

Apache Subversion

aus Wikipedia, der freien Enzyklopädie

Apache Subversion (SVN) ist eine freie Software zur Versionsverwaltung von Dateien und Verzeichnissen.

Die Versionierung erfolgt in einem zentralen Projektarchiv (engl. *repository*) in Form einer einfachen Revisionszählung. Wenn Änderungen an Inhalten verteilt auf den Computern der Bearbeiter ausgeführt werden, werden zwischen dem Projektarchiv und einem Arbeitsplatz jeweils nur die Unterschiede zu bereits vorhandenen Ständen übertragen.

Inhaltsverzeichnis

- 1 Allgemeines
- 2 Geschichte
- 3 Unterschiede zu CVS
 - 3.1 Kopien
 - 3.2 Tag- und Branchkonzept
 - 3.3 Sonstiges
- 4 Abhängigkeiten von Subversion
- 5 Distributionen
- 6 Grafische Benutzeroberflächen
- 7 Literatur
- 8 Einzelnachweise
- 9 Weblinks

Allgemeines

Subversion wird als freie Software unter einer Lizenz im Stil der Apache-Lizenz veröffentlicht.

Die Benennung „Subversion“ setzt sich aus den Worten *Sub* und *Version* zusammen und kann so als *Unterversion*, *frühere Version* verstanden werden. Eine andere Interpretation führt den Namen auf den politisch-soziologischen Begriff *Subversion* (deutsch: „Umsturz“, „Zersetzung“) zurück. Allerdings ist von den Entwicklern nicht dokumentiert worden, ob dieses Wortspiel beabsichtigt ist.

Subversion versteht sich als Weiterentwicklung von CVS und entstand als Reaktion auf weit verbreitete Kritik an CVS. In der Bedienung der Kommandozeilenversion ist es sehr ähnlich gehalten.

Mit Subversion ist es aber – im Gegensatz zu CVS – z. B. möglich, Dateien oder Verzeichnisse zu verschieben oder umzubenennen, ohne die Versionsgeschichte zu verlieren. Details finden sich im Abschnitt Unterschiede zu CVS.

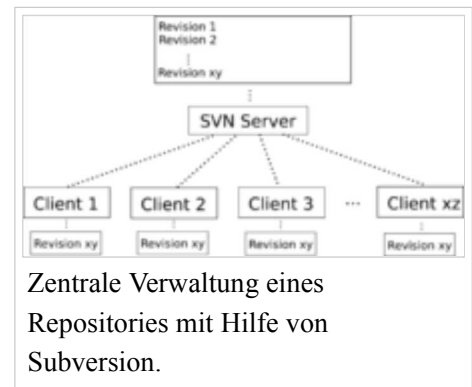
Mit cvs2svn gibt es einen Konverter, mit dem ein CVS-Repository zu Subversion konvertiert werden kann.

Apache Subversion



| | |
|---------------------------|--|
| Entwickler | CollabNet |
| Aktuelle Version | 1.7.4 (8. März 2012) |
| Betriebssystem | AIX, GNU/Linux, Windows, Mac OS X, *BSD, Solaris, OS/400 |
| Programmiersprache | C |
| Kategorie | Versionsverwaltung |
| Lizenz | Apache-Lizenz 2.0 |
| Deutschsprachig | Nach Installation eines Sprachpaketes |

subversion.apache.org
(<http://subversion.apache.org/>)



Auch für die Migration von anderen Versionsverwaltungs-Systemen (etwa PVCS, Visual Source Safe, ClearCase, MKS, Perforce, StarTeam, ...) sind verschiedene kostenfreie Import-Werkzeuge erhältlich.

Geschichte

Subversion wurde seit Anfang 2000 bei CollabNet entwickelt und erreichte am 23. Februar 2004 die stabile Version 1.0. Am 29. September 2004 erschien Version 1.1, deren größte Neuerung war, dass Projektarchive (Repositories) nicht mehr nur in einer Berkeley-Datenbank verwaltet werden können, sondern dass dazu auch direkt das Dateisystem benutzt werden kann. Außerdem wurden internationalisierte Programmausgaben ermöglicht. Die am 23. Mai 2005 erschienene Version 1.2 unterstützt nun auch Bearbeitungssperren für Dateien, was für binäre Dateien von Vorteil sein kann. Seit dem 1. Januar 2006 gibt es die Version 1.3, die in den Bereichen Server-Logging, Autorisierung, Programmiersprachen-Anbindungen, Kommando-Optionen und Leistung („Performance“) Verbesserungen bietet. Am 10. September 2006 wurde Version 1.4 veröffentlicht, sie bringt das Programm *svn* mit, welches das Spiegeln von Projektarchiven (Repositories) ermöglicht.

Die am 19. Juni 2008 veröffentlichte Version 1.5 erweiterte Subversion um das sogenannte *Merge-Tracking*, das dokumentierte Zusammenführen verschiedener Teilbäume. Version 1.6 wurde am 20. März 2009 veröffentlicht und erweitert das Merge-Tracking sowie Sicherheitsfunktionen. 2011 erschien Version 1.7 welche die Metadaten auf dem Clientcomputer zentral ablegt und nicht mehr wie bisher in jedem Projektverzeichnis.

Am 4. November 2009 gaben ehemaliger und neuer Hoster in einer gemeinsamen Presseerklärung bekannt, dass das Projekt auf Entschluss von CollabNet zur Apache Software Foundation wechseln werde.^[1] Am 7. November 2009 startete Apache Subversion im Apache Incubator, seit 17. Februar 2010 ist Apache Subversion ein Top-Level-Apache-Projekt.

Unterschiede zu CVS

CVS basiert auf dem älteren RCS-System und hat hiervon das interne Speicherformat übernommen. Dieses Format ist ausschließlich für die Verwaltung der Versionsgeschichte von einzelnen Dateien ausgelegt. CVS hat für die Verwaltung von Projekten mit mehreren Dateien eine sehr einfache Lösung gewählt: Das Projekt wird als Sammlung von Dateien betrachtet, von denen jede für sich eine eigene Historie hat. Insbesondere in der Softwareentwicklung gehören aber häufig Änderungen an mehreren Dateien in einem logischen Schritt zusammen. So muss beispielsweise in einem C-Projekt bei Anpassung einer Funktionsdeklaration in *foo.h* immer auch die Implementierung in *foo.c* angepasst werden. Dass der Schritt von *foo.h* auf die Version 1.5 mit dem Schritt von *foo.c* auf die Version 1.9 zusammenhängt, kann in CVS nicht abgebildet werden. Es kann nur durch weiche Kriterien wie zeitliche Nähe oder gleiche Commit-Nachricht gefolgert werden.

Das Versionsschema von Subversion bezieht sich aus diesem Grund nicht auf einzelne Dateien, sondern jeweils auf das ganze Projektarchiv. Dieses bekommt mit jeder Änderung eine neue „Revisionsnummer“ zugeordnet. Somit kann man einfacher – und im Gegensatz zu CVS konsistent – eine exakte Version beschreiben (z. B. „Revision 2841“ statt „Version vom 23. März 2004 20:56:31 UTC“). Die Revisionsnummer einer Datei entspricht dabei der Revisionsnummer des Projektarchivs, als sie das letzte Mal geändert wurde, die Revisionsnummer eines Verzeichnisses entspricht der höchsten Revisionsnummer der enthaltenen Dateien und Verzeichnisse. Die Abfolge der Revisionsnummern einer einzelnen Datei kann also durchaus lückenhaft sein, wenn die Datei nicht bei jeder Änderung (commit) am Repository geändert wurde. Beispielsweise könnte eine Datei bei der Revision 25 zum Projektarchiv hinzugefügt und jeweils einmal in der Revision 48 und der Revision 52 verändert worden sein. Beim Abrufen (*checkout*) einer Datei wird die größte Revisionsnummer abgerufen, die kleiner oder gleich der angeforderten ist. Wird in dem Beispiel die Revision 52 angefordert, so wird die Revision 52 der Datei abgerufen; wird hingegen die Revision 51 angefordert, liefert Subversion die Inhalte von Revision 48.

Subversion speichert Client-seitig beim Erstellen einer lokalen Kopie (checkout), beim Aktualisieren (update) und Übertragen (commit) in einem gesonderten Verzeichnis (`.svn`) pro Ordner der Arbeitskopie eine zweite Kopie jeder Datei. Dadurch verdoppelt sich der Speicherbedarf einer Arbeitskopie, allerdings bietet dies bei entfernten Projektarchiven auch einige Vorteile. So können einige Aktionen, wie Anzeige der lokalen Änderungen, ganz ohne Netzwerkzugriff erfolgen, und Subversion muss beim Übertragen nur die geänderten Teile einer Datei übertragen, während CVS die Änderungen Server-seitig berechnet und somit jeweils die gesamte Datei übertragen muss. Auch ist es möglich, jederzeit die Änderungen einer Datei gegenüber ihrer Basisversion zu ermitteln oder zurückzunehmen, ohne das Projektarchiv zu konsultieren.

Da die in den `.svn`-Verzeichnissen enthaltenen Metadaten etwa auf einem Produktivsystem nicht erwünscht sind, gibt es in Subversion das CVS nicht bekannte *export*-Kommando. Hierdurch wird eine Arbeitskopie ohne Metadaten erzeugt. Der Subversion-Client kann auf einer so erzeugten Arbeitskopie nicht arbeiten.

Die Übertragung der lokalen Änderungen (commits) geschieht dabei in Subversion atomar, das heißt, eine Änderung – auch mehrerer Dateien – wird entweder ganz oder gar nicht ins Repository gespeichert. Verbindungsabbrüche und mehrere gleichzeitige Zugriffe können somit nicht zu inkonsistenten Zuständen führen.

Mit Version 1.7.0 wurde das Metadaten-Konzept komplett geändert. Statt der `.svn`-Ordner werden die Metadaten jetzt in einer SQLite-Datenbank gespeichert.

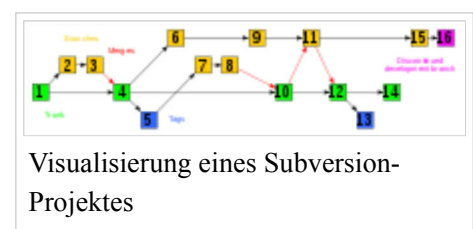
Kopien

Subversion ist in der Lage, Kopien von Dateien so zu verwalten, dass den Kopien derselbe Bearbeitungsverlauf gemeinsam ist. Dieser Umstand ist besonders dann nützlich, wenn Änderungen an einer der Dateien in die andere portiert werden sollen. Zur Erstellung einer Kopie werden die Dateien auch nicht dupliziert, sondern es wird eine datenbankinterne Verknüpfung angelegt, die im weiteren Verlauf genauso weiterverwendet werden kann wie das Original. Von dieser Technik wird häufig als „billige Kopie“^[2] gesprochen. Kopien sind insbesondere bei Umbenennung und Verschiebung von Dateien interessant. Subversion realisiert dies zwar wie CVS, indem es eine Kopie anlegt und das Original als gelöscht markiert, allerdings kommt es nicht wie in CVS zu einem Bruch im Versionsverlauf. Eine native Unterstützung für Verschiebung und Umbenennung ist auf der Entwicklerseite als mittelfristiges Ziel genannt.^[3]

Tag- und Branchkonzept

Neben dem geänderten Datenbank-Modell sticht das zu anderen Versionsverwaltungssystem völlig unterschiedliche Konzept im Bereich der Benennung von Dateiversionen (sogenanntes *tagging*) und der Verzweigung in Varianten (*branching*) hervor. Während die Benennung von tags und branches in den meisten anderen Versionsverwaltungssystemen wie git, Mercurial oder auch CVS eine klare semantische Bedeutung haben, kennt Subversion nur das

Konzept der „billigen Kopie“, die je nach Nutzungsart „Tag-“ oder „Branch-Charakter“ haben kann. Jede Kopie in Subversion ist demnach automatisch eine Variante dieser Datei oder des Verzeichnisses. Namentliche Kennzeichnungen entstehen in Subversion durch eine Kopie, die später nicht mehr verändert werden sollte. Wegen des Fehlens einer Tag- oder Branch-Semantik obliegt die Strukturierung und Verwaltung von Kennzeichnungen und Verzweigungen dem Benutzer und Administrator. Dabei hat sich bewährt, für Projekte die Basisverzeichnisse *trunk* (engl. „Stamm“), *branches* (engl. „Verzweigungen“) und *tags* (engl. „Markierungen“) anzulegen. *trunk* enthält dabei die Hauptentwicklungslinie des Projekts, in *branches* werden weitere Unterverzeichnisse mit alternativen Entwicklungspfaden verwaltet, und in *tags* eine Kopie von *trunk* oder einem der *branches* als Unterverzeichnis angelegt. Zur besseren Übersicht werden je nach Projektanforderungen *tags* und *branches* noch in weitere Unterverzeichnisse unterteilt. Als *head* bezeichnet man die neueste Version („top revision“) innerhalb einer Verzweigung.



Da *branches* und *tags* nur Kopien im Dateisystem sind, entsteht bei deren Erstellung zwangsläufig ein neuer Commit. Zudem muss sich jeder Entwickler an die vorgegebene Semantik halten. So ist es etwa möglich, *tags* und *branches* von Teilen des Dateisystems oder gar vom ganzen Dateisystem inklusive aller bisherigen *tags* und *branches* zu erstellen, in einen *tag* nachträglich Änderungen zu *committen* oder Dateien oder Verzeichnisse außerhalb der vorgesehenen Dateistruktur anzulegen. So ein regelwidriges Verhalten kann durch serverseitige Skripte unterbunden werden.

Da Dateien in Subversion auch versionskontrolliert umbenannt werden können, kann die Projektstruktur jederzeit gestiegenen oder gesunkenen Anforderungen angepasst werden.

Sonstiges

Subversion kann im Gegensatz zu CVS und anderen Versionskontrollsystemen wie SCCS, RCS oder Git auch Verzeichnisse und Metadaten verwalten. Insbesondere können Verzeichnisse auch als gelöscht markiert werden. Dies ist mit CVS nicht möglich, hier können nur leere Verzeichnisse gelöscht werden, sie können nicht ohne Verlust der Historie aller enthaltenen Dateien aus dem Repository gelöscht werden. Subversion bietet einen verbesserten Umgang mit Binärdaten. Es erkennt solche Dateien (beispielsweise Bilder oder Audiodateien) weitgehend automatisch, und es werden (wie bei Textdateien) nur die Differenzen zwischen den geänderten Versionen gespeichert. In CVS geht das umständlicher über Eintrag der Endungen von binären Dateien in `cvswrapper`, verschiedene Versionen von Dateien dieser Typen müssen aber immer vollständig gespeichert werden.

Die Metadaten erlauben die Speicherung von Attributen wie etwa Dateien, die ignoriert werden sollen, oder solche, die auf Systemen, die dies unterstützen, als ausführbar markiert werden sollen. Neben den von SVN vordefinierten Werten, zu denen etwa auch Merge-Informationen gehören, können hier auch beliebige andere Werte hinterlegt werden. Allerdings werden diese Metadaten teilweise als intransparent kritisiert, da sie je nach Client gut versteckt sind und teils von Subversion selbst gesetzt werden.

Wie CVS bietet Subversion den Netzwerkzugriff über einen eigenen Server, auf den mit SSH auch verschlüsselt zugegriffen werden kann. Zusätzlich hierzu und der Speicherung im lokalen Dateisystem gibt es auch ein Modul für den Apache-2-Webserver, mit dem die Daten auch mit der HTTP/HTTPS-Erweiterung WebDAV übertragen werden können. Somit kann die aktuelle Revision einer Datei auch mit einem gewöhnlichen Webbrowser abgerufen werden.

Einmal eingefügte (*checked in*) Dateien können als gelöscht markiert werden, sind dann aber immer noch mit ihrer gesamten Historie vorhanden. Ein vollständiges (mitsamt Versionshistorie) Entfernen von Dateien entspricht nicht den Grundsätzen einer Versionsverwaltung, lässt sich aber in Subversion als Administrator und nur mit großem Aufwand durchführen.

Subversion verwaltet das gesamte Repository in einer Datenbank, deren Dateien nicht die Struktur des Repository-Inhalts widerspiegeln. Die Integrität der Datenbank lässt sich so verzeichnisübergreifend überprüfen. Es stehen dabei aktuell zwei Backends zur Verfügung. Das in der Version 1.1 hinzugefügte `fsfs`-Backend verwendet ein eigenes Format. Das andere Backend verwendet das Berkeley-Datenbanksystem, dies hat jedoch den Nachteil, dass die Daten einerseits – abhängig von der verwendeten Version – dessen binären Inkompatibilitäten unterliegen, und andererseits den damit eingebrachten Stabilitätsproblemen. Dies kann zur Folge haben, dass ein mit einer älteren Version erstelltes Repository an die neue Version angepasst werden muss. Zudem ist der Zugriff über NFS und Windows-Netzwerkfreigaben dann nicht möglich.

Im Gegensatz zu CVS definiert Subversion die Zeichenkodierung, welche für Dateinamen und Log-Messages im Repository benutzt wird. Damit können beispielsweise auch Dateien mit Umlauten im Namen auf Systemen mit verschiedenen Zeichen-Codierungen (beispielsweise CP1252 (deutschsprachiges Windows), UTF-8 (Linux)) benutzt werden, während dies bei CVS nicht plattformunabhängig funktioniert, da dies im CVS-Client-Server-Protokoll nicht definiert ist. Allerdings ergeben sich bei Zeichen in

Dateinamen, die in UTF-8 sowohl eine composed als auch decomposed gespeichert werden können, Probleme auf MacOS-basierten Systemen. Dies betrifft etwa Umlaute, die in UTF-8 als ein Zeichen (Ä, composed) und als zwei Zeichen (A und ¨, decomposed) gespeichert werden können.^[4]

Abhängigkeiten von Subversion

Für eine Installation der Basisfunktionen muss seit Version 1.1.0 nur die *Apache Portable Runtime*-Bibliothek vorhanden sein. Zuvor war auch noch eine Berkeley-DB in einer Version 4.0 oder höher notwendig, was aber hinfällig ist, seit das Repository mit Hilfe des FSFS-Backend optional auch direkt im Dateisystem gespeichert werden kann.

Apache 2 und Neon sind für die WebDAV-Nutzung erforderlich, Python 2.x für einige mitgelieferte Test-Skripte, eine SSL-Implementierung, wenn man WebDAV verschlüsseln will. Seit Version 1.4 kann alternativ auch Serf anstatt neon für WebDAV verwendet werden.

Repositorys werden – wie bei CVS – mit dem Aufruf eines Befehls eingerichtet. Damit kann bei lokalem Zugriff sofort gearbeitet werden. Eine Server-Konfiguration verläuft abhängig von der gewählten Methode unterschiedlich, gleicht aber derjenigen anderer Systeme, CVS eingeschlossen.

Mit dem WebDAV-Modul wird der Apache-HTTP-Server auch direkt zum eigentlichen Serverprozess für Subversion-Repositories. Die Subversion-Inhalte werden dabei über eine gewöhnliche URL per HTTP adressiert. Mittels eines normalen Web-Browsers kann ein Repository durchsucht werden (zumindest die aktuelle Version). Da Subversion damit auch den normalen HTTP-Port 80 verwendet, ist oft keine zusätzliche Konfiguration von Firewall o.ä. notwendig.

Subversion kann auch mit einem eigenen Apache-unabhängigen Serverprozess verwendet werden. Dieser verwendet ein eigenes Netzwerkprotokoll, das – weil für Subversion optimiert – effizienter ist als das Apache-Modul. Der Standard-Port ist 3690.

Distributionen

Eine größere Anzahl von Freiwilligen pflegt verschiedene Binärdistributionen von Subversion, in der Regel mehrere pro Zielsystem. Ebenfalls werden Binärdistributionen zu unterschiedlichen Konditionen von einschlägigen Unternehmen herausgegeben. Das Apache-Subversion-Projekt gibt keine offiziellen Empfehlungen für bestimmte Distributionen heraus, pflegt aber eine Seite mit Weblinks der ihm bekannten Distributionen.^[5]

Grafische Benutzeroberflächen

Es gibt einige ausgereifte grafische Benutzeroberflächen (GUIs) für Subversion. Sie machen es den Benutzern besonders leicht, auf ein Subversion-Repository zuzugreifen. Hier einige Anwendungen:

- Cornerstone: Mac OS X, proprietär
- KDEsvn: Linux (KDE), Open-Source
- RapidSVN: Linux (Gnome), Windows, Open-Source
- SmartSVN: Mac OS X, Windows, Linux, OS/2, Freeware
- Subcommander: Mac OS X, Windows & Linux, Open-Source
- svnX: Mac OS X, Open-Source
- Versions: Mac OS X, proprietär
- ZigVersion: Mac OS X, Freeware
- TkCVS: Unix/Linux, Windows, und MacOS X

Integriert in Dateibrowser:

- RabbitVCS: Linux (Gnome)
- SCPlugin: Mac OS X (bis Version 10.5), Open-Source
- SmartSVN: Mac OS X, Windows, Freeware
- TortoiseSVN: Windows, Open-Source

Integriert in Entwicklungsumgebungen:

- AnkhSVN und VisualSVN für Microsoft Visual Studio
- Dreamweaver ab Version CS4 von Adobe Systems
- Oracle Data Modeler von Oracle
- Subclipse und Subversive für Eclipse
- Xcode von Apple
- IntelliJ IDEA von JetBrains Inc.
- Netbeans von Oracle Corporation
- Delphi XE/XE2 von Embarcadero Technologies

Integriert in Editoren:

- BBEdit: Mac OS X
- oXygen (Editor): Mac OS X, Windows, Linux
- TextMate: Mac OS X
- CODA Webeditor ab Version 1.5 von Panic Inc.: Mac OS X

Weiterhin sind Plugins für NetBeans, KDevelop, Code::Blocks, Vim, TYPO3 und ASCET verfügbar. Die globale Administration (Benutzerrechte, Protokolle, ...) erfolgt jedoch weiterhin über spezielle SVN-eigene Konfigurationsdateien.

Um ein Subversion-Repository lediglich zu betrachten, bieten viele Open-Source-Projekte einen Link auf ihren Webdienst an. Dieser präsentiert in übersichtlicher Form Inhalte von Dateien, Verzeichnissen und Logbüchern; auch Datei-Vergleiche sind möglich.

Literatur

- C. Michael Pilato, Ben Collins-Sussman, Brian W. Fitzpatrick: *Versionskontrolle mit Subversion* (<http://svnbook.red-bean.com/>) – Website des freien Buchprojektes.
- C. Michael Pilato, Ben Collins-Sussman, Brian W. Fitzpatrick, *Versionskontrolle mit Subversion*, O'Reilly-Verlag, 3. Auflage April 2009, ISBN 978-3-89721-897-0.
- Tobias Wassermann: *Versionsmanagement mit Subversion*, mitp-Verlag, 1. Auflage Oktober 2006, ISBN 978-3-8266-1662-4.
- Frank Budzuhn: *Subversion*, Galileo Computing, 2. aktualisierte und erweiterte Auflage 2007, ISBN 978-3-89842-879-8.

Einzelnachweise

1. Subversion Submitted to Become a Project at The Apache Software Foundation (http://www.apache.org/foundation/press/pr_2009_11_04.html)
2. Erläuterung der Bubble-Up-Methode als Grundlage „billiger Kopien“ in SVN (englisch) (<http://svn.apache.org/repos/asf/subversion/trunk/notes/subversion-design.html#server.fs.struct.bubble-up>)
3. Subversion-Bugtracker: *Issue 898* (http://subversion.tigris.org/issues/show_bug.cgi?id=898)

4. Problems with umlauts in file names on Mac OS X (<http://www.syntevo.com/smartsvn/techarticles.html?page=problems.macos-special-characters>)
5. Apache-Linkliste für Binärdistributionen (<http://subversion.apache.org/packages.html>)

Weblinks

- Offizielle Website (<http://subversion.apache.org/>) (englisch)
- Apache-Linkliste für Binärdistributionen (<http://subversion.apache.org/packages.html>) (englisch)
- Version Control with Subversion (<http://svnbook.red-bean.com/>) – Website des Online-Buchprojekts
 - aktueller Entwicklungsstand des Buches in verschiedenen Ausgaben (<http://svnbook.red-bean.com/nightly/>) – in automatisierter Übersetzung („Nightly Build“)
- Links zum Thema Subversion (http://www.dmoz.org/Computers/Software/Configuration_Management/Tools/Subversion/) im Open Directory Project

Von „http://de.wikipedia.org/w/index.php?title=Apache_Subversion&oldid=101560351“

Kategorien: Apache-Projekt | Freie Versionsverwaltungssoftware

- Diese Seite wurde zuletzt am 1. April 2012 um 22:23 Uhr geändert.
- Der Text ist unter der Lizenz „Creative Commons Attribution/Share Alike“ verfügbar; zusätzliche Bedingungen können anwendbar sein. Einzelheiten sind in den Nutzungsbedingungen beschrieben. Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.