

1 兩種資料來源

1.1 Log

1.1.1 加入方法

訊息格式

```
msg =  
string.Format("ST={0},ET={1},ProcessQTY={2},CompeleteQTY={3},UPH={4}",  
dtStart, dtEnd, pqty, QTY, UPH);
```

命令

```
SYSPara.LogSay(EnLoggerType.EnLog_SPC, msg);  
或 SYSPara.LogSay("SPC", msg);
```

1.1.2 設定

預設為 Log，經變更後儲存為 OEEReport 讀取 Log 製作報告



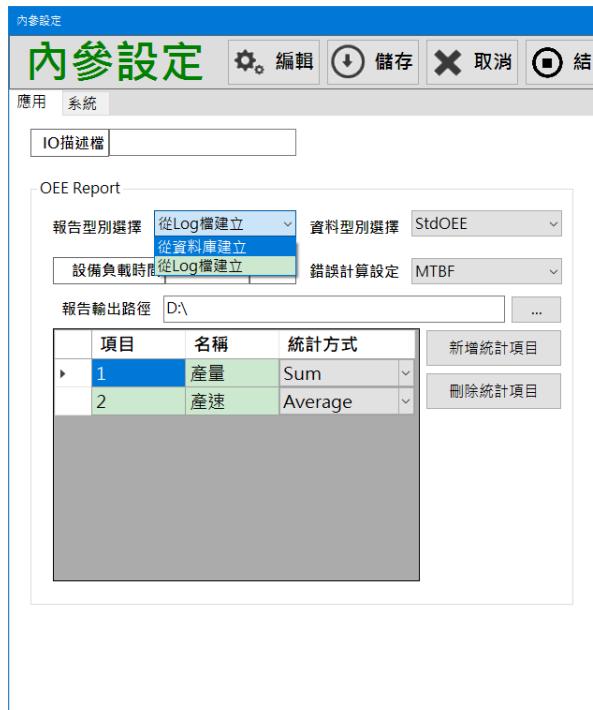
1.2 資料庫

1.2.1 加入方法

LogSay 指定 SPC 加入後會自動加入

1.1.2 設定

預設為 Log，經變更後儲存為 OEEReport 讀取資料庫製作報告



2 使用方式

2.1 Log

2.1.1 選擇起始/結束日期

| | | |
|-------------|------|-------|
| Pr | 生產料號 | |
| 2025年 3月20日 | 日 | |
| 2025年 3月25日 | | LOG分析 |
| | | |

2.1.2 選擇計算單位

這邊需要注意，OEEReport 沒有季跟半年選項，只是選擇後會依預設作分析

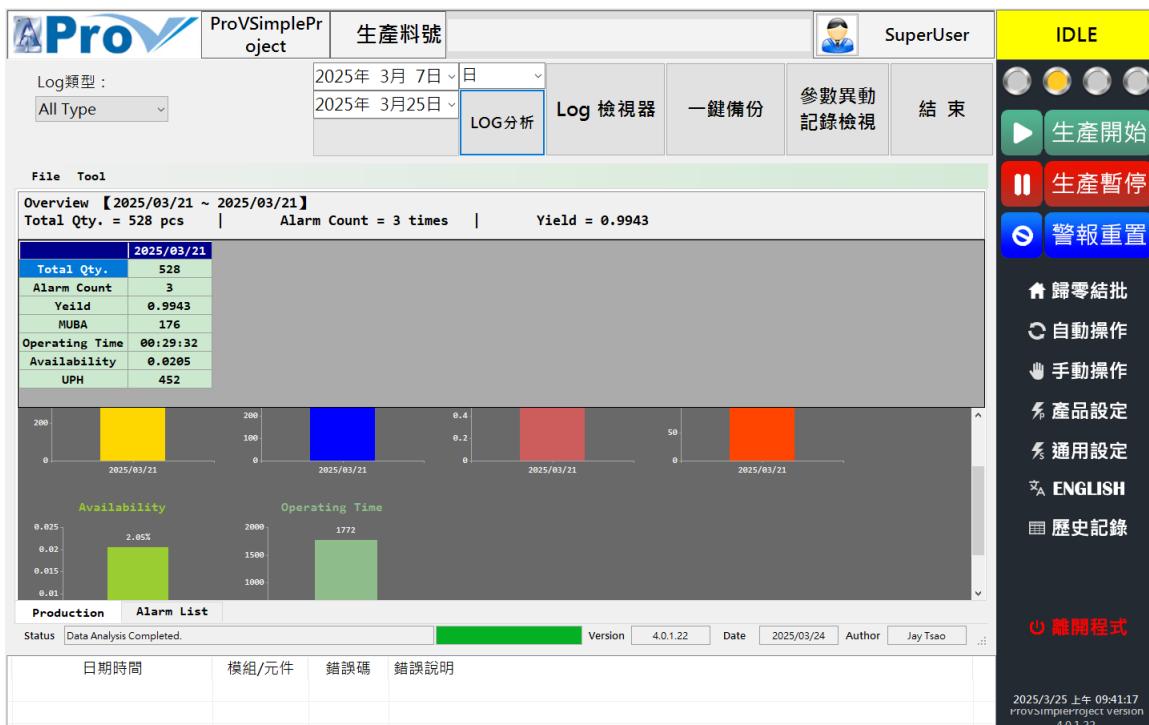
| | | |
|-------------|------|-----|
| Pr | 生產料號 | |
| 2025年 3月20日 | 日 | |
| 2025年 3月25日 | 日 | 周 |
| | | 月 |
| | | 季 |
| 斗號 | 批號 | 內容 |
| | | 半年 |
| | | 使用者 |
| | | 年 |

2.1.3 執行

按下 Log 分析後，OEEReport 會依照選擇時間抓取 D:\Log\中有使用新方式加入資料的檔來做分析，更改區間再按一次 Log 分析，則會重新計算。

2.2 資料庫

操作及設定同 Log 一樣，區別在”內參設定”選擇來源。



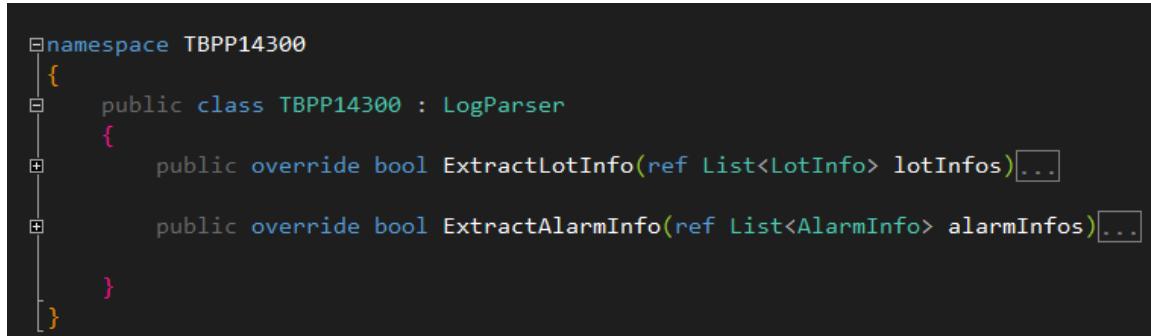
3 自定義資料型別

3.1 何謂型別

型別為資料分析的來源區分；以 StdOEE 來說就是依照第一章說明的規則來找到需要的分析資料。一般來說都是使用目前定義的標準來做即可，但是，在此之前，已經有使用 OEEReport 的人也許習慣以前的方式，所以在此保留之前的型別，且再提供框架讓其可以建立之前的型別。

3.2 如何建立

目前提供框架，之前使用過的人，只要將抓取資料方式填入框架即可。



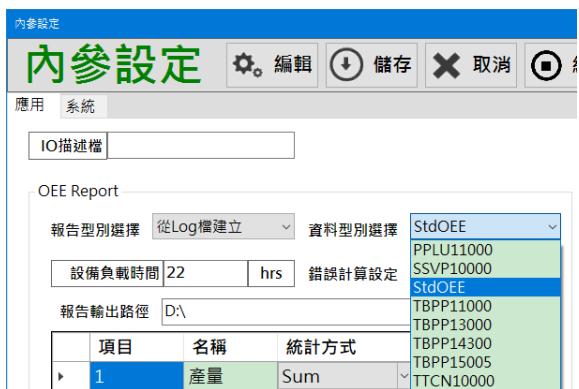
```
namespace TBPP14300
{
    public class TBPP14300 : LogParser
    {
        public override bool ExtractLotInfo(ref List<LotInfo> lotInfos)...
        public override bool ExtractAlarmInfo(ref List<AlarmInfo> alarmInfos)...
    }
}
```

3.3 存放位置

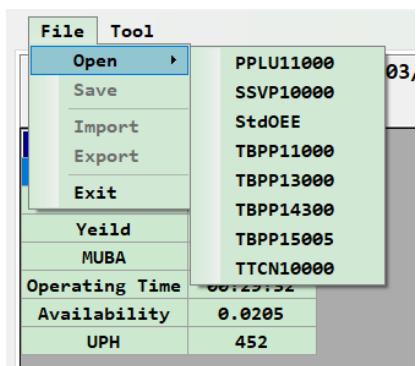
框架建立後，編譯完成會產生一.dll 檔，將之存放於 D:\ProVSimpleProject\ProVSimpleProject\ProVSimpleProject\bin\LocalData\OEEReportLib\即可。之後在內參設定的”資料型別選擇”就會出現；由於此資料夾為提供 OEEReport 用，所以請勿放其他檔案。

3.4 確認完成

完成後名稱會出現在兩個地方，一個在內參設定的資料刑別選擇



一個在 OEEReport 的 File->Open->型別選項中



4 錯誤清單

4.1 內容

錯誤清單讀取方式與前述相同，只不過使用標準型別無須多補程式，抓取資料會從舊有存放地方抓取。而自定義型別則須將方式寫於框架中。

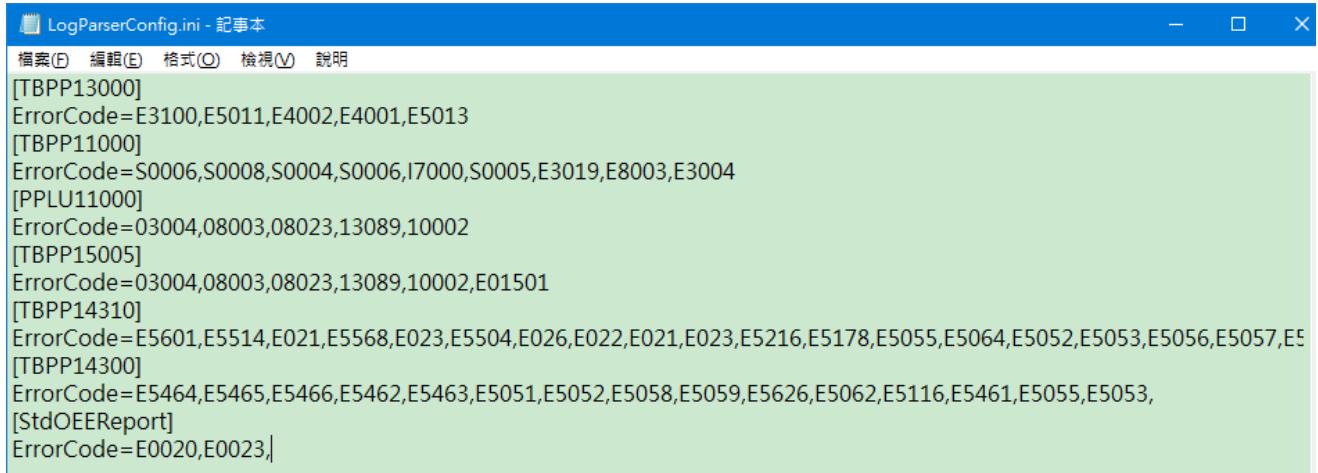
Alarm 目前可依幾項分析:Module、Recipe、ErrorCode、Message

| No | ErrorCode | Alarm Message | Count | Percentage | Cumulative Percentage |
|-------|------------|---|-------|------------|-----------------------|
| Top | Module | [motor2] 踏零時，正轉與反轉極限同時作用 | 29 | 0.5000 | 0.5000 |
| Top | Error Code | [motor1] 驅動器故障 | 8 | 0.1379 | 0.6379 |
| Top | Message | [motor1]ERR_DeviceNotInitial in AlarmReset-getAlarm | 3 | 0.0517 | 0.6897 |
| Top3 | E0020 | [motor1] 踏零時，正轉與反轉極限同時作用 | 3 | 0.0517 | 0.7414 |
| Top4 | E0099 | [motor3]ERR_DeviceNotInitial in Stop | 3 | 0.0517 | 0.7931 |
| Top5 | E0099 | [motor1]ERR_SlaveNotOPState in ReadRealEncPos | 2 | 0.0345 | 0.8276 |
| Top6 | E0099 | [motor1]ERR_SlaveNotOPState in ReadRealPos | 2 | 0.0345 | 0.8621 |
| Top7 | E0099 | [motor1]ERR_DeviceNotInitial in Stop - PRA_DEC | 2 | 0.0345 | 0.8966 |
| Top8 | E0099 | [motor1]ERR_FunctionNotEnable in GetAlarmStatus | 1 | 0.0172 | 0.9138 |
| Top9 | E0099 | [motor1] Communication Broken | 1 | 0.0172 | 0.9310 |
| Top10 | E0011 | [motor1] 馬達命令值與回授值誤差超出允許範圍 | 1 | 0.0172 | 0.9483 |
| Top11 | E0031 | [motor3]ERR_DeviceNotInitial in Stop - PRA_DEC | 1 | 0.0172 | 0.9655 |
| Top12 | E0099 | [motor3]ERR_SlaveNumberErr in GetAlarmStatus | 1 | 0.0172 | 0.9828 |
| Top13 | E0099 | [motor1] 反轉硬體極限 | 1 | 0.0172 | 1.0000 |
| Top14 | E0021 | | | | |

4.2 過濾錯誤

過濾錯誤清單定義在 D:\ProVSimpleProject\ProVSimpleProject\bin\LogParserConfig.ini

如下圖，在個別的機台型號下，填入要排除的錯誤即可；使用標準版的話，只需確認[StdOEEReport]後的 ErrorCore 是否是要排除的，若無需排除，則"="後無需填寫。



```
LogParserConfig.ini - 記事本
[TBPP13000]
ErrorCode=E3100,E5011,E4002,E4001,E5013
[TBPP11000]
ErrorCode=S0006,S0008,S0004,S0006,I7000,S0005,E3019,E8003,E3004
[PPLU11000]
ErrorCode=03004,08003,08023,13089,10002
[TBPP15005]
ErrorCode=03004,08003,08023,13089,10002,E01501
[TBPP14310]
ErrorCode=E5601,E5514,E021,E5568,E023,E5504,E026,E022,E021,E023,E5216,E5178,E5055,E5064,E5052,E5053,E5056,E5057,E5
[TBPP14300]
ErrorCode=E5464,E5465,E5466,E5462,E5463,E5051,E5052,E5058,E5059,E5626,E5062,E5116,E5461,E5055,E5053,
[StdOEEReport]
ErrorCode=E0020,E0023,
```