

Inhalt

Einleitung	15
------------------	----

TEIL I Einführung

1 Einführung in das Business Object Processing Framework	19
---	-----------

1.1 Was ist BOPF?	19
1.1.1 Typische Programmieraufgaben	19
1.1.2 Vor- und Nachteile von BOPF	22
1.2 Historische Entwicklung des Frameworks	24
1.3 Die Rolle im Programmiermodell für SAP S/4HANA	25
1.3.1 Das aktuelle Programmiermodell (vor ABAP 7.50)	26
1.3.2 Das neue Programmiermodell (nach ABAP 7.50)	27
1.3.3 Die Innovation für die Zukunft	32

2 Aufbau von Geschäftsobjekten	35
---	-----------

2.1 Knoten	36
2.1.1 Transiente Knoten	38
2.1.2 Transiente Attribute	38
2.1.3 Knotentypen	39
2.2 Knotenelemente	39
2.3 Alternative Schlüssel	40
2.3.1 Eindeutigkeit	43
2.3.2 Anlegen von alternativen Schlüsseln	44
2.3.3 Arbeiten mit alternativen Schlüsseln	44
2.4 Aktionen	44
2.4.1 Kardinalität einer Aktion	45
2.4.2 Rückgabeparameter einer Aktion	45
2.4.3 Anlegen und Implementieren einer Aktion	46
2.4.4 Aufruf einer Aktion	46

2.5	Assoziationen	46
2.5.1	Kompositionen	47
2.5.2	Allgemeine Assoziationen	50
2.5.3	Anlegen und Implementieren von Assoziationen	54
2.5.4	Assoziation nutzen	54
2.6	Ermittlungen	54
2.6.1	Auslösebedingungen und Auswertungszeitpunkte	55
2.6.2	Abhängigkeiten festlegen	57
2.6.3	Implementierung einer Ermittlung	57
2.6.4	Aufruf einer Ermittlung	58
2.7	Validierungen	58
2.7.1	Konsistenz-Validierungen	58
2.7.2	Aktions-Validierungen	59
2.7.3	Implementierung einer Validierung	59
2.7.4	Konsistenzgruppen	60
2.8	Abfragen	61
2.8.1	Kundeneigene Abfragen	62
2.8.2	Implementierung einer Abfrage	62
2.8.3	Aufruf einer Abfrage	63
2.9	Berechtigungsprüfungen	63

3 Testen von Geschäftsobjekten 65

3.1	Instanzen über Abfragen finden	66
3.1.1	Datenbanktabelle verwenden	69
3.1.2	Unterknoten	70
3.2	Eine Instanz ändern	70
3.2.1	Prüfungen ausführen	72
3.2.2	Aktionen ausführen	73
3.3	Eine neue Instanz anlegen	73
3.4	Unterknoten hinzufügen	74
3.5	Löschen von Instanzen	75
3.6	Analyse von Fehlermeldungen	76
3.7	Anzeigen der Änderungshistorie	77
3.8	Break- und Watchpoints setzen	77
3.8.1	Externe Breakpoints	78
3.8.2	BOPF-Watchpoints	79

3.9	Die Trace-Funktion aktivieren	81
3.9.1	Trace aufzeichnen	81
3.9.2	Trace stoppen	82

TEIL II Anwendungsentwicklung mit Geschäftsobjekten

4 Architektur der Consumer-API 87

4.1	Service-Manager	88
4.2	Konstanten-Schnittstelle	90
4.3	Transaktions-Manager	93
4.3.1	Methode SAVE	95
4.3.2	Methode CLEANUP	98
4.3.3	Synchronisierungspunkte	99
4.3.4	Änderungen nachverfolgen	102
4.4	Konfigurations-Manager	102
4.5	Die Factory-Klasse des Frameworks	104
4.6	Änderungsobjekte	105
4.7	Nachrichtenobjekte	110
4.8	Das auslösende Fehlerfeld: die Origin-Location	113

5 Anwendung der Consumer-API 115

5.1	Lesen von Knoteninstanzen	116
5.1.1	Die Methode QUERY	117
5.1.2	Die Methode RETRIEVE	120
5.1.3	Auslesen einer Verknüpfung mit RETRIEVE_BY_ASSOCIATION	121
5.2	Konvertieren von alternativen Schlüsseln	122
5.3	Modifizieren von Objekten	123
5.3.1	Anlegen von Hauptknoten	126
5.3.2	Anlegen von Unterknoten	127
5.3.3	Aktualisieren von Knoteninstanzen	129
5.3.4	Löschen von Knoteninstanzen	130
5.4	Sperrern/Entsperrern	131

5.5	Standardwerte für Knoten auslesen	133
5.5.1	Standardwerte für Oberknoten auslesen	133
5.5.2	Standardwerte für Unterknoten auslesen	134
5.6	Hauptknoten auslesen	135
5.7	Abfragen	137
5.7.1	Weitere Abfrageoptionen	137
5.7.2	Standard-Abfrageparameter	139
5.8	Aktionen	141
5.8.1	Aktionen aufrufen	141
5.8.2	Standardparameter für Aktionen auslesen	143
5.9	Validierungen	144
5.9.1	Konsistenz-Validierungen	144
5.9.2	Aktions-Validierungen	145
5.10	Ermittlungen	146

TEIL III Anlegen von Geschäftsobjekten

6 Die Geschäftsobjekt-API 151

6.1	Grundparameter	154
6.1.1	IO_READ: Lesen von Daten	154
6.1.2	IO_MODIFY: Ändern von Daten	156
6.1.3	IO_CHECK: Prüfen von Daten	157
6.1.4	EO_MESSAGE: Rückgabe von Nachrichten	157
6.1.5	IS_CTX: Informationen zum Aufrufkontext	159
6.2	Knoten implementieren	161
6.3	Aktionen implementieren	162
6.4	Ermittlungen implementieren	164
6.5	Validierungen implementieren	166
6.6	Abfragen implementieren	169
6.6.1	Die Methode QUERY	170
6.6.2	Die Rückgabetable ET_DATA	172
6.7	Assoziationen implementieren	172
6.8	Ausnahmen implementieren	173

7 Eigene Geschäftsobjekte mit der SAP GUI anlegen 177

7.1	Geschäftsobjekt anlegen	179
7.1.1	Persistente Struktur anlegen	184
7.1.2	Transiente Struktur anlegen	185
7.1.3	Anlage abschließen	186
7.2	Alternative Schlüssel für die SAP GUI	187
7.2.1	Alternativen Schlüssel hinzufügen	188
7.2.2	Index anlegen	190
7.3	Unterknoten hinzufügen	192
7.3.1	Standardknoten als Unterknoten anlegen	192
7.3.2	Darstellungsknoten für Geschäftsobjekte anlegen	195
7.4	Assoziationen	197
7.5	Abfragen	199
7.5.1	Eine SELECT_ALL-Abfrage anlegen	200
7.5.2	Eine SELECT_BY_ELEMENTS-Abfrage anlegen	200
7.5.3	Eine eigene Abfrage anlegen	201
7.6	Aktionen	204
7.7	Validierungen	207
7.7.1	Eindeutigkeitsprüfung für alternative Schlüssel anlegen	208
7.7.2	Konsistenz-Prüfung anlegen	209
7.7.3	Aktions-Prüfung anlegen	212
7.8	Ermittlungen	214
7.9	Autorisierungen	217
7.9.1	Ein eigenes Berechtigungsfeld anlegen	218
7.9.2	Ein eigenes Berechtigungsobjekt anlegen	219
7.9.3	Autorisierung im Geschäftsobjekt einrichten	221
7.9.4	Berechtigung an einen Nutzer vergeben	223

8 Eigene Geschäftsobjekte mit Eclipse ADT anlegen 229

8.1	Grundlegende Anlage	229
8.1.1	Persistente Struktur anlegen	235
8.1.2	Transiente Struktur anlegen	238

8.2	Alternative Schlüssel	238
8.3	Unterknoten	241
8.4	Assoziationen	243
8.5	Abfragen	246
8.5.1	Anlegen einer SELECT_BY_ELEMENTS-Abfrage	246
8.5.2	Anlegen einer eigenen Abfrage	247
8.6	Aktionen	249
8.7	Validierungen	252
8.7.1	Anlegen einer Konsistenz-Prüfung	253
8.7.2	Anlegen einer Aktions-Prüfung	255
8.8	Ermittlungen	258
8.9	Autorisierungen	261

TEIL IV Fortgeschrittene Techniken

9	Bestehende Geschäftsobjekte erweitern	265
9.1	Möglichkeiten der klassischen Erweiterung	266
9.1.1	Knotenattribute erweitern	266
9.1.2	Erweiterung einer transienten Struktur	269
9.1.3	Erweiterung der Logik in implementierenden Klassen	269
9.2	Anlegen eines Erweiterungsobjekts	271
9.3	Zusätzlichen Unterknoten anlegen	273
9.3.1	Standard-Unterknoten hinzufügen	274
9.3.2	Darstellungsknoten für Geschäftsobjekte hinzufügen	276
9.4	Knotenfunktionen erweitern	277
9.4.1	Alternative Schlüssel	278
9.4.2	Assoziationen	278
9.4.3	Abfragen	279
9.4.4	Aktionen	280
9.4.5	Validierungen	282
9.4.6	Ermittlungen	283

10

Debugging von Geschäftsobjekten

287

10.1	Laufzeitfehler mit MESSAGE_TYPE_X_TEXT	288
10.1.1	Laufzeitfehler auslesen	290
10.1.2	Über die Ausnahmeklassen	292
10.1.3	Im Aufrufstack navigieren	297
10.2	Puffer debuggen	298
10.2.1	Über Vorwärtsnavigation im Debugger debuggen	299
10.2.2	Mit Debugging-Skript debuggen	302
10.3	Nachrichten von EO_MESSAGE	304
10.4	Change-Log von EO_CHANGE	307
10.5	Eigenschaften von EO_PROPERTY	308
10.6	Änderungstabelle IT_MODIFICATION	308

11

Wiederverwendung

311

11.1	Validierungen	312
11.1.1	Wertprüfung auf Domänen	312
11.1.2	Alternative Schlüssel validieren	314
11.2	Ermittlungen	315
11.2.1	Administrationsdaten	315
11.2.2	Nummernkreise	316

12

**Eine SAPUI5-App mit CDS und BOPF
als Grundlage entwickeln**

319

12.1	CDS: die Modelldefinition	320
12.1.1	Was ist CDS?	321
12.1.2	Interface-Views mit ObjectModel-Annotationen	322
12.1.3	Consumption-Views mit UI und OData Annotations	327
12.2	BOPF: die Backend-Implementierung	331
12.2.1	BOPF-Geschäftsobjekt generieren	331
12.2.2	Arbeiten mit generierten Geschäftsobjekten	335

12.3	OData: die Schnittstelle zur Welt	335
12.3.1	Was ist OData?	336
12.3.2	Was ist SADL?	336
12.3.3	OData-Service generieren	336
12.3.4	Arbeiten mit generierten OData-Services	338
12.4	SAP Fiori: die Oberfläche	339
12.4.1	Was ist SAP Fiori?	339
12.4.2	Was ist SAPUI5?	339
12.4.3	Generieren einer SAP-Fiori-App	340
12.4.4	Arbeiten mit generierten SAP-Fiori-Apps	344

Anhang	345
---------------	-----

A	Klassendiagramme zur Consumer-API	347
B	Einrichten der BOPF-Debugging-Skripte	351
C	Funktionsumfang der Entwicklungsumgebungen	355
D	Die Autoren	359

Index	361
-------	-----