comboentry, false, false, 5);
    $vbox->pack_start($entryscroll, true, true, 0);
    $note->append_page($vbox, new GtkLabel('Titel'));}
} else{
    $entry = new GtkTextView();
    $entry->set_wrap_mode(Gtk::WRAP_WORD);
    $descbuffer = new GtkTextBuffer();
    $entry->set_buffer($descbuffer);
    $entry->set_editable(true);
    $entry->set_cursor_visible(true);
    $entryscroll = new GtkScrolledWindow();
    $entryscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
    $entryscroll->add_with_viewport($entry);
    $entryscroll->show_all();
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Haupttitel', 'Untertitel', 'Originaltitel', 'Kurztitel',
'Arbeitstitel', 'Archivtitel', '');
    foreach ($data as $str) {$comboentry->append_text($str);}
    $vbox= new GtkVBox(false, 0);
    $vbox->pack_start($comboentry, false, false, 5);
    $vbox->pack_start($entryscroll, true, true, 0);
    $note->append_page($vbox, new GtkLabel('Titel'));
}
return $note;
}
}

```

```

}

function elementtopogrdaten($id) {
    $enthmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    $attrs= new GtkTreeStore(Gtk::TYPE_STRING);
    $attrs->append(null, array("TK-Wert"));
    $iter=$attrs->append(null, array('Gauß-Krüger'));
    $attrs->append($iter, array('hoch'));
    $attrs->append($iter, array('rechts'));
    $attrs->append($iter, array('Ident. '));
    $iter=$attrs->append(null, array('Grade'));
    $attrs->append($iter, array('Länge (g|f)'));
    $attrs->append($iter, array('Breite'));
    $attrs->append(null, array('Bemerkung'));
    $attrs->append(null, array('Sonstige'));
    if(!empty($id)) {
        global $vedom;
        $element=$vedom->getElementById($id);
        if ($element->hasChildNodes()==true) {
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){
                $child=$elements->item($a);
                if ($child->nodeName == "TK"){
                    $id=$child->getAttribute("idstef");
                    $attr="TK-Wert";
                }
                if ($child->nodeName == "Hilfsfeld"){
                    $id=$child->getAttribute("idstef");
                    $attr="Sonstige";
                }
                if ($child->nodeName == "Bem"){
                    $id=$child->getAttribute("idstef");
                    $attr="Bemerkung";
                }
                if ($child->nodeName == "GK_hoch"){
                    $id=$child->getAttribute("idstef");
                    $attr="hoch";
                }
                if ($child->nodeName == "GK_rechts"){
                    $id=$child->getAttribute("idstef");
                    $attr="rechts";
                }
                if ($child->nodeName == "GK_Identifikation"){
                    $id=$child->getAttribute("idstef");
                    $attr="Ident.";
                }
                if ($child->nodeName == "Breitengrad"){
                    $id=$child->getAttribute("idstef");
                    $attr="Breite";
                }
            }
        }
    }
}

```

```

$value=$child->nodeValue;
if ($child->nodeName == "Laengengrad"){
    $id=$child->getAttribute("idstef");
    if($child->hasAttribute('Meridian_Art')) {
        $meri=$child->getAttribute('Meridian_Art');
        if ($meri=='Greenwich'){$text=' (g) '};
        elseif ($meri=='Ferro'){$text=' (f) '};
        else {$text=' '};
    } else {$text=' '};
    $attr="Länge (g|f)";
    $value=$child->nodeValue.$text;
}
$enthmodel->append(array($id, utf8_decode($attr), utf8_decode($value)));
}
}
}else{$enthmodel->append(array(' ', ' ', ' '));}
$enthlist = new GtkTreeView;
$identhcol = new GtkTreeViewColumn();
$enthattrcol = new GtkTreeViewColumn();
$enthcol = new GtkTreeViewColumn();
$identhcol->set_title('ID-Nummer');
$enthattrcol->set_title('Art');
$enthcol->set_title('Daten');
$identhcol->set_resizable(true);
$enthattrcol->set_resizable(true);
$enthcol->set_resizable(true);
$enthcell_renderer1 = new GtkCellRendererText();
$enthcell_renderer2 = new GtkCellRendererCombo();
$identhcol->pack_start($enthcell_renderer1, true);
$enthattrcol->pack_start($enthcell_renderer2, true);
$enthcol->pack_start($enthcell_renderer1, true);
$enthcell_renderer2->set_property('width', 120);
$enthcell_renderer1->set_property('editable', true);
$enthcell_renderer2->set_property('model', $attrs);
$enthcell_renderer2->set_property('text-column', 0);
$enthcell_renderer2->set_property('editable' , true);
$enthcell_renderer1->connect('edited', 'cell_edited', $enthmodel, 2);
$enthcell_renderer2->connect('edited', 'cell_edited', $enthmodel, 1);
$identhcol->set_attributes($enthcell_renderer1, 'text', 0);
$enthattrcol->set_attributes($enthcell_renderer2, 'text', 1);
$enthcol->set_attributes($enthcell_renderer1, 'text', 2);
// $bemlist->append_column($idbemcol);
$enthlist->append_column($enthattrcol);
$enthlist->append_column($enthcol);
$enthlist->set_model($enthmodel);
$enthscroll = new GtkScrolledWindow();
$enthscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$enthscroll->add_with_viewport($enthlist);
$enthscroll->show_all();

```



```

    return $enthscroll;
}

function elementumfang($id) {
    $attr='';
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
        $attr=$element->getAttribute("Mass");
        $text=utf8_decode($element->nodeValue);
    }
    $hbox= new GtkHBox(false, 0);
    $entry=new GtkEntry();
    $entry->set_text($text);
    $entry->set_size_request(45,25);
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('Blatt', 'Seiten', 'Folio', 'cm', 'VE', 'lfd.m.', 'Anzahl', 'Kartons');
    foreach ($data as $string) {$comboentry->append_text($string);
    }
    $comboentry->get_child()->set_text($attr);
    $comboentry->set_size_request(85,25);
    $hbox->pack_start($entry, true, true, 5);
    $hbox->pack_start($comboentry, false, false, 5);
    return $hbox;
}

function elementverweise($id) {
    $archref=elementarchref($id);
    $bibref=elementbibref($id);
    $fmref=elementfmref($id);
    $altuebform=elementaltuebform($id);
    $hbox= new GtkHBox(false, 5);
    $verweisnote= new GtkNoteBook();
    $verweisnote->append_page($archref, new GtkLabel('Archivalien'));
    $verweisnote->append_page($bibref, new GtkLabel('Literatur'));
    $verweisnote->append_page($fmref, new GtkLabel('Findmittel'));
    $verweisnote->append_page($altuebform, new GtkLabel('Alt. Überlieferung'));
    $hbox->pack_start($verweisnote, true, true, 5);
    return $hbox;
}

function elementvorgaenge($id) {
    global $vedom;
    if (!empty($id)) {
        $note= new GtkNotebook();
        $element=$vedom->getElementById($id);
        if (hasChild($element,'Vorgang_Sachakte') or hasChild($element,'Vorgang_Urkunde') or h
asChild($element,'Vorgang_Foto') or hasChild($element,'Vorgang_Karte')){
            $elements=$element->childNodes;
            for($a=0;$a<$elements->length;$a++){

```

```

    $schild=$elements->item($a);
    if ($schild->nodeName == "Vorgang_Sachakte"){
        $sids=$schild->getAttribute("idstef");
        $vorgang=elementvorgsachakte($sids);
        $note->append_page($vorgang, new GtkLabel('Sachakte'));
    }
    if ($schild->nodeName == "Vorgang_Urkunde"){
        $sids=$schild->getAttribute("idstef");
        $vorgang=elementvorgurkunde($sids);
        $note->append_page($vorgang, new GtkLabel('Urkunde'));
    }
    if ($schild->nodeName == "Vorgang_Foto"){
        $sids=$schild->getAttribute("idstef");
        $vorgang=elementvorgfoto($sids);
        $note->append_page($vorgang, new GtkLabel('Foto'));
    }
    if ($schild->nodeName == "Vorgang_Karte"){
        $sids=$schild->getAttribute("idstef");
        $vorgang=elementvorgkarte($sids);
        $note->append_page($vorgang, new GtkLabel('Karte'));
    }
}
return $note;
} else {$label=new GtkLabel('Noch keine Vorgänge vorhanden!'); return $label;}
} else {$label=new GtkLabel('Noch keine Vorgänge vorhanden!'); return $label;}
}

function elementvorgfoto($id){
    global $vedom;
    $ve=$vedom->getElementById($id);
    if (hasChild($ve, 'Laufzeit')==true){
        $schild=getChild($ve, 'Laufzeit');
        $sidlz=$schild->getAttribute('idstef');
    } else {$sidlz='';}
    if (hasChild($ve, 'Titel')==true){$sidthitel=$id;}
    else {$sidthitel='';}
    if (hasChild($ve, 'Enthaelt')==true){$sidenthaelt=$id;}
    else {$sidenthaelt='';}
    if (hasChild($ve, 'Umfang')==true){
        $schild=getChild($ve, 'Umfang');
        $sidumfang=$schild->getAttribute('idstef');
    } else {$sidumfang='';}
    if (hasChild($ve, 'Sperrvermerk')==true){$sidsperrvermerk=$id;}
    else {$sidsperrvermerk='';}
    if (hasChild($ve, 'Bestand_Kurz')==true){
        $schild=getChild($ve, 'Bestand_Kurz');
        $sidbestkurz=$schild->getAttribute('idstef');
    } else {$sidbestkurz='';}
    if (hasChild($ve, 'Provenienz')==true){
        $schild=getChild($ve, 'Provenienz');
    }
}

```

```

    $idprovenienz=$schild->getAttribute('idstef');
} else {$idprovenienz='';}
if (hasChild($ve, 'Bestellsig')==true){
    $schild=getChild($ve, 'Bestellsig');
    $idbestellsig=$schild->getAttribute('idstef');
} else {$idbestellsig='';}
if (hasChild($ve, 'Abg_Stelle')==true){
    $schild=getChild($ve, 'Abg_Stelle');
    $idabgstelle=$schild->getAttribute('idstef');
} else {$idabgstelle='';}
if (hasChild($ve, 'Akzession')==true){
    $schild=getChild($ve, 'Akzession');
    $idakzession=$schild->getAttribute('idstef');
} else {$idakzession='';}
if (hasChild($ve, 'Lagerung')==true){
    $schild=getChild($ve, 'Lagerung');
    $idlagerung=$schild->getAttribute('idstef');
} else {$idlagerung='';}
if (hasChild($ve, 'Zustand')==true){
    $schild=getChild($ve, 'Zustand');
    $idzustand=$schild->getAttribute('idstef');
} else {$idzustand='';}
if (hasChild($ve, 'Altsig')==true){$idaltsig=$id;}
else {$idaltsig='';}
if (hasChild($ve, 'Vor_Prov')==true){$idvorprov=$id;}
else {$idvorprov='';}
if (hasChild($ve, 'Anlass')==true){
    $schild=getChild($ve, 'Anlass');
    $idanlass=$schild->getAttribute('idstef');
} else {$idanlass='';}
if (hasChild($ve, 'Beschreibung')==true){
    $schild=getChild($ve, 'Beschreibung');
    $idbesch=$schild->getAttribute('idstef');
} else {$idbesch='';}
if (hasChild($ve, 'Person')==true){$idperson=$id;}
else {$idperson='';}
if (hasChild($ve, 'Institution')==true){$idinst=$id;}
else {$idinst='';}
if (hasChild($ve, 'Ort')==true){$idort=$id;}
else {$idort='';}
if (hasChild($ve, 'Format')==true){
    $schild=getChild($ve, 'Format');
    $idform=$schild->getAttribute('idstef');
} else {$idform='';}
if (hasChild($ve, 'Farbe')==true){
    $schild=getChild($ve, 'Farbe');
    $idfarbe=$schild->getAttribute('idstef');
} else {$idfarbe='';}
if (hasChild($ve, 'Material')==true){

```

```

    $schild=getChild($ve, 'Material');
    $idmat=$schild->getAttribute('idstef');
} else {$idmat='';}
if (hasChild($ve, 'Entstehungsstufe')==true){
    $schild=getChild($ve, 'Entstehungsstufe');
    $identstufe=$schild->getAttribute('idstef');
} else {$identstufe='';}
if (hasChild($ve, 'Bem')==true){$idbem=$id;}
else {$idbem='';}
$plhbox1= new GtkHBox(true, 8);
$bestandkurz=elementbestandkurz($idbestandkurz);
$framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
$umfang=elementumfang($idumfang);
$frameumfang = createframe('Umfang', $umfang, 'V', true);
$plhbox1->pack_start($framebestandkurz, true, true, 5);
$plhbox1->pack_start($frameumfang, true, true, 5);
$plhbox2= new GtkHBox(false, 0);
$titel=elementtitel($idtitel, 'multi');
$plhbox2->pack_start($titel, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$enthaelt=elemententhaelt($identhaelt);
$frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);
$plhbox3->pack_start($frameenthaelt, true, true, 5);
$plhbox4= new GtkHBox(false, 8);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'V', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->set_name('Sachakte1');
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$p2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);
$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprovenienz);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);

```

```

$lagerung=elementlagerung($idlagerung);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$p2hbox3= new GtkHBox(false, 0);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$p2hbox3->pack_start($framealtsig, true, true, 5);
$p2hbox3->pack_start($framevorprov, true, true, 5);
$p2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($id);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$p2hbox4->pack_start($frameverweis, true, true, 5);
$p2hbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$p2hbox5->pack_start($framebem, true, true, 5);
$p2vbox = new GtkVBox(false, 8);
$p2vbox->set_name('Sachakte2');
$p2vbox->pack_start($p2hbox1, false, false, 5);
$p2vbox->pack_start($p2hbox2, false, false, 5);
$p2vbox->pack_start($p2hbox3, true, true, 5);
$p2vbox->pack_start($p2hbox4, false, false, 5);
$p2vbox->pack_start($p2hbox5, false, false, 5);
$p3hbox1= new GtkHBox(true, 8);
$anlass=elementanlass($idanlass);
$frameanlass = createframe('Anlass', $anlass, 'V', true);
$p3hbox1->pack_start($frameanlass, true, true, 5);
$p3hbox2= new GtkHBox(true, 8);
$beschreibung=elementbeschreibung($idbesch);
$framebeschreibung = createframe('Beschreibung', $beschreibung, 'HV', true);
$person=elementperson($idperson);
$frameperson = createframe('Person', $person, 'HV', true);
$p3hbox2->pack_start($framebeschreibung, true, true, 5);
$p3hbox2->pack_start($frameperson, true, true, 5);
$p3hbox3= new GtkHBox(true, 8);
$institution=elementinstitution($idinst);
$frameinstitution = createframe('Institution', $institution, 'HV', true);
$ort=elementort($idort);
$frameort = createframe('Ort', $ort, 'HV', true);
$p3hbox3->pack_start($frameinstitution, true, true, 5);
$p3hbox3->pack_start($frameort, true, true, 5);
$p3hbox4= new GtkHBox(true, 8);

```

```

$format=elementformat($idform);
$frameformat = createframe('Format', $format, 'V', true);
$farbe=elementfarbe($idfarbe);
$framefarbe = createframe('Farbe', $farbe, 'V', true);
$p3hbox4->pack_start($frameformat, true, true, 5);
$p3hbox4->pack_start($framefarbe, true, true, 5);
$p3hbox5= new GtkHBox(true, 8);
$material=elementmaterial($idmat);
$framematerial = createframe('Material', $material, 'V', true);
$entstufe=elemententstehungsstufe($identstufe);
$frameentstufe = createframe('Entstehungsstufe', $entstufe, 'V', true);
$p3hbox5->pack_start($framematerial, true, true, 5);
$p3hbox5->pack_start($frameentstufe, true, true, 5);
$p3vbox = new GtkVBox(false, 8);
$p3vbox->pack_start($p3hbox1, false, false, 5);
$p3vbox->pack_start($p3hbox2, false, false, 5);
$p3vbox->pack_start($p3hbox3, true, true, 5);
$p3vbox->pack_start($p3hbox4, false, false, 5);
$p3vbox->pack_start($p3hbox5, false, false, 5);
$note= new GtkNotebook();
$note->append_page($p1vbox);
$note->append_page($p2vbox);
$note->append_page($p3vbox);
return $note;
}

```

```

function elementvorgkarte($id) {
    global $vedom;
    $ve=$vedom->getElementById($id);
    if (hasChild($ve, 'Laufzeit')==true) {
        $child=getChild($ve, 'Laufzeit');
        $idlz=$child->getAttribute('idstef');
    } else {$idlz='';}
    if (hasChild($ve, 'Titel')==true) {$idttitel=$id;}
    else {$idttitel='';}
    if (hasChild($ve, 'Beschreibung')==true) {
        $child=getChild($ve, 'Beschreibung');
        $idbesch=$child->getAttribute('idstef');
    } else {$idbesch='';}
    if (hasChild($ve, 'Umfang')==true) {
        $child=getChild($ve, 'Umfang');
        $idumfang=$child->getAttribute('idstef');
    } else {$idumfang='';}
    if (hasChild($ve, 'Sperrvermerk')==true) {$idsperrvermerk=$id;}
    else {$idsperrvermerk='';}
    if (hasChild($ve, 'Bestand_Kurz')==true) {
        $child=getChild($ve, 'Bestand_Kurz');
        $idbestkurz=$child->getAttribute('idstef');
    }
}

```

```

} else {$idbestkurz='';}
if (hasChild($ve, 'Provenienz')==true) {
    $child=getChild($ve, 'Provenienz');
    $idprovenienz=$child->getAttribute('idstef');
} else {$idprovenienz='';}
if (hasChild($ve, 'Bestellsig')==true) {
    $child=getChild($ve, 'Bestellsig');
    $idbestellsig=$child->getAttribute('idstef');
} else {$idbestellsig='';}
if (hasChild($ve, 'Abg_Stelle')==true) {
    $child=getChild($ve, 'Abg_Stelle');
    $idabgstelle=$child->getAttribute('idstef');
} else {$idabgstelle='';}
if (hasChild($ve, 'Akzession')==true) {
    $child=getChild($ve, 'Akzession');
    $idakzession=$child->getAttribute('idstef');
} else {$idakzession='';}
if (hasChild($ve, 'Lagerung')==true) {
    $child=getChild($ve, 'Lagerung');
    $idlagerung=$child->getAttribute('idstef');
} else {$idlagerung='';}
if (hasChild($ve, 'Zustand')==true) {
    $child=getChild($ve, 'Zustand');
    $idzustand=$child->getAttribute('idstef');
} else {$idzustand='';}
if (hasChild($ve, 'Altsig')==true) {$idaltsig=$id;}
else {$idaltsig='';}
if (hasChild($ve, 'Vor_Prov')==true) {$idvorprov=$id;}
else {$idvorprov='';}
if (hasChild($ve, 'Enthaelt')==true) {$identhaelt=$id;}
else {$identhaelt='';}
if (hasChild($ve, 'Ort')==true) {$idort=$id;}
else {$idort='';}
if (hasChild($ve, 'Person')==true) {$idperson=$id;}
else {$idperson='';}
if (hasChild($ve, 'Institution')==true) {$idinst=$id;}
else {$idinst='';}
if (hasChild($ve, 'Format')==true) {
    $child=getChild($ve, 'Format');
    $idform=$child->getAttribute('idstef');
} else {$idform='';}
if (hasChild($ve, 'Material')==true) {
    $child=getChild($ve, 'Material');
    $idmat=$child->getAttribute('idstef');
} else {$idmat='';}
if (hasChild($ve, 'Entstehungsstufe')==true) {
    $child=getChild($ve, 'Entstehungsstufe');
    $identstufe=$child->getAttribute('idstef');
} else {$identstufe='';}

```

```

if (hasChild($ve, 'Auflage')==true){
    $child=getChild($ve, 'Auflage');
    $idauflage=$child->getAttribute('idstef');}
else {$idauflage='';}
if (hasChild($ve, 'Ausfuehrung')==true){
    $child=getChild($ve, 'Ausfuehrung');
    $idausf=$child->getAttribute('idstef');
} else {$idausf='';}
if (hasChild($ve, 'Kartentyp')==true){
    $child=getChild($ve, 'Kartentyp');
    $idktyp=$child->getAttribute('idstef');
} else {$idktyp='';}
if (hasChild($ve, 'Bem')==true){$idbem=$id;}
else {$idbem='';}
if (hasChild($ve, 'Az')==true){$idaz=$id;}
else {$idaz='';}
if (hasChild($ve, 'Einzeichnung')==true){$ideinz=$id;}
else {$ideinz='';}
if (hasChild($ve, 'Massstab')==true){
    $child=getChild($ve, 'Massstab');
    $idmasss=$child->getAttribute('idstef');
} else {$idmasss='';}
if (hasChild($ve, 'Nebenkarten')==true){
    $child=getChild($ve, 'Nebenkarten');
    $idnebk=$child->getAttribute('idstef');
} else {$idnebk='';}
if (hasChild($ve, 'Topogr_Daten')==true){
    $child=getChild($ve, 'Topogr_Daten');
    $idtopogr=$child->getAttribute('idstef');
} else {$idtopogr='';}
$plhbox1= new GtkHBox(true, 8);
$bestandkurz=elementbestandkurz($idbestandkurz);
$framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
$umfang=elementumfang($idumfang);
$frameumfang = createframe('Umfang', $umfang, 'V', true);
$plhbox1->pack_start($framebestandkurz, true, true, 5);
$plhbox1->pack_start($frameumfang, true, true, 5);
$plhbox2= new GtkHBox(false, 0);
$titel=elementtitel($idtitel, 'multi');
$plhbox2->pack_start($titel, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$enthaelt=elemententhaelt($identhaelt);
$frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);
$plhbox3->pack_start($frameenthaelt, true, true, 5);
$plhbox4= new GtkHBox(false, 8);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'V', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);

```



```

$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$p2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);
$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprovenienz);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);
$lagerung=elementlagerung($idlagerung);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$nebenkarten=elementnebenkarten($idnebk);
$framenebenkarten = createframe('Nebenkarten', $nebenkarten, 'HV', true);
$p2hbox3= new GtkHBox(false, 0);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$p2hbox3->pack_start($framealtsig, true, true, 5);
$p2hbox3->pack_start($framenebenkarten, true, true, 5);
$p2hbox3->pack_start($framevorprov, true, true, 5);
$p2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($id);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$az=elementaz($idaz);
$frameaz = createframe('Aktenzeichen', $az, 'HV', true);
$p2hbox4->pack_start($frameverweis, true, true, 5);
$p2hbox4->pack_start($frameaz, true, true, 5);
$p2hbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$p2hbox5->pack_start($framebem, true, true, 5);

```

```

$2vbox = new GtkVBox(false, 8);
$2vbox->pack_start($2hbox1, false, false, 5);
$2vbox->pack_start($2hbox2, false, false, 5);
$2vbox->pack_start($2hbox3, true, true, 5);
$2vbox->pack_start($2hbox4, false, false, 5);
$2vbox->pack_start($2hbox5, false, false, 5);
$3hbox1= new GtkHBox(true, 8);
$ktyp=elementkartentyp($idktyp);
$framektyp = createframe('Kartentyp', $ktyp, 'V', true);
$ausf=elementausfuehrung($idausf);
$frameausf = createframe('Ausführung', $ausf, 'V', true);
$3hbox1->pack_start($framektyp, true, true, 5);
$3hbox1->pack_start($frameausf, true, true, 5);
$3hbox2= new GtkHBox(true, 8);
$beschreibung=elementbeschreibung($idbesch);
$framebeschreibung = createframe('Beschreibung', $beschreibung, 'HV', true);
$einz=elementeinzeichnung($ideinz);
$frameeinz = createframe('Einzeichnung', $einz, 'HV', true);
$person=elementperson($idperson);
$frameperson = createframe('Person', $person, 'HV', true);
$3hbox2->pack_start($framebeschreibung, true, true, 5);
$3hbox2->pack_start($frameeinz, true, true, 5);
$3hbox2->pack_start($frameperson, true, true, 5);
$3hbox3= new GtkHBox(true, 8);
$institution=elementinstitution($idinst);
$frameinstitution = createframe('Institution', $institution, 'HV', true);
$ort=elementort($idort);
$frameort = createframe('Ort', $ort, 'HV', true);
$stopogr=elementtopogrdaten($idtopogr);
$frametopogr = createframe('Topogr. Daten', $stopogr, 'HV', true);
$3hbox3->pack_start($frameinstitution, true, true, 5);
$3hbox3->pack_start($frameort, true, true, 5);
$3hbox3->pack_start($frametopogr, true, true, 5);
$3hbox4= new GtkHBox(true, 8);
$format=elementformat($idform);
$frameformat = createframe('Format', $format, 'V', true);
$auflage=elementauflage($idaufgabe);
$frameauflage = createframe('Auflage', $auflage, 'V', true);
$massstab=elementmassstab($idmass);
$framemassstab = createframe('Massstab', $massstab, 'V', true);
$3hbox4->pack_start($frameformat, true, true, 5);
$3hbox4->pack_start($frameauflage, true, true, 5);
$3hbox4->pack_start($framemassstab, true, true, 5);
$3hbox5= new GtkHBox(true, 8);
$material=elementmaterial($idmat);
$framematerial = createframe('Material', $material, 'V', true);
$entstufe=elemententstehungsstufe($identstufe);
$frameentstufe = createframe('Entstehungsstufe', $entstufe, 'V', true);
$3hbox5->pack_start($framematerial, true, true, 5);

```

```

$3hbox5->pack_start($frameentstufe, true, true, 5);
$3vbox = new GtkVBox(false, 8);
$3vbox->pack_start($3hbox1, false, false, 5);
$3vbox->pack_start($3hbox2, false, false, 5);
$3vbox->pack_start($3hbox3, true, true, 5);
$3vbox->pack_start($3hbox4, false, false, 5);
$3vbox->pack_start($3hbox5, false, false, 5);
$note= new GtkNotebook();
$note->append_page($1vbox);
$note->append_page($2vbox);
$note->append_page($3vbox);
return $note;
}

function elementvorgsachakte($id) {
    global $vedom;
    $ve=$vedom->getElementById($id);
    if (hasChild($ve, 'Laufzeit')==true) {
        $child=getChild($ve, 'Laufzeit');
        $idlz=$child->getAttribute('idstef');
    } else {$idlz='';}
    if (hasChild($ve, 'Titel')==true) {
        $child=getChild($ve, 'Titel');
        $idtitel=$child->getAttribute('idstef');
    } else {$idtitel='';}
    if (hasChild($ve, 'Enthaelt')==true){$identhaelt=$id;}
    else {$identhaelt='';}
    if (hasChild($ve, 'Umfang')==true) {
        $child=getChild($ve, 'Umfang');
        $idumfang=$child->getAttribute('idstef');
    } else {$idumfang='';}
    if (hasChild($ve, 'Sperrvermerk')==true){$idsperrvermerk=$id;}
    else {$idsperrvermerk='';}
    if (hasChild($ve, 'Bestand_Kurz')==true) {
        $child=getChild($ve, 'Bestand_Kurz');
        $idbestkurz=$child->getAttribute('idstef');
    } else {$idbestkurz='';}
    if (hasChild($ve, 'Provenienz')==true) {
        $child=getChild($ve, 'Provenienz');
        $idprovenienz=$child->getAttribute('idstef');
    } else {$idprovenienz='';}
    if (hasChild($ve, 'Bestellsig')==true) {
        $child=getChild($ve, 'Bestellsig');
        $idbestellsig=$child->getAttribute('idstef');
    } else {$idbestellsig='';}
    if (hasChild($ve, 'Abg_Stelle')==true) {
        $child=getChild($ve, 'Abg_Stelle');
        $idabgstelle=$child->getAttribute('idstef');
    } else {$idabgstelle='';}
    if (hasChild($ve, 'Akzession')==true) {

```

```

    $schild=getChild($ve,'Akzession');
    $idakzession=$schild->getAttribute('idstef');
} else {$idakzession='';}
if (hasChild($ve,'Lagerung')==true){
    $schild=getChild($ve,'Lagerung');
    $idlagerung=$schild->getAttribute('idstef');
} else {$idlagerung='';}
if (hasChild($ve,'Zustand')==true){
    $schild=getChild($ve,'Zustand');
    $idzustand=$schild->getAttribute('idstef');
} else {$idzustand='';}
if (hasChild($ve,'Altsig')==true){$idaltsig=$id;}
else {$idaltsig='';}
if (hasChild($ve,'Vor_Prov')==true){$idvorprov=$id;}
else {$idvorprov='';}
if (hasChild($ve,'Bem')==true){$idbem=$id;}
else {$idbem='';}
if (hasChild($ve,'Az')==true){$idaz=$id;}
else {$idaz='';}
$plhbox1= new GtkHBox(true, 8);
$bestandkurz=elementbestandkurz($idbestkurz);
$framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
$umfang=elementumfang($idumfang);
$frameumfang = createframe('Umfang', $umfang, 'V', true);
$plhbox1->pack_start($framebestandkurz, true, true, 5);
$plhbox1->pack_start($frameumfang, true, true, 5);
$plhbox2= new GtkHBox(false, 0);
$titel=elementtitel($idtitel, 'single');
$frametitel = createframe('Titel', $titel, 'V', true);
$plhbox2->pack_start($frametitel, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$enthaelt=elemententhaelt($identhaelt);
$frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);
$plhbox3->pack_start($frameenthaelt, true, true, 5);
$plhbox4= new GtkHBox(false, 8);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'V', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$sp2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);

```

```

$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprovenienz);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);
$lagerung=elementlagerung($idlagerung);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$p2hbox3->pack_start($framealtsig, true, true, 5);
$p2hbox3->pack_start($framevorprov, true, true, 5);
$p2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($id);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$az=elementaz($idaz);
$frameaz = createframe('Aktenzeichen', $az, 'HV', true);
$p2hbox4->pack_start($frameverweis, true, true, 5);
$p2hbox4->pack_start($frameaz, true, true, 5);
$p2hbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$p2hbox5->pack_start($framebem, true, true, 5);
$p2vbox = new GtkVBox(false, 8);
$p2vbox->pack_start($p2hbox1, false, false, 5);
$p2vbox->pack_start($p2hbox2, false, false, 5);
$p2vbox->pack_start($p2hbox3, true, true, 5);
$p2vbox->pack_start($p2hbox4, false, false, 5);
$p2vbox->pack_start($p2hbox5, false, false, 5);
$note= new GtkNotebook();
$note->append_page($p1vbox);
$note->append_page($p2vbox);
return $note;
}

function elementvorgurkunde($id){
    $x=new GtkLabel('urkunde');
    return $x;
}

```

```

}

function elementvorprov($id, $mode){
    global $vedom;
    $vorprovmodel = new GtkListStore(Gtk::TYPE_STRING, Gtk::TYPE_STRING);
    if ($mode == 'Bestand'){
        $vorprovcell_renderer2 = new GtkCellRendererText();
        $vorprovcell_renderer2->set_property('width', 50);
        $vorprovcell_renderer2->set_property('editable', true);
    } else {
        $provslst = new GtkTreeStore(Gtk::TYPE_STRING);
        $vorprovcell_renderer2 = new GtkCellRendererCombo();
        $vorprovcell_renderer2->set_property('model'    , $provslst);
        $vorprovcell_renderer2->set_property('text-column', 0);
        $vorprovcell_renderer2->set_property('editable' , true);
        $bestinfo=get_tags($vedom, "Bestand_Info");
        $anz=$bestinfo->length;
        for ($a=0;$a<$anz;$a++){
            $bestandinfo = $bestinfo->item($a);
            $kurz=''; $name=''; $sig='';
            if(hasChild($bestandinfo, 'Bestandsname')){
                $name=getChild($bestandinfo, 'Bestandsname');
                $name=$name->nodeValue;
            }
            if(hasChild($bestandinfo, 'Bestand_Kurz')){
                $kurz=getChild($bestandinfo, 'Bestand_Kurz');
                $kurz=$kurz->nodeValue;
            }
            if(hasChild($bestandinfo, 'Bestand_Sig')){
                $sig=getChild($bestandinfo, 'Bestand_Sig');
                $sig=$sig->nodeValue;
            }
            if($anz == 1){$iter=null;
            } else {
                if(!empty($kurz)){
                    $iter=$provslst->append(null, array(utf8_decode($kurz)));
                }
                elseif(!empty($sig)){
                    $iter=$provslst->append(null, array(utf8_decode($sig)));
                }
                elseif(!empty($name)){
                    $iter=$provslst->append(null, array(utf8_decode($name)));
                }
                else{$iter=$provslst->append(null, array('*'));}
            }
            if(tag_exists($bestandinfo, "Vor_Prov")){
                $vorprovs=get_tags($bestandinfo, "Vor_Prov");
                $anzahl=$vorprovs->length;
                for ($b=0;$b<$anzahl;$b++){
                    $vorprov = $vorprovs->item($b);
                }
            }
        }
    }
}

```

```

        $provslst->append($iter, array(utf8_decode($vorprov->nodeValue)));
    }
} else{$provslst->append($iter, array('*'));}
if ($anz == 0){$provslst->append(null, array('*'));}
}
}
if(!empty($id)){
    $element=$vedom->getElementById($id);
    if (hasChild($element,'Vor_Prov')==true){
        $elements=$element->childNodes;
        for($a=0;$a<$elements->length;$a++){
            $child=$elements->item($a);
            if ($child->nodeName == "Vor_Prov"){
                $id=$child->getAttribute("idstef");
                $value=$child->nodeValue;
                $vorprovmodel->append(array($id, utf8_decode($value)));
            }
        }
    } else{$vorprovmodel->append(array('', ''));}
} else{$vorprovmodel->append(array('', ''));}
$vorprovlist = new GtkTreeView;
$idvorprovcol = new GtkTreeViewColumn();
$vorprovcol = new GtkTreeViewColumn();
$idvorprovcol->set_title('ID-Nummer');
$vorprovcol->set_title('Vorprovenienzen');
$idvorprovcol->set_resizable(true);
$vorprovcol->set_resizable(true);
$vorprovcell_renderer1 = new GtkCellRendererText();
$idvorprovcol->pack_start($vorprovcell_renderer1, true);
$vorprovcol->pack_start($vorprovcell_renderer2, true);
$vorprovcell_renderer1->set_property('width', 50);
$vorprovcell_renderer1->set_property('editable', true);
$vorprovcell_renderer2->connect('edited', 'cell_edited', $vorprovmodel, 1);
$idvorprovcol->set_attributes($vorprovcell_renderer1, 'text', 0);
$vorprovcol->set_attributes($vorprovcell_renderer2, 'text', 1);
//$vorprovlist->append_column($idvorprovcol);
$vorprovlist->append_column($vorprovcol);
$vorprovlist->set_model($vorprovmodel);
$vorprovscroll = new GtkScrolledWindow();
$vorprovscroll->set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
$vorprovscroll->add_with_viewport($vorprovlist);
$vorprovscroll->show_all();
return $vorprovscroll;
}

function elementzustand($id){
    $text='';
    if(!empty($id)){
        global $vedom;
        $element=$vedom->getElementById($id);
    }
}

```

```

        $text=utf8_decode($element->nodeValue);
    }
    $hbox=new GtkHbox(false,0);
    $hbox->set_name('signaturpc');
    $entry=new GtkEntry();
    $entry->set_text($text);
    $hbox->pack_start($entry, true, true, 5);
    return $hbox;
}
?>

```

C.5. functions.inc.php

```

<?php
function addind(){
    $dialog = new GtkDialog('Index hinzufügen');
    $dialog->set_resizable(false);
    $dialog->add_button(GTK::STOCK_OK, Gtk::RESPONSE_OK);
    $dialog->add_button(GTK::STOCK_CANCEL, Gtk::RESPONSE_CANCEL);
    $dialog->set_position('GTK_WIN_POS_CENTER');
    $hbox = new GtkHBox(false, 8);
    $hbox->set_border_width(8);
    $stock = GtkImage::new_from_stock(GTK::STOCK_DIALOG_INFO, Gtk::ICON_SIZE_DIALOG);
    $hbox->pack_start($stock, false, false, 0);
    $vbox1= new GtkVBox(false, 8);
    $label = new GtkLabel('Indexart wählen oder neu erstellen', 1);
    $comboentry = GtkComboBoxEntry::new_text();
    $data = array('', 'Ort', 'Sache', 'Person');
    foreach ($data as $string) {
        $comboentry->append_text($string);
    }
    $comboentry->get_child()->set_text('');
    $vbox1->pack_start($label, true, false, 10);
    $vbox1->pack_start($comboentry, true, false, 10);
    $hbox->pack_start($vbox1, false, false, 0);
    $dialog->vbox->pack_start($hbox, false, false, 0);
    $dialog->show_all();
    $response = $dialog->run();
    if ($response == Gtk::RESPONSE_OK) {
        $text = $comboentry->get_active_text();
        if($text== "Ort" or $text== "Sache" or $text== "Person"){
            print "\n Index-Art: $text\n";
        } else {print "\n Index: $text \n";}
    }
    $dialog->destroy();
}

function calend(){
    $dialog = new GtkDialog('Datum wählen');
    $dialog->set_resizable(false);

```



```

$dialog->add_button(Gtk::STOCK_OK, Gtk::RESPONSE_OK);
$dialog->add_button(Gtk::STOCK_CANCEL, Gtk::RESPONSE_CANCEL);
$dialog->set_position('GTK_WIN_POS_CENTER');
$scal = new GtkCalendar();
$dialog->vbox->pack_start($scal, false, false, 0);
$dialog->show_all();
$response = $dialog->run();
if ($response == Gtk::RESPONSE_OK) {
    errorhandle('warning', "Da die Funktion get_date momentan noch nicht\n verfügbar ist, wi
rd das datum nicht zurückgegeben");
}
$dialog->destroy();
}

function cell_edited($cell_renderer, $path, $newText, $model, $col) {
    $iter = $model->get_iter_from_string($path);
    $model->set($iter, $col, $newText);
}

function cleartree($model) {
    $model->clear();
}

function clickedTree($tree, $event){
    if ($event->type != Gdk::_2BUTTON_PRESS && $event->type != Gdk::BUTTON_RELEASE) {
        return;
    }
    list($model, $arSelected) = $tree->get_selection()->get_selected_rows();
    $path = implode(':', $arSelected[0]);
    if ($event->button == 1) {
        $tree->expand_row($path, $event->type == Gdk::_2BUTTON_PRESS);
    } else if ($event->button == 2) {
        $tree->expand_row($path, true);
    } else if ($event->button == 3) {
        $tree->collapse_row($path);
    }
}

function createframe($label, $widget, $mode, $pad){
    $frame = new GtkFrame($label);
    if ($mode=='HV'){
        $hbox= new GtkHBox(false, 0);
        $hbox->pack_start($widget, $pad, $pad, 5);
        $vbox= new GtkVBox(false, 0);
        $vbox->pack_start($hbox, $pad, $pad, 5);
    }
    if ($mode=='V'){
        $vbox= new GtkVBox(false, 0);
        $vbox->pack_start($widget, $pad, $pad, 5);
    }
    if ($mode=='H'){
        $vbox= new GtkHBox(false, 0);

```

```

        $vbox->pack_start($widget, $pad, $pad, 5);
    }
    $frame->add($vbox);
    return $frame;
}

function doPopup($window, $event, $menu)
{
    if ($event->button == 3) {
        $menu->popup();
    }
}

function errorhandle($icon, $text){
    $dialog = new GtkDialog();
    $dialog->set_resizable(false);
    $dialog->add_button(Gtk::STOCK_OK, Gtk::RESPONSE_OK);
    $dialog->set_position('GTK_WIN_POS_CENTER');
    $hbox = new GtkHBox(false, 8);
    $hbox->set_border_width(8);
    if($icon== 'error'){
        $stock = GtkImage::new_from_stock(Gtk::STOCK_DIALOG_ERROR, Gtk::ICON_SIZE_DIALOG);
    }
    elseif($icon== 'question') {
        $stock = GtkImage::new_from_stock(Gtk::STOCK_DIALOG_QUESTION, Gtk::ICON_SIZE_DIALOG);
    }
    elseif($icon== 'warning') {
        $stock = GtkImage::new_from_stock(Gtk::STOCK_DIALOG_WARNING, Gtk::ICON_SIZE_DIALOG);
    }
    elseif($icon== 'authentication') {
        $stock = GtkImage::new_from_stock(Gtk::STOCK_DIALOG_QUESTION, Gtk::ICON_SIZE_DIALOG);
    }
    else {
        $stock = GtkImage::new_from_stock($icon, Gtk::ICON_SIZE_DIALOG);
    }
    $hbox->pack_start($stock, false, false, 0);
    $label = new GtkLabel($text, 0);
    $hbox->pack_start($label, true, true, 00);
    $dialog->vbox->pack_start($hbox, false, false, 0);
    $dialog->show_all();
    $response = $dialog->run();
    $dialog->destroy();
}

function get_tags($handle,$name) {
    $tags=$handle->getElementsByTagName($name);
    return $tags;
}

function getChild($handle,$tagname) {
    $childs=$handle->childNodes;
    $anz=$childs->length;
}

```

```

$c=$handle->firstChild;
$a=0;
do { if ($c->nodeName == $tagname) {return $c;}
    $c=$c->nextSibling;
    $a++;
} while ($a<=$anz);
}

function getids($id) {
    global $vedom;
    $ve=$vedom->getElementById($id);
    $tag=$ve->nodeName;
    if ($ve->hasAttribute('Gba_Art') or $ve->hasAttribute('Art')) {
        $idattgba=$id;
    } else {$idattgba='';}
    if (hasChild($ve, 'Abg_Stelle')==true) {
        if ($tag== 'Foto_Plakat' or $tag== 'Karte' or $tag== 'Sachakte' or $tag==
'Grundbuch_akte') {
            $schild=getChild($ve, 'Abg_Stelle');
            $idabgstelle=$schild->getAttribute('idstef');
        } else {$idabgstelle=$id;}
    } else {$idabgstelle='';}
    if (hasChild($ve, 'Abkuerzungen')==true) {
        $abk=getChild($ve, 'Abkuerzungen');
        $idabk=$abk->getAttribute('idstef');
    } else {$idabk='';}
    if (hasChild($ve, 'Akzession')==true) {
        if ($tag== 'Foto_Plakat' or $tag== 'Karte' or $tag== 'Sachakte' or $tag==
'Grundbuch_akte') {
            $schild=getChild($ve, 'Akzession');
            $idakzession=$schild->getAttribute('idstef');
        } else {$idakzession=$id;}
    } else {$idakzession='';}
    if (hasChild($ve, 'Altsig')==true) {$idaltsig=$id;}
    else {$idaltsig='';}
    if (hasChild($ve, 'altuebform')==true or hasChild($ve, 'archref')==true or
hasChild($ve, 'bibref')==true or hasChild($ve, 'FM_ref')==true) {
        $idverweise=$id;
    } else {$idverweise='';}
    if (hasChild($ve, 'Anlass')==true) {
        $schild=getChild($ve, 'Anlass');
        $idanlass=$schild->getAttribute('idstef');
    } else {$idanlass='';}
    if (hasChild($ve, 'Anschrift')==true) {
        $idanschrift=$id;
    } else {$idanschrift='';}
    if (hasChild($ve, 'Auflage')==true) {
        $schild=getChild($ve, 'Auflage');
        $idaufgabe=$schild->getAttribute('idstef');
    } else {$idaufgabe='';}
}

```

```

if (hasChild($ve, 'Ausfuehrung')==true) {
    $schild=getChild($ve, 'Ausfuehrung');
    $idausf=$schild->getAttribute('idstef');
} else {$idausf='';}
if (hasChild($ve, 'Az')==true) {$sidaz=$id;}
else {$sidaz='';}
if (hasChild($ve, 'Band')==true) {
    $schild=getChild($ve, 'Band');
    $idband=$schild->getAttribute('idstef');
} else {$idband='';}
if (hasChild($ve, 'Bearbeiter')==true) {$sidbearb=$id;}
else {$sidbearb='';}
if (hasChild($ve, 'Bem')==true) {$sidbem=$id;}
else {$sidbem='';}
if (hasChild($ve, 'Beschreibung')==true) {
    $schild=getChild($ve, 'Beschreibung');
    $idbesch=$schild->getAttribute('idstef');
} else {$idbesch='';}
if (hasChild($ve, 'Bestandsname')==true) {
    $schild=getChild($ve, 'Bestandsname');
    $idbestname=$schild->getAttribute('idstef');
} else {$idbestname='';}
if (hasChild($ve, 'Bestand_Kurz')==true) {
    $schild=getChild($ve, 'Bestand_Kurz');
    $idbestkurz=$schild->getAttribute('idstef');
} else {$idbestkurz='';}
if (hasChild($ve, 'Bestand_Sig')==true) {
    $schild=getChild($ve, 'Bestand_Sig');
    $idbestsig=$schild->getAttribute('idstef');
} else {$idbestsig='';}
if (hasChild($ve, 'Bestellsig')==true) {
    $schild=getChild($ve, 'Bestellsig');
    $idbestellsig=$schild->getAttribute('idstef');
} else {$idbestellsig='';}
if (hasChild($ve, 'Bibliographie')==true) {$sidbib=$id;}
else {$sidbib='';}
if (hasChild($ve, 'Blatt')==true) {
    $schild=getChild($ve, 'Blatt');
    $idblatt=$schild->getAttribute('idstef');
} else {$idblatt='';}
if (hasChild($ve, 'Eigentuemmer')==true) {
    $schild=getChild($ve, 'Eigentuemmer');
    $ideigentum=$schild->getAttribute('idstef');
} else {$ideigentum='';}
if (hasChild($ve, 'Einzeichnung')==true) {$sideinz=$id;}
else {$sideinz='';}
if (hasChild($ve, 'Enthaelt')==true) {$sidenthaelt=$id;}
else {$sidenthaelt='';}
if (hasChild($ve, 'Entstehungsstufe')==true) {

```

```

        $schild=getChild($ve,'Entstehungsstufe');
        $identstufe=$schild->getAttribute('idstef');
    } else {$identstufe='';}
    if (hasChild($ve,'Farbe')==true){
        $schild=getChild($ve,'Farbe');
        $idfarbe=$schild->getAttribute('idstef');
    } else {$idfarbe='';}
    if (hasChild($ve,'FM_Name')==true){
        $schild=getChild($ve,'FM_Name');
        $idfmname=$schild->getAttribute('idstef');
    } else {$idfmname='';}
    if (hasChild($ve,'FM_Sig')==true){
        $schild=getChild($ve,'FM_Sig');
        $idfmsig=$schild->getAttribute('idstef');
    } else {$idfmsig='';}
    if (hasChild($ve,'Format')==true){
        $schild=getChild($ve,'Format');
        $idformat=$schild->getAttribute('idstef');
    } else {$idformat='';}
    if (hasChild($ve,'Gemarkung')==true){
        $schild=getChild($ve,'Gemarkung');
        $idgemark=$schild->getAttribute('idstef');
    } else {$idgemark='';}
    if (hasChild($ve,'Hilfsfeld')==true){$idhilf=$id;}
    else {$idhilf='';}
    if (hasChild($ve,'Institution')==true){$idinst=$id;}
    else {$idinst='';}
    if (hasChild($ve,'Kartentyp')==true){
        $schild=getChild($ve,'Kartentyp');
        $idktyp=$schild->getAttribute('idstef');
    } else {$idktyp='';}
    if (hasChild($ve,'Klass_Nr')==true){
        $schild=getChild($ve,'Klass_Nr');
        $idklassnr=$schild->getAttribute('idstef');
    } else {$idklassnr='';}
    if (hasChild($ve,'Klass_Titel')==true){
        $schild=getChild($ve,'Klass_Titel');
        $idklasstitel=$schild->getAttribute('idstef');
    } else {$idklasstitel='';}
    if (hasChild($ve,'Lagerung')==true){
        $schild=getChild($ve,'Lagerung');
        $idlager=$schild->getAttribute('idstef');
    } else {$idlager='';}
    if (hasChild($ve,'Laufzeit')==true){
        $schild=getChild($ve,'Laufzeit');
        $idlz=$schild->getAttribute('idstef');
    } else {$idlz='';}
    if (hasChild($ve,'Massstab')==true){
        $schild=getChild($ve,'Massstab');
    }

```

```

        $idmasss=$schild->getAttribute('idstef');
    } else {$idmasss='';}
    if (hasChild($ve, 'Material')==true) {
        $schild=getChild($ve, 'Material');
        $idmat=$schild->getAttribute('idstef');
    } else {$idmat='';}
    if (hasChild($ve, 'Nebenkarten')==true) {
        $schild=getChild($ve, 'Nebenkarten');
        $idnebk=$schild->getAttribute('idstef');
    } else {$idnebk='';}
    if (hasChild($ve, 'Ort')==true) {$idort=$id;}
    else {$idort='';}
    if (hasChild($ve, 'Person')==true) {$idperson=$id;}
    else {$idperson='';}
    if (hasChild($ve, 'Provenienz')==true) {
        if ($tag== 'Foto_Plakat' or $tag== 'Karte' or $tag== 'Sachakte' or $tag==
'Grundbuch_akte') {
            $schild=getChild($ve, 'Provenienz');
            $idprov=$schild->getAttribute('idstef');
        } else {$idprov=$id;}
    } else {$idprov='';}
    if (hasChild($ve, 'Rechtsstatus')==true) {
        $schild=getChild($ve, 'Rechtsstatus');
        $idrecht=$schild->getAttribute('idstef');
    } else {$idrecht='';}
    if (hasChild($ve, 'Seite')==true) {
        $schild=getChild($ve, 'Seite');
        $idseite=$schild->getAttribute('idstef');
    } else {$idseite='';}
    if (hasChild($ve, 'Serientitel')==true) {
        $schild=getChild($ve, 'Serientitel');
        $idserientitel=$schild->getAttribute('idstef');
    } else {$idserientitel='';}
    if (hasChild($ve, 'Signatur')==true) {
        $schild=getChild($ve, 'Signatur');
        $idsig=$schild->getAttribute('idstef');
    } else {$idsig='';}
    if (hasChild($ve, 'Sperrvermerk')==true) {$idsperrvermerk=$id;}
    else {$idsperrvermerk='';}
    if (hasChild($ve, 'Text')==true) {$idtext=$id;}
    else {$idtext='';}
    if (hasChild($ve, 'Titel')==true) {
        if ($tag== 'Foto_Plakat' or $tag== 'Karte') {$idttitel=$id;}
        else {$schild=getChild($ve, 'Titel');
            $idttitel=$schild->getAttribute('idstef');}
    } else {$idttitel='';}
    if (hasChild($ve, 'Topogr_Daten')==true) {
        $schild=getChild($ve, 'Topogr_Daten');
        $idtopogr=$schild->getAttribute('idstef');
    } else {$idtopogr='';}

```

```

if (hasChild($ve, 'Umfang')==true){
    $child=getChild($ve, 'Umfang');
    $idumfang=$child->getAttribute('idstef');
} else {$idumfang='';}
if (hasChild($ve, 'Vorgang_Sachakte') or hasChild($ve, 'Vorgang_Urkunde') or
hasChild($ve, 'Vorgang_Foto') or hasChild($ve, 'Vorgang_Karte')){
    $vorgid=$id;
} else {$vorgid='';}
if (hasChild($ve, 'Vor_Prov')==true){$idvorprov=$id;}
else {$idvorprov='';}
if (hasChild($ve, 'Zustand')==true){
    $child=getChild($ve, 'Zustand');
    $idzustand=$child->getAttribute('idstef');
} else {$idzustand='';}

if ($tag == "Einleitung"){
    mask_einleitung($id, $idtext, $idbem, $idbib, $idabk);
}
if ($tag == "Bestand_Info"){
    mask_bestandsinfo($id, $idbestname, $idbestsig, $idbestkurz, $idtext, $idlz, $idprov, $i
dvorprov, $idabgstelle, $idakzession, $idsperrvermerk, $idbearb, $idbem, $idrecht,
$idumfang, $idlager, $idverweise);
}
if ($tag == "Findmittel_Info"){
    mask_findinfo($id, $idfmname, $idfmsig, $idlz, $idbearb, $idbem, $idsperrvermerk,
$idhilf, $idverweise);
}
if ($tag == "Klassifikation"){
    mask_klassi($id, $idsperrvermerk, $idklassnr, $idklasstitel, $idbem, $idverweise);
}
if ($tag == "Sachakte"){
    mask_sach($id, $idsig, $idlz, $idtitel, $identhaelt, $idumfang, $idsperrvermerk,
$idbestkurz, $idbestellsig, $idlager, $idzustand, $idaltsig, $idvorprov, $vorgid, $idaz,
$idbem, $idabgstelle, $idakzession, $idprov, $idverweise);
}
if ($tag == "Konkordanz"){
    mask_konkordanz($id, $idbem);
}
if ($tag == "Bandserie"){
    mask_band($id, $idsperrvermerk, $idserientitel, $idlz, $identhaelt, $idbem,
$idverweise);
}
if ($tag == "Grundbuch_akte"){
    mask_grundbuch($id, $idsig, $idlz, $idgemark, $idband, $idumfang, $idsperrvermerk,
$idbestkurz, $idbestellsig, $idzustand, $idlager, $idaltsig, $idvorprov, $idbem, $idblatt,
$idseite, $ideigentum, $idanschrift, $idattgba, $idabgstelle, $idakzession, $idprov,
$idverweise);
}
if ($tag == "Indizes"){
    mask_indizes($id, $idbem);
}
if ($tag == "Karte"){
    mask_karte($id, $idsig, $idlz, $idbesch, $idumfang, $idsperrvermerk, $idbestkurz,

```

```

$Idbestellsig, $Idlager, $Idzustand, $Idaltsig, $Idvorprov, $Identhaelt, $Idort, $Idperson,
$Idinst, $Idformat, $Idmat, $Identstufe, $Idauflage, $Idausf, $Idbem, $Idaz, $Ideinz, $Idktyp,
$Idnebk, $Idmasss, $Idtopogr, $Idabgstelle, $Idakzession, $Idprov, $Idtitel, $Idverweise);
}
if ($tag == "Foto_Plakat"){
    mask_foto($id, $idsig, $idlz, $identhaelt, $idumfang, $idsperrvermerk, $idbestkurz,
$Idbestellsig, $Idlager, $Idzustand, $Idaltsig, $Idvorprov, $Idanlass, $Idort, $Idinst,
$Idperson, $Idbesch, $Idmat, $Identstufe, $Idbem, $Idformat, $Idfarbe, $Idabgstelle,
$Idakzession, $Idprov, $Idtitel, $Idverweise);
}
if ($tag == "Film"){
    mask_film($id);
}
if ($tag == "Index"){
    mask_index($id);
}
if ($tag == "Fallakte"){
    mask_fall($id);
}
if ($tag == "Urkunde"){
    mask_urkunde($id);
}
if ($tag == "Siegel"){
    mask_siegel($id);
}
}

function hasChild($handle, $tagname) {
    $exists=false;
    $childsyesno=$handle->hasChildNodes();
    if ($childsyesno == 0){return $exists;}
    else {$childs=$handle->childNodes;
        $anz=$childs->length;
        $c=$handle->firstChild;
        $a=0;
        do {if($c->nodeName == $tagname) {$exists=true;}
            $c=$c->nextSibling;
            $a++;
        } while ($a<$anz);
    return $exists;
}
}

function idattrs($n, $mode) {
    $children=$n->childNodes;
    $anz=$children->length;
    for ($a=0; $a<$anz;$a++) {
        $c=$children->item($a);
        if ($mode== 'ovidclear'){
            if ($c->nodeType == 1){
                $user=$c->getAttributeNode('user');
                if (!empty($user)){
                    global $userid;
                }
            }
        }
    }
}

```



```

        if ($user->nodeValue == $userid) {
            $user->nodeValue="";
            $states=$c->getAttributeNode('state');
            $states->nodeValue="open";
        }
    }
}
}
if ($mode== 'veidadd'){
    if ($c->nodeType == 1){
        static $zahl= 1;
        $id="ids".$zahl++;
        if ($c->hasAttribute("idstef")) {$c->removeAttribute("idstef");}
        $c->setAttribute("idstef", $id);
    }
}
if ($mode== 'veidclear'){
    if ($c->nodeType == 1){
        if ($c->hasAttribute("idstef")) {$c->removeAttribute("idstef");}
    }
}
if ($c->hasChildNodes()) {
    idattrs($c, $mode);
}
}
}

function load_xml($file) {
    $dom = new DomDocument();
    $dom->preserveWhiteSpace=false;
    $dom->load($file);
    if(!$dom) {
        echo "Error while parsing the document\n";
        exit();
    }
    return $dom;
}

function login(){
    $dialog = new GtkDialog('Login');
    $dialog->set_resizable(false);
    $dialog->add_button(Gtk::STOCK_OK, Gtk::RESPONSE_OK);
    $dialog->add_button(Gtk::STOCK_CANCEL, Gtk::RESPONSE_CANCEL);
    $dialog->set_position('GTK_WIN_POS_CENTER');
    $dialog->set_modal('true');
    $dialog->connect_simple('destroy', array('Gtk', 'main_quit'));
    $stock = GtkImage::new_from_stock(Gtk::STOCK_DIALOG_INFO, Gtk::ICON_SIZE_DIALOG);
    $hbox1 = new GtkHBox(false, 8);
    $hbox1->set_border_width(8);
    $idlabel = new GtkLabel('Benutzer-ID', 1);

```

```

$identry = new GtkEntry;
$hbox1->pack_start($idlabel, true, false, 10);
$hbox1->pack_start($identry, true, false, 10);
$hbox2 = new GtkHBox(false, 8);
$hbox2->set_border_width(8);
$pwlabel = new GtkLabel('Passwort', 1);
$pwentry = new GtkEntry;
$hbox2->pack_start($pwlabel, true, false, 10);
$hbox2->pack_start($pwentry, true, false, 10);
$vbox1= new GtkVBox(false, 8);
$vbox1->pack_start($hbox1, true, false, 10);
$vbox1->pack_start($hbox2, true, false, 10);
$hbox = new GtkHBox(false, 8);
$hbox->set_border_width(8);
$hbox->pack_start($stock, false, false, 0);
$hbox->pack_start($vbox1, false, false, 0);
$dialo->vbox->pack_start($hbox, false, false, 0);
$dialo->show_all();
$response = $dialo->run();
if ($response == Gtk::RESPONSE_OK) {
    $id = $identry->get_text();
    $pw = $pwentry->get_text();
    if(!empty($id)) {
        global $userid;
        $userid = $id;
        $dialo->destroy();
    }
}
if ($response == Gtk::RESPONSE_CANCEL) {
    $identry->set_text('');$pwentry->set_text('');
    global $window;
    $window->destroy();
    $dialo->destroy();
}
}

function model_overview(){
    global $overviewfile;
    $overviewfile="data/archiv.xml";
    global $overviewdom;
    $overviewdom=load_xml($overviewfile);
    $root=$overviewdom->documentElement;
    set_model_iterbest($root, null);
    global $besttree;
    $besttree->show_all();
}

function model_ve($file){
    global $vedom;
    global $vefile;

```

```

$vefile=utf8_decode($file);
$vedom=load_xml($vefile);
idattrs($vedom, 'veidadd');
$root=$vedom->documentElement;
set_model_iterve($root, null);
global $vetree;
$vetree->show_all();
}

function new_konkord($n, $konmodel) {
$children=$n->childNodes;
$anz=$children->length;
for ($a=0; $a<$anz;$a++) {
$c=$children->item($a);
$count=0;
if ($c->nodeName == "Sachakte") {
$id=$c->getAttributeNode("idstef")->nodeValue;
$sig=""; $bestsig=""; $altsig=""; $lager="";
if(hasChild($c, "Signatur") == true){
$sigs=getChild($c, "Signatur");
$sig=$sigs->nodeValue;
}
if(hasChild($c, "Bestellsig") == true){
$bestsigs=getChild($c, "Bestellsig");
$bestsig=$bestsigs->nodeValue;
}
if(hasChild($c, "Altsig") == true){
$altsigs=getChild($c, "Altsig");
$altsig=$altsigs->nodeValue;
}
if(hasChild($c, "Lagerung") == true){
$lagers=getChild($c, "Lagerung");
$lager=$lagers->nodeValue;
}
if (!empty($sig)== true) {$count++;}
if (!empty($bestsig)== true) {$count++;}
if (!empty($altsig)== true) {$count++;}
if (!empty($lager)== true) {$count++;}
if($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
if(tag_exists($c, "Vorgang_Sachakte") == true){
$vorg=get_tags($c, "Vorgang_Sachakte");
for($b=0; $b<$vorg->length;$b++) {
$bestsig=""; $altsig=""; $lager="";
$vor=$vorg->item($b);
$id=$vor->getAttributeNode("idstef")->nodeValue;
if(hasChild($vor, "Bestellsig") == true){
$bestsigs=getChild($vor, "Bestellsig");
$bestsig=$bestsigs->nodeValue;
}
}
}
}
}

```

```

        if(hasChild($vor, "Altsig") == true){
            $altsigs=getChild($vor, "Altsig");
            $altsig=$altsigs->nodeValue;
        }
        if(hasChild($vor, "Lagerung") == true){
            $lagers=getChild($vor, "Lagerung");
            $lager=$lagers->nodeValue;
        }
        if (!empty($sig)== true) {$count++;}
        if (!empty($bestsig)== true) {$count++;}
        if (!empty($altsig)== true) {$count++;}
        if (!empty($lager)== true) {$count++;}
        if($count>1){$skonmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
    }
}
if(tag_exists($c, "Vorgang_Urkunde") == true){
    $vorg=get_tags($c, "Vorgang_Urkunde");
    for($b=0; $b<$vorg->length;$b++) {
        $bestsig=""; $altsig=""; $lager="";
        $vor=$vorg->item($b);
        $id=$vor->getAttributeNode("idstef")->nodeValue;
        if(hasChild($vor, "Bestellsig") == true){
            $bestsigs=getChild($vor, "Bestellsig");
            $bestsig=$bestsigs->nodeValue;
        }
        if(hasChild($vor, "Altsig") == true){
            $altsigs=getChild($vor, "Altsig");
            $altsig=$altsigs->nodeValue;
        }
        if(hasChild($vor, "Lagerung") == true){
            $lagers=getChild($vor, "Lagerung");
            $lager=$lagers->nodeValue;
        }
        if (!empty($sig)== true) {$count++;}
        if (!empty($bestsig)== true) {$count++;}
        if (!empty($altsig)== true) {$count++;}
        if (!empty($lager)== true) {$count++;}
        if($count>1){$skonmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
    }
}
if(tag_exists($c, "Vorgang_Karte") == true){
    $vorg=get_tags($c, "Vorgang_Karte");
    for($b=0; $b<$vorg->length;$b++) {
        $bestsig=""; $altsig=""; $lager="";
        $vor=$vorg->item($b);
        $id=$vor->getAttributeNode("idstef")->nodeValue;
        if(hasChild($vor, "Bestellsig") == true){
            $bestsigs=getChild($vor, "Bestellsig");

```

```

        $bestsig=$bestsigs->nodeValue;
    }
    if(hasChild($vor, "Altsig") == true){
        $altsigs=getChild($vor, "Altsig");
        $altsig=$altsigs->nodeValue;
    }
    if(hasChild($vor, "Lagerung") == true){
        $lagers=getChild($vor, "Lagerung");
        $lager=$lagers->nodeValue;
    }
    if (!empty($sig)== true) {$count++;}
    if (!empty($bestsig)== true) {$count++;}
    if (!empty($altsig)== true) {$count++;}
    if (!empty($lager)== true) {$count++;}
    if($count>1){$skonmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
    }
}
if(tag_exists($c, "Vorgang_Foto") == true){
    $vorg=get_tags($c, "Vorgang_Foto");
    for($b=0; $b<$vorg->length;$b++) {
        $bestsig=""; $altsig=""; $lager="";
        $vor=$vorg->item($b);
        $id=$vor->getAttributeNode("idstef")->nodeValue;
        if(hasChild($vor, "Bestellsig") == true){
            $bestsigs=getChild($vor, "Bestellsig");
            $bestsig=$bestsigs->nodeValue;
        }
        if(hasChild($vor, "Altsig") == true){
            $altsigs=getChild($vor, "Altsig");
            $altsig=$altsigs->nodeValue;
        }
        if(hasChild($vor, "Lagerung") == true){
            $lagers=getChild($vor, "Lagerung");
            $lager=$lagers->nodeValue;
        }
        if (!empty($sig)== true) {$count++;}
        if (!empty($bestsig)== true) {$count++;}
        if (!empty($altsig)== true) {$count++;}
        if (!empty($lager)== true) {$count++;}
        if($count>1){$skonmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
    }
}
}
if ($c->nodeName == "Fallakte") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    $sig=""; $bestsig=""; $altsig=""; $lager="";
    if(hasChild($c, "Signatur") == true){
        $sigs=getChild($c, "Signatur");

```

```

        $sig=$sigs->nodeValue;
    }
    if(hasChild($c, "Bestellsig") == true){
        $bestsigs=getChild($c, "Bestellsig");
        $bestsig=$bestsigs->nodeValue;
    }
    if(hasChild($c, "Altsig") == true){
        $altsigs=getChild($c, "Altsig");
        $altsig=$altsigs->nodeValue;
    }
    if(hasChild($c, "Lagerung") == true){
        $lagers=getChild($c, "Lagerung");
        $lager=$lagers->nodeValue;
    }
    if (!empty($sig)== true) {$count++;}
    if (!empty($bestsig)== true) {$count++;}
    if (!empty($altsig)== true) {$count++;}
    if (!empty($lager)== true) {$count++;}
    if($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
}
if ($c->nodeName == "Grundbuch_akte") {
$id=$c->getAttributeNode("idstef")->nodeValue;
$sig=""; $bestsig=""; $altsig=""; $lager="";
if(hasChild($c, "Signatur") == true){
    $sigs=getChild($c, "Signatur");
    $sig=$sigs->nodeValue;
}
if(hasChild($c, "Bestellsig") == true){
    $bestsigs=getChild($c, "Bestellsig");
    $bestsig=$bestsigs->nodeValue;
}
if(hasChild($c, "Altsig") == true){
    $altsigs=getChild($c, "Altsig");
    $altsig=$altsigs->nodeValue;
}
if(hasChild($c, "Lagerung") == true){
    $lagers=getChild($c, "Lagerung");
    $lager=$lagers->nodeValue;
}
if (!empty($sig)== true) {$count++;}
if (!empty($bestsig)== true) {$count++;}
if (!empty($altsig)== true) {$count++;}
if (!empty($lager)== true) {$count++;}
if($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
}
if ($c->nodeName == "Urkunde") {
$id=$c->getAttributeNode("idstef")->nodeValue;
$sig=""; $bestsig=""; $altsig=""; $lager="";

```

```

if(hasChild($c, "Signatur") == true){
    $sigs=getChild($c, "Signatur");
    $sig=$sigs->nodeValue;
}
if(hasChild($c, "Bestellsig") == true){
    $bestsigs=getChild($c, "Bestellsig");
    $bestsig=$bestsigs->nodeValue;
}
if(hasChild($c, "Altsig") == true){
    $altsigs=getChild($c, "Altsig");
    $altsig=$altsigs->nodeValue;
}
if(hasChild($c, "Lagerung") == true){
    $lagers=getChild($c, "Lagerung");
    $lager=$lagers->nodeValue;
}
if (!empty($sig)== true) {$count++;}
if (!empty($bestsig)== true) {$count++;}
if (!empty($altsig)== true) {$count++;}
if (!empty($lager)== true) {$count++;}
if($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
}
if ($c->nodeName == "Siegel" && $c->parentNode->nodeName == "Verzeichnungseinheiten") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
}

if ($c->nodeName == "Karte") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    $sig=""; $bestsig=""; $altsig=""; $lager="";
    if(hasChild($c, "Signatur") == true){
        $sigs=getChild($c, "Signatur");
        $sig=$sigs->nodeValue;
    }
    if(hasChild($c, "Bestellsig") == true){
        $bestsigs=getChild($c, "Bestellsig");
        $bestsig=$bestsigs->nodeValue;
    }
    if(hasChild($c, "Altsig") == true){
        $altsigs=getChild($c, "Altsig");
        $altsig=$altsigs->nodeValue;
    }
    if(hasChild($c, "Lagerung") == true){
        $lagers=getChild($c, "Lagerung");
        $lager=$lagers->nodeValue;
    }
    if (!empty($sig)== true) {$count++;}
    if (!empty($bestsig)== true) {$count++;}
    if (!empty($altsig)== true) {$count++;}
    if (!empty($lager)== true) {$count++;}
    if($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,

```

```

utf8_decode($lager));}
}
if ($c->nodeName == "Foto_Plakat") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    $sig=""; $bestsig=""; $altsig=""; $lager="";
    if (hasChild($c, "Signatur") == true){
        $sigs=getChild($c, "Signatur");
        $sig=$sigs->nodeValue;
    }
    if (hasChild($c, "Bestellsig") == true){
        $bestsigs=getChild($c, "Bestellsig");
        $bestsig=$bestsigs->nodeValue;
    }
    if (hasChild($c, "Altsig") == true){
        $altsigs=getChild($c, "Altsig");
        $altsig=$altsigs->nodeValue;
    }
    if (hasChild($c, "Lagerung") == true){
        $lagers=getChild($c, "Lagerung");
        $lager=$lagers->nodeValue;
    }
    if (!empty($sig)== true) {$count++;}
    if (!empty($bestsig)== true) {$count++;}
    if (!empty($altsig)== true) {$count++;}
    if (!empty($lager)== true) {$count++;}
    if ($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager)));}
}
if ($c->nodeName == "Film") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    $sig=""; $bestsig=""; $altsig=""; $lager="";
    if (hasChild($c, "Signatur") == true){
        $sigs=getChild($c, "Signatur");
        $sig=$sigs->nodeValue;
    }
    if (hasChild($c, "Bestellsig") == true){
        $bestsigs=getChild($c, "Bestellsig");
        $bestsig=$bestsigs->nodeValue;
    }
    if (hasChild($c, "Altsig") == true){
        $altsigs=getChild($c, "Altsig");
        $altsig=$altsigs->nodeValue;
    }
    if (hasChild($c, "Lagerung") == true){
        $lagers=getChild($c, "Lagerung");
        $lager=$lagers->nodeValue;
    }
    if (!empty($sig)== true) {$count++;}
    if (!empty($bestsig)== true) {$count++;}
    if (!empty($altsig)== true) {$count++;}
}

```



```

        if (!empty($lager)== true) {$count++;}
        if($count>1){$konmodel->append(
utf8_decode($lager));}
        if(tag_exists($c, "Film_Einheit") == true){
            $vorg=get_tags($c, "Film_Einheit");
            for($b=0; $b<$vorg->length;$b++) {
                $bestsig=""; $altsig=""; $lager="";
                $vor=$vorg->item($b);
                $id=$vor->getAttributeNode("idstef")->nodeValue;
                if(hasChild($vor, "Bestellsig") == true){
                    print "bestellsig";
                    $bestsigs=getChild($vor, "Bestellsig");
                    $bestsig=$bestsigs->nodeValue;
                }
                if(hasChild($vor, "Altsig") == true){
                    $altsigs=getChild($vor, "Altsig");
                    $altsig=$altsigs->nodeValue;
                }
                if(hasChild($vor, "Lagerung") == true){
                    $lagers=getChild($vor, "Lagerung");
                    $lager=$lagers->nodeValue;
                }
                if (!empty($sig)== true) {$count++;}
                if (!empty($bestsig)== true) {$count++;}
                if (!empty($altsig)== true) {$count++;}
                if (!empty($lager)== true) {$count++;}
                if($count>1){$konmodel->append(array($id, $sig, $bestsig, $altsig,
utf8_decode($lager));}
            }
        }
    }
}

if ($c->hasChildNodes()) {
    new_konkord($c, $konmodel);
}
}
}

function onHelp() {
    $dlg = new GtkAboutDialog();
    $dlg->set_position('GTK_WIN_POS_CENTER');

    $dlg->set_name('ASTER');
    $dlg->set_version('0.0.1');
    $dlg->set_comments('Verzeichnungstool');
    $dlg->set_copyright('No Copyright');
    $dlg->set_license("Hier kann das EULA eingesehen werden.\n"
        . "MfG S.J.");
    $pixbuf = GdkPixbuf::new_from_file('img/aster.png');
    $dlg->set_logo($pixbuf);
    $dlg->set_website('http://example.org!');
    $dlg->run();
}
}

```

```

function onQuit(){
    global $overviewdom;
    global $overviewfile;
    global $vedom;
    global $vefile;
    $overviewdom=load_xml($overviewfile);
    idattrs($overviewdom, 'ovidclear');
    $overviewdom->save($overviewfile);
    if (!empty($vedom)){idattrs($vedom, 'veidclear'); $vedom->save($vefile);}
    Gtk::main_quit();
}

function saveFile(){
    $dialog = new GtkFileChooserDialog('Datei speichern', NULL, Gtk::FILE_CHOOSER_ACTION_SAVE,
array(Gtk::STOCK_OK,Gtk::RESPONSE_OK, Gtk::STOCK_CANCEL, Gtk::RESPONSE_CANCEL));
    $response = $dialog->run();
    if ($response == Gtk::RESPONSE_OK){
        print "Save-OK button clicked, ";
        print $dialog->get_filename();
        print " selected.\n";
    }
    $dialog->destroy();
}

function selectOVTreeRow($selection) {
    global $overviewfile;
    global $overviewdom;
    global $vemodel;
    global $state;
    global $userid;
    $overviewdom=load_xml($overviewfile);
    if (!$overviewdom->schemaValidate("transform/AsterArchiv.xsd")){
        errorhandle('warning', 'Die Datei "'. $overviewfile. '"entspricht nicht dem Schema
("transform/AsterArchiv.xsd" ');
    } else {
        if(!empty($state)== true){
            $levels=$overviewdom->getElementById($state);
            $states=$levels->getAttributeNode('state');
            $states->nodeValue="open";
            $user=$levels->getAttributeNode('user');
            $user->nodeValue="";
            $state="";
            $overviewdom->save($overviewfile);
        }
    }
    list($model, $iter) = $selection->get_selected();
    if ($iter === null) {
        return;
    }
    $id = $model->get_value($iter, 2);
    $count=substr_count($id, '-');
}

```

```

if ($count>0){
    $pos=strrpos($id, '-');
    $posa=$pos+1;
    $beid=substr($id, $posa);
    $id=substr($id, 0, $pos);
} else {$beid='';}
$level=$overviewdom->getElementById($id);
cleartree($vemodel);
if ($level->getAttribute('state') == "locked" AND $level->getAttribute('user') != $userid) {
    $user=$level->getAttribute('user');
    mask_used($user);
} else {
    $states=$level->getAttributeNode('state');
    $states->nodeValue="locked";
    $user=$level->getAttributeNode('user');
    $user->nodeValue=$userid;
    $overviewdom->save($overviewfile);
    $state = $id;
    $name=$level->nodeName;
    if ($name== 'Archivinfo'){mask_archivinfo($id);}
    if ($name== 'Tektonik'){mask_tektonik($id);}
    if ($name== 'Bestand'){mask_bestand($id);}
    if ($name== 'Findmitteldatei'){
        $file=$level->nodeValue;
        model_ve($file);
        mask_findmittel($id, $beid);
    }
}
}

function selectVETreeRow($selection){
    list($model, $iter) = $selection->get_selected();
    if ($iter === null) {
        return;
    }
    $id = $model->get_value($iter, 2);
    global $vefile;
    global $vedom;
    if (!$vedom->schemaValidate("transform/AsterFindIntern.xsd")){
        errorhandle('warning', 'Die Datei "'. $vefile. "' entspricht nicht dem Schema
("transform/AsterFindIntern.xsd")');
    } else {getids($id);}
}

function set_model_iterbest($n, $iter) {
    global $bestmodel;
    global $vemodel;
    global $window;
    $home = $window->render_icon(Gtk::STOCK_HOME, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $directory = $window->render_icon(Gtk::STOCK_DIRECTORY, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $file = $window->render_icon(Gtk::STOCK_FILE, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $multi = $window->render_icon(Gtk::STOCK_DND_MULTIPLE, Gtk::ICON_SIZE_SMALL_TOOLBAR);
}

```

```

$it="";
$children=$n->childNodes;
$anz=$children->length;
for ($a=0; $a<$anz;$a++) {
    $c=$children->item($a);
    if ($c->nodeName == "Archivinfo") {
        $id=$c->getAttributeNode("id")->nodeValue;
        if (hasChild($c, 'Archivname')) {
            $child=getChild($c, 'Archivname');
            $value=$child->nodeValue;
            if (empty($value)){$value='Archivinfo';}
        } else {$value='Archivinfo';}
        $it=$bestmodel->append($iter, array($home, utf8_decode($value), $id));
    }
    if ($c->nodeName == "Tektonik") {
        $id=$c->getAttributeNode("id")->nodeValue;
        if (hasChild($c, 'Tekt_Nr')) {
            $nrs=getChild($c, 'Tekt_Nr');
            $nr=$nrs->nodeValue;
            if (!empty($nr)){$nr=$nr.' ';}
        } else {$nr=' ';}
        if (hasChild($c, 'Tekt_Titel')) {
            $titels=getChild($c, 'Tekt_Titel');
            $titel=$titels->nodeValue;
            if (empty($titel)){$titel='Tektonik';}
        } else {$titel='Tektonik';}
        $value=$nr.$titel;
        $it=$bestmodel->append($iter, array($directory, utf8_decode($value), $id));
        if (hasChild($c, 'Findmitteldatei')) {
            $dat=$c->childNodes;
            for ($b=0; $b<$dat->length;$b++) {
                $d=$dat->item($b);
                if ($d->nodeName=='Findmitteldatei'){
                    $idf=$d->getAttributeNode("id")->nodeValue;
                    $files=$d->nodeValue;
                    if (!empty($files)){
                        $iters= $bestmodel->append($it, array($multi, 'Findmittel',          $idf));
                        $dom=load_xml($files);
                        $infos=get_tags($dom, 'Bestand_Info');
                        $infozahl=$infos->length;
                        $count=1;
                        if ($infozahl>0){
                            for ($e=0; $e<$infozahl;$e++) {
                                $bestinfo=$infos->item($e);
                                if (hasChild($bestinfo, 'Bestand_Sig')) {
                                    $nrs=getChild($bestinfo, 'Bestand_Sig');
                                    $nr=$nrs->nodeValue;
                                    if (!empty($nr)){$nr=$nr.' ';}
                                } else {$nr=' ';}
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        if (hasChild($bestinfo, 'Bestandsname')) {
            $titels=getChild($bestinfo, 'Bestandsname');
            $titel=$titels->nodeValue;
            if (empty($titel)){ $titel='Bestand';}
        } else { $titel='Bestand';}
        $value=$nr.$titel;
        $ids=$idf.'-'. $count;
        $count++;
        $bestmodel->append($iters, array($file, $value, $ids));
    }
} else { $ids=$idf.'-'. $count;
    $bestmodel->append($iters, array($file, 'Bestand', $ids));
}
}
}
}
}
}

if ($c->nodeName == "Bestand") {
    $id=$c->getAttributeNode("id")->nodeValue;
    if (hasChild($c, 'Bestand_Sig')) {
        $nrs=getChild($c, 'Bestand_Sig');
        $nr=$nrs->nodeValue;
        if (!empty($nr)){ $nr=$nr.' ';}
    } else { $nr=' ';}
    if (hasChild($c, 'Bestandsname')) {
        $titels=getChild($c, 'Bestandsname');
        $titel=$titels->nodeValue;
        if (empty($titel)){ $titel='Bestand';}
    } else { $titel='Bestand';}
    $value=$nr.$titel;
    $it=$bestmodel->append($iter, array($file, utf8_decode($value), $id));
    if (hasChild($c, 'Findmitteldatei')) {
        $dat=$c->childNodes;
        $zahl=1;
        for ($b=0; $b<$dat->length;$b++) {
            $d=$dat->item($b);
            if ($d->nodeName=='Findmitteldatei'){
                $idf=$d->getAttributeNode("id")->nodeValue;
                $files=$d->nodeValue;
                if (!empty($files)){
                    $value='Findmittel '.$zahl;
                    $zahl++;
                    $bestmodel->append($it, array($multi, utf8_decode($value), $idf));
                }
            }
        }
    }
}
}
}
}
}
}

```

```

        if ($c->hasChildNodes()) {set_model_iterbest($c, $it);}
    }
}

function set_model_iterve($n, $iter) {
    global $vemodel;
    global $window;
    $home = $window->render_icon(Gtk::STOCK_HOME, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $file = $window->render_icon(Gtk::STOCK_FILE, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $directory = $window->render_icon(Gtk::STOCK_DIRECTORY, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $indexkon = $window->render_icon(Gtk::STOCK_INDEX, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $indexone = $window->render_icon(Gtk::STOCK_SELECT_FONT, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $multi = $window->render_icon(Gtk::STOCK_DND_MULTIPLE, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $justi = $window->render_icon(Gtk::STOCK_JUSTIFY_FILL, Gtk::ICON_SIZE_SMALL_TOOLBAR);
    $it="";
    $children=$n->childNodes;
    $anz=$children->length;
    for ($a=0; $a<$anz;$a++) {
        $c=$children->item($a);
        if ($c->nodeName == "Findmittel_Info") {
            $id=$c->getAttribute("idstef");
            $klass='Findmittel-Infos';
            $it=$vemodel->append($iter, array($home, utf8_decode($klass), $id));
        }
        if ($c->nodeName == "Einleitung") {
            $id=$c->getAttribute("idstef");
            $klass='Einleitung';
            $it=$vemodel->append($iter, array($justi, utf8_decode($klass), $id));
        }
        if ($c->nodeName == "Bestand_Info") {
            $id=$c->getAttribute("idstef");
            if (hasChild($c, 'Bestand_Sig')) {
                $child=getChild($c, 'Bestand_Sig');
                $sig=$child->nodeValue;
                if (!empty($sig)){$sig="($sig) ";}
            } else {$sig='';}
            if (hasChild($c, 'Bestandsname')) {
                $chld=getChild($c, 'Bestandsname');
                $name=$chld->nodeValue;
                if (empty($name)){$name='Bestandsangaben';}
            } else {$name='Bestandsangaben';}
            $klass=$sig.$name;
            $it=$vemodel->append($iter, array($file, utf8_decode($klass), $id));
        }
        if ($c->nodeName == "Klassifikation") {
            $id=$c->getAttributeNode("idstef")->nodeValue;
            $klass="Gliederungsstufe";
            if($c->firstChild->nodeName == "Klass_Nr"){
                $nrs=$c->firstChild;
                $nr=$nrs->nodeValue;
            }
        }
    }
}

```

```

    if($nrs->nextSibling->nodeName == "Klass_Titel"){
        $klassi=$nrs->nextSibling->nodeValue;
        if (!empty($klassi)){ $klass=$klassi;}
        $klass=$nr." ".$klass;
        $it=$vemodel->append($iter, array($directory, utf8_decode($klass), $id));
    }
    else{ $klass=$nr." ".$klass;
        $it=$vemodel->append($iter, array($directory, utf8_decode($klass), $id));
    }
}
elseif($c->firstChild->nodeName == "Klass_Titel"){
    $klassi=$c->firstChild->nodeValue;
    if (!empty($klassi)){ $klass=$klassi;}
    $it=$vemodel->append($iter, array($directory, utf8_decode($klass), $id));
}
else{ $it=$vemodel->append($iter, array($directory, utf8_decode($klass), $id));}
}
if ($c->nodeName == "Verzeichnungseinheiten") {
    $it=$iter;
}
if ($c->nodeName == "Sachakte") {
    $sig="";
    $tit="Sachakte";
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if (tag_exists($c, "Signatur")){
        $signatur=get_tags($c, "Signatur");
        $sigs=$signatur->item(0);
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){ $sig=" ".$sigi." ";}
    }
    if (tag_exists($c, "Titel")){
        $titel=get_tags($c, "Titel");
        $titels=$titel->item(0);
        $titeld=$titels->nodeValue;
        if (!empty($titeld)){ $tit=$titeld;}
    }
    $fulltit=$sig.$tit;
    $it=$vemodel->append($iter, array($file, utf8_decode($fulltit), $id));
}
if ($c->nodeName == "Fallakte") {
    $sig="";
    $tit="Fallakte";
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if (tag_exists($c, "Signatur")){
        $signatur=get_tags($c, "Signatur");
        $sigs=$signatur->item(0);
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){ $sig=" ".$sigi." ";}
    }
}

```

```

if (tag_exists($c, "Titel")){
    $titel=get_tags($c, "Titel");
    $titels=$titel->item(0);
    $titeld=$titels->nodeValue;
    if (!empty($titeld)){ $tit=$titeld;}
}
$fulltit=$sig.$tit;
$tit=$vemodel->append($iter, array($file, utf8_decode($fulltit), $id));
}
if ($c->nodeName == "Bandserie") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if($c->firstChild->nodeName == "Serientitel"){
        $titels=$c->firstChild;
        $titel=$titels->nodeValue;
        if (!empty($titel)){ $tit=$titel;}
        else { $tit="Bandserie";}
    } else { $tit="Bandserie";}
    $tit=$vemodel->append($iter, array($multi, utf8_decode($tit), $id));
}
if ($c->nodeName == "Grundbuch_akte") {
    $sig="";
    $tit="Grundbuch";
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if (tag_exists($c, "Signatur")){
        $signatur=get_tags($c, "Signatur");
        $sigs=$signatur->item(0);
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){ $sig="(".$sigi.") ";}
    }
    if (tag_exists($c, "Gemarkung")){
        $titel=get_tags($c, "Gemarkung");
        $titels=$titel->item(0);
        $titeld=$titels->nodeValue;
        if (!empty($titeld)){ $tit=$titeld;}
    }
    $fulltit=$sig.$tit;
    $tit=$vemodel->append($iter, array($file, utf8_decode($fulltit), $id));
}
if ($c->nodeName == "Urkunde") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    $titel="Urkunde";
    if($c->firstChild->nodeName == "Signatur"){
        $sigs=$c->firstChild;
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){ $sig="(".$sigi.") ";}
        else { $sig="";}
    }
    if($sigs->nextSibling->nodeName == "Nr"){
        $nr=$sigs->nextSibling->nodeValue;
        if (!empty($nr)){ $titel="Urk. ".$nr;}
    }
}

```



```

        $titel=$sig.$titel;
        $tit=$vemodel->append($iter, array($file, utf8_decode($titel), $id));
    } else{
        $titel=$sig.$klass;
        $tit=$vemodel->append($iter, array($file, utf8_decode($titel), $id));
    }
}
elseif($c->firstChild->nodeName == "Nr"){
    $nrs=$c->firstChild;
    $nr=$nrs->nodeValue;
    if (!empty($nr)){$titel="Urk. ".$nr;}
    $tit=$vemodel->append($iter, array($file, utf8_decode($titel), $id));
}
else {$tit=$vemodel->append($iter, array($file, utf8_decode($titel), $id));}
}
if ($c->nodeName == "Siegel" && $c->parentNode->nodeName == "Verzeichnungseinheiten") {
    $tit="Siegel";
    $sig="";
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if (tag_exists($c, "Signatur")){
        $signatur=get_tags($c, "Signatur");
        $sigs=$signatur->item(0);
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){$sig="(".$sigi.) ";}
    }
    if (tag_exists($c, "Nr")){
        $nr=get_tags($c, "Nr");
        $nrs=$nr->item(0);
        $nrd=$nrs->nodeValue;
        if (!empty($nrd)){$tit="Siegel Nr. ".$nrd;}
    }
    $fulltit=$sig.$tit;
    $tit=$vemodel->append($iter, array($file, utf8_decode($fulltit), $id));
}
if ($c->nodeName == "Karte") {
    $sig="";
    $tit="Karte";
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if (tag_exists($c, "Signatur")){
        $signatur=get_tags($c, "Signatur");
        $sigs=$signatur->item(0);
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){$sig="(".$sigi.) ";}
    }
    if (tag_exists($c, "Titel")){
        $titel=get_tags($c, "Titel");
        $titels=$titel->item(0);
        $titeld=$titels->nodeValue;
        if (!empty($titeld)){$tit=$titeld;}
    }
}

```

```

    }
    $fulltit=$sig.$tit;
    $it=$vemodel->append($iter, array($file, utf8_decode($fulltit), $id));
}
if ($c->nodeName == "Foto_Plakat") {
    $sig="";
    $tit="Foto/Plakat";
    $id=$c->getAttributeNode("idstef")->nodeValue;
    if (tag_exists($c, "Signatur")){
        $signatur=get_tags($c, "Signatur");
        $sigs=$signatur->item(0);
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){$sig="(".$sigi." ");}
    }
    if (tag_exists($c, "Titel")){
        $titel=get_tags($c, "Titel");
        $titels=$titel->item(0);
        $titeld=$titels->nodeValue;
        if (!empty($titeld)){$tit=$titeld;}
    }
    $fulltit=$sig.$tit;
    $it=$vemodel->append($iter, array($file, utf8_decode($fulltit), $id));
}
if ($c->nodeName == "Film") {
    $id=$c->getAttributeNode("idstef")->nodeValue;
    $titel="Film";
    $sig="";
    if($c->firstChild->nodeName == "Titel"){
        $tit=$c->firstChild;
        $titels=$tit->nodeValue;
        if (!empty($titels)){$titel=$titels;}
        if(hasChild($c, "Signatur") == true){
            $sigs=getChild($c, "Signatur");
            $sigi=$sigs->nodeValue;
            if (!empty($sigi)){$sig="(".$sigi." ");}
            $titel=$sig.$titel;
            $it=$vemodel->append($iter, array($file, utf8_decode($titel), $id));
        } else {$it=$vemodel->append($iter, array($file, utf8_decode($titel), $id));}
    }
    elseif($c->firstChild->nodeName == "Signatur"){
        $sigs=$c->firstChild;
        $sigi=$sigs->nodeValue;
        if (!empty($sigi)){$sig="(".$sigi." ");}
        $titel=$sig.$titel;
        $it=$vemodel->append($iter, array($file, utf8_decode($titel), $id));
    }
    else {$it=$vemodel->append($iter, array($file, utf8_decode($titel), $id));}
}
if ($c->nodeName == "Indizes") {

```

```

        $id=$c->getAttributeNode("idstef")->nodeValue;
        $it=$vemodel->append($iter, array($indexkon, utf8_decode("Indizes"), $id));
    }
    if ($c->nodeName == "Index") {
        $id=$c->getAttributeNode("idstef")->nodeValue;
        $indart=$c->getAttribute("Index_Art");
        $art=$c->getAttribute("Art");
        if (empty($indart)) {$value=$art;}
        else {$value=$indart;}
        if (empty($value)) {$value="Index";}
        $it=$vemodel->append($iter, array($indexone, utf8_decode($value), $id));
    }
    if ($c->nodeName == "Konkordanz") {
        $id=$c->getAttributeNode("idstef")->nodeValue;
        $it=$vemodel->append($iter, array($indexkon, utf8_decode("Konkordanzen"), $id));
    }
    if ($c->hasChildNodes()) {set_model_iterve($c, $it);}
}
}

function tag_exists($handle,$tagname) {
    $tag=$handle->getElementsByTagName($tagname);
    $nr=$tag->length;
    if ($nr>=1) {
        $exists=true;
    }
    else {
        $exists=false;
    }
    return $exists;
}
?>

```

C.6. masksarchiv.inc.php

```

<?php
function mask_archivinfo($id) {
    global $overviewdom;
    $ve=$overviewdom->getElementById($id);
    if (hasChild($ve, 'Archivname')==true) {$archivname=getChild($ve, 'Archivname');}
    else {$archivname='';}
    if (hasChild($ve, 'Anschrift')==true) {$anschrift=getChild($ve, 'Anschrift');}
    else {$anschrift='';}
    if (hasChild($ve, 'Telefon')==true) {$telefon=getChild($ve, 'Telefon');}
    else {$telefon='';}
    if (hasChild($ve, 'Fax')==true) {$fax=getChild($ve, 'Fax');}
    else {$fax='';}
    if (hasChild($ve, 'Email')==true) {$email=getChild($ve, 'Email');}
    else {$email='';}
    if (hasChild($ve, 'Homepage')==true) {$homepage=getChild($ve, 'Homepage');}
}

```

```

else {$homepage='';}
if (hasChild($ve, 'Land')==true){$land=getChild($ve, 'Land');}
else {$land='';}
if (hasChild($ve, 'Kompetenz')==true){$kompetenz=getChild($ve, 'Kompetenz');}
else {$kompetenz='';}
if (hasChild($ve, 'Gruendung')==true){$gruendung=getChild($ve, 'Gruendung');}
else {$gruendung='';}
if (hasChild($ve, 'Aufloesung')==true){$aufloesung=getChild($ve, 'Aufloesung');}
else {$aufloesung='';}
if (hasChild($ve, 'Bibliographie')==true){$biblio=getChild($ve, 'Bibliographie');}
else {$biblio='';}
if (hasChild($ve, 'archref')==true or hasChild($ve, 'bibref')==true or
hasChild($ve, 'fmref')==true or hasChild($ve, 'altuebform')==true){$idverweise=$id;}
else {$idverweise='';}
$plhbox1= new GtkHBox(false, 0);
$name=archivelementarchivname($archivname);
$framearchivname = createframe('Archivname', $name, 'V', true);
$gruendung=archivelementgruendung($gruendung);
$framegruendung = createframe('Gründung', $gruendung, 'V', true);
$aufloesung=archivelementaufloesung($aufloesung);
$frameaufloesung = createframe('Auflösung', $aufloesung, 'V', true);
$plhbox1->pack_start($framearchivname, true, true, 5);
$plhbox1->pack_start($framegruendung, true, true, 5);
$plhbox1->pack_start($frameaufloesung, true, true, 5);
$plhbox2= new GtkHBox(false, 0);
$anschrift=archivelementanschrift($anschrift);
$frameanschrift = createframe('Anschrift', $anschrift, 'V', true);
$land=archivelementland($land);
$frameland = createframe('Land', $land, 'V', true);
$homepage=archivelementhomepage($homepage);
$framehomepage = createframe('Homepage', $homepage, 'V', true);
$plhbox2->pack_start($frameanschrift, true, true, 5);
$plhbox2->pack_start($frameland, true, true, 5);
$plhbox2->pack_start($framehomepage, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$telefon=archivelementtelefon($telefon);
$frametelefon = createframe('Telefon', $telefon, 'V', true);
$fax=archivelementfax($fax);
$framefax = createframe('Fax', $fax, 'V', true);
$email=archivelementemail($email);
$frameemail = createframe('e-Mail', $email, 'V', true);
$plhbox3->pack_start($frametelefon, true, true, 5);
$plhbox3->pack_start($framefax, true, true, 5);
$plhbox3->pack_start($frameemail, true, true, 5);
$plhbox4= new GtkHBox(true, 0);
$kompetenz=archivelementkompetenz($kompetenz);
$framekompetenz = createframe('Kompetenzen', $kompetenz, 'V', true);
$plhbox4->pack_start($framekompetenz, true, true, 5);
$plhbox5= new GtkHBox(false, 0);
$biblio=archivelementbibliographie($biblio);

```

```

$verweis=archivelementverweise($idverweise);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$plhbox5->pack_start($biblio, true, true, 5);
$plhbox5->pack_start($frameverweis, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, false, false, 5);
$plvbox->pack_start($plhbox4, true, true, 5);
$plvbox->pack_start($plhbox5, true, true, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($plvbox, new GtkLabel('ArchivInfo'));
$note->show_all();
}

function mask_bestand($id) {
    global $overviewdom;
    $ve=$overviewdom->getElementById($id);
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->show_all();
}

function mask_findmittel($id, $bestandid) {
    global $vedom;
    $vedom->schemaValidate("transform/AsterFindIntern.xsd");
    if (empty($bestandid)){
        if (tag_exists($vedom, 'Findmittel_Info')){
            $fminfos=get_tags($vedom, 'Findmittel_Info');
            $fminfo=$fminfos->item(0);
            $idfm=$fminfo->getAttribute('idstef');
            getids($idfm);
        } else {mask_findinfo('',' ',' ',' ',' ',' ',' ',' ',' ');}
    } else {$bestandid=$bestandid-1;
        $besinfos=get_tags($vedom, 'Bestand_Info');
        $besinfo=$besinfos->item($bestandid);
        $idbes=$besinfo->getAttribute('idstef');
        getids($idbes);
    }
}

function mask_tektonik($id) {
    global $overviewdom;
    $ve=$overviewdom->getElementById($id);
    if (hasChild($ve, 'Tekt_Nr')==true){$tektnr=getChild($ve, 'Tekt_Nr');}
    else {$tektnr='';}
    if (hasChild($ve, 'Tekt_Titel')==true){$tekttitel=getChild($ve, 'Tekt_Titel');}
    else {$tekttitel='';}
}

```

```

if (hasChild($ve, 'Tekt_Beschreibung')==true) {$tektbesch=getChild($ve, 'Tekt_Beschreibung');}
else {$tektbesch='';}
if (hasChild($ve, 'Bem')==true) {$bems=get_tags($ve, 'Bem');}
else {$bems='';}
if (hasChild($ve, 'archref')==true or hasChild($ve, 'bibref')==true or
hasChild($ve, 'fmref')==true or hasChild($ve, 'altübform')==true) {$idverweise=$id;}
else {$idverweise='';}
if (hasChild($ve, 'Findmitteldatei')==true) {$dats=get_tags($ve, 'Findmitteldatei');}
else {$dats='';}
$plhbox1= new GtkHBox(false, 0);
$nr=archivelementtektnr($tektnr);
$framennr = createframe('Gliederungspunkt', $nr, 'HV', true);
$plhbox1->pack_start($framennr, false, false, 5);
$plhbox2= new GtkHBox(false, 0);
$titel=archivelementtekttitel($tekttitel);
$frametitl = createframe('Titel', $titel, 'HV', true);
$plhbox2->pack_start($frametitl, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$beschreibung=archivelementtektbeschreibung($tektbesch);
$framebeschreibung = createframe('Beschreibung', $beschreibung, 'HV', true);
$plhbox3->pack_start($framebeschreibung, true, true, 5);
$plhbox4= new GtkHBox(true, 0);
$findmitteldatei=archivelementfindmitteldatei($dats);
$framefindmittel = createframe('Findmittel', $findmitteldatei, 'HV', true);
$verweis=elementverweise($idverweise);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$plhbox4->pack_start($framefindmittel, true, true, 5);
$plhbox4->pack_start($frameverweis, true, true, 5);
$plhbox5= new GtkHBox(false, 0);
$bem=archivelementbem($bems);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$plhbox5->pack_start($framebem, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, true, true, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$plvbox->pack_start($plhbox5, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($plvbox, new GtkLabel('Tektonik'));
$note->show_all();
}

function mask_used($user) {
    $ovvbox = new GtkVBox(false, 8);
    $ovvbox->set_border_width(8);
    $label = new GtkLabel("\n\n\n\n\n\n\nBestand / Findmittel wird bereits von $user
bearbeitet.\n\n\nVersuchen Sie es später noch einmal");
    $ovvbox->pack_start($label, false, false, 0);
}

```

```

global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($ovvbox, new GtkLabel('In Benutzung'));
$note->show_all();
}
?>

```

C.7. maskfm.inc.php

```

<?php
function mask_band($id, $idsperrvermerk, $idserientitel, $idlz, $identhaelt, $idbem,
$idverweise) {
    $plhbox1= new GtkHBox(false, 0);
    $titel=elementserientitel($idserientitel);
    $frametitel = createframe('Titel', $titel, 'HV', true);
    $laufzeit=elementlaufzeit($idlz);
    $framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
    $plhbox1->pack_start($frametitel, true, true, 5);
    $plhbox1->pack_start($framelaufzeit, false, false, 5);
    $plhbox2= new GtkHBox(false, 0);
    $enthaelt=elemententhaelt($identhaelt);
    $frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);
    $plhbox2->pack_start($frameenthaelt, true, true, 5);
    $plhbox3= new GtkHBox(true, 0);
    $sperrvermerk=elementsperrvermerk($idsperrvermerk);
    $framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
    $verweis=elementverweise($idverweise);
    $frameverweis = createframe('Verweise', $verweis, 'V', true);
    $plhbox3->pack_start($framesperrvermerk, true, true, 5);
    $plhbox3->pack_start($frameverweis, true, true, 5);
    $plhbox4= new GtkHBox(false, 0);
    $bem=elementbem($idbem);
    $framebem = createframe('Bemerkungen', $bem, 'HV', true);
    $plhbox4->pack_start($framebem, true, true, 5);
    $plvbox = new GtkVBox(false, 8);
    $plvbox->set_name('Sachakte2');
    $plvbox->pack_start($plhbox1, false, false, 5);
    $plvbox->pack_start($plhbox2, true, true, 5);
    $plvbox->pack_start($plhbox3, true, true, 5);
    $plvbox->pack_start($plhbox4, false, false, 5);
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->append_page($plvbox, new GtkLabel('Bandserie'));
    $note->show_all();
}

function mask_bestandsinfo($id, $idbestname, $idbestsig, $idbestkurz, $idtext, $idlz, $idprov,
$idvorprov, $idabgstelle, $idakzession, $idsperrvermerk, $idbearb, $idbem, $idrecht,
$idumfang, $idlager, $idverweise) {

```

```

global $vedom;
$ve=$vedom->getElementById($id);
$plhbox1= new GtkHBox(true, 0);
$sig=elementbestandsig($idbestsig);
$framebestandsig = createframe('Bestandsignatur', $sig, 'V', true);
if (!empty($idbestkurz)){
    $kurzel=$vedom->getElementById($idbestkurz);
    $text=utf8_decode($kurzel->nodeValue);
} else {$text='';}
$hboxkurz=new GtkHbox(false,0);
$entry=new GtkEntry();
$entry->set_text($text);
$hboxkurz->pack_start($entry, true, true, 5);
$framebestandkurz = createframe('Bestandskürzel', $hboxkurz, 'V', true);
$umfang=elementumfang($idumfang);
$frameumfang = createframe('Umfang', $umfang, 'V', true);
$plhbox1->pack_start($framebestandsig, true, true, 5);
$plhbox1->pack_start($framebestandkurz, true, true, 5);
$plhbox1->pack_start($frameumfang, true, true, 5);
$plhbox2= new GtkHBox(false, 0);
$bestname=elementbestandsname($idbestname);
$framebestname = createframe('Bestandsname', $bestname, 'V', true);
$plhbox2->pack_start($framebestname, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$text=elementtext($idtext);
$plhbox3->pack_start($text, true, true, 5);
$plhbox4= new GtkHBox(false, 0);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 0);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$2hbox1= new GtkHBox(true, 0);
$recht=elementrechtsstatus($idrecht);
$framerecht = createframe('Rechtsstatus', $recht, 'V', true);
$lager=elementlagerung($idlager);
if (!empty($idlager)){
    $lagers=$vedom->getElementById($idlager);
    $text=utf8_decode($lagers->nodeValue);
} else {$text='';}
$hboxlag=new GtkHbox(false,0);
$entry=new GtkEntry();
$entry->set_text($text);

```



```

$ hboxlag->pack_start($entry, true, true, 5);
$ framelager = createframe('Lagerung', $ hboxlag, 'V', true);
$ p2hbox1->pack_start($ framerecht, true, true, 5);
$ p2hbox1->pack_start($ framelager, true, true, 5);
$ p2hbox2= new GtkHBox(true, 0);
$ provenienz=elementbestandprovenienz($idprov);
$ frameprovenienz = createframe('Provenienz', $provenienz, 'HV', true);
$ vorprov = elementvorprov($idvorprov, 'Bestand');
$ framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$ p2hbox2->pack_start($ frameprovenienz, true, true, 5);
$ p2hbox2->pack_start($ framevorprov, true, true, 5);
$ p2hbox3= new GtkHBox(true, 0);
$ akzession=elementbestandakzession($idakzession);
$ frameakzession = createframe('Akzession', $akzession, 'HV', true);
$ abgstelle=elementbestandabgstelle($idabgstelle);
$ frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'HV', true);
$ p2hbox3->pack_start($ frameakzession, true, true, 5);
$ p2hbox3->pack_start($ frameabgstelle, true, true, 5);
$ p2hbox4= new GtkHBox(false, 0);
$ verweis=elementverweise($idverweise);
$ frameverweis = createframe('Verweise', $verweis, 'V', true);
$ bearb=elementbearbeiter($idbearb);
$ framebearb = createframe('Bearbeiter', $bearb, 'HV', true);
$ p2hbox4->pack_start($ frameverweis, true, true, 5);
$ p2hbox4->pack_start($ framebearb, true, true, 5);
$ p2hbox5= new GtkHBox(false, 0);
$ bem=elementbem($idbem);
$ framebem = createframe('Bemerkungen', $bem, 'HV', true);
$ p2hbox5->pack_start($ framebem, true, true, 5);
$ p2vbox = new GtkVBox(false, 0);
$ p2vbox->set_name('Sachakte2');
$ p2vbox->pack_start($p2hbox1, false, false, 5);
$ p2vbox->pack_start($p2hbox2, true, true, 5);
$ p2vbox->pack_start($p2hbox3, true, true, 5);
$ p2vbox->pack_start($p2hbox4, true, true, 5);
$ p2vbox->pack_start($p2hbox5, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($p1vbox);
$note->append_page($p2vbox);
$note->show_all();
}

function mask_einleitung($id, $idtext, $idbem, $idbib, $idabk) {
    $hbox1= new GtkHBox(false, 0);
    $text=elementtext($idtext);
    $hbox1->pack_start($text, true, true, 5);
    $hbox2= new GtkHBox(true, 0);
    $bib=elementbibliographie($idbib);

```

```

$abk=elementabkuerzungen($idabk);
$frameabk = createframe('Abkürzungen', $abk, 'HV', true);
$hbox2->pack_start($bib, true, true, 5);
$hbox2->pack_start($frameabk, true, true, 5);
$hbox3= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$hbox3->pack_start($framebem, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->set_border_width(8);
$plvbox->pack_start($hbox1, true, true, 5);
$plvbox->pack_start($hbox2, true, true, 5);
$plvbox->pack_start($hbox3, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages(>0);
$note->append_page($plvbox);
$note->show_all();
}

function mask_fall($id) {
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages(>0);
    $note->show_all();
}

function mask_film($id) {
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages(>0);
    $note->show_all();
}

function mask_findinfo($id, $idfmname, $idfmsig, $idlz, $idbearb, $idbem, $idsperrvermerk,
$idhilf, $idverweise) {
    $plhbox1= new GtkHBox(false, 8);
    $fmsig=elementfmsig($idfmsig);
    $framefmsig = createframe('Findmittelsignatur', $fmsig, 'V', false);
    $fmname=elementfmname($idfmname);
    $framefmname = createframe('Findmittelname', $fmname, 'V', true);
    $plhbox1->pack_start($framefmsig, false, false, 5);
    $plhbox1->pack_start($framefmname, true, true, 5);
    $plhbox2= new GtkHBox(false, 0);
    $hilfe=elementhilfsfeld($idhilf);
    $framehilfe = createframe('Ergänzungen', $hilfe, 'HV', true);
    $plhbox2->pack_start($framehilfe, true, true, 5);
    $plhbox3= new GtkHBox(true, 8);
    $bearb=elementbearbeiter($idbearb);
    $framebearb = createframe('Bearbeiter', $bearb, 'HV', true);
    $verweis=elementverweise($idverweise);
    $frameverweis = createframe('Verweise', $verweis, 'V', true);

```

```

$plhbox3->pack_start($framebearb, true, true, 5);
$plhbox3->pack_start($frameverweis, true, true, 5);
$plhbox4= new GtkHBox(false, 8);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plhbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$plhbox5->pack_start($framebem, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$plvbox->pack_start($plhbox5, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($plvbox);
$note->show_all();
}

function mask_foto($id, $idsig, $idlz, $identhaelt, $idumfang, $idsperrvermerk, $idbestkurz,
$idbestellsig, $idlager, $idzustand, $idaltsig, $idvorprov, $idanlass, $idort, $idinst,
$idperson, $idbesch, $idmat, $identstufe, $idbem, $idformat, $idfarbe, $idabgstelle,
$idakzession, $idprov, $idtitel, $idverweise) {
    $plhbox1= new GtkHBox(true, 8);
    $sigpc=elementsignatur($idsig);
    $framesignatur = createframe('Signatur', $sigpc, 'V', true);
    $bestandkurz=elementbestandkurz($idbestkurz);
    $framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
    $umfang=elementumfang($idumfang);
    $frameumfang = createframe('Umfang', $umfang, 'V', true);
    $plhbox1->pack_start($framesignatur, true, true, 5);
    $plhbox1->pack_start($framebestandkurz, true, true, 5);
    $plhbox1->pack_start($frameumfang, true, true, 5);
    $plhbox2= new GtkHBox(false, 0);
    $titel=elementtitel($idtitel, 'multi');
    $plhbox2->pack_start($titel, true, true, 5);
    $plhbox3= new GtkHBox(false, 0);
    $enthaelt=elemententhaelt($identhaelt);
    $frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);
    $plhbox3->pack_start($frameenthaelt, true, true, 5);
    $plhbox4= new GtkHBox(false, 8);
    $laufzeit=elementlaufzeit($idlz);
    $framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
    $sperrvermerk=elementsperrvermerk($idsperrvermerk);

```

```

$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->set_name('Sachakte1');
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$p2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);
$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprov);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);
$lagerung=elementlagerung($idlager);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$p2hbox3= new GtkHBox(false, 0);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$p2hbox3->pack_start($framealtsig, true, true, 5);
$p2hbox3->pack_start($framevorprov, true, true, 5);
$p2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($idverweise);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$p2hbox4->pack_start($frameverweis, true, true, 5);
$p2hbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$p2hbox5->pack_start($framebem, true, true, 5);
$p2vbox = new GtkVBox(false, 8);
$p2vbox->set_name('Sachakte2');
$p2vbox->pack_start($p2hbox1, false, false, 5);
$p2vbox->pack_start($p2hbox2, false, false, 5);

```

```

$pvbox->pack_start($p2hbox3, true, true, 5);
$pvbox->pack_start($p2hbox4, false, false, 5);
$pvbox->pack_start($p2hbox5, false, false, 5);
$phbox1= new GtkHBox(true, 8);
$anlass=elementanlass($idanlass);
$frameanlass = createframe('Anlass', $anlass, 'V', true);
$phbox1->pack_start($frameanlass, true, true, 5);
$phbox2= new GtkHBox(true, 8);
$beschreibung=elementbeschreibung($idbesch);
$framebeschreibung = createframe('Beschreibung', $beschreibung, 'HV', true);
$person=elementperson($idperson);
$frameperson = createframe('Person', $person, 'HV', true);
$phbox2->pack_start($framebeschreibung, true, true, 5);
$phbox2->pack_start($frameperson, true, true, 5);
$phbox3= new GtkHBox(true, 8);
$institution=elementinstitution($idinst);
$frameinstitution = createframe('Institution', $institution, 'HV', true);
$ort=elementort($idort);
$frameort = createframe('Ort', $ort, 'HV', true);
$phbox3->pack_start($frameinstitution, true, true, 5);
$phbox3->pack_start($frameort, true, true, 5);
$phbox4= new GtkHBox(true, 8);
$format=elementformat($idformat);
$frameformat = createframe('Format', $format, 'V', true);
$farbe=elementfarbe($idfarbe);
$framefarbe = createframe('Farbe', $farbe, 'V', true);
$phbox4->pack_start($frameformat, true, true, 5);
$phbox4->pack_start($framefarbe, true, true, 5);
$phbox5= new GtkHBox(true, 8);
$material=elementmaterial($idmat);
$framematerial = createframe('Material', $material, 'V', true);
$entstufe=elemententstehungsstufe($identstufe);
$frameentstufe = createframe('Entstehungsstufe', $entstufe, 'V', true);
$phbox5->pack_start($framematerial, true, true, 5);
$phbox5->pack_start($frameentstufe, true, true, 5);
$pvbox = new GtkVBox(false, 8);
$pvbox->pack_start($phbox1, false, false, 5);
$pvbox->pack_start($phbox2, false, false, 5);
$pvbox->pack_start($phbox3, true, true, 5);
$pvbox->pack_start($phbox4, false, false, 5);
$pvbox->pack_start($phbox5, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($pvbox);
$note->append_page($p2vbox);
$note->append_page($pvbox);
$note->show_all();
}

```

```

function mask_grundbuch($id, $idsig, $idlz, $idgemark, $idband, $idumfang, $idsperrvermerk,
$idbestkurz, $idbestellsig, $idzustand, $idlager, $idaltsig, $idvorprov, $idbem, $idblatt,
$idseite, $ideigentum, $idanschrift, $idattgba, $idabgstelle, $idakzession, $idprov,
$idverweise) {
    $plhbox1= new GtkHBox(true, 8);
    $sigpc=elementsignatur($idsig);
    $framesignatur = createframe('Signatur', $sigpc, 'V', true);
    $gemark=elementgemarkung($idgemark);
    $framegemark = createframe('Gemarkung', $gemark, 'V', true);
    $band=elementband($idband);
    $frameband = createframe('Band', $band, 'V', true);
    $art=attributegrundbuch($idattgba);
    $plhbox1->pack_start($framesignatur, true, true, 5);
    $plhbox1->pack_start($framegemark, true, true, 5);
    $plhbox1->pack_start($frameband, true, true, 5);
    $plhbox2= new GtkHBox(false, 0);
    $bestandkurz=elementbestandkurz($idbestkurz);
    $framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
    $umfang=elementumfang($idumfang);
    $frameumfang = createframe('Umfang', $umfang, 'V', true);
    $frameart = createframe('Art', $art, 'V', true);
    $plhbox2->pack_start($framebestandkurz, true, true, 5);
    $plhbox2->pack_start($frameumfang, true, true, 5);
    $plhbox2->pack_start($frameart, true, true, 5);
    $plhbox3= new GtkHBox(false, 0);
    $seite=elementseite($idseite);
    $frameseite = createframe('Seite', $seite, 'V', true);
    $blatt=elementblatt($idblatt);
    $frameblatt = createframe('Blatt', $blatt, 'V', true);
    $plhbox3->pack_start($frameseite, true, true, 5);
    $plhbox3->pack_start($frameblatt, true, true, 5);
    $plhbox4= new GtkHBox(false, 0);
    $anschrift=elementanschrift($idanschrift);
    $frameanschrift = createframe('Anschriften', $anschrift, 'HV', true);
    $eigentum=elementeigentuemmer($ideigentum);
    $frameeigentum = createframe('Eigentümer', $eigentum, 'HV', true);
    $plhbox4->pack_start($frameanschrift, true, true, 5);
    $plhbox4->pack_start($frameeigentum, true, true, 5);
    $plhbox5= new GtkHBox(false, 8);
    $laufzeit=elementlaufzeit($idlz);
    $framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
    $sperrvermerk=elementsperrvermerk($idsperrvermerk);
    $framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
    $plhbox5->pack_start($framelaufzeit, false, false, 5);
    $plhbox5->pack_start($framesperrvermerk, true, true, 5);
    $plvbox = new GtkVBox(false, 8);
    $plvbox->pack_start($plhbox1, false, false, 5);
    $plvbox->pack_start($plhbox2, false, false, 5);
    $plvbox->pack_start($plhbox3, false, false, 5);
    $plvbox->pack_start($plhbox4, true, true, 5);

```

```

$plvbox->pack_start($plhbox5, false, false, 5);
$p2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);
$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprov);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);
$lagerung=elementlagerung($idlager);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$p2hbox3= new GtkHBox(false, 0);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$p2hbox3->pack_start($framealtsig, true, true, 5);
$p2hbox3->pack_start($framevorprov, true, true, 5);
$p2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($idverweise);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$p2hbox4->pack_start($frameverweis, true, true, 5);
$p2hbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$p2hbox5->pack_start($framebem, true, true, 5);
$p2vbox = new GtkVBox(false, 8);
$p2vbox->pack_start($p2hbox1, false, false, 5);
$p2vbox->pack_start($p2hbox2, false, false, 5);
$p2vbox->pack_start($p2hbox3, true, true, 5);
$p2vbox->pack_start($p2hbox4, false, false, 5);
$p2vbox->pack_start($p2hbox5, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($plvbox);
$note->append_page($p2vbox);
$note->show_all();
}

```

```

function mask_indizes($id, $idbem) {
    $bem=elementbem($idbem);
    $framebem = createframe('Bemerkungen', $bem, 'HV', true);
    $butbox = new GtkHButtonBox();
    $plusbutton=new GtkButton('');
    $plusbutton->set_name('vorggplus');
    $image = new GtkImage();
    $image->set_from_file('img/stock_add_24.png');
    $plusbutton->set_image($image);
    $plusbutton->set_size_request(33, 25);
    $plusbutton->connect_simple('clicked','addind');
    $butbox->add($plusbutton);
    $minusbutton=new GtkButton('');
    $minusbutton->set_name('vorgminus');
    $image = new GtkImage();
    $image->set_from_file('img/stock_remove_24.png');
    $minusbutton->set_image($image);
    $minusbutton->set_size_request(33, 25);
    $minusbutton->connect_simple('clicked','errorhandle', 'warning', "Kein unterstütztes Dateiformat");
    $butbox->add($minusbutton);
    $vbox = new GtkVBox(false, 8);
    $vbox->set_name('indizes');
    $vbox->pack_start($framebem, true, true, 5);
    $vbox->pack_end($butbox, false, false, 5);
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->append_page($vbox, new GtkLabel('Indizes'));
    $note->show_all();
}

function mask_index($id) {
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->show_all();
}

function mask_karte($id, $idsig, $idlz, $idbesch, $idumfang, $idsperrvermerk, $idbestkurz,
$idbestellsig, $idlager, $idzustand, $idaltsig, $idvorprov, $identhaelt, $idort, $idperson,
$idinst, $idformat, $idmat, $identstufe, $idauflage, $idausf, $idbem, $idaz, $ideinz, $idktyp,
$idnebk, $idmasss, $idtopogr, $idabgstelle, $idakzession, $idprov, $idtitel, $idverweise) {
    $plhbox1= new GtkHBox(true, 8);
    $sigpc=elementsignatur($idsig);
    $framesignatur = createframe('Signatur', $sigpc, 'V', true);
    $bestandkurz=elementbestandkurz($idbestkurz);
    $framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
    $umfang=elementumfang($idumfang);
    $frameumfang = createframe('Umfang', $umfang, 'V', true);
    $plhbox1->pack_start($framesignatur, true, true, 5);

```



```

$plhbox1->pack_start($framebestandkurz, true, true, 5);
$plhbox1->pack_start($frameumfang, true, true, 5);
$plhbox2= new GtkHBox(false, 0);
$titel=elementtitel($idtitel, 'multi');
$plhbox2->pack_start($titel, true, true, 5);
$plhbox3= new GtkHBox(false, 0);
$enthaelt=elemententhaelt($identhaelt);
$frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);
$plhbox3->pack_start($frameenthaelt, true, true, 5);
$plhbox4= new GtkHBox(false, 8);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$p2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);
$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprov);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);
$lagerung=elementlagerung($idlager);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$nebenkarten=elementnebenkarten($idnebk);
$framenebenkarten = createframe('Nebenkarten', $nebenkarten, 'HV', true);
$p2hbox3= new GtkHBox(false, 0);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);

```

```

$2hbox3->pack_start($framealtsig, true, true, 5);
$2hbox3->pack_start($framenebenkarten, true, true, 5);
$2hbox3->pack_start($framevorprov, true, true, 5);
$2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($idverweise);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$az=elementaz($idaz);
$frameaz = createframe('Aktenzeichen', $az, 'HV', true);
$2hbox4->pack_start($frameverweis, true, true, 5);
$2hbox4->pack_start($frameaz, true, true, 5);
$2hbox5= new GtkHBox(false, 0);
$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$2hbox5->pack_start($framebem, true, true, 5);
$2vbox = new GtkVBox(false, 8);
$2vbox->pack_start($2hbox1, false, false, 5);
$2vbox->pack_start($2hbox2, false, false, 5);
$2vbox->pack_start($2hbox3, true, true, 5);
$2vbox->pack_start($2hbox4, false, false, 5);
$2vbox->pack_start($2hbox5, false, false, 5);
$3hbox1= new GtkHBox(true, 8);
$ktyp=elementkartentyp($idktyp);
$framektyp = createframe('Kartentyp', $ktyp, 'V', true);
$ausf=elementausfuehrung($idausf);
$frameausf = createframe('Ausführung', $ausf, 'V', true);
$3hbox1->pack_start($framektyp, true, true, 5);
$3hbox1->pack_start($frameausf, true, true, 5);
$3hbox2= new GtkHBox(true, 8);
$beschreibung=elementbeschreibung($idbesch);
$framebeschreibung = createframe('Beschreibung', $beschreibung, 'HV', true);
$seinz=elementeinzeichnung($ideinz);
$frameeinz = createframe('Einzeichnung', $seinz, 'HV', true);
$person=elementperson($idperson);
$frameperson = createframe('Person', $person, 'HV', true);
$3hbox2->pack_start($framebeschreibung, true, true, 5);
$3hbox2->pack_start($frameeinz, true, true, 5);
$3hbox2->pack_start($frameperson, true, true, 5);
$3hbox3= new GtkHBox(true, 8);
$institution=elementinstitution($idinst);
$frameinstitution = createframe('Institution', $institution, 'HV', true);
$ort=elementort($idort);
$frameort = createframe('Ort', $ort, 'HV', true);
$topogr=elementtopogrdaten($idtopogr);
$frametopogr = createframe('Topogr. Daten', $topogr, 'HV', true);
$3hbox3->pack_start($frameinstitution, true, true, 5);
$3hbox3->pack_start($frameort, true, true, 5);
$3hbox3->pack_start($frametopogr, true, true, 5);
$3hbox4= new GtkHBox(true, 8);
$format=elementformat($idformat);

```

```

$frameformat = createframe('Format', $format, 'V', true);
$auflage=elementauflage($idaufgabe);
$frameauflage = createframe('Auflage', $auflage, 'V', true);
$massstab=elementmassstab($idmasss);
$framemasstab = createframe('Massstab', $massstab, 'V', true);
$phbox4->pack_start($frameformat, true, true, 5);
$phbox4->pack_start($frameauflage, true, true, 5);
$phbox4->pack_start($framemasstab, true, true, 5);
$phbox5= new GtkHBox(true, 8);
$material=elementmaterial($idmat);
$framematerial = createframe('Material', $material, 'V', true);
$entstufe=elemententstehungsstufe($identstufe);
$frameentstufe = createframe('Entstehungsstufe', $entstufe, 'V', true);
$phbox5->pack_start($framematerial, true, true, 5);
$phbox5->pack_start($frameentstufe, true, true, 5);
$pvbox = new GtkVBox(false, 8);
$pvbox->pack_start($phbox1, false, false, 5);
$pvbox->pack_start($phbox2, false, false, 5);
$pvbox->pack_start($phbox3, true, true, 5);
$pvbox->pack_start($phbox4, false, false, 5);
$pvbox->pack_start($phbox5, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($pvbox);
$note->append_page($pvbox);
$note->append_page($pvbox);
$note->show_all();
}

function mask_klassi($id, $idsperrvermerk, $idklassnr, $idklasstitel, $idbem, $idverweise) {
    $phbox1= new GtkHBox(false, 0);
    $nr=elementklassnr($idklassnr);
    $framenr = createframe('Gliederungspunkt', $nr, 'HV', true);
    $phbox1->pack_start($framenr, false, false, 5);
    $phbox2= new GtkHBox(false, 0);
    $titel=elementklasstitel($idklasstitel);
    $frametitel = createframe('Titel', $titel, 'HV', true);
    $phbox2->pack_start($frametitel, true, true, 5);
    $phbox3= new GtkHBox(true, 0);
    $sperrvermerk=elementsperrvermerk($idsperrvermerk);
    $framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
    $verweis=elementverweise($idverweise);
    $frameverweis = createframe('Verweise', $verweis, 'V', true);
    $phbox3->pack_start($framesperrvermerk, true, true, 5);
    $phbox3->pack_start($frameverweis, true, true, 5);
    $phbox4= new GtkHBox(false, 0);
    $bem=elementbem($idbem);
    $framebem = createframe('Bemerkungen', $bem, 'HV', true);
    $phbox4->pack_start($framebem, true, true, 5);
}

```

```

$plvbox = new GtkVBox(false, 8);
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, true, true, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($plvbox, new GtkLabel('Klassifikation'));
$note->show_all();
}

function mask_konkordanz($id, $idbem) {
    $plhbox1= new GtkHBox(false, 0);
    $bem=elementbem($idbem);
    $framebem = createframe('Bemerkungen', $bem, 'HV', true);
    $plhbox1->pack_start($framebem, true, true, 5);
    $plhbox2= new GtkHBox(false, 0);
    $konkord=elementkonkordanz($id);
    $framekonkord = createframe('Konkordanzen', $konkord, 'HV', true);
    $plhbox2->pack_start($framekonkord, true, true, 5);
    $vbox = new GtkVBox(false, 8);
    $vbox->pack_start($plhbox1, false, false, 5);
    $vbox->pack_start($plhbox2, true, true, 5);
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->append_page($vbox, new GtkLabel('Konkordanzen'));
    $note->show_all();
}

function mask_sach($id, $idsig, $idlz, $idtitel, $identhaelt, $idumfang, $idsperrvermerk,
$idbestkurz, $idbestellsig, $idlager, $idzustand, $idaltsig, $idvorprov, $vorgid, $idaz,
$idbem, $idabgstelle, $idakzession, $idprov, $idverweise) {
    $plhbox1= new GtkHBox(true, 8);
    $sigpc=elementsignatur($idsig);
    $framesignatur = createframe('Signatur', $sigpc, 'V', true);
    $bestandkurz=elementbestandkurz($idbestkurz);
    $framebestandkurz = createframe('Bestandskürzel', $bestandkurz, 'V', true);
    $umfang=elementumfang($idumfang);
    $frameumfang = createframe('Umfang', $umfang, 'V', true);
    $plhbox1->pack_start($framesignatur, true, true, 5);
    $plhbox1->pack_start($framebestandkurz, true, true, 5);
    $plhbox1->pack_start($frameumfang, true, true, 5);
    $plhbox2= new GtkHBox(false, 0);
    $titel=elementtitel($idtitel, 'single');
    $frametitel = createframe('Titel', $titel, 'V', true);
    $plhbox2->pack_start($frametitel, true, true, 5);
    $plhbox3= new GtkHBox(false, 0);
    $enthaelt=elemententhaelt($identhaelt);
    $frameenthaelt = createframe('Enthält', $enthaelt, 'HV', true);

```

```

$plhbox3->pack_start($frameenthalt, true, true, 5);
$plhbox4= new GtkHBox(false, 8);
$laufzeit=elementlaufzeit($idlz);
$framelaufzeit = createframe('Laufzeit', $laufzeit, 'H', true);
$sperrvermerk=elementsperrvermerk($idsperrvermerk);
$framesperrvermerk = createframe('Sperrvermerk', $sperrvermerk, 'HV', true);
$plhbox4->pack_start($framelaufzeit, false, false, 5);
$plhbox4->pack_start($framesperrvermerk, true, true, 5);
$plvbox = new GtkVBox(false, 8);
$plvbox->set_name('Sachakte1');
$plvbox->pack_start($plhbox1, false, false, 5);
$plvbox->pack_start($plhbox2, false, false, 5);
$plvbox->pack_start($plhbox3, true, true, 5);
$plvbox->pack_start($plhbox4, false, false, 5);
$p2hbox1= new GtkHBox(true, 8);
$bestellsig=elementbestellsig($idbestellsig);
$framebestellsig = createframe('Bestellsignatur', $bestellsig, 'V', true);
$abgstelle=elementabgstelle($idabgstelle);
$frameabgstelle = createframe('Abg. Stelle', $abgstelle, 'V', true);
$provenienz=elementprovenienz($idprov);
$frameprovenienz = createframe('Provenienz', $provenienz, 'V', true);
$p2hbox1->pack_start($framebestellsig, true, true, 5);
$p2hbox1->pack_start($frameprovenienz, true, true, 5);
$p2hbox1->pack_start($frameabgstelle, true, true, 5);
$p2hbox2= new GtkHBox(true, 8);
$akzession=elementakzession($idakzession);
$frameakzession = createframe('Akzession', $akzession, 'V', true);
$lagerung=elementlagerung($idlager);
$framelagerung = createframe('Lagerung', $lagerung, 'V', true);
$zustand=elementzustand($idzustand);
$framezustand = createframe('Zustand', $zustand, 'V', true);
$p2hbox2->pack_start($frameakzession, true, true, 5);
$p2hbox2->pack_start($framelagerung, true, true, 5);
$p2hbox2->pack_start($framezustand, true, true, 5);
$p2hbox3= new GtkHBox(false, 0);
$altsig=elementaltsig($idaltsig);
$framealtsig = createframe('Altsignaturen', $altsig, 'HV', true);
$vorprov = elementvorprov($idvorprov, 'anderes');
$framevorprov = createframe('Vorprovenienzen', $vorprov, 'HV', true);
$p2hbox3->pack_start($framealtsig, true, true, 5);
$p2hbox3->pack_start($framevorprov, true, true, 5);
$p2hbox4= new GtkHBox(false, 0);
$verweis=elementverweise($idverweise);
$frameverweis = createframe('Verweise', $verweis, 'V', true);
$saz=elementaz($idaz);
$frameaz = createframe('Aktenzeichen', $saz, 'HV', true);
$p2hbox4->pack_start($frameverweis, true, true, 5);
$p2hbox4->pack_start($frameaz, true, true, 5);
$p2hbox5= new GtkHBox(false, 0);

```

```

$bem=elementbem($idbem);
$framebem = createframe('Bemerkungen', $bem, 'HV', true);
$p2hbox5->pack_start($framebem, true, true, 5);
$p2vbox = new GtkVBox(false, 8);
$p2vbox->set_name('Sachakte2');
$p2vbox->pack_start($p2hbox1, false, false, 5);
$p2vbox->pack_start($p2hbox2, false, false, 5);
$p2vbox->pack_start($p2hbox3, true, true, 5);
$p2vbox->pack_start($p2hbox4, false, false, 5);
$p2vbox->pack_start($p2hbox5, false, false, 5);
$vorgvbox = new GtkVBox(false, 0);
$vorgaenge=elementvorgaenge($vorgid);
$vorgvbox->pack_start($vorgaenge, false, false, 5);
$vorgbutbox = new GtkHButtonBox();
$plusbutton=new GtkButton('');
$plusbutton->set_name('vorgplus');
$image = new GtkImage();
$image->set_from_file('img/stock_add_24.png');
$plusbutton->set_image($image);
$plusbutton->set_size_request(33, 25);
$vorgbutbox->add($plusbutton);
$minusbutton=new GtkButton('');
$minusbutton->set_name('vorgminus');
$image = new GtkImage();
$image->set_from_file('img/stock_remove_24.png');
$minusbutton->set_image($image);
$minusbutton->set_size_request(33, 25);
$vorgbutbox->add($minusbutton);
$vorgvbox->pack_end($vorgbutbox, false, false, 5);
global $note;
do {$note->remove_page($note->get_current_page());}
while ($note->get_n_pages()>0);
$note->append_page($p1vbox);
$note->append_page($p2vbox);
$note->append_page($vorgvbox, new GtkLabel('Vorgänge'));
$note->show_all();
}

function mask_siegel($id) {
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->show_all();
}

function mask_urkunde($id) {
    global $note;
    do {$note->remove_page($note->get_current_page());}
    while ($note->get_n_pages()>0);
    $note->show_all();
}

```

?>