

WordXML

WordXML ist eine leistungsstarke Erweiterung für Microsoft Word, um Inhalte auf einfache Art und Weise in das offene, medienneutrale und herstellerunabhängige [XML](#)-Format zu konvertieren. Auf Basis einer speziellen Dokumentvorlage können Inhalte ohne technische Fachkenntnisse strukturiert und als [XML](#) gespeichert werden.

Dieses Dokument enthält die die Benutzerdokumentation für WordXML und den [ResultViewer](#), der im WordXML Software-Paket enthalten ist. Im folgenden wird der [ResultViewer](#) auch kurz Viewer genannt. Er wird hauptsächlich für die Ergebniskontrolle und die [Erstellung von Ergebnistransformationen](#) verwendet.

Dieses Dokument wurde selbst mit Word erstellt und mit WordXML zur Archivierung in langlebiges und medienneutrales [XML](#) konvertiert. Aus dieser [XML](#)-Repräsentation wurde mit WordXML-Werkzeugen sowohl eine [Online-Hilfe](#) für das Web als auch eine [Windows-Hilfe](#) generiert (eine von beiden lesen Sie gerade).

Einleitung

Mit WordXML können Word-Dokumente nach [XML](#) konvertiert werden. Die [XML](#)-Dateien werden über [XSLT](#)-Stylesheets transformiert; als Ausgabe stehen die Formate [XML](#), [HTML](#)¹ und [TEXT](#) zur Verfügung².

Zur Betrachtung der Ergebnisse in den Formaten [XML](#) und [HTML](#) dient der [ResultViewer](#).

Diese Funktionalität wird grundsätzlich auch von jedem modernen Browser unterstützt. Der [ResultViewer](#) selbst basiert auf dem Internet Explorer von Microsoft. Er stellt die zentrale Komponente des Viewers dar. Der [ResultViewer](#) ergänzt die Browserfunktionalität um viele nützliche Funktionen, insbesondere vereinfacht er solche Aufgaben, die mit der [Konvertierung von Word-Dokumenten](#) Hand in Hand gehen.

Mit Hilfe von WordXML können Sie wie gewohnt [Dokumente in Microsoft Word erstellen](#) und zu spezifischen [XML](#)-Dokumenten auf Basis einer zugehörigen [DTD](#) konvertieren. Die Inhalte werden über definierte Formatvorlagen strukturiert, und dadurch bei der Konvertierung den entsprechenden Elementen der [XML](#)-Definition zugeordnet. Das Layout der [XML](#)-Dokumente wird bei der späteren Publikation z.B. im Internet über [XSLT](#)- bzw. [CSS](#)-Stylesheets erzeugt, die den einzelnen [XML](#)-Elementen eine bestimmte Formatierung zuweisen. Dadurch wird eine konsequente Trennung des Inhalts und des Layouts erreicht, d.h. Inhalte werden in [XML](#) medienneutral gespeichert und können mit beliebigen Ausgabegeräten in unterschiedlichen Layouts publiziert oder weiterverarbeitet werden. Desweiteren müssen inhaltliche Änderungen nur an einer Stelle – dem Word-Dokument – vorgenommen werden. Dies ermöglicht praktisch jedem die Pflege von Dateninhalten unabhängig von seinen Vorkenntnissen und dem eigentlichen Zielmedium.

Hinweis: Bei der Strukturierung der Inhalte mit MS Word ist nur die Auszeichnung der Textstellen mit den entsprechenden Formatvorlagen ausschlaggebend für [die Konvertierung](#). Layoutinformationen, die das Aussehen der Texte in MS Word beeinflussen (wie z.B. Fettdruck, Schriftarten, Schriftgrößen, etc.), haben keinen Einfluss auf die [Konvertierung nach XML](#) – entscheidend ist nur die Zuweisung von Formatvorlagen bei den entsprechenden Textstellen. Die Ansicht bzw. das Layout der Inhalte in MS Word stellt lediglich eine Strukturierungshilfe dar und entspricht nicht dem Layout bei der Publikation.

¹ Die im Paket mitgelieferten Stylesheets zur Transformation nach HTML, insbesondere diejenigen, die zur Erzeugung der HTML-Hilfe verwendet werden, sind [XHTML](#) konform

² Praktisch steht jedes beliebige, textbasierte Ausgabeformat zur Verfügung. Die Ausgabe ist einzig von dem dafür verwendeten Stylesheet ab. Über ein Stylesheet kann jederzeit ein eigenes Format spezifiziert werden.

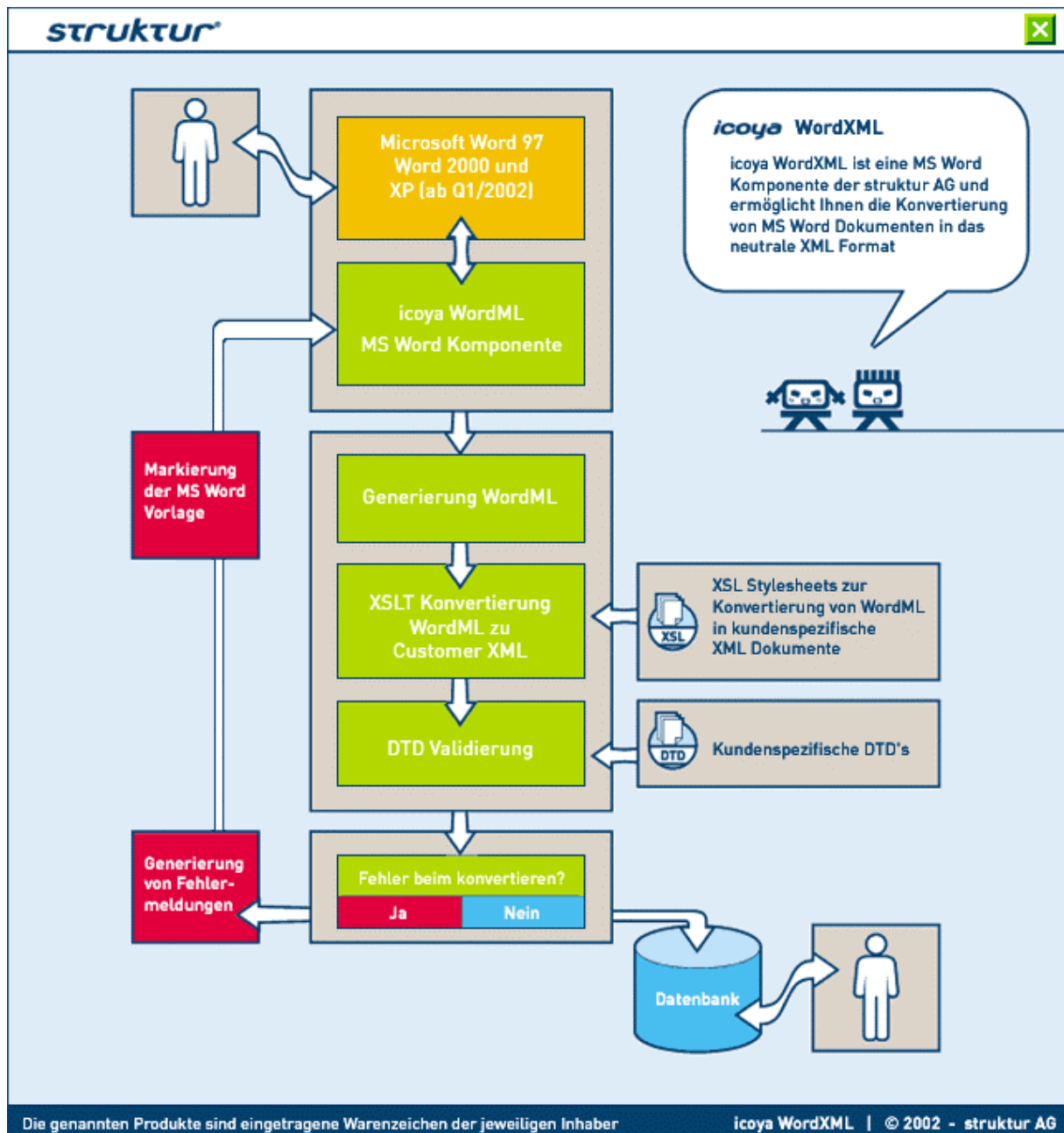


Figure 1 Schematische Darstellung der WordXML Funktionsweise

Warum XML und warum Word?

Aus welchem Grund sollte man Word-Dokumente nach [XML](#) konvertieren?

Dafür gibt es mehrere Antworten.

Zum einen liegt dies in [XML](#) selbst begründet. [XML](#) ist ein wohldefiniertes, offenes, dynamisches, tag-basiertes Datenformat, das weltweit als Standard akzeptiert ist. Es weder hersteller- noch plattformabhängig. Es ist textbasiert und somit auch von Menschen lesbar. Document Type Definitions ([DTDs](#)) bieten die Möglichkeit, die Gliederung von [XML](#)-Dokumenten so zu beschreiben, daß ihre Struktur formal Überprüft werden kann ([Validierung](#)) und sie gleichzeitig als Spezifikation für die automatische Weiterverarbeitung verwendet werden können. Sicher wird man Daten, die in [XML](#) abgespeichert sind, noch in vielen Jahren problemlos verarbeiten können. Somit bietet [XML](#) auch eine hohe Investitions- und Zukunftssicherheit.

Viele dieser Vorteile bieten andere Datenformate nicht, insbesondere wenn sie proprietär sind – wie Microsoft Word Dokumente etwa.

Außerdem bietet [XML](#) eine Reihe von begleitenden Standards, welche mächtige Mechanismen für die Verarbeitung, Auswertung und Visualisierung von [XML](#) bereitstellen.

Den Grund für den Einsatz einer Textverarbeitung bietet – trotz proprietärem Datenformat -- Word von Microsoft. Word ist wohl die am weitesten verbreitete Office-Anwendung weltweit. Praktisch ist Word an den meisten Arbeitsplätzen verfügbar. Es ist einfach zu bedienen und genießt eine hohe Akzeptanz bei den Anwendern. Während man zur Erstellung von [XML](#) technisches Wissen benötigt, ist die Anwendung von Word weitgehend intuitiv. Allerdings läßt sich [XML](#) mit jedem beliebigen Editor erzeugen, was aber mit zunehmender Größe und Komplexität immer schwieriger wird. Spätestens dann kommen spezielle Editoren und Werkzeuge ins Spiel, die in der Regel aber teuer und schwer zu verstehen sind³.

³ Der Nutzen von XML-Werkzeugen soll hier nicht in Frage gestellt werden. Ihr Einsatz geht weit über die Fähigkeiten von WordXML hinaus. Allerdings erfordern sie bei technisch unbeschlagenen Anwendern einen hohen lernaufwand. Es bleibt jedem selbst überlassen, ob der Aufwand lohnt.

Installation

Die Installation von WordXML auf Ihrem Computer ist sehr einfach. In der Regel erfolgt der Vertrieb in Form einer einzigen ausführbaren Datei. Starten Sie das Programm und folgen Sie den Anweisungen auf dem Bildschirm. Die Installation ist vollständig Dialog-gesteuert. Der Installationsvorgang dauert nur wenige Minuten. Neuerungen und Details zur Installation entnehmen Sie bitte der Datei `readme.txt`.

Im nächsten Abschnitt wird die [Installation in einzelnen Schritten](#) erklärt.

Hinweis: Stellen Sie vor der Installation sicher, daß Ihr Computer die Systemanforderungen erfüllt und lesen Sie die beigefügte Lizenzvereinbarung aufmerksam.

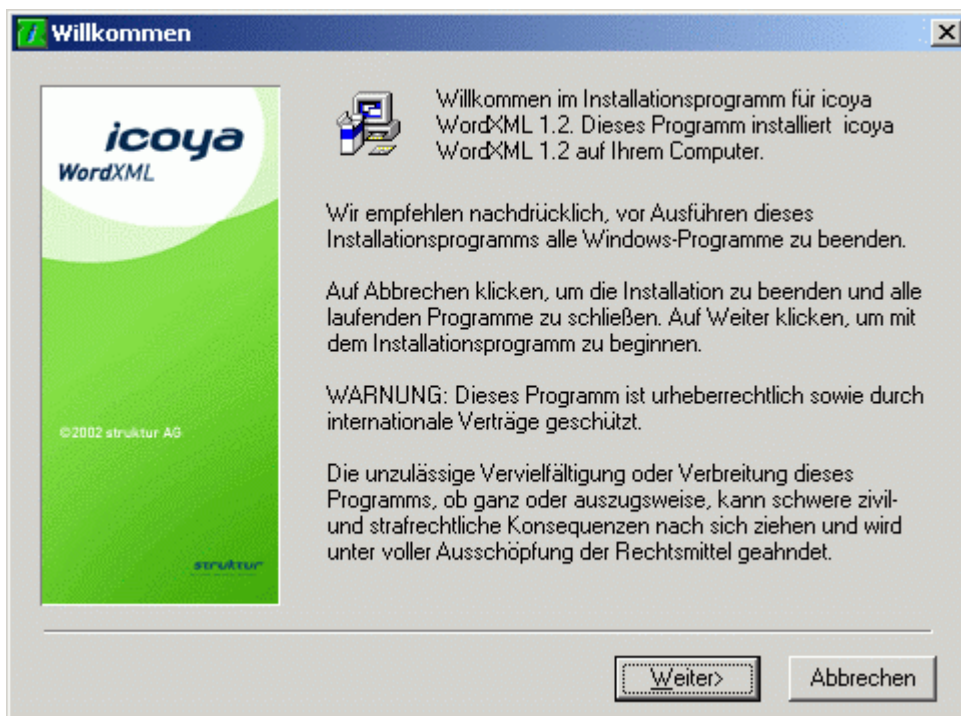


Figure 2 Ein typischer Installationsdialog

Die Installation in einzelnen Schritten

Die Installation verläuft wie folgt:

1. Schliessen Sie alle Anwendungen unter Windows.
2. Wählen Sie im Startmenü die Option [Ausführen].
3. Geben Sie in der Dialogbox „Öffnen“ den vollständigen Pfadnamen der Datei an (zum Beispiel `c:\newsoftware\WordXML\WordXML.exe`)⁴.
4. Klicken Sie auf den [OK] Button und folgen Sie den Anweisungen auf dem Bildschirm.

Hinweis: Die Installation kann auch durch einen Doppelklick auf das Symbol der Installationsdatei im Windows-Explorer oder im Fenster „Arbeitsplatz“ gestartet werden.

Hinweis: Sind während der Installation andere Programme geöffnet, so kann dies möglicherweise zu einer fehlerhaften Installation führen. Sollte eine neu installierte Version von icoya WordXML nicht problemlos funktionieren, führen Sie eine Neuinstallation durch und beenden Sie vorher alle Programme.

⁴ Das Installationsprogramm wird in der Regel nicht einfach nur WordXML.exe heißen. Normalerweise enthält es mindestens noch die Versionsnummer.

Registrierung

WordXML ist ein kommerzielles Produkt und bedarf der Registrierung. Die Registrierung der Anwender erfolgt nach den in der Lizenzvereinbarung genannten Richtlinien und Mechanismen. WordXML steht ebenfalls als Testversion zu Evaluierungszwecken zur Verfügung. Die Testversion ist gegenüber der Vollversion in ihrer Funktionalität eingeschränkt. Bestimmte Funktionen wie das [Generieren einer HTML-Hilfe](#) stehen nicht zur Verfügung und die Dokumentgröße ist in der Anzahl der Paragraphen stark limitiert. Die Evaluierungsphase dauert 30 Tage, danach sind keine Konvertierungen mehr möglich.

Die Registrierung erfolgt entweder während der Installation oder zu einem späteren Zeitpunkt über einen entsprechenden Dialog im [ResultViewer](#). Wird die Registrierung während der Installation nicht durchgeführt, wird nur die Testversion installiert. Durch eine spätere Registrierung wird der volle Funktionsumfang freigeschaltet. Auch Testversionen, die wegen Ablauf des Evaluierungszeitraums nicht mehr funktionsfähig sind, können durch eine nachträgliche Registrierung freigeschaltet werden.

HINWEIS: Wurde auf einem Rechner eine Testversion installiert und die Evaluierungsphase ist abgelaufen, so muß bei der folgenden Installation die Registrierung als Vollversion gemacht werden. Dies ist deshalb notwendig, um zu verhindern, daß die Evaluierungsphase einfach durch eine Neuinstallation der Testversion verlängert werden kann. Eine Installation der Testversion ist dann also nicht mehr möglich!

Während der Installation erscheint folgender Dialog:



WordXML Registration

Please fill in the information below for registration purposes.
The data is used only for validation of your local copy!

Name

Email

Company

Key

Figure 3 Die Registrierung

Den geforderten Schlüssel (*key*) zur Freischaltung bzw. zur Registrierung der Vollversion erhält man durch die Lizenzierung gemäß den entsprechenden Vereinbarungen. Solange der Schlüssel nicht eingegeben wird, bleibt der Button zur Registrierung deaktiviert. Die Installation der Testversion ist trotzdem möglich, allerdings müssen auch hierfür alle Eingabefelder ausgefüllt werden.

Bei Fragen zur Installation und Registrierung wenden Sie sich einfach an support@struktur.de oder besuchen Sie <http://www.struktur.de>.

Deinstallation

In der WordXML Programm-Gruppe ist die Datei `Deinstallieren` verfügbar. Durch ausführen dieser Datei wird WordXML einschliesslich aller Komponenten von Ihrem Rechner entfernt. Die Datei befindet sich in dem als Installationsverzeichnis ausgewählten Verzeichnis. Wurden bei der Installation die optionalen Verknüpfungen gewählt, so findet sich das Deinstallationsprogramm auch im Startmenu in der WordXML Programmgruppe. Außerdem kann das Programm über die Systemsteuerung entfernt werden.

Die Deinstallation in Schritten:

1. Wählen Sie im Startmenü die Option [Ausführen].
2. Geben Sie in der Dialogbox „Öffnen“ den vollständigen Namen der Datei an (zum Beispiel `c:\newsoftware\icoya\uninstall.exe`).
3. Klicken Sie auf den [OK] Button und folgen Sie den Anweisungen auf dem Bildschirm.

Die Deinstallation über die Programmgruppe:

1. Gehen Sie in der WordXML-Programmgruppe auf Deinstallieren.
2. Folgen Sie den Anweisungen auf dem Bildschirm.

Die Deinstalltion über die Systemsteuerung:

1. Wählen Sie im Startmenü die Option [Einstellungen][Systemsteuerung]. Das Systemsteuerungsfenster wird geöffnet.
2. Doppelklicken Sie auf das Symbol „Software“. Es wird ein Fenster geöffnet, das eine Liste der auf dem Computer installierten Software anzeigt. Wählen Sie aus.
3. Bestätigen Sie die Auswahl durch Klicken auf den Button „Ändern/Entfernen“

Die Beschreibung bezieht sich auf Windows2000 und kann bei anderen Windows-Versionen unwesentlich abweichen.

Systemanforderungen

Als Mindestanforderung muß der Internet Explorer in der Version 5 oder höher installiert sein; empfohlen ist die Version 5.5. Die Systemanforderungen für den [Viewer](#) entsprechen den Systemanforderungen für den aktuell verwendeten Internet Explorer. Die Systemvoraussetzungen für den Internet Explorer der Version 5.0 sind unten aufgeführt. Die Systemanforderungen für Ihren Internet Explorer Version finden Sie auf der [Webseite von Microsoft](#).

Da WordXML ein Plug-In für Microsoft Word ist, sind die Systemanforderungen von Microsoft Word bzw. das für die jeweilige Version benötigte Betriebssystem entscheidend. WordXML selbst benötigt nur etwa 25MB zusätzlichen Festplattenplatz.

Windows XP	<ul style="list-style-type: none"> • PC mit 300 MHz Taktrate oder höher • 128 Megabyte (MB) RAM oder mehr • 1.5 Gigabyte (GB) verfügbarer Festplattenspeicher • Super VGA (800 × 600) oder höher auflösende Graphikkarte und Monitor • CD-ROM oder DVD drive • Tastatur und Maus
Windows 2000	<ul style="list-style-type: none"> • PC mit 133 MHz Taktrate oder höher • 64 Megabyte (MB) RAM oder mehr • 800 Megabyte (GB) verfügbarer Festplattenspeicher • Super VGA (800 × 600) oder höher auflösende Graphikkarte und Monitor • CD-ROM oder DVD drive • Tastatur und Maus
Windows NT 4.0	<ul style="list-style-type: none"> • PC mit 133 MHz Taktrate oder höher • 64 Megabyte (MB) RAM oder mehr • 800 Megabyte (GB) verfügbarer Festplattenspeicher • Super VGA (640 × 480) oder höher auflösende Graphikkarte und Monitor • CD-ROM • Tastatur und Maus
Windows 98	<ul style="list-style-type: none"> • PC mit 133 MHz Taktrate oder höher • 32 Megabyte (MB) RAM oder mehr • 200 Megabyte (GB) verfügbarer Festplattenspeicher • Super VGA (640 × 480) oder höher auflösende Graphikkarte und Monitor • CD-ROM • Tastatur und Maus

Table 1 Betriebssystemanforderungen

Software	<ul style="list-style-type: none"> • Microsoft Word 98, Microsoft Word 2000 oder Microsoft Word 2002 • Microsoft Internet Explorer Version 5.0 oder höher
Optionale Software	Microsoft HTML Help Workshop ⁵

Table 2 Softwareanforderungen

Der Internet Explorer von Microsoft ist praktisch mit allen von WordXML unterstützten Windows-Versionen verfügbar.

⁵ Für die Erstellung eines HTML Hilfe Projektes ist der HTML Help Workshop unerlässlich. Ohne ihn läßt sich jedoch trotzdem ein HTML-Projekt erstellen. Der Workshop steht zum freien Download zur Verfügung. Hingegen ist die Hilfe-Generierung ein Zusatzmodul und nicht in der Standarddistribution enthalten.

Konventionen

Zum besseren Verständnis der Dokumentation werden hier die verwendeten Notationen und Kennzeichnungen besprochen, die für die Beschreibung der Oberfläche und von Arbeitsabläufen in diesem Dokument verwendet werden.

- Die **Namen von Menüs** werden in Anführungszeichen gesetzt.
Beispiel: ‚Datei‘-Menü
- **Graphische Schaltflächen** werden kurz als ‚Buttons‘ bezeichnet. Die Beschriftung von Buttons wird durch eckige Klammern gekennzeichnet.
Beispiele: [OK] oder [Abbruch]
- **Menüeinträge** werden ebenfalls durch eckige Klammern gekennzeichnet.
Beispiel: ... die Option [Speichern unter] im ‚Datei‘-Menü
- **Beschriftungen auf der Programmoberfläche** werden in Anführungszeichen gesetzt.
Beispiel: ... die Felder ‚Name‘ und ‚Firma‘ ...

Einige Notationen werden doppelt verwendet. Grundsätzlich sollte jedoch aus dem jeweiligen Kontext klar werden, um welchen Fall es sich handelt. In manchen Fällen sind die Begriffe auch austauschbar. Ist zum Beispiel nur von [Speichern unter] die Rede, so ist in diesem Falle gleich, ob die Funktion über das ‚Datei‘-Menü oder die entsprechende Schaltfläche auf der Werkzeugleiste aufgerufen wird.

Allgemeine Bedienungshinweise

Bei der Erstellung eines Dokuments in MS Word sind einige Punkte zu beachten, um ein gutes Konvertierungsergebnis zu erzielen. In den untenstehenden Absätzen wird ausführlich auf die jeweiligen Vorgaben und Bedingungen eingegangen. Grundsätzlich gilt: für eine Konvertierung ist es unbedingt notwendig, das zu konvertierende Dokument mit der Dokumentvorlage `xmlword.dot` zu erstellen. Beachten Sie desweiteren folgenden Hinweis zur Layoutanzeige in Word:

Das in Word angezeigte Layout der Inhalte dient der Übersichtlichkeit und erleichtert die Strukturierung. Das Layout in Word stellt jedoch keine Voransicht der späteren Publikation z.B. im Internet dar - die Layoutinformation für das Ausgabemedium wird in Stylesheets gespeichert und bei einer Publikation zu den [XML](#)-Dokumenten hinzugefügt. Eine Vorschau ist nur über den [icoya WordXML Viewer](#) möglich, der nach der Konvertierung über das Windows-Startmenü aufgerufen werden kann.

Das in Word angezeigte Layout kann über die [Formatvorlage](#) angepasst werden, allerdings dürfen dabei die Namen der Formatvorlagen nicht modifiziert werden, da sonst die Zuweisung bei der Konvertierung nicht mehr funktioniert ([Im ersten Konvertierungsschritt](#) erhalten die Tags die Namen der Styles. Die weiterverarbeitenden [XSLT](#)-Stylesheets, etwa zum [Filtern des Ergebnisses](#) oder zur [Generierung einer Windows-Hilfe](#) sind an die Tag-Namen der vorhandenen Styles angepaßt)

Grundfunktionalität

Ziel des Einsatzes von WordXML ist die einfache [Erstellung von Inhalten](#) mit einem bekannten und weit verbreiteten Programm wie Word, die Speicherung der Dokumente in dem unabhängigen und einfach weiterzuverarbeitenden Format [XML](#), und schließlich die Generierung von [HTML](#)-Seiten zur Veröffentlichung oder anderer, eventuell [selbstdefinierter Formate](#).

Der ResultViewer

Die Aufgaben des Viewers hierbei sind die Kontrolle des Ergebnisse der [Konvertierung von Word-Dokumenten](#) und hauptsächlich die Anzeige der generierten [HTML](#)-Dokumente in ihrer abschließenden Form. Der Viewer stellt die Ergebnisdateien so dar, wie sie später in einem Browser aussehen. Tatsächlich ist der Anzeigebereich des Viewer ein eingebetteter Internet Explorer. Zusätzlich können über das Viewer-Programm Werkzeuge für andere Aufgaben aufgerufen werden, etwa die [Stapelverarbeitung beliebig vieler Dokumente](#) oder die [Erzeugung eines HTML-Projekts \(einer Webseite!\) oder einer HTML Hilfe](#) für Microsoft Windows.

Zusammenfassend sind die Grundfähigkeiten des Viewers:

- Das [Öffnen jeweils einer XML-Datei](#) und die Anzeige des Quelltexts inklusive einer automatischen [Validierung](#).
- Das Öffnen mehrerer beliebiger Stylesheets zur Übersetzung der aktuell geöffneten XML-Datei nach [HTML](#), [XML](#) oder [TEXT](#)
- Die Browser-Darstellung der [HTML](#)-Dateien für jeden geöffneten Style.
- Die Anzeige des jeweils zu einer [HTML](#)-Darstellung gehörigen [XSLT](#)-Stylesheets.
- Die [Generierung von HTML Projekten](#) aus einem Word-Dokument.
- Die [Generierung von HTML Help Projekten](#), der gängigen Form von Windows Hilfedateien seit Windows 2000⁶.
- Die Bereitstellung von Werkzeugen zur [Einzelkonvertierung](#) oder [Stapelkonvertierung](#) und zur [Validierung](#).
- Das Starten eines wählbaren [externen Editors](#) zur Bearbeitung der angezeigten Dateien.

Die Kernfunktionalität des [Viewers](#) beschränkt sich also auf die Anzeige von [XML](#)-Dokumenten, insbesondere diejenigen, die [Konvertierungsergebnisse von WordXML](#) darstellen, und den daraus gewonnenen [HTML](#)-Dateien. An Stelle der [HTML](#)-Dateien kann man sich auch die zur Transformation nach [HTML](#) verwendeten [XSLT](#)-Stylesheets anzeigen lassen. Im durchschnittlichen Anwendungsfall wird man davon allerdings eher selten Gebrauch machen. Zusätzlich dient der [Viewer](#) noch als bequeme Oberfläche für [XSL-Transformationen](#) und die [Generierung von HTML und Windows-Hilfe Projekten](#).

Die Dateien können im Viewer betrachtet, nicht jedoch editiert werden. Die Grundfunktionalität entspricht demnach der eines Webbrowsers zuzüglich einer [XSLT](#)-Engine für die Transformation von [XML](#) nach [HTML](#)⁷. Es sei hier schon einmal kurz darauf hingewiesen, daß bei der [Erstellung von HTML-Seiten mit WordXML](#) alle Gestaltungsmöglichkeiten inklusive [cascading stylesheets \(CSS\)](#) zur Verfügung stehen. Die Funktionalität, die vom [ResultViewer](#) ergänzt wird, betrifft hauptsächlich die Verwaltung von Dateien und Projekten sowie die Anpassung der Programmumgebung an jeden einzelnen Benutzer.

Um schnelle Änderungen an der angezeigten Datei vorzunehmen, besteht die Möglichkeit, den jeweiligen Quelltext in einem [externen Editor](#) zu öffnen.

⁶ Prinzipiell ist eine HTML Hilfe genau wie eine Webseite aufgebaut, d.h. sie besteht aus einer Menge von HTML-Dokumenten, die untereinander verlinkt sind. Besonders an der Hilfe ist jedoch, daß alle Dateien in eine Datei ‚kompiliert‘ werden. Zusätzlich werden Dateien erstellt, die beispielsweise einen Suchindex oder ein Inhaltsverzeichnis enthalten.

⁷ Für die Kontrolle von Ergebnisdaten in den Formaten TEXT oder XML empfehlen sich die Zielapplikationen, also die Programme, welche die Daten weiterverarbeiten. Auch HTML sollte in so vielen Webbrowsern wie möglich getestet werden, falls es im Internet publiziert werden soll.

Externe Editoren

Der [ResultViewer](#) erlaubt die [Angabe eines Editors als Standard](#). Grundsätzlich kann jeder beliebige Editor angegeben werden, der folgende Bedingung erfüllt. Der Editor muß als Übergabeparameter in der Kommandozeile einen Dateinamen akzeptieren, und diese Datei dann auch öffnen⁸.

Wird kein Editorprogramm eingetragen, so wird das System nach anderen Editoren durchsucht. Im Zweifel wird die Datei mit dem Windows Notepad angezeigt, da dies auf jeden Fall verfügbar ist.

Über den [Eintrag \[Quelltext editieren\] im ‚Datei‘-Menu](#) oder die entsprechende Schaltfläche wird aus dem [Viewer](#) heraus ein Editor gestartet, in dem die aktuelle Datei geöffnet ist. Grundsätzlich sind dabei folgenden Punkte zu beachten:

- Es wird versucht, den Editor UltraEdit zu starten. Dieser muß in der Pfadvariablen der Systemumgebung eingetragen sein. Ist dies nicht der Fall oder ist UltraEdit nicht installiert, so wird das Windows Notepad gestartet.
- Nicht jede Version des UltraEdit unterstützt UTF-8, also die [Codierung](#), die standardgemäß für [XML](#) verwendet wird. Umlaute und Sonderzeichen werden dann nicht korrekt angezeigt. Dies gilt ebenfalls für das Windows Notepad. In Windows 2000 unterstützt das Notepad UTF-8.

Hinweis: Grundsätzlich sollten alle Änderungen im Word-Dokument vorgenommen werden, da das Word-Dokument der Ursprung der Daten und somit die Basis aller Konvertierungen ist. Werden Fehler beispielsweise im HTML-Code verbessert, so sind sie in der XML-Datei trotzdem noch enthalten. Wird dann auf Basis der XML-Datei ein neues HTML-Dokument erzeugt, so tritt der Fehler wieder auf.

⁸ Grundsätzlich kann anstelle eines Editors jedes beliebige Programm angegeben werden, das die Bedingung erfüllt. Dies kann sehr praktisch sein, falls getestet werden soll, ob die erzeugten Daten sich mit bestimmten Programmen automatisch weiterverarbeiten lassen.

Der ResultViewer

Dieser Abschnitt beschreibt die einzelnen Komponenten der Benutzeroberfläche und beschreibt ihre Bedeutung.

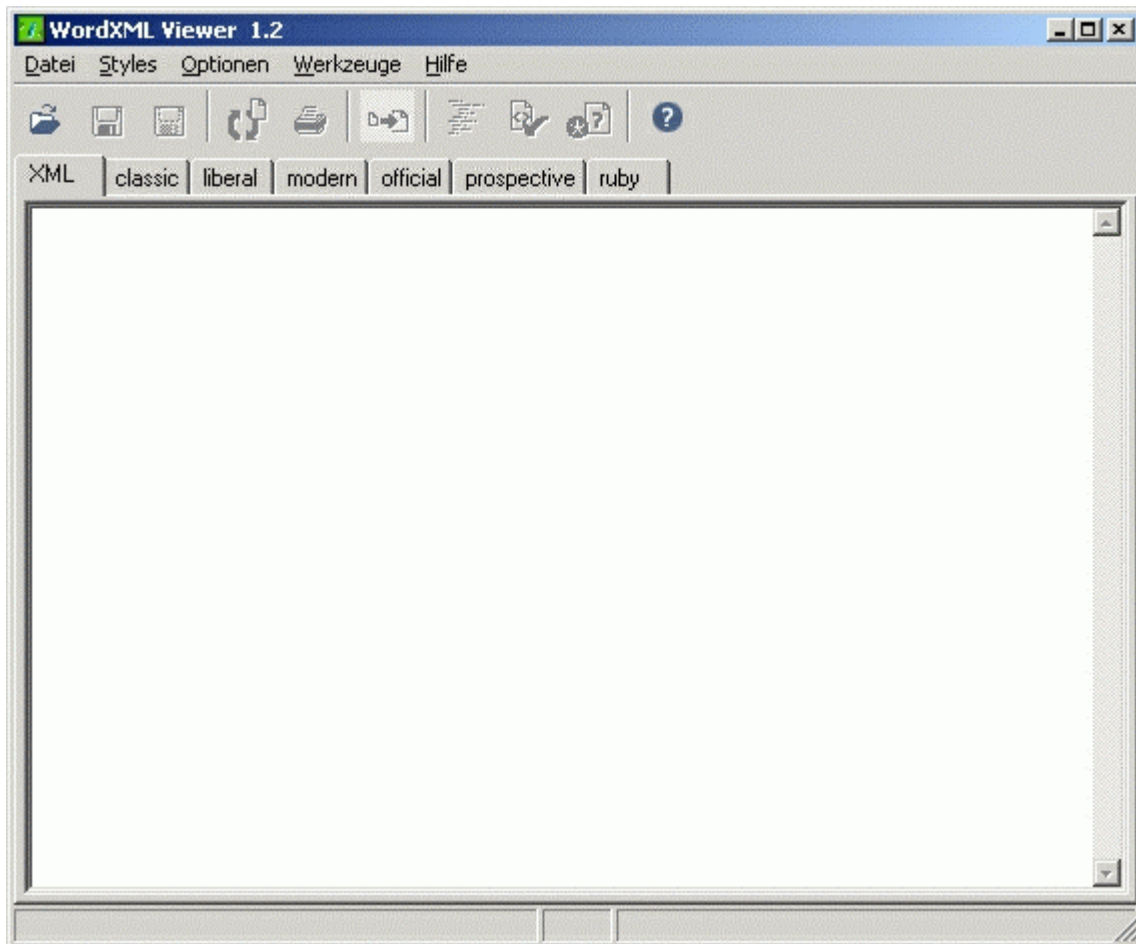


Figure 4 Die Benutzeroberfläche des ResultViewers

Bei der Installation von WordXML wird der ResultViewer automatisch mit installiert. Der Aufruf des Programms erfolgt über die Option ‚ResultViewer‘ in der WordXML Programmgruppe. Der Anzeigebereich ist eine Einbettung des zum Internet Explorer gehörigen Webbrowsers. Der Internet Explorer muß - unabhängig von WordXML - auf dem System vorhanden sein, auf dem der ResultViewer benutzt werden soll. Die Konvertierung der Dokumente ist allerdings auch ohne Internet Explorer möglich.

Hinweis: Ohne eine Installation des Internet Explorers der Version 5 oder höher ist der ResultViewer nicht lauffähig.

Beim Start des ResultViewers wird dynamisch eine Instanz der Webbrowser-Klasse des aktuell installierten Internet Explorers erzeugt. Der Viewer bleibt somit von Versionswechseln des Explorers auf dem System unberührt.

Die Titelleiste

Die Titelleiste enthält die Namen des Programms sowie die aktuelle Versionsnummer. Außerdem wird der absolute Pfad der aktuell geöffneten [XML](#)-Datei angezeigt (falls eine Datei geöffnet ist).

Das Menü ‚Datei‘

Das ‚Datei‘-Menü enthält die Grundfunktionen für den Umgang mit [XML](#)-Dateien. Das Menü enthält die gängigen Optionen:

- **Öffnen:** Öffnet eine neue [XML](#) Datei. Es erscheint wie gewohnt der ‚Datei öffnen‘-Dialog von Windows.
- **Speichern:** Die aktuell angezeigte Datei wird gespeichert ([XML](#), [XSLT](#) und [HTML](#)).
- **Speichern unter:** Die aktuell angezeigte Datei wird in einem beliebigen Pfad gespeichert, der über eine Dialog ausgewählt wird ([XML](#), [XSLT](#) und [HTML](#)).
- **Aktualisieren:** Die Anzeige der aktuell angezeigten Datei wird aktualisiert, d. h. die Datei wird neu geladen ([XML](#), [XSLT](#) und [HTML](#)).
- **Quellcode editieren:** Der Quellcode der aktuellen Datei ([HTML](#)) bzw. die aktuelle Datei selbst ([XML](#), [XSLT](#)) wird in einem [externen Editor](#) geöffnet.
- **Seite einrichten, Drucken und Druckvorschau:** Entsprechen der aus anderen Programmen bekannten Funktionalität. Drucken bezieht sich auf die aktuell angezeigte Datei ([XML](#), [XSLT](#) und [HTML](#)).
- **Dateien:** zeigt eine Liste von bis zu zehn der zuletzt geöffneten Dateien an.
- **Beenden:** Beendet das Programm

Zusätzlich wird ein Verlauf der zuletzt geöffneten Dateien angezeigt. Die Länge der Liste ist auf maximal zehn begrenzt.

Das Menü ‚Styles‘

Über das ‚Styles‘-Menu werden die im Fenster angezeigten Styles verwaltet.

Als Optionen stehen zur Verfügung:

- **Hinzufügen:** Über einen ‚Datei öffnen‘-Dialog wird ein neuer Style (in Form einer [XSLT](#)-Datei) zur aktuellen Anzeige hinzugefügt.
- **Schließen :** Schließt die Ansicht des angezeigten Styles. Wird aktuell die [XML](#)-Datei angezeigt, ist diese Option deaktiviert.
- **Ansicht umschalten:** Schaltet zwischen der [HTML](#)-Darstellung und der Anzeige des zur Erzeugung des HTML-Quellcodes verwendeten [XSLT](#)-Stylesheets um. Wird aktuell die XML-Datei angezeigt, ist diese Option deaktiviert.
- **Eigenschaften:** Öffnet einen Dialog, über den die wesentlichen Dateieigenschaften des aktuellen Stylesheets angezeigt werden. Wird gerade die [XML](#)-Datei angezeigt, ist diese Option deaktiviert.

Das Menü ‚Optionen‘

Unter dem Menü ‚Optionen‘ finden sich allgemeine Einstellungsmöglichkeiten für den ResultViewer.

Folgende Optionen können gewählt werden:

- **Einstellungen:** Es wird ein Dialog angezeigt, über den Standard-Verzeichnisse für die unterschiedlichen verwendeten Dateitypen gewählt werden können. Dies verkürzt die Navigation beim Öffnen einer Datei. Zusätzlich können noch Default-Styles angegeben werden, die sofort beim Programmstart mit geöffnet werden und eine [DTD](#), die standardmäßig zur [Validierung](#) von angezeigten [XML](#)-Dokumenten verwendet wird.
- **Quellcode-Editor:** Ein Dialog wird geöffnet, über den ein beliebiger Editor als [externer Editor](#) ausgewählt werden kann. Der Editor muß die Übergabe eines Dateinamens als Kommandozeilenparameter unterstützen.
- **Sprache auswählen:** Es erscheint ein Dialog zur Auswahl der Sprache, in der die Programmoberfläche – also die Menüs und Dialoge – angezeigt werden.
- **Registrierung:** Der [Registrierungsdialog](#) für WordXML wird aufgerufen. Nach Angabe der Benutzerdaten und des benutzerspezifischen Software-Schlüssels steht die aktuelle WordXML-Installation uneingeschränkt zur Verfügung.

Das Menü ‚Werkzeuge‘

Im ‚Werkzeuge‘-Menü finden sich Dienstprogramme für die Arbeit mit [XML](#) bzw. WordXML. Verfügbar sind im Moment:

- **Validieren:** Das aktuell angezeigte [XML](#)-Dokument wird gegen die in [Einstellungen](#) als Default eingestellte [DTD](#) validiert. Ist keine Default-[DTD](#) angegeben, wird gegen die im Dokument referenzierte [DTD](#) validiert. Ist dort ebenfalls keine [DTD](#) angegeben, wird das Dokument als gültig eingestuft⁹. Mehr zu Validierung finden Sie [hier](#).
- **Batch-Konvertierung:** Alle [XML](#)-Dateien in einem Verzeichnis werden über ein [XSLT](#)-Stylesheet nach [HTML](#) konvertiert. Das Verzeichnis und das Stylesheet können über einen Dialog eingestellt werden.
- **XSL Transformation:** Dieser Dialog bietet eine bequeme Oberfläche für die Konfiguration einer einzelnen [XSLT](#) Transformation, d. h. die Konvertierung einer XML-Datei durch ein Stylesheet in eines der Formate [XML](#), [HTML](#) oder [TEXT](#).
- **Hilfe generieren:** Über diese Option kann aus einer XML-Datei ein Windows HTML Help™ Projekt erzeugt werden. Diese Funktion erfordert das Vorhandensein der [HTML Help Workshop Software](#) von Microsoft. Diese Software ist frei verfügbar. Nach dem Aufruf dieser Option erscheint ein Dialog für die Einstellungen der Hilfe-Generierung.

⁹ Man kann sich darüber streiten, ob dies richtig ist. Aber wenn keine Regeln zur Überprüfung der Gültigkeit angegeben werden, so kann auch nichts ungültig sein!

Das Menü ‚Hilfe‘

Über das Hilfemenü wird die aktuell verfügbare Dokumentation aufgerufen.

Verfügbare Information sind:

- **Hilfe:** Zeigt diese Hilfedatei an
- **Info:** Zeigt Informationen über das Produkt und den registrierten Benutzer an.

Die ‚Werkzeugleiste‘

Die Werkzeugleiste enthält Schaltelemente (‚Buttons‘), mit deren Hilfe ein schneller Zugriff auf die am häufigsten verwendeten Funktionen des ResultViewers ermöglicht wird. Alle Funktionen können auch über die entsprechenden Menüs aufgerufen werden. Eine genaue Beschreibung der Funktionen ist der Dokumentation der entsprechenden Menüeinträge zu entnehmen.

Für folgende Funktionen sind Schaltelemente vorhanden:

- Aus dem [‚Datei‘-Menü](#): [Öffnen], [Speichern], [Speichern unter], [Aktualisieren], [Drucken], [Quellcode editieren]
- Aus dem [‚Styles‘-Menü](#): [Ansicht umschalten]
- Aus dem [‚Werkzeuge‘-Menü](#): [Validieren], [Hilfe generieren]
- Aus dem [‚Hilfe‘-Menü](#) ; [Hilfe]

Verweilt der Mauszeiger über einem Schaltelement, wird die Bedeutung der Schaltfläche sowohl unterhalb des Mauszeiger als auch im linken Feld der [Statusleiste](#) in Textform angezeigt.

Der Anzeigebereich

Die zentrale Komponente des [ResultViewers](#) ist der Anzeigebereich. Der Anzeigebereich ist ein großes Browser-Feld, in welchem jeweils eine der aktuell geöffneten Dateien dargestellt wird. Für jede geöffnete Datei existiert ein Karteikartenreiter oberhalb des Anzeigefeldes. Auf alle geöffneten Dateien kann über die Karteikartenreiter zugegriffen werden.

Für die Anordnung und Benennung der Reiter gelten folgende Konventionen:

- Der Reiter ganz links zeigt den Quelltext der aktuell geöffneten [XML](#)-Datei an. Ist keine Datei geöffnet, ist die Anzeige leer. Der Reiter ist immer mit ‚XML‘ beschriftet (der volle Dateiname steht in der [Titelleiste](#)). Diese Ansicht kann nicht geschlossen werden. Nach dem Programmstart ist nur sie geöffnet, falls keine Standard-Styles angegeben worden sind.
- Jeder weitere Reiter steht für ein geöffnetes Stylesheet. Durch anklicken des Reiters wird die Anzeige des jeweiligen Styles aktiviert. Im Anzeigebereich wird entweder die [HTML](#)-Datei oder der Quelltext des zugehörigen Stylesheets angezeigt. Zwischen diesen beiden Anzeigemodi kann über [Anzeige umschalten] im [.Styles'-Menü](#) hin- und hergewechselt werden.
- Beim Hinzufügen eines neuen Styles wird dem Anzeigebereich ein neuer Reiter hinzugefügt. Übersteigt die Anzahl der Reiter den maximal darstellbaren Bereich (die Fensterbreite des ResultViewers bzw. die Bildschirmbreite), so erscheint am Fensterrand rechts neben den Reitern eine Bildlaufleiste, mit dessen Hilfe in die nicht sichtbaren Bereiche navigiert werden kann. Der Reiter trägt den Namen der [XSLT](#)-Datei ohne die Dateiendung. Der vollständige Pfad der Datei mit Endung wird im linken äußeren Bereich der [Statusleiste](#) angezeigt.
- Wird ein Style geschlossen, so verschwindet der zugehörige Reiter. Um auf die Darstellung des Styles zuzugreifen, muß er erneut geöffnet werden.

Das Feld des Anzeigebereichs ist ein eingebetteter Internet Explorer. Was die Anzeige von [HTML](#) angeht, so wird der volle Funktionsumfang des Explorers unterstützt. Insbesondere stehen zur Erzeugung optisch ansprechender Ergebnisse neben allen Möglichkeiten, die [HTML](#) bietet, zusätzliche Hilfsmittel wie cascading stylesheets ([CSS](#)) zur Verfügung. Die vom Viewer gezeigten Ergebnisse haben genau die Form, in der die Dokumenten später beispielsweise im Internet oder Intranet veröffentlicht werden.

Die Statusleiste

Die Statusleiste befindet sich am unteren Fensterrand. Dieser ist in drei Bereiche untergliedert. Im linken Bereich wird eine kurze, kontextsensitive Hilfe zu gegenwärtig fokussierten Schaltflächen oder Menüeinträgen angezeigt. In der Mitte wird der Typ des aktuell betrachteten Styles angezeigt ([HTML](#) oder [XSL](#)). Im rechten Feld steht der absolute Pfad der verwendeten Style-Datei.

Mit WordXML arbeiten

In diesem Abschnitt werden die häufigsten Arbeitsschritte und die generelle Arbeitsweise von WordXML erklärt. Einige Funktionen werden nach dem Aufruf ohne weitere Benutzerinteraktion ausgeführt (z. B. [Datei speichern](#)). Diese Funktionen sind selbsterklärend und werden hier nicht weiter besprochen. Andere Funktionen verlangen vom Anwender die Angabe von einem oder mehreren Parametern. Mit diesen wollen wir uns hier beschäftigen.

Ein Dokument mit Word erstellen

Die Erstellung von Dokumenten mit Word sollte eigentlich klar sein. Falls es diesbezüglich offene Fragen gibt, sollte man die entsprechende Dokumentation von Microsoft Word zu Rate ziehen. Hier soll nur beschrieben werden, welche Besonderheiten in Verbindung mit WordXML auftreten.

Grundsätzlich gilt für XML, daß es keine Informationen über die Darstellung der Daten enthält (Layout), sondern die Daten nur strukturiert. Die Strukturierung von Dokumenten ist die Unterteilung etwa in Kapitel und Abschnitte, die Position und Ebene der Überschriften, das Auftreten von Tabellen und Bildern und so weiter. Das Merkmal einer Dokumentstruktur ist, daß es sehr unstrukturiert sein kann. Das heißt, das sich bestimmte Strukturierungselemente zum Beispiel selten oder gar nicht wiederholen, oder daß Abschnitte in Ihrer Größe extrem voneinander abweichen. Der extreme Gegensatz ist etwa die Strukturierung von Datensätzen aus einer Datenbank. Sie gleichen sich im Format und wiederholen sich unzählige male. Man spricht auch von dokumentzentriert und datenzentriert.

Word wird also als eine Art WYSIWYG Struktur-Editor verwendet. Da zum einen bestimmte Texteffekte mit einer Strukturierung assoziiert werden und auf der anderen Seite das Hauptziel der Konvertierung dokumentenzentrierter Anwendung die Darstellung und nicht die Verarbeitung ist, wird trotzdem bei der Konvertierung ein bisschen Layout unterstützt. Allerdings nur aus der speziell dafür erstellten Dokumentvorlage `xmlword.dot`.

Im Klartext heißt dies, daß zum Beispiel Fettschrift in Word nicht berücksichtigt wird, jedoch eine Formatvorlage `icoya_Fett`, die in Word zufällig als fettgedruckter Text dargestellt wird. Man mache sich klar, daß hier ein Unterschied besteht: Fett bedeutet in Word nur die optische Darstellung in Word, wohingegen die Verwendung der Formatvorlage diese Textstelle auszeichnet, und für eine spezielle elektronische Weiterverarbeitung vorbereitet. Dies gilt auch für schrägstellen (kursiv), unterstreichen und einfärben von Textstellen.

Die Verwendung der Dokumentvorlage erreicht man durch die Auswahl im entsprechenden Dialog, wenn man in Word ein neues Dokument öffnet.

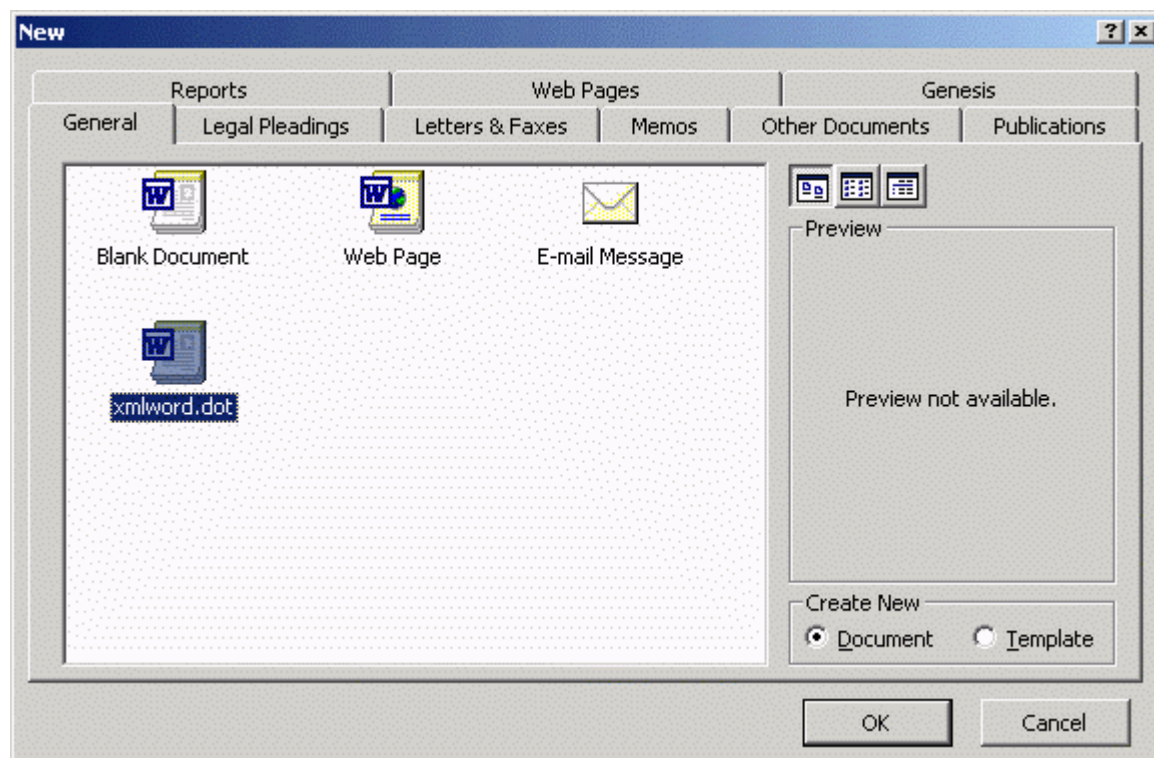


Figure 5 Die Auswahl der WordXML Dokumentvorlage

Nach der Auswahl der Dokumentvorlage kann ein neues Dokument erstellt werden. Mit dem leeren Dokument erscheint ein sogenannter Commandbar, über den auf die wichtigsten Formate und Funktionen von WordXML zugegriffen werden kann.

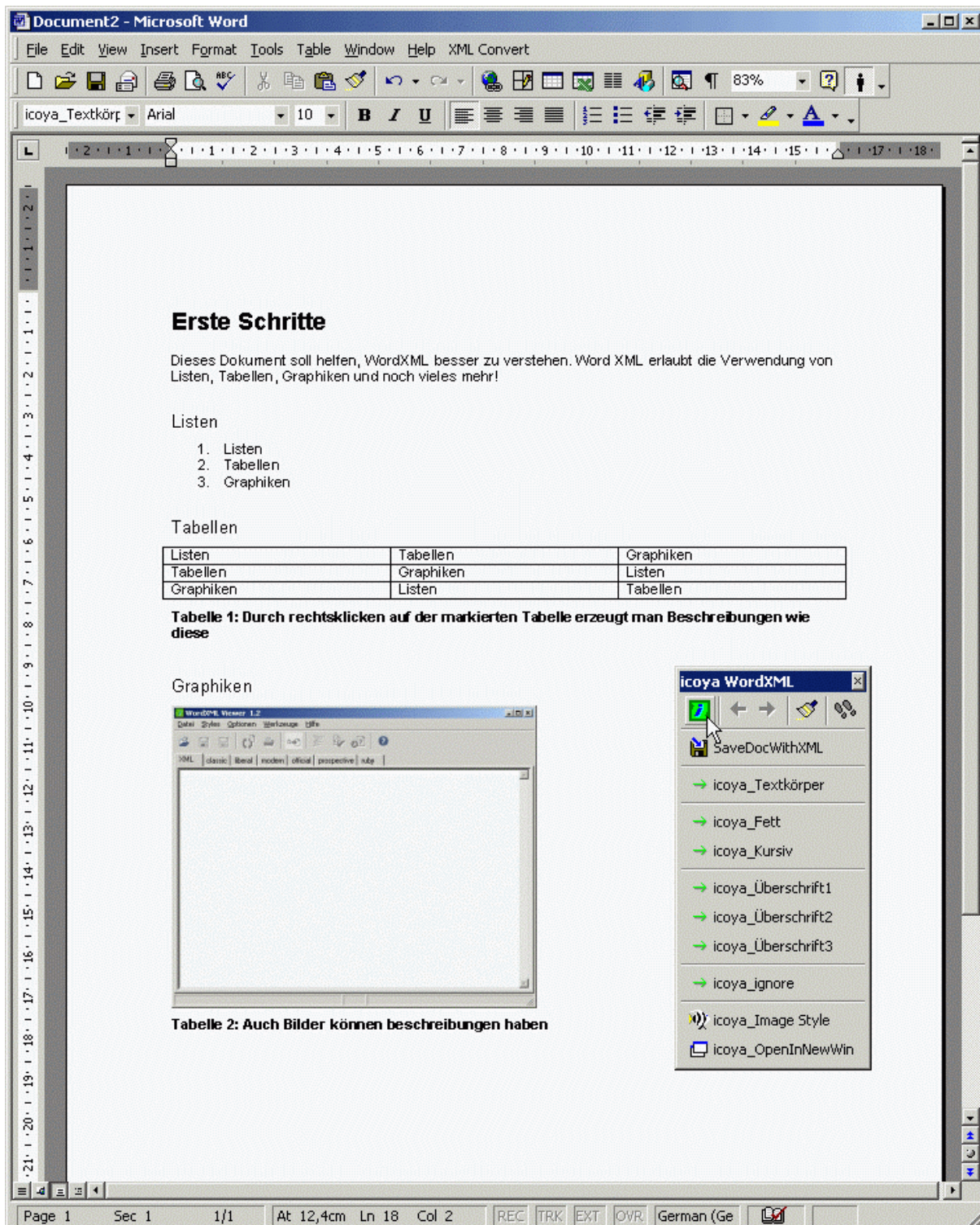


Figure 6 Ein Word-Dokument erstellen

Die Abbildung zeigt ein einfaches Dokument mit einer Graphik, einer Liste und einer Tabelle. Der Commandbar ist rechts neben der Abbildung. Über den grün-blauen Button, auf dem der Mauszeiger ruht, wird die Konvertierung aufgerufen.

Hinweis: Der WordXML Commandbar ist nur dann verfügbar, wenn die WordXML Dokumentvorlage (xmlword.dot) geladen ist. Ist er trotzdem nicht sichtbar, kann man sich über einen Rechtsklick im Toolbar-Bereich von Word ein Liste aller verfügbaren Commandbars anzeigen lassen und auch auswählen. Der Name des WordXML Commandbars ist „icoya WordXML“. Die aktuell angezeigten Commanbars sind markiert.

Für mehr Informationen über Commandbars lesen Sie die entsprechenden Word-Hilfen.

Metadaten erfassen

In Word besteht die Möglichkeit, Metadaten (Eigenschaften bzw. Properties) für ein Dokument anzulegen. Metadaten sind etwa der Name des Autors, das Erstellungsdatum oder eine Stichwortliste. Diese Daten sind garantiert in der [WXML](#)-Datei enthalten und können für bei der [Generierung von Ausgabedateien](#) verwendet werden. Insbesondere bei der [Generierung von HTML-Projekten und Windows-Hilfe-Projekten](#) spielen die Metadaten eine wichtigen Rolle.

The image shows a Windows-style dialog box titled "WordXML.doc Properties". It has several tabs: "General", "Summary", "Statistics", "Contents", and "Custom". The "General" tab is active. The fields are as follows:

- Title: WordXML
- Subject: icoya WordXML
- Author: Info; info@struktur.de
- Manager: (empty)
- Company: struktur AG; http://www.struktur.de
- Category: icoya WordXML
- Keywords: icoya, CMS, WordXML, Word, Microsoft, HTML Help,
- Comments: Dieses Dokument ist die Anleitung für WordXML. WordXML ist Software zur Konvertierung von Microsoft Word Dokumenten nach XML und HTML. Die Ausgabe von WordXML kann jeder belieben
- Hyperlink base: (empty)
- Template: xmlword.dot

At the bottom, there is a checkbox labeled "Save preview picture" which is unchecked, and two buttons: "OK" and "Cancel".

Figure 7 Metadaten und Eigenschaften eines Word-Dokuments

Wie man an der Abbildung erkennen kann, werden in den Feldern „Author“ und „Company“ durch Semikolon (;) getrennt [URLs](#) mit angegeben. In diesem Fall wird bei der [Generierung von HTML-Projekten oder der HTML-Hilfe](#) Links in den Fußzeilen der generierten Dokumente eingefügt, die den Text links vom Semikolon zeigen und auf die rechts vom Semikolon angegebene [URL](#) verlinken

Text erstellen

Die Erstellung von Text erfolgt in der gewohnten Weise. Praktisch sind nur einige wenige Einschränkungen zu beachten. Grundsätzlich kann jedoch alles verwendet werden, was Word zur Erstellung von Dokumenten anbietet, also Fußnoten, Bookmarks, Hyperlinks, etc..

Die wichtigste Einschränkung ist, daß Textformatierungen nicht berücksichtigt werden. So werden bei der Konvertierung die Formate **Fett**, *Kursiv* und Unterstrichen sowie farbige Texte einfach als normale Texte betrachtet. Da [XML](#) als Konvertierungsergebnis nur Daten strukturiert, die Daten und das Layout aber getrennt werden sollen, ist dies auch legitim. Eventuell besitzen die Zielgeräte oder – formate gar nicht die Möglichkeit, diese Attribute darzustellen. Um ein wenig Abhilfe zu schaffen, wurden für **Fett**, *Kursiv* und Unterstrichen entsprechende Textformat-Vorlagen – sogenannte Styles – in Word definiert.

Formatvorlagen werden grundsätzlich bei der Konvertierung berücksichtigt. Text, auf den eine bestimmte Formatvorlage angewendet wird, bekommt im [WXML](#) ein Attribute „style“, dessen Wert der Name der Formatvorlage ist.

Wird mit den mitgelieferten Stylesheets gearbeitet, wird der Text sogar innerhalb von Tags gesetzt, die als Namen den Namen des Styles haben, also zum Beispiel `<icoya_Fett>fetter Text</icoya_Fett>`. Will man ähnlich vorgehen, sollte man sich vergewissern, daß im Falle einer [Validierung](#) die Tags in der [DTD](#) definiert sein müssen.

Mehrere Ausgabeseiten und Seitenumbruch

Um ein Dokument auf mehrere Ausgaben zu Verteilen, muß ein [XSLT-Stylesheet](#) verwendet werden, welches diese Funktionalität unterstützt ¹⁰. Die Trennung kann entweder durch inhaltliche Merkmale veranlaßt werden, oder man setzt explizit Seitenumbrüche. Die von Word implizierte Seitenaufteilung spielt überhaupt keine Rolle. In beiden Fällen muß das verwendete Stylesheet die Trennung erkennen und entsprechend implementieren.

Um einen Seitenumbruch von Hand zu setzen Wählt man über das Word-Menü [Format][Paragraph] die Option „Page Break Before“ aus. An der Stelle, an welcher der Cursor steht, wird ein Seitenumbruch vorgenommen.

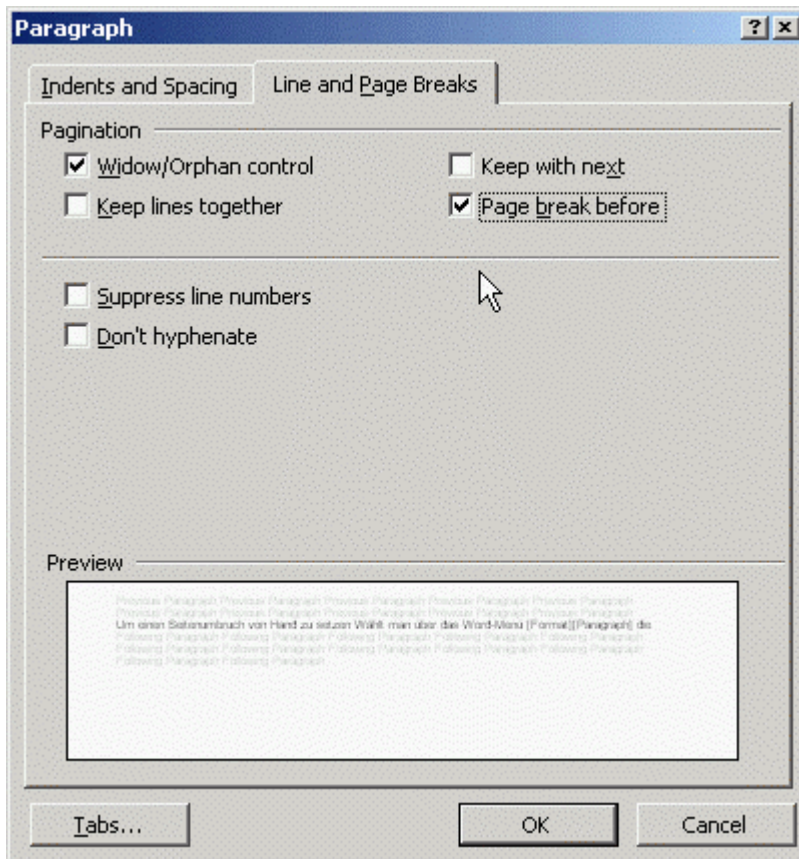


Figure 8 Seitenumbruch

¹⁰ Die Generierung von Hilfe-Dateien basiert beispielsweise auf einem solchen Stylesheet.

Bilder und Graphiken

Natürlich unterstützt WordXML auch das Einbinden von Graphiken und Bildern. Allerdings unter der Einschränkung, dass nur Bilder und Graphiken unterstützt werden, die **eingefügt und verknüpft** werden (insert and link). Andere Arten des Einfügens werden in der aktuellen Version noch nicht unterstützt.

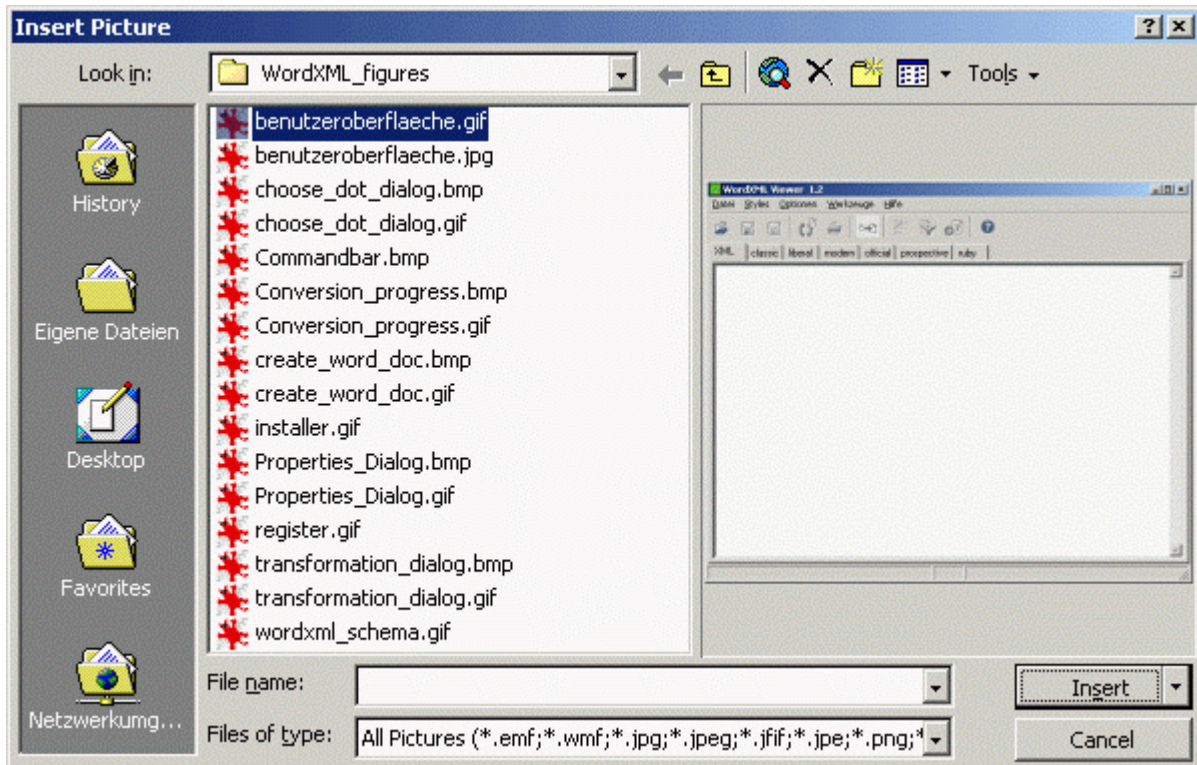


Figure 9 Bilder einfügen

Zu jedem Bild kann eine Bildbeschreibung angegeben werden. Diese erscheint dann sowohl unterhalb des Bildes als auch beispielsweise im Abbildungsverzeichnis, wenn man eine [Windows-Hilfe](#) generiert.

Tabellen und Listen

Tabellen und Listen können einfach wie in Word bekannt verwendet werden. Tabellen können genau wie Bilder eine Beschreibung haben, die – unterstützt vom entsprechenden Stylesheet – unter der Tabelle und im Tabellenverzeichnis verwendet werden.

Innerhalb einer Tabellenzelle dürfen Listen, Aufzählungen und Abbildungen verwendet werden.

In der gegenwärtigen Version wird das Verschmelzen von Tabellenzellen nicht unterstützt!

Erzeugen von Indizes

Microsoft Word unterstützt die Erstellung von Indizes zum schnellen Auffinden von Textstellen, die sich auf bestimmte Schlüsselwörter (index keys) beziehen. Um einen Index an einer bestimmten Textstelle einzufügen, setzen Sie den Cursor an die gewünschte Textstelle und rufen Sie unter dem Menü [Einfügen] die Option „Index und Tabellen“ auf. Klicken Sie auf den Button „Eintrag markieren“ und bestätigen Sie Ihre Auswahl. Es erscheint ein Dialog, in dem Sie unter anderem eine Liste von Schlüsselwörtern für diese Textstelle angeben können.

Genauere Informationen zur Erstellung von Indizes in Word-Dokumenten entnehmen Sie bitte der Dokumentation zu Microsoft Word.

Indizes in Word werden insbesondere bei der [Generierung von HTML-Hilfe](#) verwendet.

Verlinkung innerhalb des Dokuments

Innerhalb eines Dokuments können Querverweise erstellt werden. Dazu erstellt man zuerst die Sprungziele, d. h. man zeichnet die Stellen im Dokument aus, an die von mindestens einer anderen Stelle im Dokument gesprungen werden soll.

Um eine Textstelle auszuzeichnen, verfährt man wie folgt:

- Setzen Sie den Cursor an die Stelle, die ausgezeichnet werden soll
- Wählen Sie aus dem Menü [Einfügen] die Option „Lesezeichen“ (Bookmark) aus.
- Geben Sie dem Lesezeichen einen Namen ¹¹ und klicken Sie auf den Button „Hinzufügen“. Damit haben Sie das Ziel definiert.

Um ein Ziel anzuspringen, verfahren Sie wie beim erstellen eines Hyperlinks:

- Markieren Sie die Textstelle, die ein Link auf ein Lesezeichen werden soll.
- Wählen Sie aus dem Menü [Einfügen] die Option „Hyperlink“ aus.
- Wählen Sie aus dem erscheinenden Dialog die Option „Ort in diesem Dokument“ aus. In einem Fenster wird Ihnen eine schematische Darstellung aller Lesezeichen angezeigt.
- Wählen Sie ein Lesezeichen aus und bestätigen Sie die Wahl durch anklicken des Buttons „OK“.

Ein Lesezeichen kann von beliebig vielen Textstellen aus angesprungen werden. Die Quellen der Verweise sind dabei nicht von Links mit externen Zielen (zum Beispiel das WWW) zu unterscheiden. Bei der [Generierung eines HTML-Projekts oder einer HTML-Hilfe](#) werden diese internen Links automatisch zu Hyperlinks zwischen den entstehenden Dokumenten konvertiert, d. h. durch die Verwendung von Lesezeichen können Sie die gesamte Linkstruktur einer Webseite erstellen. Dieses Dokument etwa ist voll von Lesezeichen.

Externe Verlinkung, also Links auf Dokumente im World Wide Web oder auf dem Dateisystem erstellen Sie wie gewohnt. Details hierzu entnehmen Sie bitte der Hilfe zu Microsoft Word.

¹¹ Verwenden Sie „sprechende Namen“, also Namen, die eine Aussagekraft haben. Das hilft Ihnen, in Dokumenten mit vielen Lesezeichen die Übersicht zu behalten.

Die Konvertierung

Dieser Abschnitt beschreibt den Vorgang der Konvertierung und seine Ergebnisse. Ziel der Konvertierung ist ein Dokument, das inhaltlich und strukturell dem Word-Dokument entspricht¹². In einer ersten Stufe wird genau eine [XML](#)-Datei erzeugt, welche die Endung `.wxm1` hat. Diese Datei enthält eine Repräsentation des Word-Dokuments, wobei hier noch [Meta](#)- und Layoutinformationen über das Dokument enthalten sind. Dieses [XML](#) ist nicht veränderbar. Mehr dazu lesen Sie [hier](#).

Ausgehend von der [WXML](#)-Datei werden über Stylesheet-Transformationen die gewünschten Ergebnisse in der Form [XML](#), [HTML](#) oder [TEXT](#) erzeugt¹³. Hier herrscht absolute Freiheit bezüglich des gewählten Formats. Über entsprechende Stylesheets kann die Ausgabe in jede beliebige Form gebracht und damit an eventuell gegebene Umgebungen angepaßt und validiert werden.

Mit WordXML wird bereits ein Stylesheet mitgeliefert, welches [WXML](#) nach [XML](#) transformiert. Für die Ergebnisdatei existiert eine [DTD](#) sowie diverse weitere Stylesheets ([XSLT](#) und [CSS](#)) für die Konvertierung nach [HTML](#) und die [Generierung von HTML-Projekten sowie von Windows-Hilfe-Projekten](#).

Die mitgelieferten Dateien erfüllen mehrere Zwecke. Erstens dienen sie als Demoprojekt, um die Funktionalität von WordXML zu zeigen. Zweitens implementieren sie umfangreiche Funktionalität, die produktiv und professionell genutzt werden kann¹⁴ und drittes sind sie für den erfahreneren Anwender eine Anleitung für die Erstellung eigener Stylesheets¹⁵.

¹² Es sei hier noch einmal darauf hingewiesen, daß Word nicht als WYSIWYG-Editor im eigentlichen Sinn verwendet wird. Zwar gleichen die in der Installation erzeugten Ergebnisse sehr dem Ursprungsdokument, aber das liegt eher an den zur Darstellung verwendeten Stylesheets. Schliesslich trennt XML ja Daten und Strukturen vom Layout.

¹³ Immer wenn von Ausgabeformaten die Rede ist, ist mit HTML auch automatisch von XHTML die Rede. Die Ausgabe hängt immer von entsprechenden Stylesheet ab; ein Stylesheet kann auch für die Ausgabe von XHTML entworfen sein.

¹⁴ Diese Hilfe selbst wurde mit den mitgelieferten Stylesheets erstellt!

¹⁵ Auch der unerfahrene Anwender kann sich versuchen, indem er die bestehenden Stylesheets modifiziert. Allerdings sollte man nie vergessen, eine Sicherungskopie der Dateien zu erstellen.

Starten der Konvertierung

Die Konvertierung eines Dokuments, daß mit der WordXML Dokumentvorlage erstellt worden ist, kann auf zwei Arten gestartet werden; entweder über das Menü „XML Convert“ oder über den icoya-Button auf dem Commandbar.

Wird die Aktion gestartet, so erscheint im Status-Balken von Word die Meldung „Converting <datei>“, wobei <datei> für den Namen der gegenwärtig geöffneten Datei steht.

Danach werden die entsprechenden Software-Module geladen. Dies kann – insbesondere bei der ersten Konvertierung nach einem Neustart von Word – einige Sekunden dauern.

Sind die Module geladen, erscheint ein Progress-Bar auf dem Bildschirm, der über die Anzahl der Paragraphen im Dokument insgesamt sowie die Anzahl der bereits abgearbeiteten Paragraphen und die verbleibende Zeit für die Konvertierung Auskunft gibt.

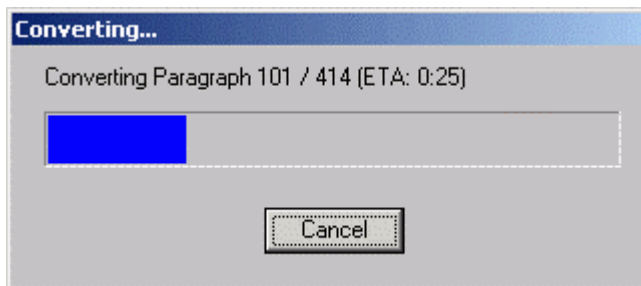


Figure 10 Statusanzeige der laufenden Konvertierung

Wird die Konvertierung erfolgreich abgeschlossen, erscheint ein „Datei speichern“-Dialog. Unter dem angegebenen Namen wird sowohl eine Datei mit der Endung `.wxm1` als auch die eigentlich Ergebnisdatei mit der dem Ausgabeformat entsprechenden Endung abgelegt.

Hinweis: es ist darauf zu achten, daß in den Ergebnisdateien verschiedene weitere Dateien referenziert werden. Dazu gehören in erster Linie die [CSS](#)-Datei, aber auch alle Bilder und eventuell eine [DTD](#). Da die Referenzen auch relative Pfade sein dürfen, ist darauf zu achten, daß das Ergebnis so gespeichert wird, daß diese Pfade noch stimmen. Fehlende Bilder oder das Ignorieren einer [CSS](#)-Datei kommen häufig von solchen falschen Pfaden.

Das Konvertierungsergebnis

Wie bereits erwähnt in der [Einleitung](#) erwähnt, besteht die Konvertierung aus mehreren Schritten. Im ersten Schritt wird [WXML](#) erzeugt. Da dieses Format sehr ausführlich und detailliert ist, eignet es sich nur bedingt für die [Validierung](#) bzw. Weiterverarbeitung zu Anzeigedaten via [XSLTs](#). Stylesheets und [DTDs](#) für [WXML](#) sind sehr umfangreich und kompilziert, da sie in direktem Verhältnis zur Größe der Tag- und Attributemengen und der Komplexität der [WXML](#)-Datei stehen¹⁶. Die folgende Abbildung zeigt den prinzipiellen Aufbau von [WXML](#).

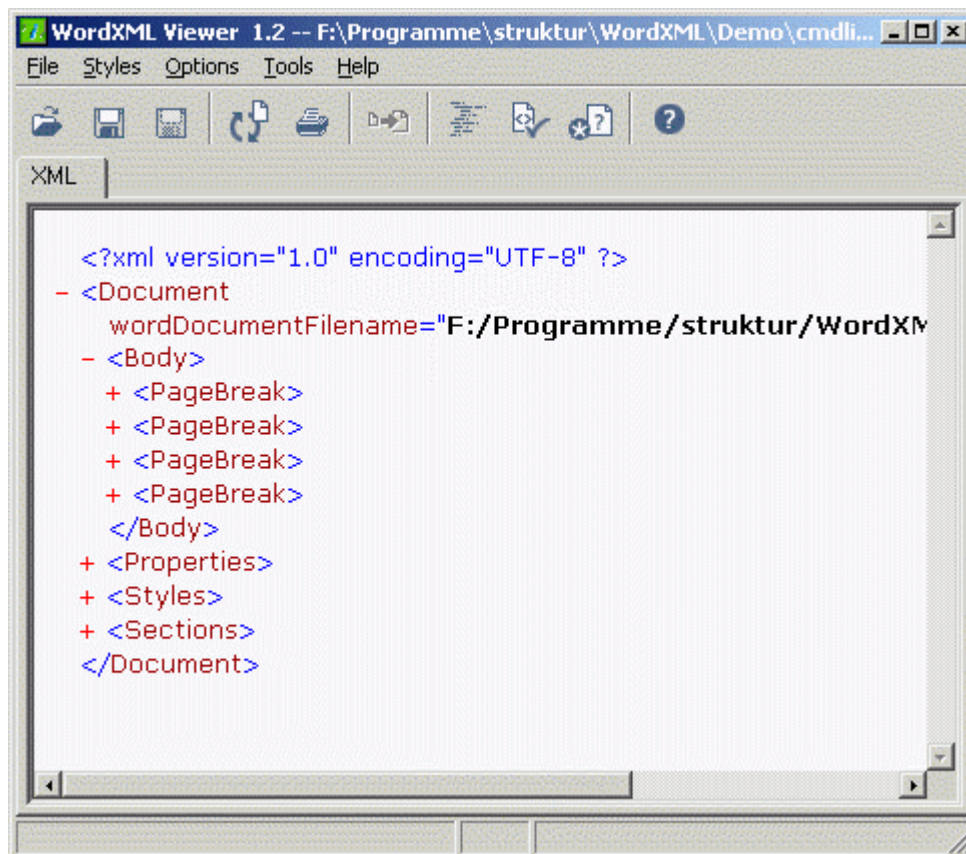


Figure 11 Struktur eines WXML-Dokuments, betrachtet im ResultViewer

Ein Minus (-) vor einem Tag bedeutet, daß das Tag expandiert wurde, d. h. sein gesamter „Inhalt“ sichtbar ist (hier: `<Document>`, `<Body>`). Ein Plus (+) bedeutet, daß das Tag kollabiert ist, was bedeutet, daß es keinen Inhalt anzeigt. Kollabierte Tags können durch anklicken des Plus-Zeichens expandiert werden.

Der prinzipielle Aufbau ist wie folgt:

- Das Wurzelement ist `<Document>`. Es enthält die gesamte Information über das Dokument, also Inhalt, Formatierung und [Metadaten](#). Ist dieses Tag kollabiert, sieht man nichts außer ihm.
- Unterhalb des Wurzelements `<Document>` sind vier Abschnitte, die durch die Tags `<Body>`, `<Properties>`, `<Styles>` und `<Sections>` gekennzeichnet bzw. geklammert sind. Alle diese Sektionen werden zur originalgetreuen Wiederherstellung des Dokuments benötigt. Für die maschinelle Weiterverarbeitung ist dahingegen meistens nur der Inhalt von `<Body>` interessant.
- Der Inhalt von `<Body>` ist in eine beliebige Anzahl von Seiten unterteilt; jede Seite beinhaltet Paragraphen, Listen, Tabellen, Bilder und so weiter. [Seitenumbrüche](#) werden bereits im Word-Dokument festgelegt. Für weitere Informationen hierüber betrachte man sich einfach eine generierte [WXML](#)-Datei, etwa die aus dem [Anwendungsbeispiel](#).

¹⁶ Viele Tags und Attribute von WXML sind in der Praxis (bislang) nicht relevant oder für den internen Gebrauch. So fallen bei den aktuell implementierten Konvertierungen nach HTML die meisten Tags und Attribute weg. Man erzeugt sie trotzdem, um eine breite Front von (zukünftigen) Anwendungen zu unterstützen.

Man kann sich auch einfach eine generierte [WXML](#)-Datei im [ResultViewer](#) anschauen und mit dem Ursprungsdokument und dem generierten [XML](#) vergleichen, um ein besseres Verständnis für dieses Format zu entwickeln. Für den Anwender spielt dieses Format eigentlich solange keine Rolle, bis er [eigene Stylesheets für die Weiterverarbeitung entwickeln](#) will.

Zur Filterung der Daten sind bereits Stylesheets im Lieferumfang enthalten. Das Stylesheet `xml2cust.xsl` akzeptiert als Eingabe eine beliebige [WXML](#)-Datei und generiert als Ausgabe ein deutlich vereinfachtes [XML](#)-Dokument. Dieser Zwischenschritt wird immer ausgeführt. Die entstehende [XML](#)-Datei ist die eigentliche Basis für weitere Arbeitsschritte, etwa die [Generierung einer Windows-Hilfe bzw. eines HTML-Projekts](#). Mehr finden Sie [hier](#).

Ein dritter Konvertierungsschritt entfernt aus dem Ergebnis-[XML](#) noch Attribute, die nur für den internen Gebrauch sind¹⁷. Eventuell werden auch Tags umbenannt. Dieser Schritt erfolgt für den Benutzer transparent, d. h. es wird keine Datei als Zwischenergebnis gespeichert. Die Ausgabe ist das Ergebnis nach dem filtern der Datei. Geleistet wird dies durch das Stylesheet `filter.xsl`. Diese Datei muß bei einer Änderung von `xml2cust.xsl` an deren neues Ausgabeformat angepaßt werden.

Hinweis: Will man ein anderes XML generieren als das vom mitgelieferten Stylesheet `xml2cust.xsl`, so muß man ein [XSLT](#)-Stylesheet entwickeln, welches als Eingabe eine beliebige [WXML](#)-Datei akzeptiert und in das gewünschte [Format](#) konvertiert¹⁸. Diese Möglichkeit bietet nahezu unbegrenzte Möglichkeiten zur Anpassung an eigene, eventuell bereits bestehende, Lösungen und Systeme. Soll die Erzeugung automatisch vor sich gehen, so muß das Stylesheet `xml2cust.xsl` mit dem Inhalt der neuen Lösung überschrieben werden. Eventuell genügen auch nur ein paar Änderungen in `xml2cust.xsl`. Auf jeden Fall ist das Stylesheet `xml2cust.xsl` eine gute Vorlage für eigene Entwicklungen. Und auf gar keinen Fall darf man es versäumen, die Datei `filter.xsl` an die neue Ausgabe anzupassen.

Die [struktur AG](#) berät Sie gerne bei der Änderung, Anpassung und Erstellung neuer Stylesheets zur Integration in Ihre eigenen Lösungen.

¹⁷ Insbesondere sind dies die Attribute „start“ und „end“, die die Byteposition des aktuellen Paragraphen im Word-Dokument angeben. Sie werden benötigt, um im Fehlerfall eine exakte Meldung zu generieren, spielen sonst aber keine Rolle.

¹⁸ Ein einfaches Beispiel wäre die Umbenennung der Tags im Vergleich zur eigentlichen Ausgabe. Aber Vorsicht, die mitgelieferten Stylesheets für die Generierung der Hilfe oder die Darstellung als HTML müssen dann nicht mehr funktionieren. Dies gilt für alle Änderungen gegenüber dem von `xml2cust.xsl` vorgegebenen Format.

Validierung des Ergebnisses

In WordXML werden [Konvertierungsergebnisse](#) optional gegen [DTDs](#) validiert, d. h. das ausgegebene XML wird auf die Gültigkeit gegenüber einer strukturbeschreibenden Grammatik – der [DTD](#) – überprüft.

Bei der Konvertierung wird explizit eine Validierung vorgenommen, wenn im Ergebnis eine [DTD](#) referenziert wird¹⁹. Fehlt diese Referenz wird nicht validiert, sonst gegen die spezifizierte [DTD](#). Schlägt die Validierung fehl, so bricht die [Konvertierung](#) mit einer Fehlermeldung ab. Eine [WXML](#)-Datei wird jedoch immer erzeugt. Ist keine [DTD](#) eingebunden, so kann eine Validierung auch noch zu einem späteren Zeitpunkt durchgeführt werden, z. B. im [ResultViewer](#).

In WordXML wird standardmäßig die [DTD](#) `custom.dtd` eingebunden. Sie befindet sich im Verzeichnis „Doctypes“ im WordXML Installationsverzeichnis.

Hinweis: Der [ResultViewer](#) benutzt als Basis-Komponente zur Anzeige den Internet Explorer von Microsoft. Dieser ist selbstvalidierend, was bedeutet, daß eine [XML](#)-Datei, die eine [DTD](#) referenziert, sofort beim Laden validiert wird. Ist die Datei nicht gültig, so kann sie auch nicht angezeigt werden. Stattdessen erscheint eine Fehlermeldung im Anzeigefenster. Eine häufige Ursache ist das spätere Einbinden von [DTDs](#) (sonst wäre bereits die Konvertierung fehlgeschlagen), oder eine nachträglich Änderung der [DTD](#) oder der [XML](#)-Datei. Von beidem ist dringend abzuraten, außer zu Entwicklungszwecken.

¹⁹ Details zur Validierung entnehmen Sie bitte einschlägiger Literatur oder dem XML-Standard des W3C.

Erzeugung von XML, HTML und TEXT

Um WordXML als Teillösung in größere, eventuell bereits bestehende Systeme zu integrieren bzw. um die Ausgabe zu bereits bestehenden [DTDs](#) konform zu machen, muß [die XML-Ausgabe](#) entsprechend angepaßt werden.

Zur Erzeugung von spezifischem [XML](#) gibt es zwei Möglichkeiten.

Die erste Möglichkeit besteht darin, ein [XSLT](#)-Stylesheet zu entwickeln, welches das im ersten Konvertierungsschritt ausgegebene [WXML](#) in das gewünschte Format transformiert. Soll das neue Format das Standardformat bei der Konvertierung werden, so muß das Stylesheet `xml2cust.xml` im Verzeichnis Stylesheets mit neuem Inhalt überschrieben oder um weitere Transformationsregeln ([XSL](#)-Templates) erweitert bzw. die bereits bestehenden Regeln modifiziert werden. Es ist darauf zu achten, daß die Datei auf keinen Fall umbenannt werden darf. Das bei der Konvertierung verwendete [XSLT](#)-Stylesheet zur Transformation von [WXML](#) in das Ergebnisformat muß immer `xml2cust.xml` heißen und sich immer im Verzeichnis „Stylesheets“ im Installationsverzeichnis von WordXML befinden. Ist das neue Format nicht das Standardformat, so kann die Transformation mit einem beliebigen [XSLT](#)-Prozessor vorgenommen werden, oder im [ResultViewer](#) über die Option „[XSL Transformation](#)“ im Menü „Werkzeuge“.

Die zweite Möglichkeit besteht darin, daß wesentlich einfachere [Ergebnis-XML](#) über [XSLT](#)-Stylesheets weiterzuverarbeiten. Die Transformation kann mit einem beliebigen XSLT-Prozessor vorgenommen werden, oder im [ResultViewer](#) über die Option „[XSL Transformation](#)“ im Menü „Werkzeuge“.

Als Ausgabeformate stehen [HTML](#), [XHTML](#), [XML](#) und [TEXT](#) zur Verfügung, wobei die Struktur und der Inhalt²⁰ des Ergebnisses natürlich nur vom verwendeten [XSLT](#)-Stylesheet abhängen. Übrigens garantiert die Verwendung von [XSLT](#)-Stylesheets die [Wohlgeformtheit](#)²¹ des Ergebnisses.

Detaillierte Informationen zu [XML](#) und [XSLT](#) entnehmen Sie bitte der entsprechenden Literatur. Eine Beschreibung der Möglichkeiten würde sonst den Rahmen dieses Dokuments sprengen. Für die professionelle Beratung und Unterstützung bei der Entwicklung von [XSLT-Stylesheets](#) steht Ihnen die [struktur AG](#) gerne zur Verfügung.

²⁰ Der Inhalt ist eine Teilmenge des Inhalts des Originaldokuments, angereichert durch Aggregation der Daten, wie man es etwa von SQL-Datenbanken kennt.

²¹ Natürlich gilt dies nur für Ausgabeformate, die – wenn überhaupt – ein Wohlgeformtheit im Sinne der von XML kennen.

Generierung einer HTML-Hilfe

[HTML](#)-Hilfe ist das gegenwärtige Standardformat für Windows-Software. Im Prinzip besteht eine Windows-Hilfe aus beliebig vielen, (nicht notwendig) miteinander verlinkten [HTML](#)-Seiten (genau wie eine Website). Aus diesem Grund unterscheidet sich die Erstellung eines [HTML](#)-Projekts überhaupt nicht von der Generierung einer Hilfedatei²². Tatsächlich wird bei WordXML immer beides gleichzeitig generiert. Im folgenden unterscheiden wir deshalb nicht mehr zwischen diesen beiden Ergebnisvarianten.

Hinweis: Die Generierung einer HTML-Hilfe ist ein Zusatz-Modul für WordXML. Es ist nicht im Standard-Lieferumfang enthalten. Wenn Sie sich nicht sicher sind, ob Ihre Version über dieses Modul verfügt, laden Sie einfach eine XML-Datei in den Resultviewer und rufen dann das Menü „Werkzeuge“ auf. Ist der Eintrag „Hilfe generieren“ deaktiviert, fehlt das Modul. Wenden Sie sich an Ihren Systemadministrator zur Überprüfung Ihrer Lizenz oder kontaktieren Sie die [struktur AG](#).

Um aus einem [HTML](#)-Projekt eine Windows-Hilfe zu generieren, müssen alle HTML-Dateien zusammen mit den Bildern kompiliert werden. Dies besorgt ein Programm namens [HTML Help Workshop](#) von Microsoft. Die Software gehört zu jeder neueren Version von Windows und ist darüberhinaus zum freien Download verfügbar.

Mit diesem Dokument benutzen Sie gerade eine mit WordXML generierte Windows-Hilfe.

Um eine HTML-Hilfe zu erzeugen, geht man wie folgt vor:

1. Man [öffnet eine XML-Datei im ResultViewer](#).
2. Über das [Menü „Werkzeuge“](#) ruft man den Dialog [„Hilfe generieren“](#) auf.
3. Man spezifiziert ein Zielverzeichnis und nimmt alle weiteren Einstellungen für die Hilfe-Generierung vor.
4. Man klickt auf den Button „Generieren“.

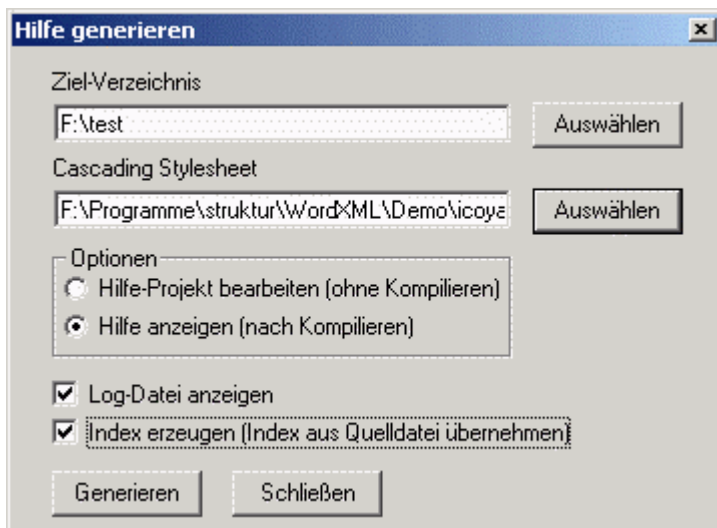


Figure 12 Einstellungen zur Generierung eines HTML-Projekts bzw. einer HTML-Hilfe

Bei der Auswahl des Zielverzeichnisses ist darauf zu achten, das eventuell [eingebettete Bilder](#) in einem Verzeichnis relativ dazu liegen, so daß die Pfade immer noch stimmen. Im Zweifel sind die Bilder dorthin zu kopieren bzw. zu verschieben.

²² Dies ist aus zwei Gründen nicht ganz wahr. Erstes wird die Erstellung von Indizes in einem HTML-Projekt nicht unterstützt, jedoch in einer Hilfe; zweitens ist ein HTML-Projekt eine Vorstufe zur Hilfe, d. h. die Hilfe muß noch kompiliert werden. Die Unterschiede sind jedoch nicht signifikant.

Einstellungen für die Hilfe-Generierung

Für die Generierung der Hilfe bzw. eines [HTML](#)-Projekts können einige Einstellungen vorgenommen werden, um das Ergebnis besser an die eigenen Wünsche und Anforderungen anpassen zu können.

Als erstes muß ein Zielverzeichnis ausgewählt werden. In diesem Verzeichnis werden alle Ergebnisdateien sowie alle temporären und Log-Dateien abgelegt. Ist kein Zielverzeichnis spezifiziert, erscheint eine Fehlermeldung auf dem Bildschirm.

Das selbe gilt für das „[Cascading Stylesheet](#)“. Verwenden Sie jedes beliebige Stylesheet, um das Erscheinungsbild Ihrer Site bzw. der Online-Hilfe zu verändern und anzupassen. Sie können selbst Stylesheets erstellen, oder eines der mitgelieferten Stylesheets verwenden.

Als Optionen stehen die Erzeugung eines Projekts zur Bearbeitung und die Kompilierung mit anschließend sofortiger Anzeige des Ergebnisses — also der Hilfe – zur Verfügung. Im ersten Fall wird der [HTML Help Workshop](#) geöffnet. Dies ist eine Art Entwicklungs-Werkzeug für HTML-Hilfen. Verwenden Sie diese Option, wenn Sie als erfahrener Anwender die Hilfe von Hand nachbearbeiten wollen oder wenn Sie sich als Anfänger mit der Erstellung von Windows-Hilfen vertraut machen wollen.

Warnung: Wenn Sie ein Hilfe-Projekt von Hand bearbeiten, verändern Sie dabei maschinell generierte Dateien. Diese Dateien werden bei der nächsten Generierung wieder neu geschrieben. Somit gehen alle Änderungen verloren! Erstellen Sie deshalb Sicherungskopien oder wechseln Sie das Zielverzeichnis.

Wenn Sie sich das Projekt anzeigen lassen und aus dem [HTML Help Workshop](#) dann sofort kompilieren, erhalten Sie das selbe Ergebnis wie wenn Sie die Option „Hilfe anzeigen (nach Kompilieren)“ im Einstellungs-Dialog wählen.

Desweiteren können Sie sich eine Log-Datei über den Generierungsvorgang anzeigen lassen. Diese Datei kann bei der Analyse eventuell auftretender Fehler hilfreich sein. Die Datei wird im [externen Editor](#) angezeigt. In der Regel wird man von dieser Option keinen Gebrauch machen.

Als letzte Option besteht die Möglichkeit einer Indexerzeugung unter Verwendung des [Index im Quelldokument](#). In einem Worddokument lassen sich Indizes erzeugen, also eine Menge von Schlüsselwörtern festlegen, über die schnell auf die jeweils für das Wort relevanten Textstellen zugegriffen werden kann. Wie Sie diesen Index erstellen, lesen Sie in der Hilfe von Word nach. Wird diese Option gewählt, so werden im erzeugten [HTML](#)-Code spezielle Tags angelegt bzw. Objekte eingebettet, die diesen speziellen Zugriffsmechanismus in der HTML-Hilfe unterstützen. In einem normalen Browser werden die Objekte als fehlend dargestellt; die Anzeige ist falsch. Wählen Sie die Option, wenn Sie im Dokument schon Indizes verwenden und eine HTML-Hilfe erstellen wollen. Sonst ist von dieser Option abzuraten, da sie im besten Fall nichts macht.

Hilfe-Projekt bearbeiten

Der [HTML-Help Workshop](#) von Microsoft ist eine kleine Entwicklungsumgebung, mit der Hilfe-Projekte von Hand erstellt und modifiziert und erweitert werden können. WordXML erledigt praktisch alle Aufgaben, die für die Generierung einer Hilfe notwendig sind. Sollte trotzdem der Wunsch bestehen, die Hilfe manuell zu verändern, so kann man im „Hilfe generieren“-Dialog die Option „Hilfe-Projekt bearbeiten“ wählen. Das Projekt wird nicht kompiliert, sondern der [HTML Help Workshop](#) von Microsoft wird geöffnet.

In der Workshop Oberfläche kann man den Index erweitern, das Inhaltsverzeichnis umsortieren, das Layout verändern, und viele Dinge mehr. Aus dem Workshop heraus kann man das modifizierte Projekt kompilieren.

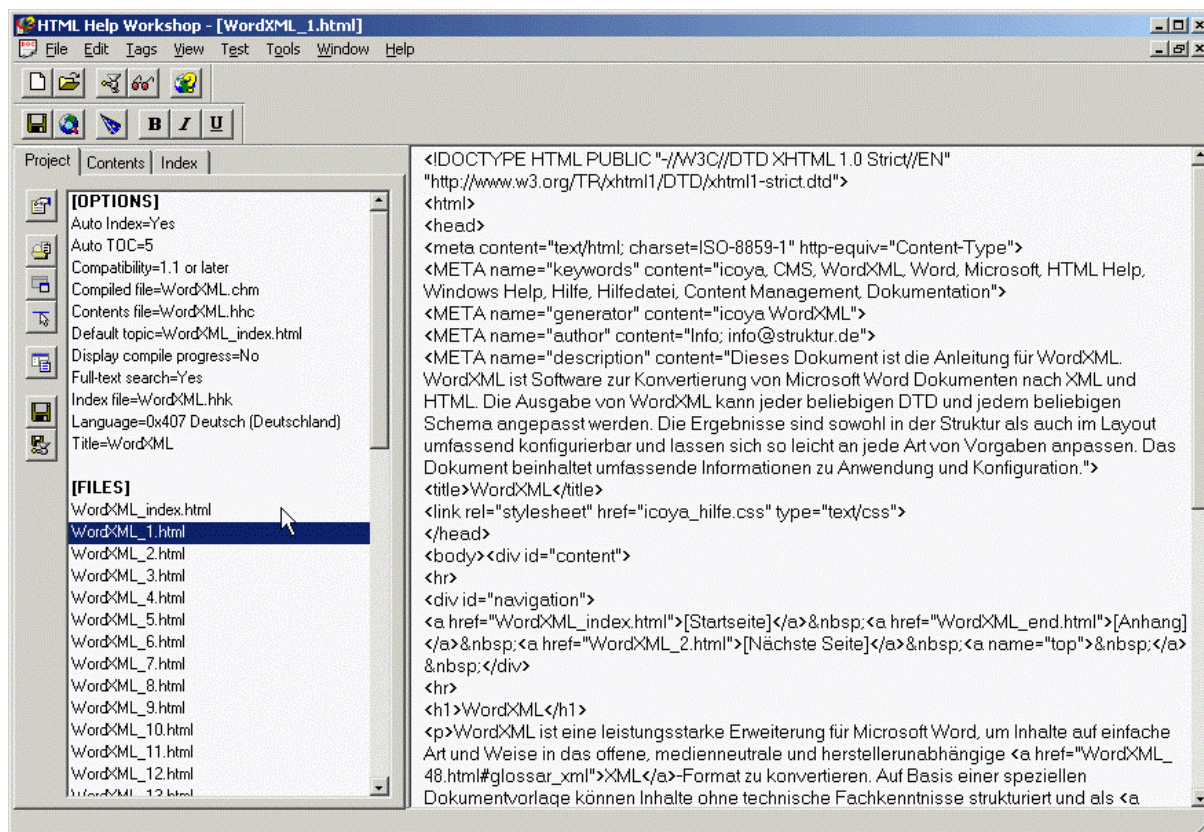


Figure 13 Der HTML Help Workshop von Microsoft

Hinweis: Wird ein Hilfe-Projekt im Workshop modifiziert, sollte man das Projekt unter einem anderen Namen abspeichern. Da die Änderungen in der Projektdatei (.hhp) gespeichert werden, diese aber von WordXML automatisch generiert wird, besteht die Gefahr des versehentlichen überschreibens bei der nächsten Generierung.

Für eine ausführliche Dokumentation des Workshops und die Hilfe-Generierung mit diesem Werkzeug sei auf die Dokumentation von Microsoft bzw. die Online-Hilfe des Workshops verwiesen.

Für all diejenigen, die sich nicht um technische Details der Hilfe-Generierung kümmern wollen und mit den standardmäßig vorhandenen Möglichkeiten zufrieden sind, gibt es die Möglichkeit, das Projekt sofort aus dem [ResultViewer](#) heraus zu kompilieren.

Hinweis: Die Generierung einer HTML-Hilfe funktioniert nur mit dem [XML](#), das mit dem mitgelieferten Stylesheet erzeugt worden ist. Ändert man das Stylesheet für die Ergebniserzeugung (also für die Transformation von [WXML](#) nach [XML](#), [XHTML](#), [HTML](#) oder [TEXT](#)), so muß man das Stylesheet, welches für die Generierung der Hilfe verantwortlich ist, den neuen Gegebenheiten anpassen.

Hilfe anzeigen

Wird diese Option im Dialog gewählt, wird sofort nach dem erfolgreichen Kompilieren des Projekts die Kompilierte Hilfedatei geöffnet. Sie ist das Endprodukt der Hilfe-Generierung und besteht aus einer einzigen Datei mit der Endung `.chm`. Diese Datei beinhaltet neben allen [HTML](#)-Seiten auch alle Bilder und das verwendete [CSS](#). Somit kann die Datei einfach verschoben oder per Email versandt oder als Teil einer Software verteilt werden.

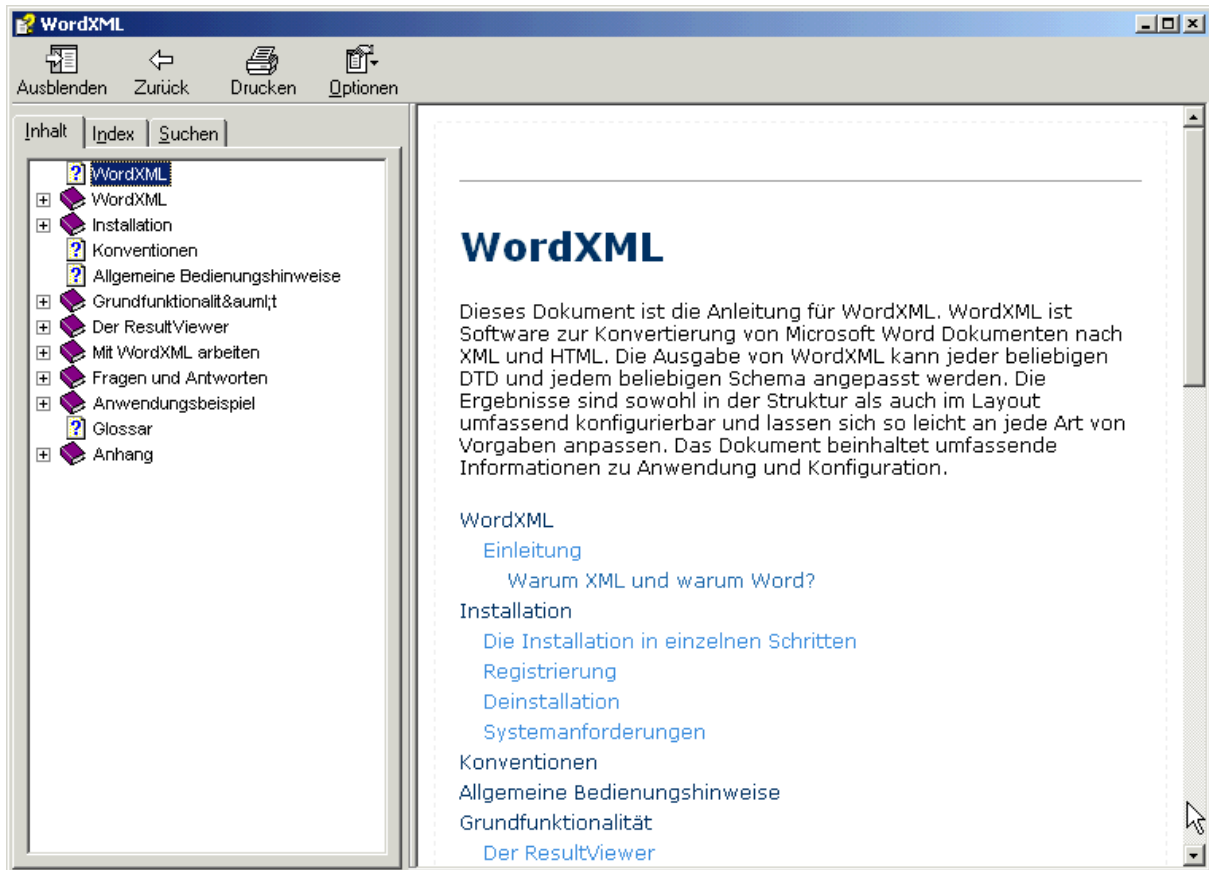


Figure 14 Ergebnis einer Hilfe-Kompilierung

Verwendung von Metadaten

Die [Verwendung von Metadaten](#) spielt bei der Generierung von Hilfe-Projekten eine große Rolle. Hier kann ein alternativer Titel für das Dokument angegeben werden. Dieser Titel wird in das `<title>`-Tag der [HTML](#)-Ergebnisdateien gespeichert und somit in der Titelleiste des [HTML Help Workshop](#) Viewers bzw. des verwendeten Browsers angezeigt.

Im den Feldern „Autor“ (Author) bzw. „Firma“ (Company) können neben dem jeweiligen Namen und durch ein Semikolon getrennt die jeweiligen [URLs](#) mit angegeben werden. Damit werden die Links in den Fußzeilen der Hilfe erzeugt.

Im Feld „Schlüsselwörter“ wird eine Komma-separierte Liste von Suchbegriffen angegeben, die ebenfalls im [HTML](#) im entsprechenden `<meta>`-Tag mitgespeichert wird.

Im Feld „Kommentar“ eingegebener Text erscheint in der Hilfe über dem Inhaltsverzeichnis. Dies gibt einem die Möglichkeit, eine zusätzliche Kurzbeschreibung des Dokuments zu speichern.

Es ist auf jeden Fall zu empfehlen, die Metadaten sehr viel und sehr oft zu bentzen.

Fragen und Antworten

In diesem Abschnitt werden Antworten auf häufig gestellte Fragen gegeben. Es werden die Bereiche [Installation](#), [Allgemeines](#), [Registrierung](#), [Arbeiten mit dem ResultViewer](#), [Generierung von HTML-Hilfe](#)) und [Deinstallation](#) abgedeckt.

Installation/Registrierung

1. Frage: Wo finde ich eine Installationsversion des [ResultViewers](#)?
Antwort: Der ResultViewer ist fester Bestandteil der WordXML Softwaredistribution. Eine gesonderte Installationsversion ist nicht verfügbar.
2. Frage: Wie installiere ich den [ResultViewer](#)?
Antwort: Der ResultViewer wird automatisch mit WordXML installiert. Nach der Installation kann er über eine Verknüpfung in der WordXML Programmgruppe gestartet werden.
3. Frage: Woher bekomme ich einen Software-Schlüssel für die Registrierung?
Antwort: Hat Ihre Firma WordXML für Sie erworben, so wenden Sie sich an Ihren Systemadministrator. Anderenfalls wenden Sie sich direkt per Email an die struktur AG unter sales@struktur.de.
4. Frage: Ich habe einen gültigen Software-Schlüssel, jedoch schlägt die Registrierung fehl. Was mache ich falsch?
Antwort: Der Software-Schlüssel wird individuell für jeden Benutzer vergeben und muß zu den Benutzerangaben passen. Überprüfen Sie die Rechtschreibung und achten Sie auf nicht sichtbare Zeichen (Leerzeichen). Gelingt die Registrierung trotzdem nicht, lassen Sie Ihren Schlüssel von der struktur AG unter Angaben der Benutzerdaten verifizieren.
5. Frage: Ich habe eine lizenzierte Version von WordXML installiert. Trotzdem stehen mir nicht alle Funktionen zur Verfügung. Warum nicht?
Antwort: Bestimmte Funktionen von WordXML bedürfen einer gesonderten Lizenzierung, beispielsweise die Generierung von Hilfe-Projekten.
6. Frage: Während der Installation wird ein Fehlerdialog mit der Meldung ‚Fehler beim Schreiben der Datei ...‘ angezeigt. Was ist die Ursache?
Antwort: Vor der Installation von WordXML sollten alle Programme geschlossen werden. Ist dies nicht der Fall, so kann es sein, daß bei der Installation auf eine Datei schreibend zugegriffen werden soll, die durch ein anderes Programm benutzt wird und somit gegen Schreibzugriffe geschützt ist. Brechen Sie die Installation ab, schließen Sie alle Programme und starten Sie die Installation erneut.
7. Frage: Wie finde ich nach der Installation die Programmgruppe von WordXML?
Antwort: Die WordXML-Programmgruppe ist direkt unter [Start][Programme] zu finden. Sofern Sie bei der Installation keinen anderen Namen vergeben haben, heißt sie auch „WordXML“.
8. Frage: Wie wähle ich das Installationsverzeichnis aus bzw. wo finde ich es?
Antwort: Das Installationsverzeichnis wird bei der Installation ausgewählt und kann jedes beliebige Verzeichnis sein, solange auf der entsprechenden Festplatte genügend Speicherplatz vorhanden ist. Sie können es finden, indem Sie in der WordXML-Programmgruppe einen Rechtsklick machen und aus dem erscheinenden Menü die Option „Eigenschaften“

Einzelheiten zur WordXML-Installation finden Sie [hier](#).

Allgemeines

- Frage:** Word bietet mir die Möglichkeit, Dokumente im [HTML](#)-Format abzuspeichern (und zukünftig wohl auch [XML](#)). Wozu brauche ich dann WordXML?

Antwort: Die Speicherung von Word-Dokumenten in [HTML](#) (oder [XML](#)) liefert eine praktisch identische Abbildung der Dokumente eben in einem anderen Format. Es fehlt die Möglichkeit, Daten schon während der Konvertierung zu filtern, zu veredeln oder auszuwerten. All diese Möglichkeiten bietet WordXML über [XSL-Transformationen](#). Der Vorgang ist für den Anwender absolut transparent. Außerdem sind die von Word erzeugten Ergebnisse überfüllt mit überflüssiger Information. Man schaue sich nur einmal eine von Word erzeugte [HTML](#)-Datei an.
- Frage:** Beim Öffnen der Word-Dokumentvorlage `xmlword.dot` erscheint nicht der Commandbar mit den Kurzbefehlen für WordXML. Wo ist er abgeblieben?

Antwort: Wenn man in Word auf dem Toolbar rechtsklickt, so erscheint eine Liste aller verfügbarer Commandbars. Aus der Liste kann dann ausgewählt werden, welche(r) Commandbar(s) angezeigt werden. Der Name des WordXML Commandbar ist „icoya WordXML“. Ist in der Liste dieser Eintrag nicht verfügbar, so ist die falsche Dokumentvorlage geöffnet oder die Vorlage ist beschädigt.

Arbeiten mit dem ResultViewer

- Frage:** Ich kann die im Kapitel ‚Die Benutzung des ResultViewers‘ genannten Beispieldateien nicht finden. Wo befinden sie sich?

Antwort: Die Dateien existieren nicht real. Die Beispiele sollten jedoch einfach auf existierende Dateien zu übertragen sein. [XML](#)-Dateien können einfach durch die Konvertierung von Word-Dokumenten erzeugt werden; Stylesheets sind im Installationsumfang von WordXML enthalten. Details hierzu sind der WordXML Dokumentation zu entnehmen.
- Frage:** Ich habe einen Style bereits geöffnet, finde aber den zugehörigen Reiter nicht. Wo kann ich ihn wiederfinden?

Antwort: Sind viele Styles geöffnet, so kann es sein, daß der aktuelle Darstellungsbereich (Fenstergröße des Bildschirms oder Bildschirmgröße im Vollbildmodus) nicht für die Anzeige aller Reiter genügt. In diesem Fall erscheint rechts oben neben den Reitern eine Bildlaufleiste. Versuchen Sie, mit dieser zum bereits geöffneten Style zu navigieren. Kann der Style trotzdem nicht gefunden werden, wurde er geschlossen. In diesem Fall öffnen Sie ihn einfach neu.
- Frage:** Ich kann die [XML](#)-Datei bzw. den zugehörigen Karteikartenreiter nicht mehr finden?

Antwort: Die XML-Ansicht kann nicht geschlossen werden; der Karteikartenreiter bleibt immer sichtbar. Ist er trotzdem nicht im Anzeigebereich, so ist die Ursache dieselbe wie in Frage 2 oben.
- Frage:** Ich verwende [CSS](#). Trotzdem werden die Layoutklassen bei der [HTML](#)-Darstellungen nicht berücksichtigt. Wie behebe ich den Fehler?

Antwort: Die CSS-Datei wird innerhalb der HTML-Datei über einen relativen Pfad referenziert. Die [CSS](#)-Datei muß sich also auch in dem entsprechenden Verzeichnis relativ zur [HTML](#)-Datei befinden. Standardmäßig muß die [CSS](#)-Datei im selben Verzeichnis wie die [HTML](#)-Dateien stehen.
- Frage:** Ich aktiviere eine [HTML](#)-Ansicht, kann aber nur Quellcode sehen. Wie zeige ich wieder [HTML](#) an?

Antwort: In diesem Fall ist der Ansichtsmodus auf ‚[XSL](#)‘ eingestellt, d. h., daß der Quellcode des zur Erzeugung der [HTML](#)-Dateien verwendeten Stylesheets angezeigt wird. Im mittleren Feld der Statusleiste wird ‚[XSL](#)‘ angezeigt. Durch [Anzeige umschalten] im ‚Styles‘-Menü wird der Anzeigemodus wieder auf ‚[HTML](#)‘ gesetzt.
- Frage:** Ich habe eine [CSS](#)-Datei verschoben und den Pfad in allen [HTML](#)-Dateien angepaßt. Trotzdem werden die Styleklassen des ‚cascading stylesheets‘ auf die bestehenden/neuen [HTML](#)-nicht angewendet. Warum?

Antwort: Funktioniert die Übernahme der Styleklassen für bestehende HTML-Dateien nicht, so kann dies zwei Gründe haben: Erstens, es werden die standardmäßig erzeugten [HTML](#)-Dateien verwendet, die automatisch bei bzw. für die Anzeige erstellt werden. Bei jedem Style-Wechsel im Anzeigebereich wird die entsprechende [HTML](#)-Datei neu erzeugt. Mit der Datei werden natürlich auch die geänderten Pfade überschrieben. Grundsätzlich sollten [HTML](#)-Dateien für den Gebrauch in einem gesonderten Verzeichnis - und nicht mit den [XML](#)-Dateien gemeinsam – gespeichert werden. Ist dies nicht die Ursache, so ist der Pfad zur [CSS](#)-Datei nicht korrekt.

Für die Änderung des Pfades für alle Dateien muß der Pfad nicht im HTML-Quellcode, sondern in der entsprechenden [XSL](#)-Datei geändert werden.

Generieren einer HTML-Hilfe oder eines HTML-Projekts

- Frage:** Ich habe eine Hilfe generiert, aber es fehlen alle Bilder

Antwort: In [HTML](#) werden Bilder über ihren [URI](#) eingebunden; dies ist entweder ihr Pfad im Dateisystem oder eine Webadresse. Wenn die Bilder in Word über relative Pfade eingebunden werden, so werden die Bilder aus den [HTML](#)-Ergebnisdateien über die gleichen relativen Pfade referenziert. Kontrollieren Sie die Pfade, indem Sie sich den Quellcode der [HTML](#)-Dateien anschauen. Kopieren bzw. verschieben Sie die Bilder gegebenenfalls. Eine gute Idee ist es auch, als Zielverzeichnis für die [HTML](#)-Dateien das Verzeichnis anzugeben, in dem das Word-Dokument liegt.
- Frage:** Ich habe versucht, ein HTML-Projekt zu erstellen, aber die Generierung scheitert. Warum?

Antwort: Dafür gibt es viele Ursachen. Ein häufiges Problem ist, daß Pfadnamen bei der Hilfe-Generierung keine Leerzeichen enthalten dürfen, da der Hilfe-Compiler dies nicht akzeptiert. Ist dies nicht die Ursache, sollte man die Log-Datei überprüfen. Sie gibt in praktisch allen Fällen Auskunft über die Ursache der Fehlfunktion. Zeigt die Log-Datei keinen Fehler an, so liegt das Problem an dem Stylesheet für die Hilfe-Generierung. Dieses erzeugt dann Code, der nicht zu compilieren ist, obwohl die Transformation ansich einwandfrei funktioniert.
- Frage:** Ich habe Änderungen im Word-Dokument bzw. im Stylesheet vorgenommen. Nach einer Neu-Generierung sind diese Änderungen jedoch nicht zu sehen. Warum?

Antwort: Dies geschieht genau dann, wenn die vorherige Hilfe-Datei noch geöffnet ist. Der Hilfe Compiler ist nicht in der Lage, geöffnete Hilfe-Dateien zu überschreiben. Schließen Sie alle Hilfe-Dateien des Projekts und starten Sie die Generierung erneut.
- Frage:** Kann ich bei der Ausgabe [XHTML](#)-Konformität erzwingen bzw überprüfen?

Antwort: Nein. Es ist Sache des verwendeten Stylesheets, die Konformität gegenüber bestimmten Standards zu gewährleisten (und somit Sache desjenigen, der das Stylesheet erstellt hat). In der aktuellen Version wird die Verifizierung nicht unterstützt, allerdings erzeugt das mitgelieferte Stylesheet [XHTML](#). Auf der Seite des [W3C](#) sind Programme verfügbar, mit denen [XHTML](#)-Konformität überprüft werden kann.
- Frage:** Ich möchte für meine Projekte eine eigene [CSS](#)-Datei verwenden. Geht das?

Antwort: Dieses Feature ist im Moment nicht implementiert, jedoch in Arbeit. Das verwendete Stylesheet trägt den Namen `icoya_hilfe.css` und wird in das angegebene Zielverzeichnis kopiert. Für einzelne Projekte kann man sich damit behelfen, daß man diese Kopie des Stylesheets modifiziert. Will man grundsätzlich ein anderes, kann man die Originaldatei im Installationsverzeichnis im Unterverzeichnis `Help` modifizieren. Insgesamt sind diese Lösungen unschön, da man eventuell mehrere gleichnamige Stylesheets erzeugt, die aber alle für ein anderes Layout sorgen.

Deinstallation

1. Frage: Wie entferne ich den ResultViewer bzw. WordXML von meinem Computer?
Antwort: Der ResultViewer ist fester Bestandteil der WordXML Softwaredistribution. Er kann nicht gesondert deinstalliert werden. Bei einer Deinstallation von WordXML wird der ResultViewer automatisch mit entfernt. WordXML kann über das Deinstallationsprogramm in der WordXML Programmgruppe oder über die Systemsteuerung deinstalliert werden.

Einzelheiten zur WordXML-Deinstallation finden sich in der entsprechenden Dokumentation.

Anwendungsbeispiel

In diesem Kapitel wird anhand von Beispielen jeder mögliche Arbeitsschritt von WordXML erläutert. Die beste Art und Weise, sich mit WordXML vertraut zu machen, ist die Beispiele nachzumachen.

Dokument erstellen

Als erstes wird ein [Dokument erstellt](#). Schreiben Sie einfach einen Text, formatieren Sie ihn mit den icoya Formatvorlagen und speichern Sie ihn ab. Verwenden Sie auch [Tabellen](#), [Listen](#) und [Seitenumbrüche](#). Vergessen Sie auch nicht, [Metainformation](#) zum Dokument zu erfassen. Für Überschriften verwenden Sie unbedingt die icoya_Überschrift-Vorlagen.

Ihr Text könnte etwa so aussehen:

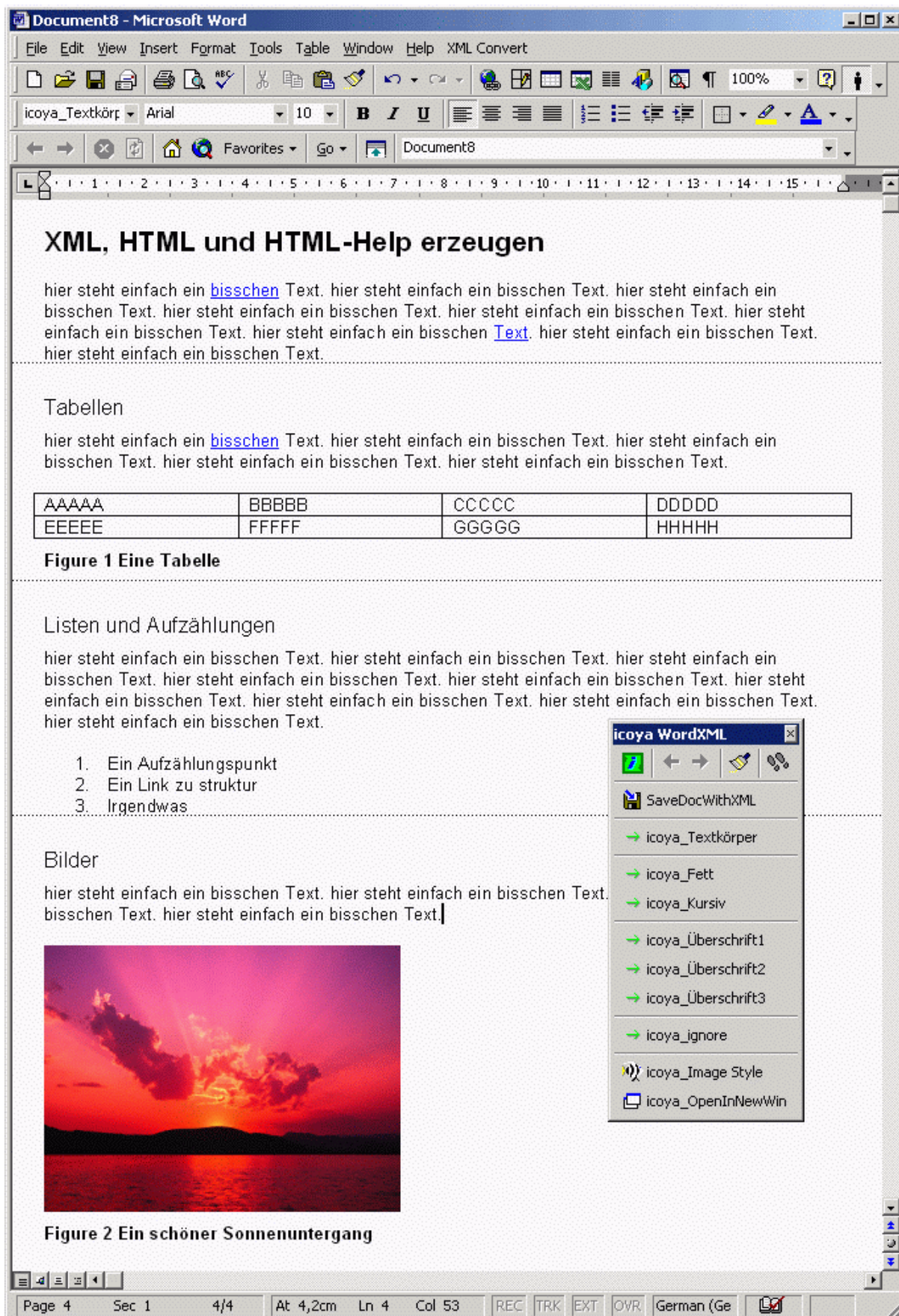


Figure 15 Ein Beispieldokument

Die gestrichelten Linien im Dokument zeigen manuell eingefügte [Seitenumbrüche](#) an. Die Überschriften sind mit den icoya_Überschrift-Vorlagen der jeweiligen Priorität formatiert.

Konvertierung

Die Konvertierung wird einfach wie [oben beschrieben](#) gestartet. Es erscheint der Progressbar, der über den Status der Konvertierung Auskunft gibt. Nach gelungener Konvertierung erscheint ein „Datei speichern“-Dialog. Im angegebenen Verzeichnis wird eine [WXML](#)-Datei und eine [XML](#)-Datei abgespeichert. Je nach Einstellung ist die [Validierung](#) ein Teil der Konvertierung.

Im Fehlerfall wird ein Dialog mit Fehlermeldungen angezeigt. Zusätzlich wird im Installationsverzeichnis von WordXML noch eine Logdatei mit dem Namen `debug.txt` angelegt.

Auch sie gibt Aufschluß darüber, aus welchen Gründen eine Konvertierung gescheitert ist.

Wir empfehlen, die ersten Versuche im Unterverzeichnis „Demo“ des WordXML-Installationsverzeichnisses zu speichern.

Der ResultViewer

Starten Sie jetzt das Programm [ResultViewer](#) in der Programmgruppe von WordXML. [Öffnen Sie die Datei](#) <ergebnis>.xml, wobei <ergebnis>.xml für den Namen der Datei steht, unter der Sie das Konvertierungsergebnis abgespeichert haben. Der Dateiname steht anschließend in der Titelleiste der Viewers.

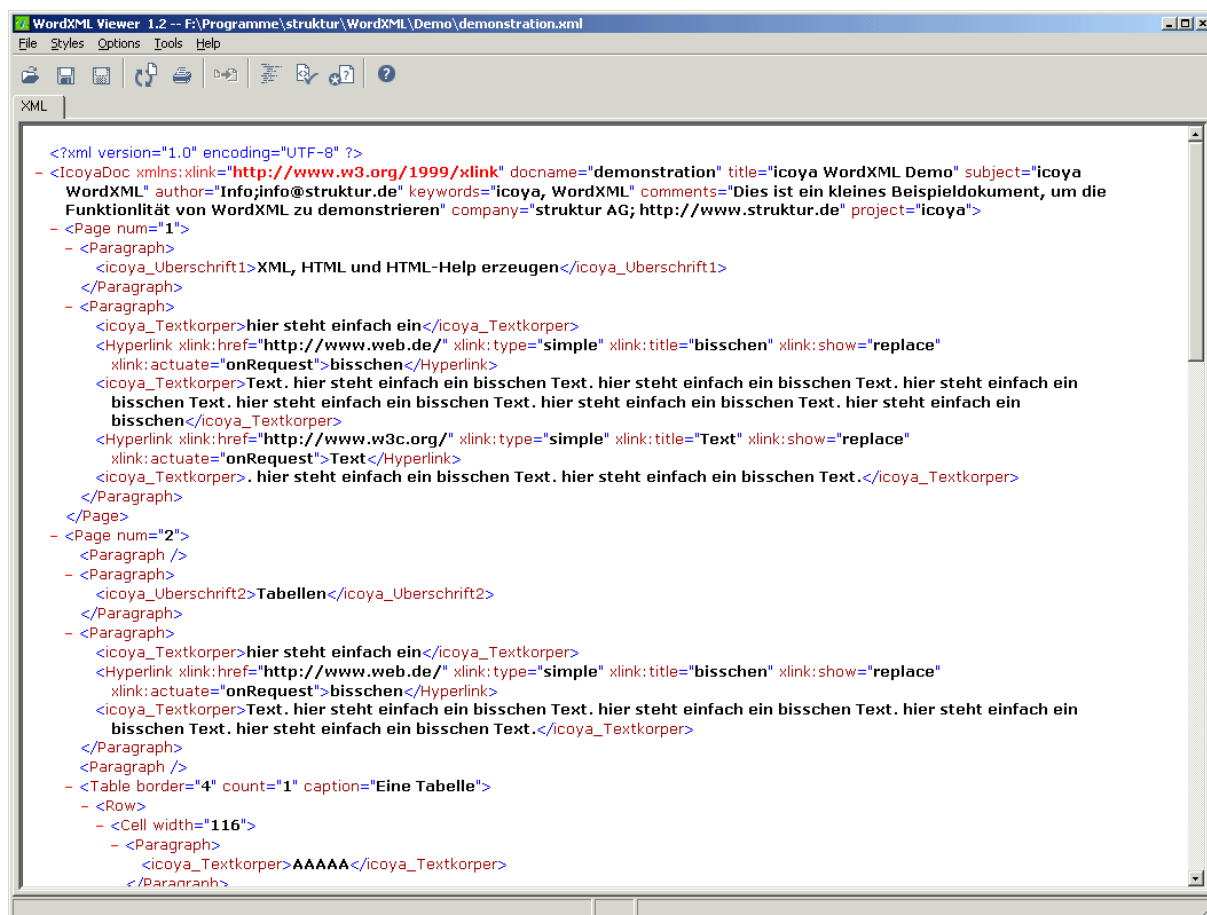


Figure 16 ResultViewer mit geöffneter Ergebnisdatei

Vergleichen Sie nach dem [Öffnen der Datei](#) ruhig einmal das Word-Dokument mit der [XML](#)-Datei. Es gibt Ihnen eine Vorstellung, wie das Ergebnis aussieht und was Sie damit machen können²³.

WordXML wird mit Beispiel-Stylesheets ausgeliefert, die sich ebenfalls im Demo-Verzeichnis unterhalb des Installationsverzeichnisses befinden. Die Stylesheets können das Ergebnis direkt nach [HTML](#) transformieren und binden ein [CSS](#) ein, welches für ein angenehmes Layout sorgt.

²³ Sie können sich ruhig auch die WXML-Datei ansehen. Lassen Sie sich aber nicht verwirren. Normalerweise bleibt Sie für den Anwender transparent, es sei denn, man will die direkte Konvertierung beeinflussen.

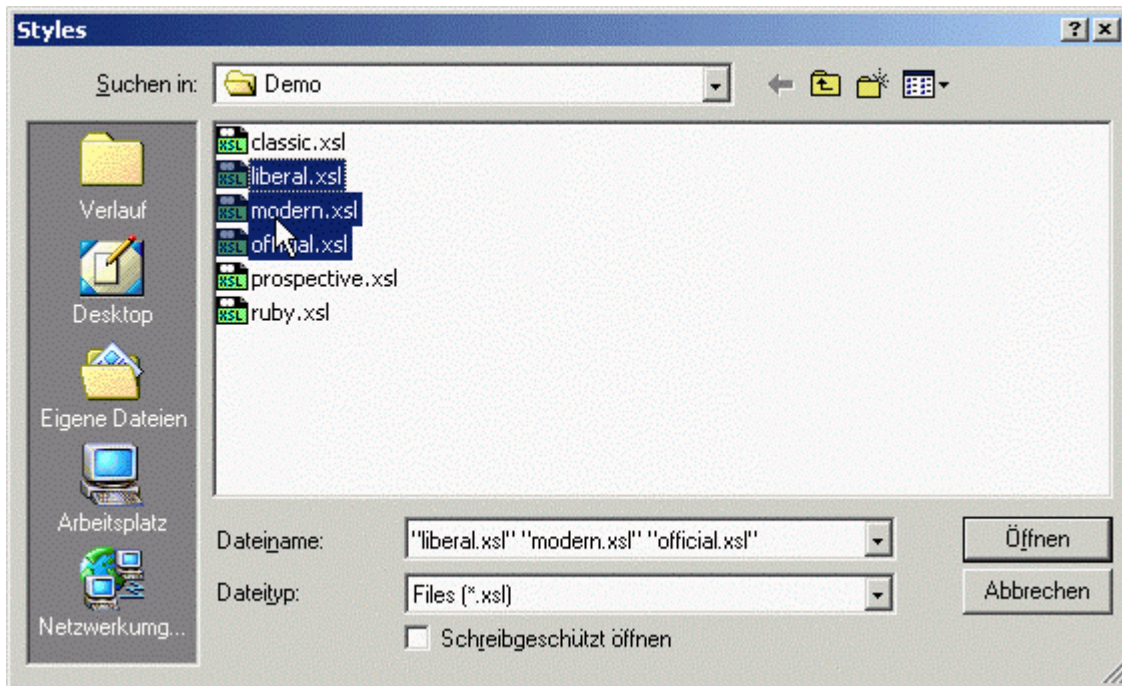


Figure 17 Auswahl eines Stylesheets für die Darstellung als HTML

Der Dialog erlaubt auch die [Auswahl mehrer Stylesheets](#) gleichzeitig. Für jedes Stylesheet wird das Ergebnis erzeugt und angezeigt. Der Zugriff auf die einzelnen Vorschauen erfolgt über die zum jeweiligen Style gehörigen Karteikartenreiter. Die erzeugte Datei wird genau so, wie sie im ResultViewer zu sehen ist, auch in jedem Browser angezeigt.

Wenn Sie den Inhalt Ihres Arbeitsverzeichnisses – hier das Demo-Verzeichnis – kontrollieren, werden Sie feststellen, daß für jeden geöffneten Style eine [HTML](#)-Datei erzeugt worden ist. Der Name der Dateien setzt sich aus dem gewählten Namen des Ergebnisses und – durch einen Unterstrich getrennt – dem Namen des verwendeten Styles zusammen. Die hier angezeigte Datei wird also unter dem Namen `demonstration_liberal.html` abgespeichert.

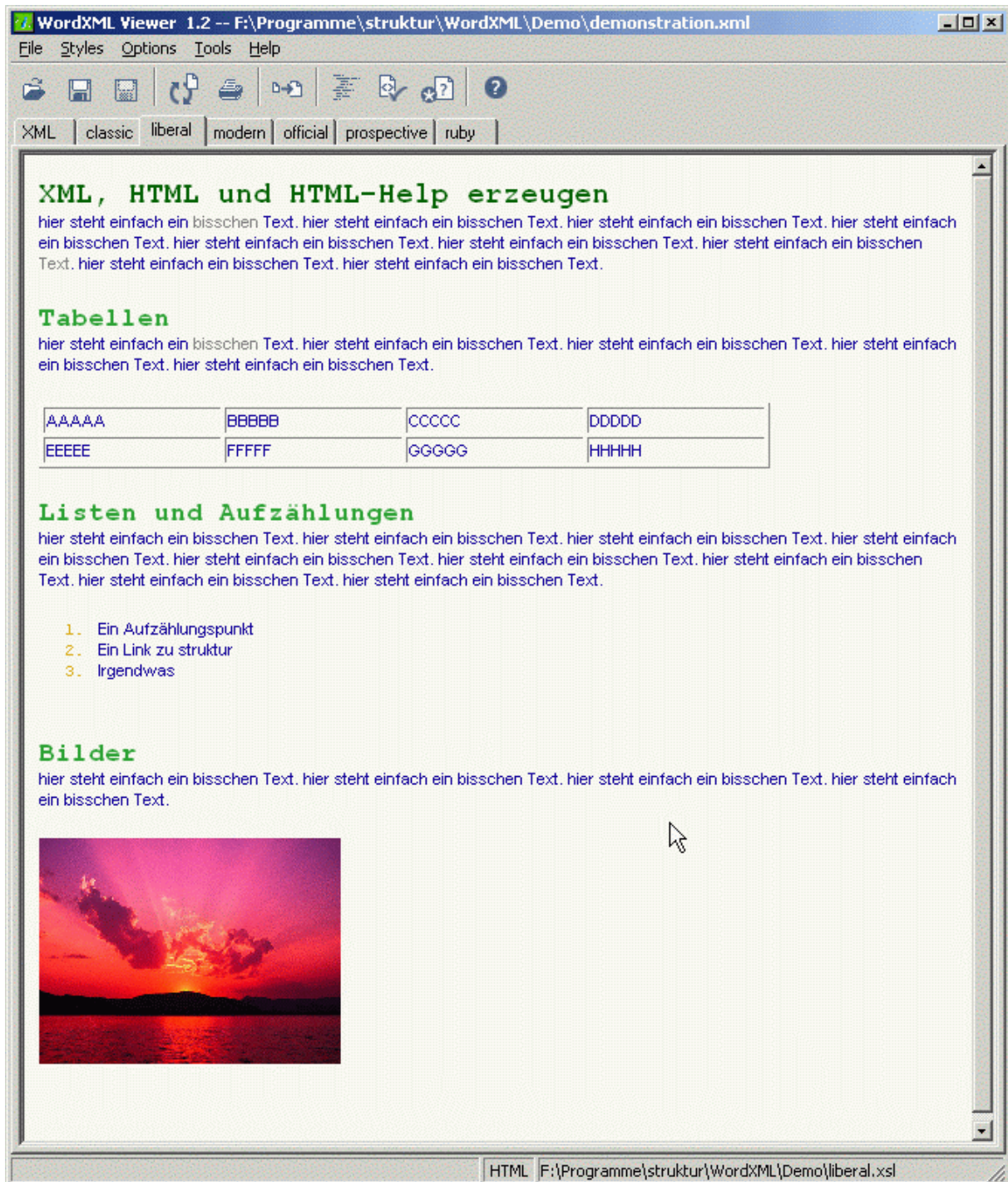


Figure 18 Ansicht des Ergebnisses mit dem HTML-Style "liberal"

Anstelle des Previews kann auch das Stylesheet angezeigt werden, welches die Ausgabe erzeugt. Dafür muß man einfach die [Ansicht umschalten](#).

HTML Hilfe generieren

Um eine Hilfe beziehungsweise ein HTML-Projekt aus unserem Demo-Dokument zu generieren, muß die [XML](#)-Datei im [ResultViewer](#) geöffnet sein. Ist die Datei geöffnet, wird der [Dialog für die Einstellungen](#) aufgerufen. Wir wählen die Option „Hilfe anzeigen“ und als Zielverzeichnis das selbe, in dem das Word-Dokument abgespeichert ist. Danach klicken wir auf den Button „Generieren“. Nach dem erfolgreich abgeschlossenen Kompilieren wird die erzeugte Hilfe automatisch geöffnet.

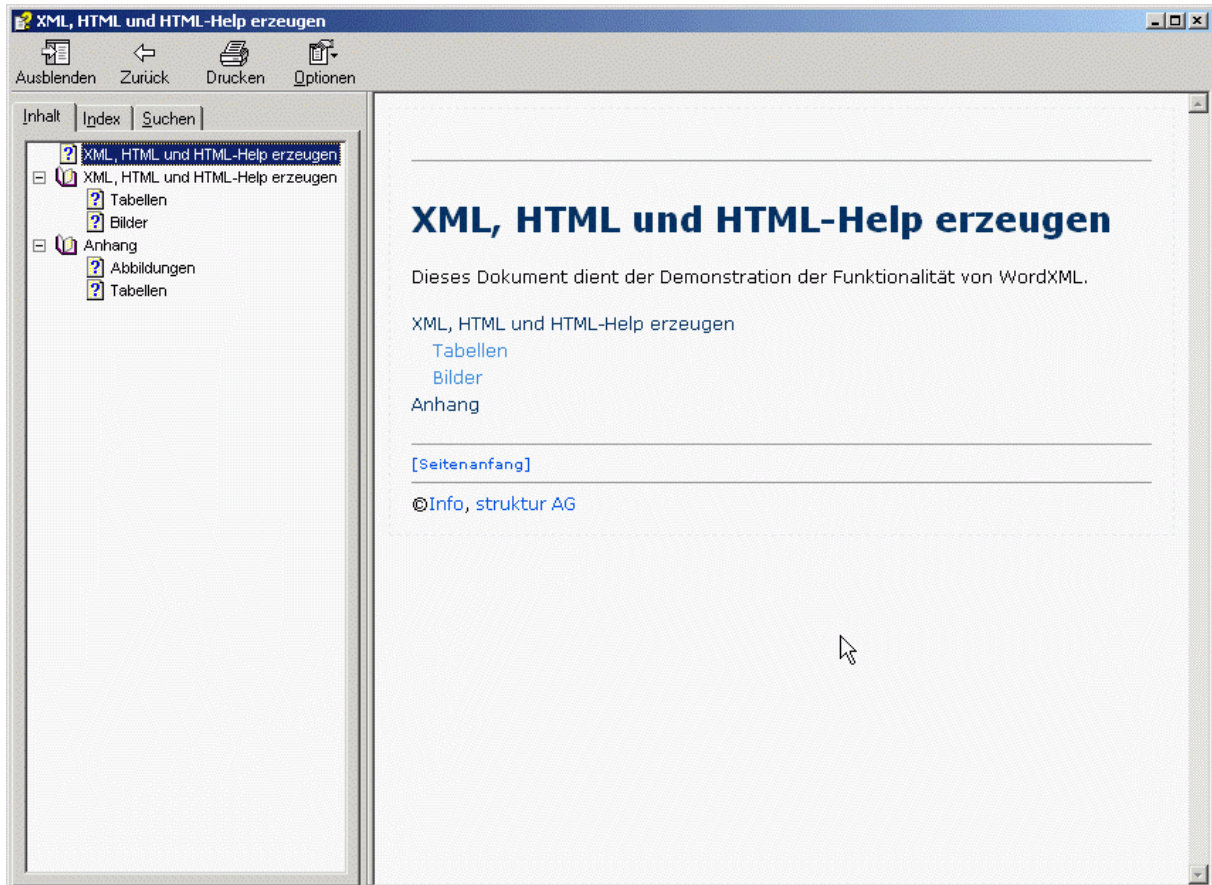


Figure 19 Aus dem Demo-Dokument generierte Hilfe

Glossar

CSS

Mit Hilfe von Cascading StyleSheets (CSS) können sogenannte Styleklassen definiert werden. Styleklassen enthalten Attribute wie zum Beispiel Fontarten und –größen, Ausrichtungen, Textstile u. ä.. Die CSS-Datei kann von einer oder mehreren [HTML](#)-Dateien referenziert werden. Den darin befindlichen [HTML](#)-Tags können Styleklassen zugewiesen werden. Somit läßt sich das Erscheinungsbild einzelner Tags oder gesamter [HTML](#)-Dateien steuern.

DTD

Die *Document Type Definition* (DTD) ist eine Grammatik zur formalen Beschreibung von [XML](#)-Dokumenten. Sie spezifiziert die Menge der Tags, die in einem XML-Dokument erlaubt sind und wie diese Tags zueinander in Beziehung stehen. Ein [XML](#)-Dokument ist **wohlgeformt**, wenn die Verschachtelung der Tags einen korrekten Klammerbegriff mit einer alles umschließenden Klammer bilden. Entspricht das Dokument zusätzlich einer DTD, so nennt man es **gültig** (valid). Weitere Informationen finden sich auf der Internetseite des World Wide Web Consortium [W3C](#).

Encoding

Die Kodierung der Zeichen einer Datei. Bei der Ausgabe beispielsweise als [HTML](#)-Datei bestimmt dies den darunterliegenden Zeichensatz, ebenso bei [TEXT](#). Bei [TEXT](#) kann die Architektur der Zielmaschine entscheidend für die korrekte Darstellung sein.

HTML

HTML steht für Hypertext Markup Language. Bei HTML handelt es sich um eine Auszeichnungssprache, mit deren Hilfe die Struktur eines Dokuments im Dokument selbst durch sogenannte Tags beschrieben werden kann. Der HTML-Standard spezifiziert die Menge aller HTML-Tags. Diese Menge ist fest.

Internetseiten sind in der Hauptsache mit HTML erstellt. Die Tags werden von Internetbrowsern ihrer Bedeutung entsprechend interpretiert und der betroffene Dokumententeil entsprechend angezeigt. HTML ist vom World Wide Web Consortium standardisiert, das eigens eine [HTML Homepage](#) unterhält.

TEXT

Im Kontext dieses Dokuments steht TEXT für die Ausgabemethode gleichen Namens bei [XSL](#)-Transformationen. TEXT erlaubt als drittes Ausgabeformat neben [XML](#) und [HTML](#) die Ausgabe beliebiger anderer textbasierter Daten wie etwa komma-separierte Dateien, Skripte oder selbstdefinierte Datenaustauschformate.

URI

Mit einem URI (Uniform Resource Identifier) werden Inhalte im Internet gekennzeichnet. Mit dem Begriff Inhalte sind in der Regel Dateien aller möglichen Formate gemeint, also Text, [HTML](#), [XML](#), Video, Sound und vieles mehr. Die häufigste Form einer URI ist eine [URL](#). Ein typischer URI spezifiziert

- Den Zugriffsmechanismus auf den Inhalt (z. B. ein Protokoll wie http, ftp oder file)
- Den Rechner, auf dem der Inhalt zu finden ist
- den spezifischen Namen des Inhalts auf diesem Computer (typischerweise ein Dateiname)

Die Teile sind optional, weshalb ein Dateiname für sich (auch ein relativer) ein URI ist.

URL

Eine URL ist die Adresse einer Datei, auf die über das Internet zugegriffen werden kann. Der Art der Datei wird über das Zugriffsprotokoll bestimmt (nicht der Dateityp!). Beispielsweise werden vom HTTP-Protokoll [HTML](#)-Seiten, Java-Applets, CGI-Skripte usw. unterstützt. Eine [URL](#) besteht aus

- Dem Zugriffsprotokoll
- Einem Rechnernamen (der Domäne)
- Einem spezifischen Dateinamen

WXML

WXML steht für „WordXML [XML](#)“. Hierbei handelt es sich um das festgelegt [Ausgabeformat](#) von WordXML in Form einer wohlgeformten [XML](#)-Datei. Die Datei enthält eine Repräsentation des Word-

Dokumentinhalte zusammen mit Struktur und Layoutinformationen. Das Format ist sehr detailliert und wird in der Regel vor einer Weiterverarbeitung noch einmal gefiltert.

Die Datei ist [UTF-8 kodiert](#). Die [Wohlgeformtheit](#) des Dokuments ist garantiert. Die Dateien werden maschinell generiert und dürfen auf keinen Fall editiert werden. Eine Veränderung der Dokumente kann zu undefinierten Ergebnissen führen. Änderungen des wxml-Formats obliegen im Zuge der Produktpflege einzig und allein der [struktur AG](#).

XHTML

Ebenfalls vom World Wide Web Consortium ([W3C](#)) festgelegt, ist [XHTML](#) eine Neuformulierung des [HTML](#) 4.0 Standards als XML-Anwendung. Tatsächlich ist XHTML die Folgeversion von HTML. Im Gegensatz zu [HTML](#) kann XHTML aber von jedermann erweitert werden. Außerdem gelten die Regeln für die [Wohlgeformtheit](#) eines [XML](#)-Dokuments

XML

XML (eXtensible Markup Language) ist eine Datenbeschreibungssprache, die ähnlich [HTML](#) die Strukturierung von Daten mit Hilfe von Tags unterstützt. Im Gegensatz zu [HTML](#) können in XML beliebig viele neue Tags eingeführt werden. Jeder kann die Menge der Tags, die er verwenden will, selbst spezifizieren. Der [XML-Standard](#) wurde vom World Wide Web Consortium verfaßt.

XSLT

XSLT steht für „eXtensible Stylesheet Language Transformations“. Ein Stylesheet enthält Anweisungen, wie Tags aus XML-Dokumenten verarbeitet werden. Eine Art der Verarbeitung ist die Ausgabe eines [HTML](#)-Tags für ein [XML](#)-Tag. So können [XML](#)-Dokumente vollständig nach [HTML](#) umgewandelt werden. Als Ausgabeformate stehen [HTML](#), [XHTML](#), [XML](#) und [TEXT](#) zur Verfügung. Das XSLT-Stylesheet legt dann die Regeln für die Umwandlung fest. Die Verarbeitung wird von sogenannten XSL-Prozessoren vorgenommen. XSLT ist ebenfalls vom World Wide Web Consortium standardisiert. Mehr Informationen hierzu gibt es auf der [XSL-Seite](#) des [W3C](#).