

Analyse du Code Original SCTL

Vue d'ensemble

Ce document analyse le code source original de l'application SCTL (Société Civile des Terres du Larzac), composée de deux projets :

- **Sctl** : Application C++ Builder pour la gestion des actionnaires et des parts sociales
- **D3Dev/Ghis** : Application Delphi pour la gestion du cadastre et des fermages

--

1. Structures de Données

1.1 Constantes Fondamentales

```
```cpp
// Main.h - Types de structures juridiques
#define TYPE_GFA 2 // Groupement Foncier Agricole
#define TYPE_ASSOC 5 // Association
#define TYPE_TSL 6 // Terres Solidaires du Larzac
#define ALL_TYPES 0 // Tous les types (filtre désactivé)
```

## 1.2 Entités Principales

### 1.2.1 Actionnaire (Personne)

Champ	Type	Description
IdPersonne	int	Identifiant unique
Civilite	String	M., Mme, etc.
Nom	String	Nom de famille
Prenom	String	Prénom
Adresse	String	Adresse ligne 1
Adresse2	String	Adresse ligne 2
CodePostal	String	Code postal
Ville	String	Ville
Tel	String	Téléphone fixe
Port	String	Téléphone portable
Fax	String	Numéro de fax
Mail	String	Adresse email
Comment	String	Commentaires libres
Divers	String	Informations diverses

Attributs booléens d'état :

Attribut	Signification
----------	---------------

Npai	N'habite Pas à l'Adresse Indiquée
Decede	Personne décédée
CR	Courrier Retourné
PasConvocAG	Ne pas convoquer aux AG
PasConvoqTSL	Ne pas convoquer TSL
Termine	Dossier terminé (soldé)
Fondateur	Membre fondateur
DeDroit	Membre de droit
Adherent	Adhérent
MisDoffice	Mis d'office
EstPersonneMorale	Personne morale (vs physique)
DcdNotarie	Décès notarié
Apport	A fait un apport
CNI	Carte d'identité fournie

## 1.2.2 Numéro de Part

Champ	Type	Description
IdNumPart	int	Identifiant unique
NumPart	int	Numéro de la part (1, 2, 3...)
IdActionnaire	int	Propriétaire actuel
IdMouvement	int	Mouvement associé
Termine	bool	Part terminée/soldée
Distribue	bool	Part distribuée
Etat	int	Code d'état de la part
IdGfa	int	Structure GFA (IDs 11-14 uniquement)
IdTSLouAssoc	int	Structure TSL ou Association (autres IDs)

**IMPORTANT - Distinction IdGfa vs IdTSLouAssoc :**

La table NumeroParts utilise deux champs distincts pour le lien avec les structures juridiques :

```
// DataBase.cpp - Logique de sauvegarde des numéros de parts
// Source: old-code/Sctl/DataBase.~cp lignes 1180-1192
```

```
// si IdGfa n'est pas compris entre 11 et 14 (Id des gfa)
// il s'agit de l'identifiant d'une nouvelle structure
if((AldGfa >= 11) && (AldGfa <= 14))
{
 hIdGfa = AldGfa; // Stocker dans IdGfa
 hIdTSLouAssoc = 0; // Laisser IdTSLouAssoc vide
```

```

 hNomChamp = "NumeroPartGfa";
}
else
{
 hIdGfa = 0; // Laisser IdGfa vide
 hIdTSLouAssoc = IdGfa; // Stocker dans IdTSLouAssoc
}

```

Champ	IDs valides	Type de structure
IdGfa	11, 12, 13, 14	Les 4 GFA originaux (GFA1, GFA2, GFA3, GFA4)
IdTSLouAssoc	39-42 (Assoc), 43-46 (TSL), autres	TSL et Associations

**Implications pour la migration :** Lors de la migration vers PostgreSQL, il faut utiliser IdTSLouAssoc comme source principale pour le lien avec les structures, avec IdGfa en fallback pour les rares cas où les IDs 11-14 sont utilisés

### 1.2.3 Mouvement

Champ	Type	Description
IdMouvement	int	Identifiant unique
IdActionnaire	int	Actionnaire concerné
IdActe	int	Acte juridique associé
DateOperation	DateTime	Date de l'opération
Sens	bool	true = acquisition (+), false = cession (-)
NbParts	int	Nombre de parts concernées
IdTypeApport	int	Type d'apport
IdTypeRembourse	int	Type de remboursement

**Visualisation du sens :**

- ImgPlus : Icône acquisition (+)
- ImgMoins : Icône cession (-)

### 1.2.4 Acte

Champ	Type	Description
IdActe	int	Identifiant unique
CodeActe	String	Code court (ex: "AGE2024")
DateActe	DateTime	Date de l'acte
LibelleActe	String	Description
IdStructure	int	Structure juridique

Provisoire	bool	Acte provisoire
------------	------	-----------------

## 1.2.5 Structure Juridique

Champ	Type	Description
IdStructure	int	Identifiant unique
NomStructure	String	Nom (GFA, TSL, Association)
TypeStructure	int	TYPE_GFA, TYPE_ASSOC, TYPE_TSL
Gfa	String	Code GFA associé

---

## 1.3 Entités Cadastre/Fermage (D3Dev/Ghis)

### 1.3.1 Parcelle

```
// UCalculs.pas - Colonnes grille cadastre
TmColonne = (
 mColNone, mColLig, mColNumCom, mColNomCom, mColParcelle,
 mColComment, mColGfa, mColLieuDit, mColDivision, mColSubdivision,
 mColTypeCad, mColClassCad, mColHectare, mColRevenu, mColExploitant,
 mColSctl, mColSurface,
 mColDebBailGfa, mColFinBailGfa, mColNumAveGfa, mColDateAveGfa,
 mColValPointGfa, mColSuppDureeGfa,
 mColDebBailSctl, mColFinBailSctl, mColNumAveSctl, mColDateAveSctl,
 mColValPointSctl, mColSuppDureeSctl,
 mColIdSubdivision, mColIdParcelle, mColIdTypeCad, mColIdClassCad,
 mColIdCommune, mColIdGfa, mColIdLieuDit, mColIdExploitant, mColAdresse
);
```

Champ	Description
IdParcelle	Identifiant unique
NumCom	Numéro de commune
NomCom	Nom de la commune
Parcelle	Référence cadastrale
LieuDit	Lieu-dit
Division	Division cadastrale
Subdivision	Subdivision
TypeCad	Type cadastral
ClassCad	Classe cadastrale
Hectare	Surface en hectares
Surface	Surface en m <sup>2</sup>
Revenu	Revenu cadastral

Exploitant	Exploitant actuel
Sctl	Appartient à la SCTL

#### Champs bail GFA :

- DebBailGfa : Début du bail GFA
- FinBailGfa : Fin du bail GFA
- NumAveGfa : Numéro avenant GFA
- DateAveGfa : Date avenant GFA
- ValPointGfa : Valeur du point GFA
- SuppDureeGfa : Supplément durée GFA

#### Champs bail SCTL :

- DebBailSctl : Début du bail SCTL
- FinBailSctl : Fin du bail SCTL
- NumAveSctl : Numéro avenant SCTL
- DateAveSctl : Date avenant SCTL
- ValPointSctl : Valeur du point SCTL
- SuppDureeSctl : Supplément durée SCTL

### 1.3.2 Fermage

```
// UCalculs.pas - Colonnes grille fermage
TfColonne = (
 fColNone, fColLig, fColNumCom, fColNomCom, fColParcelle,
 fColComment, fColGfa, fColLieuDit, fColDivision,
 fColTypeFermage, fColPointFermage, fColDureeFermage,
 fColHectare, fColRevenu, fColMontant,
 fColExploitant, fColSctl, fColEstSCTL, fColSurface, ...
);
```

Champ	Description
TypeFermage	Type de fermage
PointFermage	Nombre de points fermage
DureeFermage	Durée du fermage
Montant	Montant calculé du fermage
EstSCTL	Fermage SCTL (vs GFA)

### 1.3.3 Totaux calculés

```
// Variables globales de totaux
TfColTotaux = (
 fColTotNone, fColTotLig, fColTotTypeFermage, fColTotPointFermage,
 fColTotDureeFermage, fColTotHectare, fColTotRevenu,
 fColTotMontant, fColTotNbParcelles
```

);

## 2. Interfaces Utilisateur

## 2.1 Fenêtre Principale (Main.h - TmMainFrm)

## Application MDI (Multiple Document Interface) avec menu hiérarchique :

## Propriétés de configuration :

- **ListeActesProvisoires** : Afficher les actes provisoires
  - **ListeCessionParts** : Mode cession de parts
  - **ListeAnnulParts** : Mode annulation de parts
  - **ConfirmationSuppression** : Demander confirmation

## 2.2 Fiche Actionnaire (Action.h - TmAction)

## Formulaire principal avec onglets :

<input type="checkbox"/> <b>Fiche Actionnaire</b>		<input type="checkbox"/> <input checked="" type="checkbox"/> <b>[X]</b>
<input type="checkbox"/> <b>[Toolbar: Quitter   Rafraîchir   Imprimer   ...]</b>		
<input type="checkbox"/> <b>Général</b> <input type="checkbox"/> <b>Liste Parts</b> <input type="checkbox"/> <b>Membres PM</b> <input type="checkbox"/> <b>Total Parts</b>		
<input type="checkbox"/> <b>Civilité:</b> <input type="checkbox"/> <b>Nom:</b> <input type="text"/> <b>Prénom:</b> <input type="text"/>		
<input type="checkbox"/> <b>Adresse:</b> <input type="text"/>		
<input type="checkbox"/> <b>CP:</b> <input type="text"/> <b>Ville:</b> <input type="text"/>		
<input type="checkbox"/> <b>Tél:</b> <input type="text"/> <b>Port:</b> <input type="text"/> <b>Fax:</b> <input type="text"/>		





Recherche Multicritères

Critères Résultats Chronologie Anomalies

Nom: [ ] Prénom: [ ] Ville: [ ]

CP: [ ] Adresse: [ ]

Email: [ ] Tél: [ ]

Filtres état

Type membre

NPAI

Décédé

Fondateur

De droit

CR

Rapport

Adhérent

Mis office

Mouvements

Acte: [ ] Type Apport: [ ]

Type Remboursement: [ ]

Numéro de Part

N°: [ ] Terminé

Distribué

Min: [ ] Max: [ ] Type: (•) Tous (•) GFA

Résultats: 42 actionnaires trouvés

Colonnes de résultats :

```
enum NumColCrit {
 ColNone, ColCivil, ColNom, ColPrenom, ColAdresse, ColAdresse2,
 ColCodePost, ColVille, ColTel, ColFax, ColMail, ColComment, ColDivers
};
```

Fonctions de détection d'anomalies :

- ChercherPartsSansMouvements() : Parts orphelines
- ChercherMouvementsSansActes() : Mouvements sans acte
- ChercherNumeroPartsSansActionnaires() : Parts non attribuées
- ChercherMouvementsSansActionnaires() : Mouvements orphelins

## 2.7 Parts Restantes (PartRestantBis.h - TmPartRestantBis)

Synthèse des parts par actionnaire :

Parts Restantes

[Fusion] [Rafraîchir] [Nouveau]        
 Filtres       NPAI       Exclure NPAI       Rapport       Décédé        
 Fondateur       De droit       Mis office       Adhérent        
 Type: (•) Tous       GFA seul       SCTL seul        
 Recherche: [ ]        
 Nom      Prénom      GFA      SCTL      Total      État        
 DUPONT      Jean      3      2      5      Actif        
 MARTIN      Marie      0      4      4      NPAI        
 ...      ...      ...      ...      ...        
 Total: 1250 parts      42 actionnaires     

## 2.8 Cadastre et Fermages (D3Dev/Ghis - UCalculs.pas)

Interface de gestion des parcelles et calcul des fermages :

Cadastre et Fermages          
 Cadastre      Fermages      Totaux        
 Com.      Parcelle      Lieu-dit      Surface      Revenu      Exploit        
 001      A-125      Les Prés      2.5 ha      150 €      FERME A        
 001      B-042      La Combe      1.2 ha      80 €      FERME B        
 ...        
 Surface totale: 125.5 ha      Revenu total: 8 500 €     

## 3. Modes de Calcul

### 3.1 Calcul du Montant de Fermage

Formule principale (UBib.pas) :

```

function Bib_CalculerMontantFermage(
 APointFermage: Double; // Nombre de points fermage

```

```

ASurface: Double; // Surface en m2
ADuree: Double; // Durée (non utilisé dans calcul)
ASctl: Boolean; // true = SCTL, false = GFA
AValeurPointGFA: Double; // Valeur du point GFA
AValeurPointSCTL: Double; // Valeur du point SCTL
ASuppGFA: Boolean; // Supplément durée GFA actif
ASuppSCTL: Boolean // Supplément durée SCTL actif
): Double;
begin
 // Base : (points × surface) / 10000 (conversion m2 → hectares)
 Result := (APointFermage * ASurface) / 10000;

 if ASctl then
 begin
 // Fermage SCTL
 Result := Result * AValeurPointSCTL;
 if ASuppSCTL then
 Result := Result + ((Result * G_ValeurSuppSCTL) / 100);
 end
 else
 begin
 // Fermage GFA
 Result := Result * AValeurPointGFA;
 if ASuppGFA then
 Result := Result + ((Result * G_ValeurSuppGFA) / 100);
 end;
end;

```

**Variables globales :**

- G\_ValeurSuppGFA : Pourcentage de supplément durée GFA
- G\_ValeurSuppSCTL : Pourcentage de supplément durée SCTL

**Étapes du calcul :**

1. Calculer la base : points × surface / 10000
2. Multiplier par la valeur du point (GFA ou SCTL)
3. Si supplément durée actif : ajouter le pourcentage

## 3.2 Conversion de Surface

```

function Bib_ConvSurfaceHectare(ASurface: Double): String;
// Convertit une surface en m2 vers format "ha a ca"
// Exemple: 25432 m2 → "2 ha 54 a 32 ca"

```

```

function Bib_ReconstruireSurface(AHectare, AAre, ACentiare: Integer): Double;
// Reconvertit ha/a/ca vers m2
// Exemple: 2 ha 54 a 32 ca → 25432 m2

```

## 3.3 Calcul du Nombre de Parts

```
// Action.h - TmAction
void __fastcall CalculNbParts(void);
Calcule le nombre total de parts d'un actionnaire en sommant :
```

- Parts en nom propre
- Parts via personnes morales (si applicable)

## 3.4 Totaux des Grilles

```
// Variables de totaux calculés automatiquement
hRevenuTot: Double; // Total des revenus cadastraux
hSurfaceTot: Double; // Total des surfaces
hMontantTot: Double; // Total des montants de fermage
```

## 3.5 Valeur Cadastrale

```
function Bib_ValeurCadastral(
 AlIdCommune: Integer;
 AlIdTypeCad: Integer;
 AlIdClassCad: Integer
): Double;
// Recherche la valeur cadastrale selon commune/type/classe
```

---

# 4. Fonctions Utilitaires

## 4.1 Accès Base de Données (DataBase.h - TmSqlAccess)

### Actionnaires

Fonction	Description
SqlLoadUnActionnaire(id)	Charge un actionnaire par ID
SqlSaveUnActionnaire(stream)	Sauvegarde un actionnaire
SqlDeleteActionnaire(id)	Supprime un actionnaire
SqlLoadListeActionnaires(stream)	Liste tous les actionnaires
SqlNewMouvementDeActionnaire(id, idActe, sens, nbParts)	Crée un mouvement

### Numéros de Parts

Fonction	Description
SqlSaveNumeroParts(stream)	Sauvegarde des numéros
SqlSaveNewNumeroParts(idActionnaire, idMvt, numPart)	Crée un numéro
SqlRetraitNumeroParts(id)	Retire un numéro

SqlMarquerLesPartsCedees(idMvt)	Marque les parts cédées
SqlVerificationChampDistribue()	Vérifie les distributions
SqlVerificationChampTermine()	Vérifie les terminaisons

## Cessions

Fonction	Description
SqlCreerLesNumPartsCedeesSensMoins(params)	Cession côté cédant
SqlCreerLesNumPartsCedeesSensPlus(params)	Cession côté cessionnaire
SqlRestitutionAuGfa(idPersonne, idMvt)	Restitution au GFA

## Actes et Structures

Fonction	Description
SqlLoadActes(stream, type)	Liste les actes par type
SqlLoadStructures(stream)	Liste les structures
SqlLoadTypeApport(stream)	Types d'apport
SqlLoadTypeRembourse(stream)	Types de remboursement

## 4.2 Bibliothèque Utilitaire (UBib.pas - Bib\_\*)

### Chargement ComboBox

Fonction	Description
Bib_LoadTypeCadastre(combo)	Charge les types cadastre
Bib_LoadTypeFermage(combo)	Charge les types fermage
Bib_LoadCommunes(combo)	Charge les communes
Bib_LoadGfa(combo)	Charge les GFA
Bib_LoadExploitants(combo)	Charge les exploitants
Bib_LoadValeurPoints(combo)	Charge les valeurs points
Bib_LoadLieuDit(combo, idCommune)	Lieux-dits par commune

### Validation

Fonction	Description
Bib_VerifChampGrille(grid, col, row)	Vérifie cellule grille
Bib_VerifChampText(edit)	Vérifie champ texte
Bib_VerifChampCombo(combo)	Vérifie ComboBox

## Rapports

Fonction	Description
Bib_AfficherRapport(params)	Génère un rapport
Bib_AfficherRecapitulatif(params)	Affiche récapitulatif

## Recalcul

Fonction	Description
Bib_RecalculerCadastre()	Recalcule cadastre
Bib_RecalculerFermages()	Recalcule fermages

## 4.3 Sérialisation (mStream.h - TmStream)

Classe de sérialisation personnalisée pour transfert de données :

```
class TmStream {
 // Lecture
 int ReadInteger();
 String ReadString();
 double ReadDouble();
 bool ReadBool();
 TDateTime ReadDateTime();

 // Écriture
 void WriteInteger(int);
 void WriteString(String);
 void WriteDouble(double);
 void WriteBool(bool);
 void WriteDateTime(TDateTime);

 // Navigation
 void Reset();
 bool Eof();
};
```

## 4.4 Fonctions de Type Structure

```
// PartRestantBis.h
bool __fastcall EstDeTypeAssoc(int AldStruct); // Est une association ?
bool __fastcall EstDeTypeTSL(int AldStruct); // Est TSL ?
```

## 4.5 Gestion des Couleurs (Grilles)

Les grilles utilisent un code couleur pour les états :

- **Rouge** : NPAI ou Décédé
- **Bleu** : Fondateur
- **Vert** : Adhérent
- **Orange** : Mis d'office
- **Gris** : Terminé

```
void GererCouleurColonneNom(
 int ARow,
 bool AFondateur,
 bool ADeDroit,
 bool AAdherent,
 bool AMisOffice,
 bool ANpai,
 bool ADcd
);
```

---

## 5. Base de Données

### 5.1 Type et Format

Les données sont stockées dans des bases **Microsoft Access** (fichiers .mdb), et non Paradox comme pourrait le suggérer l'utilisation du BDE. L'accès se fait via le **BDE (Borland Database Engine)** avec le pilote DAO/ODBC pour Access.

### 5.2 Configuration de Connexion

```
// Main.cpp - Initialisation de la connexion
mSqlAccess = new TmSqlAccess;
mSqlAccess->AliasDataBase = "TerresSolidairesLarzac";
mDB = mSqlAccess;
```

L'alias BDE TerresSolidairesLarzac est configuré dans l'administrateur BDE pour pointer vers le fichier .mdb correspondant.

### 5.3 Localisation des Fichiers

#### Bases principales (en production)

Projet	Base	Chemin	Contenu
Sctl/	TSL.mdb	Sctl/Bases/TSL.mdb	Actionnaires, parts, mouvements, actes
D3Dev/ Ghis/	Sctl- Gfa.mdb	D3Dev/Ghis/Base/Sctl- Gfa.mdb	Cadastre, fermages, parcelles

## Autres bases identifiées

Fichier	Type	Description
Sctl/Bases/MaBaseTSL.mdb	Travail	Copie de travail locale
Sctl/Bases/TSL-30-03-17.mdb	Sauvegarde	Backup daté
Sctl/Bases/base christine/TSL.mdb	Utilisateur	Copie utilisateur spécifique
D3Dev/Ghis/Base/BaseCouranteSctl-Gfa.mdb	Production	Base courante cadastre
D3Dev/Ghis/Base/SauvegardeSctl-Gfa.mdb	Sauvegarde	Backup cadastre
D3Dev/LumDelLarzac/Base/LumDelLarzac.mdb	Annexe	Application Lum Del Larzac
D3Dev/Schiste/Base/Schiste.mdb	Annexe	Application Schiste

## Fichiers à ignorer (sauvegardes/copies)

Les fichiers contenant ces motifs dans leur nom sont des sauvegardes ou copies de travail :

- Copie de, Copie-, - Copie
- Sauvegarde, AvantModif, ApresModif
- Dates (ex: 26-03-2012, 070217)
- Modele, ModeleTravail, DeBase
- Vide

## 5.4 Tables Principales

### Base TSL.mdb (Actionnaires/Parts)

Table	Description
Personnes	Actionnaires (physiques et moraux)
NumeroParts	Numéros de parts sociales
Mouvements	Historique des mouvements
Actes	Actes juridiques
Libelle	Structures juridiques (GFA, TSL, Association)
TypeApport	Types d'apport
TypeRemboursement	Types de remboursement
Couleurs	Configuration des couleurs UI
Config	Paramètres de l'application
TableLog	Journal des opérations

### Base Sctl-Gfa.mdb (Cadastre/Fermages)

Table	Description
Parcelles	Données cadastrales
Communes	Référentiel des communes
LieuDit	Lieux-dits par commune
Exploitants	Exploitants agricoles
TypeCadastre	Types cadastraux
ClasseCadastre	Classes cadastrales
TypeFermage	Types de fermage
ValeurPoints	Valeurs des points fermage

## 5.5 Outils pour Lire les Fichiers .mdb sur macOS

## **mdbtools (ligne de commande)**

```
Installation
brew install mdbtools
```

```
Lister les tables
mdb-tables Sctl/Bases/TSL.mdb
```

```
Exporter une table en CSV
mdb-export Sctl/Bases/TSL.mdb Personnes > personnes.csv
```

# Voir le schéma complet  
mdb-schema Sctl/Bases/TSL.mdb

```
Exporter toutes les tables
for table in $(mdb-tables -1 Sctl/Bases/TSL.mdb); do
 mdb-export Sctl/Bases/TSL.mdb "$table" > "${table}.csv"
done
```

## Applications graphiques

- **MDB Viewer Plus** (App Store) - Visualisation simple
  - **LibreOffice Base** - Édition complète
  - **DBeaver** - Client SQL universel

## 6. Flux de Données Principal

Interface TMSqlAccess BDE Utilisateur (DataBase.h) (TQuery)

## 7. Points d'Attention pour Migration

## 6.1 Logique Métier Critique

1. **Calcul de fermage** : Formule complexe avec suppléments
  2. **Gestion des cessions** : Double mouvement (+ et -)
  3. **Numérotation des parts** : Unicité et traçabilité
  4. **Types de structure** : GFA vs SCTL vs Association

## 6.2 Contraintes d'Intégrité

- Un numéro de part appartient à un seul actionnaire
  - Tout mouvement doit être lié à un acte
  - Les cessions créent deux mouvements symétriques
  - La somme des parts doit rester constante

## 6.3 États et Transitions

Actif

Terminé

Décédé

NPAI

## Annexe : Composants Tiers Utilisés

Composant	Fournisseur	Usage
TB97	Jordan Russell	Barres d'outils

TStringGrid	Borland VCL	Grilles de données
TtsGrid	Third-party	Grilles avancées
QuickReport	QSoftware	Rapports
BDE	Borland	Accès base Paradox