

FTP文件自动采集系统 - 开发设计文档

1. 系统概述

1.1 项目简介

FTP文件自动采集系统是一个基于.NET 8.0的控制台应用程序，用于自动从FTP服务器下载Excel和PDF文件，解析文件内容，并将数据存储到SQL Server数据库中。系统支持三坐标测量报告（Excel格式）和Renishaw球杆仪诊断报告（PDF格式）的自动处理。

1.2 核心功能

- **FTP文件下载**：自动从FTP服务器下载Excel和PDF文件
- **文件解析**：解析Excel和PDF文件内容，提取结构化数据
- **数据存储**：将解析的数据存储到数据库
- **数据映射**：通过配置将文件数据映射到业务表
- **SQL执行**：支持配置化的SQL语句执行
- **文件分类**：自动将处理后的文件移动到成功/失败目录
- **日志记录**：完整的日志记录系统（文件日志和数据库日志）

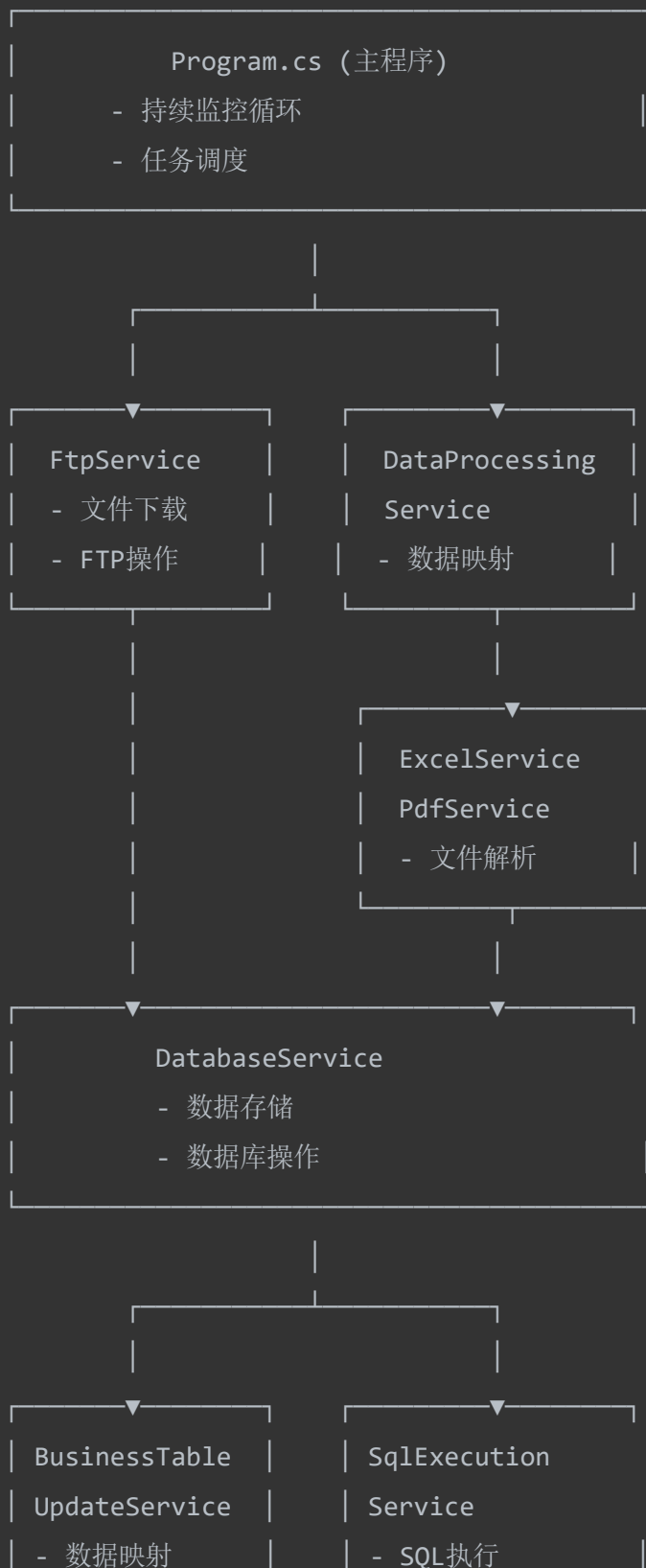
1.3 技术栈

- **开发语言**：C# (.NET 8.0)
- **数据库**：SQL Server
- **文件处理**：EPPlus (Excel), UglyToad.PdfPig (PDF)
- **日志框架**：Serilog
- **配置管理**：Microsoft.Extensions.Configuration
- **依赖注入**：Microsoft.Extensions.DependencyInjection

2. 系统架构

2.1 架构设计

系统采用分层架构设计，主要分为以下层次：



2.2 核心服务说明

2.2.1 FtpService

职责：FTP文件下载和管理

- 从FTP服务器下载Excel和PDF文件
- 支持Excel和PDF文件分别存放在不同目录
- 下载成功后自动删除FTP服务器上的文件
- 文件列表获取和文件下载

2.2.2 ExcelService

职责：Excel文件解析

- 读取三坐标测量报告（FAI格式）
- 提取表头信息（PartNumber, PartName, SerialNumber等）
- 提取测量数据行
- 智能识别数据表起始行
- 跳过表头和表单信息行

2.2.3 PdfService

职责：PDF文件解析

- 读取Renishaw球杆仪诊断报告
- 提取表头信息（TestId, Operator, TestDate, Machine, QC20W等）
- 提取诊断数据（反向间隙、反向跃冲、垂直度、圆度等）
- 支持XY/YZ/ZX标识提取
- 支持特殊符号处理

2.2.4 DatabaseService

职责：数据库操作

- 保存文件信息到FileInfo表
- 保存文件数据到FileData表
- 支持事务处理
- 数据验证和截断处理

2.2.5 DataProcessingService

职责：数据处理和映射

- 查询未处理的文件
- 调用BusinessTableUpdateService进行数据映射
- 标记文件为已处理

2.2.6 BusinessTableUpdateService

职责：业务表更新

- 根据DataMappingConfig配置生成SQL语句
- 将SQL语句保存到SqlExecutionConfig表
- 不直接执行SQL，实现读写分离

2.2.7 SqlExecutionService

职责：SQL执行管理

- 从SqlExecutionConfig表读取待执行的SQL
- SQL验证（检查危险关键字、WHERE子句等）
- 执行SQL并记录执行日志
- 支持手动添加SQL配置
- 限制受影响行数（最多10条）

2.2.8 FileClassificationService

职责：文件分类管理

- 将处理成功的文件移动到成功目录
- 将处理失败的文件移动到失败目录
- 支持Excel和PDF文件分别存放

3. 数据库设计

3.1 数据库表结构

3.1.1 FileInfo （文件信息表）

存储文件的元数据信息。

字段名	类型	说明	约束
Id	INT	主键，自增ID	PRIMARY KEY
SourceFileName	NVARCHAR(500)	源文件名	NOT NULL
FileType	NVARCHAR(50)	文件类型（Excel/PDF）	NOT NULL
PartNumber	NVARCHAR(200)	零件号（Excel）	NULL
PartName	NVARCHAR(500)	零件名称（Excel）	NULL
SerialNumber	NVARCHAR(200)	序列号	NULL
TestId	NVARCHAR(200)	测试ID（PDF）	NULL
Operator	NVARCHAR(200)	操作员	NULL
TestDate	NVARCHAR(100)	测试日期	NULL
Machine	NVARCHAR(500)	机器名称	NULL
QC20W	NVARCHAR(200)	QC20-W信息（PDF）	NULL
LastCalibration	NVARCHAR(50)	上次校准日期	NULL
ImportTime	DATETIME	导入时间	NOT NULL, DEFAULT GETDATE()

字段名	类型	说明	约束
Processed	BIT	处理状态（0=未处理，1=已处理）	NOT NULL, DEFAULT 0
ProcessedTime	DATETIME	处理时间	NULL

索引：

- `idx_FileInfo_FileType` - 按文件类型查询
- `idx_FileInfo_PartNumber` - 按零件号查询
- `idx_FileInfo_SerialNumber` - 按序列号查询
- `idx_FileInfo_TestId` - 按测试ID查询
- `idx_FileInfo_ImportTime` - 按导入时间查询
- `idx_FileInfo_Processed` - 按处理状态查询

3.1.2 FileData（文件数据表）

存储从文件中提取的所有数据，采用键值对形式。

字段名	类型	说明	约束
Id	INT	主键，自增ID	PRIMARY KEY
FileInfold	INT	文件信息ID	NOT NULL, FOREIGN KEY
SourceFileName	NVARCHAR(500)	源文件名（冗余字段）	NOT NULL
FileType	NVARCHAR(50)	文件类型	NOT NULL
RowNumber	INT	行号	NOT NULL
ColumnName	NVARCHAR(200)	列名	NOT NULL
ColumnValue	NVARCHAR(MAX)	列值	NULL
ImportTime	DATETIME	导入时间	NOT NULL, DEFAULT GETDATE()

外键：

- FK_FileData_FileInfo - 关联到FileInfo.Id，ON DELETE CASCADE

索引：

- idx_FileData_FileInfoId - 按文件信息ID查询（主关联索引）
- idx_FileData_SourceFileName - 按文件名查询
- idx_FileData_FileType - 按文件类型查询
- idx_FileData_ImportTime - 按导入时间查询

3.1.3 SystemLog（系统日志表）

存储系统运行日志。

字段名	类型	说明	约束
Id	BIGINT	主键，自增ID	PRIMARY KEY
LogTime	DATETIME	日志时间	NOT NULL, DEFAULT GETDATE()
Level	NVARCHAR(50)	日志级别	NOT NULL
Category	NVARCHAR(200)	日志分类	NULL
Message	NVARCHAR(MAX)	日志消息	NULL
Exception	NVARCHAR(MAX)	异常信息	NULL
FileName	NVARCHAR(500)	关联文件名	NULL

索引：

- idx_SystemLog_LogTime - 按日志时间查询
- idx_SystemLog_Level - 按日志级别查询
- idx_SystemLog_Category - 按日志分类查询

3.1.4 DataMappingConfig（数据映射配置表）

存储数据映射规则配置，支持标准映射和自定义SQL模板两种模式。

字段名	类型	说明	约束
Id	INT	主键, 自增ID	PRIMARY KEY
ConfigName	NVARCHAR(200)	配置名称	NOT NULL
SourceTable	NVARCHAR(200)	源表名 (通常为 FileData)	NOT NULL
SourceFileType	NVARCHAR(50)	源文件类型 (Excel/PDF)	NULL
SourceMatchField	NVARCHAR(200)	源匹配字段 (FileData的 ColumnName)	NOT NULL
SourceDataField	NVARCHAR(200)	源数据字段 (FileData的 ColumnName)	NOT NULL
TargetTable	NVARCHAR(200)	目标表名	NOT NULL
TargetMatchField	NVARCHAR(200)	目标匹配字段	NOT NULL
TargetUpdateField	NVARCHAR(200)	目标更新字段	NOT NULL
MatchCondition	NVARCHAR(500)	额外匹配条件 (SQL WHERE子句)	NULL
IsActive	BIT	是否启用	NOT NULL, DEFAULT 1
Description	NVARCHAR(500)	描述	NULL
CustomSqlTemplate	NVARCHAR(MAX)	自定义SQL模板 (支持占 位符)	NULL
UseCustomSqlTemplate	BIT	是否使用自定义SQL模板	NOT NULL, DEFAULT 0
TemplateParameters	NVARCHAR(MAX)	模板参数 (JSON格式)	NULL
CreateTime	DATETIME	创建时间	NOT NULL, DEFAULT GETDATE()
UpdateTime	DATETIME	更新时间	NOT NULL, DEFAULT GETDATE()

索引：

- idx_DataMappingConfig_IsActive - 按启用状态查询
- idx_DataMappingConfig_SourceTable - 按源表查询
- idx_DataMappingConfig_TargetTable - 按目标表查询

说明：

- **标准映射模式**（UseCustomSqlTemplate = 0）：根据SourceMatchField和SourceDataField自动生成UPDATE语句
- **自定义SQL模板模式**（UseCustomSqlTemplate = 1）：使用CustomSqlTemplate中的SQL模板，支持占位符替换

3.1.5 SqlExecutionConfig（SQL执行配置表）

存储待执行的SQL语句配置。

字段名	类型	说明	约束
Id	INT	主键，自增ID	PRIMARY KEY
ConfigName	NVARCHAR(200)	配置名称	NOT NULL
SqlType	NVARCHAR(50)	SQL类型 (Mapping/Manual)	NOT NULL
SqlStatement	NVARCHAR(MAX)	SQL语句	NOT NULL
Parameters	NVARCHAR(MAX)	参数（JSON格式）	NULL
Description	NVARCHAR(500)	描述	NULL
IsActive	BIT	是否启用	NOT NULL, DEFAULT 1
ExecutionOrder	INT	执行顺序	NOT NULL, DEFAULT 0
ValidationEnabled	BIT	启用验证（必须为1）	NOT NULL, DEFAULT 1
LastExecuteTime	DATETIME	最后执行时间	NULL
LastExecuteResult	NVARCHAR(MAX)	最后执行结果	NULL
ExecuteCount	INT	执行次数	NOT NULL, DEFAULT 0

字段名	类型	说明	约束
CreateTime	DATETIME	创建时间	NOT NULL, DEFAULT GETDATE()
UpdateTime	DATETIME	更新时间	NOT NULL, DEFAULT GETDATE()

索引：

- idx_SqlExecutionConfig_IsActive - 按启用状态查询
- idx_SqlExecutionConfig_SqlType - 按SQL类型查询
- idx_SqlExecutionConfig_ExecutionOrder - 按执行顺序查询

3.1.6 SqlExecutionLog （SQL执行日志表）

记录SQL执行历史。

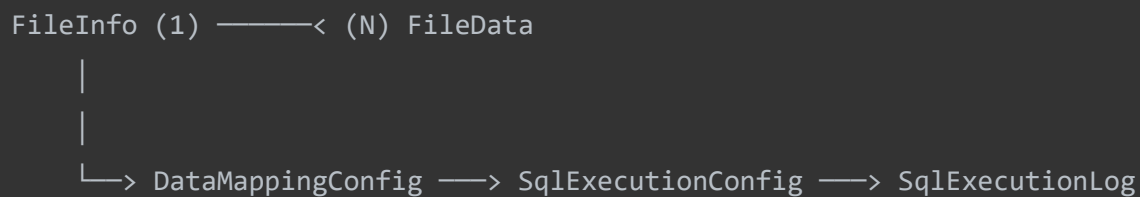
字段名	类型	说明	约束
Id	BIGINT	主键，自增ID	PRIMARY KEY
ConfigId	INT	配置ID	NOT NULL
ConfigName	NVARCHAR(200)	配置名称	NOT NULL
SqlStatement	NVARCHAR(MAX)	SQL语句	NOT NULL
Parameters	NVARCHAR(MAX)	参数	NULL
ExecutionTime	DATETIME	执行时间	NOT NULL, DEFAULT GETDATE()
ExecutionDuration	INT	执行耗时（毫秒）	NULL
RowsAffected	INT	受影响行数	NULL
IsSuccess	BIT	是否成功	NOT NULL
ErrorMessage	NVARCHAR(MAX)	错误信息	NULL
SourceFileName	NVARCHAR(500)	源文件名	NULL

索引：

- idx_SqlExecutionLog_ConfigId - 按配置ID查询

- `idx_SqlExecutionLog_ExecutionTime` - 按执行时间查询
- `idx_SqlExecutionLog_IsSuccess` - 按执行结果查询

3.2 表关系图



4. 核心工作流程

4.1 文件处理流程

1. FTP文件下载

- └─ 从Excel目录下载Excel文件
- └─ 从PDF目录下载PDF文件

2. 文件解析

- └─ Excel文件 → `ExcelService.ReadExcelFileAsync()`
 - └─ 提取表头信息
 - └─ 提取数据行
- └─ PDF文件 → `PdfService.ReadPdfFileAsync()`
 - └─ 提取表头信息
 - └─ 提取诊断数据

3. 数据存储

- └─ 保存`FileInfo`记录
- └─ 获取`FileInfo.Id`
- └─ 保存`FileData`记录（关联`FileInfoId`）

4. 文件分类

- └─ 成功 → 移动到成功目录
- └─ 失败 → 移动到失败目录

4.2 数据映射流程

1. 查询未处理的文件
 - └─ `FileInfo WHERE Processed = 0`
2. 读取文件数据
 - └─ `FileData WHERE FileInfoId = @FileInfoId`
3. 获取映射配置
 - └─ `DataMappingConfig WHERE SourceFileType = @FileType AND IsActive = 1`
4. 生成SQL语句
 - └─ 根据映射规则生成UPDATE/INSERT/MERGE语句
5. 保存SQL到配置表
 - └─ `INSERT INTO SqlExecutionConfig (SqlStatement, Parameters, ...)`
6. 标记文件为已处理
 - └─ `UPDATE FileInfo SET Processed = 1, ProcessedTime = GETDATE()`

4.3 SQL执行流程

1. 查询待执行的SQL
 - └─ `SqlExecutionConfig WHERE IsActive = 1 AND ExecuteCount = 0`
2. SQL验证（强制）
 - └─ 检查危险关键字（DROP，TRUNCATE等）
 - └─ 检查SQL类型（只允许INSERT/UPDATE/DELETE/MERGE）
 - └─ UPDATE/DELETE必须包含WHERE子句
 - └─ 语法验证
3. 执行SQL
 - └─ `SET ROWCOUNT 10`（限制受影响行数）
 - └─ 执行SQL语句
 - └─ `SET ROWCOUNT 0`
4. 记录执行日志
 - └─ `INSERT INTO SqlExecutionLog`

5. 更新配置状态

```
└─ UPDATE SqlExecutionConfig SET ExecuteCount = ExecuteCount + 1, LastExecuteTime  
= GETDATE()
```

5. 核心服务详细设计

5.1 ExcelService

5.1.1 主要方法

ReadExcelFileAsync(string filePath)

- 功能：读取Excel文件内容
- 参数：文件路径
- 返回：ExcelFileData对象
- 处理流程：
 1. 打开Excel文件（使用EPPlus）
 2. 读取指定工作表（默认第3个工作表，索引2）
 3. 提取表头信息
 4. 查找数据表起始行
 5. 读取列标题
 6. 读取数据行（跳过空行和表头行）

ExtractHeaderInfo(ExcelWorksheet worksheet, int rowCount, int colCount)

- 功能：提取表头信息
- 提取字段：PartNumber, PartName, SerialNumber, Machine等
- 使用GetCellText方法获取格式化文本

FindDataTableStartRow(ExcelWorksheet worksheet, int rowCount, int colCount)

- 功能：查找数据表起始行
- 策略：
 1. 查找常见表头关键词（Char No, Reference Location等）

2. 查找以数字开头的行（特征编号）
3. 默认返回第1行

IsHeaderOrFormRow(ExcelWorksheet worksheet, int row, int colCount)

- 功能：判断是否为表头或表单信息行
- 策略：检查是否包含关键词（Form, First Article Inspection等）

GetCellText(ExcelRange cell)

- 功能：获取单元格文本（优先使用格式化显示文本）
- 处理：支持富文本、公式、特殊字符

5.2 PdfService

5.2.1 主要方法

ReadPdfFileAsync(string filePath)

- 功能：读取PDF文件内容
- 参数：文件路径
- 返回：PdfFileData对象
- 处理流程：
 1. 打开PDF文件（使用UglyToad.PdfPig）
 2. 提取所有页面文本
 3. 提取表头信息
 4. 提取诊断数据（传入HeaderInfo以获取XY/YZ/ZX标识）

ExtractHeaderInfo(string text)

- 功能：提取表头信息
- 提取字段：
 - TestId（格式：ZX 220度 150mm 20251212-084641）
 - Operator（操作者）
 - TestDate（快速检测日期或日期）

- Machine (机器名称, 支持多种格式)
- QC20W (QC20-W信息)
- LastCalibration (上次校准日期)

ExtractDiagnosticData(string text, Dictionary<string, string> headerInfo)

- 功能: 提取诊断数据
- 提取数据项:
 - 反向间隙X/Y/Z
 - 横向间隙Z
 - 反向跃冲X/Y/Z
 - 垂直度
 - 伺服不匹配
 - 圆度
 - 运行和拟合信息
- 特殊处理:
 - 从TestId中提取XY/YZ/ZX标识
 - 在数据键名中添加标识后缀 (如: 反向间隙X_ZX_百分比)
 - 支持特殊符号 (箭头、方向符号等)
 - 支持有空格和无空格格式

5.3 DatabaseService

5.3.1 主要方法

SaveExcelDataAsync(ExcelFileData fileData)

- 功能: 保存Excel数据到数据库
- 处理流程:
 1. 开启事务
 2. 插入FileInfo记录, 获取FileInfolId
 3. 遍历数据行, 插入FileData记录
 4. 跳过ColumnValue为空的数据

5. 截断过长的列名 (>200字符) 和列值 (>10000字符)
6. 提交事务

SavePdfDataAsync(PdfFileData fileData)

- 功能：保存PDF数据到数据库
- 处理流程：
 1. 开启事务
 2. 插入FileInfo记录，获取FileInfold
 3. 遍历DiagnosticData，插入FileData记录
 4. 跳过ColumnValue为空的数据
 5. 截断过长的列名和列值
 6. 提交事务

ExecuteNonQueryAsync(string sql, Dictionary<string, object>? parameters, SqlConnection? connection, SqlTransaction? transaction)

- 功能：执行非查询SQL命令
- 支持：事务、连接复用

ExecuteScalarAsync(string sql, Dictionary<string, object>? parameters, SqlConnection? connection, SqlTransaction? transaction)

- 功能：执行标量查询
- 用途：获取自增ID等

5.4 BusinessTableUpdateService

5.4.1 主要方法

UpdateBusinessTablesFromFileDataAsync(int fileInfold, string fileType, SqlConnection connection, SqlTransaction transaction)

- 功能：根据配置规则更新业务表（生成SQL并存储）
- 处理流程：

1. 获取启用的映射配置
2. 读取文件数据行
3. 按配置规则分组处理
4. 生成SQL语句
5. 保存SQL到SqlExecutionConfig表

SaveMappingSqlToConfigAsync(...)

- 功能：保存映射SQL到配置表
- 特点：每个数据行生成一条SQL记录

5.5 SqlExecutionService

5.5.1 主要方法

ValidateSqlAsync(string sql, Dictionary<string, object>? parameters)

- 功能：验证SQL语句安全性
- 验证规则：
 - 禁止关键字：DROP, TRUNCATE, ALTER, CREATE, EXEC, EXECUTE
 - 只允许：INSERT, UPDATE, DELETE, MERGE
 - UPDATE/DELETE必须包含WHERE子句
 - 语法验证（使用SET PARSEONLY ON）

ExecuteSqlConfigByIdAsync(int configId)

- 功能：执行指定ID的SQL配置
- 处理流程：
 1. 读取配置
 2. 强制验证SQL
 3. 执行SQL（限制受影响行数）
 4. 记录执行日志
 5. 更新配置状态

AddSqlConfigAsync(...)

- 功能：手动添加SQL配置
- 特点：添加前强制验证SQL

ExecuteSqlConfigAsync(SqlExecutionConfigModel config)

- 功能：执行SQL配置
- 特点：使用SET ROWCOUNT限制受影响行数

6. 配置说明

6.1 appsettings.json配置结构

```
{
  "ConnectionStrings": {
    "SqlServer": "数据库连接字符串"
  },
  "FtpSettings": {
    "Server": "FTP服务器地址",
    "Username": "FTP用户名",
    "Password": "FTP密码",
    "ExcelRemotePath": "/Excel",
    "PdfRemotePath": "/PDF",
    "Port": 22
  },
  "FileSettings": {
    "LocalDownloadPath": "Downloads",
    "FilePattern": "*.xlsx,*.pdf",
    "ExcelSuccessPath": "Processed/Excel/Success",
    "ExcelFailedPath": "Processed/Excel/Failed",
    "PdfSuccessPath": "Processed/PDF/Success",
    "PdfFailedPath": "Processed/PDF/Failed"
  },
  "ScheduleSettings": {
    "CheckIntervalSeconds": 30
  },
  "LogSettings": {
```

```

        "EnableDatabaseLog": true
    },
    "Logging": {
        "LogLevel": {
            "Default": "Information",
            "Microsoft": "Warning"
        }
    }
}

```

6.2 配置项说明

配置项	说明	默认值
ConnectionStrings:SqlServer	SQL Server数据库连接字符串	必填
FtpSettings:Server	FTP服务器地址	必填
FtpSettings:Username	FTP用户名	可选
FtpSettings>Password	FTP密码	可选
FtpSettings:ExcelRemotePath	Excel文件远程路径	/Excel
FtpSettings:PdfRemotePath	PDF文件远程路径	/PDF
FtpSettings:Port	FTP端口	21
FileSettings:LocalDownloadPath	本地下载目录	Downloads
FileSettings:ExcelSuccessPath	Excel成功文件目录	Processed/Excel/Success
FileSettings:ExcelFailedPath	Excel失败文件目录	Processed/Excel/Failed
FileSettings:PdfSuccessPath	PDF成功文件目录	Processed/PDF/Success
FileSettings:PdfFailedPath	PDF失败文件目录	Processed/PDF/Failed
ScheduleSettings:CheckIntervalSeconds	检查间隔（秒）	30
LogSettings:EnableDatabaseLog	是否启用数据库日志	true

7. 数据映射配置

7.1 标准映射配置

标准映射配置通过以下字段定义映射规则：

- **SourceMatchField**：源匹配字段（FileData表的ColumnName），用于匹配数据行
- **SourceDataField**：源数据字段（FileData表的ColumnName），要更新的数据值
- **TargetTable**：目标业务表名
- **TargetMatchField**：目标表的匹配字段，用于查找要更新的记录
- **TargetUpdateField**：目标表的更新字段，要更新的字段名
- **MatchCondition**：额外的匹配条件（SQL WHERE子句的一部分）

示例配置：

```
INSERT INTO DataMappingConfig
(ConfigName, SourceTable, SourceFileType, SourceMatchField, SourceDataField,
 TargetTable, TargetMatchField, TargetUpdateField, Description, IsActive)
VALUES
('更新检验结果PD', 'FileData', 'Excel', 'sn', 'InspectionResultPD',
 'QC_ProcessProgrammeRecordline', 'sn', 'InspectionResultPD',
 '根据sn匹配，将Excel的InspectionResultPD值更新到QC_ProcessProgrammeRecordline', 1);
```

7.2 自定义SQL模板配置

自定义SQL模板支持复杂的JOIN查询和灵活的SQL语句，通过以下字段配置：

- **UseCustomSqlTemplate**：设置为1启用自定义SQL模板
- **CustomSqlTemplate**：SQL模板，支持占位符
- **TemplateParameters**：模板参数说明（JSON格式）

支持的占位符：

- {FileInfoId} - 文件信息ID
- {SourceFileName} - 源文件名

- {PartName} - 零件名称
- {PartNumber} - 零件号
- {SerialNumber} - 序列号
- {FileType} - 文件类型
- {ColumnName} - 列名（需要TemplateParameters指定）
- {RowNumber} - 行号（需要TemplateParameters指定）
- {ColumnValue} - 列值（需要TemplateParameters指定）

示例配置：

```
INSERT INTO DataMappingConfig
(ConfigName, SourceTable, SourceFileType, SourceMatchField, SourceDataField,
TargetTable, TargetMatchField, TargetUpdateField, Description,
UseCustomSqlTemplate, CustomSqlTemplate, TemplateParameters, IsActive)
VALUES
('更新QC记录_平面度检验结果',
'FileData', 'Excel', 'PartName', '9. Results',
'QC_ProcessProgrammeRecordline', 'sn', 'InspectionResultPD',
'根据PartName匹配, 更新QC_ProcessProgrammeRecordline的InspectionResultPD字段',
1,
N'UPDATE QC_ProcessProgrammeRecordline
SET InspectionResultPD = F.ColumnValue
FROM (
    SELECT fi.PartName, fd.RowNumber, fd.ColumnValue
    FROM FileData fd
    LEFT JOIN FileInfo fi ON fd.FileInfoId = fi.Id
    WHERE fd.ColumnName = '9. Results'
        AND fd.RowNumber = 8
        AND fd.FileInfoId = {FileInfoId}
) F
JOIN (
    SELECT qpl.Id, qpl.sn, qpl.InspectionItemName
    FROM QC_ProcessProgrammeRecordline qpl
    LEFT JOIN QC_ProcessProgrammeRecord qpr ON qpl.ProcessProgrammeRecordId = qpr.Id
    WHERE qpl.InspectionItemName = '精度检验: 大底面平面度'
) R ON F.PartName = R.sn
```

```
WHERE QC_ProcessProgrammeRecordline.Id = R.Id',  
      N'{"ColumnName":"9. Results","RowNumber":"8"}',  
      1);
```

7.3 映射规则说明

- **标准映射**：适用于简单的字段更新，系统自动生成UPDATE语句
- **自定义SQL模板**：适用于复杂的JOIN查询和多表关联更新
- **匹配逻辑**：根据SourceMatchField的值在FileData中查找匹配的数据行
- **更新逻辑**：将SourceDataField的值更新到目标表的TargetUpdateField字段

8. SQL执行配置

8.1 SQL验证规则

1. **禁止的关键字**：DROP, TRUNCATE, ALTER, CREATE, EXEC, EXECUTE
2. **允许的SQL类型**：INSERT, UPDATE, DELETE, MERGE
3. **WHERE子句要求**：UPDATE和DELETE语句必须包含WHERE子句
4. **行数限制**：DML操作最多影响10条记录（使用SET ROWCOUNT 10）
5. **验证强制**：所有SQL必须通过验证才能执行

8.2 SQL执行流程

1. 从SqlExecutionConfig表读取待执行的SQL
2. 强制验证SQL安全性
3. 设置行数限制（SET ROWCOUNT 10）
4. 执行SQL语句
5. 重置行数限制（SET ROWCOUNT 0）
6. 记录执行日志
7. 更新执行状态

9. 持续监控循环

9.1 主循环流程

```
while (!_isRunning)
{
    try
    {
        // 1. 检查并处理FTP文件
        await ProcessFtpFilesAsync();

        // 2. 检查并处理数据映射
        await ProcessDataMappingAsync();

        // 3. 检查并执行SQL
        await ProcessSqlExecutionAsync();
    }
    catch (Exception ex)
    {
        // 错误处理
    }

    // 等待指定间隔
    await Task.Delay(checkIntervalSeconds * 1000);
}
```

9.2 处理流程说明

ProcessFtpFilesAsync()

- 检查FTP服务器是否有新文件
- 下载文件
- 解析文件
- 保存数据
- 移动文件到成功/失败目录

ProcessDataMappingAsync()

- 查询未处理的文件
- 根据配置生成SQL
- 保存SQL到配置表
- 标记文件为已处理

ProcessSqlExecutionAsync()

- 查询待执行的SQL配置
- 验证SQL
- 执行SQL
- 记录执行日志

10. 错误处理和日志

10.1 日志级别

- **Information**：正常操作日志
- **Warning**：警告信息（如文件数据为空）
- **Error**：错误信息（如文件解析失败）
- **Debug**：调试信息（如Machine字段提取）

10.2 日志输出

- **控制台日志**：实时显示程序运行状态
- **文件日志**：使用Serilog写入日志文件（按天滚动）
- **数据库日志**：写入SystemLog表

10.3 错误处理策略

- **文件处理错误**：记录错误日志，移动文件到失败目录，继续处理其他文件
- **数据库错误**：回滚事务，记录错误日志，继续处理
- **SQL执行错误**：记录执行日志，更新配置状态，继续执行其他SQL

11. 部署说明

11.1 环境要求

- .NET 8.0 Runtime
- SQL Server数据库
- FTP服务器访问权限
- 足够的磁盘空间（用于文件下载和日志）

11.2 部署步骤

1. 数据库初始化

```
-- 执行 Scripts/InitializeDatabase.sql
```

2. 配置文件设置

- 编辑 appsettings.json
- 配置数据库连接字符串
- 配置FTP服务器信息
- 配置文件路径

3. 编译发布

```
dotnet publish -c Release -o publish
```

4. 运行程序

```
cd publish  
./FtpExcelProcessor
```

11.3 运行模式

- **持续监控模式：**程序持续运行，定期检查和处

12. 扩展和定制

12.1 添加新的文件类型支持

1. 创建新的Service类（如 WordService ）
2. 实现文件解析逻辑
3. 在 ProcessFtpFilesAsync 中添加文件类型判断
4. 调用相应的Service进行解析

12.2 添加新的数据映射规则

1. 在DataMappingConfig表中插入配置记录
2. 配置MappingRules JSON
3. 系统会自动应用新规则

12.3 手动执行SQL

1. 在SqlExecutionConfig表中插入SQL配置
2. 设置IsActive = 1
3. 系统会在下次循环中自动执行

13. 性能优化

13.1 数据库优化

- 使用索引加速查询
- 使用事务批量操作
- 限制单次处理文件数量

13.2 文件处理优化

- 跳过空值数据
- 截断过长数据
- 异步处理文件

13.3 内存优化

- 及时释放文件资源
- 使用流式处理大文件
- 限制日志文件大小

14. 安全考虑

14.1 SQL注入防护

- 使用参数化查询
- SQL验证机制
- 限制SQL类型和关键字

14.2 文件安全

- 文件类型验证
- 文件大小限制
- 文件路径验证

14.3 数据安全

- 事务保证数据一致性
- 错误回滚机制
- 日志记录所有操作

15. 测试建议

15.1 集成测试

- FTP文件下载测试
- 数据库操作测试
- 完整流程测试

16. 维护和监控

16.1 监控指标

- 文件处理数量
- SQL执行数量
- 错误率
- 处理耗时

16.2 日志分析

- 定期检查SystemLog表
- 分析错误日志
- 优化处理逻辑

16.3 数据库维护

- 定期清理旧日志
- 优化索引
- 备份重要数据

17. 限制和约束

17.1 数据长度限制

- SQL执行最多影响10条记录（使用SET ROWCOUNT 10限制）
- ColumnName最大长度200字符（超过自动截断）
- ColumnValue最大长度10000字符（Excel，超过自动截断）， NVARCHAR(MAX) (PDF)
- Machine字段最大长度500字符
- SourceFileName最大长度500字符

17.2 SQL执行限制

- 只允许INSERT、UPDATE、DELETE、MERGE操作
- 禁止DROP、TRUNCATE、ALTER、CREATE、EXEC、EXECUTE操作

- UPDATE和DELETE必须包含WHERE子句
- 所有SQL必须通过验证才能执行（ValidationEnabled强制为1）
- 受影响行数限制为10条

17.3 文件处理限制

- 支持的文件类型：.xlsx, .xls, .pdf
- Excel文件读取第3个工作表（索引2）
- PDF文件支持Renishaw球杆仪诊断报告格式
- 同名文件可以多次导入（不设置唯一约束）

17.4 架构约束

- 文件处理流程和SQL执行流程完全分离
- 文件处理只生成SQL，不执行SQL
- SQL执行是独立流程，从SqlExecutionConfig表读取并执行
- FileInfo和FileData通过FileInfold关联（外键，级联删除）

18. 版本历史

v1.0.0

- 初始版本
- 支持Excel和PDF文件处理
- 支持数据映射和SQL执行
- 持续监控模式
- 读写分离架构（文件处理与SQL执行分离）
- 自定义SQL模板支持
- FileInfold关联机制
- 文件分类管理（成功/失败目录）
- FTP文件下载后自动删除
- Excel和PDF文件分目录存放

19. 自定义SQL模板详细说明

19.1 占位符说明

占位符	说明	来源
{FileInfold}	文件信息ID	FileInfo表
{SourceFileName}	源文件名	FileInfo表
{PartName}	零件名称	FileInfo表
{PartNumber}	零件号	FileInfo表
{SerialNumber}	序列号	FileInfo表
{FileType}	文件类型	FileInfo表
{ColumnName}	列名	FileData表（需TemplateParameters指定）
{RowNumber}	行号	FileData表（需TemplateParameters指定）
{ColumnValue}	列值	FileData表（需TemplateParameters指定）

19.2 TemplateParameters格式

当SQL模板包含 {ColumnName} 、 {RowNumber} 或 {ColumnValue} 占位符时，需要在TemplateParameters中指定匹配条件：

```
{
  "ColumnName": "9. Results",
  "RowNumber": "8"
}
```

系统会根据这些条件从FileData中查找匹配的数据行，然后替换占位符。

19.3 处理流程

1. 读取配置的CustomSqlTemplate
2. 替换文件级占位符（{FileInfold}, {SourceFileName}等）
3. 如果包含行级占位符，解析TemplateParameters
4. 从FileData中查找匹配的数据行

5. 为每个匹配的数据行生成SQL（替换{ColumnName}, {RowNumber}, {ColumnValue}）
6. 保存生成的SQL到SqlExecutionConfig表

20. 常见问题

20.1 文件处理失败

问题：文件下载后处理失败，移动到失败目录

排查步骤：

1. 检查SystemLog表的错误日志
2. 检查文件格式是否正确
3. 检查数据库连接是否正常
4. 检查FileData表是否有数据

20.2 SQL执行失败

问题：SQL配置执行失败

排查步骤：

1. 检查SqlExecutionLog表的错误信息
2. 检查SQL语句是否正确
3. 检查目标表是否存在
4. 检查WHERE条件是否匹配到数据
5. 检查受影响行数是否超过10条限制

20.3 数据映射不生效

问题：配置了映射规则，但数据没有更新到业务表

排查步骤：

1. 检查DataMappingConfig配置是否正确
2. 检查IsActive是否为1

3. 检查SourceMatchField和SourceDataField是否存在于FileData
4. 检查是否生成了SQL（查看SqlExecutionConfig表）
5. 检查SQL是否执行（查看SqlExecutionLog表）

20.4 Machine字段未解析

问题：PDF文件的Machine字段为空

排查步骤：

1. 检查PDF文件格式是否正确
2. 检查PDF文本中是否包含"机器:"、"Machine:"或"设备:"关键字
3. 查看日志中的Machine提取信息（Debug级别）

21. 最佳实践

21.1 配置管理

- 使用有意义的ConfigName命名
- 添加详细的Description说明
- 先设置IsActive=0测试配置，确认无误后再启用
- 定期检查未使用的配置并清理

21.2 SQL编写

- 使用参数化查询防止SQL注入
- 确保WHERE条件足够精确，避免误更新
- 测试SQL语句后再添加到配置
- 使用自定义SQL模板处理复杂场景

21.3 文件管理

- 定期清理成功目录中的旧文件
- 及时处理失败目录中的文件
- 监控磁盘空间使用情况

- 备份重要文件

21.4 性能优化

- 合理设置CheckIntervalSeconds，避免过于频繁的检查
- 使用索引加速查询
- 定期清理SystemLog和SqlExecutionLog旧数据
- 监控数据库性能

22. 附录

22.1 数据库初始化脚本

执行 Scripts/InitializeDatabase.sql 初始化数据库表结构。

22.2 示例配置

参考 Examples/ 目录下的示例文件：

- CustomSqlTemplateExample.sql - 自定义SQL模板配置示例
- ManualSqlInsertExamples.sql - 手动SQL配置示例

22.3 相关文档

- README.md - 项目说明文档
- Examples/README_CUSTOM_SQL_TEMPLATE.md - 自定义SQL模板使用说明
- Examples/README_MANUAL_SQL.md - 手动SQL配置说明
- Scripts/README.md - SQL脚本说明