

Inhaltsverzeichnis

1	Einleitung	1
1.1	Warum dieses Buch?	1
1.1.1	Was dieses Buch vermittelt	1
1.1.2	Was dieses Buch nicht ist	2
1.2	Aufbau des Buches	3
1.3	Konventionen in diesem Buch	4
1.4	Installation der Beispiele	5
1.4.1	Die Eclipse IDE installieren	5
1.4.2	Eclipse Equinox installieren	5
1.4.3	Die Standard Services aus dem Incubator installieren	6
1.4.4	Die Target Platform einrichten	7
1.4.5	Installation der Codebeispiele	7
1.4.6	Arbeiten mit dem Eclipse Samples Manager	8
1.5	Danksagung	9
1.6	Website zum Buch	10
2	OSGi-Technologie im Überblick	11
2.1	Die OSGi Service Platform	12
2.1.1	Das OSGi Framework	12
2.1.2	OSGi Standard Services	13
2.1.3	Wo wird die OSGi Service Platform eingesetzt? . . .	14
2.1.4	Wo ist die OSGi Service Platform spezifiziert? . . .	15
2.2	Welche Vorteile bietet der Einsatz der OSGi Service Platform?	16
2.2.1	Modularisierung	16
2.2.2	Abhängigkeitsmanagement	17
2.2.3	Hot Deployment	18
2.2.4	Serviceorientiertes Programmiermodell	18

- 2.3 Das OSGi Framework 19
 - 2.3.1 Die Execution Environments 20
 - 2.3.2 Die Module-Schicht 21
 - 2.3.3 Die Lifecycle-Management-Schicht 22
 - 2.3.4 Die Service-Schicht 23
 - 2.3.5 Die Security-Schicht 24
 - 2.3.6 Die Framework Services 24
- 2.4 OSGi Standard Services 25
 - 2.4.1 Gemeinsame Service-Spezifikationen 26
 - 2.4.2 Weitere Services des Service Compendium 28
- 2.5 Implementierungen 29
 - 2.5.1 Open-Source-Implementierungen 29
 - 2.5.2 Kommerzielle Implementierungen 30
- 2.6 Eclipse Equinox 30
 - 2.6.1 Die Eclipse-Equinox-Distribution 30
 - 2.6.2 Eclipse Equinox als Grundlage der Eclipse-Laufzeiumgebung 31
 - 2.6.3 Die Eclipse IDE als Entwicklungsplattform für OSGi Bundles 31
- 2.7 Die OSGi Service Platform in Aktion 32
- 2.8 Zusammenfassung 37

Teil I: Das OSGi Framework

3	Bundles	41
3.1	Tutorial: Ein »Hello World«-Bundle	42
3.2	Bundles im Überblick	46
3.3	Das Bundle Manifest	47
	3.3.1 Der Plug-in Manifest Editor	47
	3.3.2 Das Bundle Manifest in textueller Form	48
	3.3.3 Bedeutung der verwendeten Bundle Manifest Header	49
3.4	Der Bundle-Aktivator	50
	3.4.1 Das Interface BundleActivator	50
3.5	Der Bundle-Kontext	50
	3.5.1 Das Interface BundleContext	50
3.6	Der Bundle-Klassenpfad	52
3.7	Zusammenfassung	53

4	Bundle-Lebenszyklus	55
4.1	Tutorial: Den Bundle-Lebenszyklus über die Equinox-Konsole steuern	56
4.2	Der Bundle-Lebenszyklus im Überblick	61
4.2.1	Installieren, Starten, Stoppen, Deinstallieren	61
4.2.2	Update und Refresh	62
4.3	Activation Policies	63
4.3.1	Motivation	63
4.3.2	Die Lazy Activation Policy	64
4.4	Programmatische Steuerung des Bundle-Lebenszyklus	65
4.4.1	Das Interface Bundle	65
4.4.2	Programmatische Installation von Bundles	67
4.5	Bundle Listener und Framework Listener	68
4.5.1	Bundle Listener	68
4.5.2	Framework Listener	69
4.6	Der Bundle Cache	70
4.6.1	Der Bundle Cache in Eclipse Equinox	70
4.7	Zusammenfassung	71
5	Package-Abhängigkeiten zwischen Bundles	73
5.1	Tutorial: Ein »Translation«-Bundle	74
5.2	Exportieren von Packages	79
5.3	Importieren von Packages	80
5.3.1	Importieren mit Import-Package	80
5.3.2	Importieren mit Require-Bundle	81
5.4	Import-Package oder Require-Bundle?	82
5.5	Arbeiten mit Bundle- und Package-Versionen	83
5.5.1	Spezifikation von Bundle- und Package-Versionen	83
5.5.2	Spezifikation von Abhängigkeiten mit Versionsbereichen	84
5.6	Dynamisches Importieren von Packages	85
5.7	Exkurs: Class Loading in Java und im OSGi Framework	87
5.7.1	Class Loading in Java	87
5.7.2	Class Loading im OSGi Framework	89
5.7.3	Class Loading über den Context Class Loader	90
5.8	Zusammenfassung	91

6	OSGi Services	93
6.1	Tutorial: Der Translation Service als OSGi Service	94
6.2	OSGi Services im Überblick	98
6.3	OSGi Services registrieren und deregistrieren	99
6.3.1	Registrieren von OSGi Services	99
6.3.2	Deregistrierung von OSGi Services	101
6.4	Service Factories	101
6.5	Services verwenden	102
6.6	Service Properties	104
6.7	Filter	106
6.8	Package- und Service-Abhängigkeiten	107
6.8.1	Kompatibilität zwischen Package- und Service- Abhängigkeit	107
6.8.2	Abfragen aller verfügbaren Services	109
6.9	Verwendung von Start-Leveln	110
6.10	Zusammenfassung	111
7	Umgang mit dynamischen Services	113
7.1	Tutorial: Ein dynamischer Translation Service	115
7.2	Services in dynamischen Systemen	120
7.2.1	Szenario 1: Service wurde nicht registriert	120
7.2.2	Szenario 2: Service wurde deregistriert	121
7.2.3	Szenario 3: Service wurde inkompatibel aktualisiert	121
7.3	Arbeiten mit dem Service Tracker	122
7.3.1	Erzeugen eines Service Trackers	123
7.3.2	Öffnen und Schließen des Service Trackers	124
7.3.3	Arbeiten mit Services	124
7.3.4	Arbeiten mit Service References	125
7.3.5	Abfragen von Zustandsinformationen	126
7.4	Das Interface ServiceTrackerCustomizer	126
7.5	Service Listener und Service Events	128
7.5.1	Service Listener verwenden	129

7.6	Exkurs: Das Whiteboard-Pattern	130
7.6.1	Beispiel: Translation Listener	131
7.6.2	Implementierung nach dem Event-Listener- Pattern	132
7.6.3	Implementierung nach dem Whiteboard-Pattern	136
7.7	Zusammenfassung	138
8	Fragment Bundles	139
8.1	Tutorial: »Language«-Fragmente	140
8.2	Fragment Bundles im Detail	144
8.3	Anwendungsszenarien für Fragment Bundles	145
8.4	Zuordnung von Fragment Bundles und Host Bundle	146
8.5	Das Framework erweitern: Extension Bundles	147
8.6	Zusammenfassung	148
9	Security	149
9.1	Tutorial: Ein sicherer Translation Service	150
9.2	Vergabe von Ausführungsrechten im OSGi Framework	156
9.2.1	Rechtevergabe in geschlossenen Systemen	157
9.2.2	Rechtevergabe in Systemen mit offenen Deploy- ment-Kanälen	157
9.2.3	Das Delegationsmodell	158
9.3	Die lokalen Berechtigungen eines Bundles	159
9.4	Globale Berechtigungen vergeben	160
9.5	OSGi-Framework-spezifische Permissions	162
9.6	Zusammenfassung	163
10	Management der OSGi Service Platform	165
10.1	Tutorial: Die Equinox-Konsole um ein greet-Kommando erweitern	166
10.2	Management Agents im Überblick	170
10.2.1	Management Bundles	170
10.2.2	Open-Source-Implementierungen	171

10.3	Die Equinox-Konsole im Detail	173
10.3.1	Kommandos zur Steuerung des Frameworks . . .	173
10.3.2	Kommandos zur Steuerung von Bundles	173
10.3.3	Kommandos zum Anzeigen von Statusinformationen	174
10.3.4	Kommandos zur Manipulation der Startlevel . . .	176
10.3.5	Weitere Kommandos	176
10.4	Die Equinox-Konsole um eigene Befehle erweitern	176
10.4.1	Das Interface CommandProvider	177
10.4.2	Das Interface CommandInterpreter	177
10.5	Exkurs: Der Knopflerfish-Desktop als alternativer Management Agent	178
10.5.1	Installation des Knopflerfish-Desktops	179
10.5.2	Arbeiten mit dem Knopflerfish-Desktop	180
10.6	Zusammenfassung	181
11	Packaging und Deployment	183
11.1	Tutorial: Ausführen des »Hello World«-Beispiels von der Kommandozeile	184
11.2	Der Plug-in-Export-Wizard	188
11.3	Automatisiertes Bauen von Bundles	189
11.4	Exkurs: Der PDE Headless Build	190
11.4.1	Schritt 1: Anlegen eines Feature-Projektes	190
11.4.2	Schritt 2: Anlegen eines Build-Projektes	191
11.4.3	Schritt 3: Anpassen der build.properties-Datei	192
11.4.4	Schritt 4: Die allElements.xml-Datei	193
11.4.5	Schritt 5: Die customTargets.xml-Datei	193
11.4.6	Schritt 6: Ausführen des PDE-Builds	194
11.5	Ausführen von Eclipse Equinox	195
11.5.1	Starten von Equinox von der Kommandozeile	195
11.5.2	Starten von Equinox aus einer Java-Anwendung	195
11.5.3	Equinox konfigurieren	197
11.6	Zusammenfassung	198

Teil II: OSGi Standard Services

12	Declarative Services	201
12.1	Tutorial: Services deklarativ beschreiben	202
12.2	Überblick über Declarative Services	208
12.3	Service Components	209
12.3.1	Die Komponenteklasse	209
12.3.2	Die Komponentenbeschreibung	211
12.4	Services bereitstellen	212
12.5	Services benutzen	213
12.5.1	Die Event-Strategie	214
12.5.2	Die Lookup-Strategie	215
12.5.3	Die Kardinalität einer Referenz	216
12.5.4	Die Policy einer Referenz	217
12.6	Der Lebenszyklus von Komponenten	219
12.6.1	Der Zustand »enabled«	219
12.6.2	Immediate und Delayed Components	219
12.6.3	Lebenszyklus von Immediate Components	219
12.6.4	Lebenszyklus von Delayed Components	220
12.7	Component Properties und Factory Components	221
12.7.1	Component Properties	221
12.7.2	Factory Components	223
12.8	Zusammenfassung	225
13	Der Config Admin Service	227
13.1	Tutorial: Ein administrierbarer Translation Service	227
13.2	Überblick über den Config Admin Service	235
13.3	Die Persistent Identity (PID)	236
13.4	Managed Services	237
13.4.1	Das Interface ManagedService	237
13.4.2	Registrieren eines Managed Service	238
13.5	Managed Service Factories	239
13.5.1	Das Interface ManagedServiceFactory	239
13.5.2	Registrieren einer Managed Service Factory	241

13.6	Konfigurationen programmatisch verwalten	241
13.6.1	Die Interfaces ConfigurationAdmin und Configuration	241
13.6.2	Erzeugen und Abfragen von Konfigurationen . .	242
13.6.3	Ändern und Löschen von Konfigurationen	244
13.7	Zusammenfassung	245
14	Der Metatype Service	247
14.1	Tutorial: Den Translation Service konfigurieren	247
14.2	Überblick über den Metatype Service	252
14.3	Beschreibung von Metatyp-Informationen in XML	253
14.3.1	Object-Class- und Attribute-Definitionen	254
14.3.2	Zuordnung zu ManagedService und Managed- ServiceFactory	257
14.3.3	Lokalisierung	257
14.4	Programmatische Abfrage von Metadaten	259
14.4.1	Das Interface MetaTypeService	259
14.4.2	Das Interface ObjectClassDefinition	260
14.4.3	Das Interface AttributeDefinition	261
14.5	Zusammenfassung	262
15	Der Event Admin Service	265
15.1	Tutorial: Ein Timer Service	266
15.2	Der Event Admin Service im Überblick	270
15.3	Die Klasse org.osgi.service.event.Event	271
15.3.1	Erzeugen von Events	272
15.3.2	Standard-Properties	272
15.3.3	Abfragen eventspezifischer Informationen	273
15.3.4	Mapping von Framework Events	274
15.4	Events veröffentlichen	274
15.4.1	Synchrones Versenden von Events	275
15.4.2	Asynchrones Versenden von Events	276
15.4.3	Fehlerbehandlung	277
15.5	Events empfangen	277
15.5.1	Das Interface org.osgi.event.EventHandler	277
15.5.2	Registrierung von Event Handlern	277
15.6	Zusammenfassung	279

16	Der Log Service	281
16.1	Tutorial: Logging in der OSGi Service Platform	281
16.2	Überblick über den Log Service	287
16.3	Verwenden des Log Service	288
16.4	Verwenden des Log Reader Service	289
16.4.1	Der LogReaderService	289
16.4.2	LogEntry	290
16.4.3	LogListener	290
16.5	Der Log Service in Eclipse Equinox	291
16.5.1	Log-Nachrichten auf der Equinox-Konsole aus- geben	291
16.5.2	Den Log Service konfigurieren	291
16.6	Zusammenfassung	292
17	Der User Admin Service	293
17.1	Tutorial: Ein Login-Kommando	293
17.2	Der User Admin Service im Überblick	301
17.3	Rollen, Benutzer und Gruppen	302
17.3.1	Rollen	302
17.3.2	Benutzer	302
17.3.3	Gruppen	303
17.4	Rollen, Benutzer und Gruppen verwalten	304
17.5	Authentifizierung	305
17.6	Autorisierung	306
17.7	Zusammenfassung	308
18	Der Preferences Service	311
18.1	Tutorial: Persistieren der letzten Login-Zeit	311
18.2	Überblick über den Preferences Service	315
18.3	Zugriff auf Preferences	317
18.3.1	Das Interface PreferencesService	317
18.3.2	Das Interface Preferences	318
18.3.3	Navigation durch den Preferences-Baum	320
18.3.4	Löschen von Preferences-Knoten	320
18.4	Schreiben und Lesen von Properties	321
18.4.1	Properties auslesen	321
18.4.2	Einfügen und Ändern von Properties	321
18.4.3	Entfernen von Properties	322

18.5	Nebenläufigkeit	322
18.6	Zusammenfassung	323
19	Der Http Service	325
19.1	Tutorial: »Hello World« über Web	326
19.2	Überblick über den Http Service	331
19.3	Das Interface <code>HttpService</code>	332
19.3.1	Registrieren und Deregistrieren von statischen Ressourcen	333
19.3.2	Registrieren und Deregistrieren von Servlets ...	334
19.3.3	Abbildung von <code>Http Requests</code> auf Servlets und Ressourcen	335
19.4	Der <code>Http Context</code>	336
19.4.1	Das Interface <code>HttpContext</code>	336
19.4.2	Der <code>Default Http Context</code>	337
19.4.3	<code>Http Context</code> und <code>Servlet Context</code>	337
19.5	Konfigurieren des <code>Http Service</code>	338
19.6	Zusammenfassung	338
20	Der Application Admin Service	341
20.1	Tutorial: Externe Anwendungen starten	341
20.2	Der <code>Application Admin Service</code> im Überblick	349
20.3	Der <code>Application Container</code>	350
20.3.1	Der <code>Application Descriptor</code>	350
20.3.2	Der <code>Application Handle</code>	353
20.4	<code>Application Manager</code>	354
20.4.1	Anwendungen ermitteln	354
20.4.2	Anwendungen starten und stoppen	355
20.4.3	Scheduling	356
20.4.4	Anwendungen zeitgesteuert ausführen	358
20.4.5	Der <code>Equinox Application Manager</code>	359
20.5	Zusammenfassung	359

Teil III: Eclipse-Equinox-spezifische Erweiterungen

21	Erweiterungen des OSGi Class Loading	363
21.1	Tutorial: Grußbotschaften aus XML-Datei einlesen	363
21.2	Buddy Class Loading	371
21.2.1	Eclipse-BuddyPolicy: dependent	372
21.2.2	Eclipse-BuddyPolicy: registered	373
21.2.3	Eclipse-BuddyPolicy: global	373
21.2.4	Verwenden der Standard-Java-Class-Loader	374
21.2.5	Buddy Class Loading vs. DynamicImport- Package	374
21.2.6	Konsequenzen des Buddy Class Loading	375
21.3	Die Sichtbarkeit exportierter Packages beschränken	376
21.3.1	Die Export-Package-Direktive x-friend	376
21.3.2	Die Export-Package-Direktive x-internal	377
21.4	Zusammenfassung	377
22	Extension Points und Extensions	379
22.1	Tutorial: Ein erweiterbarer Dialog	380
22.2	Extension Points und Extensions im Überblick	389
22.3	Extension Points definieren	390
22.3.1	Deklarative Beschreibung	390
22.3.2	Definition von Extension Points im PDE	390
22.4	Extension definieren	391
22.4.1	Eine Extension definieren	391
22.4.2	Definition von Extensions im PDE	392
22.5	Die Extension Registry	393
22.5.1	Das Interface IExtensionRegistry	393
22.5.2	Programmatischer Zugriff auf Extension Points, Extensions und Konfigurationselemente	395
22.6	Executable Extensions	397
22.6.1	Executable Extensions erzeugen	397
22.6.2	Zugriff auf Konfigurationsdaten: IExecutable- Extension	397
22.6.3	IExecutableExtensionFactory	398
22.7	Dynamische Extensions	400
22.7.1	Das Interface IRegistryChangeListener	400
22.7.2	Das Interface IExtensionTracker	401
22.7.3	Das Interface IExtensionChangeHandler	402

22.8	Extension Points vs. OSGi Services	403
22.9	Zusammenfassung	404
23	Das Equinox Application Model	407
23.1	Tutorial: Eine »Hello World« Application	408
23.2	Das Equinox Application Model im Überblick	413
23.3	Equinox-Anwendungen implementieren	414
23.3.1	Das Interface IApplication	414
23.3.2	Der Extension Point org.eclipse.core.runtime. applications	415
23.4	Ausführen von Anwendungen	417
23.4.1	Ausführen einer Anwendung beim Start des Frameworks	417
23.4.2	Verwendung des Application Admin Service ...	418
23.5	Zusammenfassung	419
24	Webanwendungen mit Eclipse Equinox	421
24.1	Tutorial: Eine Webanwendung mit JSPs und Extensions .	421
24.2	Überblick über die webspezifischen Erweiterungen	426
24.3	Verwenden der Extension Registry	427
24.3.1	Registrieren von Ressourcen	427
24.3.2	Registrieren von Servlets	428
24.3.3	Registrieren von Http-Context-Instanzen	429
24.4	Verwendung der Jasper JSP Engine	430
24.4.1	Programmatische Registrierung von JSPs	431
24.4.2	Registrierung von JSPs über die Extension Registry	432
24.5	Servlet Container in Eclipse Equinox einbetten	432
24.5.1	Das Http Service Servlet	432
24.5.2	Der Jetty Webserver als eingebetteter Servlet Container	433
24.6	Eclipse Equinox in einen Servlet Container einbetten ...	433
24.6.1	Installieren der Servlet Bridge	434
24.6.2	Konfiguration	435
24.6.3	Administration	436
24.7	Zusammenfassung	437
25	Links und Literatur	439
	Index	443