#ifndef VERSIONCHECK\_H

#define VERSIONCHECK\_H

#include <QObject>

#include <QProcess>

#include <QString>

#include <QDebug>

class VersionCheck : public QObject {

Q\_OBJECT

public:

explicit VersionCheck(QObject \*parent = nullptr) : QObject(parent) {}

bool isJavaInstalled() {

QProcess process;

process.start("java", {"-version"});

if (!process.waitForStarted(2000)) return false;

process.waitForFinished(3000);

QString output = process.readAllStandardError();

qDebug() << "[Java Output]:" << output;

return output.contains("version");

}

bool isPostgreSQLInstalled() {

QProcess process;

#ifdef Q\_OS\_WIN

process.start("psql", {"--version"});

#else

process.start("which", {"psql"});

#endif

if (!process.waitForStarted(2000)) return false;

process.waitForFinished(3000);

QString output = process.readAllStandardOutput();

qDebug() << "[PostgreSQL Output]:" << output;

return output.contains("psql") || output.contains("PostgreSQL");

}

bool isDotNet8Installed() {

QProcess process;

process.start("dotnet", {"--version"});

if (!process.waitForStarted(2000)) return false;

process.waitForFinished(3000);

QString output = process.readAllStandardOutput().trimmed();

qDebug() << "[.NET Output]:" << output;

QStringList parts = output.split('.');

if (parts.size() >= 1) {

bool ok;

int major = parts[0].toInt(&ok);

return ok && major == 8;

}

return false;

}

};

#endif // VERSIONCHECK\_H

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

bool isPostgreSQLInstalled() {

QProcess process;

process.start("cmd", {"/c", "where psql"});

process.waitForFinished(3000);

QString output = process.readAllStandardOutput();

qDebug() << "[PostgreSQL WHERE Output]:" << output;

return output.contains("psql.exe");

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

bool safeRun(QProcess &process, const QString &program, const QStringList &args, QString &output, bool useStdErr = false) {

process.start(program, args);

if (!process.waitForStarted(2000)) return false;

process.waitForFinished(3000);

output = useStdErr ? process.readAllStandardError() : process.readAllStandardOutput();

return !output.trimmed().isEmpty();

}

bool isJavaInstalled() {

QProcess process;

QString output;

bool ok = safeRun(process, "java", {"-version"}, output, true);

qDebug() << "[Java]" << output;

return ok && output.contains("version");

}

bool isPostgreSQLInstalled() {

QProcess process;

QString output;

#ifdef Q\_OS\_WIN

bool ok = safeRun(process, "cmd", {"/c", "where psql"}, output);

#else

bool ok = safeRun(process, "which", {"psql"}, output);

#endif

qDebug() << "[PostgreSQL]" << output;

return ok && output.contains("psql");

}

bool isDotNet8Installed() {

QProcess process;

QString output;

bool ok = safeRun(process, "dotnet", {"--version"}, output);

qDebug() << "[.NET]" << output;

QStringList parts = output.trimmed().split('.');

if (ok && parts.size() >= 1) {

bool parseOk = false;

int major = parts[0].toInt(&parseOk);

return parseOk && major == 8;

}

return false;

}