uverein Entwickler-Information

Generated with ROBODoc Version 4.99.41 (Feb 22 2011)

25. Oktober 2015

Inhaltsverzeichnis

1 Allgemein

INFO:

```
Allgemeine Daten
```

1.1 base.inc

NAME:

```
base.inc
Dieses Modul muss in jede PHP-Datei direkt oder indirekt eingebunden werden
Es werden die globalen Einstellungen fuer das Projekt vorgenommen
```

AUTOR:

```
Uwe Schied
```

ABHAENGIG VON:

```
base-initial.inc       (Projekteinstellungen)
database.inc           (Datenbank-Klasse)
funktionen.php         (Funktionen-Sammlung)
klasse/formular.inc    (Formularverwaltung)
```

DEFINITION:

```
Variable / Konstante    Informationen
------------------------------------------------------------------------
AKT_YEAR                aus Superglobals ausgelesen

AUSWAHL                 aus Superglobals ausgelesen

EDIT_LANG               aus Superglobals ausgelesen

SPRACHE                 aktuelle Sprache

UVEREIN_DB              Datenbankzugriff ueber die Klasse cl_database

VEREIN_CHANGED          aus Superglobals ausgelesen

VEREIN_NAME             Name des aktuellen Vereins

VIRTUALKONTO            Array mit symbolischen Werten, fuer andere Module.
                        KEY              Inhalt
                        ..............................................
                        Konten und Kontenklassen:
                          BEITRAG        Beitragskonto
```

```
                          UST             Konto fuer die Vorsteuer
                          FINANZ          Finanzkonten aus bu_klasse
                          UEBERTRAG       Konto fuer den Uebertrag
                          SCHWEBE         Schwebekonto
                        In der Datenbank erfasste Texte:
                          BEITRAG1        1. Beitragserinnerung
                          BEITRAG2        2. Beitragserinnerung
                          BEITRAG3        3. Beitragserinnerung
                          BEITRAGLEV      Lastschrift nicht eingeloest

  FORM                  Formular
  SPECIAL_CALL          Klasse cl_special
```

1.1.1 update

DEFINITION:

```
  PROJEKT_UPDATE_1705274    bu_kst um Jahr ergaenzen     in Arbeit
  PROJEKT_UPDATE_1424032    Tipp einblenden              in Arbeit
```

FUNCTION:

```
  use_update ($n)
    gibt true zurueck, wenn das Update benutzt werden kann
    oder der Testmodus aktiv ist.
```

1.2 language.php

NAME:

```
   language.php
   Diese Modul ist fuer die Uebersetzungen in andere Sprachen
   erforderlich
```

2 Beitrag

INFO:

```
Beitragskonto verwalten
```

2.1 beitrag-soll.php

INFO:

```
Automatische Eintragung des Beitrags-Solls
```

HISTORIE:

```
13.02.2010 von Test in Produktion uebertragen
18.01.2015 Faelligkeit an SEPA angepasst = heute + 9 Tage
```

BUGS:

```
keine bekannt
```

ABHAENGIG VON:

```
include_once "base.inc";
include_once "header.inc";
include_once "klassen/uverein-database.inc";
```

FUNCTION:

```
auswerten
  Die Funktion ermittelt ob der Beitrag faellig ist und
  bucht ggf. das Beitrags-Soll
```

BESCHREIBUNG:

```
-> aktuelles Datum holen
-> Zahlungsweise ermitteln
-> Beitrag entsprechend der Zahlungsweise berechnen
-> Buchungstext ermitteln
-> Wenn der Beitrag faellig ist, buchen
```

FUNCTION:

```
Hauptteil
```

BESCHREIBUNG:

```
-> Wann wurde das letzte Beitrags-Soll gebucht?
-> Alle Eintraege aus der Tabelle 'beitrags_konto' auslesen
-> Funktion 'auswerten' nur aufrufen, wenn der Betrag <> 0
-> Buchung des Beitrags-Solls merken
-> Info ausgeben
```

2.2 mahntext.php

NAME:

```
mahntext.php
```

INFO:

```
Erfassung des Mahntextes fuer das Beitrags-Inkasso
```

BUGS:

```
keine bekannt
```

3 DATAUS-Klasse

INFO:

```
Klasse fuer die Erstellung einer DATAUS-Datei fuer die Daten-
uebertragung an ein deutsches Kreditinstitut
```

3.1 dataus-datei.inc

KLASSE:

```
cl_dataus
Klasse fuer den Datenaustausch mit einem deutschen Kreditinstitut
Es wird eine Textdatei erstellt, die zur Bank hochgeladen werden
kann.
```

HISTORIE:

```
20.12.2009 Deutsche Umlaute integriert
05.07.2009 Klasse erstellt
```

BUGS:

```
keine
```

3.1.1 ausgeben ()

SICHTBAR:

```
Die Funktion ist public
```

BESCHREIBUNG:

```
ausgeben ()

Erstellt die DATAUS-Datei und bietet sie zum Download an.
```

3.1.2 buchung (BLZ, Konto-Nr, Name, Zweck, Betrag)

SICHTBAR:

```
Die Funktion ist public
```

BESCHREIBUNG:

```
buchung ($bls, $kto, $name, $zweck, $betrag)
  $blz     Bankleitzahl
  $kto     Konto-Nr
  $name    Name
  $zweck   Verwendungszweck
  $betrag  Betrag, der eingezogen werden soll

Erstellt die einzelnen Positionen des DATAUS-Satzes
```

3.1.3 checkDIN66003 (Art, Wert)

SICHTBAR:

```
Die Funktion ist nur innerhalb der Klasse aufrufbar
```

BESCHREIBUNG:

```
checkDIN66003 ($Art, $Wert)
  $Art        Art des Wertes
              1  Text
              2  Zahl
  $Wert       Zahl oder Text, der angepasst werden soll
Die Anpassung erfolgt nach den Richtlinien der DIN 66003
```

3.1.4 cl_dataus (Name, Bankleitzahl, Konto-Nr)

BESCHREIBUNG:

```
Konstruktor fuer die DATAUS-Klasse
Uebergabe-Paramter:
   Name          Fuer wen wird die Datei erstellt
   Bankleitzahl  Bankleitzahl der eigenen Bank
   Konto-Nr      Eigene Konto-Nr.
```

3.1.5 feldwert (Wert, Art, Laenge, Nachkommastellen)

SICHTBAR:

```
Die Funktion ist nur innerhalb der Klasse aufrufbar
```

BESCHREIBUNG:

```
feldwert ($Wert, $Art, $Laenge, [$Nachkomma])
  $Wert       Zahl oder Text, der angepasst werden soll
  $Art        Art des Wertes
              1  Text - wird rechts bis maximal $Laenge mit Leerzeichen
                 aufgefuellt
```

```
                2  Zahl - wird links bis maximal $Laenge mit Nullen auf-
                   gefuellt
  $Laenge       Laenge des Ergebnis-Strings
                Wenn der Wert laenger als $Laenge ist, wird er auf $Laenge
                gekuerzt - ansonsten wird der Wert je nach $Art aufgefuellt
  $Nachkomma    Anzahl der Nachkommastellen bei Zahlen - Voreinstellung ist
                keine Nachkommastellen
Die Anpassung erfolgt nach den Richtlinien der DIN 66003
```

4 Datenbank-Klasse

INFO:

```
Klassen fuer den Zugriff auf die Datenbank
```

4.1 Basisdaten +001+

NAME:

```
klassen/001.Basisdaten
```

KLASSE:

```
cl_uverein_database_001
Klasse fuer den Zugriff auf die Tabelle vereine
```

ABHAENGIG VON:

```
cl_database (database.inc)
PHP 5
```

HISTORIE:

```
06.01.2009 get_verein ergaenzt
03.01.2010 get_vereinsname ergaenzt
21.03.2009 get_jahr ergaenzt
16.03.2009 Klasse erstellt
```

AUTOR:

```
Uwe Schied
```

BUGS:

```
keine bekannt
```

4.1.1 get_jahr ()

FUNCTION:

```
get_jahr ()        aktuelles Jahr holen
```

4.1.2 get_verein ()

FUNCTION:

```
get_verein Kurzname des Vereins holen
```

4.1.3 get_vereinsname()

FUNCTION:

```
get_vereinsname() Name des Vereins ermitteln
```

ERGEBNIS:

```
Gibt den Langnamen des Vereins aus der Tabelle "vereine" zurueck
```

4.1.4 set_jahr (jahr)

FUNCTION:

```
set_jahr ($jahr)   aktuelles Jahr setzen
  $jahr            Jahr
```

4.1.5 set_verein (name)

FUNCTION:

```
set_verein ($name) Kurzname des Vereins setzen
  $name            Vereinsname
```

4.2 database-errorcode.inc

KLASSE:

```
cl_database
Fehlercodes fuer alle Datenbank-Klassen
```

HISTORIE:

```
12.12.2009 Fehlercodes ausgelagert
```

BUGS:

```
keine bekannt
```

4.2.1 Konstanten

KONSTANTE:

```
Name                        Beschreibung
------------------------------------------------------------------------
CL_DATABASE_OK                 es sind keine Fehler aufgetreten
```

```
CL_DATABASE_ERROR_CANTOPEN     Datenbank kann nicht geoeffnet werden
CL_DATABASE_ERROR_SQL          es ist ein SQL-Fehler aufgetreten
CL_DATABASE_ERROR_UNKNOWN   es ist ein unbekannter Fehler aufgetreten
CL_DATABASE_ERROR_004_INSERT   INSERT-Fehler beim Eintrag eines neuen
                               Buchungsbeleges
CL_DATABASE_ERROR_004_UPDATE   UPDATE-Fehler beim aendern eines Buchungs-
                               beleges
CL_DATABASE_ERROR_005_INSERT   INSERT-Fehler bei der Beitragsbuchung
CL_DATABASE_MAX_ERROR          nur fuer klasseninterne Nutzung
```

4.3 database.inc

KLASSE:

```
cl_database
Klasse fuer den Zugriff auf eine Postgres-Datenbank
```

HISTORIE:

```
12.12.2009 Fehlercodes ausgelager in database-errorcode.inc
08.08.2009 Schema beim oeffnen der Datenbank aktivieren
17.02.2009 Schema eingebunden
15.12.2007 Klasse erstellt
```

BUGS:

```
keine bekannt
```

4.3.1 cl_database (Datenbank, Datenbank-Schema)

DEFINITION:

```
cl_database ($name, $schema)
Konstruktor der Klasse. Die Initialisierung erfolgt durch Uebergabe des
Datenbanknamens.
  $name    = Name der Postgres-Datenbank
  $schema  = Datenbank-Schema (Voreinstellung: public)
```

BEISPIEL:

```
$datenbank = new cl_database("beispieldb");
```

4.3.2 datenbank_oeffnen ()

FUNCTION:

```
datenbank_oeffnen ()

Diese Funktion wird nur klassenintern benutzt und oeffnet die Datenbank.
```

ERGEBNIS:

```
Klasseninterne Datenbankvariablen werden gesetzt.
```

4.3.3 error_name (Fehler-Nr., Fehlertext)

FUNCTION:

```
error_name ($key, $inhalt)
  $key     Fehlercode lt. untenstehender Tabelle
  $inhalt  Text, der ausgegeben werden soll

Mit dieser Funktion koennen Fehlermeldungen eingetragen werden. Die
folgenden Fehlercodes sind vorhanden und koennen geaendert werden:

$key                         ! $inhalt
-----------------------------!-------------------------------------
CL_DATABASE_OK               ! OK
CL_DATABASE_ERROR_CANTOPEN   ! Can't open Database
CL_DATABASE_ERROR_SQL        ! Error while executing SQL-query
CL_DATABASE_ERROR_UNKNOWN    ! Unknown Error

Fehler-Array wird innerhalb der Klasse aktualisiert
```

BEISPIEL:

```
$datenbank->error_name (CL_DATABASE_OK,"Alles OK");
```

ERGEBNIS:

```
Die Funktion gibt kein Ergebnis zurueck
```

4.3.4 error_text ()

FUNCTION:

```
error_text

Diese Funktion gibt des Status der letzten Aktion zurueck. Ausgegeben
wird der durch die Funktion error_name voreingestellte Text
```

BEISPIEL:

```
echo $datenbank->error_text();
```

ERGEBNIS:

```
Zeichenkette mit der Fehlermeldung
```

4.3.5 fieldnames (Antwortzeiger)

FUNCTION:

```
fieldnames ([$qu])
  $qu    optionale Angabe des Antwortzeigers

Die Funktion liefert ein Array mit den Spaltenueberschriften = Feld-
namen zur letzten oder optional der durch den Antwortzeiger definierten
SQL-Abfrage zurueck.
```

ERGEBNIS:

```
Der Rueckgabewert ist ein Array mit den Feldnamen. Wenn die letzte An-
frage keine Felder lieferte wird ein leerer String zurueck geliefert.
```

BEISPIEL:

```
$qu = $datenbank->sql_befehl("SELECT * FROM belege");
$FELDER = $datenbank->fieldnames();
foreach ($FELDER as $key => $inhalt)
{
  echo "$key = $inhalt <BR>";
}
```

4.3.6 liste4selection (Tabelle, where, key, Anzeige)

FUNCTION:

```
liste4selection Array mit Daten fuer eine Selektion
  Tabelle        aus welcher Tabelle wird selectiert
  where          SQL-Befehl
  key            Feld mit dem Selektions-Key
  Anzeige        Welche Felder sollen angezeigt werden incl. Format
```

ERGEBNIS:

```
Es wird ein Array mit folgenden Daten zurueck gegeben:
  KEY        = key
  VALUE      = Anzeige
```

4.3.7 result_array (Antwortzeiger)

FUNCTION:

```
result_array ([$qu])
  $qu    optionale Angabe des Antwortzeigers

Array mit dem Resultat einer SQL-Abfrage
```

ERGEBNIS:

```
Die Funktion gibt das Ergebnis der letzten oder der durch den Antwort-
zeiger uebergebenen Anfrage als zweidiminensionales Array zurueck.

Die Adressierung im Array geschieht durch Angabe von Zeilen und Spalten-
Nr. oder alternativ durch Zeilen-Nr. und Feldname.

Wenn die gewuenschte Abfrage kein Ergebnis liefert ist die Rueckgabe
nicht definiert.
```

BEISPIEL:

```
$qu = $datenbank->sql_befehl("SELECT * FROM belege");
$data = $datenbank->result_array();
echo "In Feld 0 der Zeile 0 steht ".$data[0][0];
echo "Im Feld kontonr der Zeile 0 steht ".$data[0]["kontonr"];
```

4.3.8 sql_1_value (Tabelle, abzuholendes Feld, WHERE-Abfrage)

FUNCTION:

```
sql_1_value ($tabelle, $wert, $where)
  $tabelle  Welche Datenbank-Tabelle soll abgefragt werden?
  $wert     Welches Feld soll geholt werden?
  $where    Abfrage

Die Funktion holt einen Wert aus der Datenbank. Die WHERE-Formulierung
sollte daher ganz genau ueberdacht werden.
```

ERGEBNIS:

```
Es wird der Inhalt des gesuchten Feldes zurueck gegeben. Wenn WHERE kein
Ergebnis lieferte, wird ein leerer String zurueck gegeben.
```

BEISPIEL:

```
$nr = $datenbank->sql_1_value("belege","max(nr) + 1","");
// gibt die naechste Beleg-Nr. zurueck
echo "Die naechste Beleg-Nr. ist $nr";
```

4.3.9 sql_abfrage (Tabelle, WHERE-Abfrage, Sortierung, Gruppierung)

FUNCTION:

```
sql_abfrage ($tabelle, $where , [[$order],$group])
  $tabelle  Welche Datenbank-Tabelle soll abgefragt werden?
  $where    Abfrage
  $order    Optionale Sortierung der Rueckgabe
  $group    Optionale Gruppierung. Wenn hier Angaben gemacht werden, muss
            beachtet werden, dass alle Felder abgefragt werden - im
            Normalfall bleibt dieses Feld daher leer.

Die Funktion setzt aus den uebergebenene Daten einen SQL-Befehl zusammen
und gibt diesen an die Datenbank weiter.
```

ERGEBNIS:

```
Es wird ein Zeiger auf die Antwort zurueck geliefert.
```

BEISPIEL:

```
$qu = $datenbank->sql_abfrage("belege","nr>10");
// gibt alle Belege mit einer Beleg-Nr. > 10 zurueck
$i = 0; while ($i < $datenbank->sql_lines())
{
  $data = $datenbank->sql_line($i);
  echo "$data->nr $data->datum $data->text <BR>";
  $i++;
}
```

4.3.10 sql_befehl (Befehl)

FUNCTION:

```
sql_befehl ($befehl)
  $befehl    SQL-Befehl, der ausgefuehrt werden soll

Die Funktion schickt den uebergebenen SQL-Befehl direkt an die Datenbank
```

ERGEBNIS:

```
Es wird ein Zeiger auf die Antwort zurueck geliefert
```

BEISPIEL:

```
$qu = $datenbank->sql_befehl("SELECT * FROM belege");
$i = 0; while ($i < $datenbank->sql_lines($qu))
{
  $data = $datenbank->sql_lines($qu);
  echo "$data->nr $data->datum $data->text <BR>";
  $i++;
}
```

4.3.11 sql_fields (Antwortzeiger)

FUNCTION:

```
sql_fields ([$qu])
  $qu    optionale Angabe des Antwortzeigers

Die Funktion liefert die Anzahl der Spalten zur letzten oder optional
der durch den Antwortzeiger definierten SQL-Abfrage zurueck.
```

ERGEBNIS:

```
Der Rueckgabewert ist eine Zahl >= 0
```

BEISPIEL:

```
$qu = $datenbank->sql_befehl("SELECT * FROM belege");
echo "Die Abfrage hat ";
echo $datenbank->sql_fields();
echo " Spalten zurueck gegeben.";
```

4.3.12 sql_insert (Tabelle, welche Felder, Eintraege)

FUNCTION:

```
sql_insert ($tabelle, $was , $inhalt)
  $tabelle  Datenbank-Tabelle in die eingetragen werden soll
  $was      Welche Felder werden eingetragen?
  $inhalt   Die VALUES, die eingetragen werden

Die Funktion fuegt eine neue Zeile in die Datenbank ein. Die Anzahl der
Felder muss mit den in $inhalt angegebenen Feldern uebereinstimmen. Die
Funktion prueft nicht ob die Zeile tatsaechlich eingefuegt werden kann.
Konnte die Zeile nicht eingefuegt werden, ist der Errorcode auf
CL_DATABASE_ERROR_SQL gesetzt.
```

ERGEBNIS:

```
Die Funktion gibt kein Resultat zurueck.
```

BEISPIEL:

```
$datenbank->sql_insert ("belege","nr,datum,betrag","'21','14.02.2009','100.98');

Ob ein Fehler aufgetreten ist, kann ueber den Errorcode abgefragt werden.

if ($datenbank->ERRORCODE != CL_DATABASE_OK)
  echo $datenbank->error_text();
```

4.3.13 sql_line (Zeilen-Nr, Antwortzeiger)

FUNCTION:

```
sql_line ($nr, [$qu])
  $nr    Zeilen-Nr. die geholt werden soll
  $qu    optionale Angabe des Antwortzeigers

Die Funktion gibt die gewaehlte Antwortzeile zur SQL-Abfrage zurueck. $nr
wird geprueft und nur gültige Zeilen ausgewählt.
```

ERGEBNIS:

```
Es wird ein Array mit der gewuenschten Zeile zurueck gegeben.
  * wenn $nr < 0 wird die erste Zeile ausgegeben
  * wenn $nr > Anzahl der Spalten, wird die letzte Zeile ausgeben
```

BEISPIEL:

```
$qu = $datenbank->sql_befehl("SELECT * FROM belege");
$data = $datenbank->sql_line(0,$qu);
echo "$data->nr $data->datum $data->text <BR>";
```

4.3.14 sql_lines (Antwortzeiger)

FUNCTION:

```
sql_lines ([$qu])
  $qu    optionale Angabe des Antwortzeigers

Die Funktion liefert die Anzahl der Zeilen zur letzten oder optional
der durch den Antwortzeiger definierten SQL-Abfrage zurueck.
```

ERGEBNIS:

```
Der Rueckgabewert ist eine Zahl >= 0
```

BEISPIEL:

```
$qu = $datenbank->sql_befehl("SELECT * FROM belege");
echo "Die Abfrage hat ";
echo $datenbank->sql_lines();
echo " Ergebnisse gebracht.";
```

4.3.15 sql_update (Tabelle, Neue Einraege, WHERE-Abfrage)

FUNCTION:

```
sql_update ($tabelle, $set , [$condition])
  $tabelle   Datenbank-Tabelle, die geaendert werden soll
  $set       neue Eintraege
  $condition Optionale WHERE-Angabe

Mit dieser Funktion koennen ein oder mehrere Felder mit geaenderten Werten
ueberschrieben werden. Die Funktion prueft nicht ob die Aenderung tat-
saechlich durchgefuehrt werden kann. Konnte keine Aenderung erfolgen, ist
der Errorcode auf CL_DATABASE_ERROR_SQL gesetzt
```

ERGEBNIS:

```
Die Funktion gibt kein Resultat zurueck.
```

BEISPIEL:

```
$datenbank->sql_update ("belege","nr='10','nr='99');

Ob ein Fehler aufgetreten ist, kann ueber den Errorcode abgefragt werden.

if ($datenbank->ERRORCODE != CL_DATABASE_OK)
```

4.3.16 TESTMODE

VARIABLE:

```
TESTMODE
Wenn diese Variable auf true gesetzt ist, werden zu jeder Daten-
bankaktion Meldungen mit den Aufrufparametern und den Rueckgabe-
werten ausgegeben.

Voreinstellung = false

Die Variable kann ausserhalb der Klasse geaendert werden und
wirkt sich nur innerhalb der Klasse aus.
```

4.3.17 testmode (Fehlerinformation)

FUNCTION:

```
testmode ($info)

Meldungen, die ausgegeben werden, wenn TESTMODE = true

Die Funktion wird nur innerhalb der Klasse verwandt und soll
von ausserhalb nicht aufgerufen werden.
```

4.4 Konstruktor

NAME:

```
klassen/uverein-database.inc
```

KLASSE:

```
cl_uverein_database
Klasse fuer den Zugriff auf die Datenbank des Projekts uverein
```

ABHAENGIG VON:

```
Klasse                   Datei                      Hinweis
-------------------------------------------------------------------------
cl_database              database.inc               Datenbank-Klasse
cl_uverein_database_001  uverein-database.001.inc Basis-Informationen
cl_uverein_database_002  uverein-database.002.inc bu_konten
cl_uverein_database_003  uverein-database.003.inc virtual_konto
cl_uverein_database_004  uverein-database.004.inc bu_belege
cl_uverein_database_005  uverein-database.005.inc Beitragskonto
PHP 5
```

HISTORIE:

```
16.03.2009 Klasse erstellt
10.04.2009 Basis ausgelagert
10.04.2009 Tabelle bu_konten eingebunden
10.04.2009 Tabelle virtual_konto eingebunden
12.12.2009 Tabelle bu_belege eingebunden
06.01.2010 Beitragsbuchungen eingebunden
```

AUTOR:

```
Uwe Schied
```

BUGS:

```
keine bekannt
```

4.4.1 Klassendefinition (Datenbank, Datenbank-Schema)

FUNCTION:

```
Klassen-Konstruktor
$name          Name der Datenbank
$schema        optionale Angabe des Datenbank-Schemas - Voreinstellung = public
```

4.5 Tabelle: beitrag +005+

NAME:

```
klassen/uverein-database.005.inc
```

KLASSE:

```
cl_uverein_database_005
Klasse fuer den Zugriff auf die Beitragsbuchungen
```

ABHAENGIG VON:

```
Klasse                   Datei                      Hinweis
-------------------------------------------------------------------------
cl_database              database.inc               Datenbank-Klasse
cl_uverein_database_001  uverein-database.001.inc Basis-Informationen
cl_uverein_database_002  uverein-database.002.inc bu_konten
cl_uverein_database_003  uverein-database.003.inc virtual_konto
cl_uverein_database_004  uverein-database.004.inc bu_belege
PHP 5
```

BESCHREIBUNG: HISTORIE:

```
06.01.2010 Klasse erstellt
```

AUTOR:

```
Uwe Schied
```

BUGS:

```
keine bekannt
```

4.5.1 beitrag_insert

FUNCTION:

```
beitrag_insert  Beitragsbuchung erfassen
  $nr           Mitglieds-Nr
  $datum        Buchungsdatum
  $text         Buchungstext
  $haben        Betrag fuer die Haben-Buchung (Voreinstellung 0)
  $soll         Betrag fuer die Soll-Buchung (Voreinstellung 0)
  $konto        Finanzkonto fuer die Buchung
                optional -> wenn leer, wird das Konto aus virtual_konto
                ermittelt
```

BEISPIEL:

```
$UVEREIN_DB->beitrag_insert ("1","31.12.2009","Ueberweisung","20");
   = Habenbuchung ist auf Standard-Finanzkonto eingegangen
$UVEREIN_DB->beitrag_insert ("1","01.01.2010","Jahresbeitrag","0","20");
   = Beitragssoll gebucht
```

ERGEBNIS:

```
Buchung wird ins Beitragskonto eingetragen.
im Fehlerfall wird CL_DATABASE_ERROR_005_INSERT in $ERRORCODE zurueckgeliefert
```

4.5.2 get_beitraege

FUNCTION:

```
get_beitraege
  $SUM    summierte Ausgabe?
    true  die Zahlungen werden pro Finanzkonto summiert
    false es werden alle Zahlungen zurueckgeliefert
```

VOREINSTELLUNG:

```
true
```

ERGEBNIS:

```
Liefert einen Zeiger auf die Beitragszahlungen des aktuellen Jahres
```

4.5.3 get_beitragfaellig

FUNCTION:

```
get_beitragfaellig Beitragsfaelligkeit als Array holen
  $nr              Nr. aus der Tabelle beitrags_konto
```

ERGEBNIS:

```
Es wird ein Array zuruckgeliefert, das folgende KEYs enthaellt:
  KEY          Inhalt
  ------------------------------------------------------------
  TAG          Tag der 1. Faelligkeit
  MONAT        Monat der 1. Faelligkeit
  JAHR         Jahr der 1. Faelligkeit
```

4.5.4 get_beitragshoehe

FUNCTION:

```
get_beitragshoehe Hoehe des Jahresbeitrages holen
  $nr             Nr. der Beitragsart aus Tabelle beitrags_art
```

4.5.5 get_beitragstext

FUNCTION:

```
get_beitragstext Beitragsart als Text holen
  $nr           Nr. der Beitragsart aus Tabelle beitrags_art
```

4.5.6 test_log

FUNCTION:

```
test_log
Diese Funktion ist Klassen-Intern definiert und außerhalb der Klasse nicht
aufrufbar.
```

4.6 Tabelle: bu_belege +004+

NAME:

```
klassen/uverein-database.004.inc
```

KLASSE:

```
cl_uverein_database_004
Klasse fuer den Zugriff auf die Tabelle bu_belege
```

ABHAENGIG VON:

```
Klasse                  Datei                    Hinweis
-------------------------------------------------------------------
cl_database             database.inc             Datenbank-Klasse
cl_uverein_database_001 uverein-database.001.inc Basis-Informationen
cl_uverein_database_002 uverein-database.002.inc bu_konten
cl_uverein_database_003 uverein-database.003.inc virtual_konto
PHP 5
```

BESCHREIBUNG: HISTORIE:

```
12.12.2009 Klasse erstellt
```

```
12.12.2009 public  function beleg_insert erstellt
20.12.2009 private function test_log erstellt
20.12.2009 public  function beleg_update erstellt
23.12.2009 public  function get_beleg erstellt
01.01.2010 public  function sum_finanz erstellt
03.11.2012 neue Spalte "erledigt" berücksichtigen
```

AUTOR:

```
Uwe Schied
```

BUGS:

```
keine bekannt
```

4.6.1 beleg_insert

FUNCTION:

```
beleg_insert  Neuen Beleg eintragen
  $beleg      Array mit den einzutragenden Daten des Belegs
  $jahr       optionale Angabe des Buchungsjahres
```

BEISPIEL:

```
$EINTRAGEN = array ("b_nr" => "100", "b_datum" => "12.12.2009",
                    "b_konto" => "1000", "f_konto" => "2000",
                    "b_betrag" => "10.99", "b_text" => "Beispieleintrag",
                    "b_kst" => "1");
$datenbank->beleg_insert ($EINTRAGEN);
```

ERGEBNIS:

```
Beleg wird eingetragen - im Fehlerfall wird CL_DATABASE_ERROR_004_INSERT
in $ERRORCODE zurueckgeliefert
```

4.6.2 beleg_update

FUNCTION:

```
beleg_update  Beleg aendern
  $ident      Beleg, der geaendert werden soll
  $beleg      Array mit den einzutragenden Daten des Belegs
```

BEISPIEL:

```
$EINTRAGEN = array ("b_nr" => "100", "b_datum" => "12.12.2009",
                    "b_konto" => "1000", "f_konto" => "2000",
                    "b_betrag" => "10.99", "b_text" => "Beispieleintrag",
                    "b_kst" => "1");
$datenbank->beleg_update ("idx='120'",$EINTRAGEN);
```

ERGEBNIS:

```
Beleg wird geaendert - im Fehlerfall wird CL_DATABASE_ERROR_004_UPDATE
in $ERRORCODE zurueckgeliefert
```

4.6.3 find_beleg

FUNCTION:

```
find_beleg  Beleg suchen
  $datum    Datum nach dem gesucht wird
  $betrag   Buchungsbetrag
  $inhalt   Buchungstext
```

BEISPIEL:

```
$Datenbank->find_beleg ("",0,"Testbeleg");
$i = 0; while ($i < $Datenbank->sql_lines ())
{ $data = $Datenbank->sql_line($i); echo $data->b_nr." ".$data->b_text; $i++; }
```

ERGEBNIS:

```
Datenbank wird entsprechend den uebergebenen Parametern durchsucht und
die gefundenen Belege koennen ueber die Datenbankabfragen abegeholt
werden.
```

4.6.4 get_beleg

FUNCTION:

```
get_beleg  Beleg(e) holen
  $find    WHERE fuer den SQL-Befehl
  $order   optionales ORDER für den SQL-Befehl
```

BEISPIEL:

```
$Datenbank->get_beleg ("","b_nr,b_datum");
$i = 0; while ($i < $Datenbank->sql_lines ())
{ $data = $Datenbank->sql_line($i); echo $data->b_text; $i++; }
```

ERGEBNIS:

```
Datenbank wird entsprechend den uebergebenen Parametern durchsucht und
die gefundenen Belege koennen ueber die Datenbankabfragen abegeholt
werden.
```

4.6.5 sum_beitrag

FUNCTION:

```
sum_beitrag gezahlten Beitrag aus Beitragskonto holen
```

BEISPIEL:

```
$SUMME = $Datenbank->sum_beitrag();
```

ERGEBNIS:

```
Es wird der gezahlte Beitrag zurueck gegeben
```

4.6.6 sum_finanz

FUNCTION:

```
sum_finanz Summen der Finanzkonten holen
```

BEISPIEL:

```
$KTO = $Datenbank->sum_finanz();
$SUMME = 0; foreach ($KTO as $NR => $WERT)
{
  echo "<bR>$NR ".$Datenbank->get_kontenname($NR)." = $WERT";
  $SUMME += $WERT;
}
echo "<bR>GESAMT = $SUMME";
```

ERGEBNIS:

```
Es wird ein Array mit dem Inhalt der Finanzkonten erzeugt. Als
Key wird die Konto-Nr. und als Inhalt der Kontostand zurueck ge-
geben. Beitragsbuchungen aus der Tabelle 'beitrag' werden berueck-
sichtigt.
```

4.6.7 sum_konto

FUNCTION:

```
sum_konto Summe fuer ein Konto
  $nr      Konto-Nr.
  $ziel    Zielkonto waehlen
```

ERGEBNIS:

```
Die Summe der fuer das als $nr angegebene Konto wird zurueck ge-
geben. Wenn $ziel <> 0 handelt es sich um alle Buchungen fuer
das Zielkonto, ansonsten wird nur das angegebene Konto gewaehlt.
```

4.6.8 test_log

FUNCTION:

```
test_log
Diese Funktion ist Klassen-Intern definiert und außerhalb der Klasse nicht
aufrufbar.
```

4.7 Tabelle: bu_konten +002+

NAME:

```
klassen/uverein-database.002.inc
```

KLASSE:

```
cl_uverein_database_002
Klasse fuer den Zugriff auf die Tabelle bu_konten
```

ABHAENGIG VON:

```
cl_database              (database.inc)
cl_uverein_database_001  (uverein-database.001.inc)
PHP 5
```

HISTORIE:

```
21.03.2009 FUNKTION konten_liste
22.03.2009 FUNKTION kontenname
10.04.2009 Tabelle bu_konten ausgelagert in cl_uverein_database_002
```

AUTOR:

```
Uwe Schied
```

BUGS:

```
keine bekannt
```

4.7.1 get_kontenklasse (Konto-Nr)

FUNCTION:

```
get_kontenklasse   Art (Klasse) des Kontos ermitteln
$nr                Konto-Nummer
```

4.7.2 get_kontenname (Konto-Nr)

FUNCTION:

```
get_kontenname     Name eines kontos ermitteln - bu_konten.name
$nr                Konto-Nummer
```

4.7.3 get_kontodaten (Konto-Nr)

FUNCTION:

```
get_kontodaten     Zeiger auf die Daten des/der Kontos/Konten
$nr                optionale Konto-Nummer
```

4.7.4 konten_liste (Kontenart)

FUNCTION:

```
konten_liste   Array mit allen Konten zu aktuellem Verein und Jahr
$kontenart     optionale Beschraenkung auf Kontenklasse
```

ERGEBNIS:

```
Es wird ein Zeiger auf die selektierten Elemente zurueck gegeben
```

4.7.5 konten_liste4selection (Kontenart)

FUNCTION:

```
konten_liste4selection Array mit den Konten zum aktuellen Verein und
                       Jahr aufbereitet fuer Selektionen
$kontenart             Kontenklasse
```

ERGEBNIS:

```
Es wird ein Array mit folgenden Daten zurueck gegeben:
  KEY        Konto-Nr
  VALUE      Bezeichnung der Kontenklasse aus bu_klassen
             + Bezeichnung des Konto's
```

4.7.6 konto_eintragen (Konto-Nr, Name, Klasse, Ziel, Jahr, Kontostand)

FUNCTION:

```
konto_eintragen   Neues Konto erfassen oder aendern
   $nr        Konto-Nr
```

```
$name      optionale Bezeichnung des Konto's
$klasse    optionale Kontenklasse (Voreinstellung = 0)
$ziel      optionales Konto, auf das der Abschluss erfolgt
           (Voreinstellung = Bilanz)
$jahr      optionale Angabe des Jahres
           (Voreinstellung = aktuelles Jahr)
stand      optionaler Kontostand (Voreinstellung 0)
```

ERGEBNIS:

```
Das Konto wurde entweder neu eingetragen oder aktualisiert
```

4.7.7 select_konten (WHERE-Abfrage, Sortierung)

FUNCTION:

```
select_konten      Name eines kontos ermitteln - bu_konten.name
$where             optionale zusätzliche Selektion
$order             optionale Sortierung
```

ERGEBNIS:

```
Es wird ein Zeiger auf das Abfrageergebnis zurueck gegeben. Wenn kein
weiteres Selektionsmerkmal mit uebergeben wird ist dies ein Zeiger
auf alle Konten des aktuellen Vereins im aktuellen Jahr.
```

4.8 Tabelle: virtual_konto +003+

NAME:

```
klassen/uverein-database.003.inc
```

KLASSE:

```
cl_uverein_database_003
Klasse fuer den Zugriff auf die Tabelle virtual_konto
```

ABHAENGIG VON:

```
Klasse                   Datei                     Hinweis
-------------------------------------------------------------------
cl_database              database.inc              Datenbank-Klasse
cl_uverein_database_001  uverein-database.001.inc Basis-Informationen
cl_uverein_database_002  uverein-database.002.inc bu_konten
PHP 5
```

BESCHREIBUNG:

```
Schluessel          Funktion           Kommentar
-------------------------------------------------------------------
BEITRAG             get_vbeitrag       Daten fuer die Zuordnung der
                    set_vbeitrag       Beitragszahlungen
SOLLBEITRAG         get_vsollbeitrag   Letzte Buchung des Beitrags-
                    set_vsollbeitrag   solls
FINANZKONTO         get_vfinanz        Welche Kontenart haben die
                    set_vfinanz        Finanzkonten?
SCHWEBEKONTO        get_vschwebe       Schwebekonto
                    set_vschwebe
UEBERTRAG           get_vuebertrag     Daten fuer den Uebertrag
                    set_vuebertrag
VORSTEUER           get_vsteuer        Daten fuer die Vorsteuer-
                    set_vsteuer        Buchung
```

HISTORIE:

```
06.06.2010 Beitragsrueckstand ergaenzt
10.01.2010 Fehler beim uebermitteln von 0-Werten korrigiert
03.01.2010 Beitragseinzug integriert
31.05.2009 Bug bei set_vfinanz bereinigt -> break entfernt
           Bug bei set_vschwebe bereinigt -> break entfernt
10.04.2009 Klasse erstellt
```

AUTOR:

```
Uwe Schied
```

BUGS:

```
keine bekannt
```

4.8.1 get_sollbeitrag ()

FUNCTION:

```
get_vsollbeitrag  liefert das Datum des letzten Beitragseinzuges im
                  Format JJJJMMTT
```

ERGEBNIS:

```
Es wird das Datum der letzten automatischen Beuchung des Soll-Beitrages
Ist noch keine automatische Buchung erfolgt, wird 0 zurueck geliefert.
```

4.8.2 get_vbeitrag (Feldbezeichnung)

FUNCTION:

```
get_vbeitrag   Zuordnungen zur Beitragszahlung lesen
$feld          Welches Feld soll ausgelesen werden
  Moegliche Werte:
  KONTO  Beitragskonto fuer die Buchhaltung = Voreinstellung
  FINANZ Finanzkonto fuer die Voreinstellung in der Mitgliederverwaltung
  KST    Kostenstelle fuer die Beitragsbuchungen
```

ERGEBNIS:

```
Es wird die gewuenschte Informationen zurueck gegeben
```

4.8.3 get_vfinanz ()

FUNCTION:

```
get_vfinanz   Kontenart der Finanzkonten
```

ERGEBNIS:

```
Es wird die gewuenschte Informationen zurueck gegeben
```

4.8.4 get_vmahnbrief (Feld)

FUNCTION:

```
get_vmahnbrief Mahnbrief holen
Feld    Welche Mahnung soll geholt werden?
  Moegliche Werte:
  1    = 1. Erinnerung
  2    = 2. Erinnerung
  3    = 3. Erinnerung
  99   = Lastschriftretoure
```

ERGEBNIS:

```
Der gewuenschte Brief wird zurueckgeliefert
```

4.8.5 get_vmahngebuehr (Feld)

FUNCTION:

```
get_vmahngebuehr Hoehe der Mahnkosten holen
Feld    Welche Mahnung soll geholt werden?
  Moegliche Werte:
  1    = 1. Erinnerung
  2    = 2. Erinnerung
  3    = 3. Erinnerung
  99   = Lastschriftretoure
```

ERGEBNIS:

```
Der gewuenschte Gebuher wird zurueckgeliefert
```

4.8.6 get_vschwebe ()

FUNCTION:

```
get_vschwebe   Konto-Nr. des Schwebekontos
```

ERGEBNIS:

```
Es wird die gewuenschte Informationen zurueck gegeben
```

4.8.7 get_vsteuer (Feldbezeichnung)

FUNCTION:

```
get_vsteuer    Zuordnungen des Vorsteuer-Kontos
$feld          Welches Feld soll ausgelesen werden
  Moegliche Werte:
  KONTO  Vorsteuerkonto-Nr = Voreinstellung
  FINANZ Finanzkonto fuer die Vorsteuer
  KST    Kostenstelle
```

ERGEBNIS:

```
Es wird die gewuenschte Informationen zurueck gegeben
```

4.8.8 get_vuebertrag (Feldbezeichnung)

FUNCTION:

```
get_vuebertrag Zuordnungen fuer den Uebertrag
$feld          Welches Feld soll ausgelesen werden
  Moegliche Werte:
  KLASSE Kontenart = Voreinstellung
  FINANZ Finanzkonto
  KST    Kostenstelle
```

ERGEBNIS:

```
Es wird die gewuenschte Informationen zurueck gegeben
```

4.8.9 intern_eintragen

FUNCTION:

```
intern_eintragen   Zuordnung eintragen
```

GENUTZT VON:

```
Diese Funktion ist nur klassenintern verfuegbar
```

ERGEBNIS:

```
Die Zuordnung wurde entweder neu eingetragen oder aktualisiert
```

4.8.10 intern_holen

FUNCTION:

```
intern_holen   Feld auslesen
```

GENUTZT VON:

```
Diese Funktion ist nur klassenintern verfuegbar
```

ERGEBNIS:

```
Es wird der Feldinhalt zurueck geliefert
```

4.8.11 set_vbeitrag (Wert, Feldbezeichnung)

FUNCTION:

```
set_vbeitrag   Zuordnungen zur Beitragszahltung setzen
$wert          Welcher Wert soll eingetragen werden
$feld          Welches Feld soll ausgelesen werden
  Moegliche Werte:
  KONTO  Beitragskonto fuer die Buchhaltung = Voreinstellung
  FINANZ Finanzkonto fuer die Voreinstellung in der Mitgliederverwaltung
  KST    Kostenstelle fuer die Beitragsbuchungen
```

ERGEBNIS:

```
Es wird die gewuenschte Aenderung in die Tabelle geschrieben
```

4.8.12 set_vfinanz (Kontenart)

FUNCTION:

```
set_vfinanz   Kontenart der Finanzkonten
```

ERGEBNIS:

```
Es wird die gewuenschte Aenderung in die Tabelle geschrieben
```

4.8.13 set_vmahnung (Art, Wert, Feld)

FUNCTION:

```
set_vmahnung   Mahnbriefe und -kosten verwalten
Art   Art der Mahnung
  1    = 1. Erinnerung
  2    = 2. Erinnerung
  3    = 3. Erinnerung
  99   = Lastschriftretoure
Wert  Welcher Wert soll eingetragen werden
Feld  Welches Feld soll ausgelesen werden
  Moegliche Werte:
  BRIEF   Nr. des Briefes aus der Tabelle "briefe"
  GEBUEHR Hoehe der Mahngebuehr
```

ERGEBNIS:

```
Es wird die gewuenschte Aenderung in die Tabelle geschrieben
```

4.8.14 set_vschwebe (Konto-Nr)

FUNCTION:

```
set_vschwebe   Konto-Nr. des Schwebekontos
```

ERGEBNIS:

```
Es wird die gewuenschte Aenderung in die Tabelle geschrieben
```

4.8.15 set_vsollbeitrag ()

FUNCTION:

```
set_vsollbeitrag  Setzt das Datum der letzten automatischen Buchung
                  des Soll-Beitrages im Format JJJJMMTT
```

4.8.16 set_vsteuer (Wert, Feldbezeichnung)

FUNCTION:

```
set_vsteuer    Zuordnungen des Vorsteuer-Kontos
$wert          Welcher Wert soll eingetragen werden
$feld          Welches Feld soll ausgelesen werden
  Moegliche Werte:
  KONTO  Vorsteuerkonto-Nr = Voreinstellung
  FINANZ Finanzkonto fuer die Vorsteuer
  KST    Kostenstelle
```

ERGEBNIS:

```
Es wird die gewuenschte Aenderung in die Tabelle geschrieben
```

4.8.17 set_vuebertrag (Wert, Feldbezeichnung)

FUNCTION:

```
set_vuebertrag Zuordnungen fuer den Uebertrag
$wert          Welcher Wert soll eingetragen werden
$feld          Welches Feld soll ausgelesen werden
  Moegliche Werte:
  KLASSE Kontenart
  FINANZ Finanzkonto
  KST    Kostenstelle
```

ERGEBNIS:

```
Es wird die gewuenschte Aenderung in die Tabelle geschrieben
```

5 Datensicherung

INFO:

```
Funktionen zur Datensicherung und -wiederherstellung
```

5.1 0.2.1.save.php

NAME:

```
0.2.1.save.php
```

GENUTZT VON:

```
allen Versionen >= 0.2.1
                < 0.2.3-1
```

BESCHREIBUNG:

```
Folgende Daten werden gesichert:

1. Vereinsdaten
   * Name des Vereins
   * Vereinsdaten fuer die Tabelle vereine
   * CREATE Vereins-Tabellen

2. Buchhaltung
   * Buchungsklassen
   * Buchungskonten
   * Kostenstellen
   * Buchungsbelege
   * Bilanz-Daten

3. Allgemeine Daten fuer die Mitgliederverwaltung
   * Anrede-Schluessel
   * Zahlungsarten
   * Banken mit Bankleitzahl
   * Zahlungsweise
   * Beitragsarten
   * Sonderangaben aus virtual_konto
   * Standard-Briefe

4. Steuer-Berechnung

5. Mitgliederverwaltung
   * Personen-Tabelle und Vereinstabelle
   * Einzugsermaechtigungen
   * Beitragsbuchungen
   * Informationen zur Beitragszahlung
```

5.2 __save__.php

NAME:

```
__save__.php
```

GENUTZT VON:

```
allen Versionen >= 0.2.3-1
```

BESCHREIBUNG:

```
Erstellt eine mit bzip komprimierte Datei mit den aktuellen Daten
zum aktiven Verein und stellt die Datei zum Download im Browser
bereit.
```

HISTORIE:

```
07.02.2009      Start mit Version 0.2.3-1
```

5.3 restore.php

BESCHREIBUNG:

```
Stellt den mit __save__.php gesicherten Zustand wieder her
```

TODO:

```
Zustand der Sicherung wieder herstellen - bisher wird die Sicherung
nur eingelesen, aber nicht in die Datenbank zurueck geschrieben.
```

HISTORIE:

```
07.02.2009      Start mit Version 0.2.3-1
```

6 dev-scripts

INFO:

```
Scripte und Daten fuer Entwickler
```

6.1 dev-scripts/metapackage

INFO:

```
Debian-Metapackages
```

6.1.1 metapackage/meta.info

NAME:

```
meta.info
  Kontroll-Datei fuer die automatische Erstellung von
  Debian-Metapackages des Projektes
```

GENUTZT VON:

```
read_meta.bin
```

BESTANDTEILE:

```
odbc            zusaetzliche Aktivierung der ODBC-Schnittstelle
develop         Entwickler-Paket
```

6.1.2 metapackage/read_meta.c

NAME:

```
read_meta.c
```

GENUTZT VON:

```
make
```

BESCHREIBUNG:

```
Das Programm liest die Datei meta.info aus und erstellt
die fuer Debian-Pakete erforderlichen Dateien
  - control
  - changelog
```

7 Finanzbuchhaltung

INFO:

```
Funktionen zur Durchfuehrung der Finanzbuchhaltung
```

7.1 buchung_beleg.php

NAME:

```
buchung_beleg.php
```

BESCHREIBUNG:

```
Erstellt eine Aufstellung der Buchungsbelege mit allen
Belegen des aktuellen Jahres
```

7.2 buchung_kto-beleg.php

NAME:

```
buchung_kto-beleg.php
```

BESCHREIBUNG:

```
Erstellt eine Aufstellung der Buchungsbelege fuer ein
einzelnes Buchungskonto.
```

7.3 change_year.inc

NAME:

```
change_year.inc
Die Include-Datei wird eingebunden, wenn ein Jahreswechsel
stattfindet.
```

ABHAENGIG VON:

```
buchung.php
```

GENUTZT VON:

```
buchung.php
```

7.4 fkonten.summe.inc

BESCHREIBUNG:

```
Finanzkonten innerhalb der Buchungen darstellen
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

BUGS:

```
keine bekannt
```

7.5 print.belege.php

NAME:

```
print.belege.php
```

BESCHREIBUNG:

```
Erstellt eine Aufstellung der Buchungsbelege als PDF-Dokument

Uebergabe-Parameter werden ueber Super-Globals uebermittelt

ART = Art der Aufstellung
  * BELEG     alle Buchungsbelege = Voreinstellung
  * KONTO     nur Buchungsbelege zu einem bestimmten Konto
  * KST       nur Buchungsbelege zu einer bestimmten Kostenstelle
  * NR        nur eine Beleg-Nummer
  * BU_KLASSE Buchungsbelege zur Kontenklasse

WERT = Nummer des Kontos, Kostenstelle, Klasse oder Beleg-Nr.
```

7.6 sonder-menue.php

NAME:

```
sonder-menue.php
```

BESCHREIBUNG:

```
Menü für Sonderbuchungen. Hierzu gehören z.B.

* Spendenverwaltung
* Vereinsausflüge
```

HISTORIE:

```
18.02.2012 Datei angelegt
```

7.7 sonder.php

NAME:

```
sonder.php
```

BESCHREIBUNG:

```
Sonderbuchungen erstellen und verwalten
```

HISTORIE:

```
19.02.2012 Datei angelegt
```

BESTANDTEILE:

```
TAB-Datei sonderdaten.0000.SONDERDATEIEN.tab
Übersicht über die Sonderdateien mit den Feldern:

AUSWAHL=NEU
Neuanlage eines Sonderkonto's
* Kurzbezeichnung des Sonder-Kontos (max. 30 Zeichen)
* Bezeichnung des Sonderkontos (z.B. Spenden)
* Buchungskonto für Einnahmen und Ausgaben
* Kostenstelle für Einnahmen und Ausgaben

AUSWAHL=ERFASSEN
Sonderkonto neu anlegen oder Änderung erfassen

AUSWAHL=alle anderen Werte
Daten in Sonderkonto bearbeiten
TAB-Datei sonderdaten.XXXXXX.tab
Sonderdatei mit Buchungsdaten
Daten anzeigen
Name   Hinweise  Soll-Betrag   Zahlung
                               Buchungs-Nr.  Datum  Betrag
```

8 formular.inc

KLASSE:

```
cl_formular
  Diese Klasse ist fuer die Verwaltung und Auswertung der Formulare zu-
  staendig. Alle Eingaben sollen durch diese Klasse abgebildet werden.
```

HISTORIE:

```
15.12.2007 Klasse erstellt
28.12.2007 Klasse um Checkbox ergaenzt
07.02.2009 Klasse um die Funktionen getupload und upload ergaenzt
12.03.2008 Klasse um Radiobutton ergaenzt
17.03.2008 Bug "fehlerhafte Target-Angabe" beseitigt
23.05.2009 Grafik-Button erhaelt variable Groesse (set_grafikwidth)
28.03.2010 Klasse um "setwert" ergaenzt
11.05.2010 Returnwert rechnet deutsche Umlaute um
13.08.2010 Sonderfall - Variable = Array in returnwert beruecksichtigt
20.08.2011 BUG 3392871 - SUB-Formular eingefuegt (Funktionen: open_sub
           und close_sub)
11.10.2014 Ticket #19 XSS-Lücke geschlossen
```

BUGS:

```
keine bekannt
```

8.1 cl_formular

VARIABLE:

Name	privat	Inhalt
$Name	ja	Name des Formulars
$formular_ziel	ja	Ziel-Datei fuer das Formular
$formular_target	ja	Ziel-Frame fuer das Formular
$FORMULAR	ja	Das Formular selbst
$INTERNE_VARIABLEN	ja	Welche Werte wurden an das Formular uebergeben
$tausend	nein	Trennzeichen fuer Tausender (deutsch z.B. ".")
$trenner	nein	Trennzeichen fuer Nachkommastellen (z.B. ",")
$set_grafikwidth	nein	Breite des Grafik-Buttons (Voreinstellung: 25)
$SUB_FORM	ja	Nummer des Unterformulars, falls vorhanden

8.1.1 checkbox (Name, Wert, Anzeigetext, ausgewaehlt)

FUNCTION:

```
checkbox ($name, $wert, $anzeige, [$checked])
  $name       Name der Checkbox
  $wert       Rueckgabewert, wenn ausgewaehlt
  $anzeige    Text fuer die Beschriftung
  $checked    optionaler Wert. Wenn etwas uebergeben wird, ist die Box
              ausgewaehlt - ansonsten wird der Wert aus der letzen Ueber-
              mittlung des Formulars genommen. Voreinstellung = nicht aus-
              gewaehlt.

Die Funktion stellt eine Checkbox mit Beschriftung zur Verfuegung.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$FORMULAR->checkbox ("box","OK","bitte anklicken");
```

8.1.2 cl_formular (Ziel, Frame)

FUNCTION:

```
cl_formular ([$ziel],[$target])
Konstruktor der Klasse. Zur Initialisierung koennen verschiedene optionale
Parameter uebergeben werden.
  $ziel     CGI-Script fuer die Auswertung des Formulars
  $target   Frame fuer den Aufruf des CGI-Scripts fuer die Auswertung
```

BEISPIEL:

```
* Ohne Parameter wird die aktuelle Datei und der aktuelle Frame als CGI-
  Script fuer die Auswertung verwandt.
    $FORMULAR = new cl_formular ();

* Wenn nur das Ziel angegeben wird, und kein Frame wird der aktuelle Frame
  beibehalten und das uebergebene Ziel als CGI-Script verwandt.
    $FORMULAR = new cl_formular ("auswerten.php");

* Wenn Ziel und Frame uebergeben wird, erfolgt die Auswertung durch das
  als Ziel uebergeben CGI-Script im als Target uebergebenen Frame.
    $FORMULAR = new cl_formular ("auswerten.php","_blank");

* Wenn die aktuelle Datei als CGI-Script verwandt werden soll, aber ein
  anderer Frame genutzt wird, muss NULL als Ziel-Parameter angegeben
  werden.
    $FORMULAR = new cl_formular (NULL,"_blank");
```

8.1.3 close_sub ()

FUNCTION:

```
close_sub ()
Die Funktion schliesst ein Formular
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
```

8.1.4 eingabe (Name, Feldlaenge, Vorgabewert)

FUNCTION:

```
eingabe ($name, [$len], [$vorgabe])
  $name     Name des Formular-Feldes
  $len      optionaler Parameter fuer die Feldlaenge (Voreinstellung: 80)
  $vorgabe  optionale Vorgabe fuer den Feld-Inhalt (Voreinstellung ist
            Inhalt lt. letztem Sendevorgang)

Die Funktion stellt ein Eingabefeld zur Verfuegung. Die Breite des Feldes
wird durch $len festgelegt. Falls $vorgabe mit uebergeben wird, ist dies
die Vorbelegung fuer das Eingabefeld. Falls $vorgabe nicht uebergeben wird
ist die Vorbelegung der Eintrag des letzten Sendevorganges oder falls noch
kein Sendevorgang stattgefunden hat ein leerer String.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$FORMULAR->eingabe ("eingabezeile");
```

8.1.5 feld Name, Breite, Hoehe, Vorgabewert)

FUNCTION:

```
feld ($name, [$breite], [$hoehe], [$vorgabe])
  $name        Name des Eingabefeldes
  $breite      Breite Feldes (nichts = 80)
  $hoehe       Hoehe des Feldes (nichts = 10)
  $vorgabe     Inhalt der Variablen (nichts = aus Superglobals holen)
```

```
Die Funktion stellt ein Eingabefeld fuer mehrzeiligen Text zur Verfuegung.
Breite und Hoehe des Feldes werden durch $breite und $hoehe festgelegt.
Falls $vorgabe mit uebergeben wird ist dies die Vorbelegung fuer das Feld.
Falls $vorgabe nicht uebergeben wird, ist die Vorbelegung der Eintrag beim
letzten Sendevorgang oder falls noch kein Sendevorgang stattgefunden hat
ein leerer String.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$FORMULAR->feld ("eingabefeld");
```

8.1.6 fokus (name)

FUNCTION:

```
fokus ($name)
  $name   Name des Eingabefeldes
```

ERGEBNIS:

```
Die Funktion gibt den Eingabefokus an das gewuenschte Eingabefeld
```

8.1.7 getupload (Name der Variablen)

FUNCTION:

```
getupload ($name)
  $name  Name der Variablen

Die Funktion stellt eine Auswahlbox fuer die Dateiauswahl auf dem Client
zur Verfuegung
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
upload.
```

BEISPIEL:

```
$FORMULAR->getupload ("DATEI");
```

8.1.8 grafikbutton (Wert, Pfad zur Grafikdatei

FUNCTION:

```
grafikbutton ($value, $pic)
  $value     Rueckgabewert, wenn Button gedrueckt wurde
  $pic       Pfad zur Grafikdatei
  $subform   Unterformular nutzen -> Voreinstellung true

Die Funktion stellt einen Sendebutton bereit, der statt Text eine Grafik
anzeigt. Nach der Formularuebermittlung wird $value uebergeben falls der
Button angeklickt wurde.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
sende_ergebnis.
```

BEISPIEL:

```
$FORMULAR->grafikbutton ("OK","../icons/ok.png");
```

8.1.9 hide (Name, Wert)

FUNCTION:

```
hide ($name, [$wert])
  $name    Name der Variablen
  $wert    Inhalt der Variablen (wenn kein Wert uebergeben wird,
           wird der Wert aus den Superglobals direkt ausgelesen)

Variablenwerte in Formular einfuegen, die nicht angezeigt werden
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$FORMULAR->hide ("geschuetzt","Ja");
```

8.1.10 inhalt (Text)

FUNCTION:

```
inhalt ($text)
  $text  Text, der im Formular ausgegeben werden soll

Diese Funktion fuegt Text und ggf. HTML-Tags in das Formular ein. Dadurch
wird der Vordruck insgesamt uebersichtlicher.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
```

BEISPIEL:

```
$FORMULAR->inhalt ("<HR>Bemerkungsfeld:<BR>");
$FORMULAR->feld   ("bemerkungen");
$FORMULAR->inhalt ("<BR><HR>");
```

8.1.11 INT_HTML_2_UML (text)

FUNCTION:

```
INT_HTML_2_UML ($text)
  $text    Text der umgerechnet werden soll

HTML-Code wird als Umlaut zurueck gegeben
```

8.1.12 INT_UML_2_HTML (text)

FUNCTION:

```
INT_UML_2_HTML ($text)
  $text    Text der umgerechnet werden soll

Umlaute werden als HTML-Code zurueck gegeben
```

8.1.13 numeingabe2anzeige (Zahl, Nachkommastellen)

FUNCTION:

```
numeingabe2anzeige ($zahl, $nachkomma)
  $zahl      Zahl, die fuer die Anzeige optimiert werden soll
  $nachkomma darzustellende Nachkommastellen

Diese Funktion wandelt die uebergebene Zahl fuer die Darstellung am Bild-
schirm um. Die Zahl kann bereits formatiert oder unformatiert uebergeben
werden.
```

ERGEBNIS:

```
Die Funktion liefert als Rueckgabe die formatierte Zahl
```

BEISPIEL:

```
echo $FORMULAR->numeingabe2anzeige ("10.9",2); // gibt 10,90 aus
echo $FORMULAR->numeingabe2anzeige ("10,9",2); // gibt 10,90 aus
```

8.1.14 open_sub (ziel, target)

FUNCTION:

```
open_sub ([$ziel], [$target])
  $ziel     CGI-Script fuer die Formularauswertung
            Voreinstellung ist das Script des aktuellen Formulars
  $target   Frame fuer den Aufruf des CGI-Scripts
            Voreinstellung ist der Frame des aktuellen Formulars

Die Funktion oeffnet ein Unterformular
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
```

8.1.15 password(Name

FUNCTION:

```
password ($name, $len)
  $name     Name des Formular-Feldes
  $len      optionaler Parameter fuer die Feldlaenge (Voreinstellung: 80)

Die Funktion stellt ein Eingabefeld fuer Passwoerter zur Verfuegung.
Das Passwort wird waehrend der Eingabe nicht angezeigt
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$FORMULAR->password ("PW");
```

8.1.16 radio_button (Name, Wert, Anzeigetext, ausgewaehlt)

FUNCTION:

```
radio_button ($name, $wert, $anzeige, [$checked])
  $name     Name des Feldes mit Radio-Buttons
  $wert     Rueckgabewert
  $anzeige  das was angezeigt wird
  $checked  ausgewaehlt ? Voreinstellung = Nein bzw. Superglobals
            Jede Angabe bedeutet ja

Die Funktion stellt Radio-Buttons zur Verfuegung. Durch $name wird festge-
legt, welche Buttons zusammen gehoeren.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$FORMULAR->inhalt ("Zugriffsart waehlen:");
$FORMULAR->radio_button ("dbart","prod","produktiv","X");
$FORMULAR->radio_button ("dbart","test","Test");
```

8.1.17 returnglobal (Name

FUNCTION:

```
returnglobal (Name, Zahl = "")
  Name     Name der Variablen
  Vorgabe  Statt Leerzeichen soll Vorgabewert geliefert werden
           Wenn die Vorgabe eine Zahl ist, wird ein numerischer
           Wert ausgelesen.

Mit dieser Funktion wird das Formular nach der Uebermittlung ausgelesen -
falls der Wert nicht im Formular uebergeben wurde wird zusaetzlich ge-
prueft ob das Feld ueber die Browser-Funktion direkt uebergeben wurde und
ggf. eingelesen.
```

ERGEBNIS:

```
Die Funktion gibt den Inhalt des Formularfeldes bzw. alternativ einen
direkt uebergebenen Wert zurueck. Ist das Feld leer kommt als Ergebnis ein
leerer String zurueck.
```

BEISPIEL:

```
$eingabe = $FORMULAR->returnglobal ("eingabe");
```

8.1.18 returnwert (Name)

FUNCTION:

```
returnwert ($name)
  $name    Name der Variablen

Mit dieser Funktion wird das Formular nach Uebermittlung ausgelesen.
```

ERGEBNIS:

```
Die Funktion gibt den Inhalt des Formularfeldes zurueck. Ist das Feld leer
oder wurde das Formular noch nicht abgeschickt kommt als Ergebnis ein
leerer String zurueck.
```

BEISPIEL:

```
eingabe = $FORMULAR->returnwert ("eingabe");
```

8.1.19 returnzahl (Name)

FUNCTION:

```
returnzahl ($name)
  $name    Name der Variablen

Mit dieser Funktion wird das Formular nach Uebermittlung ausgelesen.
```

ERGEBNIS:

```
Die Funktion gibt den Inhalt des Formularfeldes zurueck. Ist das Feld leer
oder wurde das Formular noch nicht abgeschickt kommt als Ergebnis die Zahl
0 zurueck.
```

BEISPIEL:

```
eingabe = $FORMULAR->returnzahl ("eingabe");
```

8.1.20 schliessen ()

FUNCTION:

```
schliessen ()

Diese Funktion beendet das Formular und liefert es als Rueckgabe.
```

ERGEBNIS:

```
Die Funktion gibt das komplette Formular zurueck, das weiterverarbeitet
oder ausgegeben werden kann.
```

BEISPIEL:

```
$FORMULAR = new cl_formular ();
$FORMULAR->inhalt ("<HR>Was tun?<BR>");
$FORMULAR->senden ("Nichts");
$FORMULAR->inhalt (" ");
$FORMULAR->senden ("Etwas");
echo $FORMULAR->schliessen();
```

8.1.21 selection (Name, Array mit Selektion, selektiertes Element, Zeilenzahl)

FUNCTION:

```
selection ($name, $sel, [$selected], [$size])
  $name      Name der Variablen
  $sel       Array mit Selectionsangaben
  $selected  Optional: Welches Element ist selektiert ?
             (Voreinstellung = 1. Element bzw. lt. Superglobals)
  $size      Optional: Anzahl der angezeigten Selection
             (Voreinstellung = 1)

Diese Funktion stellt ein Selektionsfeld zur Verfuegung. Die Daten fuer
die einzelnene Selektionszeilen werden als Array ($sel) uebergeben. Der
Schluessel des Array's wird hierbei als Rueckgabewert und der Array-Inhalt
fuer die Anzeige verwandt.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktionen
returnwert und/oder returnzahl.
```

BEISPIEL:

```
$SEL = array ("OK" => "Alles OK", "KAPUT", "zerstoert");
$FORMULAR->selection ("was",$SEL);
```

8.1.22 sende_ergebnis ()

FUNCTION:

```
sende_ergebnis ()

Diese Funktion gibt das Ergebnis der letzten Uebermittlung des Formulars
zurueck. Ausgelesen werden die die Werte aus den Buttons
```

```
  * senden
  * grafikbutton
```

ERGEBNIS:

```
Die Funktion gibt den uebermittelten Wert des gedrueckten Buttons zurueck.
Wenn kein Button gedrueckt wurde wird ein leerer String zurueck gegeben.
```

BEISPIEL:

```
$auswahl = $FORMULAR->sende_ergebnis ();
```

8.1.23 senden (Name)

FUNCTION:

```
senden ($name)
  $name     Text, der angezeigt wird = gleichzeitig Ergebnis

Diese Funktion stellt einen Sende-Button bereit, der mit dem uebergebenen
Text beschriftet wird. Der Text ist gleichzeitig der Rueckgabewert, wenn
der Button angeklickt wird.
```

ERGEBNIS:

```
Die Funktion hat keinen Rueckgabewert.
Die Auswertung erfolgt nach Formularuebermittlung ueber die Funktion
sende_ergebnis.
```

BEISPIEL:

```
$FORMULAR->senden ("OK");
$FORMULAR->senden ("Abbruch");
```

8.1.24 setwert (Name, inhalt)

FUNCTION:

```
setwert ($name,$inhalt)
  $name    Name der Variablen
  $inhalt  Wert, der fuer die Variable gesetzt werden soll
```

ERGEBNIS:

```
Der Inhalt wird gesetzt und bei allen anschliessenden Leseoperationen
zurueckgeliefert.
```

BEISPIEL:

```
$FORMULAR->setwert("wasislos","Nix is los");
```

8.1.25 sub_grafikbutton (Button, Daten-Array, Ziel, Frame)

FUNCTION:

```
sub_grafikbutton ($Button, $Uebermittlung, [$Ziel], [$Target])
  $Button         Pfad zum Grafikbutton
  $Uebermittlung  Array mit Variablen, die durch dieses Unter-
                  Formular uebermittelt werden sollen
  $Ziel           Optionales Ziel des Unterformulars
  $Target         Optionaler Frame fuer das Ziel

Diese Funktion erstellt einen Textbutton, der in ein Unterformular
eingebettet wird
```

BEISPIEL:

```
$FORMULAR = new cl_formular ();
$FORMULAR->inhalt ("<HR>Was tun?<BR>");
$FORM_DATA = array ("AUSWAHL" => "OK", "NR" => 2);
$FORMULAR->sub_grafikbutton ("../icons/ok.png",$FORM_DATA);
$FORMULAR->inhalt (" ");
$FORMULAR->senden ("Etwas");
echo $FORMULAR->schliessen();
```

8.1.26 sub_mixbutton (Button, Anzeige, Daten-Array, Ziel, Frame)

FUNCTION:

```
sub_mixbutton ($Button,$Anzeige,$Uebermittlung,[$Ziel],[$Target])
  $Button         Pfad zum Grafikbutton
  $Anzeige        Text, zum Button ausgebeben wird
  $Uebermittlung  Array mit Variablen, die durch dieses Unter-
                  Formular uebermittelt werden sollen
  $Ziel           Optionales Ziel des Unterformulars
  $Target         Optionaler Frame fuer das Ziel

Diese Funktion erstellt einen Grafikbutton mit Beschriftung, der in ein
Unterformular eingebettet wird

Achtung: Die im Daten-Array uebergebenen Variablen koennen nur ueber
         returnglobal ausgelesen werden und werden nur uebermittelt, wenn
         dieser Button angeklickt wird
```

BEISPIEL:

```
$FORMULAR = new cl_formular ();
$FORMULAR->inhalt ("<HR>Was tun?<BR>");
$FORM_DATA = array ("AUSWAHL" => "OK", "NR" => 2);
$FORMULAR->sub_mixbutton ("../icons/ok.png","TESTING",$FORM_DATA);
```

```
$FORMULAR->inhalt (" ");
$FORMULAR->senden ("Etwas");
echo $FORMULAR->schliessen();
```

8.1.27 sub_textbutton (Anzeige, Daten-Array, Ziel, Frame)

FUNCTION:

```
sub_textbutton ($Anzeige, $Uebermittlung, [$Ziel], [$Target])
  $Anzeige        Text, der angezeigt wird
  $Uebermittlung  Array mit Variablen, die durch dieses Unter-
                  Formular uebermittelt werden sollen
  $Ziel           Optionales Ziel des Unterformulars
  $Target         Optionaler Frame fuer das Ziel

Diese Funktion erstellt einen Textbutton, der in ein Unterformular
eingebettet wird
```

BEISPIEL:

```
$FORMULAR = new cl_formular ();
$FORMULAR->inhalt ("<HR>Was tun?<BR>");
$FORM_DATA = array ("AUSWAHL" => "OK", "NR" => 2);
$FORMULAR->sub_textbutton ("OK ich mach was",$FORM_DATA);
$FORMULAR->inhalt (" ");
$FORMULAR->senden ("Etwas");
echo $FORMULAR->schliessen();
```

8.1.28 test ()

FUNCTION:

```
test ()

Diese Funktion ist nur fuer den Formular-Test bestimmt und sollte daher im
endgueltigen Dokument auskommentiert oder entfernt werden.
```

ERGEBNIS:

```
Die Funktion gibt alle Formularfelder mit Vorgabewert sowie alle durch den
letzten Sendevorgang uebermittelten Felder zurueck.
```

BEISPIEL:

```
echo $FORMULAR->test();
```

8.1.29 upload (Name der Variablen, Server-Pfad)

FUNCTION:

```
upload ($name, $uploadname)
  $name        Name der Variablen
  $uploadname  Pfad incl. Name unter der die Datei auf den Server hoch-
               geladen wird.
```

ERGEBNIS:

```
Die Funktion laedt die durch getupload ausgewaehlte Datei auf den Server
hoch.
```

BEISPIEL:

```
$FORMULAR->upload ("DATEI","/tmp/serverdatei");
```

9 funktionen.php

NAME:

```
funktionen.php
Die Datei enthaelt die global verwendete Funktionen-Bibliothek
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

BUGS:

```
kein BUG bekannt
```

10 Installation

INFO:

```
Installation und Update des Programms
```

10.1 db_update_blz.sh

NAME:

```
bash/db-update_blz.sh
  Das Script installiert die die Bankdaten in die Tabelle
  key_banken.

  Die Daten stammen von der DEUTSCHEN BUNDESBANK
  http://www.bundesbank.de
```

BESTANDTEILE:

```
/etc/uverein/config
  hier werden folgende Informationen abgelegt:
    [BLZ.DE.STAND]   Stand der Bankleitzahlendatei = gueltig ab
    [BLZ.DE.INSERT]  YES = Daten einbinden
                     NO  = Daten nicht einbinden
uverein.config
  hierbei handelt es sich um einen Link auf /etc/uverein/config

sql/blz-de.sql
  alle deutschen Bankleitzahlen als TAB-getrennte Datei
  Spalte 1  A = aktuell
            D = geloescht
  Spalte 2  BIC / SWIFT der Bank
  Spalte 3  Bankleitzahl
  Spalte 4  Spalte 1 = A: Bezeichnung der Bank
            Spalte 1 = D: Neue Bankleitzahl der Bank

sql/blz-de.diff.sql
  alle Aenderungen der Bankdaten seit der letzten Aktualisierung
  als TAB-getrennte Datei
  Die Spalten haben die gleiche Bedeutung wie bei sql-de.sql
```

BUGS:

```
keine bekannt
```

10.2 install.sql

NAME:

```
sql/install.sql
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
28.05.2007 Steuerabrechnung eingefuegt
           * tax
           * tax_base
           * tax_konten
20.08.2007 kurztext in ergaenzt
20.08.2007 bu_nr in tax ergaenzt
04.10.2008 Feature-Request 1293970 (optionale Felder) eingetragen
           * opt_tabledef
           * opt_tabledata
22.01.2009 Jahr in bu_konten ergaenzt
13.04.2009 komplett ueberarbeitet
03.11.2012 Feature-Request 3054813 offene Ueberweisungen verwalten
15.12.2013 Versionsverwaltung in die Datenbank integriert
27.12.2013 Mitgliederverwaltung angepasst
31.05.2014 Script für die variable Verwendung (neu und Update) vorbereitet
```

BUGS:

```
Keine bekannt
```

10.2.1 beitrag

INFO:

```
In diese Tabelle werden die Beitragsbuchungen eingetragen
```

```
CREATE TABLE IF NOT EXISTS beitrag (
 bu_nr      SERIAL,
 v_name     VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
 m_nr       INTEGER NOT NULL REFERENCES personen(nr),
 jahr       VARCHAR(10) NOT NULL DEFAULT EXTRACT(YEAR FROM CURRENT_DATE),
 f_konto    NUMERIC(10,0) NOT NULL DEFAULT '0',
 datum      DATE DEFAULT current_date,
 soll       NUMERIC(30,2) DEFAULT '0.00',
 haben      NUMERIC(30,2) DEFAULT '0.00',
 bu_text    TEXT
);
```

BESCHREIBUNG:

```
   v_name     betroffener Verein
   m_nr       Mitglieds-Nr.
   jahr       Buchungsjahr
   f_konto    Finanzkonto
   datum      Buchungsdatum
   soll       Sollbuchung - fliesst nicht in die Finanzbuchhaltung
   haben      Habenbuchung - fliesst in die Finanzbuchhaltung
   bu_text    Freier Erlaeuterungstext
```

10.2.2 beitrags_art

INFO:

```
   Die Tabelle enthaelt die Beitragsarten des Vereins. Z.B.
     * Ehrenmitglied
     * passives Mitglied
     * Mitgliedsbeitrag
```

```
CREATE TABLE IF NOT EXISTS beitrags_art (
 nr            SERIAL,
 verein        VARCHAR(20) NOT NULL REFERENCES vereine (kurzname),
 art           TEXT,
 jahresbeitrag NUMERIC(30,2) DEFAULT '0.00'
);
```

BESCHREIBUNG:

```
   nr            Nummer fuer die Beitragsart - wird automatisch vergeben
   verein        Fuer welchen Verein gilt die Beitragsart
   art           Beschreibung der Beitragsart, die in der Mitgliederverwaltung
                 angegeben wird
   jahresbeitrag Jahresbeitrag
```

10.2.3 beitrags_konto

INFO:

```
   Die Tabelle enthaelt Angaben zu Beitragsfaelligkeiten
```

```
CREATE TABLE IF NOT EXISTS beitrags_konto (
 nr            SERIAL,
 v_name        VARCHAR(20) NOT NULL REFERENCES vereine (kurzname),
 m_nr          INTEGER NOT NULL REFERENCES personen (nr),
 faellig_ab    DATE DEFAULT current_date,
```

```
 art           INTEGER NOT NULL REFERENCES beitrags_art (nr),
 zahlungsweise INTEGER NOT NULL REFERENCES zahlungsweise (nr)
);
```

BESCHREIBUNG:

```
   nr            Fortlaufende Nr. - dient als eindeutiger Identifizierer
   v_name        betroffener Verein
   m_nr          Mitglieds-Nr
   faellig_ab    1. Faelligkeitsdatum
   art           Beitragsart
   zahlungsweise Zahlungsweise
```

10.2.4 bilanz

INFO:

```
   Informationen zur Gewinn- und Verlustrechnung
```

```
CREATE TABLE IF NOT EXISTS bilanz (
 verein     VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
 jahr       VARCHAR(10) NOT NULL DEFAULT EXTRACT(YEAR FROM CURRENT_DATE),
 summe      NUMERIC(30,0) NOT NULL,
 waehrung   VARCHAR(10) NOT NULL DEFAULT 'EUR',
 schlusstxt TEXT,
 prueftxt   TEXT,
 datum      DATE NOT NULL DEFAULT CURRENT_DATE
);
```

BESCHREIBUNG:

```
   verein     betroffener Verein
   jahr       Buchungsjahr
   summe      Endsumme der Gewinn- und Verlustrechnung = Uebertrag ins
              naechste Buchungsjahr
   waehrung   Waehrung, in der gebucht wird -> Voreinstellung = EUR
   schlusstxt Schlusstext, der auf der Gewinn- und Verlustrechnung direkt nach
              den Ergebnissen aufgedruckt wird
   prueftxt   Informationstext zur Pruefung der Gewinn- und Verlustrechnung
   datum      Datum, an dem die Buchungen geprueft wurden
```

10.2.5 briefe

INFO:

```
   Brieftexte fuer verschiedene Anlaesse
```

```
CREATE TABLE IF NOT EXISTS briefe (
 verein       VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
 nr           INT4,
 info         VARCHAR(40) NOT NULL,
 inhalt       TEXT
);
```

BESCHREIBUNG:

```
   verein       betroffener Verein
   nr           Brief-Nr - auf dieses Feld wird referenziert
   info         Kurzbeschreibung des Briefes
   inhalt       Brieftext incl. Platzhaltern. Die folgenden Platzhalter sind
                definiert:
                %%BEITRAG%%      stellt eine Tabelle in den Brief mit allen
                                 Beitragsbuchungen incl. Buchungstext
                %%TERMIN28%%     wird im Brieftext durch das aktuelle Datum
                                 + 28 Tage ersetzt
```

10.2.6 bu_belege

INFO:

```
   Diese Tabelle enthaelt die Buchungsbelege fuer alle Vereine
```

```
CREATE TABLE IF NOT EXISTS bu_belege (
 idx          SERIAL,
 b_verein     VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
 b_nr         NUMERIC(10,0) NOT NULL DEFAULT 0,
 b_jahr       VARCHAR(10) NOT NULL DEFAULT EXTRACT(YEAR FROM CURRENT_DATE),
 b_datum      DATE DEFAULT current_date NOT NULL,
 b_konto      NUMERIC(10,0) NOT NULL,
 f_konto      NUMERIC(10,0) NOT NULL,
 b_betrag     NUMERIC(20,2) NOT NULL,
 b_steuer     NUMERIC(20,2),
 b_text       TEXT,
 b_kst        NUMERIC(10,0) NOT NULL,
 erledigt     BOOLEAN NOT NULL DEFAULT FALSE
);
```

BESCHREIBUNG:

```
   b_verein     betroffener Verein
   b_nr         Buchungs-Nr. (ACHTUNG: Nr. 1 ist fuer den Uebertrag reserviert
                und wird beim Jahreswechsel ueberschrieben)
   b_jahr       Buchungsjahr
   b_datum      Buchungsdatum
   b_konto      Buchungskonto
```

```
   f_konto      Finanzkonto
   b_betrag     Buchungsbetrag (brutto)
   b_steuer     Im Buchungsbetrag enthaltene Vorsteuer
   b_text       Buchungstext
   b_kst        Kostenstelle
   erledigt     Kennzeichnung fuer offene Ueberweisungen
                TRUE  = erledigt
                FALSE = Ueberweisung ist offen
```

10.2.7 bu_klassen

INFO:

```
   Diese Tabelle enthaelt die einzelnen Bereiche, in die die Konten aufge-
   teilt werden
```

```
CREATE TABLE IF NOT EXISTS bu_klassen (
 verein      VARCHAR(20) NOT NULL REFERENCES vereine (kurzname),
 kl_nr       NUMERIC(10,0) NOT NULL,
 kl_name     TEXT
);
```

BESCHREIBUNG:

```
   verein      betroffener Verein
   kl_nr       Nummer des Bereichs - auf diesen Wert wird referenziert
   kl_name     Beschreibung des Bereichs.
               z.B: - Ideeler Bereich
                    - Zweckbetrieb
                    - Finanzkonten
```

10.2.8 bu_konten

INFO:

```
   Diese Tabelle enthaelt die Buchungskonten
```

```
CREATE TABLE IF NOT EXISTS bu_konten (
 verein       VARCHAR(20) NOT NULL REFERENCES vereine (kurzname),
 kto_nr       NUMERIC(10,0) NOT NULL,
 kto_name     TEXT NOT NULL,
 kto_klasse   NUMERIC(10,0) NOT NULL,
 kto_ziel     NUMERIC(10,0) NOT NULL,
 kto_stand    NUMERIC(25,2),
 jahr         VARCHAR(10)   NOT NULL DEFAULT EXTRACT(YEAR FROM CURRENT_DATE)
);
```

BESCHREIBUNG:

```
   verein       betroffener Verein
   kto_nr       Konto-Nr.
   kto_name     Beschreibung des Kontos
   kto_klasse   Kontenbereich - siehe bu_klassen
   kto_ziel     Auf welches Konto wird dieses Konto abgeschlossen? Wenn
                kto_ziel und kto_nr identisch sind erfolgt der Abschluss
                in der Gewinn- und Verlustrechung, ansonsten handelt es
                sich um ein Unterkonto von kto_ziel
   kto_stand    Kontostand (wird z.Zt. nicht gepflegt)
   jahr         Buchungsjahr
```

10.2.9 bu_kst

INFO:

```
   Diese Tabelle enthaelt die Kostenstellen
```

```
CREATE TABLE IF NOT EXISTS bu_kst (
 verein       VARCHAR(20) NOT NULL REFERENCES vereine (kurzname),
 kst          NUMERIC(10,0) NOT NULL,
 kst_name     TEXT DEFAULT '+' NOT NULL
);
```

BESCHREIBUNG:

```
   verein       betroffener Verein
   kst          Nr. der Kostenstelle
   kst_name     Beschreibung der Kostenstelle
```

10.2.10 key_anrede

INFO:

```
   Die Tabelle enthaelt die Anredeschluessel
```

```
CREATE TABLE IF NOT EXISTS key_anrede (
 schluessel SMALLINT PRIMARY KEY,
 brief      TEXT NOT NULL,
 inhalt     TEXT NOT NULL
);
```

BESCHREIBUNG:

```
   schluessel  auf diesen Wert wird referenziert
   brief       Anrede fuer den Brief - z.B. Herr
   inhalt      Anrede im Brief - z.B. Sehr geehrter Herr
```

10.2.11 key_banken

INFO:

```
   Diese Tabelle enthaelt Bankleitzahlen und Banknamen. Die Angaben aus
   der Tabelle werden fuer die Tabellen "vereine" und "*_einzug" benoetigt.
```

```
CREATE TABLE IF NOT EXISTS key_banken (
 blz  NUMERIC(8,0) NOT NULL PRIMARY KEY,
 bank TEXT NOT NULL
);
```

BESCHREIBUNG:

```
   blz  Bankleitzahl
   bank Name der Bank
```

10.2.12 key_zahlungsarten

INFO:

```
   Die Tabelle enthaelt die Zahlungsarten fuer alle Vereine
```

```
CREATE TABLE IF NOT EXISTS key_zahlungsarten (
 nr          SMALLINT PRIMARY KEY,
 bezeichnung TEXT NOT NULL
);
```

BESCHREIBUNG:

```
   nr           auf diesen Wert wird referenziert
   bezeichnung  Beschreibung der Zahlungsart - z.B. per Rechnung
```

10.2.13 mitglied

INFO:

```
   Mitgliederverwaltung
```

```
 CREATE TABLE IF NOT EXISTS mitglied (
  verein       VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
  nr           INT8        NOT NULL REFERENCES personen(nr)          DEFAULT 1,
  eintritt     DATE        NOT NULL                                  DEFAULT CURRENT_DATE,
  ende         DATE,
  kommentar    TEXT,
```

```
  zahlungsart  SMALLINT    NOT NULL REFERENCES key_zahlungsarten(nr)  DEFAULT 1,
  bic          TEXT,
  iban         TEXT,
  bank         TEXT,
  unterschrift DATE        NOT NULL                                  DEFAULT CURRENT_DATE,
  lastschrift  INT8        NOT NULL                                  DEFAULT 0,
  aktiv        BOOLEAN                                               DEFAULT true
);
 ALTER TABLE mitglied OWNER TO :datenbankuser;
 CREATE UNIQUE INDEX mitglied_idx ON mitglied (verein,nr);
```

10.2.14 opt_tabledata

INFO:

```
   Tabelle mit den Inhalten der optionalen Felder zur Mitgliederverwaltung

CREATE TABLE IF NOT EXISTS opt_tabledata (
  opt_table     BIGINT NOT NULL REFERENCES opt_tabledef (nr),
  person        BIGINT NOT NULL REFERENCES personen (nr),
  inhalt        TEXT
);
```

BESCHREIBUNG:

```
   opt_table  Welches optionales Feld? - Referenz auf "opt_tabledata"
   person     Welches Mitglied? - Referenz auf "personen"
   inhalt     Inhalt des optionalen Feldes
```

10.2.15 opt_tabledef

INFO:

```
   Definitionstabelle fuer weitere Felder in der Mitgliederverwaltung

CREATE TABLE IF NOT EXISTS opt_tabledef (
  nr            SERIAL UNIQUE,
  verein        VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
  bezeichnung   VARCHAR(50) NOT NULL,
  beschreibung  TEXT
);
```

BESCHREIBUNG:

```
   nr            Eindeutiges Kennzeichen - existiert vereinsuebergreifend nur
                 einmal in der Tabelle
   verein        betroffener Verein
   bezeichnung   Kurzbezeichnung des Feldes (muss gefuellt werden)
   beschreibung  Ausfuehrliche Beschreibung des Feldes (kann leer bleiben)
```

10.2.16 personen

INFO:

```
Diese Tabelle enthaelt alle vereinsuebergreifenden Angaben zu den
einzelnen Mitgliedern
```

```
CREATE TABLE IF NOT EXISTS personen (
 nr         SERIAL UNIQUE,
 anrede     SMALLINT DEFAULT 1 NOT NULL,
 titel      TEXT,
 name       TEXT NOT NULL,
 vorname    TEXT NOT NULL,
 geboren    DATE DEFAULT '1800-01-01' NOT NULL,
 strasse    TEXT,
 hausnr     TEXT,
 land       VARCHAR(40),
 plz        NUMERIC(5,0),
 ort        TEXT,
 telefon    TEXT,
 email      TEXT
);
```

BESCHREIBUNG:

```
nr         Personen-Nr. = Mitglieds-Nr.
anrede     Anrede - Referenz auf key_anrede
titel      Akademischer Titel
name       Nachname
vorname    Vorname
geboren    Geburtsdatum (unbekannt = 01.01.1800)
strasse    Strasse
hausnr     Hausnummer
land       Land
plz        Postleitzahl
ort        Ort
telefon    Telefon-Nr.
email      EMail-Adresse
```

10.2.17 tax

INFO:

```
Diese Tabelle enthaelt Daten zur Umsatzsteuer-Erklaerung
```

```
CREATE TABLE IF NOT EXISTS tax (
  verein        VARCHAR(20)    NOT NULL REFERENCES vereine(kurzname),
```

```
  jahr          NUMERIC (10,0) NOT NULL DEFAULT EXTRACT (YEAR FROM CURRENT_DATE),
  datum         DATE           NOT NULL DEFAULT CURRENT_DATE,
  art           INT            NOT NULL,
  umsatz        NUMERIC (20,2),
  steuer        NUMERIC (20,2),
  bu_nr         NUMERIC (10,0) DEFAULT 0
);
```

BESCHREIBUNG:

```
   verein       betroffener Verein
   jahr         Buchungsjahr
   datum        Datum, an dem die Steuererklaerung bzw. Voranmeldung ans
                Finanzamt gesandt oder elektronisch uebertragen wurde
   art          Steuerart - siehe tax_base
   umsatz       Netto-Umsatz
   steuer       Steuer-Betrag
   bu_nr        Buchungs-Nr. unter der die Steuerabrechnung in der Buchhaltung
                eingetragen ist. 0 = noch nicht eingetragen
```

10.2.18 tax_base

INFO:

```
   Diese Tabelle enthaelt Informationen ueber die Steuer-Arten pro Verein
```

```
CREATE TABLE IF NOT EXISTS tax_base (
  verein        VARCHAR(20)     NOT NULL REFERENCES vereine(kurzname),
  nr            INT             NOT NULL DEFAULT 0,
  name          TEXT            NOT NULL,
  steuersatz    NUMERIC(8,4),
  rundung       INT             NOT NULL DEFAULT 2,
  kurztext      TEXT
);
```

BESCHREIBUNG:

```
   verein     betroffener Verein
   nr         Nr. der Steuerart - auf dieses Feld verweisen
                * tax.art
                * tax_konten.steuersatz
   name       Bezeichnung der Steuerart
   steuersatz Steuersatz
   rundung    Auf wieviele Stellen hinter dem Komma wird die Steuer gerundet?
   kurztext   Hinweise - optionale Angaben
```

10.2.19 tax_konten

INFO:

```
   Diese Tabelle enthaelt die Zuordnung der Buchungskonten zu den einzelnen
   Steuerarten

CREATE TABLE IF NOT EXISTS tax_konten (
  verein        VARCHAR(20)     NOT NULL REFERENCES vereine(kurzname),
  steuersatz    INT             NOT NULL,
  kto           NUMERIC(10,0)   NOT NULL
);
```

BESCHREIBUNG:

```
   verein     betroffener Verein
   steuersatz Steuerart - siehe tax_base
   kto        Konto-Nr., der die Steuerart zugeordnet wird
```

10.2.20 vereine

INFO:

```
   Diese Tabelle enthaelt Angaben zu den vorhandenen Vereinen

CREATE TABLE IF NOT EXISTS vereine (
 name       TEXT NOT NULL,
 kurzname   VARCHAR(20) NOT NULL UNIQUE PRIMARY KEY,
 strasse    TEXT,
 ort        TEXT,
 blz        VARCHAR(10),
 bank       TEXT,
 kontonr    NUMERIC(15,0),
 iban       TEXT,
 bic        TEXT,
 identnr    TEXT,
 email      TEXT,
 version    TEXT DEFAULT '0.2.0'
);
```

BESCHREIBUNG:

```
   name       Vereinsname
   kurzname   Eindeutiger Name des Vereins, auf den referenziert wird
   strasse    Strasse des Vereins
   ort        Ort des Vereins
   blz        Bankleitzahl des Vereinskontos
   bank       Bankbezeichnung des Vereinskontos
   kontonr    Konto-Nr. des Vereins fuer die Beitragszahlungen
   iban       IBAN
   bic        SWIFT-BIC - wird voraussichtlich bis 2/2014 benoetigt
   identnr    Glaeubiger-Identifikationsnr fuer die SEPA-Lastschrift
   email      Mail-Adresse des Vereins
   version    Versions-Nr. des Projekts - wird fuer Anpassungen benoetigt
```

10.2.21 version

INFO:

```
   Versionsverwaltung - hier steht immer die aktuell installierte Version
```

```
CREATE TABLE IF NOT EXIStS version (
 version     INTEGER NOT NULL DEFAULT 0,
 subversion  INTEGER NOT NULL DEFAULT 0,
 release     INTEGER NOT NULL DEFAULT 0,
 revision    INTEGER NOT NULL DEFAULT 0,
 typ         TEXT    NOT NULL DEFAULT 'stable'
);
```

10.2.22 virtual_konto

INFO:

```
   Diese Tabelle enthaelt Informationen ueber vereinsintern benoetigte Werte
   und Zuordnungen. Sie wird von verschiedenen Programmteilen benoetigt um
   Werte korrekt zuordnen zu koennen.
```

```
CREATE TABLE IF NOT EXISTS virtual_konto (
 verein       VARCHAR(20) NOT NULL REFERENCES vereine(kurzname),
 art          VARCHAR(20) NOT NULL,
 konto        NUMERIC(10,0) NOT NULL,
 fkonto       NUMERIC(10,0) NOT NULL,
 kst          NUMERIC(10,0) NOT NULL
);
ALTER TABLE virtual_konto OWNER TO :datenbankuser;
```

BESCHREIBUNG:

```
   verein        betroffener Verein
   art           Schluessel ueber den zugegriffen wird. Je nach Schluessel
                 enhalten die Felder konto, fkonto und kst die gesuchten
                 Werte. Folgende Zuordnungen sind vorhanden:
                    Bedeutung des Schluessels / Inhalt
     art            Feld    Was steht im Feld?
     ------------------------------------------------------------------------
     BEITRAG        Zuordnung der Beitragszahlungen aus der Mitglieder-
                    verwaltung zu den Buchhaltungskonten
                    konto   Beitragskonto fuer die Buchhaltung
                    fkonto  Finanzkonto, das als Voreinstellung in der
                            Mitgliederverwaltung eingetragen wird
                    kst     Kostenstelle fuer die Beitragsbuchungen
     BEITRAG_1      1. Beitragserinnerung
```

```
                konto   Nummer des zu Briefes - siehe "briefe"
                fkonto  Hoehe der anfallenden Mahnkosten
    BEITRAG_2   2. Beitragserinnerung
                konto   Nummer des zu Briefes - siehe "briefe"
                fkonto  Hoehe der anfallenden Mahnkosten
    BEITRAG_3   Letzte Mahnung
                konto   Nummer des zu Briefes - siehe "briefe"
                fkonto  Hoehe der anfallenden Mahnkosten
    BEITRAG_LEV Brief nach einer Lastschrift-Retoure
                konto   Nummer des zu Briefes - siehe "briefe"
                fkonto  Hoehe der anfallenden Mahnkosten
    FINANZKONTO Welcher Kontenart gehoeren die Finanzkonten an?
                konto   Kontenart der Finanzkonten - siehe "bu_klassen"
    SCHWEBEKONTO Welches Konto wird als Schwebekonto verwandt?
                konto   Konto-Nr. - siehe "bu_konten"
    UEBERTRAG   Zuordnung des Uebertrags bei Jahreswechsel
                konto   Kontenart - siehe "bu_klassen"
                fkonto  Konto-Nr. - siehe "bu_konten"
                kst     Kostenstelle - siehe "bu_kst"
    VORSTEUER   Wie wird die in bu_belege.b_steuer eingetragene Vorsteuer
                in die Buchhaltung eingetragen?
                konto   Buchungskonto - siehe "bu_konten"
                fkonto  Finanzkonto - siehe "bu_konten"
                kst     Kostenstelle - siehe "bu_kst"
```

10.2.23 zahlungsweise

INFO:

```
   Diese Tabelle enthaelt Informationen ueber die definierten Zahlungsweisen.
   Auf diese Tabelle referenzieren die fuer die Beitragszahlung zustaendigen
   Tabellen.
```

```
CREATE TABLE IF NOT EXISTS zahlungsweise (
 nr       INTEGER NOT NULL UNIQUE,
 art      TEXT,
 einmalig BOOLEAN DEFAULT false NOT NULL,
 teiler   INTEGER DEFAULT 1 NOT NULL
);
```

BESCHREIBUNG:

```
   nr       Schluessel, auf den referenziert wird
   art      Bezeichnung im Klartext
   einmalig Einmalzahlung (ja/nein)
   teiler   Teiler fuer den Jahresbeitrag
```

BESTANDTEILE:

```
nr  art              einmalig teiler
-------------------------------------
 0  einmalig         ja            1
 1  jaehrlich        nein          1
 2  halbjaehrlich    nein          2
 3  monatlich        nein         12
 4  vierteljaehrlich nein          4
```

10.3 Makefile

NAME:

```
Makefile
um die Projekt-Dokumentation mit RoboDoc zu erstellen
```

BESCHREIBUNG:

```
Abschnitt: default
 1. Robodoc wird aufgerufen um die Projektdokumentation zu
    erstellen
 2. Das erzeugte Latex-Dokument wird bereinigt und nicht be-
    nötigte Texte entfernt
 3. Das PDF-Dokument wird erstellt
 4. Temporäre Dateien wieder löschen
 5. Statistik-Daten erstellen
 6. Hilfs-Script für die Update-Koordination ausführen
 7. Aufräumen
```

10.4 muster.sql

NAME:

```
sql/muster.sql
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
14.04.2009 komplett ueberarbeitet
07.06.2009 Jahresangabe korrigiert - nur aktuelles Jahr eintragen
```

BUGS:

```
Keine bekannt
```

10.4.1 muster

INFO:

```
   Diese Tabelle existiert fuer jeden Verein und enthaelt die Informationen
   ueber die dem Verein zugeordneten Mitglieder
```

```
CREATE TABLE muster (
  nr          INT8 DEFAULT 1 NOT NULL UNIQUE REFERENCES personen(nr),
  eintritt    DATE NOT NULL,
  ende        DATE,
  kommentar   TEXT,
  zahlungsart SMALLINT DEFAULT 1 NOT NULL REFERENCES key_zahlungsarten (nr),
  aktiv       BOOLEAN DEFAULT true
);
```

BESCHREIBUNG:

```
   nr          Mitglieds-Nr. (identisch mit Nr. in der Tabelle "personen")
   eintritt    Eintrittsdatum
   ende        Vereinsaustritt
   kommentar   Bemerkungen zum Mitglied
   zahlungsart Zahlungsart fuer den Mitgliedsbeitrag
   aktiv       Voreinstellung ist true - ausgeschiedene Mitglieder = false
               Die Spalte wird durch den Cronjob "bash/uverein.hour_cron"
               stuendlich aktualisiert.
```

10.4.2 muster_einzug

INFO:

```
   Die Tabelle verwaltet die Einzugsermaechtigungen und existiert fuer jeden
   Verein
```

```
CREATE TABLE muster_einzug (
  nr    INT8 DEFAULT 1 NOT NULL UNIQUE REFERENCES muster (nr),
  blz   NUMERIC (8,0) NOT NULL REFERENCES key_banken (blz),
  konto NUMERIC (10,0) NOT NULL
);
```

BESCHREIBUNG:

```
   nr    Mitglieds-Nr. (siehe muster)
   blz   Bankleitzahl
   konto Konto-Nr.
```

10.5 update-blz.de

NAME:

```
update-blz.de in dev-scripts/update/blz/

Das Script liest die aktuelle Bankleizahlenliste der Bundesbank
ein und erstellt die SQL-Dateien fuer das Projekt.
```

BUGS:

```
keine bekannt
```

10.6 update_0.2.1-17.sql

NAME:

```
sql/update_0.2.1-17.sql
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

INFO:

```
Version 0.2.1-17
Umsatzsteuerberechnung einfuegen
```

10.7 update_0.2.3-3.install

NAME:

```
sql/update_0.2.3-3.install

Das Script installiert die SCHEMATA
  testing
  prod
in die Datenbank

Durch die Einfuehrung des Datenbank-Schemas fallen die Datenbanken
  * udb.uverein.test
  * udb.version
weg.
```

BESCHREIBUNG:

```
Erstellt
  DATENBANK-SCHEMA   Verwendungszweck
  ------------------------------------------------------------------
  testing            Fuer die Programmentwicklung - Testmodus
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
08.08.2009 Schema 'testing' erstellt
```

BUGS:

```
keine bekannt
```

10.8 update_0.2.3-3.new.sql

NAME:

```
sql/update_0.2.3-3.new.sql
Neu-Installation
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
Installations-Script neu erstellt
```

BUGS:

```
Keine bekannt
```

10.8.1 CREATE SCHEMA

```
CREATE SCHEMA ####CREATE_SCHEMA#### AUTHORIZATION "####DATABASE_USER####";
SET SEARCH_PATH = '####CREATE_SCHEMA####';
```

10.9 update_0.2.3-3.update.sql

NAME:

```
sql/update_0.2.3-3.update.sql
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

BUGS:

```
Keine bekannt
```

10.10 update_0.2.3-8.install

NAME:

```
sql/update_0.2.3-8.install

Das Script bereinigt die Installation der Test-Datenbank
  * Datenbank udb.uverein.test wird geloescht
  * alte Backup-Dateien in log werden entfernt
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
08.08.2009 Schema 'testing' erstellt
20.06.2010 testing funktioniert, daher koennen die alten
           Dateien geloescht werden
```

BUGS:

```
keine bekannt
```

10.11 update_1232630993.sql

BESCHREIBUNG:

```
Jahr in bu_konten eintragen

Neue Definition nach der Aenderung:
TABLE bu_konten (
  verein         VARCHAR(20)   NOT NULL REFERENCES vereine (kurzname),
  kto_nr         NUMERIC(10,0) NOT NULL,
  kto_name       TEXT          NOT NULL,
  kto_klasse     NUMERIC(10,0) NOT NULL,
  kto_ziel       NUMERIC(10,0) NOT NULL,
  kto_stand      NUMERIC(25,2),
  jahr           VARCHAR(10)   NOT NULL DEFAULT EXTRACT(YEAR FROM CURRENT_DATE)
)
```

HISTORIE:

```
22.01.2009  0.2.3-1 erstellt
```

10.12 Update_Makefile

NAME:

```
dev-scripts/update/Makefile
```

BESCHREIBUNG:

```
Makefile um Update-Informationen zusammen zu stellen.
```

11 Mitgliederverwaltung

INFO:

```
Verwaltung der Vereinsmitglieder
- Beginn und Ende der Mitgliedschaft
- Beitragszahlung -> Beitrag
```

11.1 beitrag.php

NAME:

```
beitrag.php
```

11.1.1 Buchungskonten festlegen

DEFINITION:

```
AUSWAHL=BU_KONTEN_FESTLEGEN
Buchungskonten und Buchungsklasse fuer die Beitragsbuchungen festlegen
```

11.2 mitglied.print.php

BESCHREIBUNG:

```
Daten zu einem Mitglied als PDF ausgeben
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
11.03.2012 Start der Programmierung
```

BUGS:

```
keine bekannt
```

11.3 mitglieder.php

BESCHREIBUNG:

```
Verwaltung der Mitglieder
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

TODO:

```
Umstellung auf Klassendefinitionen
```

BUGS:

```
keine bekannt
```

11.4 mitglieder.transfer.php

NAME:

```
mitglieder.transfer.php
```

BESCHREIBUNG:

```
Feature-Request 3035877
Mitglieder als CSV-Datei einlesen und ausgeben
```

ERGEBNIS:

```
ohne Parameter:
  Die Mitgliedsdaten werden in eine tab-getrennte Datei geschrieben
  und die Datei zum Download angeboten.

IMPORT
```

11.5 print.beitrag-offen.php

NAME:

```
print.beitrag-offen.php
```

BESCHREIBUNG:

```
Erstellt eine Aufstellung der offenen Beitraege als PDF-Dokument
```

11.6 print.beitrag.php

BESCHREIBUNG:

```
Aufstellung der Beitragsbuchungen als PDF-Dokument
```

AUTOR:

```
Uwe Schied
```

COPYRIGHT:

```
GPL
```

HISTORIE:

```
25.07.2009 Datum wird nicht angezeigt -> Problem behoben
```

BUGS:

```
keine bekannt
```

12 php

INFO:

```
PHP-Dateien zum Projekt
```

12.1 i18n

INFO:

```
Sprachdateien
```

12.2 print.belege-sum.php

NAME:

```
print.belege-sum.php
Erzeugt die Belegaufstellung als PDF - pro Buchungs-Nr. wird eine Zeile angedruckt
```

HISTORIE:

```
09.11.2008  Datei erstellt
24.01.2012 Nur bei Umbuchungen wird SOLL und HABEN in einer Zeile eingedruckt,
           ansonsten wird SOLL und HABEN in einem Wert anzeigt
```

GENUTZT VON:

```
buchung.php  AUSWAHL=BELEGE
```

VOREINSTELLUNG:

```
Seitenlaenge (bedruckbarer Bereich) wird auf 250 mm eingestellt
```

FUNCTION:

```
zahl ($x)
Zahlendarstellung mit 2 Nachkommastellen
  $x = auszuwertende Zahl
```

FUNCTION:

```
neueseite ($art)
Zwischensumme beim Seitenwechsel
  $art = Art des Seitenwechsels
         Moegliche Optionen:
         * AUTO    = automatisch
```

13 project

NAME:

```
Hauptverzeichnis fuer den Projekt-Quellcode
```

13.1 debian

INFO:

```
Dateien fuer das Debian-Paketformat
```

13.2 doc

INFO:

```
Projekt-Dokumentation und Hilfen
```

13.2.1 doc/html

INFO:

```
Hilfe zum Projekt
```

13.3 HTML

INFO:

```
Style-Sheet
```

13.3.1 HTML/grafik

INFO:

```
Bilder und Icons zum Style-Sheet
```

13.4 include

INFO:

```
Include-Dateien fuer Webserver, Passwortverwaltung usw.
```

13.4.1 configure

NAME:

```
configure
```

ERGEBNIS:

```
makefile.apache
  * APACHE_CONF   Name und Pfad der Webserver-Konfiguration
  * DBUSER        User, unter dem der Webserver laeuft
  * APACHE_START  Pfad zum Webserver-Demon

Makefile.config
  Befehle fuer die Installation
```

HISTORIE:

```
2008-11-15 Sonderkonfiguration fuer SuSE
```

HISTORIE:

```
2009-02-21 Debian Lenny Konfiguration mit Apache2
```

13.5 log

INFO:

```
Log-Dateien und durch cron erzeugte Datensicherungen
```

14 Script

INFO:

```
ausfuehrbare Scripte und Programme
```

14.1 cron.sh

NAME:

```
cron.sh
Das Script sichert die Datenbanken im Verzeichnis /usr/share/uverein/log
```

COPYRIGHT:

```
(c) GPL by Uwe Schied
```

AUTOR:

```
Uwe Schied
```

ERGEBNIS:

```
Im Verzeichnis /usr/share/uverein/log wird eine Sicherungskopie
der Datenbanken
  * udb.uverein            produktive Datenbank
  * udb.version            Versionsverwaltung
angelegt
```

```
gzip -9 > /usr/share/uverein/log/$(date +%w)-${DB}.gz
```

14.2 uverein.txt-001

NAME:

```
uverein.txt-001
  Das Script liest die Datenbank aus und erstellt im Verzeichnis log
  die Datei personen.info1.tab
  Die erstellte Datei kann in anderen Anwendungen als Tab-getrennte
  Tabelle genutzt werden.
  In der ersten Zeile sind die Tabellen-Ueberschriften
```

BUGS:

```
keine bekannt
```

15 SEPA-Klasse

INFO:

```
Klasse fuer den Datenaustausch mit Banken im SEPA-Raum.
```

15.1 SEPA-Klasse/sepa-xml.inc

KLASSE:

```
cl_sepa
Klasse fuer den Datenaustausch mit Banken im SEPA-Raum
Es wird eine XML-Datei erstellt, die zur Bank hochgeladen werden
kann.
```

HISTORIE:

```
29.12.2013 Start
```

BUGS:

```
keine
```

15.1.1 absender (Name, IBAN, BIC, Gläubiger-ID)

SICHTBAR:

```
öffentlich sichtbar
```

BESCHREIBUNG:

```
Absenderangaben
Übergabe-Parameter:
  Name
  IBAN
  BIC
  IDENTNR    Gläubiger-Identifikation (in Deutschland von der Bundesbank ausgestellt)
```

15.1.2 ausgeben ()

SICHTBAR:

```
öffentlich sichtbar
```

BESCHREIBUNG:

```
XML-Datei zusamensetzen und als Zeichenkette ausgeben
Uebergabe-Paramter:
  keine
```

15.1.3 bereitstellen ()

SICHTBAR:

```
öffentlich sichtbar
```

BESCHREIBUNG:

```
Die Funktion stellt die XML-Datei bereit
```

15.1.4 cl_sepa (art, sequenz)

BESCHREIBUNG:

```
Konstruktor fuer die SEPA-Klasse
Uebergabe-Paramter:
  art        Art des SEPA-Auftrages
             * BASIS      = Basis-Lastschrift (Voreinstellung)
  sequenz    Sequenz für die Lastschrift -> wird nicht automatisch geprüft
             * FRST       = Erst-Lastschrift
             * RCUR       = Folge-Lastschrift (Voreinstellung)
             * OOFF       = Einmal-Lastschrift
             * FNAL       = Letzmalige Lastschrift
```

15.1.5 convert_min_ohne_leerzeichen (Zeile)

SICHTBAR:

```
Die Funktion ist nur innerhalb der Klasse aufrufbar
```

BESCHREIBUNG:

```
Konvertiert Zeile in Zeichenkette mit eingeschränktem Zeichensatz
ohne Leerzeichen
Der Zeichensatz besteht aus folgenden Zeichen:
  Numerische Zeichen 0 bis 9
  Buchstaben         A bis Z und a bis z
  Sonderzeichen      :?,-(+.)/
Übergabe-Parameter:
  Zeile
Rückgabe:
  konvertierte Zeichenkette
```

15.1.6 convert_mit_leerzeichen (Zeile)

SICHTBAR:

```
Die Funktion ist nur innerhalb der Klasse aufrufbar
```

BESCHREIBUNG:

```
Konvertiert Zeile in Zeichenkette mit eingeschränktem Zeichensatz
mit Leerzeichen
Der Zeichensatz besteht aus folgenden Zeichen:
  Numerische Zeichen 0 bis 9
  Buchstaben         A bis Z und a bis z
  Sonderzeichen      :?,-(+.)/
Übergabe-Parameter:
  Zeile
Rückgabe:
  konvertierte Zeichenkette
```

15.1.7 lastschrift (Mandant, Betrag, BIC, IBAN, Name, Verwendung)

SICHTBAR:

```
öffentlich sichtbar
```

BESCHREIBUNG:

```
Einzelner Datensatz für die Lastschrift
Übergabeparameter
  * Mandant = Array mit den SEPA-Mandanten-Daten
            ID        = Mandanten-ID
            DATE      = Unterschriftsdatum
  * Betrag = Euro-Betrag für die Abbuchung
  * BIC
  * IBAN
  * Name = Name des Zahlungspflichtigen
  * Verwendung = Verwendungszweck im Kontoauszug des Mitglieds
```

15.1.8 makexml ()

SICHTBAR:

```
Die Funktion ist nur innerhalb der Klasse aufrufbar
```

BESCHREIBUNG:

```
 Die Funktion erstellt die XML-Daten
```

16 special.inc

KLASSE:

```
cl_special
Diese Klasse stellt verschiedene Funktionen fuer den Seitenaubau, Seiten-
wechsel usw. zur Verfuegung
```

HISTORIE:

```
29.11.2007 Klasse erstellt
```

BUGS:

```
keine bekannt
```

16.1 change_frame (Seite, Frame)

FUNCTION:

```
change_frame ($site, $frame)
  $site    Seite, die geladen werden soll
  $frame   Frame, in den die Seite geladen werden soll

Laedt die gewuenschte Seite in den durch $frame definierten FRAME.
```

ERGEBNIS:

```
Gibt den Befehlscode fuer die Anzeige zurueck.
```

BEISPIEL:

```
$EXTRA = new cl_special ();
echo $EXTRA->change_frame ("http://uverein.sourceforge.net","Anzeige");
```

16.2 change_site (Neue Seite)

FUNCTION:

```
change_site ($site)
  $site   Seite, die geladen werden soll

Laedt die gewuenschte Seite in den aktuellen Frame.
```

ERGEBNIS:

```
Gibt den Befehlscode fuer die Anzeige zurueck.
```

BEISPIEL:

```
$EXTRA = new cl_special ();
echo $EXTRA->change_site ("http://uverein.sourceforge.net");
```

16.3 cl_special

FUNCTION:

```
cl_special ()
Konstruktor der Klasse. Es sind keine Parameter vorhanden.
```

BEISPIEL:

```
$SPEZIAL = new cl_special ();
```

16.4 cl_special ()

16.5 reload (Neue Seite)

FUNCTION:

```
reload ($site)
  $site   Seite, die geladen werden soll

Laedt die gewuenschte Seite und ersetzt damit die komplette aktuelle Seite
```

ERGEBNIS:

```
Gibt den Befehlscode fuer die Anzeige zurueck.
```

BEISPIEL:

```
$EXTRA = new cl_special ();
echo $EXTRA->reload ("http://uverein.sourceforge.net");
```

17 tabdatei.inc

KLASSE:

```
cl_tabdatei
  Diese Klasse verwaltet eine TAB-getrennte Textdatei als Tabelle
```

HISTORIE:

```
08.03.2011 Umstellung auf PHP5-Klassenstruktur
15.12.2007 Klasse erstellt
```

BUGS:

```
keine bekannt
```

17.1 cl_tabdatei

VARIABLE:

Name	privat	Inhalt
$F_ilename	ja	Dateiname incl. Pfad der TAB-Datei
$I_nhalt	ja	Array mit dem Inhalt der TAB-Datei
$A_nzahl	nein	Anzahl der Tabellen-Zeilen
$H_eader	ja	Kopfzeile ja/nein - > wenn ja, dann ist die erste Zeile die Kopfzeile
$H_inhalt	ja	Inhalt der Kopfzeile

17.1.1 cl_tabdatei (Pfad zur Datei, Kopfzeile)

NAME:

```
cl_tabdatei ($name,$kopfzeile)
  $name      = Pfad zur TAB-Datei
  $kopfzeile = Datei enthaelt eine Spaltenueberschrift?
               true   = Spaltenueberschrift vorhanden
               false  = keine Spaltenueberschrift

Konstruktor der Klasse
```

BESCHREIBUNG:

```
Konstruktor der Klasse
```

17.1.2 readfile ()

NAME:

```
readfile ()
```

SICHTBAR:

```
Nur innerhalb der Klasse
```

BESCHREIBUNG:

```
Oeffnet die Datei und liest die Daten in das interne Array $I_nhalt ein
Wenn die Datei nicht existert, wird eine neue Datei angelegt.
```

17.1.3 tab_by_nr (Zeilen-Nr., Ergebnisfeld)

NAME:

```
tab_by_nr ($nr, $ergebnisfeld)
Holt ein Feld aus der Datenbank
  $nr            Zeilen-Nr.
  $ergebnisfeld  Spalten-Nr.
```

SICHTBAR:

```
public
```

17.1.4 tab_get (Suchfeld, gesuchter Inhalt, Ergebnisfeld)

NAME:

```
tab_get ($suchfeld, $suche, $ergebnisfeld)
Holt ein Feld abhaengig vom Suchergebnis aus der Datenbank
  $suchfeld      Spalten-Nr. des gesuchten Feldes
  $suche         gesuchter Feldinhalt
  $ergebnisfeld  Spalten-Nr. des zu holenden Feldes

Es wird nur das letzte gefundene Feld zurueck gegeben
```

SICHTBAR:

```
public
```

17.1.5 tab_getmulti (Suchfeld, gesuchter Inhalt)

NAME:

```
tab_getmulti ($suchfeld, $suche)
Durchsucht die Datenbank
  $suchfeld      Spalten-Nr. des gesuchten Feldes
  $suche         gesuchter Feldinhalt

Es wird ein Array mit den Zeilen-Nr. der gefundenen Felder zurueck gegeben
Im Element 0 steht zusaetzlich die Anzahl der gefundenen Felder
```

SICHTBAR:

```
public
```

17.1.6 tab_header ()

NAME:

```
tab_header ()
Gibt die Kopfzeile als Array zurueck. Wenn es sich um ein Datei ohne
Kopfzeile handelt, wird NULL zurueckgegeben
```

SICHTBAR:

```
public
```

17.1.7 tab_save ()

NAME:

```
tab_save
Speichert die geaenderten Daten und liest die Datei anschliessend neu ein
```

SICHTBAR:

```
public
```

17.1.8 tab_set (Zeilen-Nr., Feld, Neuer Feldinhalt)

NAME:

```
tab_set ($nr, $feld, $feldinhalt)
Traegt ein Feld in die Datenbank ein
  $nr         Zeilen-Nr. beginnt mit 0 -> wenn groesser als $A_nzahl, dann
              erfolgt ein Neueintrag, ansonsten wird der betroffene Datensatz
              ueberschrieben
  $feld       Spalte, die eingetragen werden soll
  $feldinhalt Inhalt des Datenbankfeldes
```

SICHTBAR:

```
public
```

ERGEBNIS:

```
Wenn eine Kopfzeile in der Datei enthalten ist wird die durch $feld definierte
Spalte der Kopfzeile auf $feldinhalt gesetzt wenn $nr < 0 ist
Ansonsten wird in Spalte $feld und Zeile $nr der durch $feldinhalt angegebene
Wert eingetragen.
```

17.1.9 tab_set_by_search (Suchfeld, gesuchter Inhalt, Zielfeld, Feldinhalt)

NAME:

```
tab_set_by_search ($suchfeld, $suche, $zielfeld, $feldinhalt)
Traegt ein Feld abhaengig vom Suchergebnis ein.
  $suchfeld      Spalten-Nr. des gesuchten Feldes
  $suche         gesuchter Feldinhalt
  $zielfeld      Spalten-Nr. des einzutragenden Feldes
  $feldinhalt    einzutragender Feldinhalt

Wenn die Suche mehrere Treffer ergibt, wird das Feld in allen betroffenen
Feldern eingetragen. Wenn es keine Treffer gibt wird eine neue Zeile ein-
getragen.
```

SICHTBAR:

```
public
```

17.1.10 tab_sort ()

NAME:

```
tab_sort
  Sortiert die TAB-Datei nach dem Inhalt der 1. Spalte
```

SICHTBAR:

```
public
```

18 uvereinpdf.inc

BESCHREIBUNG:

```
Erstellt PDF-Dokumente fuer das Projekt uverein
```

ABHAENGIG VON:

```
php-fpdf      http://www.fpdf.org
```

HISTORIE:

```
13.04.2008 Tabellenanzeige eingefuegt
12.04.2008 Klasse angelegt
```

ABHAENGIG VON:

```
php-fpdf
```

18.1 anzeige ()

DEFINITION:

```
function anzeige ()
PDF-Dokument ausgeben
```

18.2 cl_uvereinpdf (Kopfzeile, Fusszeile)

DEFINITION:

```
function cl_uvereinpdf (Kopfzeile, Fusszeile)
Klassendefinition

Kopfzeile = Wert fuer die Kopfzeile aller Seiten
Fusszeile = Wert fuer die Fusszeile aller Seiten
```

18.3 table_head (Tabellenzeile)

FUNCTION:

```
function table_head (Tabellenzeile)
Die Funktion erstellt die Kopfzeile der Tabelle
Die Spalteninhalte werden als Array uebergeben.

Beispiel:
$PDF->table_head (array ("1. Spalte","2. Spalte", "", "4.Spalte");
```

18.4 table_init

FUNCTION:

```
function table_init (Spalten, Schrift, Schriftgroesse, Rahmenbreite,
                     Spaltenbreiten, Spaltenausrichtungen)
Die Funktion legt die Grundwerte fuer die PDF-Tabelle fest und be-
rechnet die Spaltenbreiten anhand der uebergebenen Parameter

Spalten              = Anzahl der Tabellen-Spalten
Schrift              = Schriftart (z.B. Arial)
Schriftgroesse       = Schriftgroesse in Punkt
Rahmenbreite         = Optionale Breite des Tabellen-Rahmens -
                       Vorgabe ist 0 -> kein Rahmen
Spaltenbreiten       = Optionales Array mit den relativen Spalten-
                       breiten - Vorgabe ist gleiche Spaltenbreiten
Spaltenausrichtungen = Optionales Arry mit den Spaltenausrichtungen
                       L  Links ausgerichtet = Voreinstellung
                       J  Blocksatz
                       C  zentriert
                       R  rechtsbuendig
```

18.5 table_line (Tabellenzeile)

FUNCTION:

```
function table_line (Tabellenzeile)
Die Funktion fuegt eine Zeile in die Tabelle ein
Die Spalteninhalte werden als Array uebergeben.

Beispiel:
$PDF->table_line (array ("1. Spalte","2. Spalte", "", "4.Spalte");
```

18.6 textzeile (Text, Font, Fontsize, Ausrichtung)

FUNCTION:

```
function textzeile (Text, Font, Fontsize, Ausrichtung)
Die Funktion fuegt Text in das PDF-Dokument ein.

Text        = Text, der als PDF eingefuegt wird
Font        = Optionale Schriftart -> Voreinstellung: Arial
Fontsize    = Optionale Schrfitgroesse -> Voreinstellung: 10
Ausrichtung = Optionale Ausrichtung des Textes. Moegliche Werte:
              L  Links ausgerichtet = Voreinstellung
              J  Blocksatz
              C  zentriert
              R  rechtsbuendig
```

19 Vereine verwalten

INFO:

```
Verwaltung der einzelnen Vereine
```

19.1 verein-basis.php

BESCHREIBUNG:

```
Vereinsdaten verwalten (SEPA-Version)
```

BUGS:

```
keine bekannt
```

COPYRIGHT:

```
GPL
```

AUTOR:

```
Uwe Schied
```

BESCHREIBUNG:

```
Der Aufruf erfolgt über das Vereins-Menü - folgende Parameter
sind definiert:
* NEU              Neuanlage eines Vereins
* NEU_ERFASSEN     Daten für die Neuanlage in Datenbank eintragen
```

HISTORIE:

```
25.12.2013 Start - Neuen Verein erfassen
           Neuanlage fertig gestellt
```

19.2 verein.beitrag.php

BESCHREIBUNG:

```
Beitragsarten des Vereins verwalten
```

BUGS:

```
keine bekannt
```

COPYRIGHT:

GPL

AUTOR:

Uwe Schied

19.3 verein.change.php

BESCHREIBUNG:

Zu einem anderen Verein wechseln

BUGS:

keine bekannt

COPYRIGHT:

GPL

AUTOR:

Uwe Schied

19.4 verein.delete.php

BESCHREIBUNG:

Loescht den Verein aus der Datenbank

BUGS:

keine bekannt

COPYRIGHT:

GPL

AUTOR:

Uwe Schied