

Inhalt

1	Einleitung	9
2	Erste Schritte	13
2.1	Customizing der Workflow Engine	13
2.1.1	Customizing durchführen (Transaktion SWU3)	13
2.1.2	Systembenutzer WF-BATCH	15
2.1.3	Die logische RFC-Destination »WORKFLOW_LOCAL_xxx«	16
2.1.4	Customizing überprüfen (Transaktionen SWU3 und SWUI_VERIFY)	17
2.2	Workflows starten und Workflow-Ereignisse beobachten	21
2.2.1	Ereignisse beobachten (Transaktionen SWELS und SWEL)	22
2.2.2	Workflow-Definitionen betrachten (Transaktion SWDS)	25
2.3	Eine minimale Organisationsstruktur pflegen	30
2.3.1	Organisationseinheiten anlegen (Transaktion PPOCW)	31
2.3.2	Organisationseinheiten bearbeiten (Transaktion PPOMW)	33
2.3.3	SAP-Benutzer zu Planstellen zuordnen	35
2.3.4	Bearbeiterfindung testen (Transaktion PFAC)	37
3	Workflow-Entwicklungsumgebung zusammenstellen	41
3.1	Relevante Transaktionen	41
3.2	Workflow-Entwicklungsvorgang mit SAP-Standardfunktionalität	42
3.3	Ereignisse	43
3.3.1	Ereignistypkopplung	44
3.3.2	Ereignisinstanzkopplung	45
3.4	Verwendungsnachweis rund um SAP Business Workflow	46
3.4.1	Vom Objekt über die Standardaufgabe zum Workflow-Muster	46
3.4.2	Vom BOR-Objekt zur Standardaufgabe (klassisch)	47

3.4.3	Von der ABAP-Objects-Klasse zur Standardaufgabe (klassisch)	48
3.4.4	Von der Standardaufgabe zum Workflow-Muster	49

4 Methoden, Workitems und Ereignisse 51

4.1	Arten von Methoden im Workflow	51
4.2	Arten und Status von Workitems	54
4.3	Ereignisse und deren Zustellung	57
4.3.1	Check- und Verbrauchertyp-Funktionsbausteine.....	59
4.3.2	Ereignis-Queue	61

5 In die Bearbeiterfindung eingreifen 63

5.1	Workflow erstellen	63
5.2	Bearbeiterfindung dynamisch festlegen und auswählen	64
5.3	Workflow-Muster testen	68
5.4	Bearbeiterfindung mit Aufgabengruppen modellieren	71
5.5	Bearbeiterfindung mit Zuständigkeiten	77
5.5.1	Regel basierend auf Zuständigkeiten anlegen	78
5.5.2	Regel in das Workflow-Muster einbinden	82
5.6	Bearbeiterfindung programmieren	84
5.6.1	Klassischen Funktionsbaustein zur Bearbeiterfindung erstellen	84
5.6.2	ABAP-Klasse zur Bearbeiterfindung erstellen	90

6 Container, Datenfluss und Bedingungen im Workflow 93

6.1	Vorbereitung	93
6.2	Container – wo befinden sich die Daten, die ein Workflow benutzt?	94
6.2.1	Ereigniscontainer	96
6.2.2	Workflow-Container	96
6.2.3	Regelcontainer	97
6.2.4	Aufgabencontainer	97
6.2.5	Methodencontainer	98
6.3	ABAP-Coding mit Containern (Makros)	98
6.4	ABAP-Objects-Klassen zum Umgang mit Containern, Datenflüssen und Bedingungen	101

6.5	Codingbeispiel zu Containern, Datenflüssen und Bedingungen	103
6.5.1	Containerelemente erzeugen	103
6.5.2	Container erzeugen	105
6.5.3	Container konvertieren	107
6.5.4	Workflow-Bedingungen erstellen	107
6.5.5	Datenfluss definieren	110
6.6	Erweiterte Funktionalität im Datenfluss	113
6.7	Programmierter Datenfluss	114

7 Beispielprojekt – ABAP-Objects-Klasse für den Workflow gestalten 117

7.1	Ausgangssituation	117
7.2	Besonderheiten im Workflow-Umfeld	119
7.3	Klasse anlegen, »IF_WORKFLOW« einbinden und Schlüsselattribute definieren	121
7.4	Instanzverwaltung und -erzeugung	123
7.5	Die kleine Persistenz für »zwischen durch«	127
7.6	Fehlerbehandlung mit Ausnahmeklassen	132
7.7	Workflow-Ereignisse aus ABAP-Objects-Klassen erzeugen	134
7.8	BOR-Objekte als Attribute in ABAP-Objects-Klassen	137

8 Beispielprojekt – BOR-Objekt für den Workflow gestalten 141

8.1	Ausgangssituation	141
8.2	Neues BOR-Objekt erzeugen	143
8.3	Persistenz für BOR-Attribute anlegen	144
8.4	Schlüssel- und andere Attribute anlegen	145
8.5	BOR-Methoden anlegen	154
8.5.1	Weitere BOR-Interfaces	155
8.5.2	Methode »CREATE« überdefinieren	155
8.5.3	Methodencontainer für Parameter	156
8.5.4	Methode »DELETE« überdefinieren	159
8.6	Ausnahmen und Fehler	160
8.7	BOR-Ereignisse	161
8.8	BOR-Freigabestatus	163
8.9	BOR-Default-Vorgaben	165
8.10	Vererbung und Delegation im BOR	166

Anhang	169
A Schrittypen und Beispiel-Workflows	171
A.1 Schrittypen	171
A.2 Weitere Workflow-Techniken	175
B Wichtige Transaktionen	177
C Der Autor	181
Index	183