



Sommaire

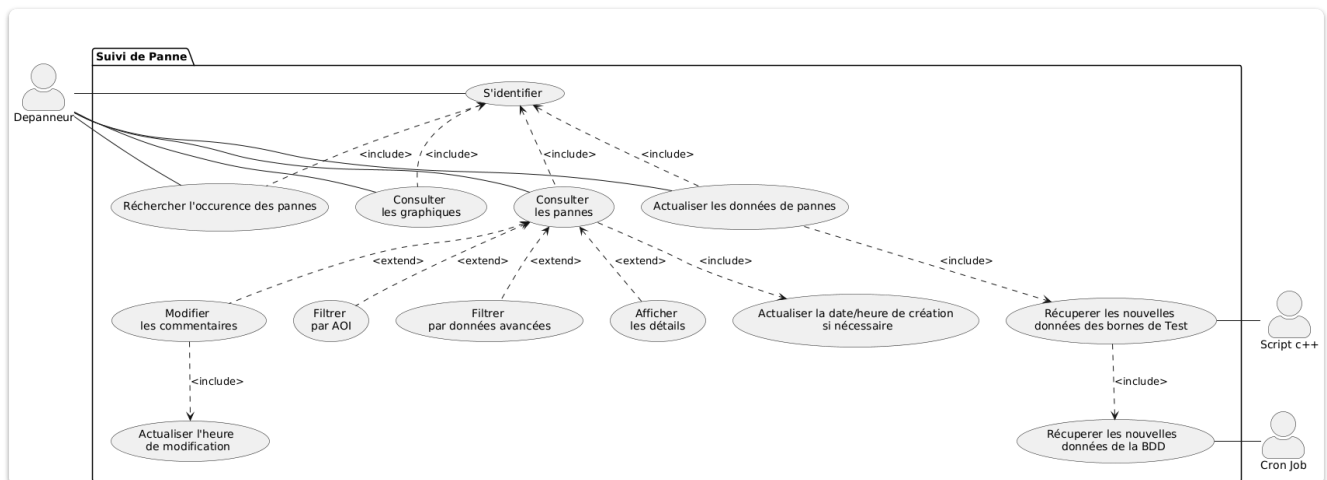
- ☐ [Présentation du Projet](#)
 - [README](#)
 - [Objectifs](#)
 - [Technologies utilisées](#)
 - [Structure du Projet](#)
- ☐ [Le Script C++](#)
 - ☐ [Formater](#)
 - [librairie](#)
 - [Scrapping des tickets](#)
 - ☐ [Formatage XML](#)
 - [logique xml](#)
 - [Résultat](#)
 - ☐ [Transfert SFTP](#)
 - [librairie](#)
 - [vcpkg](#)
 - [.env](#)
- ☐ [BDD / Cron Job](#)
 - ☐ [BDD](#)
 - [Création table](#)
 - [Diagramme de Relation](#)
 - ☐ [Cron Job](#)
 - [Récupération du fichier](#)
 - [Transfert à la base de données](#)
- ☐ [AS400 / Webix](#)
 - [AS400](#)
 - ☐ [Webix](#)
 - [Filtre](#)
 - [Affichage Test Precedent](#)

Présentation du Projet

Résumer

Le projet **Suivi de Pannes** vise à optimiser l'analyse des pannes au dépannage des cartes. Grâce à une interface web intuitive et des

outils d'analyse avancés, les dépanneurs peuvent mieux identifier les récurrences et améliorer le bon du premier coup.



README

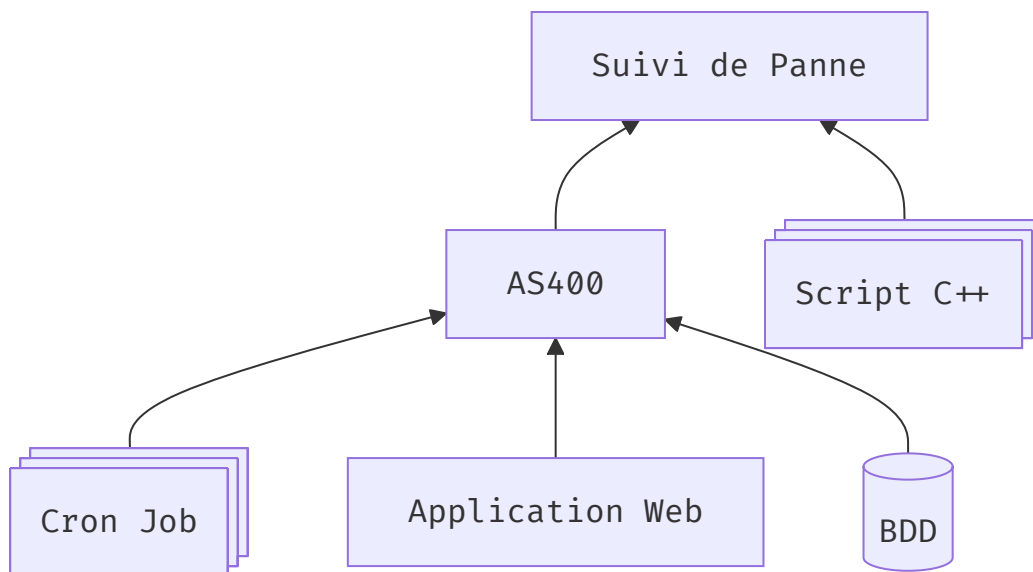
Objectifs

- Améliorer la lecture des pannes grâce à un tableau filtrable et des graphiques interactifs.
- Réduire la récurrence des pannes en facilitant l'analyse des tendances.
- Augmenter l'efficacité des interventions avec un meilleur diagnostic.
- Étendre l'outil sur l'ensemble des sites de production

Technologies utilisées

- **QT Creator** [IDE]
 - C++
 - XML [librairie interne Qt]
 - QTXml
 - QXmlStreamWriter
 - FTP [librairie open source]
 - CURL
 - OpenSSL
 - libssh2
 - vcpkg [gestionnaire de librairie]
- **IBM Db2** [BDD]
- **Webix** [Dashboard]

Structure du Projet



Le Script C++

main.cpp

```
#include <QCoreApplication>
#include <QFile>
#include <QDate>
#include "Recherche_Fichier.h"
#include "client_sftp.h"
#include <iostream>

#include <fstream>
#include <cstdlib>

#ifdef _WIN32
#define setenv(name, value, overwrite) _putenv_s(name, value)
#endif
void loadEnvFile(const std::string& filename);

int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);
    // ...

    //recherche log du jour par Poste
    QStringList Poste(QStringList() << "VP1N" << "VP2N" << "VP3N" << "VP4\\L4832V3");

    Date_jour = "2024-07-01";
    for(int i=0; i < Poste.size(); i++){

        if(QFile::exists("\\\\172.16.5.200\\L4832cavp_res$\\RESULTAT\\" + Poste.at(i) + "\\" + Date_jour + " L4832.txt")){
            qDebug() << "Fichier Existe, date: " << Date_jour;
            Recherche_Fichier Fichier("\\\\172.16.5.200\\L4832cavp_res$\\RESULTAT\\" +
                                     Poste.at(i) + "\\" + Date_jour + " L4832.txt",Poste.at(i));
            Fichier.rf_start();
        }
    }

    // Transfer SFTP
    loadEnvFile("../.env");
    // Lire une variable d'environnement
    const char* secretKey = std::getenv("FTP_IDPASS");
    if (secretKey) {
        std::cout << "Cle secrète chargee avec succes !" << std::endl;
    } else {
        std::cerr << "Erreur : FTP_IDPASS non définie !" << std::endl;
    }

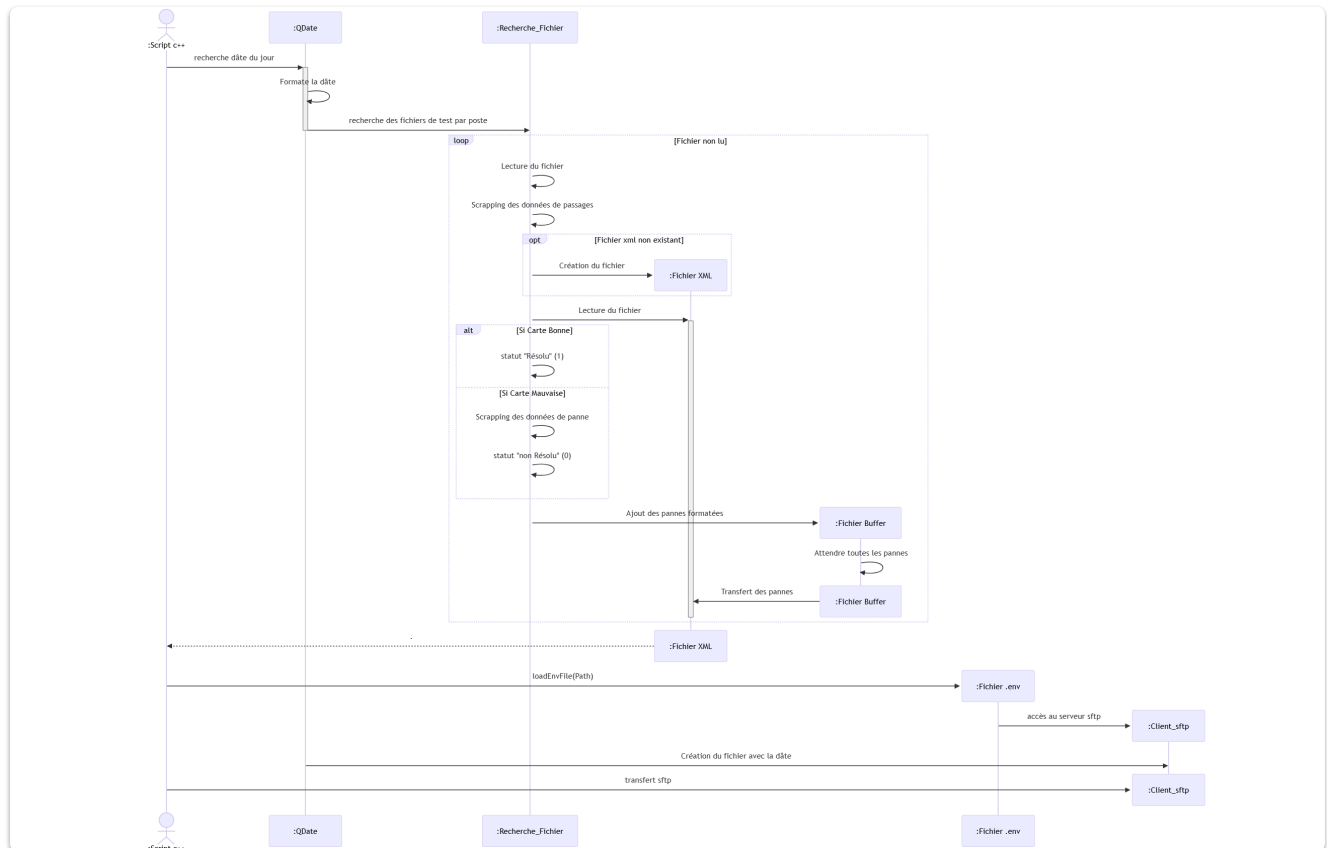
    std::string strUrl = "sftp://" + std::string(secretKey) +
        "@as400.chauvin-arnoux.com/www/phpservdev/htdocs/Webix/TFONTAINE/Webix/Tmp/SuiviPanne_" + Date_jour.toStdString() + ".xml";
    Client_SFTP as400(strUrl.data());
    as400.uploadFile("G:/production/depannage/Suivi_L4832/Depannage.xml");

    //return a.exec();
}
```

Formater

Résumer

Le script doit modifier la donnée source (ticket de dépannage .txt produit par chacun des postes de test) pour ne récupérer que les infos essentiels et les codes liés à la carte et rassembler ces dernières dans un fichier xml journalier.



librairie

- QDomStreamWriter
- QtXml/QDomDocument
- QTextStream

Scrapping des tickets

recherche_fichier.cpp

```
recherche_fichier.cpp

void Recherche_Fichier::rf_start(){
    QString texte;
    if(fichier.open(QIODevice::ReadOnly | QIODevice::Text)){ // Lecture fichier
        QTextStream stream(&fichier);
        stream.setEncoding(QStringConverter::Latin1);
        texte = stream.readAll();
        fichier.close();
    }
    else{ // erreur d'ouverture
        qDebug() << "Ouverture impossible de" << Chemin;
        return;
    }
    Text = texte.split("\n");

    for(int i=0; i < Text.size()-1; i++){
        //qDebug() << Text.at(i);
        if(Text.at(i).indexOf("FIC") > -1){ // Recup IFIC et FIC
            FIC = Text.at(i).mid(Text.at(i).indexOf("FIC"),9);
            Ind_FIC = Text.at(i).mid(Text.at(i).indexOf("FIC")+14,2);
        }

        if(Text.at(i).indexOf("$M") > -1){ // Carte Mauvaise
            recup_infoCarte(0, Text.at(i), QStringList() << Text.at(i+1) <<
Text.at(i+2) << Text.at(i+3) << Text.at(i+4));
        }
        else if(Text.at(i).indexOf("$B") > -1){ // Carte Bonne
            recup_infoCarte(1, Text.at(i));
        }
    }
}
```

```

void Recherche_Fichier::recup_infoCarte(bool statut, QString Texte, QStringList Panne) {
    // Scrap Log
    OF = Texte.mid(4,8);
    if(OF.indexOf("\t") > -1) return;

    if(Text.at(1).indexOf("3.2") > -1){
        Code_AOI = Texte.mid(23,7);
        Date_Heure = Texte.mid(31,19);
    }
    else{
        Code_AOI = Texte.mid(13,7);
        Date_Heure = Texte.mid(21,19);
    }
    if(Code_AOI.indexOf("\t") > -1) return;

    Code_Article = Recuperation_Code_Article_Recursive(Code_AOI, currentYear + 0.5);

    // Recup: Panne | Def Panne | Commentaire panne | Mesure | Lim Min/Max
    if( statut == 0 && Panne.at(0).left(2) != "♦♦"){
        Code_Panne = Panne.at(0).left(Panne.at(0).indexOf(" "));
        Definition_panne = Panne.at(0).mid(Panne.at(0).indexOf(" -")+3);
        if(Panne.at(1).left(2) != "♦♦"){ // Commentaire Panne | Mesure
            if(Panne.at(1).indexOf("Mesure : ") == -1){
                if(Panne.at(1).left(2) != "♦♦"){
                    Commentaire_panne = Panne.at(1);
                }
            }
            else{
                mesure = Panne.at(1).mid(Panne.at(1).indexOf(" :")+3,Panne.at(1).mid(
                    Panne.at(1).indexOf(" :")+3).indexOf(" ")).toDouble();
                if(Panne.at(1).mid(Panne.at(1).lastIndexOf(" ") +1,1) == "m"){
                    mesure = mesure * pow(10,-3);
                }
                else{
                    if(Panne.at(1).mid(Panne.at(1).lastIndexOf(" ") +1,1) == "n"){
                        mesure = mesure * pow(10,-9);
                    }
                    else{
                        if(Panne.at(1).mid(Panne.at(1).lastIndexOf(" ") +1,1) == "μ"){
                            mesure = mesure * pow(10,-6);
                        }
                    }
                }
            }
        }
    }
    if(Panne.at(2).left(2) != "♦♦"){ // Lim Min
        if(Panne.at(2).indexOf("Limite mini. : ") > -1){
            limite_mini = Panne.at(2).mid(Panne.at(2).indexOf(" :")+3,Panne.at(2).mid(
                Panne.at(2).indexOf(" :")+3).indexOf(" ")).toDouble();
            if(Panne.at(2).mid(Panne.at(2).lastIndexOf(" ") +1,1) == "m"){
                limite_mini = limite_mini * pow(10,-3);
            }
            else{
                if(Panne.at(2).mid(Panne.at(2).lastIndexOf(" ") +1,1) == "n"){
                    limite_mini = limite_mini * pow(10,-9);
                }
            }
        }
    }
    if(Panne.at(3).left(2) != "♦♦"){ // Lim Max
        if(Panne.at(3).indexOf("Limite maxi. : ") > -1){
            limite_max = Panne.at(3).mid(Panne.at(3).indexOf(" :")+3,Panne.at(3).mid(
                Panne.at(3).indexOf(" :")+3).indexOf(" ")).toDouble();
            if(Panne.at(3).mid(Panne.at(3).lastIndexOf(" ") +1,1) == "m"){
                limite_max = limite_max * pow(10,-3);
            }
            else{
                if(Panne.at(3).mid(Panne.at(3).lastIndexOf(" ") +1,1) == "n"){
                    limite_max = limite_max * pow(10,-9);
                }
            }
        }
    }
}

```

```

    }
}

// remplissage fichier XML
// ...

// Reset variable
// ...
}

```

Formatage XML

format:

- [dateHeure]YYYY-MM-DD hh::mm::ss
- AOI
- statutFlag
- 1 Vert Résolu
- 0 Rouge À dépanner
- 2 Orange Dépanner mais pas repasser
- 3 Bleu Résolu mais sans dépannage

logique xml

Note

Pour correctement lire un fichier xml avec **QDomDocument**, une balise mère contenant les différentes éléments est nécessaire, sauf que la librairie ne dispose pas d'un moyen de correctement insérer un element dans un autre tout en gardant un aspect lisible pour l'humain. Donc j'ai du coder la logique pour.

recherche_fichier.cpp


```

recherche_fichier.cpp

void Recherche_Fichier::recup_infoCarte(bool statut, QString Texte, QStringList Panne) {
    // Scrap Log
    // ...

    // remplissage fichier XML
    fichierXml.open(QIODevice::ReadWrite | QIODevice::Text);

    QFile buffer("G:\\production\\depannage\\Suivi_L4832\\Depannage_buffer.xml");
    buffer.open(QIODevice::WriteOnly | QIODevice::Text);

    QXmlStreamReader reader(&fichierXml);
    QXmlStreamWriter Nouveau_texte(&buffer);
    Nouveau_texte.setAutoFormatting(true);

    bool LogsElementFound = false;
    bool logAdded = false;

    while (!reader.atEnd()) {
        reader.readNext();

        if (reader.isStartElement()) {
            if (reader.name().toString() == "Logs" && !LogsElementFound) {
                // écriture debut doc et Logs
                Nouveau_texte.writeStartDocument();
                Nouveau_texte.writeStartElement("Logs");
                LogsElementFound = true;
            }
            if (reader.name().toString() == "Log") {
                // Réécriture ancienne Log
                Nouveau_texte.writeStartElement("Log");

                reader.readNextStartElement();
                while (reader.name().toString() != "Log") {
                    // Réécriture du contenu des Log
                    // Write the attributes and characters for each element within <Log>
                    Nouveau_texte.writeStartElement(reader.name().toString());
                    for (const QXmlStreamAttribute &attr : reader.attributes()) {
                        Nouveau_texte.writeAttribute(attr.name().toString(), attr.value().toString());
                    }
                    Nouveau_texte.writeCharacters(reader.readElementText());
                    Nouveau_texte.writeEndElement(); // Close the current element

                    reader.readNextStartElement();
                }
                Nouveau_texte.writeEndElement(); // End <Log>
            }
        }
        else if (reader.isEndElement() && reader.name() == "Logs") {
            // Ajout nouvelle Log et fermeture Logs
            if (!logAdded) {
                newLog(
                    Nouveau_texte,
                    Date_Heure,
                    Code_AOI,
                    Code_Article,
                    OF,
                    FIC,
                    Ind_FIC,
                    Code_Panne,
                    Definition_panne,
                    QString::number(mesure),
                    QString::number(limite_mini),
                    QString::number(limite_max),
                    Commentaire_panne,
                    QString::number(statut),
                    VP.replace("\\L4832V3", ""),
                    "VP");
                logAdded = true;
            }

            Nouveau_texte.writeEndElement();
            break; // Exit the loop after finishing the <Logs> element
        }
    }

    if (reader.hasError()) {
        qDebug() << "Error reading XML:" << reader.errorString();
    }
}

```

```

Nouveau_texte.writeEndDocument(); // End the document in the temporary file
fichierXml.close();
buffer.close();

// Replace the original file with the modified one
if (!QFile::remove(cheminXml)) {
    qDebug() << "Error deleting the original file.";
}
if (!QFile::rename("G:\\production\\depannage\\Suivi_L4832\\Depannage_buffer.xml", cheminXml)) {
    qDebug() << "Error renaming the temporary file.";
}
// if (statut = 0) qDebug() << "carte mauvaise, text: " << Definition_panne;

// Reset variable
// ...
}

```

```

void Recherche_Fichier::newLog(QXmlStreamWriter &Nouveau_texte, QString
dateHeure, QString AOI, QString Article, QString of, QString fic, QString
iFIC, QString panne, QString designationPanne, QString mesure, QString
limitMin, QString limitMax, QString commentairePanne, QString composant,
QString stateFlag, QString commentaire, QString poste, QString site, QString
testPrecedent) {

```

```

    Nouveau_texte.writeStartElement("Log");
    Nouveau_texte.writeStartElement("dateHeure");
    Nouveau_texte.writeCharacters(dateHeure.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("AOI");
    Nouveau_texte.writeCharacters(AOI.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("Article");
    Nouveau_texte.writeCharacters(Article.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("OF");
    Nouveau_texte.writeCharacters(of.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("FIC");
    Nouveau_texte.writeCharacters(fic.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("iFIC");
    Nouveau_texte.writeCharacters(iFIC.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("panne");
    Nouveau_texte.writeCharacters(panne.toUtf8());
    Nouveau_texte.writeEndElement();

```

```

    Nouveau_texte.writeStartElement("designationPanne");
    Nouveau_texte.writeCharacters(designationPanne.toUtf8());

```

```

Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("mesure");
Nouveau_texte.writeCharacters(mesure.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("limiteMin");
Nouveau_texte.writeCharacters(limitMin.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("limiteMax");
Nouveau_texte.writeCharacters(limitMax.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("commentaire");
Nouveau_texte.writeCharacters(commentaire.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("poste");
Nouveau_texte.writeCharacters(poste.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("site");
Nouveau_texte.writeCharacters(site.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("commentairePanne");
Nouveau_texte.writeCharacters(commentairePanne.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("composant");
Nouveau_texte.writeCharacters(composant.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("stateFlag");
Nouveau_texte.writeCharacters(stateFlag.toUtf8());
Nouveau_texte.writeEndElement();

Nouveau_texte.writeStartElement("testPrecedent");
Nouveau_texte.writeCharacters(testPrecedent.toUtf8());
Nouveau_texte.writeEndElement();
Nouveau_texte.writeEndElement();
}

```

Résultat

exemple de log...

```

<?xml version="1.0" encoding="UTF-8"?>

<Logs>
  <Log>

```

```

<dateHeure>01/07/2024 07:41:15</dateHeure>
<AOI>31420FW</AOI>
<Article>693608E00</Article>
<OF>01049040</OF>
<FIC>FIC592655</FIC>
<iFIC>9</iFIC>
<panne>0</panne>
<designationPanne>Vérifi.tension appliquée en J22 du poste (Q1 off) :
6V dc ±0.5V</designationPanne>
<mesure>0.014836</mesure>
<limiteMin>-6.5</limiteMin>
<limiteMax>-5.5</limiteMax>
<poste>VP2N</poste>
<site>VP</site>
<commentairePanne></commentairePanne>
<stateFlag>0</stateFlag>
</Log>
</Logs>

```

Transfert SFTP

```

int Client_SFTP::uploadFile( char *localFile) {
    CURLcode result = CURLE_GOT_NOTHING;
    FILE *f = NULL;

    // Check si le fichier existe sur le serv
    if (sftpGetRemoteFileSize() != -1) {    // Append contenu
        curlhandle = curl_easy_init();

        curl_easy_setopt(curlhandle, CURLOPT_URL, remote);
        curl_easy_setopt(curlhandle, CURLOPT_VERBOSE, 1L);

        f = fopen(localFile, "r");
        if(!f) {
            perror(NULL);
            return 0;
        }

        curl_easy_setopt(curlhandle, CURLOPT_UPLOAD, 1L);
        curl_easy_setopt(curlhandle, CURLOPT_READFUNCTION, readfunc);
        curl_easy_setopt(curlhandle, CURLOPT_READDATA, f);
        curl_easy_setopt(curlhandle, CURLOPT_APPEND, 1L);

        curl_easy_perform(curlhandle);
    }
    else {    // Créer fichier et ajouté contenu
        curlhandle = curl_easy_init();

        curl_easy_setopt(curlhandle, CURLOPT_URL, remote);
        curl_easy_setopt(curlhandle, CURLOPT_VERBOSE, 1L);
    }
}

```

```

f = fopen(localFile, "r");
if(!f) {
    perror(NULL);
    return 0;
}

curl_easy_setopt(curlhandle, CURLOPT_UPLOAD, 1L);
curl_easy_setopt(curlhandle, CURLOPT_READFUNCTION, readfunc);
curl_easy_setopt(curlhandle, CURLOPT_READDATA, f);
curl_easy_setopt(curlhandle, CURLOPT_NEW_FILE_PERMS, 0664L);

curl_easy_perform(curlhandle);
}
curl_easy_cleanup(curlhandle);

if(result == CURLE_OK) {
    // Resumer du fichier

    char buffer [100];
    while (!feof(f))
    {
        if ( fgets (buffer , 100 , f) == NULL ) break;
        fputs (buffer , stdout);
    }
    fclose(f);
    return 1;
}
else {
    fprintf(stderr, "%s\n", curl_easy_strerror(result));
    fclose(f);
    return 0;
}
}

```

librairie

- curl

Note

QT6 ne propose plus de librairie supportant **le protocole SFTP**. C'est pour ça que j'utilise **Curl** une librairie tiers et open source. Pour pouvoir l'exploiter j'utilise le gestionnaire de librairie **vcpkg** et un fichier **manifest.json** pour sélectionner la bonne librairie.

vcpkg

docs

manifest.json

```
{
  "$schema": "https://raw.githubusercontent.com/microsoft/vcpkg-tool/main/docs/vcpkg.schema.json",
  "name": "mypackage",
  "version-string": "0.0.1",
  "dependencies": [
    {
      "name": "curl",
      "default-features": false,
      "features": ["ssh", "openssl", "tool", "sspi", "ssl", "schannel", "non-http"]
    }
  ]
}
```

.env

Note

Pour simplifier le changement d'identifiant et mot de passe pour la requête sftp et pour éviter qu'ils soient codé en dur dans le code.

```
#include <cstdlib>

#ifdef _WIN32
#define setenv(name, value, overwrite) _putenv_s(name, value)
#endif
void loadEnvFile(const std::string& filename);

int main(int argc, char *argv[]) {
    // ...
    loadEnvFile("../.env");
    // Lire une variable d'environnement
    const char* secretKey = std::getenv("FTP_IDPASS");
    if (secretKey) {
        std::cout << "Cle secrète chargée avec succès !" << std::endl;
    } else {
        std::cerr << "Erreur : FTP_IDPASS non définie !" << std::endl;
    }
    // ...
}

void loadEnvFile(const std::string& filename) {
    std::ifstream file(filename);
    if (!file.is_open()) {
        std::cerr << "Erreur : Impossible d'ouvrir le fichier "
                  << filename
                  << std::endl;
        return;
    }
}
```

```

std::string line;
while (std::getline(file, line)) {
    size_t delimiterPos = line.find('=');
    if (delimiterPos != std::string::npos) {
        std::string key = line.substr(0, delimiterPos);
        std::string value = line.substr(delimiterPos + 1);

        // Définir la variable d'environnement
        setenv(key.c_str(), value.c_str(), 1);
    }
}

file.close();
}

```

BDD / Cron Job

BDD

- Nom Long: Suivie_panne_carteElectronique
- Nom Court: SVPANNCTP0
- Format: FSVPANNCT

[Script.sql](#)

#todo

- ☐ mettre à jour les images de la requêtes

Création table

```

CREATE OR REPLACE TABLE DEV_FILE.Suivie_panne_carteElectronique FOR SYSTEM NAME SVPANNCTP0 (
  Id                FOR SVP_ID    INT GENERATED BY DEFAULT AS IDENTITY NOT NULL,
  Poste             FOR SVP_POST  CHAR(6)    DEFAULT '' NOT NULL,
  Site              FOR SVP_SITE  CHAR(2)    DEFAULT '' NOT NULL,
  Panne             FOR SVP_PANN  CHAR(5)    DEFAULT '' NOT NULL,
  Statut_flag       FOR SVP_STAT  CHAR(1)    DEFAULT '' NOT NULL,

  Numero_aoi        FOR NUM_AOI   CHAR(40)   DEFAULT '' NOT NULL,
  Numero_of         FOR NUM_OF    CHAR(8)    DEFAULT '' NOT NULL,
  Numero_fic        FOR NUM_FIC   CHAR(15)   DEFAULT '' NOT NULL,
  Indice_fic        FOR IND_FIC   CHAR(2)    DEFAULT '' NOT NULL,

  Mesure            FOR MESR_PANN  DECIMAL(2,6) DEFAULT 0 NOT NULL,
  Limite_min        FOR MESR_LIMIN DECIMAL(2,2) DEFAULT 0 NOT NULL,
  Limite_max        FOR MESR_LIMAX DECIMAL(2,2) DEFAULT 0 NOT NULL,

  Designation_panne FOR DESI_PANN  CHAR(5000)  DEFAULT '' NOT NULL,
  Commentaire_panne FOR COMM_PANN  CHAR(5000)  DEFAULT '' NOT NULL,
  Composant         FOR CPST_PANN  CHAR(21)    DEFAULT '' NOT NULL,
  Commentaire       FOR WBX_COMM   CHAR(5000)  DEFAULT '' NOT NULL,

  Creation_utilisateur FOR CRT_UTIL CHAR(10)    DEFAULT '' NOT NULL,
  Creation_date       FOR CRT_DATE  CHAR(10)    DEFAULT '' NOT NULL,
  Creation_heure      FOR CRT_HEUR  CHAR(8)     DEFAULT '' NOT NULL,
  Modification_utilisateur FOR MOD_UTIL CHAR(10)    DEFAULT '' NOT NULL,
  Modification_date   FOR MOD_DATE  CHAR(10)    DEFAULT '' NOT NULL,
  Modification_heure  FOR MOD_HEUR  CHAR(8)     DEFAULT '' NOT NULL,
) RCDFMT FSVPANNCT;

```



```

--LABEL TABLE
LABEL ON TABLE DEV_FILE.Suivie_panne_carteElectronique IS 'Suivie de Panne - Carte
Electronique';
--COMMENT TABLE
COMMENT ON TABLE DEV_FILE.Suivie_panne_carteElectronique IS 'Suivie de Panne - Carte
Electronique';

--LABEL COLUMN
LABEL ON COLUMN DEV_FILE.Suivie_panne_carteElectronique (
Id                                TEXT IS                                'Id auto increment'
Poste                             TEXT IS                                'Poste de Dépannage utilisée'
Site                              TEXT IS                                'Site de Production'
Panne                             TEXT IS                                'Numéro de panne'
Statut_flag                       TEXT IS                                'État (ex : résolu, en cours)'
Numero_aoi                        TEXT IS                                'Numero AOI de la carte'
Numero_of                         TEXT IS                                'Numero OF de la carte'
Numero_fic                       TEXT IS                                'Numero FIC de la carte'
Indice_fic                       TEXT IS                                'Indice FIC de la carte'
Mesure                           TEXT IS                                'Mesure Tension de la panne'
Limite_min                       TEXT IS                                'Limite Min mesurer'
Limite_max                       TEXT IS                                'Limite Max mesurer'
Designation_panne                TEXT IS                                'Désignation ou type de la panne'
Commentaire_panne                TEXT IS                                'Commentaire relatif à la panne'
Composant                        TEXT IS                                'Composant concerné par la panne'
Commentaire                      TEXT IS                                'Commentaire Dépanneur'
Creation_utilisateur             TEXT IS                                'Utilisateur ayant créer le suivi de panne'
Creation_date                    TEXT IS                                'Date de la Panne'
Creation_heure                   TEXT IS                                'Heure de la Panne'
Modification_utilisateur         TEXT IS                                'Utilisateur ayant fait la dernière modification'
Modification_date                TEXT IS                                'Date de la dernière modification'
Modification_heure               TEXT IS                                'Heure de la dernière modification'
);

--COMMENT COLUMN
COMMENT ON COLUMN DEV_FILE.Suivie_panne_carteElectronique (
Id                                IS                                'Id auto increment'
Poste                             IS                                'Poste de Dépannage utilisée'
Site                              IS                                'Site de Production'
Panne                             IS                                'Numéro de panne'
Statut_flag                       IS                                'État (ex : résolu, en cours)'
Numero_aoi                        IS                                'Numero AOI de la carte'
Numero_of                         IS                                'Numero OF de la carte'
Numero_fic                       IS                                'Numero FIC de la carte'
Indice_fic                       IS                                'Indice FIC de la carte'
Mesure                           IS                                'Mesure Tension de la panne'
Limite_min                       IS                                'Limite Min mesurer'
Limite_max                       IS                                'Limite Max mesurer'
Designation_panne                IS                                'Désignation ou type de la panne'
Commentaire_panne                IS                                'Commentaire relatif à la panne'
Composant                        IS                                'Composant concerné par la panne'
Commentaire                      IS                                'Commentaire Dépanneur'
Creation_utilisateur             IS                                'Utilisateur ayant créer le suivi de panne'
Creation_date                    IS                                'Date de la Panne'
Creation_heure                   IS                                'Heure de la Panne'
Modification_utilisateur         IS                                'Utilisateur ayant fait la dernière modification'
Modification_date                IS                                'Date de la dernière modification'
Modification_heure               IS                                'Heure de la dernière modification'
);

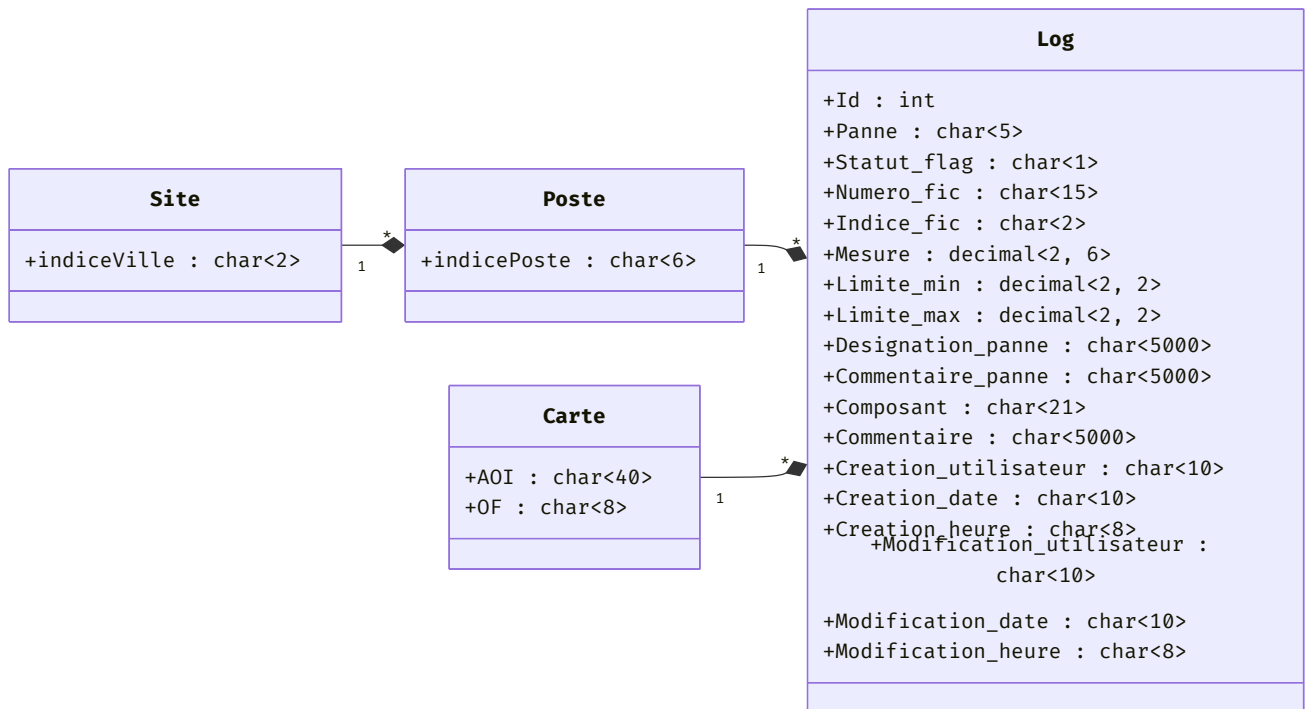
--CREATE INDEX
CREATE INDEX DEV_FILE.SVPANNCT_L1 ON DEV_FILE.Suivie_panne_carteElectronique (Id);
COMMENT ON INDEX DEV_FILE.SVPANNCT_L1 IS 'SVPANNCT - (Id)';
LABEL ON INDEX DEV_FILE.SVPANNCT_L1 IS 'SVPANNCT - (Id)';

CREATE INDEX DEV_FILE.SVPANNCT_L2 ON DEV_FILE.Suivie_panne_carteElectronique (Numero_aoi);
COMMENT ON INDEX DEV_FILE.SVPANNCT_L2 IS 'SVPANNCT - (Numero_aoi)';
LABEL ON INDEX DEV_FILE.SVPANNCT_L2 IS 'SVPANNCT - (Numero_aoi)';

CREATE INDEX DEV_FILE.SVPANNCT_L3 ON DEV_FILE.Suivie_panne_carteElectronique (Numero_of);
COMMENT ON INDEX DEV_FILE.SVPANNCT_L3 IS 'SVPANNCT - (Numero_of)';
LABEL ON INDEX DEV_FILE.SVPANNCT_L3 IS 'SVPANNCT - (Numero_of)';

```

Diagramme de Relation



Cron Job

- Tous les 00h05 de chaque jour
- Executable par l'actualisation de la base sur Webix

[docs](#)

Récupération du fichier

```

GestPrd_SuiviPanne_integration.php

try {
    // Charger le fichier XML
    $V_xmlSuiviPanne = simplexml_load_file($V_xmlFilePath) or die('Erreur de chargement du fichier XML');
    foreach ($V_xmlSuiviPanne->Log as $V_xmlLog) { // phpcs:ignore
        $V_dateHeure = explode(' ', $V_xmlLog->dateHeure);
        $V_date = addslashes(trim(date('Y-m-d', strtotime(str_replace('/', '-', $V_dateHeure[0])))));
        $V_heure = addslashes(date('H:i:s', strtotime($V_dateHeure[1])));

        $V_logTemp = new Log(
            P_Poste: addslashes($V_xmlLog->poste),
            P_Site: addslashes($V_xmlLog->site),
            P_Panne: addslashes($V_xmlLog->panne),
            P_Statut_flag: addslashes($V_xmlLog->stateFlag),
            P_Numero_aoi: addslashes($V_xmlLog->AOI),
            P_Numero_of: addslashes($V_xmlLog->OF),
            P_Numero_fic: addslashes($V_xmlLog->FIC),
            P_Indice_fic: addslashes($V_xmlLog->iFIC),
            P_Mesure: (float) addslashes($V_xmlLog->mesure),
            P_Limite_min: (float) addslashes($V_xmlLog->limiteMin),
            P_Limite_max: (float) addslashes($V_xmlLog->limiteMax),
            P_Designation_panne: addslashes($V_xmlLog->designationPanne),
            P_Commentaire_panne: addslashes($V_xmlLog->commentairePanne),
            P_Composant: addslashes($V_xmlLog->composant),
            P_Commentaire: addslashes($V_xmlLog->commentaire),
            P_Creation_date: $V_date, // formate la date pour l'IBM DB2
            P_Creation_heure: $V_heure // format l'heure pour l'IBM DB2
        );
        $V_logList[] = $V_logTemp; // Ajout à la liste de Logs
    }

    // Transfert BDD
}

```

Transfert à la base de données

```

GestPrd_SuiviPanne_integration.php

try {
    // Transfert BDD
    $V_requeteSql = 'INSERT INTO ' . $V_tableSuiviPanne . ' (
        SVP_POST, SVP_SITE, SVP_PANN, SVP_STAT,
        NUM_AOI, NUM_OF, NUM_FIC, IND_FIC,
        MESR_PANN, MESR_LIMIN, MESR_LIMAX,
        DESI_PANN, COMM_PANN, CPST_PANN, WBX_COMM,
        CRT_DATE, CRT_HEUR
    ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)';

    $V_stmt = odbc_prepare($V_connector, $V_requeteSql);

    foreach ($V_logList as $V_log) { // Insertion de chaque log dans la requête
        // phpcs:disable
        $V_params = [
            trim($V_log->V_Poste),
            trim($V_log->V_Site),
            trim($V_log->V_Panne),
            trim($V_log->V_Statut_flag),
            trim($V_log->V_Numero_aoi),
            trim($V_log->V_Numero_of),
            trim($V_log->V_Numero_fic),
            trim($V_log->V_Indice_fic),
            floatval($V_log->V_Mesure),
            floatval($V_log->V_Limite_min),
            floatval($V_log->V_Limite_max),
            addslashes($V_log->V_Designation_panne),
            addslashes($V_log->V_Commentaire_panne),
            addslashes($V_log->V_Composant),
            addslashes($V_log->V_Commentaire),
            date('Y-m-d', strtotime(str_replace('/', '-', trim($V_log->V_Creation_date)))),
            date('H:i:s', strtotime(trim($V_log->V_Creation_heure)))
        ];
        // phpcs:enable

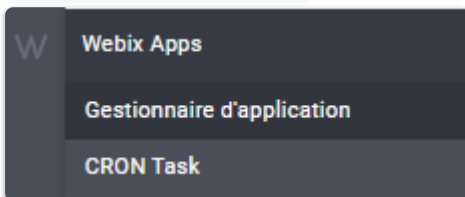
        print_r($V_requeteSql . "\n");
        $V_executeSql = odbc_execute($V_stmt, $V_params);
        print_r('Requête code résultat: ' . $V_executeSql . "\n");
    }
}

```

AS400 / Webix

AS400

Ctrl + Shift + F10



Webix

