Main

**const** cDynSourceIDDL='05DOXDMCSZDL3FUD00C5OG4NF4'; *// dyn DL*

cReportIDDL='4000000001'; *// ID tisk sestavy DL*

cDynSourceIDII='40SBPEINEFD13ACM03KIU0CLP4'; *// dyn fakt. vyd*

cReportIDII='1300000101'; *// ID tisk sestavy fakt. vyd.*

cDynSourceIDPRV='0PDOXDMCSZDL3FUD00C5OG4NF4'; *// dyn PRV*

cReportIDPRV='2P20000101'; *// ID tisk sestavy PRV*

cURL='http://s14.mailstep.cz/api/mailstage-test/';

cEnd='.json';

cUser='user';

cPassword='password';

cReceiptCardDocQueue='I000000101';

cStoreID='1000000101';

cFirmID='2JF0000101';

cDivisionID='1000000101';

cCodeSeparator='';

cPOZPPDFTempDir = '%TEMP%';

cPDFtkFile = 'pdftk.exe';

cPDFtkPath = 'C:\Program Files (x86)\PDFtk\bin';

cLogin='.json?login='+cUser+'&pswd='+cPassword;

**function** printTempPdf(aOS:TNxCustomObjectSpace; aID:**String**;aDynsourceID,aReportID:**String**):**String**;

**var** mIDs:TStringList;

**begin**

Result:='';

**if** **not** NxIsEmptyOID(aID) **then**

**begin**

mIDs:=TStringList.Create;

mIDs.Add(aID);

**try**

Result:= CFxReportManager.PrintByIDsToTempFile(NxCreateContext(aOS),mIDs,aDynSourceID,aReportID,pekPDF);

**finally**

mIDs.Free;

**end**;

**end**;

**end**;

**function** AttachBlobBase64(AOS: TNxCustomObjectSpace;AFileName:**String**;):**String**;

**var**

mFS: TFileStream;

mContent: TNxCustomBusinessObject;

mMon: TNxBusinessMoniker;

**begin**

result:='';

*//OutputDebugMessage('AttachFile - FileName: ' + AFileName);*

**if** FileExists(AFileName) **then** **begin**

mFS := TFileStream.Create(AFileName, fmOpenRead);

mContent := AOS.CreateObject('NUDTXZFOQUJOJJ0EVSTTGN4NPS');

**try**

mContent.New;

mContent.Prefill;

TNxCustomBlob(mContent).BlobData.CopyFrom(mFS, mFS.Size);

result:=EncodeBase64(mContent.GetFieldValueAsBytes('BlobData'));

**finally**

mContent.Free;

mFS.Free;

**end**;

**end**

**else**

OutputDebugString('CHYBA - soubor neexistuje');

**end**;

*//---------------//*

*{HTTPS KOMUNIKACE}*

**function** GetHTTP(**var** WinHttpRequest: Variant): Boolean;

**begin**

**try**

**if** **not** VarIsType(WinHttpRequest, varDispatch) **then**

WinHttpRequest := CreateOleObject('WinHttp.WinHttpRequest.5.1');

Result := True;

**except**

WinHttpRequest := **nil**;

Result := False;

RaiseException(ExceptionMessage);

**end**;

**end**;

**function** WSPost(**var** aLogStr,

AURL: **string**;

aToken : **String**;

**var** aJSONString:**String**;

**var** aResponseText: **string**; aConditions:**String**='';): Boolean;

**var**

aJSONRequestBody:TJSONSuperObject;

mMessage, mStr, mFunctionName: **string**;

mWinHTTP: Variant;

mStatusText: **string**;

mBytes: TBytes;

aText:**String**;

**begin**

**try**

Result := true;

**if** GetHTTP(mWinHTTP) **then** **begin**

aJSONRequestBody:=TJSONSuperObject.Create;

**try**

**if** aConditions='GET'**then** mWinHTTP.Open('GET', AURL)

**else**

mWinHTTP.Open('POST', CfxInternet.PathEncode(AURL));

*// mWinHTTP.SetRequestHeader('Content-Type', 'application/json; charset=utf-8');*

mWinHTTP.SetRequestHeader('Content-Type', 'application/x-www-form-urlencoded'); *// for POST data*

*// mWinHTTP.SetRequestHeader('Authorization', ' Token '+ aToken);*

**if** aConditions='GET' **then** mWinHTTP.Send('')

**else**

mWinHTTP.Send(CFxInternet.PathEncode('login='+cUser+'&pswd='+cPassword+aConditions+'&data\_file='+aJSONString));

aLogStr:= aLogStr+ #10#13+ mWinHTTP.StatusText;

Result := CheckResponsHeader(aLogStr,mWinHTTP.Status);

aResponseText := TEncoding.UTF8.GetString(mWinHTTP.ResponseBody);

**finally**

TJSONSuperObject.Free;

**end**;

**end**;

**except**

aLogStr := aLogStr+ #10#13 + 'Nelze navázat spojení se serverem.' + #10#13 + ExceptionMessage;

Result := False;

**end**;

**end**;

**function** CheckResponsHeader(**var** aLogStr:**String**;

aHTTPResponseCode:Longint):Boolean;

**begin**

Result:=false;

**if** aHTTPResponseCode = 200 **then**

**begin**

Result := true;

**end**

**else**

**begin**

aLogStr := aLogStr + #10#13 + 'Chyba spojení: Code ' + IntToStr(aHTTPResponseCode);

**end**;

**case** aHTTPResponseCode **of**

204 : aLogStr := aLogStr + #10#13 + 'Resources not found ';

400 : aLogStr := aLogStr + #10#13 + 'Missing regnos parameter; Too many regnos passed; Bad regno format; Bad country\_code ';

401 : aLogStr := aLogStr + #10#13 + 'Not authorized ';

403 : aLogStr := aLogStr + #10#13 + 'Expired account; Assigned API calls limit exhausted ';

**end**;

**end**;

*// vytvoří JSON pro přidání skladové karty*

**function** createJsonSCLine(aBO:TNxCustomBusinessObject;):**String**; *//mBO.. SC*

**var** mWeight:extended;

mSQL,mCode,mEAN:**String**;

mUnitsList:TStringList;

mDS: TMemoryDataset;

mWeightUnit:**String**;

mLots,mWeightINT:integer;

mPicture:**String**;

mPictData:TBytes;

mLength:integer;

**begin**

result:='';

**if** Assigned(aBO) **then**

**begin**

mPicture:='null';

mDS:= TMemoryDataset.Create(**nil**);

**if** **not** nxIsEmptyOID(aBO.GetFieldValueAsString('Picture\_ID')) **then**

**begin**

mPictData:=aBO.GetFieldValueAsBytes('Picture\_ID.BlobData');

mLength:=Length(mPictData);

**if** (mLength>0) **and** (mLength<=102400) **then**

**begin**

mPicture:=EncodeBase64(mPictData);

**if** NxLeft(mPicture,24)='DgAAAFROeEdESVBQaWN0dXJl' **then**

**begin**

mPicture:='"'+NxRight(mPicture,Length(mPicture)-NxCharPos('/',mPicture)+1)+'"';

**end**

**else**

**begin**

*//NxShowMessage('Nepřenesen obrázek','Skl. karta: '+aBO.DisplayName+' má obrázek v jiném formátu než jpg, proto nebyl přenesen na MailStep.',mdError,false,nil);*

mPicture:='null';

**end**;

**end**

**end**;

mSQL:='Select code, weight,WeightUnit, EAN, X\_MailStepID from StoreUnits where Parent\_ID='+quotedStr(aBO.OID);

*//NxShowSimpleMessage(sql,mSiteForm);*

**if** FillDatasetFromSQL(aBO.ObjectSpace,mSQL,mDS)>0 **then**

**begin**

mDS.Open;

mDS.First;

**while** **not** mDS.EoF **do**

**begin**

mLots:=0;

**if** (aBO.GetFieldValueAsInteger('Category') **in** [1,2]) **then** mLots:=1;

mWeightUnit:=varToStr(mDS.FieldValues('WeightUnit'));

mWeight:=NxIBStrToFloat(mDS.FieldValues('weight'));

**if** (mWeightUnit='1') **then** mWeight:=mWeight\*1000

**else** **if** (mWeightUnit='2') **then** mWeight:=mWeight\*1000000;

mWeightINT:=Round(mWeight);

mEAN:=mDS.FieldValues('EAN');

**if** result<>'' **then** result:=result+','+#13#10;

mCode:=mDS.FieldValues('X\_MailStepID');

**if** mCode='' **then** *// ještě nebyl přiřazen mailStepKod, zakládám novou položku*

**begin**

mCode:=mDS.FieldValues('Code');

**if** mCode=aBO.GetFieldValueAsString('MainUnitCode') **then** mCode:=aBO.GetFieldValueAsString('Code')

**else** mCode:=aBO.GetFieldValueAsString('Code')+cCodeSeparator+mCode;

**end**;

result:= result+ '{"prod\_id":"'+parseStr(mCode)+'","name":"'+parseStr(aBO.GetFieldValueAsString('Name'))+'","descr":"'+parseStr(aBO.GetFieldValueAsString('Name'))+'","barcode":"'+parseStr(mEAN)+'","weight":"'+IntToStr(mWeightINT)+'","virt":"0","lots":"'+intToStr(mLots)+'","image":'+mPicture+'}';

mDS.Next;

**end**;

*// NxShowEditorSite(NxCreateContext(aBO.ObjectSpace),result,true); // Zobrazení vrácených dat*

**end**

**else** NxShowMessage('Chyba skl. karty','Nebyly nalezeny žádné jednotky pro skl. kartu: '+aBO.DisplayName,mdError,false,**nil**);

**end**;

**end**;

**function** createJsonSCLineFromRow(aBO:TNxCustomBusinessObject; ):**String**; *// BO BillOfDeliveryRow*

**begin**

result:='';

**if** Assigned(aBO) **then**

**begin**

result:= '{"prod\_id":"'+parseStr(aBO.GetFieldValueAsString('StoreCard\_ID.Code'))+'","name":"'+parseStr(aBO.GetFieldValueAsString('StoreCard\_ID.Name'))+'","descr":"'+parseStr(aBO.GetFieldValueAsString('StoreCard\_ID.Name'))+'","barcode":"'+parseStr(aBO.GetFieldValueAsString('StoreCard\_ID.EAN'))+'","weight":"0","virt":"0","image":null}'

**end**;

**end**;

**function** createNewSCFromBoRow(aBO:TNxCustomBusinessObject):boolean; *//aBO.. BillOfDeliveryRow*

**var** mBO:TNxCustomBusinessObject;

**begin**

result:=false;

**if** Assigned(aBO) **then**

**begin**

mBO:=aBO.ObjectSpace.CreateObject(Class\_StoreCard);

**try**

mBO.Load(aBO.GetFieldValueAsString('StoreCard\_ID'),**nil**);

result:=ExportSC(mBO);

**finally**

mBO.free

**end**;

**end**;

**end**;

*/////////////////////////////////////*

*/////// Export StoreCard*

*////////////////////////////////////*

**function** ExportSC(mBO:TNxCustomBusinessObject):boolean; *//mBO.. StoreCard*

**var** mJSONString,mLogStr,mResponseText,mToken:**String**;

i:integer;

mBORows:TNxCustomBusinessMonikerCollection;

mBORow:TNxCustomBusinessObject;

**begin**

result:=false;

mToken:='';

**if** Assigned(mBO) **and** **not** mBO.GetFieldValueAsBoolean('X\_MailStageDontSend') **then**

**begin**

mJSONString:='['+createJsonSCLine(mBO)+']';

**if** **not** (WSPost(mLogStr,cURL+'product/set'+cEnd,mToken,mJSONString,mResponseText)) **then** NxShowMessage('Chyba připojení k serveru','Nelze se připojit k serveru pro nahrání skl. karty '+mBO.DisplayName+'.',mdError,false,**nil**)

**else**

**begin**

Result:=(Pos('"error":',mResponseText)=0) **and** (Pos('"errors":[{',mResponseText)=0);

**if** Result **then**

**begin**

mBO.SetFieldValueAsBoolean('X\_MailStage',true);

mBO.SetFieldValueAsDateTime('X\_MailStageDate',Now);

mBORows:=mBO.GetLoadedCollectionMonikerForFieldCode(mBO.GetFieldCode('StoreUnits'));

**for** i:=0 **to** mBORows.Count-1 **do**

**begin**

mBORow:=mBORows.BusinessObject[i];

**if** (mBORow.GetFieldValueAsString('X\_MailStepID')='') **then** *// pouze pokud jednotka ještě neměla přidělen MailStepKod*

**if** (mBORow.GetFieldValueAsString('Code')=mBO.GetFieldValueAsString('MainUnitCode')) **then** mBORow.SetFieldValueAsString('X\_MailStepID',mBO.GetFieldValueAsString('Code'))

**else** mBORow.SetFieldValueAsString('X\_MailStepID',mBO.GetFieldValueAsString('Code')+cCodeSeparator+mBORow.GetFieldValueAsString('Code'));

**end**;

**if** mBO.Validate **and** mBO.NeedSave **then** mBO.Save;

**end**

**else** NxShowMessage('Chyba importu do katalogu','Import skončil s chybou: '+#13#10+mResponseText,mdError,false,**nil**);

**end**;

**end**;

**end**;

*//// --Export*

**function** parseStr(S:**String**):**String**;

**begin**

result:=NxSearchReplace(S, '"','\"', [srAll]);

result:=NxSearchReplace(result, '&','(and)', [srAll]);

result := NxSearchReplace(result, ' ', ' ', [srAll]); *// paznak (tab) za mezeru*

**end**;

**function** ScExists(ACode:**String**; aOS:TNxCustomObjectSpace;):boolean;

**var** mLogStr,mResponseText,mJSONString,mToken:**String**;

mRes:**String**;

**begin**

result:=false;

**if** ACode<>'' **then**

result:=GetFirstSQLResult('Select id from StoreUnits where X\_MailstepID='+quotedStr(ACode),aOS)<>'';

**end**;

**function** CancelOrder(aBO:TNxCustomBusinessObject; aOS:TNxCustomObjectSpace;):boolean;

**var** mLogStr,mResponseText,mJSONString,mToken:**String**;

mRes:**String**;

**begin**

result:=false;

**if** **not** (WSPost(mLogStr,cURL+'order/cancel'+cEnd,mToken,mJSONString,mResponseText,'&order\_id='+aBO.GetFieldValueAsString('X\_OrderID'))) **then** NxShowMessage('Chyba připojení','Nelze se připojit na server.',mdError,false,**nil**)

**else**

**begin**

**if** (Pos('"error":',mResponseText)=0) **and** (Pos('"errors":[{',mResponseText)=0) **then**

**begin**

result:=true;

*//NxShowMessage('Importováno','Import vpořádku doběhl.',mdExclamation,false,nil);*

**end**

**else**

**begin**

NxShowMessage('Neúspěšné storno','Doklad '+aBO.DisplayName+' ('+aBO.GetFieldValueAsString('X\_OrderID')+') nelze stornovat: '+#13#10+ParseErrors(mResponseText),mdError,false,**nil**);

**end**;

**end**;

**end**;

**function** StockChange(aConditions:**String**;):**String**;

**var** mLogStr,mResponseText,mJSONString,mToken:**String**;

**begin**

result:='';

mJSONString:='';

**if** **not** (WSPost(mLogStr,cURL+'stockChange/get'+cEnd,mToken,mJSONString,mResponseText,aConditions)) **then** *//showmessage('Nelze se spojit se serverem.')*

**else**

**begin**

result:=mResponseText;

**end**;

**end**;

**procedure** StockChangeParseValues(OS: TNxCustomObjectSpace;

**var** Success: Boolean; **var** LogInfoStr: **String**);

**var** mJSONString,mToken,mResponseText:**String**;

mData : TJSONSuperObject;

mPole: TJSONSuperObjectArray;

i,j:integer;

mResults:TStringList;

mBO:TNxCustomBusinessObject;

mDateTime:TDateTime;

mPrev:**String**;

mErrList:TStringList;

mLot:**String**;

mFirmID:**String**;

mDRC:TMemoryDataset;

mSCIDs:TStringList;

mOrderRows:TStringList;

mOrderQuantity:TStringList;

mOrderID:**String**;

mDS:TMemoryDataset;

mSQL:**String**;

mStateDateTime:**String**;

mPreskocen:boolean;

mPrijemek:integer;

**begin**

Success := True;

mResponseText:='[]';

mJSONString:='';

mPrev:='';

**if** (LogInfoStr<>'') **then**

**begin**

mStateDateTime:=LogInfoStr;

**end**

**else**

**begin**

mSQL:='select Name from defrollData where clsid=''SKKEGMEH0YMOXCGP3NQIZPXSSC'' and Code=''lastState''';

mStateDateTime:=GetFirstSQLResult(mSQL,OS);

mStateDateTime:=NxTrim(mStateDateTime,'"');

**end**;

**if** mStateDateTime<>'' **then** mResponseText:=StockChange('&date='+mStateDateTime+'&type=1');

**if** mResponseText='' **then** Success:=false;

DateTimeToString(mStateDateTime,'YYYY-MM-DD hh:mm',Now);

*// showmessage(mResponseText);*

LogInfoStr:='';

**begin**

*//NxShowEditorSite(NxCreateContext(OS),mResponseText,true); // Zobrazení vrácených dat*

**if** mResponseText<>'[]' **then**

**begin**

mResults:=TStringList.Create;

mErrList:=TStringList.Create;

mOrderRows:=TStringList.Create;

mOrderQuantity:=TStringList.Create;

mPreskocen:=false;

mPrijemek:=0;

**try**

mData := TJSONSuperObject.ParseString(mResponseText, True);

**try**

mPole := mData.AsArray;

mPrev:='';

**for** i:=0 **to** mPole.Length-1 **do**

**begin**

**if** GetFirstSQLResult('Select id from storeDocuments where DocumentType=''20'' AND Description='+quotedStr(mPole.O[i].S['ms\_id']),OS)<>'' **then**

**begin**

**if** (mPrev<>'') **and** (mPrev<>mPole.O[i].S['ms\_id']) **and** (**not** mPreskocen) **and** assigned(mBO) **then** *// uložíme poslední příjemku*

**begin**

**if** **not** mBO.Validate **then**

**begin**

mBO.GetValidateErrors(mErrList);

LogInfoStr:=LogInfoStr+#13#10+'Příjemku '+mPrev+' se nepodařilo uložit, je nevalidní:'+#13#10+mErrList.Text;

**end**

**else** **if** mBO.NeedSave **then**

**begin**

mBO.Save;

inc(mPrijemek);

**for** j:=0 **to** mOrderRows.Count-1 **do** **begin**

OS.SQLExecute('update IssuedOrders2 set DeliveredQuantity=DeliveredQuantity+(UnitRate\*'+mOrderQuantity[j]+') where id='+quotedStr(mOrderRows[j]));

**end**;

mOrderRows.Clear;

mOrderQuantity.Clear;

**end**;

mBO.Free;

**end**;

**if** mPrev<>mPole.O[i].S['ms\_id'] **then** LogInfoStr:=LogInfoStr+#13#10+'Preskakuji jiz importovanou prijemku '+mPole.O[i].S['ms\_id'];

**if** (mPrev='') **then** mPrev:=mPole.O[i].S['ms\_id'];

mPreskocen:=true;

continue;

**end**;

mResults.Add(mPole.O[i].S['ms\_id']+' : kod: '+mPole.O[i].S['prod\_id']+' : mn: '+mPole.O[i].S['qnty']+' : sklad: '+mPole.O[i].S['store\_id']+' : sar: '+mPole.O[i].S['lot']+' : OP:');

**if** (mPole.O[i].I['cancel']=0) **and** (mPole.O[i].I['qnty']>0) **then** *//and False then*

**begin**

mFirmID:=cFirmID;

**if** mPrev='' **then**

**begin**

mBO:=OS.CreateObject(Class\_ReceiptCard);

mBO.New;

mBO.Prefill;

mBO.SetFieldValueAsString('DocQueue\_ID',cReceiptCardDocQueue);

mDateTime:=CfxDate.StrToDateEx(NxLeft(mPole.O[i].S['date'],10),'YYYY-MM-DD','-');

*//mDateTime:=StrToDateTime(DateTimeToStr(mDateTime)+NxRight(mPole.O[i].S['date'],6));*

mBO.SetFieldValueAsDateTime('DocDate$Date',mDateTime);

mPrev:=mPole.O[i].S['ms\_id'];

mBO.SetFieldValueAsString('Description',mPole.O[i].S['ms\_id']);

mBO.SetFieldValueAsString('Firm\_ID',cFirmID);

mOrderID:=mPole.O[i].S['advice\_id'];

**if** **not** NxIsEmptyOID(mOrderID) **then**

**begin**

mDS:=TMemoryDataset.Create(**nil**);

**try**

mSQL:='Select \* from IssuedOrders io where id='+quotedStr(mOrderID);

**if** FillDatasetFromSQL(OS,mSQL,mDS)>0 **then**

**begin**

mDS.Open;

mDS.First;

mBO.SetFieldValueAsString('Firm\_ID', mDS.FieldByName('Firm\_ID').AsString);

mBO.SetFieldValueAsString('FirmOffice\_ID', mDS.FieldByName('FirmOffice\_ID').AsString);

*// mBO.SetFieldValueAsString('Description', mDS.FieldByName('Description').AsString);*

mBO.SetFieldValueAsString('Currency\_ID', mDS.FieldByName('Currency\_ID').AsString);

mBO.SetFieldValueAsString('Person\_ID', mDS.FieldByName('Person\_ID').AsString);

mDS.Close;

**end**

**else**

**begin**

LogInfoStr:=LogInfoStr+#13#10+'ERR: Pro příjemku '+mPole.O[i].S['ms\_id']+' se nepodařilo se dohledat OV ('+mOrderID+')';

**end**;

**finally**

mDS.Free;

**end**;

**end**;

**end**;

*// doplňujeme řádky, dokud jsou shodná čísla dokladu*

**if** mPrev=mPole.O[i].S['ms\_id'] **then**

**begin**

addReceiptCardRow(mBO,mPole.O[i].S['prod\_id'],mPole.O[i].I['qnty'],mPole.O[i].S['store\_id'],mPole.O[i].S['lot'],mPole.O[i].S['advice\_id'],mOrderRows,mOrderQuantity,LogInfoStr);

**end**

**else** *// jinak čísla dokladů nejsou shodná*

**begin**

**if** mPrev<>'' **then**

**begin**

**if** **not** mPreskocen **then** **begin**

**if** **not** mBO.Validate **then**

**begin**

mBO.GetValidateErrors(mErrList);

LogInfoStr:=LogInfoStr+#13#10+'Příjemku '+mPrev+' se nepodařilo uložit, je nevalidní:'+#13#10+mErrList.Text;

**end**

**else** **if** mBO.NeedSave **then**

**begin**

mBO.Save;

inc(mPrijemek);

**for** j:=0 **to** mOrderRows.Count-1 **do** **begin**

OS.SQLExecute('update IssuedOrders2 set DeliveredQuantity=DeliveredQuantity+(UnitRate\*'+mOrderQuantity[j]+') where id='+quotedStr(mOrderRows[j]));

**end**;

mOrderRows.Clear;

mOrderQuantity.Clear;

**end**;

mBO.Free;

**end**;

mPreskocen:=false;

mBO:=OS.CreateObject(Class\_ReceiptCard);

mBO.New;

mBO.Prefill;

mBO.SetFieldValueAsString('DocQueue\_ID',cReceiptCardDocQueue);

mDateTime:=CfxDate.StrToDateEx(NxLeft(mPole.O[i].S['date'],10),'YYYY-MM-DD','-');

mBO.SetFieldValueAsDateTime('DocDate$Date',mDateTime);

mPrev:=mPole.O[i].S['ms\_id'];

mBO.SetFieldValueAsString('Description',mPole.O[i].S['ms\_id']);

mBO.SetFieldValueAsString('Firm\_ID',cFirmID);

mOrderID:=mPole.O[i].S['advice\_id'];

**if** **not** NxIsEmptyOID(mOrderID) **then**

**begin**

mDS:=TMemoryDataset.Create(**nil**);

**try**

mSQL:='Select \* from IssuedOrders where id='+quotedStr(mOrderID);

**if** FillDatasetFromSQL(OS,mSQL,mDS)>0 **then**

**begin**

mDS.Open;

mDS.First;

mBO.SetFieldValueAsString('Firm\_ID', mDS.FieldByName('Firm\_ID').AsString);

mBO.SetFieldValueAsString('FirmOffice\_ID', mDS.FieldByName('FirmOffice\_ID').AsString);

mBO.SetFieldValueAsString('Currency\_ID', mDS.FieldByName('Currency\_ID').AsString);

mBO.SetFieldValueAsString('Person\_ID', mDS.FieldByName('Person\_ID').AsString);

**end**

**else**

**begin**

LogInfoStr:=LogInfoStr+#13#10+'ERR: Pro příjemku '+mPole.O[i].S['ms\_id']+' se nepodařilo se dohledat OV ('+mOrderID+')';

**end**;

**finally**

mDS.Free;

**end**;

**end**;

addReceiptCardRow(mBO,mPole.O[i].S['prod\_id'],mPole.O[i].I['qnty'],mPole.O[i].S['store\_id'],mPole.O[i].S['lot'],mPole.O[i].S['advice\_id'],mOrderRows,mOrderQuantity,LogInfoStr);

**end**;

**end**;

**end**;

**end**; *// prošlé všechny importované řádky*

**if** (mPrev<>'') **and** **not** mPreskocen **then** *// uložíme poslední příjemku*

**begin**

**if** **not** mBO.Validate **then**

**begin**

mBO.GetValidateErrors(mErrList);

LogInfoStr:=LogInfoStr+#13#10+'Příjemku '+mPrev+' se nepodařilo uložit, je nevalidní:'+#13#10+mErrList.Text;

**end**

**else** **if** mBO.NeedSave **then**

**begin**

mBO.Save;

inc(mPrijemek);

**for** j:=0 **to** mOrderRows.Count-1 **do** **begin**

OS.SQLExecute('update IssuedOrders2 set DeliveredQuantity=DeliveredQuantity+(UnitRate\*'+mOrderQuantity[j]+') where id='+quotedStr(mOrderRows[j]));

**end**;

mOrderRows.Clear;

mOrderQuantity.Clear;

**end**;

**end**;

**if** mResults.Count>0 **then** **begin**

*// NxShowEditorSite(NxCreateContext(OS),mResults.Text,true);*

success:=true;

**end**;

LogInfoStr:=LogInfoStr+#13#10+'Proběhlo načtení '+intToStr(mPole.Length)+' řádků příjemek. - bylo vytvořeno '+intToStr(mPrijemek)+' příjemek.';

**finally**

mData.Free;

**end**;

**finally**

mOrderRows.Free;

mOrderQuantity.Free;

mResults.Free;

mErrList.Free;

**end**;

**end** **else** LogInfoStr:=LogInfoStr+#13#10+'Nejsou žádné nové příjemky.';

**end**;

**if** Success **then**

**begin**

mSQL:='update defrollData set Name='+quotedStr(mStateDateTime)+' where clsid=''SKKEGMEH0YMOXCGP3NQIZPXSSC'' and Code=''lastState''';

OS.SQLExecute(mSQL);

**end**;

**end**;

**procedure** addReceiptCardRow(**var** aBO:TNxCustomBusinessObject; aProductMS:**String**; aQnt:integer; aStoreCode,aLot,aOrderID:**String**;**var** aOrderRows:TStringList; **var** aOrderQuantity:TStringList; **var** aLog:**String**;);

**var** mBORows:TNxCustomBusinessMonikerCollection;

mBORow:TNxCustomBusinessObject;

mSc,mID,mUnit,mStoreBatchID:**String**;

mStore:**String**;

mStoreBatches:TNxCustomBusinessMonikerCollection;

mStoreBatch:TNxCustomBusinessObject;

mOS:TNxCustomObjectSpace;

mDS:TMemoryDataset;

mSQL:**String**;

mUnitPrice:Extended;

mDay,mDate:**String**;

mMonth,mYear:char;

mNmb:integer;

mBatchDate:TDate;

**begin**

**if** assigned(aBO) **and** (aQnt>0) **and** (aProductMS<>'') **then**

**begin**

mOS:=aBO.ObjectSpace;

mUnit:=GetFirstSQLResult('Select Parent\_ID||'';''||code from StoreUnits where X\_MailStepID='+quotedStr(aProductMS),mOS,'');

**if** (mUnit<>'') **then**

**begin**

mUnit:=NxTrim(mUnit,'"');

mID := NxToken(mUnit,';');

mStore:=GetFirstSQLResult('Select id from Stores where hidden=''N'' and code='+quotedStr(aStoreCode),mOS,cStoreID);

mBORows:=aBO.GetLoadedCollectionMonikerForFieldCode(aBO.GetFieldCode('Rows'));

mBORow:=mBORows.AddNewObject;

mBORow.SetFieldValueAsInteger('RowType',3);

mBORow.SetFieldValueAsString('Division\_ID',cDivisionID);

mBORow.SetFieldValueAsString('Store\_ID',mStore);

mBORow.SetFieldValueAsString('StoreCard\_ID',mID);

mBORow.SetFieldValueAsString('qUnit',mUnit);

mBORow.SetFieldValueAsFloat('UnitQuantity', aQnt);

*// zatím jen vytvořím vazbu*

**if** **not** NxIsEmptyOID(aOrderID) **then** **begin**

mDS:=TMemoryDataset.Create(**nil**);

**try**

mSQL:='Select io2.id,io2.unitPrice,io2.totalPrice,io2.quantity,io2.UnitRate,io2.Store\_ID,io2.Division\_ID,io2.BusOrder\_ID,io2.BusTransaction\_ID,io2.BusProject\_ID from IssuedOrders2 io2 join IssuedOrders io on io2.Parent\_ID=io.id and io.id='+quotedStr(aOrderID)+ ' where io2.StoreCard\_ID='+quotedStr(mID)+' AND ((io2.Quantity/io2.UnitRate)='+intToStr(aQnt)+') AND io2.qUnit='+quotedStr(mUnit);

**if** FillDatasetFromSQL(mOS,mSQL,mDS)>0 **then**

**begin**

mBORow.SetFieldValueAsString('Provide\_ID', aOrderID);

mDS.Open;

mDS.First;

mBORow.SetFieldValueAsString('ProvideRow\_ID', mDS.FieldByName('id').AsString);

mBORow.SetFieldValueAsString('Division\_ID', mDS.FieldByName('Division\_ID').AsString);

mBORow.SetFieldValueAsString('BusOrder\_ID', mDS.FieldByName('BusOrder\_ID').AsString);

mBORow.SetFieldValueAsString('BusTransaction\_ID', mDS.FieldByName('BusTransaction\_ID').AsString);

mBORow.SetFieldValueAsString('BusProject\_ID', mDS.FieldByName('BusProject\_ID').AsString);

mBORow.SetFieldValueAsString('Store\_ID', mDS.FieldByName('Store\_ID').AsString);

*// zapamatuji si řádek a množství, abych mohl po uložení opravit deliveredQuantity*

aOrderRows.add(mDS.FieldByName('id').AsString);

aOrderQuantity.add(IntToStr(aQnt));

mUnitPrice:=mDS.FieldByName('unitPrice').AsFloat;

**if** (mUnitPrice=0) **and** (mDS.FieldByName('quantity').AsFloat>0) **then**

**begin**

mUnitPrice:=mDS.FieldByName('totalPrice').AsFloat/mDS.FieldByName('quantity').AsFloat/mDS.FieldByName('unitRate').AsFloat;

**end**;

mBORow.SetFieldValueAsFloat('UnitPrice', mUnitPrice);

mDS.Close;

**end** **else** aLog:=aLog + #13#10+ 'ERR: Nepodařilo se dohledat řádek OV ('+aOrderID+') pro kartu: '+mID+' , jedn: '+mUnit+' a množství: '+intToStr(aQnt); *//showmessage(mSQL);*

**finally**

mDS.Free;

**end**;

**end**;

**if** (aLot<>'null') **and** (mBORow.GetFieldValueAsInteger('StoreCard\_ID.Category')**in** [2,3]) **then**

**begin**

mStoreBatches:=mBORow.GetLoadedCollectionMonikerForFieldCode(mBORow.GetFieldCode('DocRowBatches'));

mStoreBatch:=mStoreBatches.AddNewObject;

mStoreBatch.Prefill;

mStoreBatch.SetFieldValueAsString('qUnit',mUnit);

mStoreBatch.SetFieldValueAsFloat('UnitQuantity', aQnt);

mStoreBatchID:=NxTrim(GetFirstSQLResult('Select id from StoreBatches where hidden=''N'' and StoreCard\_ID='+quotedStr(mID)+' and name='+quotedStr(aLot),mOS,''),'"');

**if** mStoreBatchID<>'' **then**

**begin**

mStoreBatch.SetFieldValueAsString('StoreBatch\_ID',mStoreBatchID)

**end**

**else** *// nova sarze*

**begin**

mStoreBatch.SetFieldValueAsBoolean('NewBatch',true);

mStoreBatch.SetFieldValueAsString('NewBatchName',aLot);

*//specificke ziskavani Data expirace pro Bona Naturu*

mDay:=NxRight(aLot,2);

**if** TryStrToInt(mDay,mNmb) **and** (length(aLot)>3) **then**

**begin**

mMonth:=aLot[Length(aLot)-2];

mYear:=aLot[Length(aLot)-3];

mNmb:=ord(mMonth)-ord('A')+1;

mDate:=mDay+'.'+intToStr(mNmb);

mNmb:=ord(mYear)-ord('A')+2004+3;*// doba výroby + 3 letá expirace*

mDate:=mDate+'.'+intToStr(mNmb);

**if** (TryStrToDate(mDate,mBatchDate)) **then**

**begin**

mStoreBatch.SetFieldValueAsDateTime('NewBatchExpirationDate$DATE',mBatchDate);

**end**;

**end**;

**end**

*// else RaiseException('Pro skladovou kartu '+aBO.DisplayName+' neexistuje šarže '+aLot);*

**end**;

*// určitě by to chtělo nějaké ceny*

**end**;

**end**;

**end**;

*{*

*// Kontrola změn stavů*

*}*

**procedure** CheckStates(OS: TNxCustomObjectSpace;

**var** Success: Boolean; **var** LogInfoStr: **String**);

**var** mJSONString,mToken,mResponseText,mLogStr:**String**;

mData : TJSONSuperObject;

mPole: TJSONSuperObjectArray;

i:integer;

mResults:TStringList;

mBO:TNxCustomBusinessObject;

mDateTime:TDateTime;

mDoc,mID:**String**;

**begin**

Success := True;

LogInfoStr := '';

mResponseText:='';

mJSONString:='';

**if** **not** (WSPost(mLogStr,cURL+'status/get'+cLogin,mToken,mJSONString,mResponseText)) **then**

**begin**

LogInfoStr := 'ERR: Nepodařilo se spojit se serverem'+#13#10;

**end**

**else**

**begin**

*//NxShowEditorSite(NxCreateContext(Sender.Site.BaseObjectSpace),mJSONString,true);*

**if** mResponseText<>'[]' **then**

**begin**

mResults:=TStringList.Create;

**try**

mData := TJSONSuperObject.ParseString(mResponseText, True);

**try**

mPole := mData.AsArray;

**for** i:=0 **to** mPole.Length-1 **do**

**begin**

OS.SQLSelect('Select id||'';''||DocumentType from StoreDocuments where DocumentType in ('+quotedStr('21')+','+quotedStr('22')+') AND X\_OrderID='+mPole.O[i].S['order\_id'],mResults);

**if** mResults.Count=1 **then**

**begin**

mDoc:=NxTrim(mResults[0],'"');

mID:= NxToken(mDoc,';');

**if** mDoc='22' **then** *// PRV*

mBO:=OS.CreateObject(Class\_OutgoingTransfer)

**else** *// jinak DL*

mBO:=OS.CreateObject(Class\_BillOfDelivery);

**try**

mBO.Load(mID,**nil**);

mBO.SetFieldValueAsInteger('X\_State',mPole.O[i].I['status\_id']);

mDateTime:=CfxDate.StrToDateEx(NxLeft(mPole.O[i].S['status\_date'],10),'YYYY-MM-DD','-');

mDateTime:=StrToDateTime(DateTimeToStr(mDateTime)+NxRight(mPole.O[i].S['status\_date'],6));

mBO.SetFieldValueAsDateTime('X\_StateDateTime',mDateTime);

**if** mPole.O[i].S['package\_id']<>'null' **then** mBO.SetFieldValueAsString('X\_PackageNumber',mPole.O[i].S['package\_id']);

**if** mBO.Validate **and** mBO.NeedSave **then** mBO.Save;

**finally**

mBO.Free;

**end**;

**end**;

**if** mResults.Count>1 **then**

**begin**

LogInfoStr:=LogInfoStr+'ERR: Více objednávek má stejné číslo '+mPole.O[i].S['order\_id']+#13#10;

Success:=false;

**end**;

**end**;

LogInfoStr:=LogInfoStr+'Proběhla změna stavů '+intToStr(mPole.Length)+' objednávek';

**finally**

mData.Free;

**end**;

**finally**

mResults.Free;

**end**;

**end** **else** LogInfoStr:=LogInfoStr+'Nejsou žádné změny stavů objednávek.';

**end**;

**end**;

**function** ParseErrors(aRespond:**String**):**String**;

**var** mResults:TStringList;

mData : TJSONSuperObject;

mPole: TJSONSuperObjectArray;

i:integer;

**begin**

result:='';

**if** (Pos('"error":',aRespond)>0) **or** (Pos('"errors":[{',aRespond)>0) **then**

**begin**

mResults:=TStringList.Create;

**try**

**try**

mData := TJSONSuperObject.ParseString(aRespond, True);

**try**

**try**

mPole := mData.A['errors'];

**for** i:=0 **to** mPole.Length-1 **do**

**begin**

mResults.Add('DL: '+mPole.O[i].S['order\_id']+' chyba: '+mPole.O[i].S['message']);

**end**;

**Except**

RaiseException('Neočekávaná návratová hodnota:'+#13#10+aRespond);

**end**;

**finally**

mPole.Free;

**end**;

**finally**

mData.Free;

**end**;

**if** mResults.Count>0 **then** result:=mResults.Text;

**finally**

mResults.Free;

**end**;

**end**;

**end**;

**function** getErrSqlIds(aRespond:**String**):**String**;

**var** mResults:TStringList;

mData : TJSONSuperObject;

mPole: TJSONSuperObjectArray;

i:integer;

**begin**

result:='';

**if** (Pos('"error":',aRespond)>0) **or** (Pos('"errors":[{',aRespond)>0) **then**

**begin**

mResults:=TStringList.Create;

**try**

**try**

mData := TJSONSuperObject.ParseString(aRespond, True);

**try**

**try**

mPole := mData.A['errors'];

**for** i:=0 **to** mPole.Length-1 **do**

**begin**

mResults.Add(quotedStr(mPole.O[i].S['order\_id']));

**end**;

**Except**

RaiseException('Neočekávaná návratová hodnota:'+#13#10+aRespond);

**end**;

**finally**

mPole.Free;

**end**;

**finally**

mData.Free;

**end**;

**if** mResults.Count>0 **then** result:=mResults.CommaText;

**finally**

mResults.Free;

**end**;

**end**;

**end**;

Orders

**uses** 'eu.abra.peti.MailStep.lib';

*{*

*Vyvolává se po provedení inicializace agendy/formuláře. V tento okamžik je již na formuláři dostupný SiteContext.*

*}*

**procedure** InitSite\_Hook(Self: TSiteForm);

**var** mAction,mAction2: TAction;

**begin**

mAction := Self.GetNewAction;

mAction.name:='mGenJSONDL';

mAction.ShowControl := True;

mAction.ShowMenuItem := True;

mAction.Category := 'tabList,tabDetail';

mAction.Caption := 'Expedovat';

mAction.OnExecute :=@exportDLsMailStep;

mAction2 := Self.GetNewAction;

mAction2.name:='mStornoMSDL';

mAction2.ShowControl := True;

mAction2.ShowMenuItem := True;

mAction2.Category := 'tabList,tabDetail';

mAction2.Caption := 'Zrušit Expedici';

mAction2.OnExecute :=@StornoMailStepClick;

**end**;

**procedure** StornoMailStepClick(Sender:TComponent);

**var** mIDs:TStringList;

mBO,mBORow:TNxCustomBusinessObject;

mJSONString:**String**;

mLogStr,mResponseText,mToken:**String**;

i:integer;

mOS:TNxCustomObjectSpace;

mErrs:TStringList;

**begin**

**if** (Assigned(Sender) **And** Assigned(Sender.DynSite) **And** Assigned(Sender.DynSite.CurrentObject)) **then**

**begin**

mOS:=Sender.Site.BaseObjectSpace;

mErrs:=TStringList.Create;

mIDs:=TStringList.Create;

**try**

Sender.Site.List.GetSelectedId(mIDs);

**if** mIDs.Count>0 **then**

**begin**

**for** i:=0 **to** mIDs.count-1 **do**

**begin**

mBO:=mOs.CreateObject(Sender.DynSite.CurrentObject.CLSID);

**try**

mBO.Load(mIDs[i],**nil**);

**if** mBO.GetFieldValueAsInteger('X\_State')=0 **then**

**begin**

**if** **not** CancelOrder(mBO,mOS) **then** mErrs.add(mBO.DisplayName +' se nepodařilo stornovat.')

**else** mErrs.add(mBO.DisplayName +' Stornován.');

**end** **else** NxShowMessage('Objednávku nelze zrušit','Stav objednávky již neumožňuje její zrušení.',mdError,false,**nil**);

**finally**

mBO.Free;

**end**;

**end**;

**if** mErrs.Count>0 **then** NxShowMessage('Zpracováno',mErrs.Text,mdExclamation,false,**nil**);

**end**

**else** NxShowMessage('Žádné záznamy','Nebyly vybrány žádné záznamy.',mdError,false,**nil**);

**finally**

mErrs.Free;

mIDs.Free;

**end**;

**end**;

**end**;

**procedure** exportDLsMailStep(Sender:TComponent);

**var** mIDs:TStringList;

mBO,mBORow:TNxCustomBusinessObject;

mJSONString,mJSONwithoutPDF:**String**;

mLogStr,mResponseText,mToken:**String**;

i:integer;

mOS:TNxCustomObjectSpace;

mNow:Double;

mRowsCount:integer;

**begin**

mToken:='';

**if** (Assigned(Sender) **And** Assigned(Sender.DynSite) **and** Assigned(Sender.DynSite.CurrentObject)) **then**

**begin**

mOS:=Sender.Site.BaseObjectSpace;

mIDs:=TStringList.Create;

**try**

Sender.Site.List.GetSelectedId(mIDs);

**if** mIDs.Count>0 **then**

**begin**

mJSONString:='['+#13#10;

**for** i:=0 **to** mIDs.Count-1 **do**

**begin**

mBO:=mOS.CreateObject(Sender.DynSite.CurrentObject.CLSID);

**try**

mBO.Load(mIDs[i],**nil**);

**if** i>0 **then** mJSONString:=mJSONString+','+createJsonDL(mBO,mOS,mRowsCount,mJSONwithoutPDF)+#13#10

**else**

mJSONString:=mJSONString+createJsonDL(mBO,mOS,mRowsCount,mJSONwithoutPDF)+#13#10;

**finally**

mBO.Free;

**end**;

**end**;

mJSONString:=mJSONString+']';

mNow:=Now;

*// NxShowEditorSite(NxCreateContext(Sender.Site.BaseObjectSpace),mJSONString,true); // dotaz do MailStepu*

*//Exit;*

**if** **not** (WSPost(mLogStr,cURL+'order/set'+cEnd,mToken,mJSONString,mResponseText)) **then**

NxShowMessage('Chyba připojení','Nelze se připojit na server.',mdError,false,**nil**)

**else**

**begin** *// neúspěšné nahrání*

**if** (Pos('"error":',mResponseText)=0) **and** (Pos('"errors":[{',mResponseText)=0) **then**

**begin**

mOS.SqlExecute('Update StoreDocuments set X\_MS\_CreatedDateTime='+ReplaceStr(FloatToStr(mNow),',','.')+' where id in ('''+ReplaceStr(mIDs.CommaText,',',''',''')+''') and X\_MS\_CreatedDateTime=0');

NxShowMessage('Importováno','Import vpořádku doběhl. ('+intToStr(mRowsCount)+' řádků do MS)',mdExclamation,false,**nil**);

mOS.SqlExecute('Update StoreDocuments set X\_MS\_JSON='+quotedStr(ReplaceStr(mJSONwithoutPDF,'''',''''''))+' where id in ('''+ReplaceStr(mIDs.CommaText,',',''',''')+''')');

**end**

**else**

**begin**

mOS.SqlExecute('Update StoreDocuments set X\_MS\_CreatedDateTime='+ReplaceStr(FloatToStr(mNow),',','.')+' where id in ('''+ReplaceStr(mIDs.CommaText,',',''',''')+''') and (X\_OrderID not in ('+getErrSqlIds(mResponseText)+')) and X\_MS\_CreatedDateTime=0');

NxShowMessage('Chyba importu','Import skončil s chybou: '+#13#10+ParseErrors(mResponseText),mdError,false,**nil**);

**end**;

**end**;

**end**

**else** NxShowMessage('Žádné záznamy','Nebyly vybrány žádné záznamy.',mdError,false,**nil**);

**finally**

mIDs.Free;

**end**;

**end**;

**end**;

**function** createJsonDL(mBO:TNxCustomBusinessObject; aOS:TNxCustomObjectSpace; **var** aRowsCount:integer; **var** aJSONwithoutPDF:**String**):**String**;

**var** mBORow:TNxCustomBusinessObject;

mRowColl:TNxCustomBusinessMonikerCollection;

i,y:integer;

mCourier:integer;

mIIid:**String**;

mPDFOutputDir, mOutputFile, mShellParams,mStreet,mHnmb:**String**;

mNbr:integer;

mReportDL,mReportFV,mReportPRV:**String**;

mOrderID:**String**;

mIIBO:TNxCustomBusinessObject;

mDobirka:double;

mDobirkaMena:**String**;

mCurrRowsCount:integer;

**begin**

result:='';

**if** Assigned(mBO) **then**

**begin**

mCourier:=1;

**if** **not** NxIsEmptyOID(mBO.GetFieldValueAsString('TransportationType\_ID')) **then**

mCourier:=mBO.GetFieldValueAsInteger('TransportationType\_ID.X\_Courier');

**if** **not** ((mCourier>1) **and** (mCourier<17)) **then** mCourier:=1;

mStreet:=mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.Street');

mHnmb:=NxRight(mStreet,Length(mStreet)-NxCharPosR(' ',mStreet));

**if** mHnmb <>'' **then**

**begin**

**if** (TryStrToInt(mHnmb[1],mNbr)) **then**

**begin**

mStreet:=NxLeft(mStreet,NxCharPosR(' ',mStreet)-1);

**end** **else** mHnmb:='';

**end**;

mOrderID:=intToStr(mBO.GetFieldValueAsInteger('X\_OrderID'));

mDobirka:=0;

mDobirkaMena:='';

mIIid:=GetFirstSQLResult('Select ii.id from issuedInvoices ii join IssuedInvoices2 ii2 on (ii.id=ii2.Parent\_ID and ii2.Provide\_ID='+QuotedStr(mBO.OID)+')',aOS);

mIIid:=NxTrim(mIIid,'"');

**if** **not** NxIsEmptyOID(mIIid) **then**

**begin**

mIIBO:=aOS.CreateObject(Class\_IssuedInvoice);

**try**

mIIBO.Load(mIIid,**nil**);

mOrderID:=mIIBO.GetFieldValueAsString('VarSymbol');

mBO.SetFieldValueAsInteger('X\_OrderID',StrToInt(mOrderID));

mDobirka:=mIIBO.GetFieldValueAsFloat('Amount');

mDobirkaMena:=mIIBO.GetFieldValueAsString('Currency\_ID.Code');

**finally**

mIIBO.Free;

**end**;

**end**;

result:= '{"order\_id":"'+mOrderID+'","company":"'+parseStr(mBO.GetFieldValueAsString('Firm\_ID.Name'))+'","street":"'+parseStr(mStreet)+'","house\_no":"'+parseStr(mHnmb)+'","city":"'+parseStr(mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.City'))+'","zip":"'+parseStr(ReplaceStr(mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.PostCode'),' ',''))+'","country":"'+parseStr(mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.CountryCode'))+'","phone":"'+parseStr(mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.PhoneNumber1'))+'","email":"'+parseStr(mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.EMail'))+'","surname":"'+parseStr(mBO.GetFieldValueAsString('FirmOffice\_ID.Address\_ID.Recipient'))+'","vat\_id":"'+parseStr(mBO.GetFieldValueAsString('Firm\_ID.VATIdentNumber'))+'",'+#13#10;

result:=result+'"company\_id":"'+parseStr(mBO.GetFieldValueAsString('Firm\_ID.OrgIdentNumber'))+'","courier\_id":"'+intToStr(mCourier-1)+'","order\_date":"'+CFxDate.DateToStr(mBO.GetFieldValueAsDateTime('DocDate$Date'), 'YYYY/MM/DD', '-')+'","doc\_id":"'+parseStr(replaceStr(mBO.DisplayName,'/','-'))+'",'+#13#10;

**if** **not** NxIsEmptyOID(mIIid) **then** result:=result+'"var\_sym": "'+mOrderID+'","price": "'+NxFloatToIBStr(mDobirka)+'","currency": "'+mDobirkaMena+'",';

**if** (mDobirka>0) **and** mBO.GetFieldValueAsBoolean('X\_MS\_CashOndelivery') **then** result:=result+'"cash\_ondel": "'+NxFloatToIBStr(mDobirka)+'",'+#13#10;

result:=result+'"items": ['+#13#10;

mRowColl:=mBO.GetLoadedCollectionMonikerForFieldCode(mBO.GetFieldCode('Rows'));

y := 0;

**for** i:=0 **to** mRowColl.Count-1 **do**

**begin**

mBORow:=mRowColl.BusinessObject[i];

**if** (mBORow.GetFieldValueAsInteger('RowType')=3) **and** (**not** mBORow.GetFieldValueAsBoolean('StoreCard\_ID.X\_MailStageDontSend')) **then**

**begin**

**if** y>0 **then** **begin**

result:=result+','+CreateJsonRow(mBORow,mCurrRowsCount) ;

**end** **else** **begin**

result:=result+CreateJsonRow(mBORow,mCurrRowsCount);

**end**;

y := y + mCurrRowsCount;

**end**;

**end**;

aRowsCount:=y;

result:=result+']'+#13#10;

aJSONwithoutPDF:=result+'}';

*// připojení PDF s dodacím listem*

**if** **not** NxIsEmptyOID(mBO.GetFieldValueAsString('Firm\_ID.X\_ReportDL')) **then** mReportDL:=mBO.GetFieldValueAsString('Firm\_ID.X\_ReportDL')

**else** mReportDL:=cReportIDDL;

**if** **not** NxIsEmptyOID(mBO.GetFieldValueAsString('Firm\_ID.X\_ReportPRV')) **then** mReportPRV:=mBO.GetFieldValueAsString('Firm\_ID.X\_ReportPRV')

**else** mReportPRV:=cReportIDPRV;

**if** **not** NxIsEmptyOID(mBO.GetFieldValueAsString('Firm\_ID.X\_ReportFV')) **then** mReportFV:=mBO.GetFieldValueAsString('Firm\_ID.X\_ReportFV')

**else** mReportFV:=cReportIDII;

**if** **not** NxIsEmptyOID(mIIid) **and** (mBO.GetFieldValueAsBoolean('X\_AttachIssuedInvoice')) **then**

**begin**

*// sloučení PDF*

mOutputFile := '~TMP\_FV\_DL\_'+NxGetActualUserID(aOS)+'.PDF';

mPDFOutputDir := NxGetTempDir;

**if** mBO.ClassName='TNxBillOfDelivery' **then** *// tiskneme dodak*

mShellParams := printTempPdf(aOS,mIIid,cDynSourceIDII,mReportFV)+' '+printTempPdf(aOS,mBO.OID,cDynSourceIDDL,mReportDL) + ' output '+mPDFOutputDir+'\'+mOutputFile

**else** *// jinak tiskneme prevodku vydej*

mShellParams := printTempPdf(aOS,mIIid,cDynSourceIDII,mReportFV)+' '+printTempPdf(aOS,mBO.OID,cDynSourceIDPRV,mReportPRV) + ' output '+mPDFOutputDir+'\'+mOutputFile;

**if** (**not** ShellAPI.Execute('open', cPDFtkFile, mShellParams, cPDFtkPath)) **then**

RaiseException('Chyba při sloučení vygenerovaných PDF.');

i := 0;

**while** **not** FileExists(mPDFOutputDir + '\' + mOutputFile) **do** **begin**

NxSleep(1000);

Inc(i);

**if** i = 10 **then** break;

**end**;

**if** FileExists(mPDFOutputDir + '\' + mOutputFile) **then**

**begin**

result:=result+',"doc\_tmpl":"1","doc\_file":"'+AttachBlobBase64(aOS,mPDFOutputDir + '\' + mOutputFile)+'"'+#13#10;

**end**

**else** **begin**

NxShowMessage('Chyba nahrávání souborů','Pro doklad '+mBO.DisplayName+' se nepodařilo připojit PDF soubor s doklady.',mdError,false,**nil**);

**end**;

**end** *// neexistuje faktura*

**else**

**begin**

**if** mBO.ClassName='TNxBillOfDelivery' **then** *// tiskneme dodak*

result:=result+',"doc\_tmpl":"1","doc\_file":"'+AttachBlobBase64(aOS,printTempPdf(aOS,mBO.OID,cDynSourceIDDL,mReportDL))+'"'+#13#10

**else** *// jinak PRV*

result:=result+',"doc\_tmpl":"1","doc\_file":"'+AttachBlobBase64(aOS,printTempPdf(aOS,mBO.OID,cDynSourceIDPRV,mReportPRV))+'"'+#13#10;

**end**;

result:=result+'}';

**if** mBO.Validate **and** mBO.NeedSave **then** mBO.Save; *// Pozor, tzn neposílat již odeslané zásilky, ke kterým původně nebyla FV a nyní již je*

**end**;

**end**;

**function** createJsonRow(aBO:TNxCustomBusinessObject; **var** aRowsCount:integer;):**String**; *// pracuje s řádky DL - vytváří JSON pro Export řádků do MailStep*

**var** mCode:**String**;

mSQL:**String**;

mStoreBatches:TNxCustomBusinessMonikerCollection;

mStoreBatch:TNxCustomBusinessObject;

i:integer;

mOS:TNxCustomObjectSpace;

**begin**

result:='';

aRowsCount:=0;

**if** Assigned(aBO) **then**

**begin**

mOS:=aBO.ObjectSpace;

mSQL:='Select X\_MailStepID from StoreUnits where Parent\_ID='+quotedStr(aBO.GetFieldValueAsString('StoreCard\_ID'))+' AND code='+quotedStr(aBO.GetFieldValueAsString('QUnit'));

mCode:=GetFirstSQLResult(mSQL,mOS);

mCode:=NxTrim(mCode,'"');

**if** (mCode='') **then***//or (not (SCExists(mCode,mOS))) then*

**begin**

createNewSCFromBoRow(aBO);

**end**;

**if** mCode='' **then** mCode:=GetFirstSQLResult(mSQL,aBO.ObjectSpace);

mStoreBatches:=aBO.GetLoadedCollectionMonikerForFieldCode(aBO.GetFieldCode('DocRowBatches'));

**if** mStoreBatches.Count>0 **then**

**begin**

**for** i:=0 **to** mStoreBatches.Count-1 **do**

**begin**

mStoreBatch:=mStoreBatches.BusinessObject[i];

**if** i>0 **then** result:=result+',';

result:=result+'{"prod\_id":"'+parseStr(mCode)+'","qnty":"'+FloatToStr(mStoreBatch.GetFieldValueAsFloat('UnitQuantity'))+'","lot":"'+mStoreBatch.GetFieldValueAsString('StoreBatch\_ID.Name')+'"}'+#13#10;

inc(aRowsCount);

**end**;

**end**

**else** *// nejsou připojeny žádné šarže, pošlu bez nich*

**begin**

aRowsCount:=1;

result:= '{"prod\_id":"'+parseStr(mCode)+'","qnty":"'+FloatToStr(aBO.GetFieldValueAsFloat('UnitQuantity'))+'"}'+#13#10;

**end**;

**end**;

**end**;

**begin**

**end**.