

Inhalt

Versionshinweis	19
Vorwort	21
Einleitung	23
Ruby ist ein Edelstein	24
Was für eine Programmiersprache ist Ruby?	25
Ist Ruby für mich geeignet?	26
Warum haben wir dieses Buch geschrieben?	26
Ruby-Versionen	27
Ruby installieren	27
Ruby ausführen	28
Ressourcen	29
Danksagungen	30
Konventionen bei der Notation	31
I Überblick	33
I Facetten von Ruby	35
2 Ruby.new	37
2.1 Ruby ist eine objektorientierte Sprache	37
2.2 Ruby-Grundlagen	39
2.3 Arrays und Hashes	42
2.4 Kontrollstrukturen	44
2.5 Reguläre Ausdrücke	45
2.6 Blöcke und Iteratoren	47
2.7 Lesen und Schreiben	49
2.8 Der nächste Schritt	50

3	Klassen, Objekte und Variablen	51
3.1	Vererbung und Nachrichten	53
3.1.1	Vererbung und Mixins	55
3.2	Objekte und Attribute	56
3.2.1	Schreibbare Attribute	57
3.2.2	Virtuelle Attribute	58
3.3	Klassenvariablen und Klassenmethoden	59
3.3.1	Klassenvariablen	59
3.3.2	Klassenmethoden	60
3.3.3	Singletons und andere Konstruktoren	61
3.4	Zugriffskontrolle	63
3.4.1	Die Zugriffskontrolle angeben	64
3.5	Variablen	66
4	Container, Blöcke und Iteratoren	69
4.1	Container	69
4.1.1	Arrays	69
4.1.2	Hashes	72
4.1.3	Den Container SongList implementieren	72
4.2	Blöcke und Iteratoren	75
4.2.1	Iteratoren implementieren	76
4.2.2	Ruby im Vergleich zu C++ und Java	78
4.2.3	Blöcke für Transaktionen	80
4.2.4	Blöcke können abgeschlossene Einheiten sein	82
5	Standardtypen	85
5.1	Zahlen	85
5.2	Zeichenketten	87
5.2.1	Mit Zeichenketten arbeiten	89
5.3	Wertebereiche	93
5.3.1	Wertebereiche als Folgen	93
5.3.2	Wertebereiche als Bedingungen	94
5.3.3	Wertebereiche als Intervalle	95
5.4	Reguläre Ausdrücke	95
5.4.1	Muster	96
5.4.2	Musterbasierte Substitution	101
5.4.3	Objektorientierte reguläre Ausdrücke	104
6	Mehr über Methoden	107
6.1	Eine Methode definieren	107
6.1.1	Argumentlisten mit variabler Länge	108
6.1.2	Methoden und Blöcke	108
6.2	Eine Methode aufrufen	109
6.2.1	Arrays in Methodenaufrufen erweitern	110
6.2.2	Blöcke dynamischer gestalten	110
6.2.3	Hash-Argumente sammeln	112

7	Ausdrücke	115
7.1	Operatorausdrücke	116
7.2	Verschiedene Ausdrücke	117
7.2.1	Befehlserweiterung	117
7.2.2	Rückwärtsgerichtete Anführungszeichen sind weich	117
7.3	Zuweisung	118
7.3.1	Parallele Zuweisung	120
7.3.2	Andere Zuweisungsarten	121
7.4	Bedingte Ausführung	121
7.4.1	Boolesche Ausdrücke	122
7.4.2	Defined?, And, Or und Not	122
7.4.3	If- und Unless-Ausdrücke	124
7.5	Case-Ausdrücke	126
7.6	Schleifen	127
7.6.1	Iteratoren	129
7.6.2	For ... In	130
7.6.3	Break, Redo und Next	131
7.6.4	Retry	132
7.7	Variablengültigkeitsbereich und Schleifen	133
8	Ausnahmen, Catch und Throw	135
8.1	Die Klasse Exception	135
8.2	Ausnahmen behandeln	136
8.2.1	Aufräumen	138
8.2.2	Und noch einmal von vorn	139
8.3	Ausnahmen auslösen	140
8.3.1	Informationen zu Ausnahmen hinzufügen	141
8.4	Catch und Throw	141
9	Module	143
9.1	Namensräume	143
9.2	Mixins	144
9.2.1	Instanzvariablen in Mixins	147
9.3	Iteratoren und das Modul Enumerable	149
9.4	Andere Dateien einfügen	149
10	Einfache Ein- und Ausgabe	151
10.1	Was ist ein IO-Objekt?	151
10.2	Dateien öffnen und schließen	152
10.3	Dateien lesen und schreiben	152
10.3.1	Iteratoren für das Lesen	153
10.3.2	In Dateien schreiben	154
10.3.3	Aber ich vermisse meinen IO-Stream aus C++	155
10.4	Mit Netzwerken kommunizieren	155

11	Threads und Prozesse	157
11.1	Mit mehreren Threads arbeiten	157
11.1.1	Ruby-Threads erstellen	157
11.1.2	Threads und Ausnahmen	160
11.2	Den Thread-Zeitplaner steuern	161
11.3	Gegenseitiger Ausschluss	162
11.3.1	Die Klasse Mutex	162
11.3.2	Bedingungsvariablen	164
11.4	Mehrere Prozesse ausführen	165
11.4.1	Neue Prozesse erzeugen	165
11.4.2	Unabhängige Kinder	167
11.4.3	Blöcke und Teilprozesse	168
12	Fehlerbehebung	169
12.1	Rubys Debugger	169
12.2	Interaktives Ruby	171
12.2.1	Unterstützung von Editoren	173
12.3	Aber es funktioniert nicht!	174
12.4	Aber es ist zu langsam!	175
12.4.1	Lokale Variablen außerhalb von Blöcken erstellen	175
12.4.2	Verwenden Sie den Profiler	176
II	Ruby im Kontext	179
13	Ruby und seine Welt	181
13.1	Kommandozeilenargumente	181
13.1.1	Kommandozeilenoptionen	182
13.1.2	ARGV	184
13.2	Programmbeendigung	184
13.3	Umgebungsvariablen	185
13.3.1	In Umgebungsvariablen schreiben	185
13.4	Wo Ruby seine Module findet	186
13.5	Erstellte Umgebung	187
14	Ruby und das Web	189
14.1	CGI-Skripte schreiben	189
14.1.1	cgi.rb verwenden	189
14.1.2	Anführungszeichen	189
14.1.3	Formulare	190
14.1.4	Cookies	192
14.1.5	Sitzungen	193
14.2	Ruby in HTML einbetten	194
14.2.1	eruby verwenden	194
14.2.2	eruby auf Apache installieren	196
14.3	Die Leistung verbessern	196

15	Ruby Tk	199
15.1	Einfache Tk-Anwendung	199
15.2	Widgets	200
15.2.1	Widget-Optionen definieren	200
15.2.2	Widget-Daten holen	201
15.2.3	Optionen dynamisch setzen und holen	202
15.2.4	Beispielanwendung	202
15.3	Bindungsereignisse	203
15.4	Zeichenfläche	205
15.5	Scrolling	207
15.5.1	Eine Sache noch	207
15.6	Von der Perl/Tk-Dokumentation übersetzen	209
15.6.1	Objekterzeugung	210
15.6.2	Optionen	210
15.6.3	Variablenreferenzen	210
16	Ruby und Microsoft Windows	211
16.1	Ruby-Ports	211
16.2	Ruby unter Windows ausführen	211
16.3	Win32API	212
16.4	Windows-Automation	212
16.4.1	Eigenschaften holen und setzen	213
16.4.2	Benannte Argumente	213
16.4.3	for each	214
16.4.4	Ein Beispiel	214
16.4.5	Optimierung	215
17	Ruby erweitern	217
17.1	Ruby-Objekte in C	217
17.1.1	VALUE als Zeiger	218
17.1.2	VALUE als direktes Objekt	219
17.2	Ruby in C schreiben	219
17.2.1	Ruby-Ausdrücke in C auswerten	222
17.3	Daten mit Ruby und C gemeinsam nutzen	222
17.3.1	Variablen direkt gemeinsam nutzen	223
17.3.2	C-Strukturen einhüllen	223
17.3.3	API: C-Datentyp-Wrapping	224
17.3.4	Ein Beispiel	228
17.4	Speicherzuweisung	230
17.4.1	API: Speicherzuweisung	230
17.5	Eine Erweiterung erzeugen	230
17.5.1	Eine Makefile mit extconf.rb erzeugen	231
17.5.2	Statische Verknüpfung	234
17.6	Einen Ruby-Interpreter einbetten	235
17.6.1	API: Eingebettete Ruby-API	235
17.7	Ruby mit anderen Sprachen verbinden	236

17.8	Die C-API von Ruby	236
17.8.1	API: Objekte definieren	237
17.8.2	API: Methoden definieren	237
17.8.3	API: Variablen und Konstanten definieren	239
17.8.4	Methoden aufrufen	240
17.8.5	Ausnahmen	241
17.8.6	Iteratoren	242
17.8.7	Auf Variablen zugreifen	243
17.8.8	Objektstatus	243
17.8.9	Häufig verwendete Methoden	244
III	Ruby im Detail	247
18	Die Programmiersprache Ruby	249
18.1	Quell-Layout	249
18.1.1	BEGIN- und END-Blöcke	250
18.2	Die Grundtypen	251
18.2.1	Ganze Zahlen und Gleitkommazahlen	251
18.2.2	Wertebereiche	254
18.2.3	Arrays	255
18.2.4	Hashes	255
18.2.5	Symbole	256
18.2.6	Reguläre Ausdrücke	256
18.3	Namen	261
18.3.1	Variablen/Methoden-Mehrdeutigkeit	262
18.4	Variablen und Konstanten	263
18.4.1	Gültigkeitsbereich von Konstanten und Variablen	264
18.4.2	Vordefinierte Variablen	266
18.5	Ausdrücke	271
18.5.1	Einzelne Terme	271
18.5.2	Operatorausdrücke	272
18.5.3	Mehr über Zuweisungen	273
18.5.4	Blockausdrücke	274
18.5.5	Boolesche Ausdrücke	274
18.5.6	if- und unless-Ausdrücke	277
18.5.7	Ternäre Operatoren	277
18.5.8	case-Ausdrücke	277
18.5.9	Schleifen	278
18.5.10	Break, Redo, Next und Retry	279
18.6	Methodendefinition	279
18.6.1	Methodenargumente	280
18.7	Eine Methode aufrufen	281
18.7.1	super	281
18.7.2	Operatormethoden	283
18.7.3	Attributzuweisung	283
18.7.4	Elementreferenzoperator	284
18.8	Aliasverwendung	284

18.9	Klassendefinition	285
18.9.1	Objekte von Klassen erzeugen	286
18.9.2	Klassenattributdeklarationen	286
18.10	Moduldefinitionen	286
18.10.1	Mixins-Module einbeziehen	287
18.11	Zugriffskontrolle	288
18.12	Blöcke, abgeschlossene Einheiten und Proc-Objekte	289
18.12.1	Proc-Objekte	289
18.13	Ausnahmen	290
18.13.1	Ausnahmen verursachen	290
18.14	catch und throw	292
19	Klassen und Objekte	293
19.1	Wie Klassen und Objekte interagieren	293
19.1.1	Ihr Basisobjekt	294
19.1.2	Was ist das Meta?	295
19.1.3	Objektspezifische Klassen	296
19.1.4	Mixin-Module	298
19.1.5	Objekte erweitern	299
19.2	Klassen- und Moduldefinitionen	300
19.2.1	Klassennamen sind Konstanten	305
19.3	Top-Level-Ausführungsumgebung	306
19.4	Vererbung und Sichtbarkeit	306
19.4.1	Objekte einfrieren	307
20	Ruby sichern	309
20.1	Sicherheitsebenen	310
20.2	Mangelhafte Objekte	311
21	Reflection, ObjectSpace und verteilte Programmierung mit Ruby	315
21.1	Objekte ansehen	316
21.1.1	In Objekte hineinblicken	316
21.2	Klassen betrachten	317
21.2.1	In Klassen hineinblicken	318
21.3	Methoden dynamisch aufrufen	318
21.3.1	Leistungsüberlegungen	321
21.4	System-Hooks	321
21.4.1	Laufzeit-Callbacks	323
21.5	Die Ausführung Ihres Programms verfolgen	324
21.5.1	Wie sind wir hierher gekommen?	325
21.6	Marshaling und verteilte Ruby-Programmierung	325
21.6.1	Benutzerdefinierte Serialisierungsstrategie	326
21.6.2	Verteilte Ruby-Programmierung	327
21.7	Kompilierzeit? Laufzeit? Jederzeit!	329

IV Ruby-Bibliotheksreferenz**331****22 Integrierte Klassen****333**

22.1	Alphabetische Liste	334
22.2	Die Klasse Array (Elternklasse: Object)	336
22.2.1	Mixins	336
22.2.2	Klassenmethoden	337
22.2.3	Instanzmethode	337
22.3	Die Klasse Bignum (Elternklasse: Integer)	352
22.3.1	Instanzmethode	353
22.4	Die Klasse Binding (Elternklasse: Object)	355
22.5	Die Klasse Class (Elternklasse: Module)	355
22.5.1	Klassenmethoden	356
22.5.2	Instanzmethode	357
22.6	Die Klasse Continuation (Elternklasse: Object)	357
22.6.1	Instanzmethode	358
22.7	Die Klasse Dir (Elternklasse: Object)	358
22.7.1	Mixins	358
22.7.2	Klassenmethoden	359
22.7.3	Instanzmethode	362
22.8	Die Klasse Exception (Elternklasse: Object)	363
22.8.1	Klassenmethoden	363
22.8.2	Instanzmethode	364
22.9	Die Klasse FalseClass (Elternklasse: Object)	365
22.9.1	Instanzmethode	365
22.10	Die Klasse File (Elternklasse: IO)	366
22.10.1	Mixins	367
22.10.2	Klassenmethoden	367
22.10.3	atime	367
22.10.4	Instanzmethode	373
22.11	Die Klasse File::Stat (Elternklasse: Object)	375
22.11.1	Mixins	375
22.11.2	Instanzmethode	375
22.12	Die Klasse Fixnum (Elternklasse: Integer)	382
22.12.1	Instanzmethode	382
22.13	Die Klasse Float (Elternklasse: Numeric)	384
22.13.1	Instanzmethode	384
22.14	Die Klasse Hash (Elternklasse: Object)	386
22.14.1	Mixins	387
22.14.2	Klassenmethoden	387
22.14.3	Instanzmethode	387
22.15	Die Klasse Integer (Elternklasse: Numeric)	396
22.15.1	Instanzmethode	396
22.16	Die Klasse IO (Elternklasse: Object)	398
22.16.1	Mixins	398
22.16.2	Klassenmethoden	399
22.16.3	Instanzmethode	399
		401

22.17	Die Klasse MatchData (Elternklasse: Object)	412
22.17.1	Instanzmethoden	413
22.18	Die Klasse Method (Elternklasse: Object)	415
22.18.1	Instanzmethoden	415
22.19	Die Klasse Module (Elternklasse: Object)	416
22.19.1	Klassenmethoden	417
22.19.2	Instanzmethoden	417
22.19.3	Private Instanzmethoden	423
22.20	Die Klasse NilClass (Elternklasse: Object)	428
22.20.1	Instanzmethoden	428
22.21	Die Klasse Numeric (Elternklasse: Object)	429
22.21.1	Mixins	430
22.21.2	Instanzmethoden	430
22.22	Die Klasse Object	432
22.22.1	Instanzmethoden	432
22.23	Die Klasse Proc (Elternklasse: Object)	441
22.23.1	Klassenmethoden	442
22.23.2	Instanzmethoden	442
22.24	Die Klasse Range (Elternklasse: Object)	443
22.24.1	Mixins	444
22.24.2	Klassenmethoden	444
22.24.3	Instanzmethoden	444
22.25	Die Klasse Regexp (Elternklasse: Object)	446
22.25.1	Klassenkonstanten	446
22.25.2	Klassenmethoden	446
22.25.3	Instanzmethoden	447
22.26	Die Klasse String (Elternklasse: Object)	449
22.26.1	Mixins	449
22.26.2	Klassenmethoden	449
22.26.3	Instanzmethoden	449
22.27	Die Klasse Struct (Elternklasse: Object)	471
22.27.1	Mixins	472
22.27.2	Klassenmethoden	472
22.27.3	Instanzmethoden	473
22.28	Die Klasse Struct::Tms (Elternklasse: Struct)	475
22.29	Die Klasse Symbol (Elternklasse: Object)	476
22.29.1	Instanzmethoden	476
22.30	Die Klasse Thread (Elternklasse: Object)	477
22.30.1	Klassenmethoden	477
22.30.2	Instanzmethoden	481
22.31	Die Klasse ThreadGroup (Elternklasse: Object)	485
22.31.1	ThreadGroup-Konstanten	485
22.31.2	Klassenmethoden	485
22.31.3	Instanzmethoden	485
22.32	Die Klasse Time (Elternklasse: Object)	486
22.32.1	Mixins	487
22.32.2	Klassenmethoden	487

22.32.3	Instanzmethode	489
22.33	Die Klasse TrueClass (Elternklasse: Object)	495
22.33.1	Instanzmethode	496
23	Integrierte Module	497
23.1	Alphabetische Liste	497
23.2	Das Modul Comparable	498
23.2.1	Instanzmethode	498
23.3	Das Modul Enumerable	499
23.3.1	Instanzmethode	499
23.4	Das Modul Errno	503
23.5	Das Modul FileTest	504
23.5.1	Instanzmethode	504
23.6	Das Modul GC	508
23.6.1	Klassenmethode	508
23.6.2	Instanzmethode	509
23.7	Das Modul Kernel	509
23.7.1	Klassenmethode	509
23.8	Das Modul Marshal	531
23.8.1	Klassenmethode	531
23.9	Das Modul Math	532
23.9.1	Konstanten	532
23.9.2	Klassenmethode	532
23.10	Das Modul ObjectSpace	534
23.10.1	Klassenmethode	534
23.11	Das Modul Process	535
23.11.1	Konstanten	536
23.11.2	Klassenmethode	536
24	Standardbibliothek	541
24.1	Die Klasse Complex (Elternklasse: Numeric) require »complex«	541
24.1.1	Klassenkonstanten	541
24.1.2	Klassenmethode	541
24.1.3	Instanzmethode	542
24.2	Die Klasse Date (Elternklasse: Object) require »date«	544
24.2.1	Mixins	546
24.2.2	Klassenmethode	546
24.2.3	Instanzmethode	548
24.3	Die Bibliothek English require »English«	552
24.4	Das Modul Find require »find«	553
24.4.1	Modulmethode	553
24.5	Die Klasse File (Elternklasse: IO) require »ftools«	553
24.5.1	Klassenmethode	553
		554

24.6	Die Klasse GetoptLong (Elternklasse: Object) require »getoptlong«	556
24.6.1	Klassenkonstanten	557
24.6.2	Klassenmethoden	557
24.6.3	Instanzmethode	557
24.7	Das Modul mkmf require »mkmf«	560
24.7.1	Modulkonstanten	560
24.7.2	Instanzmethode	560
24.8	Das Modul ParseDate require »parsedate«	562
24.8.1	Modulmethoden	562
24.9	Die Bibliothek profile require »profile«	563
24.10	Die Klasse PStore (Elternklasse: Object) require »pstore«	563
24.10.1	Klassenmethoden	564
24.10.2	Instanzmethode	565
24.11	Die Klasse Tempfile (Elternklasse: IO) require »tempfile«	566
24.11.1	Klassenmethoden	566
24.11.2	Instanzmethode	567
24.12	Die Klasse Mutex (Elternklasse: Object) require »thread«	567
24.12.1	Instanzmethode	568
24.13	Die Klasse ConditionVariable (Elternklasse: Object) require »thread«	569
24.13.1	Instanzvariablen	569
24.14	Die Bibliothek timeout require »timeout«	570
24.15	Die Klasse WeakRef (Elternklasse: Delegator) require »weakref«	570
24.15.1	Klassenmethoden	571
24.15.2	Instanzmethode	572
25	Bibliotheken für den objektorientierten Entwurf	573
25.1	Das Besucher-Muster	573
25.2	Die Bibliothek delegate require »delegate«	574
25.3	Die Bibliothek observer	576
25.4	Die Bibliothek singleton	578
26	Netzwerk- und Webbibliotheken	579
26.1	Zugriff auf der Socket-Ebene	579
26.2	Die Klasse BasicSocket (Elternklasse: IO) require »socket«	581
26.2.1	Klassenmethoden	581

26.2.2	Instanzmethoden	582
26.3	Die Klasse IPSocket (Elternklasse: BasicSocket) require »socket«	583
26.3.1	Klassenmethoden	583
26.3.2	Instanzmethoden	583
26.4	Die Klasse TCPSocket (Elternklasse: IPSocket) require »socket«	584
26.4.1	Klassenmethoden	584
26.4.2	Instanzmethoden	584
26.5	Die Klasse SOCKSSocket (Elternklasse: TCPSocket) require »socket«	585
26.5.1	Klassenmethoden	585
26.5.2	Instanzmethoden	585
26.6	Die Klasse TCPServer (Elternklasse: TCPSocket) require »socket«	585
26.6.1	Klassenmethoden	586
26.6.2	Instanzmethoden	586
26.7	Die Klasse UDPSocket (Elternklasse: IPSocket) require »socket«	586
26.7.1	Klassenmethoden	587
26.7.2	Instanzmethoden	587
26.8	Die Klasse UNIXSocket (Elternklasse: BasicSocket) require »socket«	588
26.8.1	Klassenmethoden	589
26.8.2	Instanzmethoden	589
26.9	Die Klasse UNIXServer (Elternklasse: UNIXSocket) require »socket«	590
26.9.1	Klassenmethoden	590
26.9.2	Instanzmethoden	590
26.10	Die Klasse Socket (Elternklasse: BasicSocket) require »socket«	591
26.10.1	Konstanten	591
26.10.2	Klassenmethoden	592
26.10.3	Instanzmethoden	594
26.11	Zugriff auf höherer Ebene	595
26.12	Die Klasse Net::FTP (Elternklasse: Object) require »net/ftp«	595
26.12.1	Klassenkonstanten	595
26.12.2	Klassenmethoden	595
26.12.3	Instanzmethoden	596
26.13	Die Klasse Net::HTTP (Elternklasse: Net::Protocol) require »net/http«	600
26.13.1	Klassenmethoden	602
26.13.2	Instanzmethoden	602
26.14	Die Klasse Net::HTTPResponse require »net/http«	603
26.14.1	Instanzmethoden	603

26.15	Die Klasse Net::POP (Elternklasse: Net::Protocol) require »net/pop«	604
26.15.1	Klassenmethoden	605
26.15.2	Instanzmethode	605
26.16	Die Klasse Net::APOP (Elternklasse: Net::POP3) require »net/pop«	606
26.16.1	Instanzmethode	606
26.17	Die Klasse Net::POPmail (Elternklasse: Object) require »net/pop«	607
26.17.1	Instanzmethode	607
26.18	Die Klasse Net::SMTP (Elternklasse: Net::Protocol) require »net/smtp«	608
26.18.1	Klassenmethoden	608
26.18.2	Instanzmethode	609
26.19	Die Klasse Net::Telnet (Elternklasse: [Socket]) require »net/telnet«	610
26.19.1	Klassenmethoden	611
26.19.2	Instanzmethode	612
26.20	CGI-Entwicklung	613
26.21	Die Klasse CGI (Elternklasse: Object) require »cgi«	613
26.21.1	Klassenmethoden	615
26.21.2	Instanzmethode	617
26.21.3	HTML-Ausgabemethoden	618
26.21.4	Werte in mehrteiligen Formularen	621
26.22	Die Klasse CGI::Session (Elternklasse: Object) require »cgi/session«	621
26.22.1	Klassenmethoden	622
26.22.2	Instanzmethode	622
27	Microsoft Windows-Unterstützung	625
27.1	Die Klasse WIN32OLE (Elternklasse: Object) require »win32ole«	625
27.1.1	Konstante	625
27.1.2	Klassenmethoden	625
27.1.3	Instanzmethode	626
27.2	Die Klasse WIN32OLE_EVENT (Elternklasse: Object) require »win32ole«	626
27.2.1	Klassenmethoden	627
27.2.2	Instanzmethode	628
27.3	Die Klasse Win32API (Elternklasse: Object) require »Win32API«	628
27.3.1	Klassenmethode	629
27.3.2	Instanzmethode	629

A	Eingebettete Dokumentation	631
A.1	Inline-Formatierung	633
A.2	Querverweise	634
A.3	Methodennamen	635
A.4	Andere Dateien einfügen	635
A.5	rdtool verwenden	635
A.6	Obligatorische Haftungsausschlusserklärung	636
B	Interaktive Ruby-Shell	637
B.1	Kommandozeile	637
B.2	Die Initialisierungsdatei	638
B.3	Befehle	639
B.3.1	Konfiguration der Eingabeaufforderung	641
B.4	Beschränkungen	643
B.5	rtags, xmp und die Klasse Frame	643
B.5.1	rtags	644
B.5.2	xmp	644
B.5.3	Die Klasse Frame	645
C	Unterstützung	647
C.1	Websites	647
C.2	Download-Sites	648
C.3	Usenet-Newsgroup	648
C.4	Mailinglisten	648
C.5	Programmfehler melden	649
D	Bibliografie	651
	Index	653