



Software

RW-Buch
Hinweise zur
Datenübergabe gemäß GDPdU

Copyright © 2023 Wachtmann Computer-Service

Wachtmann Computer-Service
Im Kohlpott 5, D-32120 Hiddenhausen
Tel: 0 52 21 – 6 71 40 Fax: 0 52 21 – 6 73 58
Internet: www.rwsoftware.de
Email: post@rwsoftware.de

Inhaltsverzeichnis

1. Vorwort.....	3
2. Allgemeines.....	4
2.1. Aufruf in RW-Buch.....	4
2.1.1. Zeitraum einstellen.....	4
2.1.2. Betriebsprüfung (Datenträgerüberlassung).....	4
2.1.3. Zusätzlich Daten auf externes Laufwerk kopieren.....	4
2.1.4. Speicherort der Archive.....	5
2.2. Verarbeitung in 2 Stufen.....	6
2.3. Optional – Daten auf externes Laufwerk kopieren.....	7
3. Hinweise zum Datenaufbau.....	9
3.1. Buchungen.....	9
3.1.1. Datengrundlagen.....	9
3.1.2. Datenstrukturen.....	10
3.2. Konten und Kontoblätter.....	11
3.2.1. Datengrundlagen Konten.....	11
3.2.2. Datenstrukturen Konten.....	12
3.2.3. Datengrundlagen Kontoblätter.....	13
3.2.4. Datenstrukturen Kontoblätter.....	14
3.3. Personenkonten und Personenkontoblätter.....	15
3.3.1. Datengrundlagen.....	15
3.3.2. Datenstrukturen Personenkonten.....	16
3.3.3. Datenstrukturen Personenkontoblätter.....	16
3.4. Buchungshistorie.....	17
3.5. Zusammenfassung Datenstrukturen.....	18
3.6. Dateien nach dem Beschreibungsstandard.....	19

1. Vorwort

Aufgrund gesetzlicher Vorschriften stellen wir hiermit einige ergänzende Informationen zum Datenaufbau und zu den Datenstrukturen bereit, die für den Datenzugriff gemäß GDPdU bzw. nach den GoBD verwendet werden.

Ab RW-Buch 14 wurde eine Archivierungsfunktion erweitert, die Daten in einer MS-Access-Datenbank jahresweise bereitstellt, falls das Datenformat RW-Buch-Datenbank gewählt wurde. Die Tabellen enthalten alle Daten aus der Buchhaltung für das jeweilige Jahr. Die Tabellen und Spalten sind in der Regel selbsterklärend. Folgende Ausführungen beziehen sich auf Verknüpfungen zwischen Buchungen und Splittbuchungen sowie zwischen Konten und Kontenwerte.

Ab der Version 18 wurden die Tabellen nochmals um kleinere Anpassungen erweitert, ab Version 19 von RW-Buch sind einige weitere Tabellen bzw. Abfragen hinzugekommen. In Version 20 sind einige weitere Datenfelder hinzugekommen, die für einige Tabellen ergänzt wurden, ebenso in Version 22 mit der Jahresanpassung für 2012.

Ab RW-Buch 27 sind grundlegende Erweiterungen in Bezug auf die neuen Buchhaltungsregeln nach GoBD erfolgt, die auch für den Datenzugriff nach GDPdU relevant sind. Ab der Version 27 wird zusätzlich eine Buchungshistorie geführt. Die Daten der Historie werden ebenfalls in die Datenaufbereitung einbezogen. Außerdem steht ab Version 27 ein Aktionsprotokoll zur Verfügung, das im Rahmen des internen Kontrollsystems geführt wird.

Ab RW-Buch 34 wurden weitere umfangreiche Änderungen für die Datenträgerüberlassung nach dem Beschreibungsstandard vorgenommen. Das Format mit einer Datei index.xml mit Inhalten in CSV-Dateien wurde erweitert. Diese Dateien werden jetzt zusätzlich zu der Access-Datenbank erzeugt. Die Tabellen aus der Access-Datenbank werden dabei in CSV-Dateien umgewandelt, die Beschreibung der einzelnen Dateien und Datenfelder erfolgt in der Datei index.xml, die nach dem Beschreibungsstandard aufgebaut ist. Dadurch sind diese Daten in der Regel einfacher für den Betriebsprüfer zu verarbeiten.

Die Datenfelder, die mit den Buchstaben OP beginnen, werden nur in der größeren Version RW-Buch OP eingesetzt, in der Grundversion sind diese Felder nicht vorhanden.

Falls Sie eine ältere Version von RW-Buch einsetzen, empfehlen wir den Umstieg auf die momentan aktuelle Version 34. Umstiegsangebote finden Sie auf unserer Webseite.

2. Allgemeines

2.1. Aufruf in RW-Buch

Der Aufruf erfolgt in RW-Buch über Stammdaten / Archivierung. Folgender Bildschirm erscheint:

Archivierungsfunktion

Zeitraum für die Archivierung einstellen

Für das Geschäftsjahr plus Folgejahre

1. Betriebsprüfung (Datenträgerüberlassung) 2. Interne Zwecke

Firma	Ort	Hinweis
Testfirma	Mühlheim	Hinweistext für die Datenüberlassung

Für eine Betriebsprüfung können für das Geschäftsjahr und maximal 2 Folgejahre die Dateien für eine Datenrägerüberlassung erzeugt werden. Beachten Sie bitte auch die mitgelieferten PDF-Hinweise (GDPdU-Erläuterungen.pdf). Die Verarbeitung erfolgt in 2 Schritten:

- Es wird zuerst eine Access-Datenbank (Access 97) erzeugt, die nur Daten des jeweiligen Jahres enthält, außerdem diverse weitere Tabellen, z.B. für Kontoblätter etc.
Die Datei wird im Ordner: C:\Users\User\RWBUWIN34OP\Archiv angelegt u. mit der MandantNr u. Jahreszahl versehen.
- Die in Schritt 1 erzeugten Tabellen werden dann nach dem Beschreibungsstandard für die Datenträgerüberlassung in einer Datei index.xml u. diversen CSV-Dateien umgewandelt.
Die Eingaben Firma, Ort, Hinweis werden zusätzlich in der index.xml hinterlegt.
Die Dateien werden im Ordner: C:\Users\User\RWBUWIN34OP\Archiv\Mandant<MandantNr>\<Jahreszahl> gespeichert.

Zusätzlich Daten auf externes Laufwerk kopieren

Archiv erstellen Abbruch

2.1.1. Zeitraum einstellen

Im oberen Bereich **1** können Sie den Zeitraum für die Archivierung einstellen. Sie geben das Geschäftsjahr an und können außerdem bis maximal 2 Folgejahre angeben. Für diesen Jahresbereich werden die entsprechenden Daten aufbereitet.

2.1.2. Betriebsprüfung (Datenträgerüberlassung)

In der Mitte **2** wählen Sie im Karteidialog Betriebsprüfung (Datenträgerüberlassung) aus. Auf der Seite können Sie einige allgemeine Hinweise eintragen, die in die Datei index.xml eingesetzt werden.

2.1.3. Zusätzlich Daten auf externes Laufwerk kopieren

Über diese Option unten links **3** können Sie einstellen, dass die erzeugten Dateien im Anschluss auf ein externes Laufwerk kopiert werden sollen. Vor der Kopieraktion wird das Laufwerk über einen Windows-Dialog abgefragt. Hier könnte z.B. ein USB-Laufwerk gewählt werden, bitte sorgen Sie vorher dafür, dass ein Laufwerk für die Kopieraktion verfügbar ist, ein Schreibschutz bei einem USB-Stick muss z.B. vorher aufgehoben werden.

2.1.4. Speicherort der Archive

Die Speicherung der Dateien erfolgt im eingestellten Datenlaufwerk im Ordner Archiv. Falls der Ordner im Datenlaufwerk nicht vorhanden ist, wird dieser erzeugt. Z.B.:

Datenlaufwerk: C:\RWBWIN
Archivordner: C:\RWBWIN\Archiv

Damit die einzelnen Jahre exakt unterschieden werden können, erfolgt eine Umbenennung der RW-Buch-Datenbank nach der Aufbereitung der Daten. Dabei wird folgendes Schema angewandt:

Jahr<Archivierungsjahr>RWBU<MandantNr>.MDB

z.B. Mandant 5 im Datenlaufwerk C:\RWBWIN\Daten wird für das Archivierungsjahr 2019 zu:

C:\RWBWIN\Daten\Archiv\Jahr2019RWBU005.MDB

Für die Jahresdaten nach dem Beschreibungsstandard werden Mandanten- und Jahresordner erzeugt:

Mandantenordner: C:\RWBWIN\Archiv\Mandant<MandantNr>
Jahresordner: C:\RWBWIN\Archiv\Mandant<MandantNr>\<Jahreszahl>

Falls der Mandant 5 für die Jahre 2019 bis 2021 verarbeitet werden sollte, würden sich dafür folgende Ordner als Beispiel ergeben:

C:\RWBWIN\Daten\Archiv\Mandant005\2019\
C:\RWBWIN\Daten\Archiv\Mandant005\2020\
C:\RWBWIN\Daten\Archiv\Mandant005\2021\

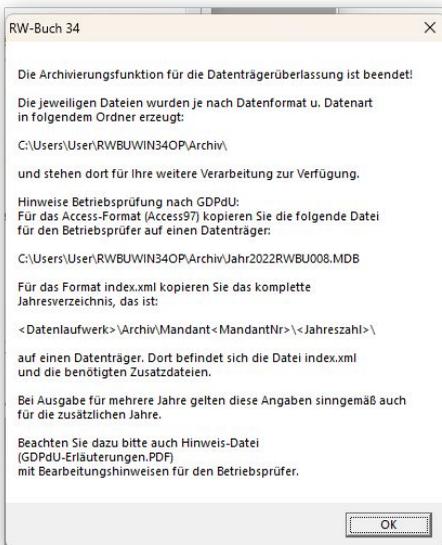
In jedem dieser Ordner werden diverse Dateien nach dem Beschreibungsstandard erzeugt.

2.2. Verarbeitung in 2 Stufen

Die Verarbeitung der Daten erfolgt in 2 Stufen durch den Befehlsschalter 'Archiv erstellen':

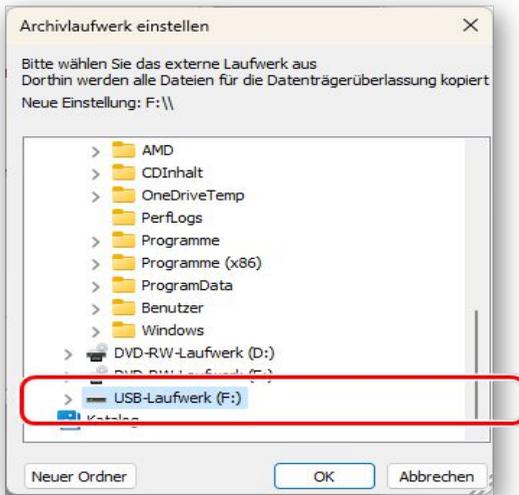
- **Jahresdaten in einer RW-Buch-Datenbank (Access97-Format)**
Bei der Aufbereitung der Daten wird zuerst die RW-Buch-Datenbank als Access97-Datenbank erzeugt, die nur die Daten für das jeweilige Jahr enthält. Die Benennung und Speicherort der Datei ergibt sich aus dem vorherigen Kapitel. Detaillierte Hinweise zur Datenaufbereitung finden Sie weiter hinten.
- **Jahresdaten nach dem Beschreibungsstandard**
Hier werden in den oben beschriebenen Mandanten- und Jahresordnern die Datei index.xml und die diverse CSV-Dateien erzeugt. Außerdem ist die Datei gdpdu-01-08-2002.dtd enthalten, die wir dorthin kopieren. Darin sind die Beschreibungsvorschriften enthalten.

Nach der Verarbeitung erscheint ein Hinweisfenster, der Ihnen die Aktionen nochmals anzeigt:

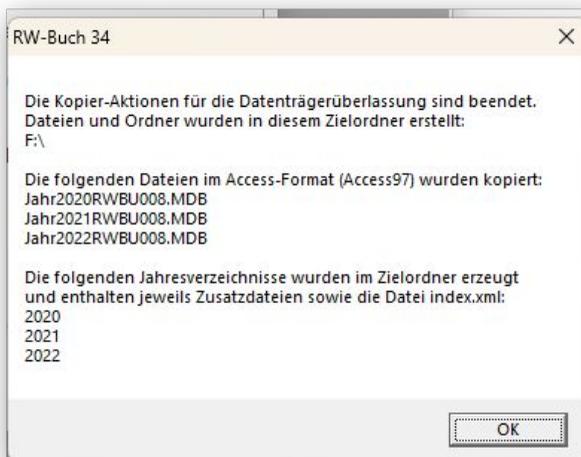


2.3. Optional – Daten auf externes Laufwerk kopieren

Falls die Option für das Kopieren auf ein externes Laufwerk eingeschaltet wurde, kommt im Anschluss an die Erzeugung der Jahresdaten eine Abfrage, auf welches Laufwerk die Dateien kopiert werden sollen:

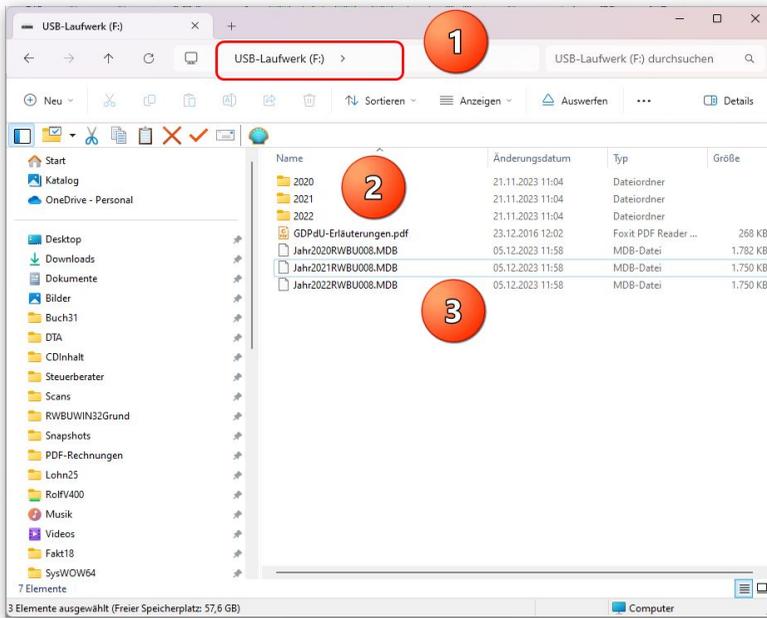


Nach der Kopieraktion kommt ein weiteres Hinweisfenster mit der Angabe, was genau kopiert wurde:

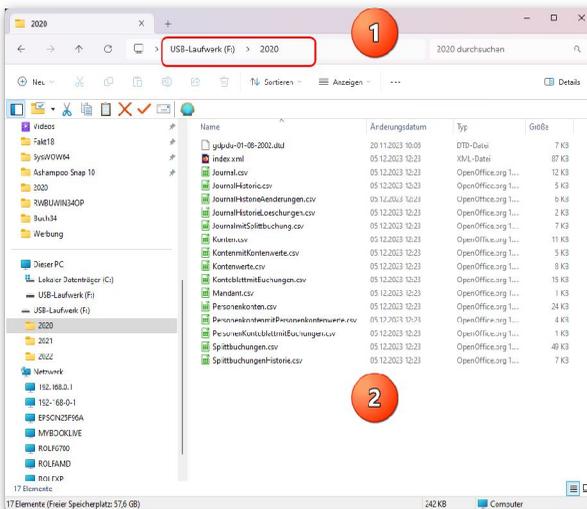


RW-Buch GDPdU-Hinweise *Optional – Daten auf externes Laufwerk kopieren*

Der Inhalt des Laufwerks könnte dann z.B. im Explorer so aussehen, falls auf einen USB-Stick im Laufwerk F: kopiert wurde:



Auf der gewählten Kopierebene **1** befinden sich die einzelnen Jahresdateien als einzelne Dateien im Access97-Format **3**, die GDPdU-Hinweise als PDF-Datei und die Jahresordner **2** mit den Dateien nach dem Beschreibungsstandard. Ein solcher Jahresordner sieht z.B. so aus:



Beim Kopieren auf ein externes Laufwerk wird kein Mandantordner erzeugt, da wir davon ausgehen, dass das Medium nur die Daten eines einzelnen Mandanten enthalten sollte.

Falls Sie dennoch eine Trennung vornehmen möchten, müssten Sie bei der Auswahl für das externe Kopieren als Kopierebene einen Unterordner benutzen oder neu anlegen, also jeweils pro Mandant einen separaten Ordner.

3. Hinweise zum Datenaufbau

3.1. Buchungen

3.1.1. Datengrundlagen

Die Tabelle JOURNAL enthält die Hauptbuchungen. Falls eine Buchung als Splittbuchung erfolgt ist, besteht eine Verknüpfung mit der Tabelle SPLITTBUCHUNGEN. Dabei handelt es sich um eine 1:n-Verbindung, die über das gemeinsame Feld LFDNR erzeugt wird. In einfacher Form sieht das als SQL-Befehl so aus:

```
SELECT Journal.*, Splittbuchungen.*
FROM Journal, Splittbuchungen, Journal LEFT JOIN Splittbuchungen ON Journal.LfdNr = Splittbuchungen.LfdNr
ORDER BY Journal.Belegdatum, Journal.LfdNr, Splittbuchungen.LfdZeile
```

Zum einfacheren Zugriff haben wir ab RW-Buch 16, Release 6 eine weitere Tabelle in die Archiv-Datenbank eingefügt, JOURNALMITSPITTBUCHUNG, die die o.g. Verknüpfung umsetzt und außerdem nur die wichtigsten Buchungsangaben enthält. Dabei wurde folgender SQL-Befehl eingesetzt:

```
SELECT
  Belegdatum,
  Belegnummer,
  Identnummer,
  Journal.LfdNr,
  Splittbuchungen.LfdNr,
  IIF(NOT Splittbuchung, Journal.Buchungstext, Splittbuchungen.Buchungstext) AS BuchungText,
  IIF(NOT Splittbuchung,
    IIF(EA, Gegenkonto, Ausgangskonto),
    IIF(NOT ISNULL(Soll) AND Soll <> 0, KontoNr, ""))
  AS SollKto,
  IIF(NOT Splittbuchung, IIF(EA, Netto, Brutto), Soll) AS Sollbetrag,
  IIF(NOT Splittbuchung,
    IIF(NOT EA, Gegenkonto, Ausgangskonto),
    IIF(NOT ISNULL(Haben) AND Haben <> 0, KontoNr, ""))
  AS HabenKto,
  IIF(NOT Splittbuchung, IIF(EA, Brutto, Netto), Haben) as Habenbetrag,
  IIF(NOT Splittbuchung, UStKonto, "") AS UmsatzsteuerKto,
  IIF(NOT Splittbuchung, IIF(EA, UStBetrag, 0)) AS UmsatzsteuerBetragSoll,
  IIF(NOT Splittbuchung, IIF(NOT EA, UStBetrag, 0)) AS UmsatzsteuerBetragHaben,
  IIF(NOT Splittbuchung, UStProzent, 0) AS UmsatzsteuerProzent,
  IIF(NOT Splittbuchung, Journal.Kostenstelle, Splittbuchungen.Kostenstelle) AS
  KostenstellenNr,
  UStIdNr, Währung, Währungsbetrag,
  IIF(Splittbuchung,
    Splittbuchungen.OPVorgangsNr,
    IIF(NOT ISNULL(Journal.OPKontoNr) AND TRIM(Journal.OPKontoNr) <> "", Journal.OPVor-
    gangsNr, ""))
  AS OPVorgNr,
```

IIF(NOT Splittbuchung, Journal.OPKontoNr, Splittbuchungen.OPKontoNr) AS OPKonto,
IIF(NOT Splittbuchung,
Journal.OPKontoArt + Journal.OPVorgangsArt + Journal.OPZahlungsart,
Splittbuchungen.OPKontoArt + Splittbuchungen.OPVorgangsArt + Splittbuchungen.OPZahlungs-
Art)
AS OPART
FROM Journal, Splittbuchungen,
Journal LEFT JOIN Splittbuchungen ON Journal.LfdNr = Splittbuchungen.LfdNr
ORDER BY Journal.Belegdatum, Journal.LfdNr, Splittbuchungen.LfdZeile

3.1.2. Datenstrukturen

Für die Buchungen stehen diese Strukturen in der Access-Datenbank bereit:

<i>Tabelle</i>	<i>Bemerkung</i>
Journal	Originäre Tabelle mit den zugrunde liegenden Daten, die bei der Eingabe gespeichert wurden. Pro Buchung wird ein Datensatz gespeichert.
Splittbuchungen	Für Splittbuchungen wird zu dem Datensatz im Journal mindestens 2 Datensätze in der Tabelle Splittbuchungen erzeugt. Dabei handelt es sich um eine 1:n-Verbindung, die über das gemeinsame Feld LFDNR erzeugt wird.
JournalmitSplittbuchung	Aufbereitete Daten aus den beiden vorgenannten Tabellen anhand des o.g. SQL-Befehls. Durch den SQL-Befehl werden Soll- und Habenbetrag, Soll- und Habenkonto etc. umgesetzt, so dass diese Tabelle für eine Prüfung der Buchungen am besten geeignet ist.
AbfrageJournalmitSplittbuchung	Selektionsabfrage, die ebenfalls die vorgenannten Umsetzungen vornimmt

3.2. Konten und Kontoblätter

Die Konten enthalten die allgemeinen Angaben zu den Sachkonten, also Kontonummer, Bezeichnung und diverse weitere Informationen. Die Kontoblätter enthalten die Bewegungsdaten, die für die einzelnen Konten in den Buchungen erfasst wurden.

3.2.1. Datengrundlagen Konten

Die Tabelle KONTEN enthält die grundlegenden Kontenangaben. Die Kontenwerte sind dagegen jahresweise in der Tabelle KONTENWERTE enthalten. Die Verbindung wird über das gemeinsame Feld KONTONR erzeugt. In einfacher Form sieht das als SQL-Befehl so aus:

```
SELECT * FROM Konten, Kontenwerte,  
Konten LEFT JOIN Kontenwerte ON Konten.KontoNr = Kontenwerte.KontoNr  
ORDER BY Konten.KontoNr
```

Zum einfacheren Zugriff haben wir ab RW-Buch 16, Release 6 eine weitere Tabelle in die Archiv-Datenbank eingefügt, KONTENMITKONTENWERTE, die die o.g. Verknüpfung umsetzt und außerdem eine Addition der einzelnen Monatswerte vornimmt, um die Salden zu ermitteln. Dabei wurde folgender SQL-Befehl eingesetzt:

```
SELECT  Konten.KontoNr, Bezeichnung,  
(IIF(ISNULL(ABSoll), 0, ABSoll) - IIF(ISNULL(ABHaben), 0, ABHaben)) AS Vorsaldo,  
(IIF(ISNULL(Monat01Soll), 0, Monat01Soll))  
  + (IIF(ISNULL(Monat02Soll), 0, Monat02Soll))  
  + (IIF(ISNULL(Monat03Soll), 0, Monat03Soll))  
  + (IIF(ISNULL(Monat04Soll), 0, Monat04Soll))  
  + (IIF(ISNULL(Monat05Soll), 0, Monat05Soll))  
  + (IIF(ISNULL(Monat06Soll), 0, Monat06Soll))  
  + (IIF(ISNULL(Monat07Soll), 0, Monat07Soll))  
  + (IIF(ISNULL(Monat08Soll), 0, Monat08Soll))  
  + (IIF(ISNULL(Monat09Soll), 0, Monat09Soll))  
  + (IIF(ISNULL(Monat10Soll), 0, Monat10Soll))  
  + (IIF(ISNULL(Monat11Soll), 0, Monat11Soll))  
  + (IIF(ISNULL(Monat12Soll), 0, Monat12Soll)) AS Soll,  
(IIF(ISNULL(Monat01Haben), 0, Monat01Haben))  
  + (IIF(ISNULL(Monat02Haben), 0, Monat02Haben))  
  + (IIF(ISNULL(Monat03Haben), 0, Monat03Haben))  
  + (IIF(ISNULL(Monat04Haben), 0, Monat04Haben))  
  + (IIF(ISNULL(Monat05Haben), 0, Monat05Haben))  
  + (IIF(ISNULL(Monat06Haben), 0, Monat06Haben))  
  + (IIF(ISNULL(Monat07Haben), 0, Monat07Haben))  
  + (IIF(ISNULL(Monat08Haben), 0, Monat08Haben))  
  + (IIF(ISNULL(Monat09Haben), 0, Monat09Haben))  
  + (IIF(ISNULL(Monat10Haben), 0, Monat10Haben))  
  + (IIF(ISNULL(Monat11Haben), 0, Monat11Haben))  
  + (IIF(ISNULL(Monat12Haben), 0, Monat12Haben)) AS Haben,  
(Vorsaldo + (Soll - Haben)) AS Endsaldo  
FROM Konten  
LEFT JOIN Kontenwerte ON Konten.KontoNr = Kontenwerte.KontoNr  
ORDER BY Konten.KontoNr
```

3.2.2. Datenstrukturen Konten

Für die Konten stehen diese Strukturen in der Access-Datenbank bereit:

<i>Tabelle</i>	<i>Bemerkung</i>
Konten	Originäre Tabelle mit den zugrundeliegenden Daten, die bei der Eingabe gespeichert wurden. Pro Sachkonto wird ein Datensatz gespeichert.
Kontenwerte	Für die verbuchten Kontenwerte wird zu dem Konto pro Jahr ein Datensatz gespeichert. Dabei handelt es sich um eine 1:n-Verbindung, die über das gemeinsame Feld KONTONR erzeugt wird.
KontenmitKontenwerte	Aufbereitete Daten aus den beiden vorgenannten Tabellen anhand des o.g. SQL-Befehls. Durch den SQL-Befehl werden Soll- und Habenbetrag, Vor- und Endsaldo etc. umgesetzt, so dass diese Tabelle für eine Prüfung der Konten am besten geeignet ist.
AbfrageKontenmitKontenwerte	Selektionsabfrage, die ebenfalls die vorgenannten Umsetzungen vornimmt

3.2.3. Datengrundlagen Kontoblätter

Ab RW-Buch 19 werden auch Daten für Kontoblätter in der Datenbank aufbereitet, ab Version 20 sind einige Datenfelder erweitert worden, ebenso wie in Version 22. Dabei wird eine ähnliche Datenaufbereitung wie im Journal verwendet, die zugrunde liegenden Tabellen sind wie bei den Buchungen Journal und Splittbuchungen. Die Aufbereitung verläuft 2-stufig, zuerst wird eine so genannte Parameter-Abfrage erzeugt. Diese wird dann für jedes Konto aufgerufen, für das Buchungen vorgenommen wurden. Die Parameter-Abfrage sieht so aus:

```
PARAMETERS Suchkonto Text;
SELECT IIF(NOT Splittbuchung,IIF(Ausgangskonto = SuchKonto,
    IIF(EA, 0, Brutto), IIF(Gegenkonto = SuchKonto,
    IIF(EA, Netto, 0), IIF(UStkonto = SuchKonto,
    IIF(EA, UStBetrag, 0)))) , IIF(Soll <> 0, Soll, 0)) AS Sollbetrag,
IIF(NOT Splittbuchung,IIF(Ausgangskonto = SuchKonto,
    IIF(EA, Brutto, 0), IIF(Gegenkonto = SuchKonto,
    IIF(EA, 0, Netto), IIF(UStkonto = SuchKonto,
    IIF(EA, 0, UStBetrag)))) , IIF(Haben <> 0, Haben, 0)) AS Habenbetrag,
IIF(NOT Splittbuchung, Journal.Buchungstext, Splittbuchungen.Buchungstext) AS
    BuchungText,
IIF(NOT Splittbuchung, IIF(Ausgangskonto = SuchKonto,
    Gegenkonto + IIF(ISNULL(UStkonto) OR TRIM(UStkonto) = ",", ',' + UStkonto),
    IIF(Gegenkonto = SuchKonto,
    Ausgangskonto + IIF(ISNULL(UStkonto) OR TRIM(UStkonto) = ",", ',' + UStkonto),
    IIF(UStkonto = SuchKonto, Ausgangskonto + ',' + Gegenkonto, '')), 'SSSS')
    AS Gegenkonten,
    Journal.*, Splittbuchungen.*
FROM Journal, Splittbuchungen,
Journal LEFT JOIN Splittbuchungen ON Journal.LfdNr = Splittbuchungen.LfdNr
WHERE ((Journal.Splittbuchung AND SuchKonto = Splittbuchungen.Kontonr) OR
(NOT Journal.Splittbuchung AND SuchKonto IN(Ausgangskonto, Gegenkonto, UStkonto)))
ORDER BY Journal.Belegdatum ASC, Journal.LfdNr ASC, Splittbuchungen.LfdZeile ASC
```

Die auf diese Weise gefundenen Daten in einer weiteren Tabelle dann gespeichert. Der SQL-Befehl zur Erzeugung der Tabelle sieht so aus:

```
CREATE TABLE KontoblättermitBuchungen
(AusgabeKontoNr Text, AusgabeKontoBezeichnung Text, Datum DateTime, Beleg Text,
Identnummer Text, UStldNr Text, Kostenstelle Double, Gegenkonten Text,
Buchungstext Text, Vorsaldo Currency, Soll Currency, Haben Currency, Neusaldo Currency)
```

3.2.4. Datenstrukturen Kontoblätter

Für die Kontenblätter stehen diese Strukturen in der Access-Datenbank bereit:

<i>Tabelle</i>	<i>Bemerkung</i>
KontoblättermitBuchungen	Aufbereitete Daten, die für jedes Konto aufgrund der nachfolgende Parameter-Abfrage aufgerufen werden. Pro Buchung wird ein Datensatz gespeichert. Zu jeder Buchung wird der Vor- und auch der Endsaldo ermittelt.
AbfrageEinzelKontoblattmitBuchungen	Hier ist die vorgenannte und beschriebene Parameter-Abfrage gespeichert. Zur Laufzeit wird die zu suchende KontoNr als Eingabe erwartet.

3.3. Personenkonten und Personenkontoblätter

3.3.1. Datengrundlagen

Ähnlich wie die Sachkonten stehen in der größeren Version RW-Buch OP auch separate Stammdaten für Debitoren und Kreditoren zur Verfügung, die in der Grundversion nicht vorhanden sind. In der Tabelle PersonenkontenmitPersonenkontenwerte befinden sich die Salden aus den Personenkonten, die aus den Tabellen Personenkonten und Personenkontenwerte gebildet werden. Mit folgendem SQL-Befehl wird die Tabelle erzeugt:

```
CREATE TABLE PersonenkontenmitPersonenkontenwerte
(OPKontoNr Text, Suchname Text, Vorsaldo Currency,
Soll Currency, Haben Currency, Endsaldo Currency)
```

Zur Laufzeit werden die Werte dann entsprechend eingesetzt. Der Vorsaldo besteht dabei aus einem eventuell vorhandenen Anfangsbestand (außerhalb des Programms erfasste Werte) und den summierten Jahreswerten bis zum Ausgabjahr.

Ähnlich wie ein Sachkontoblatt wird dazu auch ein Personenkontoblatt erzeugt. Die Parameter-Abfrage ist dazu etwas abgewandelt und sieht so aus:

```
PARAMETERS Suchkonto Text;
SELECT
  IIF(Splittbuchung, IIF(Soll <> 0, Soll, 0),
    IIF(NOT ISNULL(Journal.OPKontoNr) AND TRIM(Journal.OPKontoNr) <> ",
      IIF(Journal.OPKontoArt + Journal.OPVorgangsArt + Journal.OPZahlungsArt IN
        ('KRN', 'LRZ', 'KGZ', 'LGN', 'LAN', 'KAZ'), Brutto, 0))) AS Sollbetrag,
    IIF(Splittbuchung, IIF(Haben <> 0, Haben, 0),
      IIF(NOT ISNULL(Journal.OPKontoNr) AND TRIM(Journal.OPKontoNr) <> ",
        IIF(Journal.OPKontoArt + Journal.OPVorgangsArt + Journal.OPZahlungsArt IN
          ('KRZ', 'LRN', 'KGN', 'LGZ', 'KAN', 'LAZ'), Brutto, 0))) AS Habenbetrag,
    IIF(NOT Splittbuchung, Journal.Buchungstext, Splittbuchungen.Buchungstext) AS
      Buchung Text,
    IIF(Splittbuchung, KontoNr,
      IIF(NOT ISNULL(Journal.OPKontoNr) AND TRIM(Journal.OPKontoNr) <> ",
        Ausgangskonto + IIF(ISNULL(Gegenkonto) OR TRIM(Gegenkonto) = ", ", '!' +
          Gegenkonto) + IIF(ISNULL(UStkonto) OR TRIM(UStkonto) = ", ", '!' + UStkonto), ")")
      AS Sachkonten,
    IIF(Splittbuchung, Splittbuchungen.OPVorgangsNr,
      IIF(NOT ISNULL(Journal.OPKontoNr) AND TRIM(Journal.OPKontoNr) <> ",
        Journal.OPVorgangsNr, ")") AS OPVorgNr,
    IIF(NOT Splittbuchung, Journal.OPKontoNr, Splittbuchungen.OPKontoNr) AS OPKonto,
    IIF(NOT Splittbuchung, Journal.OPKontoArt + Journal.OPVorgangsArt +
      Journal.OPZahlungsart, Splittbuchungen.OPKontoArt + Splittbuchungen.OPVorgangsArt +
      Splittbuchungen.OPZahlungsart ) AS OPArt,
    Journal.*, Splittbuchungen.*
FROM Journal, Splittbuchungen,
  Journal LEFT JOIN Splittbuchungen ON Journal.LfdNr = Splittbuchungen.LfdNr
WHERE ((Splittbuchung AND Splittbuchungen.OPKontomr = Suchkonto) OR
  (NOT Splittbuchung AND Journal.OPKontoNr = Suchkonto))
ORDER BY Journal.Belegdatum ASC, Journal.LfdNr ASC, Splittbuchungen.LfdZeile ASC
```

Diese Parameter-Abfrage wird wie ein Sachkontoblatt zur Laufzeit für jedes Personenkonto aufgerufen und zur Speicherung in dieser Tabelle verwandt:

```
CREATE TABLE PersonenKontoblättermitBuchungen
  (AusgabeKontoNr Text, AusgabeKontoBezeichnung Text, Datum DateTime, Beleg Text,
  Identnummer Text, UStIdNr Text, Kostenstelle Double, Sachkonten Text,
  Buchungstext Text, Vorsaldo Currency, Soll Currency, Haben Currency, Neusaldo Currency)
```

3.3.2. Datenstrukturen Personenkonten

Für die Personenkonten stehen diese Strukturen in der Access-Datenbank bereit:

<i>Tabelle</i>	<i>Bemerkung</i>
Personenkonten	Originäre Tabelle mit den zugrundeliegenden Daten, die bei der Eingabe gespeichert wurden. Pro Personenkonto wird ein Datensatz gespeichert.
PersonenkontenWerte	Für die verbuchten Kontenwerte wird zu dem Konto pro Jahr ein Datensatz gespeichert. Dabei handelt es sich um eine 1:n-Verbindung, die über das gemeinsame Feld OPKONTONR erzeugt wird.
PersonenkontenmitPersonenkontenwerte	Aufbereitete Daten aus den beiden vorgenannten Tabellen wodurch Soll- und Habenbetrag, Vor- und Endsaldo etc. umgesetzt werden, so dass diese Tabelle für eine Prüfung der Personenkonten am besten geeignet ist.

3.3.3. Datenstrukturen Personenkontoblätter

Für die Kontenblätter der Personenkonten stehen diese Strukturen in der Access-Datenbank bereit:

<i>Tabelle</i>	<i>Bemerkung</i>
PersonenKontoblättermitBuchungen	Aufbereitete Daten, die für jedes Personenkonto aufgrund der nachfolgende Parameter-Abfrage aufgerufen werden. Pro Buchung wird ein Datensatz gespeichert. Zu jeder Buchung wird der Vor- und auch der Endsaldo ermittelt.
AbfrageEinzelPersonenKontoblattmitBuchungen	Hier ist die vorgenannte und beschriebene Parameter-Abfrage gespeichert. Zur Laufzeit wird die zu suchende PersonenkontoNr als Eingabe erwartet.

3.4. Buchungshistorie

Ab Version 27 steht eine Buchungshistorie zur Verfügung, die alle Änderungen und Löschungen aufnimmt, die im Programm erfolgt sind. Dazu steht die Tabelle JournalHistorie bereit, in der die Daten enthalten sind. Die Strukturen entsprechen dem Journal, zusätzlich wird das Änderungsdatum und eventuell ein Löschungsdatum gespeichert. Bei einer Splittbuchung wird zusätzlich die Tabelle SplittbuchungenHistorie verwendet, die Verbindung erfolgt wie im normalen Journal über das Datenfeld LfdNr.

Zur Vereinfachung erzeugen wir jeweils eine separate Tabelle für Änderungen und Löschungen und jeweils eine Abfrage, mit der ebenfalls die gleichen Daten angezeigt werden können.

<i>Tabelle</i>	<i>Bemerkung</i>
JournalHistorie	Originäre Tabelle mit den zugrunde liegenden Daten, die für die Buchungshistorie gespeichert werden.
SplittbuchungenHistorie	Verbindungstabelle mit Splittingzeilen, falls die Hauptbuchung eine Splittbuchung ist.
JournalHistorieÄnderungen	Selektierte Daten aus der Tabelle JournalHistorie, die nur die Änderungen enthält.
JournalHistorieLöschungen	Selektierte Daten aus der Tabelle JournalHistorie, die nur die Löschungen enthält.

Hinweis zur Verbindung der Tabelle JournalHistorie mit den Journaldaten in der Tabelle Journal: Bei Änderungen besteht die Verbindung über das gleichnamige Datenfeld LfdNr in beiden Tabellen. Bei Löschungen wird die LfdNr in das separate Feld LfdNRLöschung eingetragen und anschließend Lfd-Nr gelöscht, da die Verbindung nach dem Löschen nicht mehr existiert, sondern nur in der Historie nachvollziehbar ist.

Ähnlich wie die Strukturen für die normalen Buchungen stellen wir auch hier den entsprechenden SQL-Befehl zur Verfügung, zuerst für die Tabelle JournalHistorieÄnderungen:

```

SELECT JournalHistorie.LfdNr, LfdNrHistorie,
       IIF(NOT Splittbuchung, IIF(EA, IIF(ISNULL(Netto), 0, Netto), IIF(ISNULL(Brutto),
       0, Brutto)), IIF(ISNULL(Soll), 0, Soll)) AS Sollbetrag,
       IIF(NOT Splittbuchung, IIF(EA, IIF(ISNULL(Brutto), 0, Brutto), IIF(ISNULL
(Netto),
       0, Netto)), IIF(ISNULL(Haben), 0, Haben)) AS Habenbetrag,
       IIF(NOT Splittbuchung, IIF(EA, Gegenkonto, Ausgangskonto), IIF(NOT ISNULL
(Soll) AND Soll <> 0, KontoNr, "")) AS SollKto,
       IIF(NOT Splittbuchung, IIF(NOT EA, Gegenkonto, Ausgangskonto), IIF(NOT IS
NULL(Haben) AND Haben <> 0, KontoNr, "")) AS HabenKto,
       IIF(NOT Splittbuchung, JournalHistorie.Buchungstext,
       SplittbuchungenHistorie.Buchungstext) AS BuchungText,
       IIF(NOT Splittbuchung, UStKonto, "") AS UmsatzsteuerKto,
       IIF(NOT Splittbuchung, IIF(ISNULL(UStBetrag),0,UStBetrag), 0)
       AS UmsatzsteuerBetrag,
       JournalHistorie.*, SplittbuchungenHistorie.*
FROM JournalHistorie
LEFT JOIN SplittbuchungenHistorie ON
       JournalHistorie.LfdNrHistorie = SplittbuchungenHistorie.LfdNr
    
```

```
WHERE JournalHistorie.LfdNr > 0  
ORDER BY JournalHistorie.LfdNr, LfdNrHistorie DESC;
```

Dann für die Tabelle JournalHistorieLöschungen:

```
SELECT LfdNrLöschung, LfdNrHistorie,  
       IIF(NOT Splittbuchung, IIF(EA, IIF(ISNULL(Netto), 0, Netto), IIF(ISNULL(Brutto),  
       0, Brutto)), IIF(ISNull(Soll), 0, Soll)) AS Sollbetrag,  
       IIF(NOT Splittbuchung, IIF(EA, IIF(ISNULL(Brutto), 0, Brutto), IIF(ISNULL  
(Netto),  
       IIF(ISNULL(Haben), 0, Haben)) AS Habenbetrag,  
       IIF(NOT Splittbuchung, IIF(EA, Gegenkonto, Ausgangskonto), IIF(NOT ISNULL  
(Soll) AND Soll <> 0, KontoNr, "")) AS SollKto,  
       IIF(NOT Splittbuchung, IIF(NOT EA, Gegenkonto, Ausgangskonto), IIF(NOT IS  
       NULL(Haben) AND Haben <> 0, KontoNr, "")) AS HabenKto,  
       IIF(NOT Splittbuchung, JournalHistorie.Buchungstext,  
       SplittbuchungenHistorie.Buchungstext) AS BuchungText,  
       IIF(NOT Splittbuchung, UStKonto, "") AS UmsatzsteuerKto,  
       IIF(NOT Splittbuchung, IIF(ISNULL(UStBetrag),0,UStBetrag), 0)  
       AS UmsatzsteuerBetrag,  
       JournalHistorie.*, SplittbuchungenHistorie.*  
FROM JournalHistorie  
LEFT JOIN SplittbuchungenHistorie ON  
       JournalHistorie.LfdNrHistorie = SplittbuchungenHistorie.LfdNr  
WHERE LfdNrLöschung > 0  
ORDER BY LfdNrLöschung, LfdNrHistorie DESC;
```

3.5. Zusammenfassung Datenstrukturen

Als Zusammenfassung der vorgenannten Ausführungen zu den Datenstrukturen empfehlen wir für eine Betriebsprüfung die Verwendung der folgenden Tabellen, falls das Format Access97 verwendet werden kann:

<i>Tabelle</i>	<i>Bemerkung</i>
JournalmitSplittbuchung	Buchungsdaten
KontenmitKontenwerte	Sachkonten mit Vor- und Endsalden
KontoblättermitBuchungen	Sachkontoblatt
PersonenkontenmitPersonenkontenwerte	Personenkonten mit Vor- und Endsalden (Verfügbar nur in der OP-Version)
PersonenKontoblättermitBuchungen	Kontoblätter für Personenkonten (Verfügbar nur in der OP-Version)

3.6. Dateien nach dem Beschreibungsstandard

Für die Dateien nach dem Beschreibungsstandard werden folgende Dateien erzeugt, die jeweils auf Basis der weiter vorne beschriebenen Datengrundlagen und Strukturen mit dem Inhalt aus den entsprechenden Access-Tabellen der RW-Buch-Datenbank gefüllt werden:

<i>Datei</i>	<i>Bemerkung</i>
Gdpdu-01-08-2002.dtd	Strukturinformationen und Elementdefinitionen für den Aufbau der Datei index.xml
index.xml	Beschreibung der erzeugten Dateien und die jeweils dafür vorhandenen Datenfelder. Außerdem sind hier die allgemeinen Hinweistexte enthalten, die weiter vorne beschrieben sind.
Journal.csv	Basisdaten mit Normalbuchungen
Splittbuchungen.csv	Basisdaten mit Splittbuchungen
JournalmitSplittbuchung.csv	Buchungsdaten mit Normal- und Splittbuchungen
Konten.csv	Basisdaten Konten
Kontenwerte.csv	Basisdaten Kontenjahreswerte
KontenmitKontenwerte.csv	Sachkonten mit Vor- und Endsalden
KontoblattmitBuchungen.csv	Sachkontoblatt
Mandant.csv	Basisdaten mit Firmenangaben, Adresse etc.
Personenkonten.csv	Basisdaten Personenkonten (Verfügbar nur in der OP-Version)
PersonenkontenmitPersonenkontenwerte.csv	Personenkonten mit Vor- und Endsalden (Verfügbar nur in der OP-Version)
PersonenKontoblattmitBuchungen.csv	Kontoblätter für Personenkonten (Verfügbar nur in der OP-Version)
JournalHistorie.csv	Basisdaten Journalhistorie
SplittbuchungenHistorie.csv	Basisdaten Splittbuchungshistorie
JournalHistorieAenderungen.csv	Journalhistorie nur Änderungen
JournalHistorieLoeschungen.csv	Journalhistorie nur Löschungen

Die CSV-Dateien enthalten in der ersten Zeile die Feldnamen (Spaltennamen), der Datenbereich beginnt daher ab der Zeile 2. In der index.xml ist das durch eine Range-Angabe umgesetzt.

Falls Probleme bei der Verarbeitung auftauchen sollten, setzen Sie sich bitte mit einer kurzen Fehlerbeschreibung mit uns in Verbindung, damit wir das prüfen und gegebenenfalls korrigieren können.