

zpub - Redakteurhandbuch

Thomas Breitner

zpub - Redakteurhandbuch

von Thomas Breitner

\$Revision\$

Veröffentlicht 2013

Copyright © 2013, 2010, 2009 zpub.de | Thomas und Joachim Breitner

Inhaltsverzeichnis

Über zpub, das zentrale Dokumentationssystem	viii
Systemvoraussetzungen	x
1. Installation der Client-Anwendungen	1
Liste der Downloadlinks:	1
Installation Subversion Client	1
Installation Serna Free XML-Editor	2
2. Verwaltung der Dokumente	3
Das Webinterface	3
Aufbau Webinterface	4
Freigabe von Dokumenten	5
3. Nutzung von Subversion	6
Kurzdarstellung eines kompletten Editiervorgangs	6
Auschecken des SVN-Repositories	7
Daten aktualisieren und übertragen	10
SVN Konflikte	11
Konflikte manuell lösen	12
4. Editieren von Dokumenten	15
Neues Dokument erstellen	15
Kopieren einer Vorlage als neues Dokument	15
Dokument löschen	17
DocBook-Struktur	17
Arbeiten mit Serna Free	18
Praktisches	19
Arbeiten mit Elementen	20
Markup <i>escapen</i>	21
IDS	21
Kommentare	21
Validieren	22
Wiederverwendbarer Inhalt	22
5. Ausgabeformate	23
PDF	23
PDF Metadaten	23
Landscape figure	23
Landscape table	24
A. DocBook-Elemente	27
application Programmname	28
callout Annotationen	28
figure Grafikeinbindung	28
Einbindung via figure	31
Einbindung via figure mit Untertitel (caption)	31
Einbindung via screenshot	32
footnote Fußnote	32
glossary Glossar	32
indexterm Indexeintrag	33
inlinemediaobject Grafikeinbindung (inline)	34
itemizedlist Ungeordnete Liste	34
keycap/keysym Tastensymbole	34
link Allgemeiner Link	35
menuchoiceMenüstrukturen	35
note Anmerkungen	35
orderedlist Geordnete Liste	36
table Tabellen	36
tip Tipps	37
trademark Produktname	37
ulink Externer Link	37

warning Warnungen	37
xref Interner Link	38
B. FAQ / Troubleshooting	39
C. XMLmind XML-Editor	41
Glossar	42
Stichwortverzeichnis	44

Abbildungsverzeichnis

1. zpub-Aufbau	viii
3.1. Screenshot: SVN auschecken...	7
3.2. Screenshot: Projektarchiv	7
3.3. Screenshot: SVN Zertifikatsabfrage	8
3.4. Screenshot: SVN Anmeldedaten	8
3.5. Screenshot: SVN ausgecheckt	8
3.6. SVN-Symbol für aktuellen Ordner	9
3.7. SVN - Symbol für aktuelle Datei	9
3.8. TortoiseSVN Kontextmenü - Aktualisieren	10
3.9. SVN-Übertragungsmeldung	11
3.10. zpub: Archiv	11
3.11. SVN-Meldung: Datei veraltet	13
3.12. SVN-Meldung: Konflikt	13
3.13. TortoiseMerge: Konflikte manuell auflösen	14
4.1. Versionierte Dateien exportieren	16
4.2. SVN-Eigenschaften importieren	17
4.3. DocBook-Baumansicht	18
4.4. Element-Auswahl choice/choose	19
4.5. Aktueller Knoten	19
4.6. Dialog Insert Element	20
5.1. Landscape figure	24
A.1. Screenshot (unskaliert: 698x406px, 96dpi)	29
A.2. Screenshot (unskaliert: 698x406px,192dpi)	30
A.3. Screenshot (kein ClearType; unskaliert: 698x406px,192dpi)	30
A.4. Pixelleiste (unskaliert)	31
A.5. Pixelleiste (Breite: 80%)	31
A.6. Pixelleiste (Breite: 5cm)	31
A.7. Abbildung figure	31
A.8. Beispielabbildung mit Untertitel figure	31
A.9. Screenshot-Titel	32

Tabellenverzeichnis

5.1. Landscape table	26
A.1. Tastensymbole	34
A.3. Beispieltabelle mit alternierenden Zeilehintergrundfarben	36
A.4. CALS-Tabelle	37

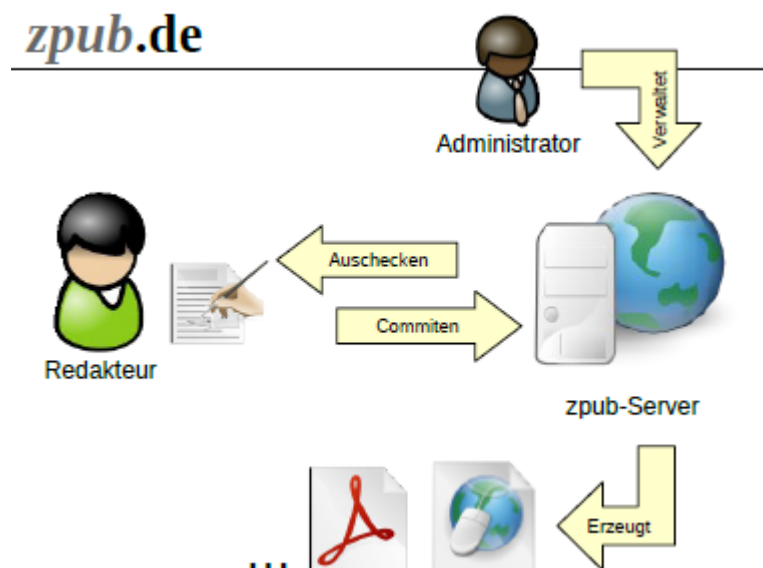
Liste der Beispiele

4.1. Verlinkung von Elementen	21
A.1. ID-Vergabe/Verlinkung von Elementen	28
A.2. Programmnamen	28
A.3. Skalierungsbeispiele für Abbildungen	31
A.4. Abbildung ohne Untertitel	31
A.5. Abbildung mit Untertitel (<i>caption</i>)	31
A.6. Screenshot	32
A.7. Fußnoten-Beispiel	32
A.8. Quelltext-Beispiel: Glossar	33
A.9. <i>indexterm</i> Textbeispiel	33
A.10. <i>Indexterm</i> Quellcodebeispiel	33
A.11. <i>inlinemediaobject</i>	34
A.12. <i>menuchoice</i> und <i>shortcut</i>	35
A.13. Produktbezeichnungen	37
A.14. <i>xref</i> -Vergabe/Verlinkung von Elementen	38

Über zpub, das zentrale Dokumentationssystem

zpub bietet ein Redaktions- und Publikationssystem auf der Basis von DocBook-XML.

Abbildung 1. zpub-Aufbau



zpub bietet im Einzelnen:

- zentrale Dokument-Erstellung und Pflege
- zpub benötigt auf Clientseite lediglich SVN-Zugriff und einen geeigneten Editor für DocBook-XML-Dateien
- strikte Trennung von Inhalt [DocBook-XML] und Präsentation
- automatische Generierung verschiedener Ausgabeformate:
 - xhtml für Ihren Internet-Auftritt
 - pdf für druckfertige Handbücher
 - chm zur Integration in Ihre Windows-Anwendungen
 - rtf zur einfachen Weiterverarbeitung in anderen Anwendung
- jederzeit die aktuelle Versionen in allen Ausgabeformaten verfügbar – sowie alle vorherigen Versionen
- Versionskontrolle via SVN
- automatische Benachrichtigung von Gruppenmitgliedern
- zpub setzt auf anerkannten Open Source-Projekten wie fop, xslt, svn, apache etc. auf

zpub hilft, sich auf den Inhalt zu konzentrieren.

Freie Software. zpub ist **Freie Software** und kann kostenlos heruntergeladen, eingesetzt und verändert werden.

Lizenz. zpub.de ist Freie Software und wird unter der European Union Public License v.1.1 [<https://joinup.ec.europa.eu/software/page/eupl/licence-eupl>] vertrieben.

Entwicklungsversion. Den aktuellsten Quellcode von zpub.de finden Sie im zpub-Git-Repository auf GitHub [<http://github.com/nomeata/zpub>].

Wir bieten Ihnen dazu eine Reihe von Dienstleistungen an:

- Support: Wir helfen Ihnen, zpub erfolgreich auf- und einzusetzen.
- Entwicklung: Ihnen fehlt bei zpub noch ein Feature? Wir entwickeln es für Sie.
- Hosting: Wir betreiben Ihre zpub-Instanz auf unseren Servern. So können Sie ohne große Investitionen und Verpflichtungen günstig in zpub einsteigen.

Systemvoraussetzungen

Auf Redakteursseite wird lediglich eine SVN-Client-Anwendung sowie ein beliebiger Editor für Text- bzw. XML-Dateien benötigt.

Die Benutzerkennungen werden vom DEMO-zpub-Administrator erstellt.

Tipp

Dieses Handbuch geht von einer Installation des Subversion-Clients TortoiseSVN [<http://tortoisesvn.tigris.org/>] sowie eines UTF-8-fähigen Texteditors bzw. des XML-Editors Serna Free [<http://sourceforge.net/projects/sernafree.mirror/>] aus und nutzt diese für Beschreibungen und Abbildungen. Die Abbildungen beziehen sich auf eine jeweils aktuelle Version der Softwarekomponenten. Zukünftige Softwareversionen können daher unterschiedliche Menüs und Designs haben. Beide genannten Softwareprojekte werden aktiv gepflegt, es sollten daher immer die aktuellsten Versionen genutzt werden.

Je nach persönlich präferierter Arbeitsumgebung des Redakteurs kann selbstverständlich auch eine andere Softwareausstattung genutzt werden.

Anmerkung

Für das Editieren von DocBook-XML-Dokumenten hat sich der inzwischen als Open Source-freigegebene XML-Editor Serna Free [<http://sourceforge.net/projects/sernafree.mirror/>] bewährt. Dieser bietet eine übersichtliche Bedienoberfläche sowie komfortables Editieren.

Warnung

Das Webfrontend wie auch der SVN-Zugriff nutzen eine SSL-verschlüsselte Verbindung. Diese ist mit einem eigenen Zertifikat, welches von CA-Cert signiert ist, gesichert. Sollte das Root-Zertifikat von CA-Cert nicht auf dem Rechner des Benutzers installiert sein, so erscheint beim Zugriff auf die Webseiten des Frontends sowie bei dem Zugriff auf das SVN-Repository eine Fehlermeldung, die auf ein unbekanntes Zertifikat hinweist. In beiden Fällen soll das Zertifikat dauerhaft akzeptiert werden, um sicher und komfortabel mit dem System zu arbeiten.

Im folgenden wird davon ausgegangen, dass auf Redakteursseite das Betriebssystem Windows XP zum Einsatz kommt. Sämtliche Komponenten sind auch für andere Betriebssysteme- und Versionen verfügbar.

Kapitel 1. Installation der Client-Anwendungen

Für dieses Handbuch wird clientseitig von der Installation des TortoiseSVN-Clients sowie des XML-Editors Serna Free [<http://sourceforge.net/projects/sernafree.mirror/>] ausgegangen.

Liste der Downloadlinks:

- Serna Free [<http://sourceforge.net/projects/sernafree.mirror/>]
- TortoiseSVN [<http://tortoisesvn.net/downloads.html>] (Installation erfordert Neustart)
- TortoiseSVN (dt. Sprachpaket ("*Language Pack*")) [<http://tortoisesvn.net/downloads.html>] (nach obigem Neustart zu installieren, optional)
- Notepad++ [<http://notepad-plus-plus.org/download>] (*Texteditor, optional*)

Anmerkung

Es sollten immer die aktuellsten Versionen der Softwarepakete genutzt werden.

Sämtliche Downloads verlinken auf Versionen für das Betriebssystem Windows™ in der 32bit-Variante. Versionen für andere Betriebssysteme bzw. 64bit-Versionen existieren eventuell oder es existieren äquivalente Programme.

Die genannten Programme sind entweder unter einer OpenSource-Lizenz veröffentlicht oder kostenfrei nutzbar.

Installation Subversion Client

Für dieses Handbuch wird beispielhaft mit dem Subversion Client "TortoiseSVN" gearbeitet. Das Installationspaket zum Download befindet sich auf der Projektseite <http://tortoisesvn.net/downloads>. Nach der Installation des korrekten Pakets muß der Rechner neu gestartet werden. Der TortoiseSVN-Client wird als Windows Explorer-Erweiterung ins System eingebunden.

Die Installation ist ausführlich auf den Projektseiten der Software [http://tortoisesvn.net/docs/release/TortoiseSVN_de/tsvn-intro-install.html#id547135] beschrieben.

Es existiert ein deutsches Sprachpaket [<http://tortoisesvn.net/downloads.html>] (32 Bit und 64 Bit) für diesen Subversion-Client. Es ist zu beachten, dass die Versionsnummer des Tortoise-SVN-Clients sowie die Versionsnummer des Sprachpakets identisch sind. Das Sprachpaket wird nach erfolgreicher Installation des Tortoise-SVN-Clients installiert. Die verfügbaren Sprachen können wie folgt aktiviert werden:

- Rechtsklick
- Kontextmenüeintrag 'TortoiseSVN'
- Kontextmenüeintrag 'Settings'
- Im erscheinenden Einstellungsdialog unter 'General' -> 'Language' die gewünschte Sprache aus dem Dropdown-Menü auswählen

Installation Serna Free XML-Editor

Serna Free Download [<http://sourceforge.net/projects/sernafree.mirror/>] - basiert auf Java, installiert jedoch eine eigene JRE, eine gesonderte Installation von Oracles Java JRE ist daher nicht notwendig.

Kapitel 2. Verwaltung der Dokumente

zpub stellt eine einfach zu bedienende, grafische, browsergestützte Oberfläche für die im SVN verwalteten Dokumente zur Verfügung. Dabei kann die Historie eines Dokuments verfolgt werden, Änderungen können zeitlich und per Redakteur zugeordnet werden, Dokumente können freigegeben werden, Gruppen von Redakteuren gebildet werden, Ausgabeformate angesehen und heruntergeladen werden und der Status aktueller Ausgabegenerierungsprozesse angezeigt werden.

Das Webinterface von zpub hält je nach den Rechten des angemeldeten Benutzers unterschiedliche Informationen und Funktionen parat. Bisher existieren zwei Benutzerrollen:

1. Redakteure

Redakteure sind in der Lage, sämtlich relevanten Informationen und Funktionen zum Verwalten und Einsehen der von Ihnen bearbeiteten Dokumente zu nutzen.

2. Administratoren

Administratoren bekommen zusätzlich zu den Informationen und Funktionen der Redakteure weitere Funktionen dargestellt. Diese Funktionen sind in der folgenden Übersicht gekennzeichnet. Die Zuordnung der Rollen geschieht über Max Mustermann, max.mustermann@demo.zpub.de [mailto:max.mustermann@demo.zpub.de].

Das Webinterface

Die Verwaltung von Dokumenten sowie die automatische Generierung der Ausgabeformate geschieht serverseitig und ist über ein Webinterface zu administrieren.

Anmerkung

URL des Webinterfaces:

<https://demo.zpub.de/>

Die Seiten des Webinterfaces werden bei jedem Aufruf generiert und beinhalten - sofern gerade aufgerufen oder aktualisiert - immer den aktuellen Stand. Werden relative Zeitangaben gemacht, so gelten diese relativ zum Zeitpunkt des Seitenaufrufs bzw. der letzten Aktualisierung.

Tipp

Das die Verbindung zu zpub läuft über eine verschlüsselte Verbindung. Das selbst-signierte SSL-Zertifikat muß akzeptiert und bestätigt werden.

Das Webinterface bietet Funktionen zu den einzelnen Dokumenten wie auch zur Administration des Systems.

Das Menü ist hierarchisch mit zwei Ebenen aufgebaut, wobei zu jedem Dokument eine Unterebene existiert. Die Menüpunkte sind im Folgenden erläutert (Übersicht):

- Dokumentübersicht
- Dokument „Handbuch“
 - Archiv
 - Abonnenten
- Status

- Benutzer

Aufbau Webinterface

Dokumentübersicht

In der Dokumentübersicht werden die verwalteten Dokumente aufgeführt. Die verwalteten Dokumente sind ebenfalls in der Navigation vorhanden.

Dokument „Handbuch“

Die Übersicht über ein Dokument enthält Angaben über die letzten Änderungen, den Status der Freigabe sowie die erzeugten Ausgabeformate. Existieren zu einem Dokument Änderungen, die aktueller sind als die letzte freigegebene Version, kann dieses Dokument freigegeben werden.

Archiv

Zu jedem Dokument existiert ein Archiv, welches in tabellarischer Darstellung die Bearbeitungsschritte des Dokuments mit Angaben zum jeweiligen Redakteur, der inhaltlichen Änderung, des Änderungsdatums und des Freigabestatuses enthält. Die Tabelle ist nach den laufenden Revisions-Nummern sortiert, die auch in den Arbeitsversionen der Dokumente eingefügt sind. Weiter können noch nicht freigegebene Revisionen hier freigegeben werden.

Das Archiv beinhaltet alle Revisionen seit der Verwaltung der Dokumente mit zpub. Revisionen werden nicht gelöscht, die gesamte Historie bleibt erhalten und wird lediglich vom Speicherplatz auf dem zpub-Server begrenzt.

Versionierung

Jede via SVN übertragene Änderung stößt auf dem zpub-Server eine Neugenerierung sämtlicher Ausgabeformate an. Die zu einer Version bzw. Revision erstellten Ausgabeformate werden dauerhaft abgespeichert, wobei sie weiterhin verlinkt und abrufbar sind. Die Links beinhalten die Revisionsangaben und sind wie folgt aufgebaut:

`https://demo.zpub.de/zpub-Technik/archive/13 /handbuch/zpub-Technik.pdf`

Revisionsangabe

Um immer auf die aktuellste Dokumentrevision zuzugreifen, existiert ein *latest*-Link zu jedem Ausgabeformat:

`https://demo.zpub.de/zpub-Technik/latest/handbuch/zpub-Technik.pdf`

Die Revisionsangabe ist durch **latest** ersetzt. Dieser Link kann, da er sich nicht ändert, z.B. für Bookmarks etc. verwendet werden.

Abonnenten

Dieser Punkt ist nur bei entsprechenden Rechten sichtbar.

Unter "Abonnenten" sind die Email-Adressen der Redakteure einzutragen, die bei neuen Versionen dieses Dokuments per Email benachrichtigt werden sollen. Die Benachrichtigungsemail enthält Angaben zum Redakteur, der die Änderung durchgeführt hat, den Zeitpunkt der Änderung, der Kurzbeschreibung der Änderung sowie Links zu den generierten Ausgabeformaten.

Status

"Status" fasst Systemmeldungen sowie Meldungen der automatischen Ausgabegenerierung zusammen. Die sogenannten "job"-Angaben sind nach "Wartenden Aufträgen", "Aufträge in Bearbeitung"

sowie "Fehlgeschlagene Aufträge" eingeordnet. Sollte ein Auftrag fehlgeschlagen sein, so ist hier die Fehlermeldung ("Logdatei") einsehbar und es werden Optionen zu einem erneuten Versuch und des Löschens des Auftrages angeboten.

Benutzer

Dieser Punkt ist nur bei entsprechenden Rechten sichtbar.

Listet alle Benutzer des Systems auf. Hier eingetragene Benutzerkennungen sind zum einen für die Anmeldung am Webfrontend (die Seiten, die über <http://demo.zpub.de> [<https://demo.zpub.de>] erreichbar sind) wie auch für die Nutzung des Versions-Verwaltungssystems gültig.

Freigabe von Dokumenten

Dokumente werden standardmäßig mit einem "Draft"-Wasserzeichen sowie Revisions-, Redakteursangaben und Zeitstempel versehen. Während ein Dokument also von einem Redakteur oder einer Redakteursgruppe "in Bearbeitung" ist, wird dies durch das Wasserzeichen kenntlich gemacht. Diese Dokumente (Revisionen) sind nur für den internen Gebrauch bestimmt. Erreicht ein Dokument einen *finalen* Stand (Revision), kann diese Revision *freigegeben* werden. Dies geschieht über die entsprechenden Buttons (auf der Dokumentseite sowie der zugehörigen Archiv-Seite). Dabei generiert *freigegeben* die entsprechende Revision des Dokuments erneut - jedoch ohne das Wasserzeichen und die Revisionsmarkierungen.

Kapitel 3. Nutzung von Subversion

Die Arbeit über Subversion ermöglicht das gleichzeitige, verteilte Arbeiten an einem Dokument. Änderungen werden ebenso wie die Dokumente selber zentral verwaltet und versioniert. Weiter bietet Subversion die Möglichkeit, ältere Zustände eines Dokuments einzusehen und gegebenenfalls wiederherzustellen.

Mit dem Subversion-Client greift ein Redakteur auf die zentral verwalteten Dokumente zu. Dabei wird mit lokalen Kopien der Dateien gearbeitet. Nach Änderungen bzw. dem Editieren der Dokumente werden diese wieder in das zentrale Subversion-Repository 'committed' ("übergeben"). Arbeiten mehrere Redakteure an einem Dokument, so weiß der Subversion-Client beim 'commiten' des Dokuments auf eventuelle Konflikte hin.

Tipp

URL des Subversion-Repositories:

<https://demo.zpub.de/svn/>

Tipp

Der TortoiseSVN-Client bindet sich als Windows Explorer-Erweiterung im System ein. Seine Funktionalität stellt er beim Rechtsklick auf eine Datei im Explorer zur Verfügung.

Das TortoiseSVN-Projekt pflegt eine gute, detaillierte und deutsche Dokumentation - siehe http://tortoisesvn.net/docs/release/TortoiseSVN_de/index.html.

Warnung

Wird ein SVN-Repository in eine lokale Kopie ausgescheckt, so werden neben den eigentlichen Dokument-Quelldateien SVN-Systemdateien und Ordner in der lokalen Kopie erstellt. Diese befinden sich in jedem Ordner, der unter der Versionsverwaltung steht, als `.svn`-Ordner. Diese Ordner, die als versteckte Ordner gegebenenfalls nicht angezeigt werden, dürfen nicht manuell verändert werden, da sie SVN-Verwaltungsdaten enthalten.

TortoiseSVN bietet eine Reihe nützlicher Funktion zum Verwalten der Dokumente sowie der Rückverfolgung von Änderungen, die über das TortoiseSVN-Kontextmenü zugänglich sind. Insbesondere ist die Log-Funktion und die Vergleichsversion für unterschiedliche Revisionen genannt.

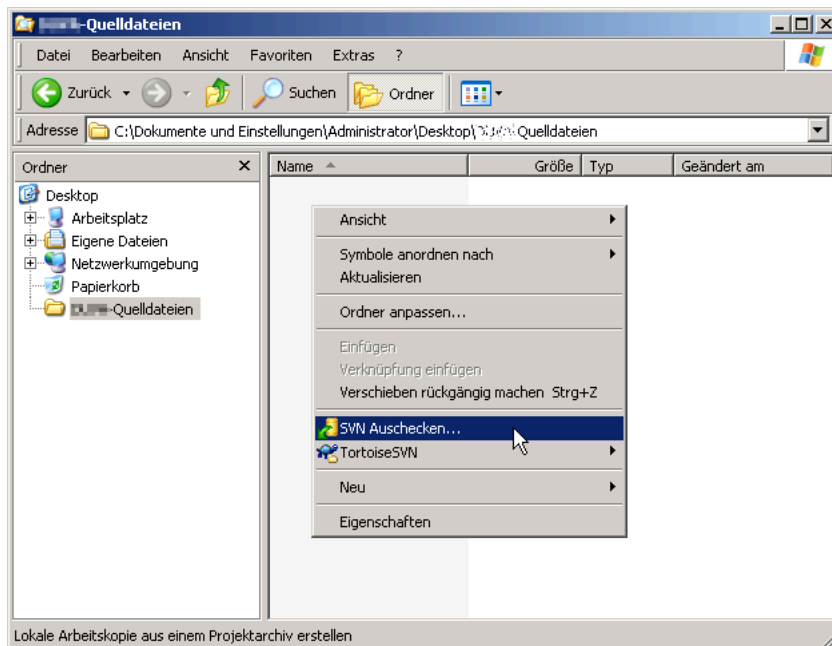
Kurzdarstellung eines kompletten Editiervorgangs

1. Erstellen eines lokalen Ordners (hier: DEMO-Quelldateien), welcher die DEMO-Dokumente enthalten soll (einmalig)
2. Auschecken (einmalig) bzw. Updaten des zpub-DEMO-Repositories in dem soeben erstellten Ordner
3. Öffnen der gewünschten XML-Dokumentdatei mit dem XML-Editor
4. Bearbeiten des Dokuments
5. Comitten/Übertragen der der Änderungen an das zpub-DEMO-Repository
6. Über das zpub-DEMO-Webinterface [<https://demo.zpub.de/>] werden die generierten aktuellen Ausgabedokumente (pdf, chm, html) sowie grundlegende Daten zur Historie des Dokuments sowie dessen Freigabe-Status abgebildet.

Auschecken des SVN-Repositoryes

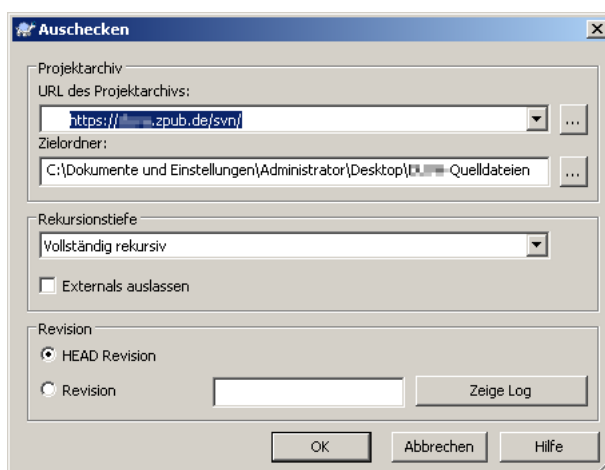
1. Erstellen eines leeren Ordners
2. Wechseln in den neuen Ordner
3. Rechtsklick, Auswahl des Kontextmenüpunkts "SVN auschecken..."

Abbildung 3.1. Screenshot: SVN auschecken...



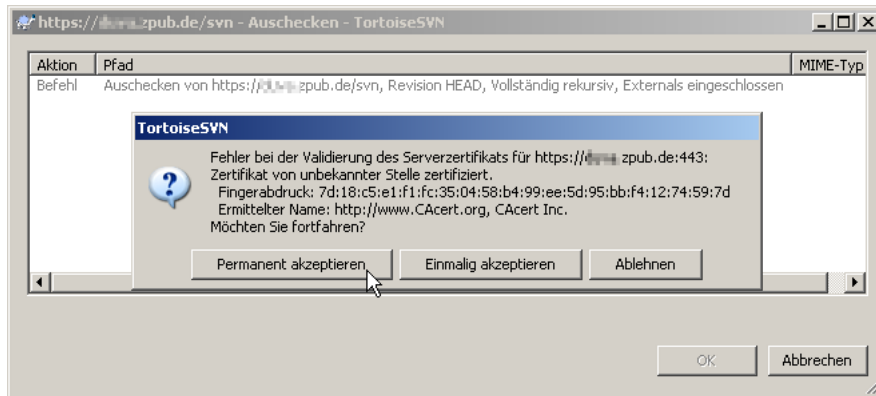
4. Im erscheinenden Dialog die URL zum DEMO-SVN-Repository eintragen

Abbildung 3.2. Screenshot: Projektarchiv



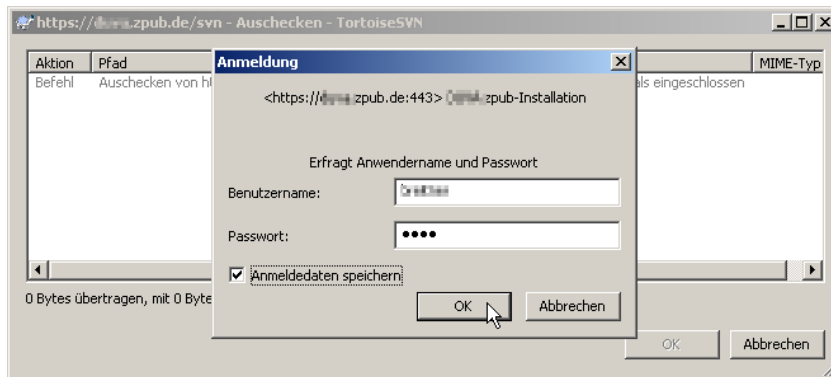
5. Zertifikatnachfrage mit "Permanent akzeptieren" bestätigen

Abbildung 3.3. Screenshot: SVN Zertifikatsabfrage



6. Anmelddaten eingeben und um spätere Nachfragen zu umgehen Option "Anmelddaten speichern" aktivieren.

Abbildung 3.4. Screenshot: SVN Anmelddaten

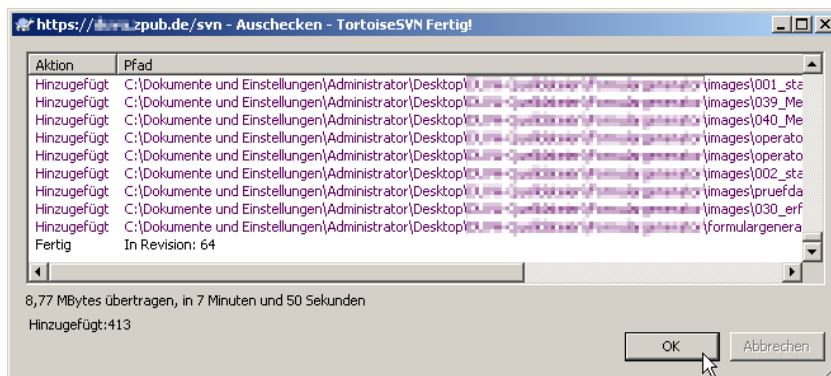


Anmerkung

Die Benutzer müssen zuvor im Webinterface unter dem Punkt Benutzer angelegt werden.

Sobald der Anmeldedialog mit "Ok" bestätigt wird, beginnt das Auschecken der Dateien im Repository in die lokale Kopie. Dies kann einige Minuten dauern, auch ohne dass sich der Fortschrittsdialog ändert.

Abbildung 3.5. Screenshot: SVN ausgecheckt



7. Nach dem ersten Auschecken ist eine lokale Kopie der Quelldateien des entsprechenden Dokuments auf dem Redakteursrechner vorhanden. Die Quelldateien setzen sich aus der DocBook-XML-Datei des Dokuments sowie gegebenenfalls zugehörigen Bildern.

TortoiseSVN ergänzt die Standard-Datei- und Ordner-Icons um kleine Zusatzsymbole, die den aktuellen SVN-Status des entsprechenden Elements darstellen. Nach einer Aktualisierung (s.o.) sind die Dateien der lokalen Arbeitskopie und des SVN-Repositories auf dem selben Stand, die lokale Kopie ist damit "aktuell" und wird mit einem grünen Häkchen versehen.

Abbildung 3.6. SVN-Symbol für aktuellen Ordner

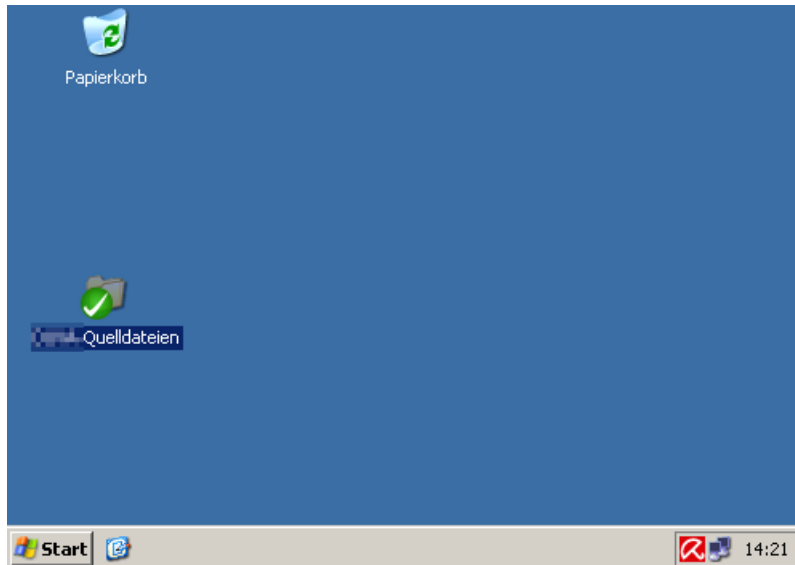
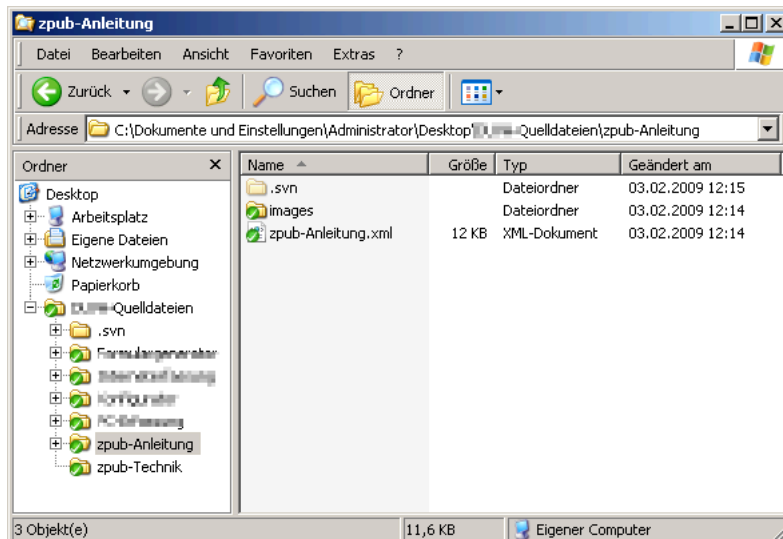


Abbildung 3.7. SVN - Symbol für aktuelle Datei



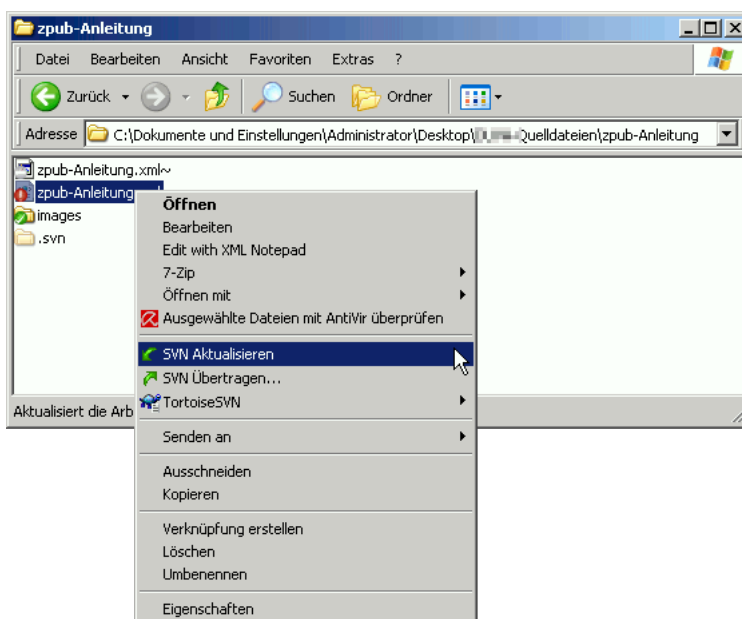
8. Es liegt nun eine lokale Kopie aller Quelldateien vor. Diese Dateien können - unabhängig von einer (Internet-) Verbindung zum SVN-Server bearbeitet werden.
9. Für weitere Funktion des SVN-Clients nutzen Sie bitte die Dokumentation des Herstellers, der installierten Hilfe zur Software oder erforschen die Software über das umfangreiche Kontextmenü, welches durch einen Rechtsklick auf eine SVN-verwaltete Datei/Ordner erscheint.

Daten aktualisieren und übertragen

Vor jedem neuen Editiervorgang sollte die lokale Kopie des SVN-Repositories auf den aktuellen Stand gebracht werden, es könnten inzwischen Änderungen von anderen Redakteuren gemacht und übertragen worden sein. Nach dem Editieren eines Dokuments wird dieses mit dem TortoiseSVN-Client wieder in das zentrale SVN-Repository übertragen und steht ab dann auch den anderen Redakteuren zur Verfügung.

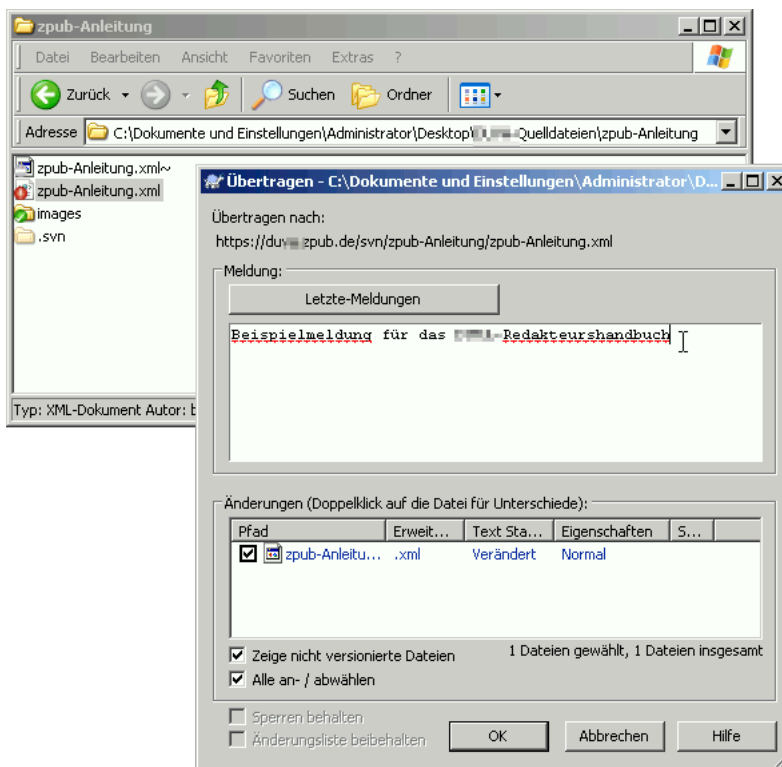
Das Aktualisieren eines Dokuments - bzw. der Abgleich des lokalen Dokuments mit dem Dokument auf dem SVN-Server - geschieht über einen Rechtsklick auf die entsprechende Datei. Das aufklappende Kontextmenü beinhaltet direkt die beiden grundlegendsten SVN-Aktionen "SVN aktualisieren" und "SVN übertragen...".

Abbildung 3.8. TortoiseSVN Kontextmenü - Aktualisieren



Nach dem Ausführen des Befehls "SVN Aktualisieren" kann das Dokument editiert werden. Im Anschluß an das Editieren ist die lokale, aktuelle Version wieder auf den SVN-Server zu übertragen. Das Vorgehen ist analog zum soeben beschriebenen "aktualisieren", es ist lediglich der Punkt "SVN Übertragen..." auszuwählen. Vor der Übertragung kann und soll eine eigene, kurze Meldung zu den gemachten Änderungen angegeben werden. Diese Meldung wird über die zpub.de-Webseite den angeschlossenen Redakteuren veröffentlicht.

Abbildung 3.9. SVN-Übertragungsmeldung



Anmerkung

Es ist nützlich, "oft und ausführlich" zu übertragen, d.h. die eigenen Änderungen in kleinen, überschaubaren Einheiten zu übertragen und zu jeder Einheit eine gute, beschreibende Meldung anzugeben. Jede Übertragung von Änderungen erzeugt eine neue Revision, und das Versionsverwaltungssystem SVN operiert auf Revisionen. Man kann zu alten Revisionen "zurückspringen", man kann sich Änderungen zwischen beliebigen Revisionen anzeigen lassen, man kann der Historie der Revisionen anhand der "sprechenden" Meldungen folgen.

Neben diesen technischen Möglichkeiten ist jedoch die "zwischenmenschliche Nachvollziehbarkeit" von Änderungen innerhalb einer Projekt-/Redakteursgruppe hervorzuheben. zpub hilft durch die Archiv-Ansicht, einen Überblick über die Historie eines Dokuments zu behalten:

Abbildung 3.10. zpub: Archiv

Rev.	Autor	Datum
6	demo „customer-entity korrigiert“	1. November 2009 um 15:12
5	demo „Sprachlicher Fehler behoben“	1. November 2009 um 15:11
4	demo „SVN-Keywörd gesetzt“	1. November 2009 um 15:08
3	demo „Technik-Doku in öffentliche zpub-Instanz aufgenommen“	1. November 2009 um 15:07

SVN Konflikte

Über das Versionsverwaltungssystem SVN werden die Dokumente und zugehörigen Dateien etc. verwaltet: sie werden mit einer eindeutigen Revisions-Nummer, einem Zeitstempel und der Benutzerken-

nung versehen. Durch die zentrale Verwaltung der Dokumente auf dem SVN-Server, können mehrere Redakteure an den Dokumenten arbeiten. Der Workflow eines Redakteurs beginnt i.d.R. mit dem Aktualisieren seiner lokalen Kopie, dem Editieren und endet mit dem Übertragen der Änderungen an den SVN-Server.

Bearbeiten zwei Redakteure (hier: Hinz und Kunz) jeweils ihre privaten, lokalen Arbeitskopien (hier: Datei test.xml) und übertragen ihre Änderungen wieder an das SVN-Projektarchiv, können Konflikte entstehen:

1. Automatisch auflösbare Konflikte

Hinz editiert im ersten Kapitel und überträgt seine Änderungen. Kunz editiert im zweiten Kapitel und überträgt seine Änderungen. Bei dieser zweiten Übertragung erkennt SVN, dass sich inzwischen etwas an der Datei test.xml getan hat - die Änderungen von Hinz. Diese Änderungen aber in der Datei, die Kunz vorliegt, nicht enthalten sind. Daher fordert SVN Kunz auf, vor dem Übertragen zuerst die Datei test.xml seiner lokalen Arbeitskopie auf den aktuellen Stand zu bringen. Bei der Aktualisierung erkennt SVN, dass es Änderungen an zwei Stellen (Kapitel ein und zwei) gibt, die nicht miteinander im Konflikt stehen und führt die Änderungen zusammen. In der Datei sind nun sowohl die Änderungen von Hinz im ersten Kapitel wie auch von Kunz im zweiten Kapitel vorhanden. Diese Datei kann nun Kunz an das SVN-Projektarchiv übermitteln, damit alle Beteiligten den neuen Stand zur Verfügung haben.

2. Manuell aufzulösende Konflikte

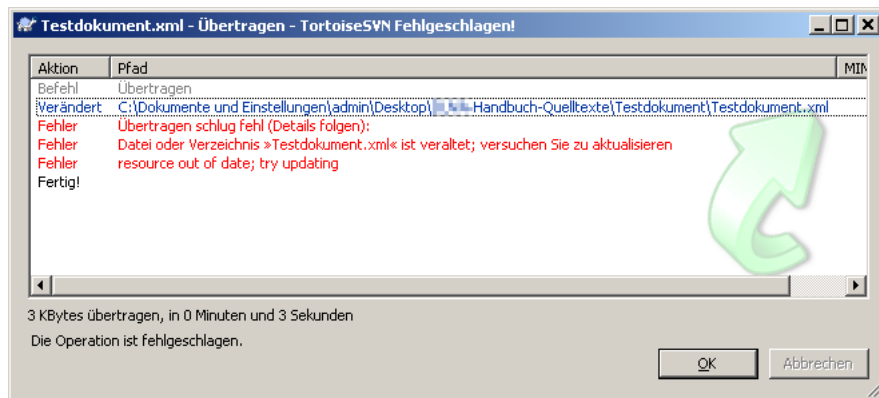
Hinz editiert den Abbildungstitel von Abbildung fünf. Kunz - der nichts von den Editiervorgängen von Hinz weiß - editiert ebenfalls den Abbildungstitel von Abbildung fünf. Hinz und Kunz übertragen ihre Änderungen an das SVN-Projektarchiv. Derjenige, der als zweiter (hier: Kunz) seine Änderungen übertragen will, wird vom SVN aufgefordert, seine lokale Datei zuerst auf den aktuellen Stand zu bringen, da diese ja die Änderungen des zuerst übertragenden Redakteurs (hier: Hinz) noch nicht enthält. Aktualisiert nun Kunz seine Datei, erkennt das SVN die Änderungen beider Redakteure an "der selben Stelle". Dieser Konflikt kann nicht automatisch aufgelöst werden, hier muß ein intelligentes Wesen entscheiden, welcher Abbildungstitel denn nun der Richtige ist. Das SVN markiert die Datei als "in Konflikt" und überlässt das Auflösen des Konflikts Kunz. Allerdings unterstützt es Kunz, indem es ihm die entsprechende Stelle (Zeile) in der Datei sowie die beiden konkurrierenden Abbildungstitel anzeigt. Indem Kunz sich nun für einen der Abbildungstitel entscheidet, führt er die Dateien manuell zusammen. Die Datei ist nun nicht mehr "in Konflikt" und kann von Kunz an das SVN-Projektarchiv übertragen werden.

Anmerkung

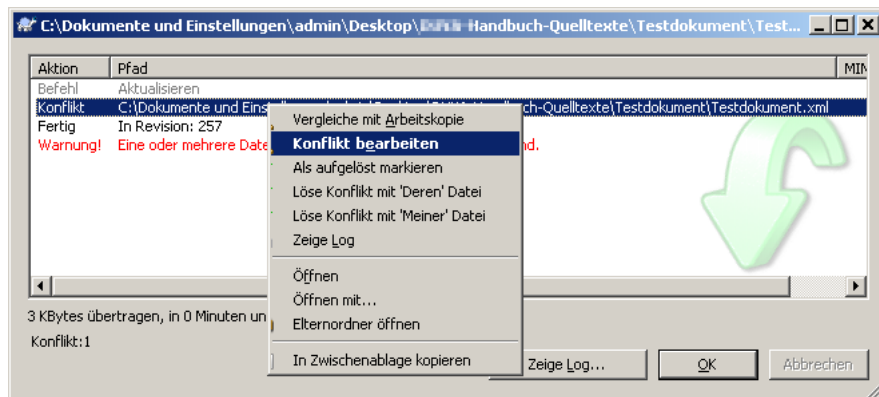
Das TortoiseSVN-Projekt pflegt eine eigenständige - auch auf Deutsch verfügbare - Dokumentation [http://tortoisesvn.net/docs/release/TortoiseMerge_de/index.html] zu dem Programmmodul TortoiseMerge, welche detailliert auf Aspekte wie Revisionierung, Zusammenführen und Konflikte eingeht.

Konflikte manuell lösen

Konflikte in einem Dokument, die sich von SVN nicht automatisch auflösen lassen (siehe voriges Kapitel), müssen vom Redakteur manuell gelöst werden. Der Redakteur wird beim Versuch, eine konfliktbehaftete Datei an das Projektarchiv zu übertragen mit einem Meldungsfenster über den Konflikt informiert. Dabei wird der Redakteur zuerst aufgefordert, seine Datei zu aktualisieren:

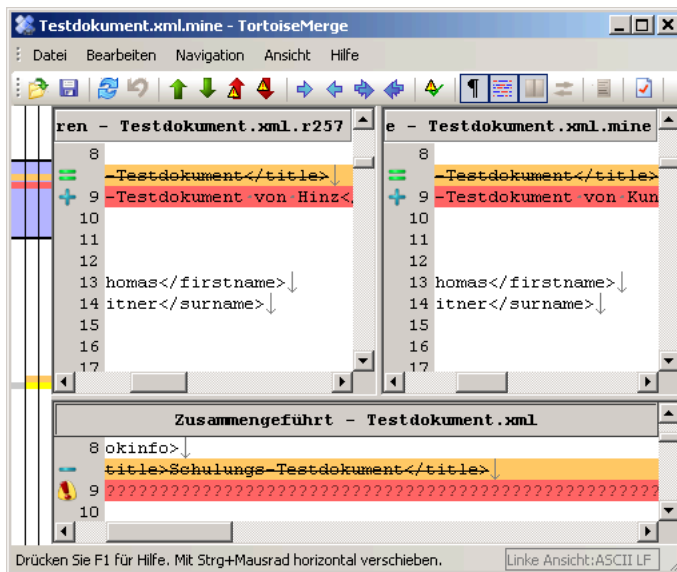
Abbildung 3.11. SVN-Meldung: Datei veraltet


Die beanstandete Datei (hier: `Testdokument.xml`) ist demnach zuerst zu aktualisieren. Kann die Datei beim Aktualisieren nicht automatisch zusammengeführt werden, wird eine Konfliktmeldung ausgegeben. Über einen Rechtsklick auf die Konfliktmeldung und den Eintrag **Konflikt bearbeiten** kann direkt das Programm TortoiseMerge aufgerufen werden:

Abbildung 3.12. SVN-Meldung: Konflikt

Es öffnet sich nun das Programm TortoiseMerge, welches in einer dreigeteilten Fensteransicht den Redakteur unterstützt, die konfligierenden Änderungen aufzulösen. In unserem Beispielfall sieht dies wie folgt aus:

Abbildung 3.13. TortoiseMerge: Konflikte manuell auflösen



Die dreigeteilte Fensteransicht beinhaltet oben links den Bereich für "Deren" Datei, links oben für "Meine" Datei sowie unten das Ergebnis der zusammengeführten Datei. Die konfliktbehaftete Zeile ist in "Deren" wie in "Meiner" Datei rot hervorgehoben. Weder in "Deren" noch in "Meiner" Datei kann editiert werden - hier kann man sich über einen Rechtsklick auf die Konfliktzeile lediglich dafür entscheiden, welche Version der Zeile man als Lösung betrachtet. Kommt weder die Zeile aus "Deren" Datei, noch die korrespondierende Zeile aus "Meiner" Datei in Betracht, muß die Datei neu editiert/korrigiert werden. Dies geschieht im unteren Fensterteil "Zusammengeführt". Im Beispiel (siehe Abbildung 3.13, „TortoiseMerge: Konflikte manuell auflösen“) ist die Zeile ebenfalls rot hervorgehoben und - da keine automatische Zusammenführung möglich ist - mit Fragezeichen gefüllt. Diese Zeile (sowie die gesamte Datei) kann nun editiert werden. Ist kein Konflikt mehr vorhanden, ist auch keine Zeile in der unteren Fensterzeile mehr rot eingefärbt. Die Datei kann nun gespeichert und "als aufgelöst" markiert werden. Dies geschieht über den Button .

Kapitel 4. Editieren von Dokumenten

zpub-Dokumente sind DocBook 4.5 [<http://www.docbook.org/xml/4.5/>]-Dokumente¹. Der aktuelle Workflow von zpub geht von dieser DocBook-Version aus und die Stylesheets für die Ausgabeformate erwarten ein valides DocBook-4.5-Dokument. Die offizielle Dokumentation zu DocBook 4.5 existiert <http://www.docbook.org/tdg/en/html/docbook.html> online.

Neues Dokument erstellen

Ein neues Dokument besteht aus einem Verzeichnis, der DocBook-Datei sowie optional einem images-Verzeichnis. Das Verzeichnislayout lautet wie folgt:

```
`-- Verzeichnisname
   |-- images
   #   |-- bild_01.png
   #   |-- bild_02.png
   |-- Dokumentname.xml
```

Projektverzeichnis. Verzeichnisname wird für die Benennung des Projekts auf der zpub-Webseite genutzt. Es ist nicht zwingend notwendig, dass Projektverzeichnisname und Dokumentdateiname übereinstimmen, bietet sich jedoch an.

Verzeichnis für die Ablage von Grafiken, die im DocBook-Dokument verlinkt werden. Das DocBook-Dokument. Dateiname wird ebenfalls für die Ausgabeformate genutzt.

Ein (minimales) Grundgerüst einer (zpub-)DocBook4.5-Datei ist im folgenden dargestellt:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-op
<book lang="de">
  <title>Dokumenttitel</title>
  <chapter>
    <title>Kapiteltitel</title>
    <para>Inhalt</para>
  </chapter>
</book>
```

Die Verzeichnisstruktur inklusive der Dateien (DocBook-Datei sowie eventuell Grafiken) wird zpub durch das Hinzufügen zum SVN-Repository sowie dem anschließenden commit bekannt gemacht. Der Commit der Datei(en) stößt bei zpub die Generierung der konfigurierten Ausgabeformate, die Versionierung und das Update der zpub-Webseite an.

Kopieren einer Vorlage als neues Dokument

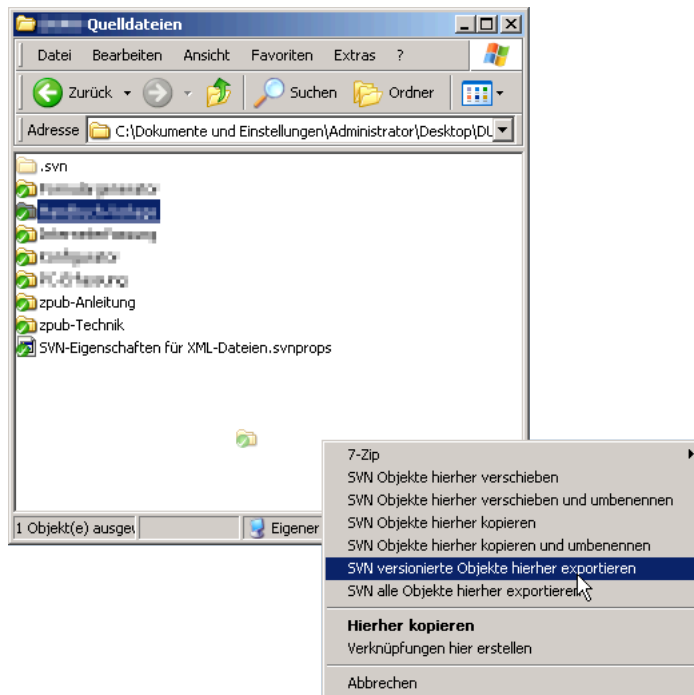
Im Verzeichnis der zpub-Quelldokumente befindet sich eine minimale Vorlage für ein neues Handbuch-Projekt im Verzeichnis `Handbuch-Vorlage`. Diese Datei kann als Ausgangspunkt für ein neues Projekt genutzt werden. Selbstverständlich lassen sich auch vorhandene Handbücher als Vorlage nutzen. Die folgenden Schritte führen zu einem neuen, leeren Dokument:

1. Exportieren des Vorlageordners:

Der Vorlageordner (hier: `Handbuch-Vorlage`) wird über die SVN-Funktion 'SVN versionierte Objekte hierher exportieren' kopiert. Diese Funktion steht beim klicken-und-ziehen mit der rechten Maustaste zur Verfügung:

¹Ein Hinweis zur Nutzung von Serna Free mit DocBook5 findet sich z.B. auf <https://wiki.hpdd.intel.com/> [<https://wiki.hpdd.intel.com/display/PUB/Making+changes+to+the+Lustre+Manual+source#MakingchangestotheLustreManualsource-serna>].

Abbildung 4.1. Versionierte Dateien exportieren



2. Umbenennen von Verzeichnis und der XML-Datei

Beim Umbenennen müssen das Verzeichnis und die Handbuch-XML-Datei nicht zwingend den selben Name besitzen. Bei der Benennung dürfen keine Sonderzeichen wie Umlaute, Leerzeichen etc. verwendet werden.

Anmerkung

Der Verzeichnisname und der Dateiname der XML-Datei müssen nicht übereinstimmen - zweckmäßig ist dies jedoch. Der Verzeichnisname wird für die Benennung des Projekts auf der zpub-Webseite genutzt, der Dateiname für die Benennung der Ausgabeformate.

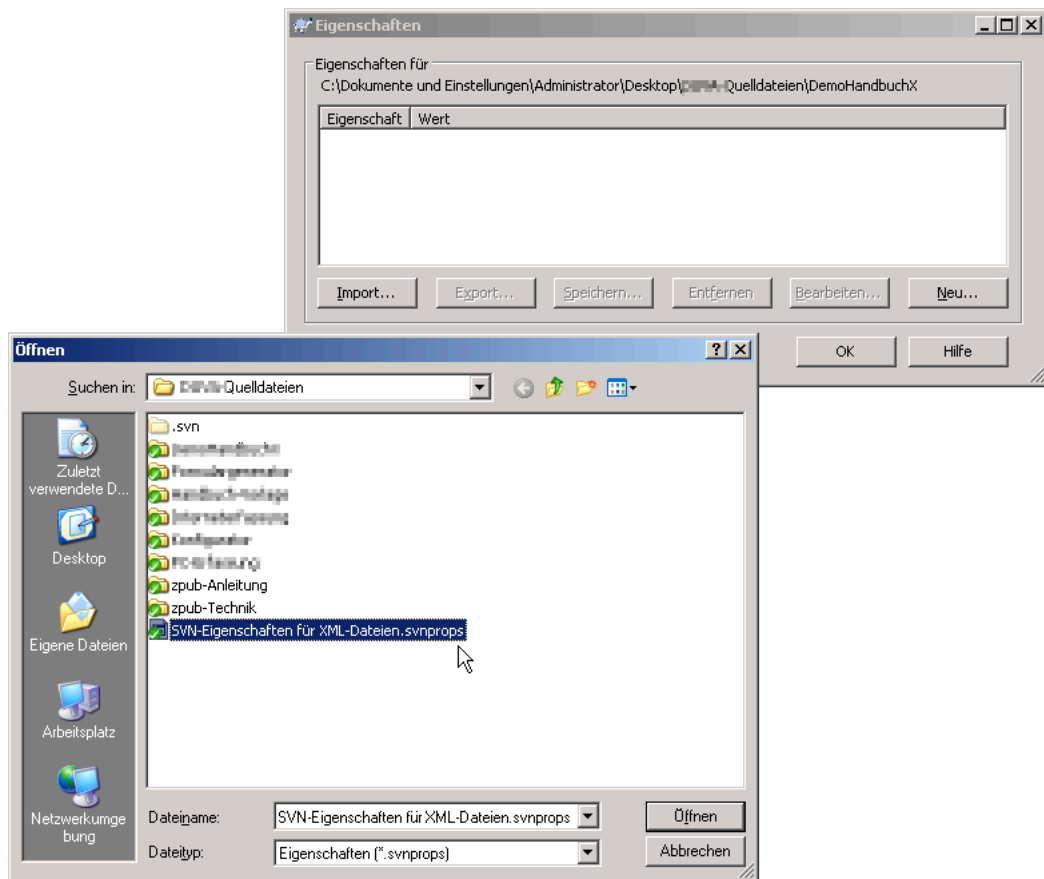
3. Verzeichnisse und Dateien via SVN dem zentralen Repository hinzufügen.

Dies geschieht über den Eintrag 'SVN hinzufügen' im Kontextmenü von TortoiseSVN beim Rechtsklick auf das neue Verzeichnis. Es werden automatisch rekursiv alle Dateien/Unterverzeichnisse des Verzeichnisses hinzugefügt.

4. SVN-Eigenschaften importieren

Die benötigten Eigenschaften liegen vorkonfiguriert in Dateiform vor und müssen lediglich noch importiert werden. Dies geschieht über das Kontextmenü (Rechtsklick) der XML-Datei im neuen Verzeichnis. Hier sind die folgenden Punkte auszuwählen: TortoiseSVN → Eigenschaften. In diesem Dialog ist über den Button Import... die Datei 'SVN-Eigenschaften für XML-Dateien.svnprops' zu importieren, die sich im Oberverzeichnis der Handbücher befindet.

Abbildung 4.2. SVN-Eigenschaften importieren



- Handbuch editieren und/oder Verzeichnisse/Dateien via SVN ins zentrale Repository übertragen. Dies geschieht über den Eintrag 'SVN übertragen' im Kontextmenü des neuen Verzeichnisses.

Dokument löschen

Ein Dokument wird inklusive seiner vorhandenen Revisionen gelöscht, indem das dazugehörige Projektverzeichnis aus dem SVN-Repository gelöscht wird.

Anmerkung

Auch ein Löschvorgang im SVN-Repository muss „committed“ werden.

DocBook-Struktur

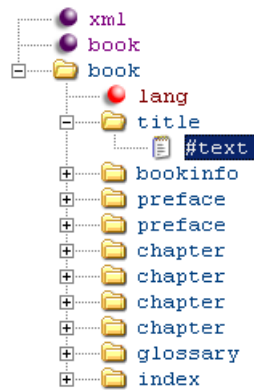
Die Quelldateien der DEMO-zpub-Dokumente sind XML-Dateien, die dem DocBook-Standard entsprechen. Sie können mit jedem beliebigen, UTF-8-fähigen Texteditor bearbeitet werden. Ob die Bearbeitung mit einem einfachen Editor oder einem speziellen XML-Editor geschieht, ist dem Redakteur sowie der technischen Ausstattung der Redakteurs-Rechner überlassen. Der hier vorgestellte Editor Serna Free stellt einen Mittelweg dar, der die Vorteile beider Verfahren vereint und dem Redakteur die technische Seite des XML-Dialekts abnimmt.

Eine umfassende Dokumentation von Serna ist auf der Herstellerseite [<http://www.serna-xmleditor.com/wp-content/plugins/download-monitor/download.php?id=2>] zu finden.

Im Folgenden wird der Editor vereinfacht Serna genannt.

Bei der Arbeit an einem DocBook-Dokument hilft es, wenn man sich über die interne Struktur eines DocBook-Dokuments im Klaren ist. Ein DocBook-Dokument ist streng hierarchisch in einer Baumstruktur aufgebaut. Jeder Knoten entspricht einem DocBook-Tag, also einer Art Anweisung, um was für eine Art von Information es sich beim Inhalt dieses Knotens handelt. Soll nun neuer Inhalt eingefügt werden, so muss zuerst ein dem Inhalt entsprechender Knoten eingefügt werden. Die folgende Abbildung zeigt den Aufbau eines DocBook-Dokuments auf erster Ebene:

Abbildung 4.3. DocBook-Baumansicht



Um die inhaltliche Ebene zu demonstrieren, ist der `title`-Tag ausgeklappt. Der `#text`-Platzhalter nimmt den tatsächlichen Inhalt auf - in diesem Falle den Titel dieses Buches/Dokuments.

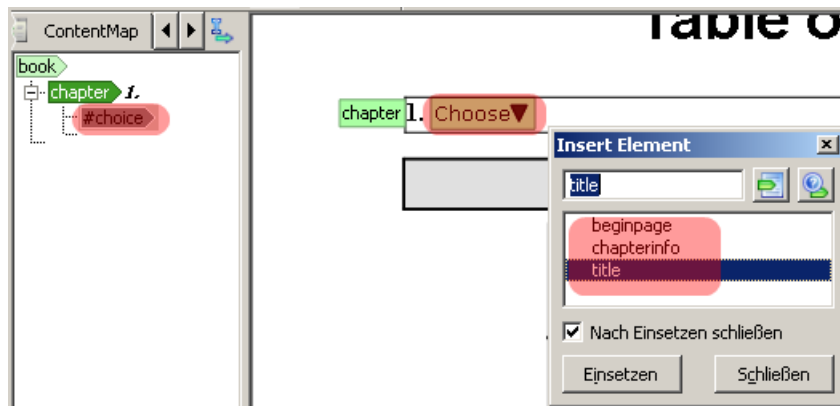
Arbeiten mit Serna Free

Das Editieren von DocBook-Dokumenten ähnelt dem gewohnten Editieren mit einer Textverarbeitung: Texte können eingefügt, geändert, kopiert, gelöscht, verschoben werden. Es existieren jedoch einige grundlegende Unterschiede:

- Es werden nur Inhalte bearbeitet, keine Formatierungen
- Es existieren Beschränkungen, welcher Inhalt wo vorkommen kann oder muß
- Inhalte werden mit Tags "ausgezeichnet", welche den auszuzeichnenden Inhalt möglichst gut beschreiben.
- Es können z.B. auch nicht mehrere aufeinanderfolgende Leerzeichen eingegeben werden. Auch existieren keine Tabulatoren oder sonstige Möglichkeiten zur Definition von selbst grundlegenden Formatierungen.
- Die Taste **Return** bewirkt nicht einen gewöhnlichen Zeilenumbruch, sondern fügt ein neues Absatz-Element (`para`) ein.
- Einige Bereiche/Texte sind nicht direkt editierbar. So werden z.B. die Verzeichnisse für den Inhalt, die Abbildungen, die Nummerierung der Abbildungen und Kapitel etc. automatisch vorgenommen.

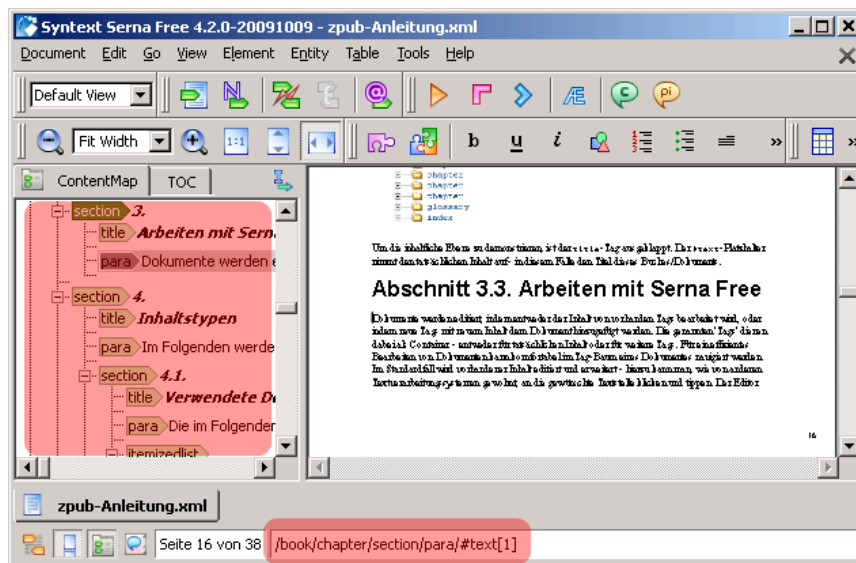
In vielen Fällen hilft der Editor, sich im Wald der DocBook-Tags zurechtzufinden. Da der Editor nur valide Eingaben zulässt, bzw. nur valide Dokumente bearbeitet, und dem Benutzer bei der Auswahl eines passenden und validen DocBook-Tags unter die Arme greift, muß der Anwender kein DocBook-Experte sein. Existieren bei der Eingabe mehrere Möglichkeiten, mit welchem Element (DocBook-Tag) es "weitergeht", so stellt der Editor dem Benutzer auch nur die an dieser Stelle möglichen Elemente zur Auswahl. Ist an dieser Stelle zwingend ein bestimmtes Element nötig, so fordert der Editor offensiv zu einer Auswahl aus:

Abbildung 4.4. Element-Auswahl choice/choose



Dokumente werden editiert, indem entweder der Inhalt von vorhanden Tags bearbeitet wird, oder indem neue Tags mit neuem Inhalt dem Dokument hinzugefügt werden. Die genannten 'Tags' dienen dabei als Container - entweder für tatsächlichen Inhalt oder für weitere Tags. Für ein effizientes Bearbeiten von Dokumenten kann komfortabel im Tag-Baum eines Dokumentes navigiert werden. Im Standardfall wird vorhandener Inhalt editiert und erweitert - hierzu kann man, wie von anderen Textverarbeitungsprogrammen gewohnt, an die gewünschte Textstelle klicken und tippen. Der Editor zeigt zu jedem Element, welches den Focus besitzt, zu welchem DocBook-Tag es in der Hierarchie gehört (der entsprechende Pfad wird in der Fußzeile des Editors sowie in der ContentMap bzw. TOC in der Seitenleiste angezeigt):

Abbildung 4.5. Aktueller Knoten



In der Seitenleiste kann zwischen den Registerkarten ContentMap und TOC umgeschaltet werden. In der Ansicht ContentMap werden die XML-Knoten wie auch der Inhalt hierarchisch dargestellt, in der Ansicht TOC werden lediglich die Knoten mit eventuell vorhandenen <title>-Tags dargestellt. In beiden Ansichten kann komfortabel durch das Dokument navigiert werden, momentan nicht benötigte Abschnitte können ein- bzw. aufgeklappt werden.

Praktisches

Ein schnelles und effizientes Editieren mit Serna Free wird durch viele praktische Helfer unterstützt.

- Neues Element an aktueller Position einfügen: **Strg+Return**


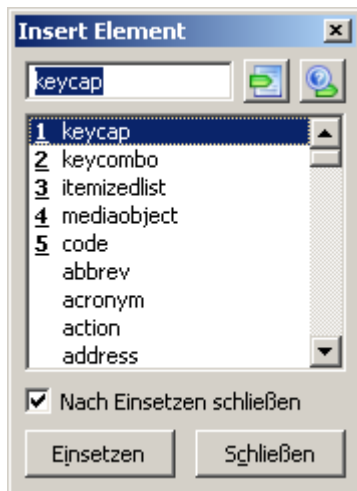
- Textfragment mit einem Element "ummanteln" (beispielsweise, um einen Begriff zu verlinken): Text markieren und **Strg+Return**
- Navigieren in den Elementen in der Edit-Ansicht: Cursor-Tasten. Dabei wird das fokussierte Element hervorgehoben.
- Kommentare einfügen: **Strg+I**. Kommentare werden nicht publiziert und können somit für eigene Anmerkungen oder Marker innerhalb des Dokuments genutzt werden. Auch können durch Kommentare "Diskussionen" mit anderen Redakteuren geführt werden. Sind Kommentare vorhanden, werden diese mit einem Kommentar-Symbol  angezeigt. Ein Doppelklick auf das Symbol öffnet ein Fenster mit dem Inhalt des Kommentars.
- Inline-Content: Die Tastenkombination **Strg+Return** öffnet das Auswahlfenster für ein neues Element an der aktuellen Cursor-Position. Mit den Cursortasten kann direkt ein Element ausgewählt und mit Return bestätigt werden. Die - falls vorhanden - in dieser Sitzung schon einmal genutzten Elemente werden dabei zu Beginn der Element-Liste dargestellt:

Abbildung 4.6. Dialog Insert Element



Hat man ein Inline-Element eingefügt, im Falle obiger Abbildung einen Tastendruck beschrieben, so kann nach dem Ende des Elements durch die Taste Ende wieder in das ursprüngliche Element gewechselt werden. Dabei wird das Inline-Element automatisch geschlossen (`<keycap> . . . </keycap>`).

- Tastatur-Shortcuts

Eine Übersicht der Tastatur-Shortcuts finden Sie unter [Help → Shortcut Keys...](#)

- Abbildungen und Tabellen, die nicht in Verzeichnissen aufgeführt werden sollen

Abbildungen und Tabellen werden automatisch in den Abbildungs- und Tabellenverzeichnissen aufgeführt. Soll dies bei einzelnen Abbildungen/Tabellen nicht geschehen, so sind diese als `informalfigure` bzw. `informaltable` ausgezeichnet werden.

Arbeiten mit Elementen

Das DocBook-Dokument wird über DocBook-Elemente aufgebaut. Dabei werden drei Arten von Elementen unterschieden:

1. Strukturierende Elemente

Strukturierende Elemente geben den Aufbau bzw. die Gliederung und die Dokumentart des Dokuments wieder. Darunter fallen z.B. `book`, `chapter`, `glossary`, `index` usw.

2. Block-Elemente

Block-Elemente zeichnen den Inhalt aus und erzwingen i.d.R. einen Zeilenumbruch vor und nach dem Element. Darunter fallen bspw. `para`, `itemizedlist`, `figure` usw.

3. Inline-Elemente

Inline-Elemente "fliessen mit dem Text" und erzwingen keinen Zeilenumbruch. Ein `xref` oder `link` gehört beispielsweise dazu.

Beim Editieren muß der Redakteur den Text in adäquate DocBook-Elemente "verpacken" - dazu stehen eine fast unüberschaubare Menge an unterschiedlichen Elementen zur Verfügung. Eine Auflistung sämtlicher Elemente ist auf der DocBook-Webseite [<http://www.docbook.org/tdg/en/html/part2.html>] untergebracht.

Von den über 400 gelisteten Elementen werden jedoch meist nur eine sehr viel geringere Anzahl für die eigenen Dokumentationen benötigt. In Anhang A, *DocBook-Elemente* wird die Verwendung und Bedeutung der wichtigsten Elemente erläutert.

Markup *escapen*

Anmerkung

Soll DocBook- oder XML-Markup lediglich ausgegeben statt ausgewertet werden, so müssen Teile davon "escaped" werden. Beispielsweise die Zeichen, die ein DocBook-/XML-Element einleiten (<) und beenden (>), müssen als Entity notiert (< sowie >) werden, damit sie beim Generieren der Ausgabeformate auch "buchstäblich" ausgegeben werden.

Bei längeren Textpassagen, die eventuell XML-Markup enthalten, kann die gesamte Textpassage als CDATA markiert und escaped werden:

```
<section>
  <title>Beispiel CDATA</title>
  <para>Die DocBook-Elemente dieses Abschnitts werden ausgegeben statt inte
</section>
```

CDATA-Quellcode.

```
<![CDATA[
  zu escapender Quellcode
]]>
```

IDS

Mit einer `id` kann ein eindeutiger Identifizierer innerhalb eines Dokument für ein Element vergeben werden. Über Links wie `link` oder `xref` können diese identifizierbaren Elemente angesprungen werden. Im Folgenden ein Quelltext-Beispiel:

Beispiel 4.1. Verlinkung von Elementen

```
<section id="anker">
  <title>Beispieltitel</title>
  <para>Nullam consequat. Phasellus iaculis, urna non sodales pulvinar, tellus
  pellentesque mi, eu tempor nisl nisi non lectus. </para>
  <para>Nam accumsan tincidunt elit. Suspendisse rutrum luctus nisi. Siehe a
</section>
```

Kommentare

Kommentare im Sinne von XML-Kommentaren erscheinen nicht in den Ausgabeformaten.

Anmerkung

Kommentare im DocBook-Quelltext sind für die Zusammenarbeit mit anderen Redakteuren sowie für eigene Notizen im Dokument sinnvoll und zur Kommentierung wird ausdrücklich aufgerufen. Auf Quelltext-Ebene wird ein Kommentar in XML-Syntax (`<!-- Kommentar /-->`) eingegeben.

Validieren

zpub kann ausschließlich valide DocBook-XML-Dokumente gemäß der *DTD* verarbeiten. Sollte der verwendete Editor nicht sicherstellen, dass nur valide Dokumente produziert werden, muß dies der Redakteur übernehmen. Hierfür stehen unterschiedliche Hilfsmittel bereit. Unter einem GNU/Linux-Betriebssystem beispielsweise via:

```
xmllint --noout --postvalid Dateiname.xml
```

Anmerkung

Bevor zpub versucht, die unterschiedlichen Ausgabeformate eines Dokuments zu generieren, wird intern `xmllint` zum Überprüfen der Syntax der `xml`-Datei genutzt. Die Ausgaben hierbei - insbesondere die Fehlermeldungen - informieren über eventuell gefundene Probleme mit der Docbook-XML-Datei des Dokuments. Für jede Revision eines Dokuments kann die Logdatei, die auch die `xmllint`-Ausgaben enthält unter der folgenden URL abgerufen werden:

```
https:// \$Instanz .zpub.de/ \$Dokument /archive/ \$Revision / \$Style /zpub-render.log
```

Wiederverwendbarer Inhalt

Für einige Text-/Dokumentfragmente bietet es sich eventuell an, diese ausserhalb eines spezifischen Dokuments zu pflegen, da dieser Inhalt in mehreren Dokumenten vorkommen soll und nur zentral an einer Stelle gepflegt werden soll. So kann beispielsweise ein Glossar verwaltet werden.

Dazu kann man auf oberster Verzeichnisebene der lokalen Arbeitskopie ein Verzeichnis `common` anlegen. Inhalte daraus können in den Handbüchern verwendet werden: Bilder wie gewohnt, Dokumentfragmente per `XInclude`. Werden Dateien in `common` geändert werden von zpub genau die betroffenen Handbücher neu gebaut.

Kapitel 5. Ausgabeformate

zpub generiert automatisch nach jedem SVN-commit die konfigurierten Ausgabeformat - z.B. PDF, XHTML, CHM und EPUB. Das Layout und Design der Ausgabeformate wird über Stylesheets bestimmt und kann individuell angepasst werden. So können die Dokumente beispielsweise mit dem eigenen Branding/Corporate Design versehen werden. Die Anpassungen werden im Verzeichnis `style/` der zpub-Instanz vorgenommen.

PDF

PDF Metadaten

Um die PDF Metadaten (siehe PDF Reader: Datei → Eigenschaften... vollständig abzubilden, sind in der Docbook-Datei die folgenden Angaben zu machen:

Für die "Schlüsselworte" (es sind mehrere keywords möglich):

```
<bookinfo>
...
<keywordset>
  <keyword>Qualitätsmanagement</keyword>
  <keyword>DIN EN ISO 9001:2008</keyword>
</keywordset>
...
</bookinfo>
```

Für den "Betreff" (es sind mehrere subjectterms möglich):

```
<bookinfo>
...
<subjectset>
<subject>
  <subjectterm>Qualitätsmanagement Handbücher</subjectterm>
</subject>
</subjectset>
...
</bookinfo>
```

Landscape figure

Eine figure im Querformat unter Ausnutzung des maximal vorhandenen Viewports ausgeben.

Markup.

```
<figure>
  <?landscapeFigure?>
  <title>Landscape <code>figure</code></title>
  <mediaobject>
    <imageobject>
      <imagedata fileref="images/20091010_144209_preview.jpg" scalefit="1" v
    </imageobject>
  </mediaobject>
</figure>
```

Abbildung 5.1. Landscape figure



Landscape table

Eine table im Querformat unter Ausnutzung des maximal vorhandenen Viewports ausgeben.

Markup.

```
<table frame="none" orient="land" width="100">  
  <title>Landscape <code>table</code></title>  
  ...  
</table>
```

Tabelle 5.1.1. Landscape table

Abkürzung	Ausgeschrieben
QMB	Qualitätsmanagementbeauftragter
VA	Verfahrensweisung
QMH	Qualitätsmanagementhandbuch
QMS	Qualitätsmanagementsystem
FB	Formblatt
FB	Formblatt
FB	Formblatt
FB	Formblatt
QMB	Qualitätsmanagementbeauftragter
VA	Verfahrensweisung
QMH	Qualitätsmanagementhandbuch
QMS	Qualitätsmanagementsystem
FB	Formblatt
FB	Formblatt
QMB	Qualitätsmanagementbeauftragter
VA	Verfahrensweisung
FB	Formblatt
QMB	Qualitätsmanagementbeauftragter
VA	Verfahrensweisung

Anhang A. DocBook-Elemente

DocBook-Elemente: Übersicht

Dieser Anhang stellt einige der gebräuchlichsten DocBook-Elemente vor.

- `book` - Definiert ein Buch; hier: ein Dokument.
- `preface` - Ein Vorwort. Kann nur direkt für den `book`-Tag verwendet werden.
- `chapter` - Ein Kapitel. Kapitel können nicht verschachtelt werden. Sollen Kapitel weiter untergliedert werden, so geschieht dies durch eine oder mehrere `sections`.
- `section` - Eine Unterteilung eines Kapitels.
- `title` - Ein Titel. Titel sind u.a. für `chapter` und `section` sowie Abbildungen, Tabellen etc. notwendig. Diese Titel erscheinen ebenfalls in den generierten Verzeichnissen.
- `para` - Ein Absatz. Der Standard-Tag für Fließtext.
- `itemizedlist` - Eine ungeordnete (Bullet-)Liste.
- `orderedlist` - Eine geordnete (standardmäßig durchnummerierte) Liste.
- `figure`, `screenshot`, `figure(screenshot)` - Abbildungen, Fotos, Screenshots u.ä. Siehe unten.
- `table` - Eine Tabelle. Siehe unten.
- `keycap`, `keySYM` - Eine Taste.
- `keycombo` - Eine Tastenkombination, bestehend aus Tasten im `keycap` Tag.
- `ulink` - Ein externer Link. Siehe unten.
- `xref`, `link` - Ein interner Link. Siehe unten.

Typografisches¹.

- Für kursiv gesetzten Text kann `<emphasis>...</emphasis>` verwendet werden.
Ausgabebeispiel: *Dieser Text ist <emphasis>*.
- Für Text in Fettdruck kann `<emphasis role="strong"> ...</emphasis>` verwendet werden.
Ausgabebeispiel: **Dieser Text ist fett gedruckt..**
- Für Text in dicktengleicher Schrift bzw. Monospace-Schriften können, je nach inhaltlicher Anforderung, unterschiedliche Elemente verwendet werden. Beispielsweise:
 - `<literal>...</literal>`
 - `<computeroutput>...</computeroutput>`
 - `<filename>...</filename>`
 - `<code>...</code>`
 - `<programlisting>...</programlisting>`

Ein DocBook-Dokument enthält ausschließlich inhaltliches Markup und kein Markup zur Präsentation. Zeilen- und Seitenumbrüche werden daher im Stylesheet für das jeweilige Ausgabemedium berechnet. Zeilen- oder Seitenumbrüche können jedoch erzwungen werden:

¹DocBook dient der inhaltlichen Auszeichnung des Textes und nicht der Formatierung. Daher kommen diese typographischen Elemente nur dort zum Einsatz, wo es inhaltlich erforderlich ist - beispielsweise um Begrifflichkeiten oder Vorgänge zu betonen.

erzwungener Zeilenumbruch `<?linebreak?>`

erzwungener Seitenumbruch `<?hard-pagebreak?>`

Links. Für Verlinkungen in einem Docbook-Dokument stehen drei Möglichkeiten zur Verfügung:

1. `link` : Verlinkung
2. `ulink` : Verlinkung auf externe URL
3. `xref` : dokumentinterne Verlinkung mit bei Bedarf automatisch generiertem Linktext

Für die Verlinkung auf andere Bereiche im selben Dokument ist der Zielbereich mit einer `id` zu identifizieren.

IDs

Mit einer `id` kann ein eindeutiger Identifizierer innerhalb eines Dokument für ein Element vergeben werden. Über Links wie `link` oder `xref` können diese identifizierbaren Elemente angesprungen werden. Im Folgenden ein Quelltext-Beispiel:

Beispiel A.1. ID-Vergabe/Verlinkung von Elementen

```
<section id="anker">
  <title>Beispieltitel</title>
  <para>Nullam consequat. Phasellus iaculis.</para>
  <para> Nam accumsan tincidunt elit. Suspendisse rutrum luctus nisi. Siehe a
</section>
```

application Programmname

Programmnamen können mit den entsprechenden DocBook-Elementen ausgezeichnet werden.

- **Beispiel A.2. Programmnamen**

Dateien vom Typ `odt` können mit OpenOffice oder LibreOffice bearbeitet werden.

- **Quelltext.**

Dateien vom Typ `<code>odt</code>` können mit `<application>OpenOffice</application>`

callout Annotationen

Callouts sind mit Positionszahlen und entsprechenden Erläuterungen versehenende Bereiche in Grafiken oder Texten

```
[boot loader]
  timeout=30
  default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
  multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Profession
```

Der `timeout` definiert die Zeit bis zum automatischen Booten des `default`-Eintrags, wenn mehrere Betriebssystemeinträge vorhanden sind.

Definiert die möglichen zu startenden Betriebssysteme.

figure Grafikeinbindung

Abbildungen können "generisch" via `figure` oder - in den meisten Fällen spezifischer - über `screenshot` eingebunden werden.

Bei der Einbindung von Grafiken bzw. Screenshots sollte auf die folgenden Punkte geachtet werden:

Größe/Dimensionen Grafiken sollten nicht breiter als 550 Pixel sein. Diese Empfehlung gilt, wenn als Ausgabeformat u.a. PDF gewählt ist und somit eine fixe (meist DIN A4) Breite und Höhe gegeben ist. Gegebenenfalls können passende Ausschnitte gewählt werden, oder die Abbildung in einer Grafikbearbeitungssoftware skaliert werden.

Ist die Abbildung größer als der dafür zur Verfügung stehende Raum, wird - in der PDF-Ausgabe - die Grafik rechts und/oder unten abgeschnitten. Sollen auch größere Abbildungen unbeschnitten dargestellt werden, müssen diese entweder im Vorfeld mit einer Grafikbearbeitung skaliert werden, oder es wird das Attribut `width="X%"` für das Element `<imagedata>` genutzt. Siehe auch Beispiel A.3, „Skalierungsbeispiele für Abbildungen“ und „Landscape figure“.

Dateiformat Bevorzugt wird das Dateiformat Portable Network Graphics (PNG) gewählt. Dies wird von dem von zpub genutzten fo-Formatierer `fop` nativ unterstützt, bietet "echte" Transparenz via Alphakanal und verlustfreie Kompression.

Weiter ist auch die gute Unterstützung von Scalable Vector Graphics (SVG, deutsch skalierbare Vektorgrafiken) zu nennen.

Auflösung Wenn möglich sollten die Abbildungen und Screenshots in einer Auflösung von 96dpi vorliegen (sofern die Bildbearbeitung die Auflösung in die Grafikdatei einbettet), dies garantiert qualitativ beste Ergebnisse. Ein Screenshot unter Microsoft Windows XP/7™ hat eine Auflösung von 96dpi.

Abbildung A.1. Screenshot (unskaliert: 698x406px, 96dpi)



Eine Verdoppelung der DPI - bei gleichbleibenden Dimensionen - führt zu kompakteren Grafiken, die im Ausdruck in Größe und Schärfe/Lesbarkeit als passender empfunden werden. Alternativ können auch - sollte die Dimensionen der Ausgabe zu klein geraten - geringere Erhöhungen der DPI genutzt werden. Die geänderte Auflösung muß auch im Dateiformat gespeichert werden, was je nach Bildbearbeitungssoftware und Format unterschiedlich zu erreichen ist. Beim Speichern ist darauf zu achten, dass als Format "PNG" und "Auflösung speichern" gewählt ist:

Abbildung A.2. Screenshot (unskaliert: 698x406px,192dpi)

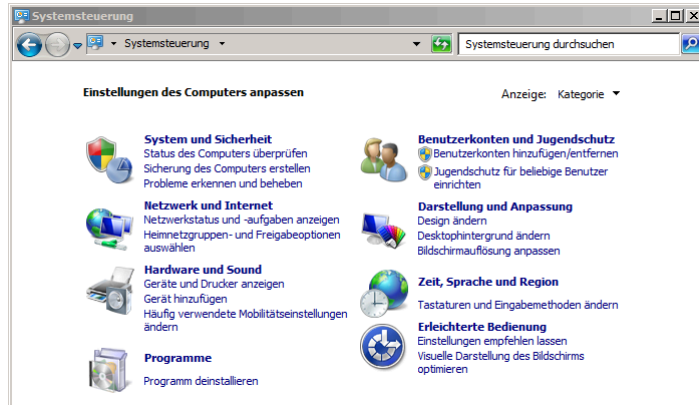


Abbildung A.3. Screenshot (kein ClearType; unskaliert: 698x406px,192dpi)



Screenshots

Screenshots können auf unterschiedlichste Weise erzeugt werden - über die Taste Druck beispielsweise wird der aktuelle Bildschirminhalt in der Zwischenablage gespeichert, über Programme wie XnView kann der Bildschirm- oder Fensterinhalt präziser erfasst und zumindest rudimentär bearbeitet werden. Ab Microsoft™ Windows 7 können sämtliche Screenshoterstellung (einzelnes Fenster, gesamter Desktop, individueller Bereich etc.) über das zu Windows gehörende Snipping Tool komfortabel erstellt werden.

Die Bildqualität (Lesbarkeit von Schriften) von Screenshots wird durch eine Skalierung stark beeinträchtigt und sollte daher vermieden werden. Eine Erhöhung des DPI-Werts dagegen bringt erzeugt in der PDF-Ausgabe qualitativ hochwertige, jedoch in ihren Dimensionen kleinere Abbil-

dungen. Nicht-grafische Abbildungen wie Photos können dagegen relativ problemlos skaliert werden.

Beispiel A.3. Skalierungsbeispiele für Abbildungen

Abbildung A.4. Pixelleiste (unskaliert)

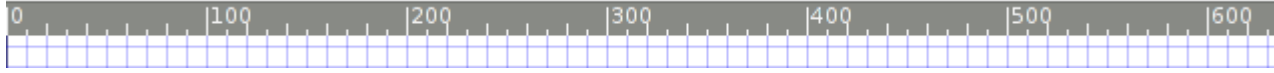


Abbildung A.5. Pixelleiste (Breite: 80%)

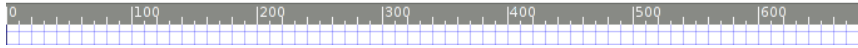
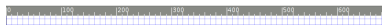
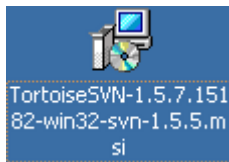


Abbildung A.6. Pixelleiste (Breite: 5cm)



Einbindung via figure

Abbildung A.7. Abbildung figure

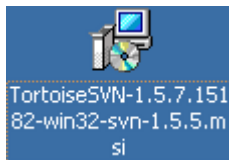


Beispiel A.4. Abbildung ohne Untertitel

```
<figure>
  <title>Abbildungstitel</title>
  <mediaobject>
    <imageobject>
      <imagedata fileref="images/dateiname.png" />
    </imageobject>
  </mediaobject>
</figure>
```

Einbindung via figure mit Untertitel (caption)

Abbildung A.8. Beispielabbildung mit Untertitel figure



Untertitel des Abbildung - caption

Beispiel A.5. Abbildung mit Untertitel (caption)

```
<figure>
  <title>Abbildungstitel</title>
  <mediaobject>
    <imageobject>
```

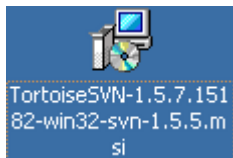
```

    <imagedata fileref="images/dateiname.png" />
  </imageobject>
  <caption>
    <para>Untertitel des Abbildung - caption</para>
  </caption>
</mediaobject>
</figure>

```

Einbindung via screenshot

Abbildung A.9. Screenshot-Titel



Beispiel A.6. Screenshot

```

<figure>
<title>Screenshot-Titel</title>
  <screenshot>
    <mediaobject>
      <imageobject>
        <imagedata fileref="images/dateiname.png" />
      </imageobject>
    </mediaobject>
  </screenshot>
</figure>

```

footnote Fußnote

Beispiel A.7. Fußnoten-Beispiel

Maecenas eu ipsum. Praesent sed velit. Nullam consequat². Phasellus iaculis, urna non sodales pulvinar, tellus ligula pellentesque mi, eu tempor nisl nisi non lectus.

Quellcode <footnote>Fußnotentext</footnote>

glossary Glossar

Zur zentralen Erläuterung von Begrifflichkeiten kann ein Glossar angelegt werden.

Das Glossar besteht aus einem Bereich (glossary), das die Begrifflichkeiten und deren Erläuterung aufnimmt, sowie aus den Links im Dokument auf diese Erläuterungen.

Das Glossar ist aus den folgenden Elementen aufgebaut:

Elemente eines Glossars

<glossary>	Container-Element für das Glossar. Normalerweise am Ende eines Dokuments.
<glossentry id=" ">	Nimmt den eigentlichen Glossar-Begriff (<glossterm>) auf sowie die Beschreibung (<glossdef>) auf.

²Hier wird lediglich Blindtext verwendet.

`<glossterm id="">` Verweist im Fließtext auf die Begriffserläuterung im Glossar. Über die `id` werden die Begriffe verlinkt.

Anmerkung

Um ein konsistentes Benennungsschema zu erhalten und mögliche Inferenzen mit anderen `ids` gering zu halten, können Glossar-`ids` mit einem Präfix wie "gloss-" versehen werden.

Das Glossar wird automatisch alphabetisch sortiert (dies kann über einen Stylesheet-Parameter definiert werden).

Beispiel A.8. Quelltext-Beispiel: Glossar

Verweis innerhalb des Dokuments auf den Glossareintrag:

```
<!-- ... -->
<para>Grundlage bildet die <glossterm linkend="gloss_docbook">DocBook</glossterm>
<!-- ... -->
```

Definition des Glossareintrags im Glossar:

```
<!-- ... -->
<glossary>
<!-- ... -->
<glossentry id="gloss_docbook">
  <glossterm>DocBook</glossterm>
  <glossdef>
    <para>Definition von DocBook...</para>
  </glossdef>
</glossentry>
<!-- ... -->
<glossary>
<!-- ... -->
```

Grundlage bildet die *DocBook*-Datei

indexterm Indexeintrag

Ein `Indexterm` definiert einen Begriff, der im automatisch generierten Index am Ende des Dokuments ("Stichwortverzeichnis") aufgeführt werden soll. Es können dabei auch `Indexterme` verschachtelt werden, um Zusammengehörigkeiten von Begrifflichkeiten herauszuarbeiten. Es sind dabei drei Ebenen möglich. Der `Indexterm` selber wird im Ausgabeformat nicht gerendert, er dient lediglich als Anker für den Begriff im Stichwortverzeichnis. Der `indexterm` erzeugt keine Ausgabe an der Stelle, an der er eingefügt wird, er wird quasi an einen vorhandenen Begriff, ein vorhandenes Kapitel o.ä. angehängt (siehe folgendes Quellcode-Beispiel):

Beispiel A.9. indexterm Textbeispiel

GNU ist das im Rahmen des GNU-Projekts in Entwicklung befindliche, vollständig freie Betriebssystem. Die Bezeichnung ist ein rekursives Akronym von GNU's Not Unix. Es wird mit dem Ziel entwickelt, eine vollständig freie Alternative für Unix zu bieten, und steht unter der GNU General Public License (GPL). GNU ist POSIX-kompatibel. Es besteht aus dem Mach Microkernel, den Services GNU Hurd und der weiteren Software des GNU-Projekts.

Beispiel A.10. Indexterm Quellcodebeispiel


```
<section>
```

```
<title>GNU</title>
<para>
GNU ist das im Rahmen des GNU-Projekts<indexterm><primary>GNU</primary></indexterm>
</para>
</section>
```

Anmerkung

Für die Ausgabe siehe Stichwortverzeichnis.

inlinemediaobject Grafikeinbindung (inline)

Abbildungen und Grafiken können auch *inline* genutzt werden, beispielsweise um Schaltflächen im Fließtext unterzubringen. Beispielsweise, um einen Button  zu beschreiben.

Per `inlinemediaobject` eingebundene Grafiken erhalten keinen Titel und werden in Verzeichnissen nicht aufgeführt.

Beispiel A.11. inlinemediaobject

```
<inlinemediaobject>
  <imageobject>
    <imagedata fileref="images/dateiname.png" />
  </imageobject>
</inlinemediaobject>
```

itemizedlist Ungeordnete Liste

- Listenpunkt
- Listenpunkt
- Listenpunkt

keycap/keysym Tastensymbole

Für die Darstellung von Tastensymbolen oder Tastenkombinationen - vor allem in der PDF-Ausgabe - sind die gängigsten "Tastatursymbole" hinterlegt.

Die Folgenden Tastatursymbole werden aktuell unterstützt:

Tabelle A.1. Tastensymbole

Taste	Quellcode	Ausgabe
Return	<code><keycap>Return</keycap></code>	Return
Escape/Esc	<code><keycap>Esc</keycap></code>	Esc
Steuerung/Strg	<code><keycap>Strg</keycap></code>	Strg
Umsch/Shift	<code><keycap>Umsch</keycap></code>	Umsch
Alt	<code><keycap>Alt</keycap></code>	Alt
Einfg/Einfügen	<code><keycap>Einfg</keycap></code>	Einfg

Taste	Quellcode	Ausgabe
Entf/Entfernen	<code><keycap>Entf</keycap></code>	Entf
Cursor: Links, Rechts, Oben, Unten	<code><keycap>Links</keycap>, <keycap>Rechts</keycap>, <keycap>Oben</keycap>, <keycap>Unten</keycap></code>	Links, Rechts, Oben, Unten
Ende	<code><keycap>Ende</keycap></code>	Ende

Bei der Eingabe ist auf korrekte Groß-/Kleinschreibung laut Spalte "Quellcode" zu achten.

Anmerkung

Sollten für Ihr Projekt Tastensymbole fehlen, implementieren wir [mailto:zpub.de] diese gerne für Sie.

link Allgemeiner Link

Soll der Linktext nicht automatisch generiert werden, bietet sich das Element `link` an. Es erwartet die Angabe einer `id` für das Linkziel und umschließt den zu verlinkenden Text:

```
<link linkend='ziel-id'>Linktext</link>
```

Beispiel. Donec eu felis ac urna aliquam ultrices pellentesque non diam. Phasellus imperdiet lacus non tortor elementum ornare. Ut aliquet semper rutrum. Cras dapibus auctor dui ac suscipit.

menuchoice Menüstrukturen

Das Navigieren in (verschachtelten) Menüs wird über entsprechende DocBook-Elemente abgebildet. Dabei kann auch ein eventuell vorhandener Tastatur-Shortcut angegeben werden.

- **Beispiel A.12. menuchoice und shortcut**

Zum Ausdrucken des Dokuments wählen Sie Datei → Drucken... (**Strg+P**).

- **Quelltext.**

```
<para>Zum Ausdrucken des Dokuments wählen Sie
<menuchoice>
  <shortcut>
    <keycombo><keysym>Strg</keysym><keysym>P</keysym></keycombo>
  </shortcut>
  <guimenu>Datei</guimenu>
  <guimenuitem>Drucken...</guimenuitem>
</menuchoice>.
</para>
```

note Anmerkungen

Anmerkung

Mauris nec nisl sed metus sagittis lobortis. Aenean nunc magna, bibendum sollicitudin, luctus id, posuere vel, neque. Cras malesuada congue neque.

```
<note>
  <para></para>
</note>
```

Anmerkung mit Titel

note, tip sowie warning können optional auch einen title enthalten.

```
<note>
  <title></title>
  <para></para>
</note>
```

orderedlist Geordnete Liste

1. Listenpunkt
2. Listenpunkt
3. Listenpunkt

table Tabellen

Tabellen können auf zwei Arten definiert werden: nach dem HTML-Modell oder nach dem CALS-Modell.

HTML-Tabelle. Tabellen nach dem HTML-Modell sind nach dem Schema `table - tr - td` aufgebaut:

Tabelle A.2. Beispieltabelle (HTML-Tabelle):

Spaltentitel	Spaltentitel
Test	Test
Test	Test
Test	Test
Test	Test

Tipp

Sollen bei der PDF-Ausgabe alternierende Hintergrundfarben für die Tabellenzeilen ausgegeben werden, ist für das `table`-Element das Attribut `tabstyle="striped"` anzugeben. Die HTML-Ausgabe verwendet in jedem Fall alternierende Hintergrundfarben.

Tabelle A.3. Beispieltabelle mit alternierenden Zeilehintergrundfarben

Labelname im G20/G21/G22-Befehl	Registerwert bei Programmausführung	Programmfortsetzung bei Aufruf von UP
G20\$TOOL-%R5%-TEST\$	R5 = 18	\$TOOL-18-TEST\$
G20\$TOOL-%R5%-TEST\$	R5 = 1444	\$TOOL-1444-TEST\$
G22E5\$TOOL-%R5%-TEST\$	R5 = 333	G98\$TOOL-333-TEST\$
G20\$%R1004%\$	R1004 = 18	\$18\$
G22\$%R163%\$	R163 = 15	G98\$15\$

Anmerkung

Die CSS-Regeln für die Tabellendarstellung in der HTML-Ausgabe werden von allen modernen Browsern korrekt ausgeführt. Lediglich Microsoft Internet Explorer bis zu Version 8 ist nicht in der Lage, bestimmte Regeln zu interpretieren.

CALS-Tabelle. Tabellen nach dem CALS-Modell sind nach dem Schema `table - row - entry` aufgebaut:

Tabelle A.4. CALS-Tabelle

Spaltentitel	Spaltentitel
Test	Test
Test	Test
Test	Test
Test	Test

tip Tipps

Tipp

Mauris nec nisl sed metus sagittis lobortis. Aenean nunc magna, bibendum sollicitudin, luctus id, posuere vel, neque. Cras malesuada congue neque.

```
<tip>
  <para></para>
</tip>
```

trademark Produktname

Produktname können mit den entsprechenden DocBook-Elementen ausgezeichnet werden.

- **Beispiel A.13. Produktbezeichnungen**

Dateien vom Typ `docx` und `doc` können mit Microsoft Word™ bearbeitet werden.

- **Quelltext.**

Dateien vom Typ `<code>docx</code>` und `<code>doc</code>` können mit `<trademark>`

ulink Externer Link

Für die Verlinkung von externen Ressourcen wird mit `ulink` gearbeitet. `ulink` erwartet lediglich das Attribut `url` und umschließt den zu verlinkenden Text. Ist der zu verlinkende Text leer (`<ulink url="..." />` oder `<ulink url="..."></ulink>`) wird die URL als Linktext ausgegeben.

Beispiel. Donec eu felis ac urna aliquam [<http://www.zpub.de/de/>] ultrices pellentesque non diam. Phasellus imper- diet lacus non tortor elementum ornare. Ut aliquet semper rutrum. Cras dapibus <http://www.breitnerundbreitner.de/> auctor dui ac suscipit.

Wird die Ausgabe der URL im Fließtext unterdrückt, da ein individueller Linktext angegeben ist, wird die verlinkte URL in der PDF-Ausgabe als Fußnote dargestellt.

warning Warnungen

Warnung

Cras ut tortor. Suspendisse semper libero nec massa. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent at purus. Vivamus nulla magna, molestie vitae, adipiscing a, fermentum et, lacus.

```
<warning>
  <para></para>
</warning>
```

Anmerkung

Statt `warning` kann auch `important` oder `caution` genutzt werden, auf die Ausgabe hat dies jedoch keinen Einfluss.

xref Interner Link

Mit `xref` kann ohne die Angabe eines Linktexts auf einen mit einer `id` referenzierten Bereich verlinkt werden, wobei der Linktext aus dem Zielbereich (normalerweise dem `title`-Element generiert wird. Wird der Link für die PDF-Ausgabe generiert, wird dem Linktext noch die Seitenangabe der Zielseite hinten angestellt.

Beispiel A.14. xref-Vergabe/Verlinkung von Elementen

```
<section id="anker">
  <title>Beispieltitel</title>
  <para>Nullam consequat. Phasellus iaculis.</para>
  <para> Nam accumsan tincidunt elit. Suspendisse rutrum luctus nisi. Siehe auch
</section>
```

Der folgende Link ist im Quellcode als `<xref linkend="docbook_elemente" />` angegeben, die Ausgabe dagegen sieht wie folgt aus: Anhang A, *DocBook-Elemente*. Noch ein Beispiellink: siehe auch Kapitel 2, *Verwaltung der Dokumente*.

Anhang B. FAQ / Troubleshooting

B.1. SVN

B.1.1. Keine Verbindung zum SVN-Server

Sollte für die Internetverbindung ein Proxy notwendig sein, so ist dieser auch im SVN-Client einzutragen:

- Rechtsklick
- Kontextmenüeintrag 'TortoiseSVN'
- Kontextmenüeintrag 'Settings'/'Einstellungen'
- Im erscheinenden Einstellungsdialog unter 'Network'/'Netzwerk' die korrekten Proxy-Einstellungen für Ihr Netzwerk vornehmen. Diese Einstellungen erhalten Sie von Ihrem Netzwerk-Administrator/ EDV-Ansprechpartner.

B.2. Ausgabe

B.2.1. Es wird keine Ausgabe erzeugt

Wenn nach dem Übertragen des Dokuments auf dem Webserver keine neue Version generiert wird, kann das folgenden Grund haben:

Das Dokument ist nicht valide. Dies sollte nicht passieren, wenn es mit einem XML-Editor wie XMLmind bearbeitet wurde. Bearbeiten es dagegen mit einem generellen Text-Editor, können ungültige XML-Dateien erzeugt werden – etwa wenn ein Tag nicht geschlossen wird. Die Logdatei im Webfrontend enthält Angaben, wo in der Datei das Problem zu finden ist.

B.2.2. Es fehlen Bilder in der Ausgabe

Es wurde ein Bild in das Dokument eingefügt, welches dann aber nicht in der Ausgabe auftaucht. Dies kann vorkommen, wenn der Dateiname des Bildes im Dokument nicht zum wirklichen Ort des Bildes passt. Ist beispielsweise das Bild `bild.jpg` im Ordner `images` gespeichert, so muss der Verweis `images/bild.jpg` heißen.

Ein weiterer Grund für ein fehlendes Bild ist, wenn vergessen wurde, das Bild zum SVN-Repository hinzuzufügen. Dies erkennt man am Fragezeichen-Symbol über dem Dateisymbol. In diesem Fall kann über das Kontext-Menüs des Windows-Explorers die Funktion SVN Hinzufügen... die Datei hinzugefügt und anschließend committed werden.

B.2.3. Die SVN-Revisionsangaben werden nicht aktualisiert

In der Draft-Ansicht enthalten die Dokumente Informationen zur SVN-Revisionsnummer, zum Zeitpunkt der letzten Änderung (Zeitzone UTC) und zum letzten Autor. Wenn diese Daten nach Änderungen am Dokument nicht oder nicht mehr aktualisiert werden, hat das einen der folgenden Gründe:

Die SVN-Properties (siehe SVN-Eigenschaften importieren) wurden für die XML-Quelldatei des Dokuments nicht importiert. Dieser Schritt kann auch nachträglich wie beschrieben durchgeführt werden.

Beim Speichern wurde wegen zu geringer Zeilenlänge die Markierung auseinandergerissen. Eventuell sind Einstellungen zur Zeilenlänge im Editor so anzupassen, dass die `<release-info>`-Angabe nicht umgebrochen wird.

B.2.4. Die Ausgabe von PDF/HTML/etc gefällt nicht

Wenden Sie sich mit Ihren Wünschen an uns [<mailto:mail@zpub.de>] - wir unterstützen Sie gerne.

Anhang C. XMLmind XML-Editor

Für das Editieren der DocBook-XML-Dokumente können wie schon erwähnt auch andere XML- oder Text-Editoren genutzt werden.

Die Pflege einer eigenen Dokumentation wurde aufgrund der Wahl von Serna Free als Referenz-Editor entfernt. Statt dessen verweisen wir auf die ausführliche Dokumentation [http://www.xmlmind.com/xmleditor/_distrib/doc/help/index.html], die vom Hersteller XMLmind [<http://www.xmlmind.com/>] gepflegt wird. gepflegt.

Glossar

Spezielle Begrifflichkeiten sind im Folgenden aufgeführt.

A bis Z

DocBook	DocBook ist ein Dokumentenformat, das in einer für SGML und XML vorliegenden Dokumenttypdefinition (DTD) festgelegt ist. Es eignet sich besonders zur Erstellung von Büchern, Artikeln und Dokumentationen im technischen Umfeld (Hardware oder Software). DocBook ist ein offener Standard, der von der Organization for the Advancement of Structured Information Standards (OASIS) gepflegt wird.
Arbeitsversionen	Eine Arbeitsversion eines Dokuments ist vergleichbar mit einem Entwurf, das Änderungen enthält, die noch nicht freigegeben sind. Diese Dokumente sind in den Ausgabeformaten immer mit genauen technischen Angaben zur Revision sowie einem "Draft"-Wasserzeichen gekennzeichnet und nur für den internen Gebrauch bestimmt. Erst wenn eine Arbeitsversion freigegeben wird, ändert sie ihren Status in eine "finale Version", welche auch nicht mehr seitenweise (im Falle des PDF-Ausgabeformats) die Revisionsangaben und das "Draft"-Wasserzeichen enthält.
DocBook-Tag	Siehe Tag.
Repository	Bei SVN werden Quellcodedateien oder andere Dateien aus dem Repository „ausgecheckt“, d. h. auf den Rechner eines Programmierers geladen. Nach der Bearbeitung werden die geänderten Dateien wieder in das Repository „eingchecked“, wobei die Veränderung protokolliert wird. Die Versionenverwaltung dient der Dokumentation der Systementwicklung, erlaubt aber auch die jederzeitige Rekonstruktion von früheren Zuständen der Systembeschreibung.
DTD	Siehe Dokumenttypdefinition.
Dokumenttypdefinition	<p>Eine Dokumenttypdefinition (englisch Document Type Definition, DTD, auch Schema-Definition oder DOCTYPE) ist ein Satz an Regeln, der benutzt wird, um Dokumente eines bestimmten Typs zu deklarieren. Ein Dokumenttyp ist dabei eine Klasse ähnlicher Dokumente, wie beispielsweise Telefonbücher oder Inventurdatensätze. Die Dokumenttypdefinition besteht dabei aus Elementtypen, Attributen von Elementen, Entitäten und Notationen. Konkret heißt das, dass in einer DTD die Reihenfolge, die Verschachtelung der Elemente und die Art des Inhalts von Attributen festgelegt wird – kurz gesagt: die Struktur des Dokuments.</p> <p>— Wikipedia [http://de.wikipedia.org/wiki/Dokumenttypdefinition]</p>
Tag	"In Auszeichnungssprachen (engl. mark-up languages) wie SGML, XML oder HTML bezeichnen Tags die in Kleiner- und Größerzeichen eingeschlossenen Kürzel, die beispielsweise in HTML dazu dienen, Textelemente auszuzeichnen oder in XML Daten zu klassifizieren und strukturieren. Dabei kommen sogenannte öffnende und schließende Tags zum Einsatz. Letztere sind durch einen führenden Schrägstrich gekennzeichnet. Tags können als benannte Klammern aufgefasst werden. Analog zur korrekten Klammerung arithmetischer Ausdrücke dürfen diese immer nur paarweise auftreten (in HTML existieren auch Ausnahmen). Auf diese Weise erzeugen sie eine Baumstruktur." Siehe Wikipedia-Artikel 'Tag (Informatik)'.

XML

Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungssprache“), abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten.

Stichwortverzeichnis

G

- GNU, 33
 - GNU-Tools, 33
 - Kernel, 33
 - Mach, 33

I

- Installation, 1
 - Installation Subversion Client, 1

L

- Listen
 - Geordnete Liste, 36
 - Ungeordnete Liste, 34

S

- Subversion
 - Nutzung, 6
- SVN, 4