

Werner Hertleif
Christoph Wachter

SAP Smart Forms

Formulare komfortabel erstellen



Galileo Press 

Inhalt

1 Einleitung 13

1.1	Formularerstellung im neuen Stil	13
1.2	Handhabung des Buches	14
1.2.1	Buchaufbau	14
1.2.2	Der richtige Einstieg	17
1.2.3	Generelle Hinweise	17

2 Schnelleinstieg 19

2.1	Generelles zur Formularentwicklung	19
2.1.1	Übersicht	19
2.1.2	Layout des Formulars	20
2.1.3	Ablauflogik des Formulars	21
2.1.4	Daten im Formular	23
2.1.5	Ausgabe	24
2.1.6	Werkzeuge	24
2.1.7	Weitere Schritte	25
2.2	Übungsbeispiel	25
2.2.1	Voraussetzungen	25
2.2.2	Vorbereitung: Standardtext erstellen	27
2.2.3	Ausgabe Musterformular	30
2.2.4	Eigenes Formular als Arbeitskopie erstellen	36
2.2.5	Änderungen am eigenen Formular	38
2.2.6	Eigenes Rahmenprogramm erstellen	47
2.3	Knotentypen in der Übersicht	50

3 Werkzeuge 55

3.1	Übersicht	55
3.2	Style Builder	57
3.2.1	Übersicht	57
3.2.2	Funktionen des Style Builders	58
3.2.3	Eigenen Stil durch Kopie erstellen	59
3.2.4	Kopfdaten eines Stils	61
3.2.5	Absatzformate pflegen	62
3.2.6	Übungsbeispiel: Gliederungen	66

3.2.7	Zeichenformate	68
3.2.8	Übungsbeispiele: Inhalte des Stils	70
3.3	Form Builder	71
3.3.1	Bearbeitungsfunktionen	71
3.3.2	Form Painter	79
3.3.3	Table Painter	81
3.4	Formulare prüfen, testen, aktivieren	84
3.4.1	Formular aktivieren	85
3.4.2	Formular testen	87
3.4.3	Formular generieren	88
3.5	Formular-Dokumentation	88

4 Layout des Formulars 91

4.1	Übersicht	91
4.2	Generelle Layoutvorgaben	92
4.2.1	Globale Einstellungen	93
4.2.2	Rahmen und Schattierungen	94
4.3	Seite-Knoten	96
4.3.1	Neuanlage von Entwurfsseiten	97
4.4	Fenster-Knoten	99
4.4.1	Neuanlage eines Fensters	100
4.4.2	Übungsbeispiel: Fenster erzeugen	102
4.5	Schablonen und Ausgabetabellen	103
4.5.1	Übersicht	103
4.5.2	Übungsbeispiel: Schablone-Knoten einrichten	106
4.5.3	Mehrere Schablonen kombinieren	112
4.5.4	Ausgabetablelle ohne direkten Datenzugriff	112

5 Elementare Knotentypen 115

5.1	Text-Knoten	115
5.1.1	Allgemeine Eigenschaften	115
5.1.2	Textelement	119
5.1.3	Übungsbeispiel: Text-Knoten anlegen	120
5.1.4	Include-Text (SAPscript-Texte)	121
5.1.5	Textbausteine einbinden	125
5.1.6	Übungsbeispiel: Textbaustein einbinden	126
5.1.7	Textbausteine anlegen und pflegen	128
5.1.8	Übungsbeispiel: Textbaustein anlegen	129
5.2	Adresse-Knoten	130
5.2.1	Adresse-Knoten anlegen	131

5.3	Grafiken und Hintergrundbild	135
5.3.1	Einbindung in das Formular	135
5.3.2	Grafiken in Fenster, Schablone oder Ausgabetabelle	137
5.3.3	Hintergrundbild zur Seite	137
5.3.4	Optimierung der Druckausgabe	139
5.4	Ordner-Knoten	139
5.4.1	Übungsbeispiel: Ordner anlegen	141
5.5	Sprache im Formular	141
5.5.1	Sprache bei Formularausgabe	142

6 Daten im Formular 143

6.1	Übersicht	143
6.2	Felder als Platzhalter	145
6.2.1	Feldliste	145
6.2.2	Felder im Text-Knoten	145
6.2.3	Felder als Knotenattribut	148
6.2.4	Felder mit Aufbereitungsoptionen	149
6.3	Datenstrukturen und Datendefinitionen	152
6.3.1	Vorgehensweise	152
6.3.2	Übersicht Datenstrukturen	153
6.3.3	Datendefinition	159
6.3.4	ABAP-Datentypen verwenden	160
6.3.5	Bezug auf ABAP-Dictionary	161
6.3.6	Bezug auf interne Daten	166
6.3.7	Individuelle Datentypen im Formular	167
6.3.8	Syntax der Felder	168
6.4	Quelle der Daten	168
6.4.1	Formularschnittstelle	169
6.4.2	Globale Definitionen	177
6.4.3	Systemfelder	179
6.5	Ausgabe von Feldern mit Währungs- oder Mengenbezug	180

7 Ablauflogik des Formulars 185

7.1	Übersicht	185
7.2	Dynamische Datenausgabe	188
7.2.1	Übersicht	188
7.2.2	Knotentypen zur Ausgabe von internen Tabellen	190
7.2.3	Schleife	192
7.2.4	Einzelnen Datensatz über Schleife lesen	198
7.2.5	Übungsbeispiel: Schleife	200
7.2.6	Ausgabetabelle	201
7.2.7	Zeitpunkte (Ereignisknoten)	204
7.2.8	Übungsbeispiel: Flugrechnung überarbeiten	205

7.3	Logische Abfragen	210
7.3.1	Bedingungen	210
7.3.2	Verzweigungen über Alternative-Knoten	213
7.4	Abfolge und Nummerierung der Ausgabeseiten	214
7.5	Kommando-Knoten	217
7.5.1	Übersicht	217
7.5.2	Print-Controls	219
7.5.3	Attribute zum Spoolauftrag	221
7.6	Komplexer Abschnitt	222
7.7	Nachprozessierung	223
7.8	Zusammenfassung der Formularprozessierung	225

8 ABAP-Programme im Formular 229

8.1	Wozu Programmcode im Formular?	229
8.2	Bearbeitungsfunktionen im Programm-Knoten	232
8.3	Besonderheiten im Programm-Knoten	236
8.4	ABAP-Grundlagen	238
8.4.1	Übersicht	238
8.4.2	Datendefinition	240
8.4.3	Wertzuweisungen	241
8.4.4	Datenbanktabellen abfragen	243
8.4.5	Schleifen	245
8.4.6	Kontrollstrukturen (über Bedingungen)	245
8.4.7	Unterprogramme	246
8.4.8	Testfunktion	248
8.5	Beispiel: Adresse über Funktionsbaustein einbinden	248
8.5.1	Übersicht	248
8.5.2	Verwendung einer Adressnummer aus der ZAV	250
8.5.3	Einbindung bei direkter Übergabe von Adressinhalten	251
8.5.4	Übungsbeispiel	253
8.6	Beispiel: Summenbildung in der Flugrechnung	254
8.6.1	Übungsvorschlag: Ausgabe in Hauswährung	257

9 Rahmenprogramm 259

9.1	Übersicht	259
9.2	Festlegung der Merkmale zur Datenbereitstellung	261
9.2.1	Flugrechnung: Selektionsbild zur Dateneingabe	261
9.2.2	Abwicklung über Nachrichtensteuerung	264
9.3	Datenbeschaffung	268
9.3.1	Flugrechnung: Lesen der Datenbanktabellen	268
9.3.2	Lieferung: Daten zusammenstellen	270

9.4	Funktionsbaustein des Formulars ansprechen	271
9.4.1	Name des Funktionsbausteins ermitteln	271
9.4.2	Aufruf Funktionsbaustein zum Formular	272
9.5	Übungsbeispiel: Flugrechnung an mehrere Kunden	274
9.6	Welche Daten verwendet das Formular?	280
9.6.1	Übersicht	280
9.6.2	Beispiel: Übergabe Flugverbindungen je nach Bedarf	283
9.6.3	Reduzierte Datenbereitstellung beim Lieferschein	283
9.6.4	Beispiel: Kennzeichnung von Durchschlägen	286
9.7	Standardparameter der Formularschnittstelle	290
9.7.1	Übersicht	290
9.7.2	Import-Standardparameter	291
9.7.3	Übungsbeispiel: Import-Standardparameter	297
9.7.4	Export-Standardparameter	301
9.7.5	Übungsbeispiel: Export-Standardparameter	303
9.8	Laufzeitfehler/Ausnahmen	305
9.8.1	Übersicht	305
9.8.2	Fehlerbehandlung im Rahmenprogramm	308
9.8.3	Individuelle Fehlerbehandlung im Formular	311
9.8.4	Überwachung der Formularausführung per TRACE	317

10 Besondere Ausgabeverfahren 319

10.1	Versenden über E-Mail oder Telefax	319
10.1.1	Übersicht	319
10.1.2	Übungsbeispiel: E-Mail-Versand	322
10.1.3	Zusätzliche Möglichkeiten bei DEVICE='MAIL'	325
10.1.4	Anwendungsfall: mySAP CRM	326
10.2	Versenden per Telefax	327
10.3	XSF-Ausgabe	329
10.3.1	Übersicht	329
10.3.2	Übergabeformen der XSF-Ausgabe	330
10.3.3	XSF-Ausgabe statisch aktivieren	331
10.3.4	XSF-Ausgabe dynamisch über Formularschnittstelle vorgeben	332
10.3.5	Übungsbeispiel: Download XSF-Ausgabe	333
10.4	Archivierung	337
10.5	Etikettendruck	338
10.5.1	Übersicht	338
10.5.2	Basisformular anlegen und einbinden	340
10.5.3	Einrichten Etikettendruck im Formular	341
10.5.4	Ergebnis der Formularausgabe	347
10.5.5	Vorschlag für den Ausbau des Formulars	348

11 SAPscript-Formulare migrieren 349

11.1	Übersicht	349
11.2	Stile migrieren	351
11.3	Formulare migrieren	352
11.3.1	Automatische Migrationsfunktion	352
11.3.2	Manuelle Nachbearbeitung	354
11.3.3	Testfähigkeit des Formulars erreichen	355
11.3.4	SAPscript-Kommandos umsetzen	358
11.3.5	Ausgabefähigkeit über Rahmenprogramm erzeugen	362
11.3.6	Übergeordnete Ablauflogik	363
11.3.7	Standard-Schnittstellenparameter	368

12 Neuerungen ab Basis-Release 6.10 373

12.1	SAP Web Application Server 6.10	373
12.1.1	Neuerungen im Form Builder, Form Painter und Table Painter ..	374
12.1.2	Neue Tabellenausgabe	376
12.1.3	Das Kopienfenster	377
12.1.4	Das Abschlussfenster	378
12.1.5	Neue Systemfelder	379
12.2	Web-Formulare für Internet-Anwendungen	379
12.2.1	Die HTML-Ausgabe	381
12.2.2	Verwendung eines Web-Formulars	381
12.2.3	Eingabeelemente	382
12.2.4	Mehr Details zu HTML-Formularen	384
12.2.5	Aufruf eines Web-Formulars mit einer BSP-Applikation	386
12.2.6	Formulareingaben auswerten	389
12.2.7	Web-Eigenschaften im Form Builder pflegen	390
12.2.8	Einfache Eingabetypen verwenden	391
12.2.9	Gruppierende Eingabetypen verwenden	392
12.3	SAP Web Application Server 6.20	393
12.3.1	Seitenschutz für Tabellenzeilen	393
12.3.2	Hoch- und Herunterladen von Textbausteinen und Smart Styles	394
12.3.3	Web-Formulare	394

13 Umfeld Formularentwicklung 397

13.1	Übersicht	397
13.2	Transport und Übersetzung	397
13.2.1	Transport	398
13.2.2	Übersetzung von Texten	399
13.2.3	Sprache der Formularbearbeitung	402

13.3	Textbearbeitung und -ablage	403
13.3.1	Übersicht	403
13.3.2	Inline-Editor	403
13.3.3	SAPscript-Editor (Ganzseiteneditor)	407
13.3.4	Sonderzeichen im Text	409
13.3.5	Textverwaltung	415
13.4	Grafikverwaltung	420
13.5	Nachrichtenfindung und -steuerung	423
13.5.1	Übersicht	423
13.5.2	Nachrichtenfindung über Konditionstechnik	424
13.5.3	Formulare des Finanzwesens einbinden	434
13.5.4	QM-Formulare einbinden	435
13.6	Druck- und Spoolsystem	437
13.6.1	Übersicht	437
13.6.2	Vorgaben im Spooldialogbild	439
13.6.3	Spoolsteuerung (Ausgabesteuerung)	441
13.6.4	Spoolverwaltung	445
13.6.5	Gerätetypen (Print-Controls)	449
13.6.6	Fonts und Barcodes im SAP-System	452
13.7	Entwicklungswerkzeuge	457
13.7.1	Übersicht (Object Navigator)	457
13.7.2	ABAP-Editor	460
13.7.3	Function Builder	464
13.7.4	ABAP-Debugger	470
13.7.5	ABAP-Dictionary	480

14 Preconfigured Smart Forms 487

14.1	Überblick – Preconfigured Smart Forms	487
14.2	Verfügbare Formulare, Details und Hinweise zur Konfiguration	487
14.3	Tipps und Tricks	496

A Anhang 497

A.1	Hinweise im SAPnet (SAPnotes/OSS-Meldungen)	497
A.2	Das SAP-Flugdatenmodell	497
A.3	Beispielformulare zur Flugrechnung	502
A.3.1	Ausgelieferte Musterformulare	502
A.3.2	Musterausdruck zur Flugrechnung (SF_EXAMPLE_01)	502
A.3.3	Quelltexte zum Rahmenprogramm (Original)	504
A.3.4	Musterausdruck zur Flugrechnung nach Änderungen	506
A.3.5	Quelltext zur Flugrechnung nach den Übungsbeispielen	507

A.4	Beispiel zum E-Mail-Versand über Smart Forms	510
A.4.1	Quelltext	510
A.4.2	Programmerstellung	516
A.5	Musterformulare Automotive	519
A.5.1	Übersicht	519
A.5.2	Warenanhänger nach VDA 4902 (KLT-Label)	520
A.5.3	Lieferschein nach DIN 4994/4992	520
A.5.4	Frachtauftrag nach VDA 4992	522

Index

523

7 Ablauflogik des Formulars

7.1 Übersicht

In den vorhergehenden Kapiteln haben wir erläutert, wie Sie das Layout einer Seite entwerfen können und wie dort Texte, Grafiken und auch Daten über entsprechende Knoten eingebunden werden.

Wir haben uns aber bisher nur kurz damit befasst, in welcher Reihenfolge die Knoten bei der Ausgabe abgearbeitet (*prozessiert*) werden, wie Ausgaben über mehrere Seiten erfolgen und wie man diese Abläufe steuern kann. Mit diesen Funktionen befinden Sie sich in der Ablauflogik des Formulars, die wir kurz als *Formularlogik* bezeichnen wollen.

Bevor wir die vielfältigen Möglichkeiten zur Gestaltung der Formularlogik detailliert betrachten, möchten wir Ihnen zunächst eine kurze Zusammenfassung der Grundlagen geben, auf die wir im Zusammenhang der Formularlogik treffen.

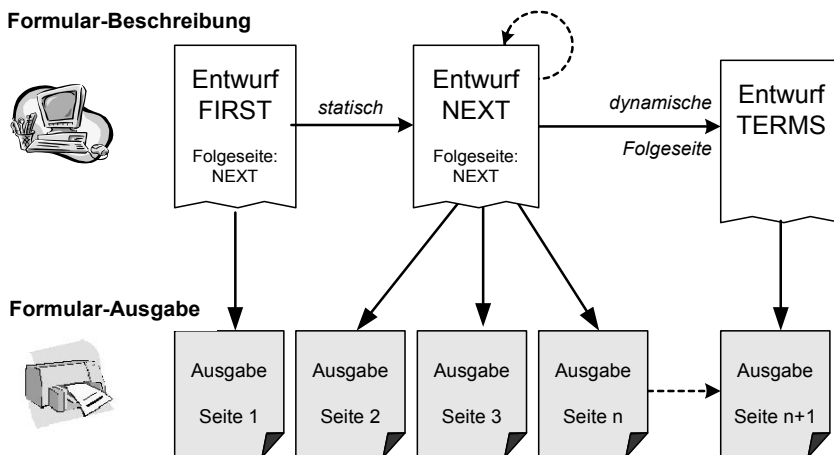


Abbildung 7.1 Abarbeitung von Seiten in der Formularlogik

Jedes Formular besteht aus einer oder mehreren Entwurfsseiten (siehe Abbildung 7.1). Die erste Seite im Navigationsbaum ist die Startseite. Üblicherweise wird dafür der Name FIRST gewählt, dieses Kürzel verwenden wir gelegentlich auch hier im Buch. Mit ihr beginnt bei der Ausgabe die Abarbeitung (*Prozessierung*) des Formulars. Die Prozessierung innerhalb dieser Seite erfolgt in der Reihenfolge, in der die zugeordneten Knoten im Navigationsbaum des Form Builders angelegt sind.

Wenn eine Ausgabeseite gefüllt ist, muss die Ausgabesteuerung entscheiden, ob noch weitere Seiten erzeugt werden müssen. Diese Entscheidung wird danach getroffen, ob bereits alle Texte und/oder Daten ausgegeben worden sind oder nicht. Welches Seitenlayout bei der neuen Ausgabeseite verwendet wird, entscheidet der Eintrag zur Folgeseite bei den Attributen der bisherigen Entwurfsseite (die übliche Kurzbezeichnung der Folgeseite ist NEXT). Wegen der festen Zuordnung dieser Folgeseite spricht man auch von einer *statischen* Abwicklung.

Alternativ kann sich die Folgeseite aber auch *dynamisch* über einen manuellen Seitenumbruch ergeben (d.h. aufgrund individueller Bedingungen zur Laufzeit). In Abbildung 7.1 ist dies der Fall, um beispielhaft eine abschließende Seite TERMS mit Angabe von Geschäftsbedingungen auszugeben. Der dynamische Seitenumbruch erfolgt, sobald alle Datensätze abgearbeitet sind.

Hinweis: Wenn wir von der Ausgabe des Formulars sprechen, ist damit die Erzeugung eines Zwischendokuments gemeint, das alle notwendigen Anweisungen enthält, um die weitere Abwicklung über das SAP-Spoolssystem zu ermöglichen. Derjenige Programmteil innerhalb der Laufzeitumgebung von Smart Forms, der die Umsetzung der Formularelemente in dieses Zwischendokument vornimmt, wird auch als *Formular-Prozessor* oder *Composer* bezeichnet.

Haupt- und Nebenfenster

Jede Entwurfsseite enthält in ihren Layoutangaben ein oder mehrere Fenster. In den Fenstern befinden sich die Texte und Daten, wobei alle Angaben zur Formatierung der Ausgabe einem hinterlegten Stil entnommen werden.

Maximal ein Fenster im Formular kann als *Hauptfenster* gekennzeichnet werden. Nur in diesem Fenster kann das System die Inhalte der Ausgabe darauf überwachen, ob sie auf die aktuelle Ausgabeseite passen oder ob eine neue Seite eröffnet werden muss. Wird für die Folgeseite ein anderes Layout verwendet, muss auch das Hauptfenster in diesem Layout vorhanden sein (damit die Ausgabe der Inhalte in diesem Fenster fortgesetzt werden kann).

Alle anderen Fenster werden als *Nebenfenster* bezeichnet und haben keinen direkten Einfluss auf den Seitenumbruch. Trotzdem können sich auch Nebenfenster auf verschiedenen Entwurfsseiten wiederholen. Auf jeder neuen Ausgabeseite wird auch der Inhalt des Fensters neu begonnen (und nicht etwa von einer vorherigen Seite fortgesetzt).

Beim Hauptfenster kann also der Inhalt auf mehrere Seiten verteilt sein (Fließtext und Daten mit automatischem Seitenumbruch); bei Nebenfenstern wird der Inhalt jeweils wiederholt.

Abarbeitung einer Entwurfsseite

Die Prozessierung einer Seite (und damit der enthaltenen Knoten) erfolgt in der Reihenfolge, in der die Knoten im Navigationsbaum angelegt sind. Um eine Vorstellung von dieser Reihenfolge zu bekommen, kann es vorteilhaft sein, den Navigationsbaum der Seite komplett aufzuklappen. Dann stehen die einzelnen Knoten in der verwendeten Reihenfolge (Abbildung 7.2 zeigt einen Ausschnitt).



Abbildung 7.2 Abarbeitung des Formulars bei Ausgabe

Die Reihenfolge der Abarbeitung steht nicht im Zusammenhang mit der Position, die ein Knoten bei Ausgabe auf der Seite hat. Einzige Ausnahme ist die relative Positionierung z. B. von Text- oder Grafik-Knoten. Dann hängt die jeweilige Position von den vorherigen Ausgabeinhalten ab.

Die Reihenfolge der Abarbeitung müssen Sie insbesondere auch dann beachten, wenn mit Feldern gearbeitet wird. Stellen Sie sicher, dass die erforderlichen Daten einem Knoten zugewiesen werden, der im Navigationsbaum oberhalb desjenigen Knotens steht, durch den die Ausgabe der Daten erfolgt.

Durch Verschieben einzelner Knoten im Navigationsbaum (oder auch ganzer Zweige) können Sie deren Reihenfolge im Navigationsbaum ändern; verwenden Sie dafür vorzugsweise die Mausfunktionen des Form Builders (Kontextmenü, Drag&Drop). Mit etwas Übung werden Sie auch das notwendige Fingerspitzengefühl für die Angabe des Ziels besitzen. In Einzelfällen fragt das System nach, ob ein Knoten an einer bestimmten Stelle als Nachfolger- oder Unterknoten eingefügt werden soll.

Individuelle Formularlogik

Sie haben darüber hinaus verschiedene Möglichkeiten, um die Prozessierung des Formulars individuell zu beeinflussen. Verwenden Sie dazu die generellen Knotenattribute oder spezielle Knotentypen, auf die wir im Folgenden ausführlicher eingehen wollen:

- ▶ Verwenden Sie Schleife-Knoten, um Ausgaben beliebig oft zu wiederholen. Damit wird die Ausgabe von Daten aus internen Tabellen komfortabel möglich. Um die Ausgabe dabei gleichzeitig tabellarisch zu formatieren, steht der Tabelle-Knoten zur Verfügung. Die Datenausgabe über solche Schleifen in einem Hauptfenster ist der gängigste Weg, Ausgaben mit automatischem Seitenumbruch zu erzeugen.
- ▶ Zu jedem Formular-Knoten können Sie Bedingungen hinterlegen. Trifft eine solche Bedingung bei Ausgabe des Formulars nicht zu, wird der jeweilige Knoten auf der aktuellen Ausgabeseite nicht abgearbeitet (einschließlich aller Unterknoten).
- ▶ Verwenden Sie Alternative-Knoten, wenn sich die Prozessierung zur Laufzeit zwischen verschiedenen Zweigen im Navigationsbaum (und deren Unterknoten) entscheiden soll.
- ▶ Über einen Kommando-Knoten und seine Bedingungen stellen Sie ein, wann ein dynamischer Seitenumbruch mit individueller Folgeseite erfolgen soll.

In den folgenden Kapiteln werden wir diese und weitere Möglichkeiten zur Beeinflussung der Formularlogik im Detail vorstellen. Wir beginnen im ersten Schritt mit Schleife- und Tabelle-Knoten. Man spricht in diesem Zusammenhang auch von *dynamischer* Ausgabe, weil die Anzahl der Datensätze (in unserer Flugrechnung z. B. die Anzahl der Rechnungspositionen) und damit die Länge der Ausgabe zum Zeitpunkt des Formularentwurfs nicht bekannt ist. Dass die Zahl der Datensätze nicht bekannt ist, stellt den zentrale Unterschied zur Ausgabe über Schablonen dar. Bei der dynamischen Ausgabe muss ein Seitenumbruch zur Laufzeit von der Ausgabesteuerung selbst vorgenommen werden.

7.2 Dynamische Datenausgabe

7.2.1 Übersicht

Bei Ausgabe eines Formulars werden üblicherweise im Datenbeschaffungsteil des Rahmenprogramms die erforderlichen Daten aus Datenbanktabellen gelesen, in interne Tabellen übertragen und dann per Formularschnittstelle an das Formular übergeben. In Sonderfällen kann die Beschaffung von Daten auch über einen individuellen Programm-Knoten im Formular erfolgen (darauf kommen wir noch zurück).

Diese internen Tabellen bestehen aus Angaben zu Spalten mit den eigentlichen Feldnamen und einer beliebigen Zahl von Zeilen (ähnlich wie in einem Tabellenkalkulationsblatt). Diese internen Tabellen können **nicht** wie die Inhalte von Variablen oder Feldleisten direkt in einen Text-Knoten platziert werden. Die Ausgabe muss statt dessen zeilenweise unter Verwendung von Arbeitsbereichen erfolgen.

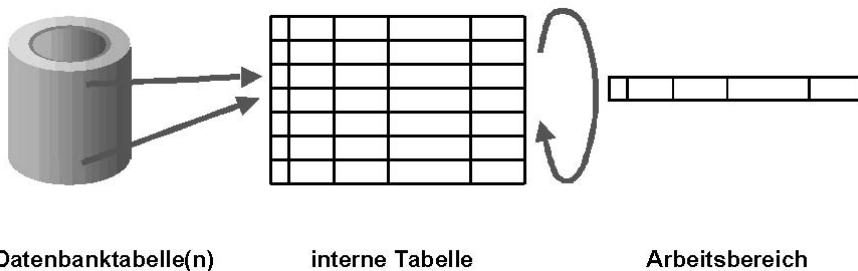


Abbildung 7.3 Schleife über interne Tabelle

Die Abarbeitung der internen Tabellen erfolgt in folgenden Stufen:

- ▶ Jede Zeile der internen Tabelle wird einmal gelesen und dabei in eine Feldleiste mit gleichem Spaltenaufbau kopiert, die wir als *Arbeitsbereich* bezeichnen (üblich ist teilweise auch die Bezeichnung *Ausgabebereich*).
- ▶ Nach jedem Lesevorgang können die benötigten Felder des Arbeitsbereichs in der gewünschten Form ausgegeben werden.
- ▶ Nach Abarbeitung aller Ausgaben wechselt das System automatisch zur nächsten Zeile der internen Tabelle.

Dieser Durchlauf über alle Einträge einer internen Tabelle wird als *Schleife* bezeichnet. Verwendet wird für diese Funktion unter Smart Forms ein Schleife-Knoten oder alternativ ein Tabelle-Knoten, wie wir noch sehen werden. Für den Durchlauf stehen zwei wichtige Optionen zur Verfügung:

- ▶ Falls nicht alle Datensätze der internen Tabelle für die Ausgabe relevant sein sollten, können Sie Selektionsbedingungen hinterlegen. Es werden dann nur diejenigen Zeilen gelesen und kopiert, die den Bedingungen entsprechen (bei den Knotenattributen als WHERE-Bedingungen bezeichnet). Als Sonderfall können Sie solche Bedingungen auch nutzen, um nur einen einzelnen Datensatz zu lesen.
- ▶ Für die Festlegung einer gewünschten Ausgabereihenfolge können Sie die interne Tabelle mit Sortierkriterien versehen. Bei einer sortierten Ausgabe haben Sie zusätzlich die Möglichkeit, Zwischenwerte je Sortierstufe einzufügen.

Wenn Sie eine interne Tabelle über eine solche Schleife ausgeben, ist üblicherweise nicht bekannt, wie viele Datensätze enthalten sind. Deshalb sollten Sie interne Tabellen in der Regel in einem Hauptfenster ausgeben, damit die Ausgabesteuerung für einen passenden Seitenumbruch sorgen kann.

Bei den meisten Formularen wird auch die Ausgabe der Datensätze tabellarischen Charakter haben, d.h. das Format einer einzelnen Ausgabezeile wird nur noch wiederholt (wie z.B. bei den Positionen unserer Flugrechnung). Folglich ist es auch sinnvoll, diese tabellarische Ausgabeform im Formular durch Definition einzelner Zeilen zu beschreiben. Im einfachsten Fall kann dies über einen Text-Knoten mit Tabulatoren geschehen oder über eine Folge mehrerer Text-Knoten. Noch komfortabler ist die Gestaltung mit Hilfe eines Tabelle-Knotens. In diesem Fall können Sie die Ausgabezeilen als Zeilentypen ähnlich wie bei einer Schablone definieren.

7.2.2 Knotentypen zur Ausgabe von internen Tabellen

Für die Ausgabe der Daten, die in internen Tabellen vorliegen, sind die folgenden Knotentypen vorgesehen:

► Schleife

Zugriff auf Daten einer internen Tabelle. Die interne Tabelle wird in einer Schleife durchlaufen, wodurch jeder einzelne Datensatz über einen Arbeitsbereich ausgegeben werden kann. Die Art der Ausgabe wird nicht vorgegeben, sondern muss durch entsprechende Unterknoten erfolgen. Die Knotenattribute enthalten Selektionsbedingungen und Vorgaben zur Sortierung. Da die Schleife alle untergeordneten Knoten zusammenfasst, kann sie auch Funktionen eines Ordner-Knotens ausführen; z.B. zur automatischen Erzeugung von Kopf- und Fußbereichen.

► Tabelle

Der Tabelle-Knoten steuert die strukturierte Ausgabe der Daten in Form von Ausgabetablellen. Dafür finden Sie bei den Knotenattributen eine Registerkarte mit Angaben zum Tabellenlayout (in Form von Zeilentypen). Die Eingaben erfolgen ähnlich wie zu einer Schablone. Bei allen zugehörigen Unterknoten, die ausgaberelevant sind, müssen Bezüge auf die definierten Zellen im Layout der Ausgabetablelle hinterlegt sein.

Bei Bedarf kann der Tabelle-Knoten zusätzlich auch den Zugriff auf Daten einer internen Tabelle steuern. Er besitzt dadurch auch alle Funktionen einer Schleife. Eine Kombination aus beiden Funktionen ist sogar die üblichste Anwendungsform für diesen Knotentyp.

► **Komplexer Abschnitt**

Dieser Knotentyp simuliert die Eigenschaften der Knotentypen **Ordner**, **Schleife**, **Tabelle** oder **Schablone**. Je nach Eintrag auf der ersten Registerkarte wird jeweils eine der Alternativen aktiviert. Der Knotentyp stammt aus der Anfangszeit der Smart Forms-Entwicklung und sollte nicht mehr verwendet werden. Wir werden deshalb nicht weiter auf ihn eingehen.

Hinweis: Im Original unserer Flugrechnung werden allerdings mehrere Knoten vom Typ **Komplexer Abschnitt** bei Ausgabe der Positionen verwendet. Wir werden diese Knoten in einer der folgenden Übungen durch Schleife- bzw. Tabelle-Knoten ersetzen (und dabei gleichzeitig die Struktur im Hauptfenster etwas übersichtlicher gestalten).

In komplexeren Formularen sind auch Kombinationen verschiedener Knotentypen üblich:

- Einem Schleife-Knoten können Sie mehrere Tabelle-Knoten als Unterknoten zuordnen, um damit die Ausgabeform zu gestalten.
- Alternativ können Sie unterhalb eines Tabelle-Knotens auch eine oder mehrere Schleife-Knoten anlegen, die dann auf gemeinsame Eigenschaften zur Ausgabetable zurückgreifen.

Welchen Aufbau Sie wählen, hängt natürlich von der konkreten Anwendung ab. Der Tabelle-Knoten selbst lässt sich nicht weiter schachteln; er kann also nicht Unterknoten eines anderen Tabelle-Knotens sein.

Weiteres Vorgehen

In den folgenden Kapiteln werden wir die genannten Möglichkeiten zur Ausgabe von internen Tabellen im Detail erläutern und dabei auch wieder auf das Beispielformular der Flugrechnung zurückgreifen. Daher werden folgende Einzelthemen behandelt:

- Wir werden die Schleife als zentrales Element aller genannten Knoten ausführlich vorstellen; damit sind gleichzeitig auch die äquivalenten Funktionen in den Knotentypen **Tabelle** und **Komplexer Abschnitt** gemeint.
- Über die Funktion der Ausgabetable in den Knotentypen **Tabelle** und **Komplexer Abschnitt** gestalten Sie die Ausgabeform aller betreffenden Texte, Daten oder auch Grafiken.
- Bei allen drei Knotentypen können Sie Ereignisknoten nutzen, um bestimmte Unterknoten in einem Kopf- oder Fußbereich ausführen zu lassen. Die Ereignis-

nisknoten aktivieren Sie durch Festlegung der zugehörigen Zeitpunkte, zu denen die zugeordneten Unterknoten ausgeführt werden sollen.

- ▶ Zur weiteren Erläuterung bieten wir Ihnen abschließend die Möglichkeit, das Übungsbeispiel der Flugrechnung bezogen auf die Ausgabe der Positionen (Flugbuchungen) grundlegend zu ändern. Sie werden dadurch eine größere Übersichtlichkeit erreichen.

7.2.3 Schleife

Alle drei genannten Knotentypen (**Schleife**, **Tabelle** und **Komplexer Abschnitt**) können die Funktion einer Schleife zur Verfügung stellen: Verwenden Sie die Registerkarte **Daten**, um den Durchlauf über eine interne Tabelle mit seinen Eigenschaften zu beschreiben. Die Attribute zur Funktion der Schleife sind für alle drei Knotentypen gleich; wir können uns also im Folgenden auf die Beschreibung eines Falles beschränken.

In unserer Flugrechnung werden Flugbuchungen als Rechnungspositionen ausgegeben (siehe den Musterausdruck im Anhang). Die Abbildung 7.4 zeigt den Knoten LOOP_BOOK im Hauptfenster MAIN (etwas versteckt unterhalb des Knotens TABLE), der die Eigenschaften der zugehörigen Schleife beschreibt.

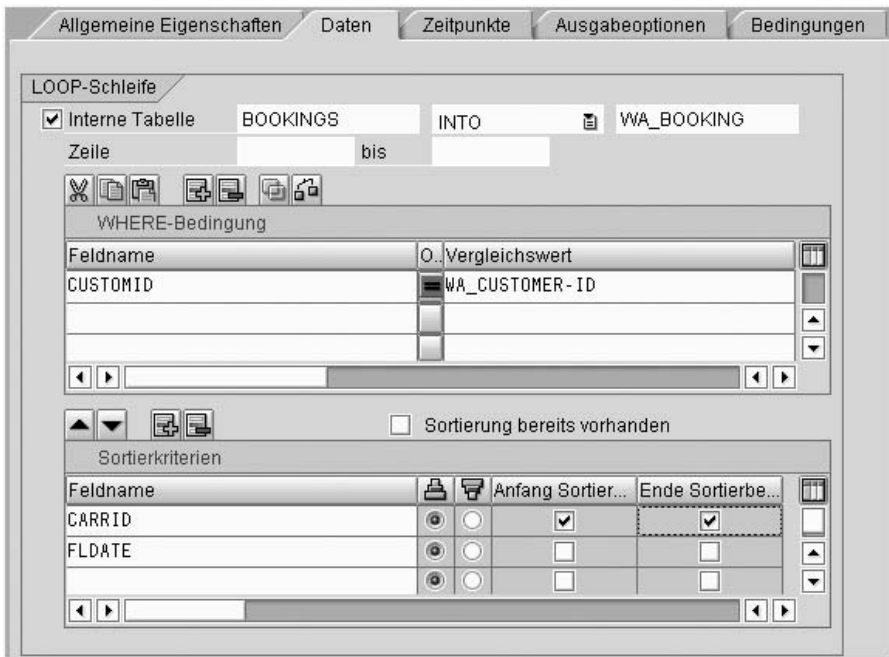


Abbildung 7.4 Merkmale einer Schleife

Bei dem Knoten LOOP_BOOK handelt es sich um einen **Komplexen Abschnitt**: Durch die Vorgaben auf der Registerkarte **Allgemeine Eigenschaften** arbeitet er wie ein Schleife-Knoten (Ausgabeart **einfach** und **wiederholte Abarbeitung**).

Daten zur Schleife

Die Registerkarte **Daten** enthält alle Attribute für den Datenzugriff innerhalb der Schleife. Die ersten und wichtigsten Angaben gelten der internen Tabelle, die durchlaufen werden soll. Dabei fällt zunächst der logische Schalter vor dem Attribut **Interne Tabelle** auf:

- ▶ Er ist im Standardfall gesetzt und sorgt dafür, dass die danach eingetragene interne Tabelle auch wirklich durchlaufen wird.
- ▶ Wenn Sie die Schleifenausführung über das Attribut deaktivieren, wird die Schleife auch nicht durchlaufen. Vom jeweiligen Knoten bleiben dann nur noch die Restfunktionen, d. h. er verhält sich ähnlich wie ein Ordner. Bei einem Tabellen-Knoten bleibt zusätzlich auch das Ausgabelayout stehen. Die Zeilentypen können also auch weiterhin zur Ausgabe von Unterknoten verwendet werden (z. B. als Ersatz für Schablone, siehe auch Hinweise in Abschnitt 4.5). Bei dem Knotentyp **Komplexer Abschnitt** (wie hier in unserem Beispiel) wird bei deaktiviertem Schalter sogar die gesamte Registerkarte **Daten** aus der Anzeige entfernt. Sie lässt sich aber nachträglich über die Registerkarte **Allgemeine Eigenschaften** wieder anwählen.

Tipp: Wurde der Schalter vor dem Attribut **Interne Tabelle** deaktiviert, so bleiben bei allen Knotentypen trotzdem die vorher eingegebenen Inhalte im Hintergrund stehen. Dies kann in der Testphase eines Formulars ganz vorteilhaft sein, wenn eine Schleife kurzzeitig von ihren Daten getrennt werden soll.

Festlegung der internen Tabelle

In unserem Beispiel ist BOOKINGS als diejenige interne Tabelle eingestellt, die in der Schleife durchlaufen werden soll. Sie enthält alle Flugbuchungen mit ihren charakteristischen Eigenschaften wie Gesellschaft, Datum und Preis, die dann bei Ausgabe als Rechnungspositionen erscheinen. Der Inhalt der internen Tabelle wird im Zuge der Schleifenbearbeitung zeilenweise in die Feldleiste WA_BOOKING als Arbeitsbereich kopiert.

Tipp: In unserem Beispiel besteht der Name jeder Feldleiste, die als Arbeitsbereich fungiert, aus dem Kürzel 'WA' (als Hinweis auf *Work-Area*) und dem Namen der internen Tabelle, für die sie eingesetzt wird. Nutzen Sie solche Formalien bei der Formularentwicklung. Sie sorgen damit vor allem auch in komplexen Formularen für Transparenz.

Nach vollständiger Abarbeitung der Schleife steht im zugewiesenen Arbeitsbereich der letzte Datensatz. Auf dessen Inhalt können Sie dann auch außerhalb der Schleife zugreifen.

Die interne Tabelle BOOKINGS wird in der Flugrechnung über die Formularschnittstelle zur Verfügung gestellt, die Feldleiste ist bei den globalen Daten im Formular definiert. Durch den Bezug auf einen einheitlichen Datentyp im ABAP-Dictionary ist sichergestellt, dass der Feldaufbau beider Komponenten gleich ist. In der Feldliste des Form Builders können Sie jederzeit die Namen der enthaltenen Felder kontrollieren.

Sonderfall 1: Interne Tabelle mit Kopfzeile

Über eine spezielle Typisierung können interne Tabellen so angelegt werden, dass sie selbst schon über einen Arbeitsbereich verfügen. Dieser Arbeitsbereich hat dabei den gleichen Namen, wie die interne Tabelle. Man spricht in diesem Fall auch von einer **internen Tabelle mit Kopfzeile**. Wenn Sie eine solche interne Tabelle auf der Registerkarte **Daten** eintragen, erkennt dies der Form Builder und übernimmt automatisch die Kopfzeile in die Angabe zum gleichnamigen Arbeitsbereich.

Aber auch bei einer internen Tabelle mit Kopfzeile können Sie weiterhin eine andere eigenständige Feldleiste als Arbeitsbereich eintragen. Dies trägt zu einer größeren Übersichtlichkeit bei. Interne Tabellen mit Kopfzeilen sollen laut SAP-Empfehlung heute ohnehin nicht mehr verwendet werden.

Sonderfall 2: Verwendung von Feldsymbolen

Für die Übertragung der Daten in den Arbeitsbereich verwenden wir die Zuweisungsart INTO. Die Alternative über ASSIGNING ist für den Fall vorgesehen, dass Sie den Arbeitsbereich über ein *Feldsymbol* abbilden wollen. Bei diesem Verfahren nutzt das System Zeiger, so dass die einzelnen Zeilen nicht in eine eigenständige Feldleiste kopiert werden müssen. Das kann vor allem bei großen internen Tabellen Vorteile in der Ausführungsgeschwindigkeit bringen.

Wenn Sie mit einem Feldsymbol arbeiten wollen, müssen Sie dieses zuvor auf der gleichnamigen Registerkarte bei den Globalen Definitionen eintragen. Wir werden im weiteren Verlauf aber nicht näher auf dieses Verfahren eingehen.

Datensätze einschränken

Wie in Abbildung 7.4 dargestellt, gibt es zwei Möglichkeiten, um beim Durchlauf der Schleife über die interne Tabelle die Zahl der angesprochenen Datensätze einzuschränken: per Zeilennummer oder über WHERE-Bedingungen.

Einschränken über Zeilennummer

Da in einer internen Tabelle jeder Datensatz eine eindeutige Nummer besitzt, können Sie durch Angabe von Zeilennummern bei **von/bis** die Anzahl der Datensätze einschränken. Diese Möglichkeit ist jedoch nur selten wirklich sinnvoll anwendbar. Um alle Datensätze in die Schleife einzuschließen, können die Eingabefelder leer bleiben.

Tipp: Eine Schleife mit Angabe **von Zeile 1 bis Zeile 1** liest nur den Inhalt der ersten Zeile einer internen Tabelle. Das kann eine sinnvolle Anwendung sein, wenn Sie z.B. auf ein einzelnes Feld zurückgreifen wollen, dessen Inhalt in allen Tabellenzeilen gleich ist. Bei Ausgabe des Feldes werden Sie dessen Inhalt wie übergeordnete Kopfdaten ausgeben wollen (also außerhalb der eigentlichen Schleife über alle Datensätze). In diesem Fall können Sie einen weiteren Schleife-Knoten dazu verwenden, nur die erste Tabellenzeile zu lesen, um das betreffende Feld auszugeben.

Einschränken über WHERE-Bedingungen

Weit häufiger erfolgt die Einschränkung der Datensätze über eine WHERE-Bedingung. Dann werden nur diejenigen Datensätze in den Arbeitsbereich kopiert und ausgegeben, die den eingetragenen Bedingungen entsprechen. Die Eingabe erfolgt in der gleichen Weise, wie es bei den logischen Ausdrücken auf der Registerkarte **Bedingungen** üblich ist (auf die wir in Abschnitt 7.3.1 noch ausführlich eingehen werden). Einziger Unterschied. Ein logischer Ausdruck hier in der Schleife darf nur Komponenten der internen Tabelle abfragen.

Hinweis: Wie in Abbildung 7.4 gezeigt, steht der Feldname zur Komponente der internen Tabelle immer in der linken Spalte. Der Feldname ist in verkürzter Version einzutragen, d.h. ohne den vorangestellten Namen der internen Tabelle. Das wird bei den ersten Versuchen mit Smart Forms häufig missachtet (die Knotenprüfung liefert dann eine Fehlermeldung).

Da in unserem Beispiel die interne Tabelle BOOKINGS unter Umständen auch Daten anderer Kunden enthalten kann, wurde die Schleife mit einer WHERE-Bedingung versehen (siehe Abbildung 7.4). Das Feld CUSTOMID enthält zu jeder Flugbuchung die zugehörige Kundennummer; der Inhalt wird deshalb mit der Nummer des Kunden verglichen, an den die Rechnung gerichtet ist. Diese Kundennummer wiederum befindet sich im Feld ID des Arbeitsbereichs WA_CUSTOMER, der ja auch für die Ausgabe der Anschrift sorgt.

Sortieren und Gruppieren

Häufig ist es erforderlich, dass die Ausgabe der Datensätze in einer vorgegebenen Reihenfolge geschieht. Allerdings ist nicht immer sichergestellt, dass die Datensätze schon in der gewünschten Sortierung in der ausgewerteten internen Tabelle vorliegen. Dann können Sie entsprechende Vorgaben als Attribute zur Schleife hinterlegen. Die Eingaben erfolgen im untersten Abschnitt der Registerkarte **Daten** (siehe Abbildung 7.4).

Beispiel

In unserem Beispiel haben wir eine Sortierung eingefügt, die im Original der Flugrechnung nicht enthalten ist.

Ihre Buchungen:			
Flug	Datum	Preis	
AA017	16.12.2000	1.200,00 USD	Kopfbereich im Hauptfenster
AA017	31.12.2000	1.200,00 USD	
Summe für AA		2.400,00 USD	
LH400	17.11.2000	581,00 DEM	Ende Sortierstufe zu AA
LH402	17.11.2000	669,00 DEM	
LH403	12.12.2000	610,00 DEM	
Summe für LH		1.860,00 DEM	Ende Sortierstufe zu LH
Gesamtsumme		2.400,00 USD	
		1.860,00 DEM	

Abbildung 7.5 Sortierung und Gruppierung

Bei Ausgabe der Positionen haben die Knotenattribute in unserem Beispiel folgende Auswirkungen:

- ▶ Die erste Sortierstufe mit Angabe von CARRID bewirkt, dass die Ausgabe der Flugbuchungen in der Reihenfolge der Fluggesellschaften erfolgt (nach Kürzel).
- ▶ Die zweite Stufe mit FLDATE bewirkt, dass danach die Sortierung nach dem Flugdatum erfolgt (also unterhalb der jeweiligen Fluggesellschaft).

In dem Ausgabebeispiel sehen Sie das Kürzel der jeweiligen Fluggesellschaft in der Spalte **Flug**, dahinter das Flugdatum: In beiden Fällen ist eine *aufsteigende Sortierung* gewählt (siehe zugehörige Einträge bei den Knotenattributen). Allerdings ist die Ausgabe keine durchgehende Liste:

- ▶ Mit Beginn jeder neuen Fluggesellschaft erscheint eine neue Überschrift mit dem ausführlichen Namen der Gesellschaft.
- ▶ Am Ende wird für jede Fluggesellschaft der Anteil am Rechnungsbetrag als Zwischensumme dargestellt.

Die Steuerung dazu erfolgt über *Ereignisknoten*, die automatisch vom Form Builder eingefügt werden, wenn die Attribute **Anfang Sort.** bzw. **Ende Sort.** gesetzt sind (hier zur Sortierstufe CARRID).

Allgemeine Hinweise

Da die Sortierung sich immer auf die verwendete interne Tabelle der Schleife bezieht, müssen die eingetragenen Felder, nach deren Inhalt die Sortierung erfolgen soll, auch in dieser internen Tabelle vorhanden sein. Sie müssen den Feldnamen deshalb wieder in verkürzter Version eintragen (d.h. ohne den vorangestellten Namen der internen Tabelle).

Wenn Sie mehrere Felder eintragen, bestimmt die Reihenfolge der Feldnamen deren Priorität bei der Sortierung. Dies lässt sich mit den schwarzen Pfeilen/Symbolen (oben links) auch nachträglich noch ändern. Jeder Eintrag in der Liste repräsentiert dann eine Sortierungsstufe.

Über die Attribute **Anfang Sortierbegriff** bzw. **Ende Sortierbegriff** können Sie je Sortierungsstufe eine *Gruppenstufe* definieren. Dann erscheint (wie in unserem Beispiel) jeweils ein Ereignisknoten im Navigationsbaum des Form Builders. Dort können Sie weitere Unterknoten anlegen, um zu den gewünschten Zeitpunkten Überschriften oder Zwischensummen auszugeben. Die Ereignisknoten werden bei Ausgabe exakt dann angesprochen, wenn sich der Inhalt des angesprochenen Feldes ändert.

Wenn Sie keine Sortierungen vorgeben, werden die Daten so ausgelesen, wie sie in der verwendeten internen Tabellen vorliegen. Es wird also die vom Rahmenprogramm vorgenommene Sortierung berücksichtigt.

Hinweis: Es ist durchaus üblich, dass eine interne Tabelle schon in der gewünschten Sortierung vom Rahmenprogramm zur Verfügung gestellt wird. Wenn Sie zu dieser Sortierung z.B. noch zusätzlich eine Gruppenstufe mit Ereignisknoten erzeugen wollen, müssen Sie zwangsläufig die Felder der Sortierung auch im Schleife-Knoten nochmals eintragen. Die Ausgabesteuerung von Smart Forms kann in diesem Fall aber nicht erkennen, dass die interne Tabelle bereits in der vorgesehenen Form sortiert ist und würde sie deshalb nochmals sortieren (was unnötige Zeit beansprucht). Wählen Sie das Attribut **Sortierung bereits vorhanden**, um dies zu vermeiden.

Ausgaben zu einer Schleife

Betrachten wir jetzt noch die Art und Weise, wie im Beispiel der Flugrechnung die Inhalte des Arbeitsbereichs WA_BOOKING mit den einzelnen Flugbuchungen über Unterknoten ausgegeben werden. Öffnen Sie dazu den Unterknoten LOOP_CONNECT. Hier finden Sie beginnend mit BODY_COL1 fünf Text-Knoten für die einzelnen Spalten der Ausgabetablelle. Öffnen Sie z.B. den letzten Text-Knoten: Sie sehen nun die Feldangaben für Preis und Währung. Beide sind in WA_BOOKING enthalten.

Bei den Ausgabeoptionen finden Sie passende Einträge, um die einzelnen Text-Knoten den Zellen einer Ausgabetablelle zuzuordnen. Darauf werden wir in Abschnitt 7.2.6 bei Behandlung der Ausgabetablellen zurückkommen.

7.2.4 Einzelnen Datensatz über Schleife lesen

Im vorangegangenen Abschnitt haben wir bereits erwähnt, wie die Schleife verwendet werden kann, um einen einzelnen Tabelleneintrag durch Vorgabe einer Zeilennummer zu lesen. Häufiger wird sich der jeweilige Datensatz über eine WHERE-Bedingung ergeben. Die Inhalte der abgefragten Felder sind dann auch erst zur Laufzeit bekannt. Im Vergleich zum ersten Fall handelt es sich also um einen dynamischen Zugriff auf einen Datensatz der internen Tabelle. Wir wollen den Fall im Folgenden wieder anhand eines Beispiels aus der Flugrechnung erläutern.

Öffnen Sie in der Flugrechnung den Knoten BODY_COL4: Die Uhrzeit des Abflugs wird offensichtlich nicht aus dem Arbeitsbereich WA_BOOKING sondern aus WA_CONNECTION gelesen. Der Grund dafür ist einfach: Bei regelmäßigen Linieneinflügen ist die Startzeit immer gleich und wird deshalb nicht direkt zum Flug, sondern zur übergeordneten Flugverbindung gespeichert (siehe auch die Hinweise zum Flugdatenmodell im Anhang).

Damit zu jeder Flugbuchung auch die richtige Uhrzeit ausgegeben werden kann, muss in WA_CONNECTION die passende Verbindung zur aktuellen Flugbuchung stehen. Dafür sorgt der Knoten LOOP_CONNECT, ebenfalls vom Typ **Komplexer Abschnitt**. Darunter sind die genannten Text-Knoten angelegt. Wählen Sie wie in Abbildung 7.6 die Registerkarte **Daten** bei den Knotenattributen.

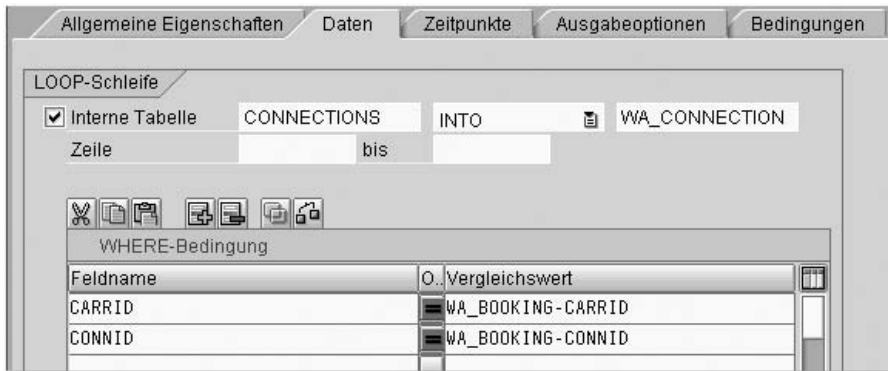


Abbildung 7.6 Schleife zum Lesen eines Datensatzes

Der Knoten arbeitet über die Voreinstellungen bei **Allgemeine Eigenschaften** wieder als eine Schleife. Über die hinterlegten WHERE-Bedingungen wird hier allerdings nur ein einziger Datensatz aus der internen Tabelle der Verbindungen gelesen und in den zugehörigen Arbeitsbereich kopiert. Die Einschränkung beim Lesen erfolgt über zwei Merkmale der jeweils aktuellen Flugbuchung:

- ▶ Feld CARRID mit Angabe der aktuellen Fluggesellschaft
- ▶ Feld CONNID mit Angabe der aktuellen Verbindungsnummer

Da eine Flugbuchung immer nur einer Verbindung zugeordnet sein kann, wird hier grundsätzlich nur ein Datensatz gelesen. Deshalb kann die im Feld DEPTIME hinterlegte Abflugzeit der Verbindung dann im Text-Knoten BODY_COL4 ausgegeben werden.

Schleife im Programmcode

Die Funktion des Schleife-Knotens wird also in diesem Beispiel etwas zweckentfremdet, um einen bestimmten Datensatz aus einer internen Tabellen zu lesen. Eine solche Funktion können Sie sonst nur über einen Programm-Knoten realisieren. Das setzt allerdings ABAP-Grundkenntnisse voraus.

Hintergrund: Mit Aktivierung des Formulars wird ein Funktionsbaustein erzeugt, dessen Programmcode alle angelegten Knoten als Folge von ABAP-Anweisungen enthält. Die eben beschriebene Schleife wird dabei in die folgende LOOP-Anweisung übersetzt:

```
*&-----&*
LOOP AT CONNECTIONS INTO WA_CONNECTION
      WHERE CARRID EQ WA_BOOKING-CARRID
           AND CONNID EQ WA_BOOKING-CONNID.
ENDLOOP.
*&-----&*
```

Listing 71 Schleife-Knoten im generierten Funktionsbaustein

Das Ergebnis des Lesevorgangs im Arbeitsbereich WA_BOOKING steht nicht nur innerhalb der Schleife zur Verfügung, sondern so lange, bis dessen Inhalt erneut überschrieben wird. Ausgaben zum Arbeitsbereich der Schleife können infolge dessen über beliebige Knoten erfolgen, ohne dass diese als Unterknoten der Schleife angelegt sein müssen.

Hinweis: Wird ein Arbeitsbereich über einen Schleife-Knoten gelesen und durch Felder angesprochen, die nicht unterhalb des Knotens angelegt sind, kann die Gesamtprüfung des Formulars zu einer Warnung führen. Der Form Builder geht in diesem Fall davon aus, dass der betreffende Arbeitsbereich keine gültigen Wert hat. Dann sollten Sie den Arbeitsbereich einmalig an anderer Stelle als Ausgangsparameter angeben; z. B. unter der Registerkarte **Initialisierung** bei den Globalen Definitionen (siehe auch Hinweise unter Abschnitt 3.4). Auch in dem folgenden Übungsbeispiel wird dieser Fall auftreten.

7.2.5 Übungsbeispiel: Schleife

Aus den bisherigen Darstellungen in diesem Kapitel geht hervor, dass der Knoten LOOP_CONNECT eigentlich keine echte Schleife darstellt, sondern »nur« zur Datenbeschaffung für die Variable dient, die das Flugdatum enthält.

Die eigentliche Ausgabe der Positionen ist über den Knoten LOOP_BOOK gesteuert, der nacheinander alle Flugbuchungen durchläuft. Diesem Knoten sollten dann auch alle Unterknoten zugeordnet sein, die für die Ausgabe sorgen (im Augenblick stehen diese unterhalb von LOOP_CONNECT).

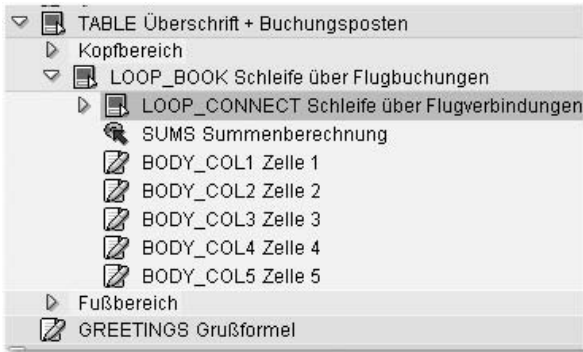


Abbildung 7.7 Positionsausgaben hinter LOOP_CONNECT

Verschieben Sie die Knoten so per Maus, dass die bisherigen Unterknoten von LOOP_CONNECT jetzt hinter diesem Knoten folgen (siehe Abbildung 7.7). Achten Sie darauf, dass die Reihenfolge der Knoten erhalten bleibt. Geben Sie das Formular erneut über Programm Z_SF_EXAMPLE_01 aus (vorheriges Aktivieren nicht vergessen), und kontrollieren Sie, ob die Abflugzeit jetzt wirklich noch enthalten ist.

Hinweis: Leider liefert die Gesamtprüfung zum Formular jetzt eine neue Warnung mit dem Inhalt »Feld WA_CONNECTION-DEPTIME besitzt keinen definierten Wert«. Das Phänomen haben wir weiter oben schon erläutert und Ihnen auch das »Gegenmittel« genannt: Tragen Sie WA_CONNECTION bei den Globalen Definitionen unter der Registerkarte **Initialisierung** als Ausgangsparameter ein. Dann verläuft auch die Gesamtprüfung ohne Fehlermeldung.

Wir werden am Ende dieses Kapitels als Übungsbeispiel größere Änderungen an dem Aufbau des Zweiges vornehmen, der zur Ausgabe der Rechnungspositionen dient (siehe Abbildung 7.2.8). Dabei können Sie auch neue Knoten vom Typ **Schleife** anlegen bzw. die bisherigen Knoten **Komplexen Abschnitt** ersetzen.

7.2.6 Ausgabetabelle

Im letzten Abschnitt haben wir erläutert, wie die Positionen der Flugrechnung in einer Schleife durchlaufen und über Text-Knoten ausgegeben werden. Wir haben aber noch nicht betrachtet, wie diese Ausgaben in die tabellarische Darstellung gebracht werden. Dazu wird das Layout einer Ausgabetabelle verwendet, die wir grundlegend schon in Abschnitt 4.5 vorgestellt haben.

Überprüfen Sie deshalb zunächst die Registerkarte **Tabelle** im Knoten TABLE, die Sie in Abbildung 7.8 sehen.



Abbildung 7.8 Ausgabetablelle

Auch in diesem Fall handelt es sich wieder um einen Knoten vom Typ **Komplexer Abschnitt**. Auf der Registerkarte **Allgemeine Eigenschaften** ist er diesmal allerdings konfiguriert als Tabellen-Knoten mit **wiederholter Abarbeitung**. Mit diesen Vorgaben wird über die Eintragungen unter der Registerkarte **Daten** wieder eine interne Tabelle durchlaufen. Auf deren Inhalt gehen wir später ein (wir werden zum Schluss auch diesen Knoten überarbeiten).

Jeder Tabelle-Knoten enthält als Besonderheit die Registerkarte **Tabelle**: Hier ist das Layout der Ausgabetablelle definiert, die von den zugehörigen Unterknoten angesprochen werden kann. Die Definition erfolgt ähnlich wie bei einer Schablone wieder über Zeilentypen: Sie werden die drei in unserem Beispiel angelegten Zeilentypen sofort auf einem Musterausdruck des Formulars wiedererkennen. Eine kleine Besonderheit ist, dass die Zeilentypen TABLE_HEADER für die Kopfangaben und TABLE_POS für die einzelnen Positionen in unserem Fall die gleiche Spaltenteilung haben; man wäre also im Grunde auch mit einem Zeilentyp weniger ausgekommen.

Die einzelnen Spalten sind wieder mit ihrer Breite eingetragen. Was im Vergleich zur Schablone fehlt, ist die Zeilenhöhe. Sie wird dynamisch bestimmt und hängt deshalb vom Inhalt der Zelle und dem dort verwendeten Stil ab. Die Zelle mit der größten Höhe bestimmt auch die Höhe der ganzen Zeile. Die einzelnen Zeilen können deshalb bei der Ausgabe auch eine unterschiedliche Höhe haben (abhängig von den jeweiligen Inhalten).

Bei jedem ausgaberelevanten Unterknoten zu einer Ausgabetablelle muss der gewünschte Zeilentyp angegeben werden. Über das Attribut **Default** im Tabellenlayout legen Sie fest, welcher Zeilentyp verwendet werden soll, wenn bei der Ausgabe eines Unterknotens kein Zeilentyp angegeben ist.

Über das Attribut **Schutz vor Umbruch** legen Sie fest, ob der betreffende Zeilentyp gegen Seitenumbruch geschützt sein soll. In diesem Fall wird der Inhalt einer Zeile immer komplett auf einer Seite ausgegeben.

Vor Eingabe der Zeilentypen müssen Sie wie bei der Schablone eine Gesamttabellenbreite sowie die Positionierung innerhalb des übergeordneten Fensters festlegen. Die Ausgabetablelle darf ebenfalls nicht breiter sein als das Fenster. Die Summe aller Zellen muss in der Breite mit der gesamten Tabellenbreite übereinstimmen. Für das Design der Zeilentypen steht auch hier der grafische Table Painter zur Verfügung; er überprüft die Übereinstimmung der Zeilentypen mit der Gesamtbreite.

Zuordnung der Zellen im Unterknoten

Wir haben bisher gesehen, wie das Layout der Ausgabetablelle erstellt wird. Betrachten wir nun, wie ausgaberelevante Unterknoten den Zellen innerhalb der Ausgabetablelle zugeordnet werden. Öffnen Sie dazu z.B. den Text-Knoten BODY_COL1 zur Ausgabe der Fluggesellschaft, diesmal aber mit der Registerkarte **Ausgabeoptionen**. Dort steht jetzt ein weiterer Abschnitt für die Zuordnung innerhalb der Ausgabetablelle zur Verfügung (siehe Abbildung 7.9).



Abbildung 7.9 Ausgabeoptionen zur Tabelle

Wir haben in Abschnitt 4.5 bei Betrachtung von Schablonen gesehen, dass Text- oder Grafik-Knoten über feste Zeilen- und Spaltenangaben den Schablonenzellen zugewiesen sind. Das ergibt sich aus dem festen Aufbau einer Schablone. Bei einer Ausgabetablelle existieren hingegen keine Zeilenangaben, denn es können beliebig viele sein (je nach Datenmenge). Deshalb muss hier eine dynamische Zuordnung der Text-Knoten erfolgen:

- ▶ Da der Knoten zu Abbildung 7.9 das Kürzel der Fluggesellschaft ausgibt und es sich dabei um die erste Spalte in der Ausgabetablelle handelt, muss hier das Attribut für **neue Zeile** gesetzt sein.
- ▶ Über das untere Attribut **neue Zeile** wird zusätzlich eine neue Zeile angesprungen (es ist zugleich die erste, da keine Zeile übersprungen wird).

Überprüfen Sie zusätzlich noch die nächsten Knoten BODY_COL2 usw.: dort ist immer nur das Attribut **neue Zelle** gesetzt.

Ist bei einem ausgaberelevanten Knoten weder **neue Zeile** noch **neue Zelle** gesetzt, so wird der Inhalt an den vorherigen Knoten angehängt. Bei einem Text-Knoten müssen Sie in diesem Fall wieder die Angaben zu **Neuer Absatz** etc. auf der ersten Registerkarte des Knotens beachten. Mit Eröffnung einer neuen Zelle beginnt die Formatierung dagegen immer mit einem neuen Absatz (wie es im aktuellen Text eingestellt ist).

7.2.7 Zeitpunkte (Ereignisknoten)

Da jeder der Knotentypen mit der Möglichkeit zur dynamischen Datenausgabe auch untergeordnete Knoten zusammenfasst, ist er zusätzlich mit den Funktionen eines Ordner-Knotens ausgestattet. Das ermöglicht die automatische Erzeugung von Ereignisknoten für Kopf- und Fußbereiche, z. B. zur Ausgabe von Überschriften oder abschließenden Summen.

Die zugehörigen Einstellungen erfolgen bei den jeweiligen Knotenattributen mit Hilfe der Registerkarte **Zeitpunkte**. Öffnen Sie (wie in Abbildung 7.10) den Knoten TABLE in unserer Flugrechnung, in dem die bisherigen Kopf- und Fußbereiche angelegt sind.

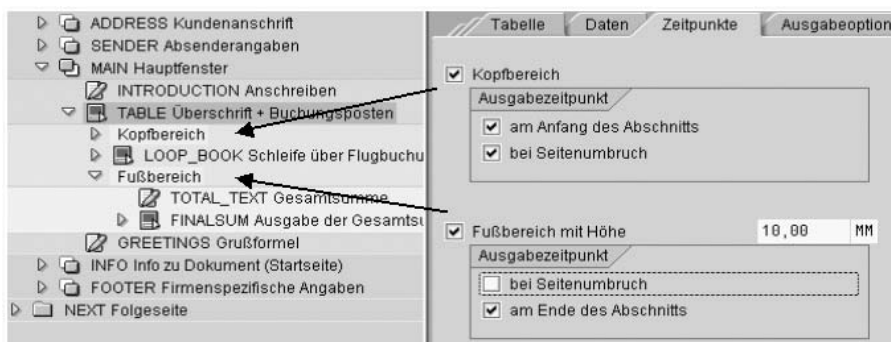


Abbildung 7.10 Ereignispunkte bei dynamischer Datenausgabe

Die Ereignisknoten können Sie zu den folgenden Zeitpunkten ausgeben lassen:

- ▶ Am Anfang und am Ende des Abschnitts, wobei ein Abschnitt durch die Abfolge aller Unterknoten definiert ist
- ▶ Bei einem Seitenumbruch (jeweils am Beginn oder am Ende einer Seite)

Für den Fußbereich benötigt das Formular eine Höhenangabe, damit bei Berechnung des Seitenumbruchs der benötigte Platz für den Fußbereich einkalkuliert werden kann.

Wenn Sie einen Kopf- und/oder Fußbereich anwählen, erzeugt der Form Builder im Navigationsbaum automatisch den passenden Ereignisknoten (mit der Bezeichnung **Kopfbereich** etc.). Dort können Sie jetzt weitere Unterknoten einfügen. In unserem Beispiel sind bereits die Texte zur Tabellenüberschrift bzw. die Rechnungssummen in den Währungen der einzelnen Flugbuchungen angelegt.

Hinweis: Ein Fußbereich wird üblicherweise für die Ausgabe von Zwischensummen verwendet. Für diese Anwendung muss sichergestellt sein, dass die entsprechenden Summen vorher vollständig berechnet worden sind. Wie die Berechnung der Gesamtsummen erfolgt, werden wir später noch ausführlich darstellen (siehe Abschnitt 8.6).

Ereignisknoten können Sie nicht weiter schachteln. Wenn also einmal ein Kopf- oder Fußbereich für einen Abschnitt definiert ist, können Sie keinen weiteren untergeordneten Abschnitt mit eigenem Kopf- oder Fußbereich anlegen. Zusätzlich müssen Kopf- und Fußbereich, falls beide Bereiche benutzt werden sollen, immer im gleichen Knoten definiert sein.

Tipp: Beachten Sie deshalb beim Design des Formulars, dass bei kombinierter Verwendung von Tabelle- und Schleife-Knoten die Ereignisknoten nicht auf jeder Ebene zur Verfügung stehen, sondern vom jeweils übergeordneten Knoten beansprucht werden.

7.2.8 Übungsbeispiel: Flugrechnung überarbeiten

Im letzten Kapitel haben wir im Knoten TABLE nur die Vorgaben zur Gestaltung der Ausgabetabelle überprüft. Rufen Sie den Knoten noch einmal auf, jetzt aber mit der Registerkarte **Daten**: Hier werden alle Angaben zum aktuellen Kunden aus einer internen Tabelle in den zugehörigen Arbeitsbereich WA_CUSTOMER kopiert.

Da das Formular aber immer nur an einen Kunden gerichtet ist, ist auch dies wieder keine echte Schleifenabwicklung (diesen Sonderfall zur Schleife haben wir in Abschnitt 7.2.3 angesprochen).

Der gefüllte Arbeitsbereich wird darüber hinaus eigentlich auch nicht für die Positionsdaten benötigt, sondern dient primär zur Ausgabe der Kundenanschrift im ADRESS-Fenster.

Hinweis: Die Ausgabe der Adresse hat bisher nur funktioniert, weil der zugehörige Adress-Knoten ADRESS im Navigationsbaum hinter MAIN und damit auch hinter TABLE folgt. Verschieben Sie probeweise das gesamte Adressfenster im Navigationsbaum vor MAIN und geben Sie das Formular erneut über das Rahmenprogramm aus. Jetzt wird die Adresse nicht mehr enthalten sein.

Mit dieser Funktion zur Beschaffung der Empfängeradresse besitzt der Knoten TABLE eine Funktion, wie sie eigentlich im MAIN-Fenster nicht zu erwarten wäre. Aber schon deshalb besitzt der Zweig zur Ausgabe der Rechnungspositionen einen Knoten-Aufbau, der sich in einigen Punkten optimieren lässt. Benötigt werden eigentlich nur:

- ▶ Ein einziger Tabelle-Knoten, in dem die ganze Schleifenabwicklung über die Flugbuchungen und die Ausgabe der Positionen erfolgt
- ▶ Ein Knoten für die Beschaffung der Uhrzeit einer Verbindung (wie bisher realisierbar über eine Schleife zum Lesen eines einzelnen Datensatzes)
- ▶ Ein Knoten außerhalb von MAIN für die Beschaffung der Adresse, wobei die Ausgabe der Adresse **nach** diesem Knoten erfolgen muss (wie bisher realisierbar über eine Schleife zum Lesen eines einzelnen Datensatzes)

Übersicht zur Strukturänderung

Im folgenden Übungsbeispiel wollen wir eine neue, übersichtlichere Knotenstruktur zur Ausgabe der Positionstabelle anlegen. Dies sind die Eckpunkte unseres Vorgehens:

- ▶ Die Ausgabe soll (wie soeben beschrieben) nur noch über einen neuen, zentralen Tabelle-Knoten gesteuert sein.
- ▶ Die ausgegebenen Unterknoten werden in die neue Struktur verschoben; sie können also wie bisher weiterverwendet werden.
- ▶ Statt der Knoten vom Typ **Komplexe Abschnitte** werden echte Tabelle- bzw. Schleife-Knoten mit gleichen Eigenschaften angelegt.

In Abbildung 7.11 ist das Ergebnis einer solchen Änderung dargestellt. Insbesondere die Struktur des MAIN-Fensters ist durch weniger Ebenen im Navigationsbaum einfacher geworden. Die Beschaffung der Daten zur Adresse erfolgt jetzt da, wo sie eigentlich hingehört: im Fenster zur Adressausgabe.

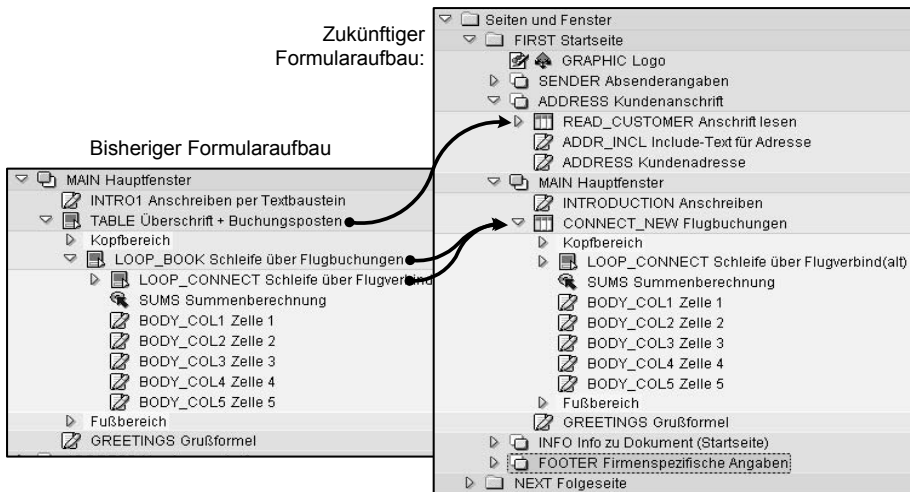


Abbildung 7.11 Formularoptimierung

Gehen Sie Schritt für Schritt vor; da die notwendigen Knoten größtenteils in der Vorlage schon vorhanden sind, dürfte das Übungsbeispiel leicht nachvollziehbar sein. Auf der Basis der neuen Struktur werden wir später weitere Änderungen vornehmen.

Schritt 1: Tabellen-Knoten anlegen

Aus der Betrachtung zum Knoten TABLE ergibt sich, dass er für die Ausgabe der Flugbuchungen im Hauptfenster nicht erforderlich ist. Wir werden ihn allerdings wegen der darin enthaltenen Zuweisung der Empfängeradresse beibehalten. Für die Ausgabe der eigentlichen Rechnungspositionen wollen wir jedoch einen neuen Tabelle-Knoten anlegen, der sowohl die Formatierung der Ausgabetabelle enthält, als auch die Schleife über alle Flugbuchungen.

Erzeugen Sie im MAIN-Fenster den neuen Tabellen-Knoten oberhalb von GREETINGS über das Kontextmenü mit **Anlegen • Tabelle**. Wählen Sie als Namen BOOK_NEW und eine passende Beschreibung. Die Tabellenbreite ist 15,25 cm. Per Default ist im Tabelle-Knoten die Registerkarte **Daten** aktiviert. Füllen Sie für den Knoten nacheinander die folgenden Registerkarten:

► Tabelle

Übertragen Sie das Design der Ausgabetabelle aus dem bisherigen Knoten TABLE (Breite, Zeilentypen, Linienmuster). Sie können hierfür auch die schnellere SAP-Kopierfunktion mit erweiterter Zwischenablage (**Strg + Y** etc.) verwenden. Wenn Ihre Positionen wie bisher mit sichtbaren Rändern erscheinen sollen, wählen Sie zusätzlich das passende Muster (über die zugehörige Taste).

► **Daten**

Die Schleife soll wie im bisherigen Knoten LOOP_BOOK über die interne Tabelle der Flugbuchungen laufen. Übertragen Sie die Einstellungen.

► **Zeitpunkte**

Übernehmen Sie die Vorgaben für Kopf- und Fußbereich wieder vom Knoten TABLE.

Überprüfen Sie die einzelnen Knoten; auch die Gesamtprüfung darf keine Fehler melden.

Schritt 2: Bisherige Unterknoten verschieben

Verschieben Sie jetzt die untergeordneten Knoten in die neuen Zweige des Navigationsbaums (per Maus über Drag&Drop):

- Beginnen Sie mit den Unterknoten im Kopf- und Fußbereich, die bisher unterhalb von TABLE angelegt waren. Achten Sie darauf, dass die Reihenfolge der Unterknoten auch unterhalb von BOOK_NEW erhalten bleibt.
- Verschieben Sie dann diejenigen Knoten direkt zum neuen Knoten BOOK_NEW, die für die Ausgabe der Positionen zuständig sind. Diese befinden sich unterhalb von LOOP_BOOK. Im Übungsbeispiel zu Abschnitt 7.2.4 haben wir auch bereits die Knoten zur Ausgabe der Positionen aus der Ebene unterhalb von LOOP_CONNECT hierhin verschoben.

Der neue Zweig im Navigationsbaum sollte jetzt etwa so aussehen wie in Abbildung 7.12 gezeigt.

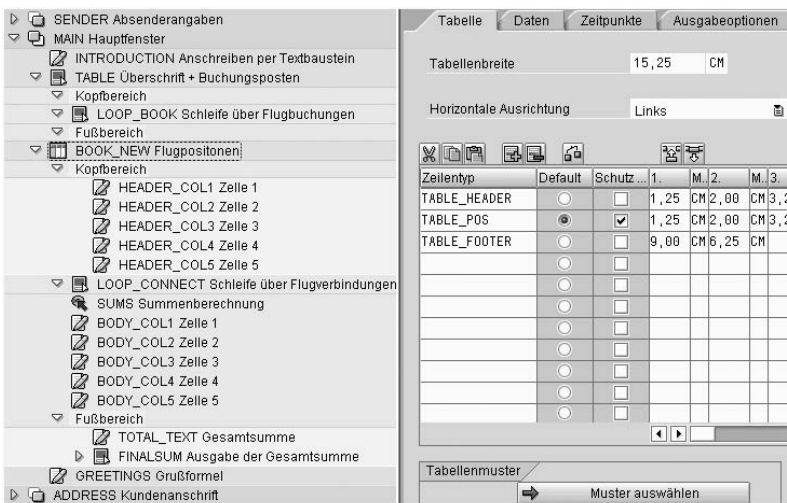


Abbildung 7.12 Geänderte Positionsausgabe

Alle relevanten Inhalte des bisherigen Knotens TABLE sind nach BOOK_NEW übertragen worden. Die bisherige Schleifenabwicklung über TABLE hat keine Inhalte mehr; wir werden sie im nächsten Schritt entfernen.

Überprüfen Sie das gesamte Formular; es dürfen keine neuen Fehler auftreten. Testen Sie die Ausgabe über das Rahmenprogramm Z_SF_EXAMPLE_01. Das Ergebnis sollte den vorherigen Ausgaben gleichen.

Schritt 3: Alte Knoten bzw. Einstellungen entfernen

Abschließend noch etwas für die Schönheit:

- ▶ Der bisherige Knoten LOOP_BOOK ist vollständig in den neuen Tabellen-Knoten aufgegangen; Sie können ihn löschen.
- ▶ Entfernen Sie auch die Kopf- und Fußbereiche unterhalb von TABLE; sie würden jetzt höchstens noch Leerzeilen bei der Ausgabe erzeugen (auf der Registerkarte **Zeitpunkte**).
- ▶ Da der bisherige Knoten TABLE jetzt nur noch zur Beschaffung der Kundendaten dient, ändern Sie den Namen auf READ_CUSTOMER.
- ▶ Es ist sicher auch sinnvoller, das Beschaffen der Adresse dort vorzunehmen, wo sie gebraucht wird: Verschieben Sie den Knoten READ_CUSTOMER also in das Fenster ADRESS. Er sollte dort der erste Eintrag sein, damit die anschließenden Text-Knoten auch Inhalte haben.

Da für diesen Knoten jetzt die Gestaltung der Ausgabetablelle nicht mehr relevant ist, sollten Sie unter der Registerkarte **Allgemeine Eigenschaften** die Ausgabeart auf **einfach** ändern (sonst meldet die Gesamtprüfung, die Tabelle passe nicht ins Fenster). Durch diese Option verhält sich der Knoten vom Typ **Komplexer Abschnitt** jetzt wie ein Schleife-Knoten.

- ▶ Verschieben Sie abschließend das gesamte Fenster ADRESS vor MAIN.

Dieses Vorgehen hat zwei Gründe:

- ▶ Die Navigation im Navigationsbaum fällt leichter, wenn diejenigen Fenster, die auf dem Papier oben platziert sind, auch im Navigationsbaum zuerst aufgeführt werden.
- ▶ Die Kundennummer muss vor Ausführung des MAIN-Fensters bekannt sein, denn über die Kundennummer in WA_CUSTOMER-ID haben wir auch die Flugbuchungen im Tabelle-Knoten BOOK_NEW gefunden.

Die neue Struktur im Navigationsbaum sollte jetzt auf dem Stand sein, den wir in Abbildung 7.11 zu Beginn dieses Kapitels gezeigt haben.

Optionale Änderungen

Sowohl der Knoten READ_CUSTOMER als auch LOOP_CONNECT unterhalb von BOOK_NEW sind im Augenblick noch mit Hilfe des Knotentyps **Komplexer Abschnitt** realisiert. Sie dienen jeweils nur noch dazu, einen einzelnen Datensatz zu lesen und in eine Feldleiste zu kopieren. Sie haben keine weiteren Unterknoten. Ersetzen Sie die Knoten durch echte Schleife-Knoten. Gehen Sie wie folgt vor:

- ▶ Legen Sie hinter dem bisherigen Knoten jeweils einen neuen Schleife-Knoten an. Dadurch wird der neue Knoten zeitlich nach dem alten ausgeführt: der dabei gelesene Datensatz stammt folglich auch immer aus dem neuen Knoten.
- ▶ Vergeben Sie die Attribute wie im bisherigen Knoten (**Interne Tabelle, Arbeitsbereich, Bedingungen**).
- ▶ Überprüfen Sie das Formular und testen Sie die Ausgabe.
- ▶ Löschen Sie dann die bisherigen Knoten vom Typ **Komplexer Abschnitt** und geben Sie den neuen Knoten die Namen des bisherigen Knotens.

7.3 Logische Abfragen

Die gängigste Form zur Beeinflussung der Ablauflogik des Formulars besteht in der Definition von logischen Abfragen. Zwei Möglichkeiten stehen zur Verfügung:

- ▶ Zu jedem beliebigen Knoten im Formular (unterhalb des Seite-Knotens) können *Bedingungen* hinterlegt sein. Damit können Sie die Ausführung des Knotens und ggf. auch aller seiner Unterknoten freischalten oder unterbinden.
- ▶ Der *Alternative*-Knoten stellt im Grunde nur eine Erweiterung von Bedingungen für den Fall dar, dass zwei Knoten unter entgegengesetzten Voraussetzungen ausgeführt werden sollen. Der Knotentyp sorgt dann für eine transparentere Darstellung und verringert den Eingabeaufwand.

7.3.1 Bedingungen

Alle manuellen Knotentypen unterhalb einer Seite (mit Ausnahme des Programm-Knotens) besitzen eine Registerkarte **Bedingungen**, um die Ausführung des Knotens an die jeweilige Formularlogik anzupassen.

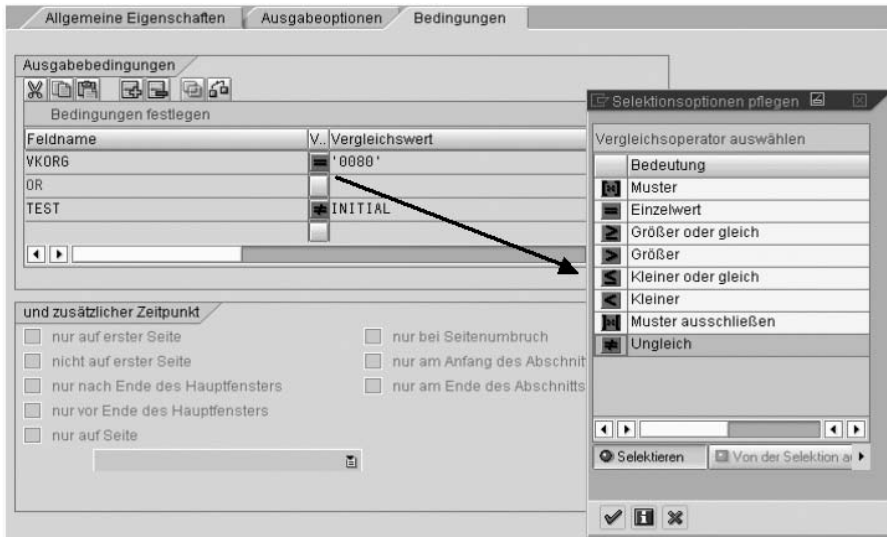


Abbildung 7:13 Bedingungen zum Knoten

Sobald eine Bedingung angelegt wurde, ist dies sofort auch an der Darstellung des zugehörigen Knotens im Navigationsbaum erkennbar. Smart Forms unterscheidet zwei Typen von Bedingungen:

- ▶ Logische Ausdrücke
- ▶ Ausgabezeitpunkte

Wir beschreiben zunächst die jeweiligen Eigenschaften.

Überwachung logischer Ausdrücke

In jeder Zeile der Bedingungen können Sie einen logischen Ausdruck mit zwei Operanden eintragen. Jeder der Operanden kann ein Feld oder ein Wert sein, denn trotz der Bezeichnungen **Feldname** und **Vergleichswert** sind beide Spalten gleichwertig. Geben Sie die Feldnamen ohne &-Klammerung ein, die Werte dagegen in Hochkommata (bei numerischen Werten auch ohne Hochkommata).

Der Vergleichsoperator steht zwischen den einzelnen Operanden und ergibt sich aus der hinterlegten Liste wie in Abbildung 7:13 dargestellt. Ohne direkte Vorgabe durch den Anwender erzeugt der Form Builder bei Neueingabe per Default eine Überprüfung auf Gleichheit.

Durch zeilenweise Eingabe mehrerer Bedingungen können Sie auch komplexe logische Abfragen einstellen. Dabei werden die Einzelabfragen in den Zeilen über eine logische UND-Verknüpfung verknüpft. Um eine ODER-Verknüpfung zwi-

schen zwei Zeilen zu erreichen, wählen Sie die entsprechende Taste in der Symbolleiste oder geben Sie die Buchstaben OR in die erste Spalte ein. Nach Bestätigung der Eingabe wird die Zeile in beiden Fällen automatisch grau dargestellt, um weitere Ergänzungen zu verhindern.

Tipp: ABAP kennt bei logischen Abfragen neben dem Vergleich von Operanden auch eine Abfrage auf **is initial**. Dabei wird überprüft, ob dem zugehörigen Feld nach seiner Deklaration ein gültiger Wert zugewiesen wurde. Diese Überprüfung können Sie auch als Bedingung unter Smart Forms hinterlegen, indem Sie beim zweiten Operanden INITIAL eingeben (siehe Beispiel in Abbildung 7.13).

Überwachung von Ausgabezeitpunkten

Über den unteren Abschnitt **Zusätzlicher Zeitpunkt** verknüpfen Sie die Ausführung des Knotens zusätzlich mit besonderen Ereignissen der Formularprozessierung. Die Bedingungen sind eigentlich selbsterklärend, deshalb geben wir hier nur kurze Hinweise:

- ▶ Bei den Eigenschaften **nur/nicht auf erster Seite** sind immer einzelne Ausgabeseiten gemeint, nicht die Entwurfsseiten.
- ▶ Die Bedingungen **nur am/vor Ende des Hauptfensters** sind nur sinnvoll anwendbar bei Knoten in Nebenfenstern, die zusätzlich im Navigationsbaum hinter dem Hauptfenster stehen. Die hinterlegte Bedingung basiert auf einer Abfrage zur Systemvariablen SFSY-PAGEBREAK. Diese wird genau dann gesetzt, wenn das Hauptfenster komplett abgearbeitet ist. Anwendungsbeispiel: die Ausgabe einer Zwischensumme innerhalb eines Nebenfensters, das auf allen Seiten außer der letzten erscheinen soll.
- ▶ Die Bedingung **nur auf Seite** erfordert die Angabe einer Entwurfsseite. In vielen Fällen können Sie alternativ dazu auch gleich den zugehörigen Knoten auf der entsprechenden Entwurfsseite löschen. Doch das geht nur, wenn das zugehörige Fenster ausschließlich auf dieser Seite existiert. Als Kopie auf mehreren Seiten (wie es z.B. bei MAIN üblicherweise der Fall ist) würde der betreffende Knoten sonst auf allen Seiten entfernt.

Die Einstellungen rechts zu **Seitenumbruch** und **Anfang/Ende von Abschnitten** sind nur bei Knoten anwählbar, die unterhalb von Ereignisknoten angelegt sind. Ereignisknoten wiederum existieren z.B. bei den Knotentypen Ordner und Schleife; sie ermöglichen die automatische Erstellung von Kopf- und Fußbereichen (siehe z.B. Abschnitt 5.4).

Verknüpfung der Überwachungen

Die Ergebnisse der Einzelüberwachungen zu logischen Ausdrücken und den Ausgabezeitpunkten sind untereinander mit UND verknüpft. Nur wenn die Gesamtbedingung erfüllt ist, wird der Knoten abgearbeitet. Ist sie nicht erfüllt, wird dieser Knoten zusammen mit allen Unterknoten ignoriert.

Im Grunde sind auch alle Unterknoten bei der Auswertung der Bedingungen mit einem logischen UND zum übergeordneten Knoten verbunden. Darüber können Sie letztendlich eine beliebig komplexe Abfragelogik im Formular hinterlegen.

Tipp: Die Verwendung von Bedingungen empfiehlt sich häufig auch in der Entwicklungsphase eines Formulars, z.B. bei der Ursachenforschung zu Laufzeitfehlern. Über Bedingungen können Sie dann testweise ganze Zweige »auskommentieren« und so die Fehlerquelle einkreisen.

7.3.2 Verzweigungen über Alternative-Knoten

Der Alternative-Knoten bietet sich als elegante Lösung an, wenn zwei Knoten unter entgegengesetzten Bedingungen ausgeführt werden sollen.

Mit Neuanlage eines solchen Knotentyps erzeugt der Form Builder automatisch zwei weitere Ereignisknoten TRUE und FALSE als direkte Unterknoten im Navigationsbaum: Dort fügen Sie die eigentlichen Unterknoten ein, die abhängig vom Ergebnis der logischen Abfrage ausgeführt werden sollen (ggf. können das auch weitere Alternative-Knoten sein).

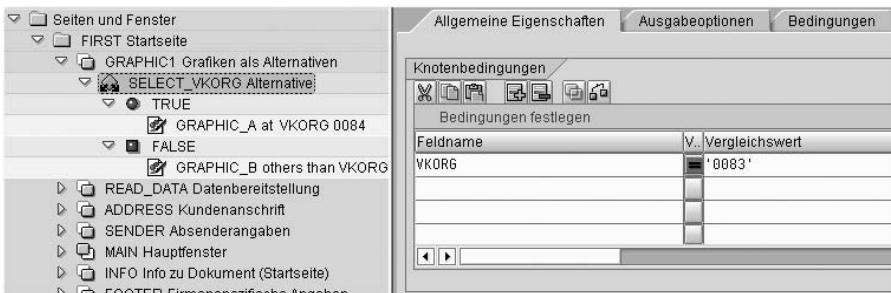


Abbildung 7.14 Verzweigungen über Alternative-Knoten

In dem einfachen Beispiel der Abbildung 7.14 wird – abhängig von der Abfrage zur Verkaufsorganisation – bei der Ausgabe zwischen zwei Grafikalalternativen gewählt, die jeweils im Zweig TRUE bzw. FALSE angelegt sind.

Wichtig: Den Inhalt der Abfrage bestimmen Sie über die Registerkarte **Allgemeine Eigenschaften** des Alternative-Knotens, wobei die Eingabemöglichkeiten exakt denen auf der Registerkarte **Bedingungen** entsprechen (was anfangs leicht zu Verwechslungen führt). Die letztgenannte Registerkarte existiert zudem ebenfalls. Darüber können Sie wieder die Abarbeitung des gesamten Alternative-Knotens mit einer Bedingung versehen.

Bei dem Beispiel der Abbildung 7.14 hätte man das gleiche Ergebnis auch über zwei Grafik-Knoten erreichen können. Dazu müsste jeweils nur als Bedingung die Abfrage zur Verkaufsorganisation entgegengesetzt hinterlegt sein (einmal auf *Gleichheit* und einmal auf *Ungleichheit*).

Hinweis: Der Alternative-Knoten wird im Quelltext des generierten Funktionsbausteins zum Formular in die ABAP-Anweisung `IF ... THEN ... ELSE ... ENDIF` übersetzt.

7.4 Abfolge und Nummerierung der Ausgabeseiten

Über das Systemfeld SFSY-PAGE können Sie die aktuelle Seitenzahl in einen Text-Knoten im Formular ausgeben oder auch in sonstigen Knotenattributen abfragen, wie z. B. bei Bedingungen (siehe auch Abschnitt 6.4.3). Der Inhalt des Seitenzählers hängt von der Abfolge der Seiten im Formulars ab. Hierauf wiederum haben die folgenden Faktoren Einfluss:

- ▶ Die Menge an Datensätzen, die auszugeben sind, und der Platz im Hauptfenster, der dafür zur Verfügung steht
- ▶ Die eingestellten Knotenattribute der jeweiligen Entwurfsseite
- ▶ Die Verwendung dynamischer Seitenumbrüche per Kommando-Knoten

Reihenfolge der Seiten

Normalerweise erfolgt ein Seitenwechsel immer dann, wenn das Hauptfenster einer Seite gefüllt ist. Dann wird als nächstes die Seite prozessiert, die bei dem aktuellen Seite-Knoten als Folgeseite eingetragen ist (auf der Registerkarte **Allgemeine Eigenschaften**). In unserem Beispiel der Flugrechnung arbeiten die Entwurfsseiten FIRST und NEXT entsprechend dieser statischen Logik. Auf diese Abwicklung sind wir auch schon zu Beginn dieses Kapitels eingegangen.

Eine Abweichung von diesem Ablauf tritt ein, wenn z. B. am Ende der normalen Formularausgabe ein weiteres Blatt ausgegeben werden soll (etwa als Geschäftsbedingungen). Dann muss die Ausgabesteuerung dynamisch entscheiden können, wann auf welche andere Seite gesprungen werden soll. Die Vorgabe können Sie über einen Kommando-Knoten im Formular festlegen.

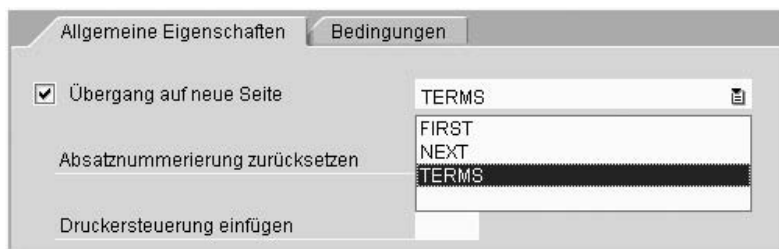


Abbildung 7.15 Dynamischer Seitenwechsel im Kommando-Knoten

Wählen Sie bei Bedarf die Option **Übergang auf neue Seite** und ordnen Sie das Kürzel der Seite zu (siehe Abbildung 7.15). Wann der Kommando-Knoten ausgeführt werden soll, legen Sie wie üblich auf der Registerkarte **Bedingungen** fest.

Grundsätzlich können Sie einen Kommando-Knoten an beliebiger Stelle im Navigationsbaum einfügen (siehe Abschnitt 7.5). Um aber die Option **Übergang auf neue Seite** verwenden zu können, müssen zusätzlich zwei Voraussetzungen erfüllt sein:

1. Der Kommando-Knoten muss in diesem Fall im Hauptfenster angelegt sein. Ansonsten erscheint bei der Ausgabe des Formulars eine Laufzeitfehlermeldung.
2. Gleichzeitig darf sich der dynamische Seitenumbruch nicht unterhalb eines Tabelle-Knotens befinden.

Alle Knoten, die innerhalb des Hauptfensters nach einem Kommando mit manuellem Seitenwechsel folgen, werden auf der aktuellen Seite nicht mehr prozessiert. Ein Kommando-Knoten mit dynamischem Seitenumbruch ist damit normalerweise der letzte Knoten im Hauptfenster. Die angelegten Neben-Fenster werden aber auch auf dieser Seite noch ausgegeben.

Die zweite oben genannte Einschränkung bezieht sich explizit nicht auf den Schleife-Knoten: Problematisch für Smart Forms sind in diesem Zusammenhang die Funktionen zur Ausgabetablelle, nicht die eigentliche Schleife über die Daten einer internen Tabelle.

Daraus ergibt sich ein interessanter Anwendungsfall: Nehmen wir an, es soll ein Formular derart erstellt werden, dass für jede Positionszeile eine eigene Formularseite gedruckt werden soll. Angewendet auf unsere Flugrechnung könnte die Aufgabe z.B. lauten, dass für jede Position ein eigenes Ticket auszugeben ist. In diesem Fall muss nach jeder Position auch ein Seitenumbruch erfolgen.

Hier als Vorschlag zwei Lösungsmöglichkeiten:

- ▶ Sie benutzen für die Schleifenabwicklung über die Positionen einen Schleifenknoten, wobei am Ende jedes Durchlaufs der Seitenumbruch über den Kommando-Knoten erzwungen wird.
- ▶ Falls Sie die Ausgaben überwiegend über Nebenfenster organisieren, könnten Sie auch mit einem sehr kleinen Hauptfenster arbeiten, das nach jeder Position komplett gefüllt ist. Das führt automatisch zum gewünschten Seitenumbruch ohne speziellen Kommando-Knoten. In diesem Fall kann im Hauptfenster der Lauf über alle Positionen auch mit einem Tabelle-Knoten erfolgen.

Seitenzähler

Den aktuellen Stand der Seitennummerierung können Sie im Formular über die folgenden Systemfelder abfragen:

SFSY-PAGE	Für die aktuelle Seitennummer
SFSY-FORMPAGES	Für die Gesamtseitenzahl des Formulars
SFSY-JOBPAGE	Für die Gesamtseitenzahl aller Formulare im Druckjob

Über diese Systemfelder können Sie interne Seitenzähler der Ausgabesteuerung abfragen.

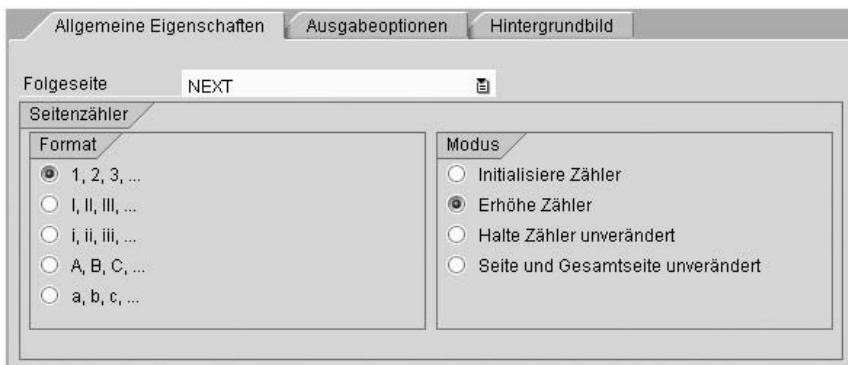


Abbildung 7:16 Eigenschaften der Seitenzähler

Für die internen Seitenzähler können Sie auf der Registerkarte **Allgemeine Eigenschaften** bei den Attributen des Seite-Knotens einige Eigenschaften festlegen (Abbildung 7.16):

- ▶ Im Abschnitt **Format** geben Sie vor, wie die Seitenzahl ausgegeben werden soll (in arabischen Ziffern, römischen Ziffern oder auch als Buchstaben).
- ▶ Im Abschnitt **Modus** legen Sie fest, wie die Werte der verschiedenen Seitenzähler beeinflusst werden sollen.
 - ▶ Die Optionen **Initialisieren** , **Erhöhen** und **Halten** verändern nur die aktuelle Seitenzahl in SFSY-PAGE. Die Gesamtseitenzahlen in SFSY-FORMPAGES bzw. SFSY-JOBPAGES werden unabhängig davon immer um eins erhöht
 - ▶ Die Option **Seite und Gesamtseite unverändert** bewirkt, dass neben SFSY-PAGE auch SFSY-FORMPAGES unverändert bleibt

7.5 Kommando-Knoten

7.5.1 Übersicht

Der Kommando-Knoten enthält eine Zusammenstellung unterschiedlicher Funktionen, die wir teilweise schon an anderer Stelle eingegangen sind.

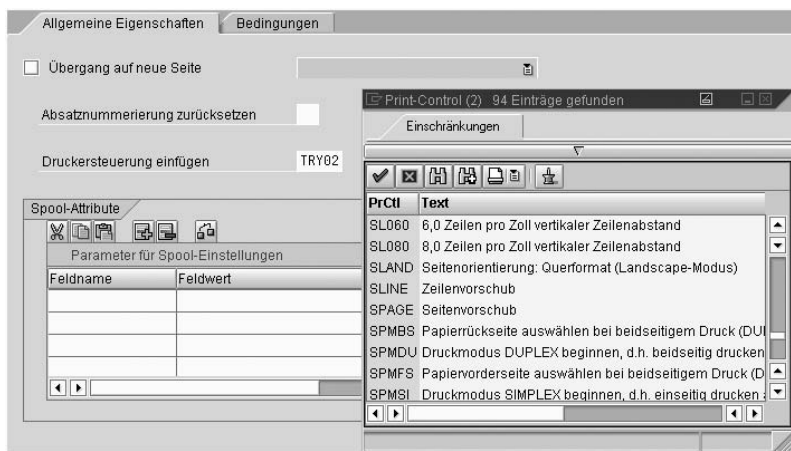


Abbildung 7.17 Kommando-Knoten mit Liste der Print-Controls

Hier zunächst eine Übersicht zu den enthaltenen Funktionen:

▶ Übergang auf neue Seite

Diese Funktion darf nur im Hauptfenster verwendet werden; und dort auch nur außerhalb einer Tabelle. Hierauf sind wir bereits im Abschnitt 7.4 eingegangen.

► **Absatznummerierung zurücksetzen**

Für ein Absatzformat mit Gliederungsfunktionen wird die Nummerierung auf den Startwert zurückgesetzt. Die Gliederungsfunktionen haben wir bei Vorstellung des Style Builders im Kapitel 3 ausführlich erläutert.

► **Druckersteuerung einfügen**

Einfügen eines Print-Controls, um individuelle Funktionen des Ausgabegeräts zu nutzen

► **Vergabe von Spoolattributen**

Für den erzeugten Ausgabeauftrag können Merkmale vergeben werden, über die die betreffenden Aufträge im Spooler auswertbar sind.

Wir wollen uns im Folgenden nur noch mit den beiden letzten Optionen näher befassen.

Übersicht zur Ausgabesteuerung

In den bisherigen Kapiteln des Buches haben wir häufig von der **Ausgabe** des Formulars gesprochen und dabei bislang meist ignoriert, dass diese Ausgabe genau genommen aus zwei einzelnen Schritten besteht. Wichtige Zwischenstation ist dabei der Spooler, der alle Druckaufträge im SAP-System verwaltet:

- Im ersten Schritt wird die Ausgabe als Spoolauftrag an den Spooler übergeben. Der Inhalt des Auftrags ist das Ergebnis der Formularbearbeitung, hat aber noch keine Formatierung, die für das Zielgerät (Drucker, Fax etc.) direkt verständlich ist. Das erzeugte Dokument besitzt zu diesem Zeitpunkt ein geräteunabhängiges Zwischenformat (das *Output Text Format*, kurz OTF-Format).
- Im Spooler werden alle Spoolaufträge gesammelt. Sie können in Ihren Eigenschaften ggf. geändert werden (z. B. durch Zuordnung eines anderen Ausgabegeräts). Nach endgültiger Bestätigung durch den Anwender erzeugt der Spooler den eigentlichen Ausgabeauftrag und sendet ihn zum Ausgabegerät. Dazu wird zunächst mit Hilfe des *Druckertreibers* das bestehende Dokument aus dem OTF-Zwischenformat in ein individuelles Format übersetzt, das vom jeweiligen Zielgerät abhängt.

Die Umsetzung im Schritt 2 ist der Zeitpunkt, bei dem *Print-Controls* zum Einsatz kommen. Sie übersetzen jede der allgemeinen Formatanweisungen des OTF-Formats in konkrete Steuersequenzen des Ausgabegerätes. Die Print-Controls beschreiben damit die eigentlichen Funktionen des Ausgabegerätes.

Bei diesem automatisierten Vorgehen werden nicht wirklich alle Funktionen des Zielgerätes berücksichtigt, sonst müsste nahezu jeder Gerätetyp auch einen eigenen, sehr individuellen Druckertreiber haben. Zusätzlich müsste auch noch der

Form Builder diese individuellen Funktionen unterstützen, was dort wieder zu höherer Komplexität führt

Um trotzdem im Einzelfall auch individuelle Möglichkeiten eines Ausgabegerätes nutzen zu können, bietet Smart Forms den Aufruf von Print-Controls über den Kommando-Knoten.

7.5.2 Print-Controls

Über Print-Controls im Kommando-Knoten erhält das Formular einen direkten Zugriff auf Einzelfunktionen des Ausgabegerätes. Dieser Aufruf bedingt allerdings Abhängigkeiten oder sogar Risiken. Dazu zunächst einige Beispiele:

- ▶ Ein Formular wird dadurch abhängig vom Ausgabegerät; dies kann bei unterschiedlichen Geräten auch zu unterschiedlichen Ergebnissen führen.
- ▶ Welche Funktion sich hinter einem direkt aufgerufenen Print-Control verbirgt, ist für die Prüfroutinen im Formular nicht ersichtlich. Es wird auch keine Überprüfung vorgenommen, ob das verwendete Print-Control beim jeweiligen Ausgabegeräte überhaupt existiert.
- ▶ Die im Print-Control enthaltenen Druckerkommandos dürfen keinerlei Auswirkung auf die aktuellen Druckereinstellungen haben, da der Druckertreiber keine Informationen darüber hat, welche Änderungen durch ein beliebiges Print-Control ausgelöst werden. Der Druckertreiber geht davon aus, dass das Print-Control keine Nachwirkungen auf die danach ausgegebenen Texte und Grafiken hat (ggf. muss ein weiteres Print-Control die Einstellungen wieder zurücknehmen).
- ▶ Wenn die verwendeten Print-Controls den Zeilenvorschub verwenden, wird die automatische Berechnung des Seitenumbruchs nicht mehr funktionieren.

Trotz dieser Risiken: Gerade in speziellen Anwendungsfällen werden Sie über die trickreichen Print-Controls die passende Lösung finden! Deshalb wollen wir in der folgenden Darstellung auch technische Hintergründe behandeln.

Hinweis: Alle verfügbaren Print-Controls, die Sie über einen Kommando-Knoten in das Formular einbinden können, werden zentral über die Transaktion zur Spooladministration verwaltet. Der Aufruf erfolgt vom SAP-Menü über den Pfad **Werkzeuge • CCMS • Spool • Spoolverwaltung SPAD**. Wählen Sie dort unter der Registerkarte **Gerätetypen** die Taste **Print-Control** (allerdings nur verfügbar im Modus der erweiterten Administration).

Print-Controls bestehen aus einem fünfstelligen Kürzel und einer Beschreibung. An den Anfangsbuchstaben im Namenskürzel ist die Gruppierung der Print-Controls entsprechend den Anwendungsfällen erkennbar. Beispiele hierfür sind:

BAR..	Barcodefunktionen
CI...	Einstellungen zum Zeichenabstand
COL..	Einstellungen zur Zeichenfarbe
LI...	Einstellung des Zeilenabstands (Zeilen pro Zoll)
S....	Print-Controls für SAPscript (verwendet im Druckertreiber)
TRY..	Wahl des Papiereinzugschachtes

Innerhalb dieser Gruppen sind jeweils die konkreten Print-Controls angelegt; hier wieder einige Beispiele:

CI010	Zeichenabstand 10 Zeichen pro Zoll einstellen
COL2V	Schrift invers
TRY02	Schacht 2 des Druckers für den Ausdruck wählen

Normalerweise werden bei Umsetzung der Formularausgabe in den Befehlssatz des Ausgabegerätes automatisch auch die passenden Print-Controls verwendet. Verantwortlich für die Umsetzung von Ausgaben zu Smart Forms ist der *SAPscript-Druckertreiber*; er verwendet überwiegend Print-Controls mit den Anfangsbuchstaben 'S' und 'T'. Für die Steuerung von Fonts und Barcodes sind das z. B. die folgenden Print-Controls:

SBPxx	Barcode mit Nummer 'xx' einschalten
SBSxx	Barcode mit Nummer 'xx' ausschalten
SFxxx	Druckerfont Nummer 'xxx' wählen
SLxxx	Zeilenabstand 'xxx' einstellen

Die einzelnen konkreten Druckerfonts bzw. Barcodes pflegen Sie in der SAP-Fontverwaltung (siehe Abschnitt 13.6.6).

Bei Umwandlung des Spoolauftrags in einen Ausgabeauftrag für das konkrete Ausgabegerät wird jedes Print-Control mittels Druckertreiber in die passenden Befehle (ESC-Sequenzen) des jeweiligen Gerätetyps umgewandelt (siehe auch Abschnitt 13.6).

Hinweis: Es kann immer nur ein Print-Control je Kommando-Knoten ausgegeben werden. Im Eingabefeld ist über die Werthilfe eine Liste aller Print-Controls aufrufbar, die in der Spooladministration angelegt worden sind (siehe auch Abbildung 7.17). Es handelt sich hier um eine Gesamtliste: Vergewissern Sie sich, dass ein gewähltes Print-Control bei den verwendeten Ausgabegeräten auch mit der gewünschten Steuerungsfunktion hinterlegt ist.

7.5.3 Attribute zum Spoolauftrag

Wie oben erwähnt, wird bei jeder Formularausgabe zunächst ein Spoolauftrag erzeugt und im Spooler zwischengespeichert. Eine solcher Spoolauftrag enthält dann u. a. auch eine Reihe von Verwaltungsinformationen. In Abschnitt 13.6 werden wir noch ausführlich darauf eingehen.

Sie können zu jedem dieser Spoolaufträge darüber hinaus noch beliebige freie Attribute als zusätzliche Verwaltungsinformationen mitgeben, die dann als individuelle Merkmale z. B. für die Auswertung zur Verfügung stehen. Um die Attribute innerhalb der Spoolverwaltung (SP01) anzuzeigen, wählen Sie dort den Menüpfad **Springen • Auftragsinformationen** und dann die Registerkarte **Freie Attribute** (die allerdings nur existiert, wenn Attribute angelegt sind). Diese Spoolattribute können Sie über einen Kommando-Knoten auch direkt aus dem Formular heraus erzeugen.

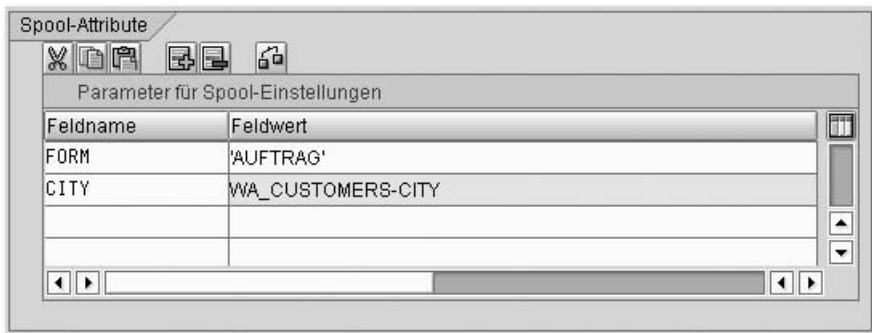


Abbildung 7.18 Kommando-Knoten mit Spoolattributen

Die einzelnen Spoolattribute bestehen aus einem beliebigen Namen für das betreffende Feld und einem entsprechenden Inhalt (Feldwert). In unserem Beispiel sind zwei Attribute definiert:

- ▶ FORM gibt den Inhalt des Formulars wieder, hier mit dem Feldwert AUFTRAG als Hinweis auf eine Auftragsbestätigung
- ▶ Der Parameter CITY enthält die Stadt des Auftraggebers; hier dynamisch definiert über ein Feld, da diese Angabe vom jeweils aktuellen Auftraggeber abhängt.

Eine spätere Auswertung der Spooldatei könnte folgende Abfrage enthalten: Liste alle Auftragsbestätigungen (d.h. Feldname FORM = 'AUFTRAG'), die in eine bestimmte Stadt – z.B. Berlin – gegangen sind (d.h. zusätzlich Feldname CITY = 'BERLIN').

Hinweis: In Release-Stand 4.6C muss bei dynamischer Vergabe des Feldinhalts zusätzlich eine Klammerung des Feldnamens über das &-Sonderzeichen erfolgen; dieser Zustand gilt als Programmfehler (siehe SAPnotes im Anhang).

Eine Anwendung der Spoolattribute kann die Portooptimierung sein: In diesem Fall wird als Attribut eine (verkürzte) Postleitzahl mitgegeben (siehe z.B. Report RSPOPRNT, die Spoolattribute sind in der Datenbanktabelle TSP02A gespeichert).

7.6 Komplexer Abschnitt

In einem **Komplexen Abschnitt** werden die Eigenschaften der Knotentypen **Ordner**, **Schablone**, **Tabelle** und **Schleife** in einem einzigen Knotentyp vereinigt. Über entsprechende Optionen auf der Registerkarte **Allgemeine Eigenschaften** wählen Sie den Knotentyp, der jeweils simuliert werden soll. Es werden also nicht Eigenschaften der verschiedenen Knotentypen kombiniert.

Der Knotentyp **Komplexer Abschnitt** stammt aus ersten Versionen von Smart Forms und sollte heute nicht weiter verwendet werden. Da er im Grunde keine neuen Funktionen bietet, verzichten wir hier auf eine detaillierte Beschreibung.

Sie können eine gewünschte Funktion auch immer über denjenigen Knotentyp realisieren, der dafür eigentlich vorgesehen ist. Die Verwendung der individuellen Knotentypen bringt zudem für die Darstellung einen kleinen zusätzlichen Vorteil: Über das zugehörige Icon im Navigationsbaum erkennen Sie schneller den Knotentyp und damit die grundlegende Funktionalität.

Allerdings waren in der Originalausführung zur Flugrechnung, die wir als Übungsbeispiel verwenden, mehrere Knoten vom Typ **Komplexer Abschnitt** angelegt. Im Übungsbeispiel des Abschnitts 7.2.8 haben wir diese Knoten beispielhaft ersetzt.

7.7 Nachprozessierung

Immer wieder besteht im Rahmen der Formularentwicklung die Anforderung, dass Daten auf der ersten Seite des Formulars auszugeben sind, die eigentlich erst am Ende der Formularausgabe bekannt sind.

Beispiele: Ausgabe von Summenwerten, die erst während der Formularausgabe berechnet werden, wie z.B. bei individuellen Rechnungssummen oder Angaben zum Gesamtgewicht eines Lieferscheins. Häufig sind solche Angaben zusätzlich mit besonderen Ausgabeformatierungen verbunden (etwa Strichcode oder Angaben für Kuvertiermaschinen). Eine weitere Anforderung kann die besondere Kennzeichnung von Durchschlägen sein, wenn ein Formular in mehrfacher Ausführung ausgegeben wird.

In der bisher vorgestellten Ablauflogik zum Formular ist aber festgelegt, dass ein Feld nur korrekt ausgegeben werden kann, wenn sein Inhalt zum Zeitpunkt der Ausgabe schon bekannt ist. Und das ist bei den hier genannten Anforderungen nicht mehr der Fall. Um auch diese Ausgaben zu ermöglichen, sind grundsätzlich mindestens zwei Durchläufe bei der Prozessierung der Formulars erforderlich:

- ▶ Im ersten Durchlauf werden alle benötigten Werte berechnet.
- ▶ Im zweiten Durchlauf stehen diese Werte von Beginn an zur Verfügung und lassen sich dementsprechend ausgeben.

Hinweis: Weiter oben haben Sie bereits die Systemfelder SFSY-FORMPAGES und SFSY-JOBPAGES kennen gelernt (für die Gesamtzahl der Ausgabeseiten zum Formular bzw. Spoolauftrag). Deren Inhalt ist streng genommen auch erst am Ende der Formularprozessierung bekannt. Diese Felder werden aber vom Formularprozessor grundsätzlich als Sonderfall betrachtet und auch erst nachträglich über einen zweiten Verarbeitungsdurchlauf in die Ausgabeseiten eingefügt.

Mit Basis-Release 6.10 wird es möglich sein, beliebige Variablen in einem zweiten Durchlauf auszugeben. Die Steuerung erfolgt über einen speziellen Fenster-Knoten, der zu diesem Zweck als Nebenfenster angelegt sein muss. Für diese Fenster kann dann ein Knotenattribut **Nachprozessierung** gesetzt werden, das für die Prozessierung im zweiten Durchlauf sorgt. Damit sind dann auch die anfangs genannten Beispiele realisierbar (siehe auch Kapitel 12).

In Release 4.6C steht diese spezielle Knoteneigenschaft noch nicht zur Verfügung. Die Logik der Nachprozessierung ist aber bereits implementiert und wird im Prinzip bei der Ausgabe der Gesamtseitenzahl auch schon angewendet. Ab Support Package 12 gibt es deshalb auch für Release 4.6C eine Zwischenlösung, um die gewünschte Nachprozessierung eines gesamten Fensters zu erreichen.

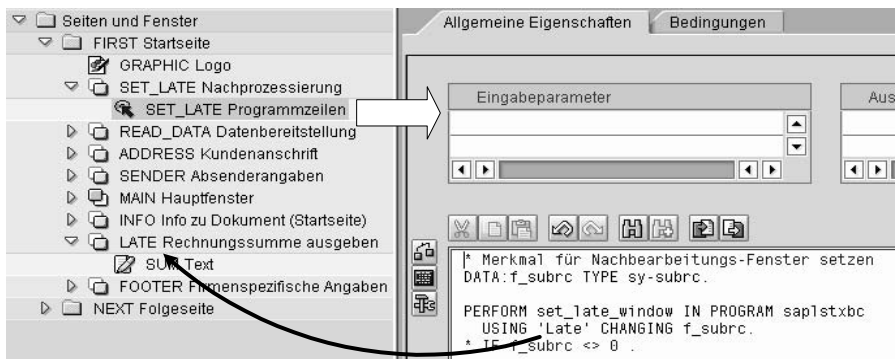


Abbildung 7.19 Nachprozessierung

Bei installiertem Support Package haben Sie die Möglichkeit, das Attribut zur Nachprozessierung eines Fensters über einen zusätzlichen Programm-Knoten zu setzen (statt über die Knotenattribute wie ab Release 6.10). Hier eine kurze Erläuterung zur Vorgehensweise:

1. Legen Sie alle Fenster im Formular an; auch die Nebenfenster, die erst im zweiten Durchlauf prozessiert werden sollen (z. B. mit Namen LATE1, LATE2). In den Unterknoten dieser Nachbearbeitungsfenster können Sie auf alle Felder im Formular zurückgreifen. Ausgegeben werden die Inhalte, wie sie sich am Ende des ersten Durchlaufs darstellen (z. B. mit berechneten Gesamtsummen).
2. Legen Sie zusätzlich auf der ersten Seite und an oberster Stelle im Navigationsbaum ein weiteres Nebenfenster an (wie SET_LATE in obiger Abbildung). Das Fenster wird selbst keine Ausgaben erzeugen.
3. Erstellen Sie statt dessen unterhalb dieses Fenster-Knotens einen Programm-Knoten. Dort muss für alle angelegten Nachbearbeitungsfenster jeweils eine FORM-Routine aufgerufen werden, die das zusätzliche Merkmal setzt. Hier die Codierung für den Fall, dass zwei Fenster (LATE1 und LATE2) erst im zweiten Durchlauf abgearbeitet werden sollen:

```
*&-----&*
* Merkmal für Nachbearbeitungsfenster setzen
DATA: f_subrc TYPE sy-subrc.
```

```

PERFORM set_late_window IN PROGRAM saplstxbc
  USING 'Late1' CHANGING f_subrc.
* IF f_subrc <> 0 .
PERFORM set_late_window IN PROGRAM saplstxbc
  USING 'Late2' CHANGING f_subrc.
*&-----&*

```

Listing 7.2 Nachprozessierung

Für weitere Details siehe die entsprechenden Hinweise im SAPnet (siehe Anhang). Ab Basis-Release 6.10 sollten Sie dann aber wirklich den vorgesehenen Eintrag bei den Knotenattributen des jeweiligen Fensters vornehmen. Löschen Sie bei Releasewechsel auch den zusätzlich angelegten Programm-Knoten.

7.8 Zusammenfassung der Formularprozessierung

In den vorhergehenden Abschnitten haben wir alle Knotentypen und -attribute vorgestellt haben, die Einfluss auf die Prozessierung des Formulars haben. Abschließend wollen wir die Abläufe nochmals aus Sicht der Ablaufsteuerung zur Formularausgabe zusammenfassen. Sie werden auf diese Kenntnisse bei Entwurf eines Formulars aber kaum angewiesen sein.

Wir unterscheiden bei der Prozessierung zwei Ebenen:

- ▶ Die Steuerung der seitenweisen Ausgabe
- ▶ Die Prozessierung der einzelnen Knoten innerhalb einer Seite

In einer ersten Näherung können Sie sich an den Zweigen des Navigationsbaums im Form Builder orientieren:

- ▶ Die Ausgabe beginnt mit der Startseite (also der ersten Seite im Navigationsbaum), es sein denn, bei den Parametern der Formularschnittstelle ist ausdrücklich eine andere Startseite vorgegeben.
- ▶ Auf jeder Entwurfsseite werden die Fenster entsprechend der Reihenfolge im Navigationsbaum durchlaufen. Die Prozessierung jedes einzelnen Fensters erfolgt dabei wieder in der Reihenfolge der zugeordneten Unterknoten.

Es werden solange Informationen zu einem Nebenfenster ausgegeben, bis einer der beiden folgenden Fälle eingetreten ist:

- ▶ Alle Knoten sind im Fenster abgearbeitet.
- ▶ Das Fenster ist auf der Ausgabeseite vollständig gefüllt.

Steuerung der Seitenausgabe

Wenn die aktuell bearbeitete Entwurfsseite ein Hauptfenster enthält, ist dessen Verhalten dafür verantwortlich, ob von der Ausgabesteuerung eine weitere Ausgabeseite angelegt wird oder ob die aktuelle Seite gleichzeitig die letzte Seite ist. Die Ablaufsteuerung zur Seitenausgabe wird aktiv, wenn eine der folgenden Bedingungen erfüllt ist:

- ▶ Alle im Hauptfenster enthaltenen Knoten sind abgearbeitet.
- ▶ Es sind noch **nicht** alle enthaltenen Knoten abgearbeitet, aber der Platz auf der Ausgabeseite ist erschöpft.
- ▶ Über einen Kommando-Knoten wurde ein manueller Seitenwechsel ausgelöst.

In allen Fällen wird zunächst die Ausgabe der aktuellen Seite beendet, wobei insbesondere die noch verbleibenden Nebenfenster ausgegeben werden, weil sie im Navigationsbaum hinter dem Hauptfenster folgen. In den beiden letzten Fällen wird darauf folgend automatisch eine neue Ausgabeseite angelegt:

- ▶ Ist der Platz auf der Ausgabeseite erschöpft, so ist dies eine zugeordnete statische Folgeseite.
- ▶ Wird über einen Kommando-Knoten ein manueller Seitenwechsel ausgelöst, so ist dies eine dynamische Folgeseite, wie sie im auslösenden Kommando-Knoten eingetragen ist.

Da im Fall, dass alle im Hauptfenster enthaltenen Knoten abgearbeitet sind, keine weitere Ausgabeseite folgt, ist dann auch die gesamte Prozessierung des Formulars beendet.

Bei einer Entwurfsseite ohne Hauptfenster wird die Folgeseite immer aus der Angabe zur statischen Folgeseite gelesen: Ein manueller Seitenumbruch mit dynamischer Folgeseite ist in diesem Fall nicht möglich, weil ein entsprechender Kommando-Knoten dort nicht ausgeführt werden kann (ein solcher ist nur in einem Hauptfenster erlaubt). Falls keine Angabe zur Folgeseite existiert, ist die aktuelle Ausgabeseite gleichzeitig auch die letzte Seite.

Prozessierung der einzelnen Knoten

Die Knoten im Navigationsbaum unterscheiden sich in ihrem Typ und haben dementsprechend unterschiedliche Eigenschaften (Art der Abarbeitung, Anzahl der Unterknoten etc.). Allen Knoten können Ausgabeoptionen und Bedingungen hinterlegt sein. Die Abarbeitung eines einzelnen Knotens enthält folgende Schritte:

- ▶ Ist der Knoten mit einer Bedingung versehen, so wird er mitsamt seinen Unterknoten nur abgearbeitet, wenn diese Bedingung erfüllt ist.
- ▶ Ist die Bedingung erfüllt oder überhaupt keine Bedingung hinterlegt, so wird die Aktion ausgeführt, die dem jeweiligen Knoten zugeordnet ist (z. B. Ausgabe eines Textes, Abarbeitung von Programmzeilen). Dabei werden auch die eingestellten Ausgabeoptionen berücksichtigt. Beispiele hierfür sind:
 - ▶ Die Ausgabe von Rahmen und Schattierungen
 - ▶ Die Zuordnung von Zellen einer Schablone oder Ausgabetablelle
 - ▶ Die Angabe eines Stils ermöglicht individuelle Absatz- und Zeichenformate.
 - ▶ Bei gesetztem Seitenschutz werden alle Ausgabeninhalte des Knotens bzw. seiner Unterknoten auf der gleichen Seite ausgegeben.
- ▶ Besitzt der jeweilige Knoten eigene Unterknoten, so werden als nächstes diese Unterknoten abgearbeitet. Die Reihenfolge der Abarbeitung ist vom Typ des Knotens abhängig; möglich sind diese Reihenfolgen:
 - ▶ Sequenziell von oben nach unten
 - ▶ Nur zu bestimmten Zeitpunkten
 - ▶ Als Alternative aufgrund von logischen Abfragen
 - ▶ Wiederholt bei Steuerung über eine Schleife

Index

A

ABAP-Datentypen 160
ABAP-Debugger 470
ABAP-Dictionary 161, 162, 171, 231, 480
ABAP-Editor 47, 460
 Anweisungsmuster 234, 273
 Bearbeitungsfunktionen 232
 Besonderheiten unter Smart Forms 236
 Eingebaute Hilfen 235
 Syntaxcheck 234
ABAP-Grundlagen 238
 Selektionsbild 261, 275
ABAP-Programmierung 23
 Programmcode im Formular 229
Ablauflogik 19
Absatzformat 57, 70
 Absatznummerierung zurücksetzen 218
 Namenskürzel 62
 pflegen 62
 Ränder und Abstände 63
Abschnitt 204
Adressaufkleber 339
Adressen 130
 Adresse-Knoten 131
 über Funktionsbaustein einbinden 248
 ZAV 130, 250
 ZAV Customizing 134
Alternative-Knoten 188, 213
Änderungsmanagement 88
Anwendungsprogramm 259
Arbeitsbereich 189
Archivierung 319
Aufbereitungsoptionen 149
Aufzählungen 65
Ausgabeauftrag 220, 445
Ausgabebereich 20, 92, 99, 105, 116, 137
Ausgabegerät 447
Ausgabemedien 319, 437
 Archivierung 337
 Drucker 319, 428
 E-Mail 319, 406
 HTML 381

Telefax 327

XML 329

XSF 387

Ausgabeoptionen 32, 53, 57, 94, 110, 140
 Rahmung und Schattierung 118
 Zeile und Spalte 111
 zur Ausgabetable 203
Ausgabeseite 21
Ausgabetable 96, 103, 201
 Layout 202
 Registerkarte Tabelle 202
 Schutz vor Umbruch 203
 Zeilentypen 202
 Zelle im Unterknoten zuordnen 203
Ausnahmen 176

B

Barcode 69, 452, 456
BCI (Business Communication Interface) 319
BDS (Business Document Service) 420
Bedingungen 32, 53, 188, 210
Beispielformular 25, 37
Benutzerparameter 440
Bereichsmenü 55
 Administration 55
 SAPscript 56
Bezugstyp 160
Buchaufbau 14
Business Workplace 320

C

Composer 186, 447
control_parameters 292

D

Daten 22, 23, 143
 ABAP Datentypen 160
 Arbeitsbereich 189, 194, 198, 278
 Bezugstyp 160, 172
 Datenbanksystem 153
 Datenbanktable 161, 164, 188, 243, 481
 Datendefinition 159

- Datenobjekt 156
 - Datentypen 153, 159, 481
 - formularintern 167
 - Dictionary 161
 - Feld 154
 - Feldleiste 154, 284
 - Feldsymbole 177, 194
 - Interne Tabelle 155, 189, 193
 - komplex strukturiert 156, 162
 - Quelle 143, 168, 243
 - Strukturen 144, 153
 - Transparente Tabellen 153, 164
 - Datenbanktabelle 481
 - Datenbeschaffung 24, 188, 229
 - Performance 230, 284
 - Wege 230
 - Datendefinition 144, 152, 254
 - Globale Daten 177
 - Konstante 178
 - mit LIKE 159
 - mit TYPE 159
 - Typisierung 156
 - unter ABAP 240
 - Variablen 156, 178
 - Vorschlagswert 178
 - Datenelement 162
 - Datum 149, 179
 - Formatierung 142
 - Dokumentation 88
 - Download 333
 - Drag&Drop 208
 - Druckerkommando 219
 - Druckersteuerzeichen 218
 - Druckertest 413
 - Druckertreiber 219, 447, 448
 - Druckjob 437
 - Druckmodus 99
 - Druckprogramm 259
 - Drucksystem 437
 - Durchschläge 223, 280, 286
- E**
- Einzelprüfung 39
 - Elementares Feld 154
 - E-Mail 319
 - Entwicklerregistrierung 27
 - Entwicklungs-klasse 37, 57
 - Entwicklungswerkzeuge 457
 - Entwurfsseite 96
 - Ereignisknoten 40, 74, 76, 140, 197, 204
 - Fußbereich 39
 - Zwischensumme 205
 - Etikettenausgabe 347
 - Etikettendruck 338
 - Ausgabeformatierung 344
 - Basisformular anlegen 340
 - Basisformular einbinden 341
 - Datendeklaration 342
 - im Formular einrichten 341
- F**
- Farbdrucker 69
 - Fax-Ausgabe 319
 - Fehlerbehandlung 305
 - Ausnahmen 176, 308
 - Ausnahmen im Formular 313
 - Fehlermeldung 306
 - im Rahmenprogramm 308
 - interne Fehlerkonstanten 312
 - internes Protokoll 306, 310
 - Laufzeitfehler 305
 - Standard-Meldungen 307
 - Systemfelder 305
 - Fehlerliste 85
 - Fehlermeldungen 39, 85
 - Felder 86, 187
 - als Knotenattribut 148
 - im Debugger 475
 - im Text-Knoten 145
 - mit Aufbereitungsoptionen 149
 - Syntax 168
 - Systemfelder 179
 - Feldleiste 154
 - Feldliste 72, 145
 - Feldsymbole 177
 - Fenster 20
 - Fenster-Knoten
 - Abschlußfenster 378
 - anlegen 100
 - auf mehreren Seiten 101
 - Hauptfenster 100
 - Kopienfenster 377
 - Nebfenster 100
 - Flugdatenmodell 25
 - Flugrechnung 26
 - Arbeitskopie 36

- Folgeseite 186
- Fontpflege 452
- Fonts 55, 69, 452
 - TrueType-Fonts 455
- Form Builder 31, 71, 141
 - Drag&Drop 47, 187, 208, 233
 - Grundeinstellungen 77
 - Navigationsbaum 185, 209, 213
 - Teilwerkzeuge 71
- Form Painter 31, 72, 79, 101
- Formroutinen 231
- Formular 19
 - Ablauflogik 19, 21, 185, 214
 - aktivieren 38, 84
 - angelegt 31
 - ausgeben 24, 35
 - Datenflussanalyse 86
 - Design 26
 - Dokumentation 88
 - dynamische Ausgabe 188, 204
 - Erstseite 21
 - Folgeseite 21, 186, 214
 - Formularlogik 185
 - generieren 31, 56, 88
 - Gesamtprüfung 38, 86, 200, 237
 - Herunter-/Hochladen 375, 394
 - kopieren 36
 - Layout 19, 20, 91
 - Name 56
 - Prozessierung 187, 223, 225
 - prüfen 84
 - Seitenränder 448
 - Seitenumbruch 214
 - sichern 84
 - Status 34, 38, 84
 - testen 34, 38, 84, 87, 464
 - zurück zu aktiver Version 84
- Formularattribute 34, 38, 93
- Formularlogik 185
- Formular-Prozessor 186
- Formularschnittstelle 34, 169, 440
 - Anzahl Kopien 287
 - Ausnahmen 176
 - Export-Parameter 173, 301
 - Feldsymbole 177
 - Import-Parameter 170, 291
 - obligatorische Parameter 170
 - Standardparameter 170, 174, 274, 290

- Tabellen 175
- Typen 177
- Wertübergabe 174
- zentrale Datenübergabestruktur 171

- Formular-Templates 487
- Function Builder 34, 87, 464
- Funktionsbaustein 23, 31, 56, 87, 88
 - Eigenschaften 465
 - Name feststellen 271
- Funktionstasten 18
- Fußbereich 204

G

- Gerätetyp 448, 449
- Gesamtprüfung 38
- Gesamtsummen 205
- Gliederung 65
- Globale Daten 177
- Globale Definitionen 177
 - Formroutinen 231
 - Globale Daten 177
 - Initialisierung 231
 - Typen 167, 231
- Globale Einstellungen 34, 51, 72, 93
- Grafik 55
 - Auflösung 42
 - BMP-Format 420
 - im Form Painter 80
 - importieren 422
 - Klassifikation 423
 - Performance Druckausgabe 139
 - SAPSCRIPT_SMALL_DOG 41
 - selektieren 41, 136
 - TIF-Format 420
 - Verwaltung 420
- Grafik-Knoten 41, 135
- Ausgabebereiche 137

H

- Hauptfenster 22, 99, 104, 186, 189
 - Breite 100
- Hintergrundbild 81, 83, 135, 137
- HTML-Ausgabe 319
- Hyperlink 406

I

- Include-Text 121
- Dynamische Textnamen 124

Objekttyp 124
Standardtext 125
Text-ID 124
Textschlüssel 121
Inline-Editor 44, 72, 119, 403
Interne Tabelle 155
interne Tabelle
 mit Kopfzeile 175
Internet-Anwendungen 379, 386
ITF Textformat 419

K

Knoten 24, 32
 Einzelprüfung 39, 85
 im Funktionsbaustein 469, 479
 kopieren 46, 75
 Kurzbezeichnung 75
 Nachfolgerknoten 73
 über mehrere Seiten 75
 verschieben 46, 75
 verwalten 46
 Vorgängerknoten 73
Knotenattribute 32, 53, 76
 statisch/dynamisch 76
Knotentypen
 elementar 52, 115
 Grafische Symbole im Navigationsbaum 73
 mit Layouteigenschaften 51, 116
 Übersicht 50
 zur Ablaufsteuerung 52
Kommando-Knoten 188, 217
 Absatznummerierung 218
 Druckersteuerzeichen 218
 dynamischer Seitenumbruch 215
 Spoolattribute 218, 221
Komplexer Abschnitt 191, 193, 202, 222
Konstante 178
Kontextmenü 46, 74
Kopfbereich 204

L

Laufzeitfehler 305
Layout 19
LIKE 159
Linie 95
Logische Abfragen 210
 Alternativen 213

Ausgabezeitpunkte 212
Bedingungen 210
Logische Ausdrücke 211
Lokales Objekt 37

M

Makro 314
Maßeinheiten 62, 78, 93
Meldungen 38
Mengenangaben 152, 180
Menüweg 18, 55
Migration 349
Muster 109
Musterformular 26, 36
mySAP CRM 326, 407

N

Nachprozessierung 223, 378
Nachrichtenfindung 424
 Customizing 425
 Konditionstechnik 424
Nachrichtensteuerung 423
 NAST 264, 433
Navigationsbaum 32
 Bearbeitungsfunktionen 74
 Grafische Knotensymbole 73
 Zweige 73
Nebenfenster 22, 99, 104, 186
Nummerierung und Gliederung 65

O

Object Navigator 457
Objektkatalogeintrag 37, 57
Ordner-Knoten 111, 139, 204
 Übungsbeispiel 141
OTF Ausgabeformat 303, 326, 447
output_options 292

P

Papierfach/einzug 98
PDF Ausgabeformat 301
PDF-Format 319
Portooptimierung 222
Preconfigured Smart Forms 487
Print-Controls 98, 218, 219, 450
Programm-Knoten 180, 230
 Adressbeispiel 248
 Datenbeschaffung 188

- Editorfunktionen 232
- Ein-/Ausgangsparameter 237
- Nachprozessierung 224
- Prüfen 234, 237
- Summenbildung 254
- Prozessierung 73
- Prüftabelle 183

Q

- QM Formulare einbinden 435

R

- Rahmen 94
- Rahmenprogramm 19
 - aktivieren 49
 - anlegen 26, 47
 - ausführen 35, 49
 - Ausgabe über Schleife 277
 - Datenbeschaffung 188, 230, 268, 276
 - Datenbeschaffung Lieferschein 283
 - Formularname 268
 - Funktionsbaustein aufrufen 271, 272
 - Funktionsübersicht 259
 - im Formular verwendete Parameter 237, 280
 - kopieren 48
 - mehrere Formulare 283
 - Nachrichtensteuerung 264
 - Name des Funktionsbausteins 271, 274
 - prüfen 49
 - Spoolsteuerung 297
 - Standardparameter und Nachrichtensteuerung 296
 - Übungsbeispiel 274, 283, 286, 297
- Rechnungsformular 20
- Referenzfeld 180
- Reporting 35
- Ressourcenname 98

S

- SAP Easy Access 33
- SAPconnect 319
- SAP-Ikonen 410
- SAPscript 27, 56
 - Editor 407
 - Formulare migrieren 349, 352
 - Kommandos 358

- Kompatibilität 27
- Schnittstellenparameter 368
- Standardtext 33
- Stil 30, 58
- Stile migrieren 351
- Symbole 357
 - Zeileneditor 29
- SAPscript-Text 121
- SAP-Symbole 410
- Schablone 339
- Schablone-Knoten 81, 103, 338
 - anlegen 105
 - Breite 106, 109
 - kombinieren 112
 - Layout 103
 - Muster 109
 - prüfen 106, 110
 - Übungsbeispiel 106
 - Unterknoten 110
- Schattierung 94
- Schleife 189
 - ABAP-Programmcode 199
 - Ausgabe über Schleife 198
 - Einzelnen Datensatz lesen 198
 - Ereignisknoten 197
 - Gruppenstufe 197
 - Interne Tabelle mit Kopfzeile 194
 - Registerkarte Daten 192, 199
 - Sortieren und Gruppieren 196
 - Sortierung 189
 - Übungsbeispiel 200, 210
 - von Daten trennen 193
 - WHERE-Bedingung 189, 195, 198
 - Zeilennummer 195
 - Zwischenwerte 189
- Schleife-Knoten 188, 190, 205
- Schnelleinstieg 19, 26
- Schrift 64, 69
 - unterstreichen 69
- Schulungsbeispiel 25
- Seite-Knoten 96
 - anlegen 97
 - Attribute 97
 - Entwurfsseite 21
 - Hintergrundbild 135
 - Seitenzähler 217
- Seiten und Fenster 33, 72
- Seitenformat 93, 98

- Seitennummer 179
 - Seitenränder 448
 - Seitenumbruch 100, 204
 - automatische Berechnung 219
 - dynamisch 186
 - manuell 97, 186
 - Seitenzahl 148
 - Seitenzähler 216, 223
 - SF_EXAMPLE_01 36
 - Smart Forms 17
 - Eröffnungsbild 31, 56
 - Werkzeuge 55
 - Smart Forms-Templates 487
 - Smart Style 19, 44
 - Sonderzeichen 410
 - Spoolattribute 221
 - Spoolauftrag 221, 298, 330, 440
 - Status 442
 - Spooldialogbild 298, 439
 - Spoolsteuerung 438, 441
 - Anzahl Exemplare 287
 - Spoolsystem 437
 - Spoolverwaltung 221, 344, 414, 438, 445
 - Gerätetypen 219
 - Print Controls 219
 - Spoolauftrag 220
 - Sprache 141, 402
 - bei Formularausgabe 142
 - Standardabsatz 61
 - Standardtext 26, 27
 - anlegen 29
 - kopieren 28
 - Stil 19, 31, 44, 57, 89, 94
 - Absatzformat 115, 123
 - aktivieren 60
 - in Formular einbinden 60
 - Kopfdaten 57, 61
 - Namen ändern 60
 - pflegen 60
 - Standardabsatz 61
 - Varianten 60
 - Zeichenformat 115
 - Strichcode 69
 - Striche 110
 - Struktur 162
 - Style Builder 57
 - Stilbaum 59
 - Systemfelder 179, 305, 379
- T**
- Tabelle als Begriff 104
 - Tabelle-Knoten 81, 190, 205
 - anlegen 207
 - Ausgabetabelle 201
 - Layout 103
 - Muster 109
 - ohne Datenzugriff 112
 - Registerkarte Tabelle 202
 - Übungsbeispiel 205
 - Tabellentyp 162
 - Table Painter 81, 104, 108
 - Aufruf 82
 - Mausfunktionen 82
 - Raster 83
 - verlassen 83
 - Vordruck und Hintergrundbild 83
 - Tabulatoren 64, 81
 - Textablage 403, 418
 - Textart 416
 - Textbaustein 31, 125
 - Dynamischer Textname 127
 - in Formular einbinden 125
 - Neuanlage und Pflege 128
 - Text übernehmen 127
 - Überungsbeispiel 129
 - Übungsbeispiel 126
 - Verwendeter Stil 127
 - Textbearbeitung 403
 - Texteditor 44, 403
 - Textfluss 64
 - Text-ID 28, 415
 - Text-Knoten 115, 403
 - Bezug zu Ausgabebereich 116
 - Hyperlink 406
 - Include-Text 121
 - Rahmen und Schattierung 118
 - Sprache 142
 - Textbaustein 125
 - Textelement 119
 - Textfluss 116
 - Textformatierung 115
 - Texttypen 117
 - Übungsbeispiel 120
 - Textobjekt 415
 - Textverwaltung 415
 - Trace 317
 - Transaktion 55

Transportauftrag 56, 398, 422
Transport-Organizer 398
TYPE 159
Typisierung 156

U

Überschriften 204
Übersetzung 399
Übungsbeispiel
 Ausgabe in Hauswährung 257
 Durchschläge 286
 Fenster-Knoten 102
 Flugrechnung überarbeiten 205
 Flugverbindungen 283
 Gliederungen 66
 Ordner-Knoten 141
 Programm-Knoten 248
 Rechnung an mehrere Kunden 274
 Rückgabe im OTF Format 303
 Schablone-Knoten 106
 Schleife 200
 Standardparameter der Formular-
 schnittstelle 297
 Stilinhalte 70
 Summenbildung 254
 Textbaustein 126, 129
 Text-Knoten 120
URL einfügen 407

V

Variablen 156
Verwendungsnachweis 178
Vordruck 103

W

Währungsbeträge 152, 180
Warnmeldungen 38, 39, 85
Web Application Server 373, 386, 393
Web-Formulare 379, 394
Werkzeuge 24

X

XML-Ausgabe 319
XSF 24, 94
XSF-Ausgabe 329

Z

Z_SF_EXAMPLE_01 37
ZAV 130
Zeichenformat 57, 68, 71
Zeichensatz 413
Zeilenabstand 64
Zeilentypen 81, 104, 107, 202
Zeitpunkte 204
Zwischenablage (erweitert) 77
Zwischensummen 205