



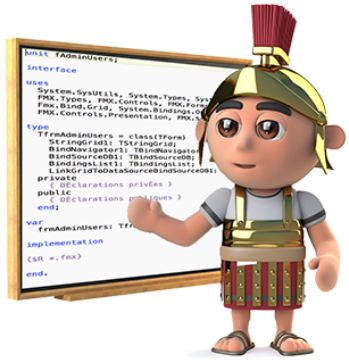


# Au programme

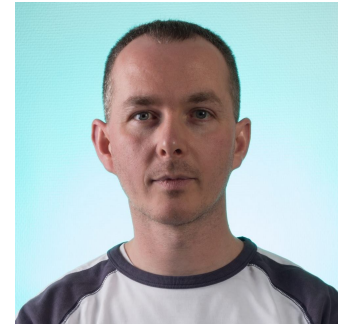
- Introduction
- La problématique
- La solution du jour
- Conception et versionage de la base de données
- Conception du module de données
- Conclusion



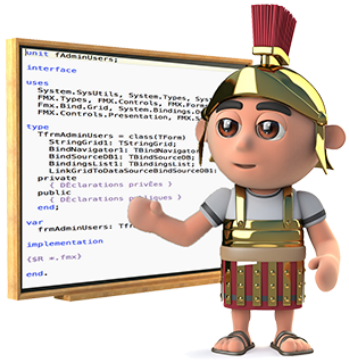
# Présentation



# A propos de Patrick Prémartin



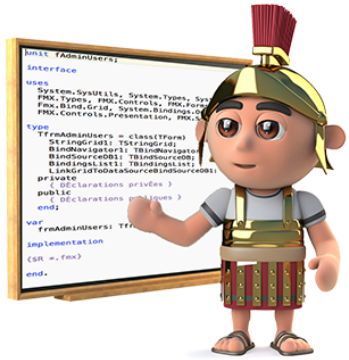
- Ancien utilisateur de Turbo Pascal, amoureux de Delphi depuis toujours, développeur web depuis NCSA Mosaic...
- Prestataire et formateur à travers sa société Olf Software.
- Editeur du blog <https://developpeur-pascal.fr/>
- MVP Embarcadero depuis octobre 2017.
- Auteur d'articles, de livres, webinariste et parfois même conférencier.



## A propos de Olf Software



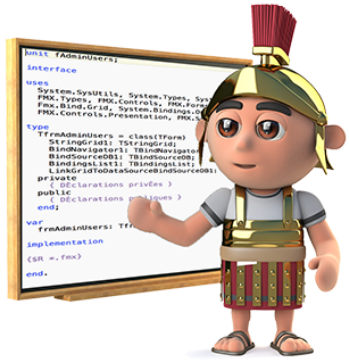
- Depuis octobre 2001 la société de services Olf Software propose des prestations de conseil, développement et formation sur les technologies liées au web et au développement logiciel.
- La société édite également des livres, des sites Internet et des logiciels.
- Site web principal : <https://olfsoftware.fr>



# Formations Delphi

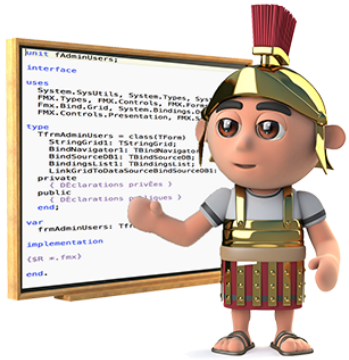


- Olf Software propose des formations en lien avec Delphi.
- Des formations inter entreprises sont organisées régulièrement et annoncées sur <https://se-former-a-delphi.fr/>
- Des formations intra entreprise personnalisées sont faisables sur demande partout en France et en Europe.
- Olf Software est un organisme de formation professionnelle enregistré en préfecture et référencé sur Datadock.  
Nos formations peuvent être prises en charge par les OPCA/OPCO.



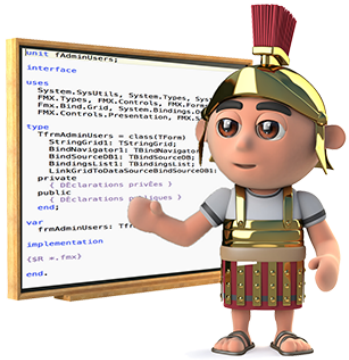
# Les versions de Delphi

- Plusieurs licences de Delphi existent (vendu seul ou dans RAD Studio):
  - Academic : pour les établissements d'enseignement et leurs étudiants
  - Community Edition : gratuit pour tous, en licence commerciale limitée
  - Professional, Enterprise, Architect : pour les entreprises, selon leurs besoins
- Pour toutes informations sur les licences et les tarifs contactez la société Barnsten sur <https://www.barnsten.com/fr/>



# Les webinaires précédents

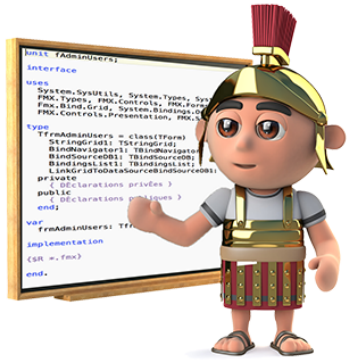
- Je participe régulièrement à des webinaires. Une page leur est consacrée sur mon blog à l'adresse <https://vasur.fr/webinairesdelphi>
- Vous y trouverez les rediffusions en vidéos lorsqu'elles sont disponibles, un PDF de mes présentations, le lien vers les sources des exemples présentés et des informations complémentaires.
- S'y trouvent aussi les dates et liens d'inscriptions pour les prochains webinaires, conférences et formations.



## En bref

- Présentation de la société Olf Software  
<https://www.olfsoftware.fr>
- Présentation de Patrick Prémartin  
<https://www.linkedin.com/in/patrickpremartin/>
- Les différentes versions de Delphi  
<https://www.embarcadero.com/fr/products/delphi>
- Voir ou revoir les webinaires précédents  
<https://vasur.fr/webinairesdelphi>

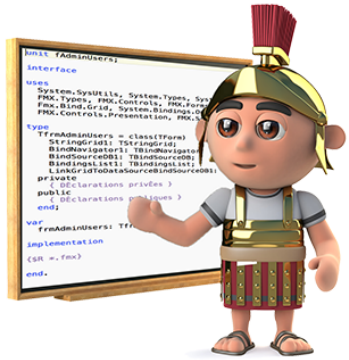




# Introduction

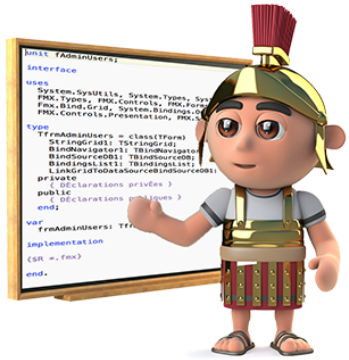
- Delphi et C++Builder permettent de faire toutes sortes de logiciels pour les environnements Windows, macOS, Android, iOS et Linux<sup>(\*)</sup>.
- Un outil de déploiement est intégré à RAD Studio pour créer des logiciels d'installation ou les exécutables à envoyer sur les différents magasins d'applications ou directement sur les appareils visés.
- Grâce au PAServer ([voir le webinaire du 5 novembre](#)) et à l'outil de déploiement on peut diffuser des fichiers complémentaires au programme exécutable.

(\*) Linux uniquement sur Delphi pour le moment



# Introduction

- La plupart des logiciels ont besoin de stocker des données. De nombreuses solutions existent et sont exploitables très simplement sous Delphi :
  - Fichiers textes (Tinfile, TextFile, TXMLDocument),
  - Fichiers binaires (File, File of Record),
  - Bases de données client/serveur (dbExpress, FireDAC, IBX),
  - Bases de données embarquées (SQLite, IBLite, IB ToGo, Firebird Embedded, MySQL Embedded)
  - Liste loin d'être exhaustive tant il y a de solutions dans le commerce et en open source pour gérer des données avec Delphi.



# Introduction

- Lorsqu'on veut travailler en SQL on peut utiliser les moteurs habituels pour accéder à de « vraies bases » SQL comme on peut passer par FireDAC pour travailler avec des API REST en SQL à l'aide des drivers CDATA désormais fournis avec les licences Enterprise et Architect ou disponibles à la vente pour les autres versions de Delphi.
- Tout ça est très bien lorsqu'on a un réseau local, une connexion Internet, un serveur sur lequel mettre le moteur de bases de données mais ce n'est pas toujours le cas, surtout pour les applications grand public.

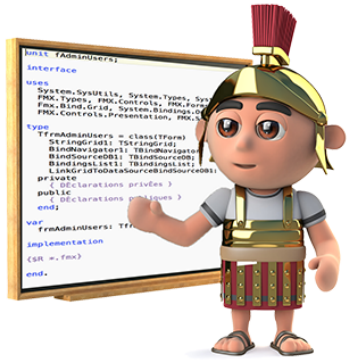


# Introduction

- La solution est donc d'embarquer des bases de données directement avec le programme, qu'il soit en mode desktop ou mobile.

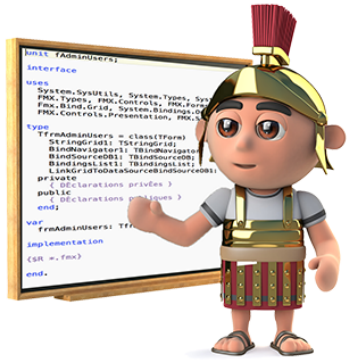


# La problématique



# La problématique

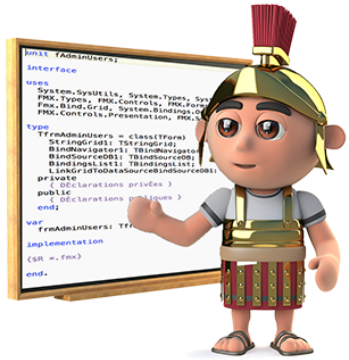
- Embarquer une base de données avec un programme c'est sympa, mais que se passe-t-il lorsqu'on fait des mises à jour des données côté éditeur, des mises à jours de données côté utilisateur et des changements dans la structure de la base de données ?
- Quand les données ne sont modifiées que par l'éditeur, la base embarquée est un fichier classique et peut écraser la version disponible chez l'utilisateur.
- Dans les autres cas, c'est compliqué...



# La problématique

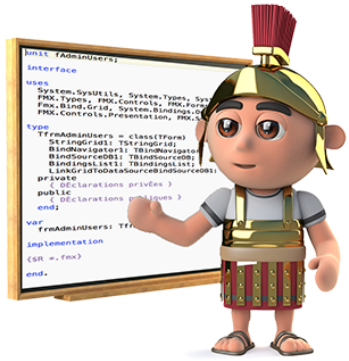
- ... ou ça ne l'est pas, à vous de voir !



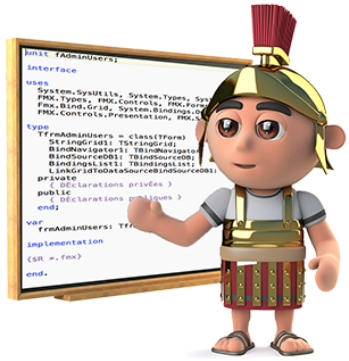


# La problématique

- Pour la suite une connaissance basique de FireDAC et de son architecture est utile. Vous pouvez regarder les webinaires suivants :
  - [Utilisation de FireDAC avec la VCL](#)
  - [Utilisation de FireDAC avec FMX](#)
- Vous pouvez également regarder le webinaire que nous avons consacré aux solutions de synchronisation de données :
  - [Des solutions de synchronisation de données](#)

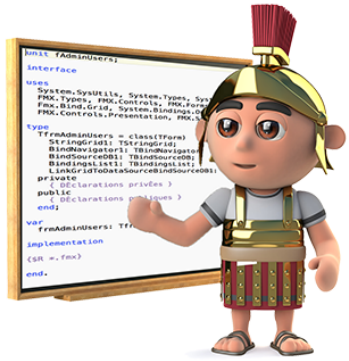


# La solution du jour



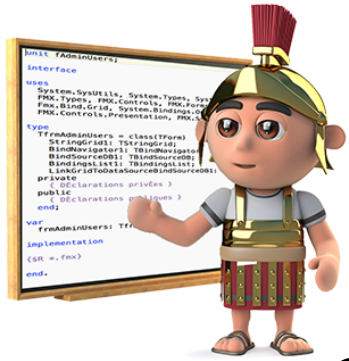
# La solution du jour

- La solution que je vous propose aujourd'hui permet de résoudre les points suivants :
  - Avoir une base de données locale sur toutes les plateformes
  - La maintenance de la structure de la base de données
  - Son utilisation dans un projet



## La solution du jour

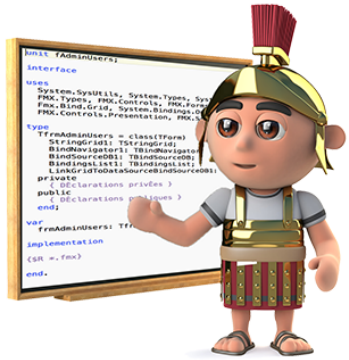
- Nous allons utiliser une base de données SQLite et un projet FireMonkey.
- Ca fonctionnerait aussi pour un projet VCL, IntraWeb ou n'importe quel autre type de projet sur lequel les composants FireDAC fonctionnent.
- Je fais la démonstration sur Delphi mais c'est aussi valable pour les utilisateurs de C++Builder sous réserve d'adapter le code source.



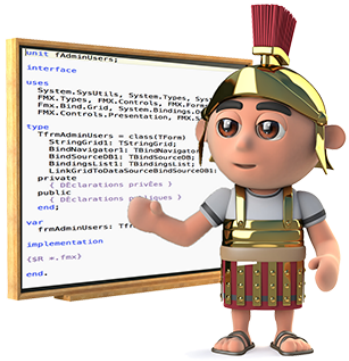
```
Public Interface IAdminUsers
Interface
Uses
System.Xml.Linq, System.Type, System.Linq
PFX.Type, PFX.Controls, PFX.Presentation
PFX.BindGrid, System.Collections.Generic
PFX.Controls.Presentation, PFX

Type
TAdminUsers = class(Fox)
String() As String()
BindGrid() As TBindGrid
BindSource() As TBindSource
LinkGridToDataSource() As TBindSource
Private
(Declarations omises)
Public
(Declarations omises)
End
End
Var
TAdminUsers: TAdminUsers
Implementation
(SCR = Fox)
End
```

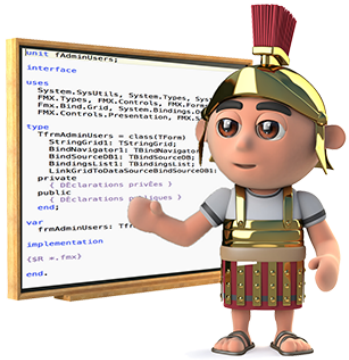
# Conception et versionage de la base de données



- Pour la création de la base de données j'utilise TMS Data Modeler de TMS Software.
- C'est un logiciel payant, disponible dans la licence TMS All Access ou en achat direct.
- Il permet de dessiner des MCD, gérer les relations entre les tables et d'une manière générale faire tout ce qu'on fait pour créer des bases de données.
- Il gère plusieurs moteurs de bases SQL et permet d'exporter les requêtes de création et de modification.



- J'ai choisi TMS Data Modeler car il est simple d'utilisation et que je l'ai avec ma licence All Access.
- D'autres logiciels ayant les mêmes fonctionnalités existent sur le marché et en open source.
- Les détenteurs de licences Architect de Delphi, C++Builder et RAD Studio bénéficient d'une licence pour [Aqua Data Studio](#) qui va au delà de la simple conception de base de données.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Test
{
    public interface IAdminUsers
    {
        void AddAdminUsers();
        void DeleteAdminUsers();
        void UpdateAdminUsers();
        void GetAdminUsers();
    }

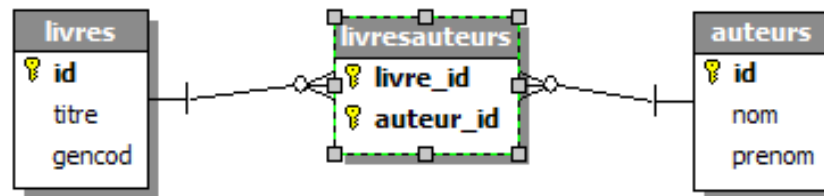
    public class AdminUsers : IAdminUsers
    {
        public void AddAdminUsers()
        {
            // TODO: Add Admin Users
        }

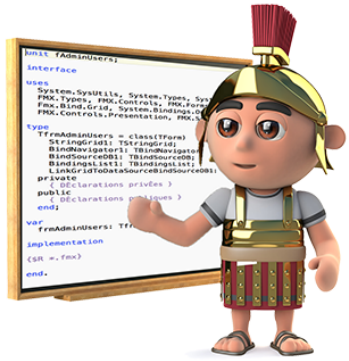
        public void DeleteAdminUsers()
        {
            // TODO: Delete Admin Users
        }

        public void UpdateAdminUsers()
        {
            // TODO: Update Admin Users
        }

        public void GetAdminUsers()
        {
            // TODO: Get Admin Users
        }
    }
}
```

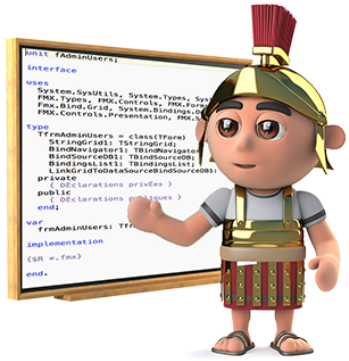
- Pour la démo je crée une base SQLite avec deux tables : *livres* et *auteurs*.
- Les *livres* ont un titre, un GENCOD, un ou plusieurs auteurs.
- Les *auteurs* ont un nom et un prénom.



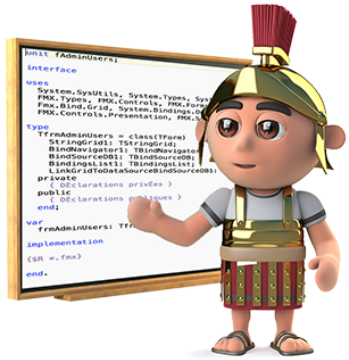


- Une fois la base créée on exporte les requêtes SQL permettant de la créer et on l'archive dans TMS Data Modeler.

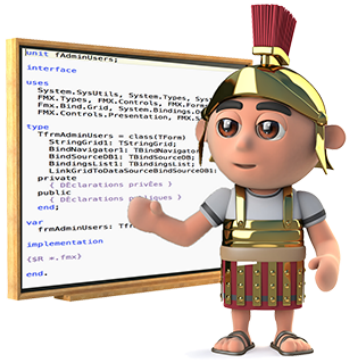




# Conception du module de données pour le projet

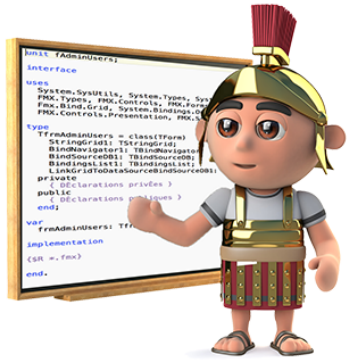


- Côté Delphi j'utilise un TDataModule pour gérer ma base de données.
- Il nous faut un composant TFDConnexion pour gérer la base et ses fonctionnalités. Il utilisera un driver SQLite.
- Pensez à ajouter le driver SQLite à vos projets depuis l'outil de déploiement puisqu'il est désormais à lier aux programmes sur macOS. (pour des histoires de sandbox)
- En onBeforeConnect je paramètre ma connexion et pointe sur le fichier de la base selon l'OS.

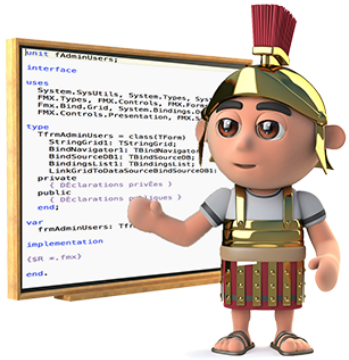


- En onAfterConnect je teste la version en cours de la base de données et la mets à jour si nécessaire.
- Les requêtes de création et de mise à jour sont stockées dans un TFDScrip<sup>(\*)</sup> mais on peut aussi les envoyer manuellement directement depuis le source par des appels à TFDConnexion.ExecSQL().

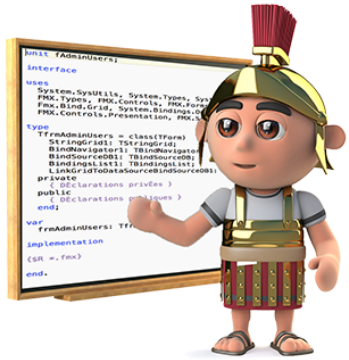
(\*) TFDScrip était « cassé » en 10.3.2 Rio, corrigé en 10.3.3 Rio



# Conclusion

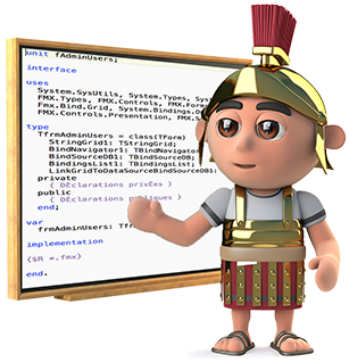


- Cette technique fonctionne bien entendu pour tout type de bases de données SQL.
- Pensez à faire des backups des bases de données locales surtout avant d'en modifier la structure.
- Si vous le pouvez proposez une synchronisation sur un service de cloud ou d'exporter les données pour que vos utilisateurs ne perdent rien en cas de crash ou d'anomalie lors d'une mise à jour de la base.



# Conclusion

- Comme pour les webinaires précédents, ce PDF et l'éventuel replay seront mis à disposition sur le blog <https://developpeur-pascal.fr>
- Prochain événement en live :  
=> rencontres développeurs / meetups, conférences, présentations sur demandes, contactez-nous si vous êtes intéressés.
- Prochain webinaire le mardi 28 janvier 2019 consacré à l'utilisation du SVG sous Delphi.



# Questions / réponses



(c) Patrick Prémartin / [developpeur-pascal.fr](http://developpeur-pascal.fr)