**BI程式開發注意事項 (可參考Sewing R02)**

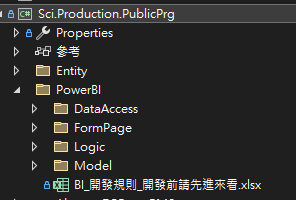
1. 判斷BI的需求性與**PMS耦合性是否過高**或者需要**更彈性的技術(API)**? 假如是請C#共用方式撰寫，不是就看方便性決定，使用舊的方法也是可以(Store Procedure)。

Ex: 某隻R系列報表照抄過來，某隻R系列報表的加總結果。

1. C#的程式請寫在” **Sci.Production.Prg**”底下
   * 畫面寫在FormPage UserControl 底下，使用FlowLayoutPanel來排版，利用ADD方式加入控制項，加入時機點是看傳入的參數是否有值。(第一組日期區間[SDate、EDate]、第二組日期區間[SDate2、EDate2]、單一日期[SDate])
   * PMS邏輯寫在Logic，請依照**PMS程式名稱命名class**，該隻Report邏輯只將有跟BI相關的在移置這裡。

Ex: 假如該隻Report有三份Excel，但這隻BI只要其中一份，只要將這一份移過來就好，其他兩隻就不用動作。

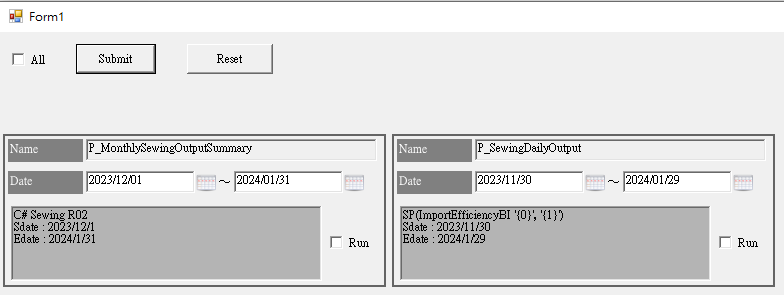
* + BI邏輯寫在DataAccess，命名規則**P\_Import\_{TableName}**，寫入方法用insert update delete不要用Merge，最後需更新BITableInfo，需確認**是否傳回台北要確認**。
  + Model為放置ViewModel位置，建議都使用強行別，這是個好習慣。

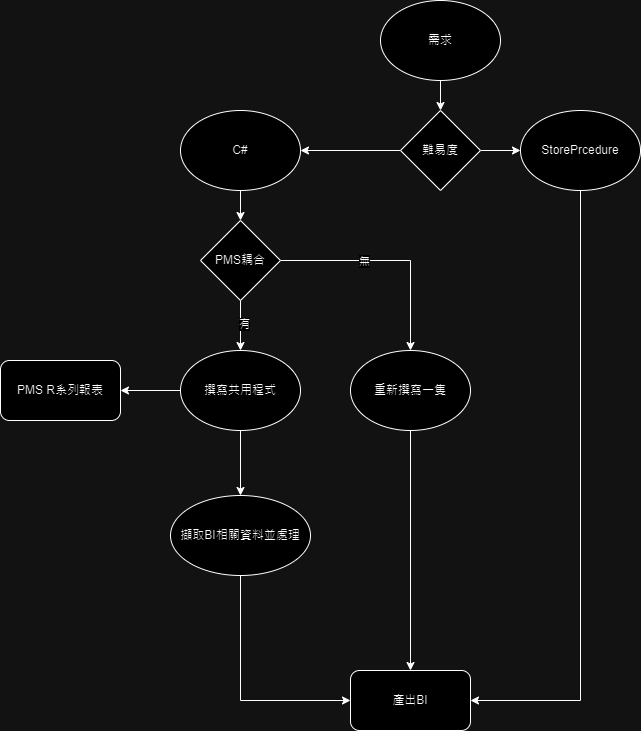


1. 不管是C#還是Store Procedure皆需在BITaskInfo Table寫入資料。
   * P\_MonthlySewingOutputSummary 為C#
   * P\_SewingDailyOutput 為Store Procedure



執行畫面





**BI 開發規則**

1. BI的條件需要由參數傳入，不是寫死在語法內。
2. 初始規則應該跟每日轉的規則一樣，且不需要寫在語法內。

Ex: P\_Import\_ICRAnalysis、Production.dbo.[GetFabricInspLabSummaryReport]

1. BITableInfo需紀錄Table，不是SP
2. 語法中沒用到openquery 就不需要用動態語法。
3. SP名稱統一

Ex: P\_Import\_XXX，但後面的XXX就不要再有'\_'

**BI結果確認**

1. Job Log 組別”PMS” 作業名稱” Factory BI transfer”

