

Versionsgeschichte zum Report-Explorer von SchILD

Integration in SchILD

Ab der Version 2.0.8 von SchILD ist die Reportverwaltung direkt in SchILD integriert, d.h. es wird kein separates Plugin.(Reporting.dll) mehr benötigt. Die Versionsnummer ist daher ab dieser Version identisch mit der von SchILD.

Version 2.0.10.1

Neue Prozeduren und Funktionen

Prozedur AlleFaecherLaden(Laden: Boolean):

Mit der Prozedur "AlleFaecherLaden(Laden: Boolean)" kann erreicht werden, dass in einem Report auch Fächer ohne Noten geladen werden. Dies betrifft insbesondere solche Datenquellen, die ursprünglich in erster Linie für die Zeugnisausgabe entwickelt wurden, in letzter Zeit aber auch häufiger z.B. für Konferenzlisten genutzt werden. Über den Aufruf "AlleFaecherladen(true);" (z.B. im Ereignis "ReportBeforePrint") kann erreicht werden, dass für den betreffenden Report die globale Einstellung "Auch Leistungsdaten ohne Noten laden" geändert wird (ohne dass dies auf einen späteren Zeugnisdruck Auswirkungen hat).

Version 2.0.9.1

Neue Datenquellen

Betriebe zur Schuelerauswahl

Liefert die Betriebe (in alphabet. Reihenfolge), die bei den Schülern der aktuellen Auswahl jeweils an erster Stelle bei "Allg. Adressen" stehen (und damit im Verständnis von SchILD die Betriebe sind, an denen die Schüler derzeit ausgebildet werden). Diese Datenquelle muss in einer Serienbriefvorlage die Hauptdatenquelle sein.

Schueler zu aktuellem Betrieb

Liefert die Schüler (in alphabet. Reihenfolg, zzgl. Angaben zu Vertragsbeginnsw.), die dem jeweiligen Betrieb aus "Betriebe_zur_Schuelerauswahl" zugeordnet sind.

Auch hier wird die aktuelle Schülerauswahl berücksichtigt.

Die Datenquelle Schueler_zu_aktuellem_Betrieb muss als Subreport zu Betriebe_zur_Schuelerauswahl verwendet werden.

Beispiel:

Es wird in SchILD auf die Klasse AM05A gefiltert. Dann liefert *Betriebe_zur_Schuelerauswahl* nur solche Betriebe, die bei den Schülern aus AM05A als erster in der Reihenfolge eingetragen sind.

Schueler_zu_aktuellem_Betrieb liefert dann nur die Schüler, die die AM05A besuchen und aktuell bei dem Betrieb ausgebildet werden.

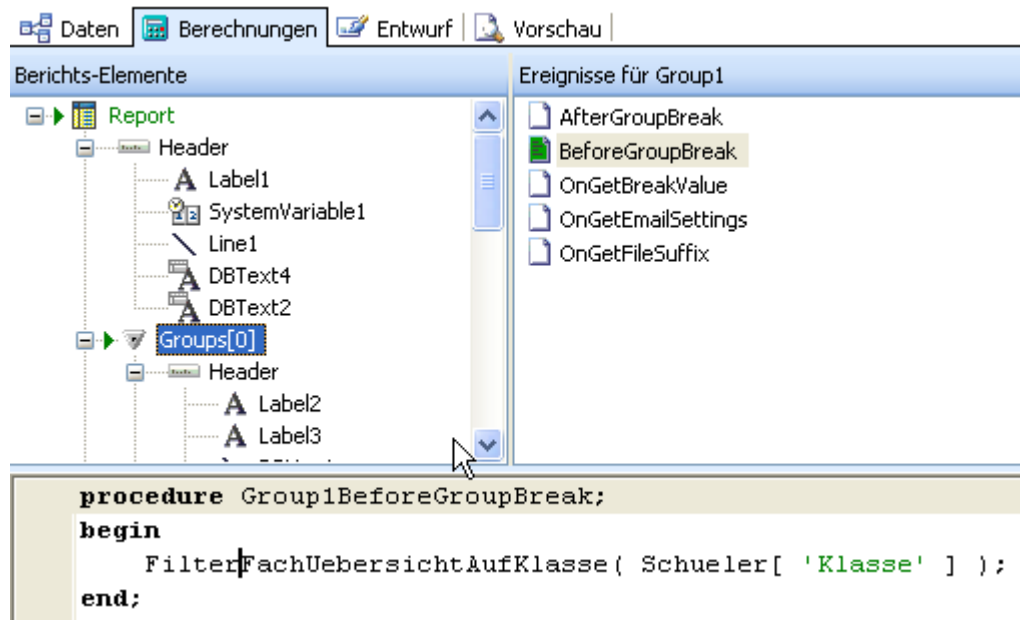
Wenn also Max Mustermann (Klasse AM05A) und Stefan Sonderling (Klasse AM05B) beide bei "Schmitz und Sohn" ausgebildet werden, wird "Schmitz und Sohn" bei Filterung in SchILD auf AM05A in *Betriebe_zur_Schuelerauswahl* erscheinen. Max Mustermann ist dann in *Schueler_zu_aktuellem_Betrieb* vorhanden, nicht aber Stefan Sonderling (weil in AM05B).

Version 2.0.8.5

Neue Prozeduren und Funktionen

Procedure FilterFachUebersichtAufKlasse(const klasse: string);

Mit dieser Prozedur kann bei einem Gruppenwechsel (wobei die Gruppierung auf "Klasse" sein muss) eine Fachübersicht neu initialisiert werden. Der Aufruf erfolgt typischerweise im Ereignis "BeforeGroupBreak" (siehe Screenshot), als Aufruf-Parameter wird die Klasse übergeben. Dadurch ist es möglich, Fachübersichten für mehrere Klassen (oder die ganze Schule) in einem Durchlauf auszugeben, wobei nur die jeweils von der betreffenden Klasse belegten Fächer in der Übersicht erscheinen.



```
procedure Group1BeforeGroupBreak;
begin
  FilterFachUebersichtAufKlasse( Schueler[ 'Klasse' ] );
end;
```

Wichtig: Der Aufruf sollte nur in Verbindung mit der Datenquelle "Fachuebersicht" erfolgen. Ein Aufruf von "FaecherVorbelegen" ist nicht notwendig (wird intern von "FilterFachUebersichtAufKlasse" gemacht) und wäre auch kontraproduktiv (erhöhter Zeitaufwand wegen doppeltem Aufruf).

Datenquelle "FachUebersicht"

Auch die Angaben zur Fächergruppe werden jetzt aus den gliederungs-bezogenen Einstellungen (sofern vorhanden) übernommen.

Verzeichnisse „Header_Footer“ und „Subreports“

In beiden Verzeichnissen können jetzt auch Unterverzeichnisse angelegt und genutzt werden. Werden in einem Bericht dynamisch nachladbare Subreports aus solchen Unterverzeichnissen verwendet, so wird der relative Pfad zum jeweiligen Hauptverzeichnis (Header_Footer bzw. Subreports) gespeichert.

Version 2.0.8.2

Neue Felder

Datenquelle Schueler

In den Datenquellen "Schueler" gibt es nun ein Feld "BriefAdresse", das die vollständige Adresse (mit Namen und Zeilenumbrüchen) enthält.

Wichtig. Das Feld muss in einem DBMemo ausgegeben werden.

Datenquelle Erzieher

In den Datenquellen "Erzieher" gibt es nun ein Feld "BriefAdresse", das die vollständige Adresse (mit Namen und Zeilenumbrüchen) enthält.

Außerdem gibt es nun ein Feld "FormaleAnrede2", das die Anrede für einen Brief ("Sehr geehrte Frau...") mit einem Zeilenumbruch zwischen zwei Namen enthält.

Wichtig. Beide Felder müssen als DBMemo ausgegeben werden.

Datenquelle GSDaten

In der Datenquelle "GSDaten" wurde nun auch die Staatsangehörigkeit aufgenommen.

Datenquelle FHRFaecher

In der Datenquelle "FHRFaecher" steht jetzt auch das "Aufgabenfeld" zur Verfügung.

Datenquelle Lernabschnitte

In der Datenquelle "Lernabschnitte" stehen jetzt auch Angaben zum Abteilungsleiter zur Verfügung.

Geänderte Prozeduren und Funktionen

Prozedur "SetzeFilter"

Die Prozedur "SetzeFilter" kann jetzt auf zwei unterschiedliche Arten verwendet werden:

- Falls die Filterbedingung ein Hochkomma oder den Befehl "QuotedStr" enthält (typischerweise bei SQL-Befehlen) arbeitet die Prozedur wie bisher. In der Filterbedingung können aber nur "echte" Felder (d.h. die in der betreffenden Tabelle auch existieren) verwendet werden
- Daneben kann die Filterbedingung nun aber auch mit einer einfacheren Syntax verwendet werden. Dabei können auch berechnete oder "Lookup"-Felder verwendet werden.

Beispiele:

SetzeFilter(Schueler, 'Klasse=05*'):: Es werden nur Schüler berücksichtigt, deren Klasse mit "05" beginnt

SetzeFilter(Schueler, 'Klasse=*A'):: Es werden nur Schüler berücksichtigt, deren Klasse mit "A" endet

SetzeFilter(Schueler, 'Klasse=05A,06B,07C'):: Es werden nur Schüler berücksichtigt, die in den Klassen "05A", "06B" oder "07C" sind

Vorteil: Einfachere und flexiblere Verwendung,
Nachteil: Langsamer als bisheriges Verfahren

Separates Plugin (Reporting.dll)

Version 2.1.4.6

Neue Datenquelle

Datenquelle SchuelerGSDaten

Stellt die Grundschuldaten (z.B. Einzelnoten und Durchschnittsnoten) zur Verfügung

Neue Felder

Datenquelle Schueler

ZieldifferentesLernen

Neue Prozeduren und Funktionen

Procedure LocateRecord(Pipeline: TppDbPipeline; const Feld, Wert: String);'

Mit dieser Prozedur kann in einer Datenquelle der erste Datensatz, der die Bedingung „Feld=Wert“ erfüllt, gefunden werden.

Beispiel

```
LocateRecord( Lehrer, 'Kuerzel', SchuelerLeistungen[ ,FachLehrer' ] )
```

function RecordCount(Pipeline: TppDbPipeline):integer;'

Liefert die Anzahl der Datensätze in einer Datenquelle.

Beispiel: IblAnzahl.Caption := IntToStr(RecordCount(Schueler));

Geänderte Prozeduren und Funktionen

Die Verwendung der Prozedur „SetzeFilter“ wurde vereinfacht. Bisher war es nur möglich, auf "echte" Datenbankfelder zu filtern, nun können aber auch berechnete oder „Lookup“-Felder verwendet werden.

Soll auf unterschiedliche Feldinhalte gefiltert werden, so brauchen die entsprechenden Werte jetzt nur hoch hintereinander (getrennt durch Semikolon) angegeben zu werden (bisher waren dazu mehrere „OR“-Konstrukte notwendig).

Beispiel:

```
SetzeFilter( Erzieherfunktion, 'Funktion=Klassenpflegschaftsvorsitzende/r;stv.  
Klassenpflegschaftsvorsitzende/r' );
```

Soll auf „nicht leer“ gefiltert werden, so ist die wie folgt einzugeben:

```
SetzeFilter(Schueler,'Fahrschuelerart<>''');
```

Wichtig: Im Designmodus funktioniert SetzeFilter zur Zeit noch nicht immer korrekt, im normalen Ausgabemodus aber wohl.

Version 2.1.4.2

Allgemeines

- Neues Layout der Schaltersymbole
- Im Report-Explorer werden jetzt auch RTF- und PDF-Dateien angezeigt. Bei Doppelklick auf eine solche Datei öffnet sich WordPad oder der jeweils vorhandene PDF-Viewer

Neue Felder

Datenquelle FachUebersichtHeader

FachLehrer: Kürzel des jeweiligen Fachlehrers, falls in der jeweiligen Auswahl mehrere Fachlehrer für ein Fach vorkommen, wird eine durch Komma getrennte Liste der Kürzel ausgegeben. Das Feld kann für Überschriften einer Fächertabelle verwendet werden.

Neue Prozeduren und Funktionen

Function ThemaProjektkurs: string;

Diese Funktion liefert das Thema des Projektkurses (sofern vorhanden) des jeweiligen Schülers.

Function EditPlainText: string;

Mit dieser Funktion kann ein einfacher Editor zur Eingabe von Text aufgerufen werden. Es wird der eingegebene Text zurückgeliefert.

Version 2.1.3.1

Neue Felder

Datenquellen „Lernabschnitte“

Zeugnisart: Kürzel für die Zeugnisart

ZeugnisartText: Klartext für die Zeugnisart

Folgende Einträge stehen zur Verfügung:

Halbjahrszeugnis (HJ)

Jahrszeugnis - Versetzungsvermerk (JV)

Jahrszeugnis - Wiederholer' (JNV)

Jahrszeugnis - Anforderungen erfüllt (JAE), nur bei BK

Jahrszeugnis - Anforderungen nicht erfüllt (JANE), nur bei BK

Jahrszeugnis - freiwilliger Wiederholer (JFW)

Jahrszeugnis - Überspringer (JÜ)

Jahrszeugnis - Nachprüfung (JNP)

Abgangszeugnis - ohne Abschluss (AZO)

Abgangszeugnis - Versetzungsvermerk (AZV);

Abschlusszeugnis (AB)

Überweisungszeugnis (ÜZ), nicht bei BK

Datenquellen „Abteilungen“, „SchuelerKlassen“

Raum, Durchwahl und E-Mail

Version 2.1.2.1

Neue Funktionalitäten

Report-Designer

Es steht jetzt eine Zoom-Funktion für den Arbeitsbereich zur Verfügung (Schieberegler unten in der Fußzeile)

Neue Felder

Datenquellen „Schueler“, „HalbjahrUebersicht“, „Lernabschnitte“, „SchuelerLaufbahn“

In diesen Datenquellen gibt es nun das Feld „SekStufe“, das die jeweilige Sekundarstufe (ergänzt um das Halbjahr) ausgibt, z.B. in der Form
SII-1_1 (für "erster Jahrgang der Sek.II, 1. Halbjahr")
SII-3_2 (für "dritter Jahrgang der Sek.II, 2. Halbjahr") usw.

Je nach Schulform verbirgt sich dahinter z.B. EF.1 oder Q2.2 oder 11.1 bzw. 13.2 usw. Das gilt insbesondere auch für WBK's, dort gibt es nämlich das spezielle Problem, dass SII-3_2 aus Schülersicht nicht unbedingt im 2.Schulhalbjahr liegen muss. Wenn nämlich der Schulbesuch zum Sommersemester (2. Schulhalbjahr) begonnen wurde, liegt das 2. Halbjahr (aus Schülersicht) im 1. Schulhalbjahr und umgekehrt.

Das Feld kann dazu benutzt werden, um z.B. Übersichten der gymn. Oberstufe unabhängig von der Schulform entwickeln zu können.

Version 2.1.1.4

Neue Funktionalitäten

Serienmails

Reports, die auf die Datenquellen „Schueler“ oder „Erzieher“ zugreifen und eine Gruppierung auf die folgenden Felder haben, können nun als Serienmails verschickt werden:

- Datenquelle „Schueler“: Felder ID, LSSchulnr (d.h. zuletzt besuchte Schule), SchulwechselNr (d.h. aufnehmende Schule)
- Datenquelle „Erzieher“: Feld ID

Die entsprechende E-Mail-Adresse wird automatisch anhand der Gruppierung ermittelt:

- Datenquelle „Schueler“, Gruppierungsfeld „ID“: E-Mail-Adresse des Schülers
- Datenquelle „Schueler“, Gruppierungsfeld „LSSchulnr“: E-Mail-Adresse der zuletzt besuchten Schule
- Datenquelle „Schueler“, Gruppierungsfeld „SchulwechselNr“: E-Mail-Adresse der aufnehmenden Schule
- Datenquelle „Erzieher“, Gruppierungsfeld „ID“: E-Mail-Adresse der Erziehungsberechtigten

Wenn diese Voraussetzungen gegeben sind, steht bei der Ausgabe die zusätzliche Option „als Serien-E-Mail versenden“ zur Verfügung.

Wichtig: Eine spezielle Anpassung für den Serienmail-Versand in der Reportvorlage ist nicht mehr notwendig.

Neue Felder

Datenquelle „Schueler“

EntlassjahrgangIntern: Der Entlassjahrgang als „interne Bezeichnung“ (im Gegensatz zum Feld „Entlassjahrgang“, in dem der Entlassjahrgang als Statistik-Bezeichnung ausgegeben wird)

LSEMail: E-Mail-Adresse der zuletzt besuchten Schule

ZSEMail: Email-Adresse der zukünftigen (d.h. aufnehmenden Schule)

Berufsqualifikation: Qualifikation, die ein Schüler bei Eintritt in die Schule mitbringt (für BK und WBK)

Datenquelle „BKAbschlussFaecher“

Lernentw: Fachbezogene Bemerkung (wird aus den jeweils aktuellen Leistungsdaten ermittelt)

VorherAbgeschl: Angabe zum Epochalunterricht (wird aus den jeweils aktuellen Leistungsdaten ermittelt)

AbschlussJahrgang: Gibt den Jahrgang an, in dem eine Fach, das in einem vorherigen Abschnitt abgeschlossen wurde, zuletzt unterrichtet wurde

Hochrechnung: Falls negativ, handelt es sich um ein Fach, das in einem vorherigen Halbjahr zuletzt unterrichtet wurde („abgeschlossenes Fach“)

GewichtungAB: „Allgemein-bildende“ Gewichtung des Faches, dabei werden auch evtl. vorhandene Gliederungs- oder Fachklassen-bezogene Einstellungen berücksichtigt

GewichtungBB: „Berufsbezogene“ Gewichtung des Faches, dabei werden auch evtl. vorhandene Gliederungs- oder Fachklassen-bezogene Einstellungen berücksichtigt

Neue Prozeduren und Funktionen

Function StatistikKuerzelAusFachKuerzel (const fach_krz: string): string:

Ermittelt das Statistik-Kürzel eines Faches aus dem internen Fachkürzel. Damit kann z.B. ermittelt werden, um was für ein Fach es sich im Einzelfall handelt (unabhängig von der jeweils internen Bezeichnung). Dadurch können Reports, die bestimmte Fächer auswerten, unabhängig von der internen Bezeichnung gemacht werden.

Parameter:

fach_krz: Das interne Fachkürzel

Beispiel:

```
If StatistikKuerzelAusFachKuerzel(SchuelerLeistungen[ 'FachKrz' ]) = 'D'  
then  
    ShowMessage( 'Das Fach ist DEUTSCH' );
```

Version 2.1.1.3

Geschwindigkeitsverhalten

Es wurden erhebliche Anstrengungen unternommen, das Geschwindigkeitsverhalten bei der Erstellung von Reports zu optimieren. Unter anderem wird nun beim Laden einer Reportvorlage analysiert, welche "Datenquellen" der betreffende Report verwendet. Als Kriterium dient dabei die Verknüpfung eines "datensensitiven" Elementes (z.B. DBText, Subreport) an eine Datenquelle (z.B. Schueler, Erzieher usw.). Nur Datenquellen, die mit einem solchen Element verknüpft sind, werden tatsächlich geöffnet, dies führt insbesondere bei Leistungsübersichten zu einer erheblich schnelleren Ausgabe der Reports.

Dies bedingt aber auch, dass Datenquellen, die nicht an datensensitive Elemente gebunden sind, nicht automatisch geöffnet werden. Wenn z.B. auf Erzieherdaten nur im Rahmen einer Programmierung zugegriffen wird (um z.B. ein einfaches Label-Element zu befüllen), wird die Erzieher-Datenquelle nicht geöffnet und der Text des betreffenden Label-Elementes bleibt leer. In einem solchen Fall gibt es mehrere einfache Möglichkeiten, den Report so umzugestalten, dass die betreffenden Datenquellen dennoch geöffnet werden:

- Verwenden Sie ein unsichtbares DBText-Element, das mit der betreffenden Datenquelle verknüpft wird
- Verwenden Sie (z.B. im BeforePrint-Ereignis des Reportes) die Prozedur "PipelineOeffnen" (s.u.) um die betreffende Datenquelle explizit zu öffnen
Beispiel:

```
procedure ReportBeforePrint;  
begin  
    PipelineOeffnen( Erzieher );  
end;
```

Neue Funktionalitäten

Zeugnisparameter

Die „Zeugnisparameter“ (Elternsprechtag, Wiederbeginn des Unterrichtes usw.) werden jetzt „global“ und halbjahresbezogen gespeichert. Das erleichtert den nachträglichen Zeugnisdruck aus einem vergangenen Halbjahr.

Neue Felder

Datenquelle „Schueler“

InternJahrgang: Der aktuelle Jahrgang als schulinterne Bezeichnung

EntlassjahrgangIntern: Der Entlassjahrgang als schulinterne Bezeichnung

AnmeldeDatum: Datum der Anmeldung

DSN_Text: Durchschnittsnote als Text (z.B. „befriedigend“)

Dokumentenverzeichnis: Das schülerspezifische Dokumentenverzeichnis

Datenquelle Leistungsübersicht

Wochenstunden eines Faches

Neue Prozeduren und Funktionen

procedure PipelineOeffnen(aPipeline: TppDBpipeline);

Öffnet die als Parameter angegebene Datenquelle

Beispiel: siehe oben

procedure DatensatzZeigerBewegen (aPipeline: TppDatapipeline; aSchritte: integer)

Bewegt den Datensatzzeiger der als Parameter angegebenen Datenquelle

Parameter:

aPipeline: Die betreffende Datenquelle

aSchritte: Bewegungsschritte (-1000: Erster Datensatz, 1000: Letzter Datensatz,

-1: vorheriger Datensatz, 1: Nächster Datensatz)

Beispiel:

```
DatensatzZeigerBewegen( Lernabschnitte, -1 );
```

function PluginVersion: string;

Liefert die Version der Reporting.dll

Parameter:

Keine

Beispiel:

```
procedure ReportBeforePrint;  
var  
    vers: string;  
begin  
    vers := PluginVersion;  
    if vers >= '2' then  
        begin  
  
        end;  
end;
```

function AnzahlElemente(const text, trennzeichen: string): integer;

Liefert die Anzahl der durch das Trennzeichen getrennten Einzelelemente in "text"

Parameter:

text: der zu analysierende Text

trennzeichen: Das Trennzeichen

Beispiel:

```
txt := '1,2,3,4':  
if AnzahlElemente( txt, ',' ) > 2 then
```

function EinzelElement(const text, trennzeichen: string; const position: integer):

string

Liefert das Einzelelement an Position "position" aus dem durch das Trennzeichen getrennten Text "text"

Parameter

text: der zu analysierende Text

trennzeichen: Das Trennzeichen

position: Die Position des Einzelelementes im Text (beginnend bei 1)

Beispiel:

```
txt := '1,2,3,4':  
if einzel_element( txt, ',', 2 ) = '2' then
```

procedure SetQuartalsnotenInUebersicht(MitQuartalsnoten: Boolean);

Steuert, ob in Leistungsübersichten auch Quartalsnoten erscheinen sollen

Beispiel:

```
procedure ReportBeforePrint;  
begin  
{Darstellung von Quartalsnoten einschalten}  
  SetQuartalsnotenInUebersicht( true );  
End;
```

procedure FilterLaufbahnStufen(const Stufenliste: String);

Filtert für Leistungsübersichten auf bestimmte Jahrgangsstufen. Damit ist es möglich, z.B. Formulare für die gymn. Oberstufe schulformunabhängig zu entwickeln

Parameter:

Stufenliste: Die zu betrachtenden Jahrgangsstufen, getrennt durch Komma

Beispiel:

```
procedure ReportBeforePrint;
var
  vers: string;
begin
  {Filter auf Q1 und Q2 bzw. 12 und 13 bzw. 5 und 6 (bei WBK)}
  FilterLaufbahnStufen('SII-2,SII-3');
end;
```

function PraktikumsTage(const nur betriebspraktikum: boolean): integer;

Liefert die Anzahl der Praktikumstage eines Schülers (optional nur solche mit Kennung „Betriebspraktikum“). Voraussetzung ist, dass das Praktikum bei dem jeweiligen Betrieb als solches gekennzeichnet ist (mit Anfangs- und Enddatum)

Beispiel:

```
procedure DetailBeforePrint;
Begin
  {Alle Praktikumstage ausgeben}
  LblPraktikumstage.Text := IntToStr( PraktikumsTage( false ) );
End;
```

function LernfelderZuFachklasse(fachklasse_id: integer): string;

Gibt die Lernfelder einer Fachklasse aus

Parameter:

fachklasse_id: Die Fachklasse_ID des Schülers (aus Stammdaten oder Lernabschnittsdaten)

Beispiel:

```
procedure DetailBeforePrint;
begin
  RichText_LF.RichText := LernfelderZuFachklasse( Schueler[ 'Fachklasse_ID'
] );
End;
```

procedure ZeugnisDatenquellenOeffnen;

Öffnet alle in Zeugnissen benötigten Datenquellen. In Zeugnissen, in denen die einzelnen Fächerbereiche (z.B. Deutsch, Mathematik, Fremdsprachen usw.) über dynamisch nachladbare Subreports realisiert sind, scheitert die automatische Erkennung, welche Datenquellen zu öffnen sind, da die betreffenden Subreports erst geladen werden, nachdem die Analyse erfolgt ist. Der einmalige Aufruf dieser Prozedur erspart dann den mehrfachen Aufruf von „PipelineOeffnen“ für jede einzelne Datenquelle.

Beispiel:

```
procedure ReportBeforePrint;  
begin  
    ZeugnisDatenquellenOeffnen;  
end;
```

procedure DateinameFuerDVAuswaehlen;

Der Aufruf dieser Prozedur (typischerweise im Ereignis ReportBeforePrint) bewirkt, dass vor der Ausgabe eines Reportes als PDF-Datei in der Dokumentenverwaltung ein individueller Dateiname eingegeben werden kann. Dieser Name wird dann bei der Ermittlung des „schülerspezifischen“ Dateinamens anstelle des Namens der Reportvorlage verwendet. Dies ist besonders bei Serienbriefvorlagen hilfreich, bei denen der Briefinhalt zur Laufzeit aus einer RTF-Datei geladen wird.

Beispiel.

Die Reportvorlage heißt „RTFBrief.rtm“. Normalerweise würde dann die in der Dokumentenverwaltung des Schülers abgelegte Datei z.B. den Namen „RTFBrief_2013_05_24.pdf“ erhalten, d.h. aus dem Dateinamen geht die Bedeutung des Schreibens nicht hervor. Durch Aufruf von „DateinameFuerDVAuswaehlen“ kann aber ein Dialog aufgerufen werden, in dem ein individueller Name definiert werden kann (z.B. „Hausaufgabenmahnbrief“), woraus sich dann der Dateinamen „Hausaufgabenmahnbrief_2013_05_24.pdf“ ergibt.

Beispiel:

```
procedure ReportBeforePrint;  
begin  
    DateinameFuerDVAuswaehlen;  
end;
```

Hinweis: Wenn der RTF-Text aus einer externen Datei geladen wird, so wird deren Dateiname verwendet, eine manuelle Eingabe erfolgt dann nicht.