

Inhaltsverzeichnis

I	Einstieg in Eclipse und CDT	1
1	Installation	3
1.1	C/C++-Compiler	3
1.2	Das richtige Eclipse-Archiv finden	4
1.3	Installation unter Linux	5
1.3.1	Compiler und Tools	6
1.3.2	Java	7
1.3.3	Eclipse	7
1.4	Installation unter Windows	8
1.4.1	Compiler und Tools	8
1.4.2	Java	14
1.4.3	Eclipse	14
1.4.4	Wascanal	15
1.5	Die Eclipse.ini-Datei	16
1.6	Willkommen	18
1.7	Eclipse erweitern	19
1.7.1	CDT nachträglich installieren	20
1.7.2	Update-Sites	22
1.7.3	Features verwalten	25
1.7.4	Die CDT-Features im Detail	26
1.7.5	Eclipse Marketplace	27
1.7.6	p2	29
2	Erste Schritte	31
2.1	Konzepte des Eclipse-Frameworks	31
2.1.1	Workspace	31
2.1.2	Ressourcen	32
2.1.3	Eclipse-Workbench	32
2.1.4	Menüleiste	32
2.1.5	Werkzeugleiste	34
2.1.6	Statusleiste	34
2.1.7	Editoren und Views	34
2.1.8	Perspektiven	37

2.2	Das erste Projekt	38
2.2.1	Eclipse anpassen	39
2.2.2	Ablauf	40
2.2.3	Ein neues Projekt anlegen	41
2.2.4	Die Quelldatei erzeugen	41
2.2.5	Übersetzen	43
2.2.6	Ausführen	45
2.2.7	Programmfehler aufspüren	45
2.2.8	Zusammenfassung und Ausblick	48
2.3	Das Eclipse-Hilfesystem	48
2.3.1	Dokumentationsaufbau	48
2.3.2	Hilfefenster	49
2.3.3	Kontextbezogene Hilfe	51
2.4	Eclipse über die Tastatur steuern	52
2.5	Nützliche Einstellungen	53
2.5.1	Voreinstellungsdialog	54
2.5.2	Hierarchie der Voreinstellungen	54
2.5.3	Allgemeine Workspace-Einstellungen	55
2.5.4	Workspace-Auswahl beim Starten und Beenden	57
2.5.5	Netzwerkeinstellungen	58
2.5.6	Einstellungen für SSH-Client	58
2.5.7	Passwortspeicher	60
2.5.8	Komponenten (de-)aktivieren	60
2.6	Die Benutzerschnittstelle anpassen	61
2.6.1	Werkzeug- und Menüleiste anpassen	62
2.6.2	Perspektiven verwalten	65
2.6.3	Tastatursteuerung anpassen	65
2.6.4	Erscheinungsbild	67
2.7	Verzeichnislayout auf dem Datenträger	68
2.7.1	Programmverzeichnis	69
2.7.2	Workspace	72

II Entwicklung mit CDT

73

3	Projekte erstellen und bearbeiten	75
3.1	Konzepte von CDT	76
3.1.1	Parser	76
3.1.2	Indexer	76
3.1.3	Piktogramme	77
3.1.4	Kurze Einführung in das Build-System	78
3.2	C/C++-Projekte in CDT erzeugen	79
3.2.1	Projekte anlegen	80

3.2.2	Dateien erzeugen	83
3.2.3	Verzeichnisse hinzufügen	88
3.2.4	Vorhandene Projekte importieren	89
3.2.5	Projekt entfernen	93
3.2.6	Projekttyp ändern	94
3.3	Eigenschaften von Ressourcen	94
3.4	Projekt: Dirscanner	96
3.4.1	Boost installieren	97
3.4.2	Dirscanner entwickeln	99
3.5	Quelltexte bearbeiten	104
3.5.1	Grundsätzlicher Aufbau eines Editorfeldes	104
3.5.2	Textpassagen markieren	104
3.5.3	Quelltext formatieren	105
3.5.4	Editierhilfen	107
3.5.5	Präsentation der Quelltexte	107
3.5.6	Annotationen	111
3.5.7	Code-Analyse	113
3.5.8	Navigation	115
3.5.9	Content-Assistenz	116
3.5.10	Makros untersuchen	119
3.5.11	Find/Replace	120
3.5.12	Setters und Getters erzeugen	124
3.5.13	Methode implementieren	124
3.5.14	Kommentare für externe Dokumentationswerkzeuge	125
3.5.15	Bookmarks	127
3.5.16	Makefile	128
3.5.17	Undo/Redo-Funktion	128
3.6	Dateienvergleich	129
3.6.1	Zwei Dateien vergleichen	129
3.6.2	Drei Dateien vergleichen	131
3.6.3	Lokale Historie	132
3.7	Working-Sets	133
3.7.1	Ein Working-Set erstellen	134
3.7.2	Working-Sets einschalten	136
3.7.3	Workbench-Working-Sets	136
3.8	Tasks	136
3.8.1	Tasks hinzufügen	137
3.8.2	Tasks durch spezielle Schlüsselwörter	137
3.8.3	Tasks anzeigen lassen	138
3.9	Refaktorisierung	139
3.9.1	Ressourcen umbenennen	139
3.9.2	Bezeichner umbenennen	140
3.9.3	Konstanten extrahieren	143

3.9.4	Lokale Variablen extrahieren	144
3.9.5	Eine Funktion oder Methode extrahieren	145
3.9.6	Methoden verstecken	148
3.9.7	Refaktorisierungshistorie	148
3.10	Parser- und Indexerdetails	149
3.10.1	Parser	149
3.10.2	Indexeroptionen anpassen	151
3.10.3	Weitere Indexer-Parameter	155
3.10.4	Indexdatenbank analysieren	155
3.10.5	Probleme mit dem Indexer	156
3.11	Views der C/C++-Perspektive	157
3.11.1	Outline	157
3.11.2	Include Browser	158
3.11.3	Call Hierarchy	160
3.11.4	Type Hierarchy	161
3.12	Navigations-Views	162
3.12.1	Navigator	164
3.12.2	C/C++ Project	164
3.12.3	Project Explorer	166
3.13	Elemente finden	166
3.14	Suchen und Ersetzen	167
3.14.1	File Search	169
3.14.2	C/C++ Search	170
3.15	Textschablonen anpassen	171
3.15.1	Editorschablonen	172
3.15.2	Quelltextschablonen	174
3.16	Projektspezifische Metadateien	175
3.16.1	Metadateien im Projektverzeichnis	175
3.16.2	Metadateien im Workspace-Verzeichnis	176
4	Der Build-Prozess	177
4.1	Builder	178
4.1.1	Die vier Modi eines Builders	178
4.1.2	Builder-Assoziationen	179
4.1.3	Builder-Konfiguration	180
4.1.4	Zwei Builder für C/C++-Projekte	181
4.2	Build-Variablen	182
4.2.1	Kennzeichen von Build-Variablen	182
4.2.2	Build-Variablen benutzen	183
4.2.3	Gültigkeitsbereich von Build-Variablen	183
4.2.4	Workspace-weite Build-Variablen definieren	185
4.2.5	Umgebungsvariablen als Build-Variablen	186
4.2.6	Dynamische Build-Variablen	186

4.3	Kontrolle über Managed Build	187
4.3.1	Konfigurationen verwalten	187
4.3.2	Quellverzeichnisse	187
4.3.3	Dateien vom Build-Prozess ausschließen	189
4.3.4	Custom Build	190
4.4	Einstellungen für C/C++-Projekte.....	190
4.4.1	C/C++ Build	190
4.4.2	Discovery Options	193
4.4.3	Paths and Symbols.....	195
4.4.4	Environment.....	197
4.4.5	Logging	199
4.4.6	Settings	199
4.4.7	Tool Chain Editor	206
4.4.8	Build Variables	208
4.5	Einstellungen für Ressourcen.....	208
4.5.1	Discovery Options	210
4.5.2	Settings	210
4.5.3	Tool Chain Editor	212
4.5.4	Path and Symbols.....	213
4.6	Makefile-Projekte	213
4.6.1	Der <i>Make Targets</i> -View	213
4.6.2	Make Targets hinzufügen	214
4.6.3	Make Targets bauen	215
4.6.4	Make Targets entfernen	216
4.6.5	Besonderheiten	216
4.7	Compiler-Ausgaben	216
4.7.1	Console	216
4.7.2	Voreinstellungen zur Build-Konsole	218
4.7.3	Wie CDT Kommandoausgaben verarbeitet	218
4.7.4	Error Parser festlegen	219
4.7.5	Fehler finden	220
4.7.6	Fehler filtern	221
4.8	Code-Generatoren verwenden	223
4.8.1	Beispiel: mathematische Ausdrücke auswerten	224
4.8.2	Bison installieren	225
4.8.3	Projekt erstellen und Quelltexte einpflegen	225
4.8.4	Einstellungen anpassen	228
4.8.5	Projekt übersetzen und ausführen	230
4.9	Cross-Compiling	231
4.9.1	Cross-Compiling für Make-Projekte	231
4.9.2	Cross-Compiling für Managed Build	231
4.9.3	Beispiel: AmigaOS4	232
4.9.4	Cross-Compiling-Plugin	234

4.10	Remote Compiling	234
4.10.1	Das Programm ssh einrichten	235
4.10.2	Das Dateisystem einrichten	236
4.10.3	Ein Remote-Build-Kommando erstellen	237
4.10.4	Remote Compiling in Eclipse anwenden	238
4.11	Microsoft Visual C++ einbinden	239
4.11.1	Visual C++ installieren	240
4.11.2	Das Projekt einrichten	241
4.11.3	Den Build-Prozess anpassen	243
5	Ausführen und Debugging	249
5.1	Programmstart-Dialoge	250
5.1.1	Main	252
5.1.2	Arguments	253
5.1.3	Environment	253
5.1.4	Common	253
5.2	Doxygen einbinden	255
5.2.1	Konfigurationsdatei anlegen	255
5.2.2	Doxygen-Tool einrichten	256
5.2.3	Doxygen aufrufen	257
5.3	Die Launch-Konsole	258
5.4	Programme debuggen	259
5.4.1	Programme im Debug-Modus starten	259
5.4.2	Multiple Debug-Launcher	260
5.4.3	Debug-Perspektive	261
5.4.4	Quelltextpfad lokalisieren	262
5.4.5	Der Debug-View	263
5.5	Den Programmablauf gezielt unterbrechen	267
5.5.1	Breakpoints-View	267
5.5.2	Zeilen-Breakpoints	269
5.5.3	Data-Breakpoints	271
5.5.4	Address-Breakpoints	271
5.5.5	Event-Breakpoints	272
5.5.6	Eigenschaften von Breakpoints	272
5.6	Nützliche Views beim Debuggen	275
5.6.1	Variables	276
5.6.2	Expressions	278
5.6.3	Register	279
5.6.4	Disassembly	281
5.6.5	Signale	281
5.6.6	Module	282
5.6.7	Memory	283
5.6.8	Executables	287

5.7	Launch Groups	288
5.7.1	Auf entfernten Rechnern debuggen	289

III Fortgeschrittene Entwicklung 291

6	Versionsverwaltung	293
6.1	Grundlagen	294
6.1.1	Motivation	294
6.1.2	Begriffe	296
6.2	Serverseitige CVS-Komponenten installieren	297
6.2.1	Windows	297
6.2.2	Linux	298
6.3	CVS	298
6.3.1	CVS-Unterstützung von Eclipse einrichten	299
6.3.2	Perspektive <i>CVS Repository Exploring</i>	299
6.3.3	Repositorys verwalten	299
6.3.4	Projekte auschecken	304
6.3.5	Dekorationen	305
6.3.6	Dateien aktualisieren	305
6.3.7	Konflikte	306
6.3.8	Veränderte Dateien einchecken	306
6.3.9	Neue Dateien einchecken	308
6.3.10	Dateien entfernen	308
6.3.11	Ressourcen synchronisieren	309
6.3.12	Dateien markieren	309
6.3.13	Historie anzeigen	309
6.3.14	Ältere Revisionen wiederherstellen	311
6.3.15	Neue Projekte einchecken	312
6.3.16	Dateien ignorieren	314
6.3.17	Binärdateien	317
6.3.18	Dateirevisionen vergleichen	317
6.3.19	Einen Patch erzeugen	318
6.3.20	Projekte entkoppeln	319
6.3.21	Abschließende Bemerkungen	319
6.4	SVN	320
6.4.1	Installation	320
6.4.2	Die Perspektive <i>SVN Repository Exploring</i>	321
6.4.3	Ein neues Repository hinzufügen	321
6.4.4	Mit Repositorys arbeiten	322
6.4.5	Projekte auschecken	322
6.4.6	Dekorationen	324
6.4.7	Neue Projekte einchecken	324

6.4.8	Dateien markieren	325
6.4.9	Historie anzeigen	325
6.4.10	Dateien ignorieren	327
6.4.11	Propertys von Ressourcen	328
6.5	Git	330
6.5.1	Konzepte	330
6.5.2	Installation	331
6.5.3	Die Perspektive <i>Git Repository Exploring</i>	332
6.5.4	Ein entferntes Repository klonen	333
6.5.5	Projekt in den Workspace importieren	334
6.5.6	Mit Projekten unter Git arbeiten	335
6.5.7	Weitere Informationen	338
6.6	Team-Synchronizing-Perspektive	338
6.6.1	<i>Synchronize-View</i>	338
6.6.2	Konflikte	341
7	Mylyn	343
7.1	Mylyn installieren	344
7.2	Einführendes Beispiel	345
7.3	Konzepte	347
7.4	Task Connectors	349
7.4.1	Bugzilla	349
7.4.2	Mit SourceForge verbinden	352
8	Target Management	357
8.1	Das Target-Management-Plugin installieren	357
8.2	Konzepte	358
8.3	Die Perspektive <i>Remote System Explorer</i>	359
8.4	Der <i>Remote Systems-View</i>	360
8.5	Der <i>Remote Monitor-View</i>	361
8.6	Eine neue Verbindung anlegen	361
8.7	Die Verbindung aktivieren	364
8.8	Durch Ressourcen navigieren	364
8.9	Filter und Filterpools erzeugen	365
8.10	Profile verwalten	367
8.11	Ein Projekt auf den entfernten Rechner exportieren	367
8.12	Entfernte Projekte	367
8.13	DataStore	368
8.13.1	DataStore auf Linux einrichten	368
8.13.2	DataStore auf Windows einrichten	369
8.14	Den entfernten Rechner (fern-)steuern	369
8.14.1	Kommandos aufrufen	369
8.14.2	Terminals	370

8.15	Programme auf dem entfernten Rechner ausführen und debuggen	371
9	Linux-Tools	373
9.1	Linux-Tools installieren.....	373
9.2	Autotools	374
9.2.1	Autotools-Projekt erzeugen	374
9.2.2	Editoren für Autotools-Dateien	375
9.2.3	Autotools-Projekte übersetzen.....	375
9.2.4	Autotools manuell aufrufen	376
9.2.5	Globale Voreinstellungen.....	376
9.2.6	Projekteinstellungen	376
9.3	Valgrind-Plugin	377
9.3.1	Hintergründe	377
9.3.2	Erste Schritte	378
9.3.3	Valgrind konfigurieren	381
9.3.4	Weitere Anwendungsbeispiele	382
9.4	Profiling mit GProf	387
9.5	Weitere Plugins	387
	Literaturverzeichnis	391
	Glossar	393
	Index	394