



Trinity5

# Web JFDesigner User Guide

Based on Trinity JCS Lite

v20260130

© 2026 NetPro Information Service Ltd.

# Version History

Trinity JCS Lite

Date.	Author	Description	Chapter
20260130	Angelina	建立文件	

# 目錄

Version History .....	2
目錄 .....	3
Web JFDesigner 介紹 .....	5
Web JFDesigner 使用者介面 .....	5
Web JFDesigner 的基本操作 .....	7
Web JFDesigner 的其他操作 .....	12
作業階層設定(Business Entity 與 Category) .....	16
Business Entity .....	17
Category .....	19
作業階層權限設定 .....	22
權限的種類 .....	22
執行授權設定 .....	23
權限查詢 .....	24
House Keeping 設定 .....	26
Mutex 設定 .....	30
Resource .....	34
Schedule .....	35
Import/Export .....	36
Import/Export Job .....	36
Import/Export Resource .....	46
Import 參數設定 .....	73
Job/Job Step/JobDAG 設定 .....	76
作業(Job)設定 .....	76
作業流程(JobFlow) .....	95
作業條件式流程(JobDAG) .....	107
Job 強化功能設定說明 .....	123
訊息通知(Notification) .....	140
Job Stream 設定 .....	149
Job Dependency 設定 .....	152
Dependent Job 與 Trigger By .....	161
作業步驟(Job Step)設定 .....	163
作業步驟(Job Step)類型介紹 .....	171
Job Step - External Command .....	171

Job Step - SQL Script.....	174
Job Step – FTP .....	178
Job Step – Mail.....	181
Statistic 頁面說明.....	184
Step Postcondition.....	188
Step Variable.....	190
JTextField & JTextArea & SQL Executor 控制字元的行為.....	194
<b>版本控管(VersionControl)介紹 .....</b>	<b>195</b>
功能目的 .....	195
功能操作相依關連.....	196
執行 CheckOut .....	198
執行 CheckIn 新增 Job(Flow) 版本.....	200
執行 ReleaseLock 以釋放 CheckOut 狀態.....	202
開啟 VersionHistory 觀覽歷屆 Job(Flow) 版本 .....	203
開啟 ViewStatus 查看 Job(Flow) 現階狀態.....	204
ForceReleaseLock 強行釋放任一使用者對 Job(Flow) 之 CheckOut 狀態.....	205
執行 RollBack 對已存在的 Job(Flow) 選擇版本回復 .....	206
執行 Compare 對 Job 作版本差異比對 .....	208
Batch Version Control .....	211
<b>作業建立數量查詢.....</b>	<b>214</b>
<b>附錄 A. 工作變數的使用.....</b>	<b>215</b>
西元年工作變數.....	216
民國年工作變數.....	226
時間工作變數 .....	231

# Web JFDesigner 介紹

Trinity Web JFDesigner 是一套能夠設定作業任務和流程任務的工具，透過 Web JFDesigner 的操作介面可以讓使用者輕鬆地利用 Web 頁面選擇欄位搭配各項完成複雜的任務設定，而不需要寫大量繁雜的 XML 設定檔。

此文件詳細解釋 Trinity Web JFDesigner 功能和使用說明。

## Web JFDesigner 使用者介面

開啟 Trinity Home 之後，點選 Web JFDesigner 圖示進入。



### 登入 Web JFDesigner

若 Trinity Home 已登入，點選 Web JFDesigner 進入畫面將自動登入；若 Trinity Home 未登入，則跳出下圖登入畫面。

Log in

Enter your account

Enter your password

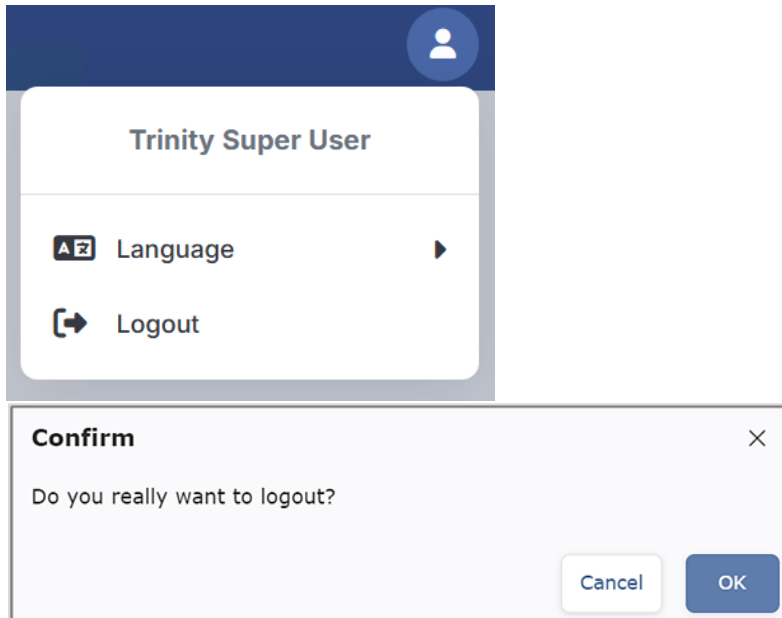
Language

English

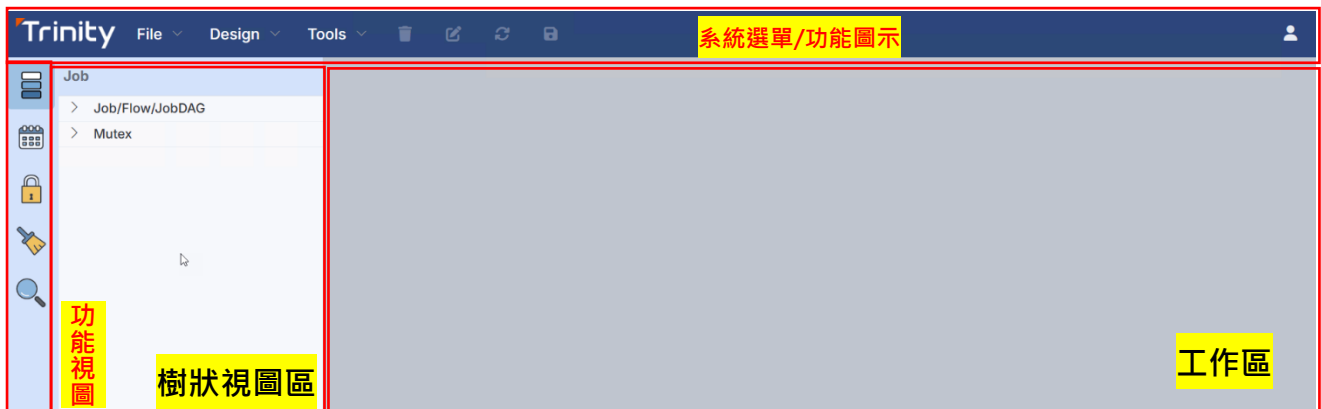
Login

## 登出 Web JFDesigner

點選右上方帳號資訊，選擇「Logout」可登出 Web JFDesigner，執行過程會顯示對話視窗，詢問是否確定要執行登出動作，按下「Ok」會結束與 Trinity 數據庫的連線，而按下「Cancel」則會放棄登出的動作繼續回到 Web JFDesigner 中。



在 Web JFDesigner 中的工作視窗可分成左方功能視圖、樹狀視圖區及右方工作區。於功能視圖指定樹狀視圖區顯示項目，工作區用以顯示所選取物件項目之編輯設定畫面。上方系統選單與功能圖示按鈕可供選擇以便對選取的項目進行操作。



## 結束 Web JFDesigner

點選視窗關閉圖示即可結束 Web JFDesigner 的執行，執行過程會顯示對話視窗，詢問是否確定要離開，按下「離開」會結束 Web JFDesigner 的執行，而按下「取消」則會放棄結束的動作。

# Web JFDesigner 的基本操作

在這一章中，我們會詳述 Web JFDesigner 中的一些基本操作項目。

## 功能視圖的主項目圖示

左邊的功能視圖中有六個主要的項目圖示，它們分別是：

Job、Schedule、Permission、House Keeping、Resource

- Job

此項目下會顯示目前 Job/Flow/JobDAG 設定的 Business Entity 項目、其下的 Category 項目與作業與流程項目，這裡會以 Business Entity -> Category -> Job/Flow 的階層關係來展現作業與流程及 Mutex 項目顯示新增的 Mutex，之後章節會有詳述說明。

- Schedule

此項目顯示目前系統中既存的 Frequency 物件、Working Calendar 物件或 Exclude Frequency 物件；若有自定義的 Frequency Category 存在的話，也會依照階層關係來展現。

- Permission

此項目可對物件進行授權動作，這些物件包含了 Business Entity、Category、Job 及 Mutex，於後續章節詳述。

- House Keeping Rule

此項目下顯示新增的 House Keeping，於後續章節詳述。

- Resource

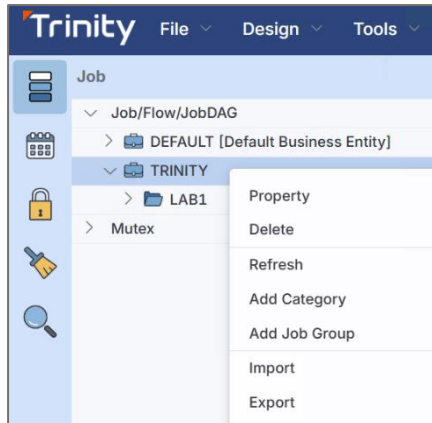
此項目下包含了 Domain、Connection 及 Agent，需在 Admin UI 中進行設定。這些項目在 Web JFDesigner 中僅提供查詢及執行 Resource Export/Import 的功能。

## 系統選單

視窗上方的系統選單與功能圖示隨著在樹狀視圖中所選取的子項目而啟用該子項目所對應的功能，若某些功能在所選取的功能是無法執行的，則會以失效(Disabled)的狀態顯示。

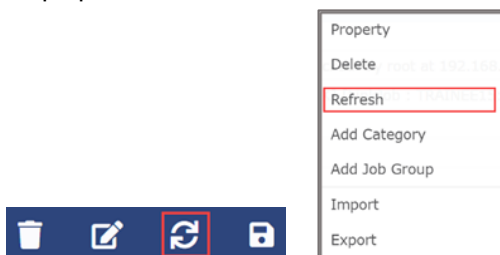
## 使用滑鼠右鍵顯示 Popup 功能選單

可以在欲執行操作的樹狀子項目中按下滑鼠右鍵，此時會顯示一個 Popup 功能選單。此選單部分內容與系統選單及功能圖示重疊。



## 更新樹狀視圖中的物件

當要強制 Web JFDesigner 重新讀取某一項樹狀視圖中的物件子項目時，可以在選取此物件子項目後執行功能圖示中的「Refresh」功能。如此會針對所選取的樹狀物件子項目重新從數據庫中讀取所屬的物件定義並更新在樹狀視圖中。這在多人同時對 Trinity 數據庫進行操作時，透過「Refresh」可立刻看到其他人所做的變更。使用者可以按滑鼠右鍵由 Popup 功能選單中來執行「Refresh」，也可以從功能圖示中點選 Refresh 圖示。

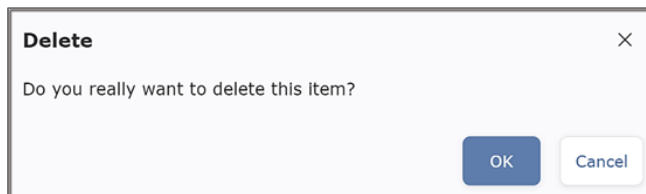


## 刪除樹狀視圖中的物件

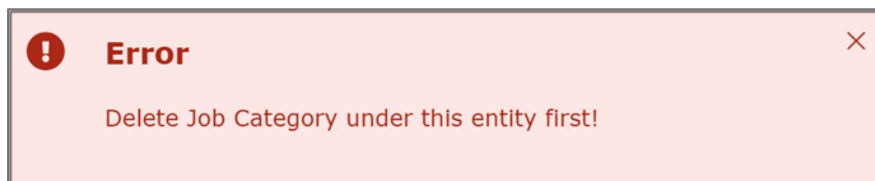
如果要對樹狀視圖中某一項物件子項目進行刪除動作的話，使用者必須先選取想要刪除的物件子項目，按下滑鼠右鍵從 Popup Menu 中選取「Delete」，也可以從功能圖示中直接點選「Delete」的圖示。



點選之後 Web JFDesigner 會顯示一個對話視窗詢問使用者是否確定要刪除此物件項目，如下圖所顯示的對話視窗。按下「Confirm」會執行刪除物件的動作，而按下「Cancel」則會放棄刪除物件的動作。



某些物件可能會因為物件定義的相關性而無法執行刪除動作，例如下圖所顯示的是因為一個 Business Entity 下還包含了一些 Category 項目而顯示的錯誤訊息。使用者無法刪除還包含 Category 的 Business Entity，只有空的 Business Entity 才能被刪除掉。

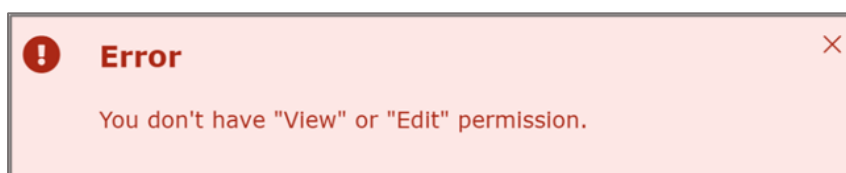


## 編輯樹狀視圖中的物件

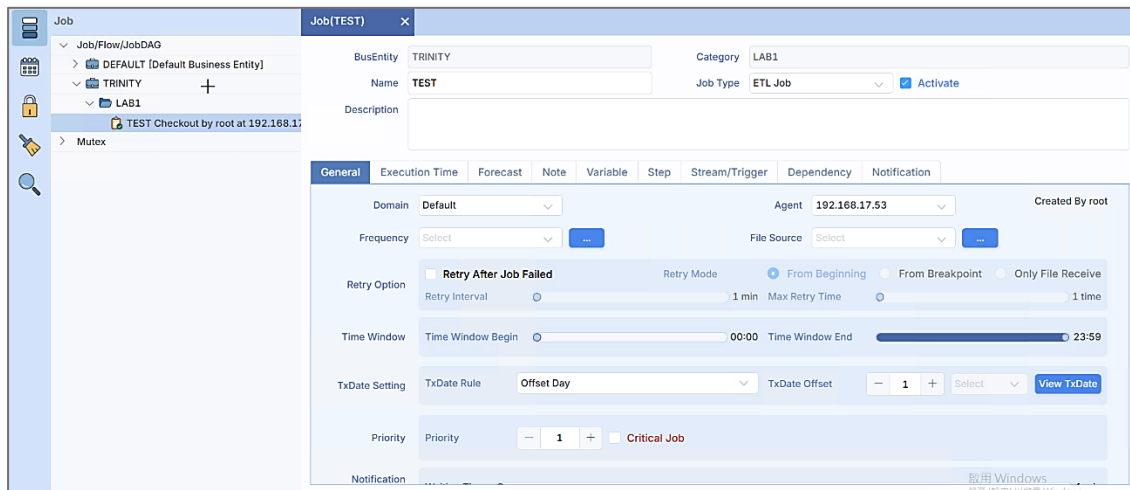
如果要對樹狀視圖中某一項物件子項目進行編輯動作的話，使用者必須先選取想要編輯的物件子項目，按下滑鼠右鍵從 Popup Menu 中選取「Property」，也可以從功能圖示中直接點選「Property」的圖示。



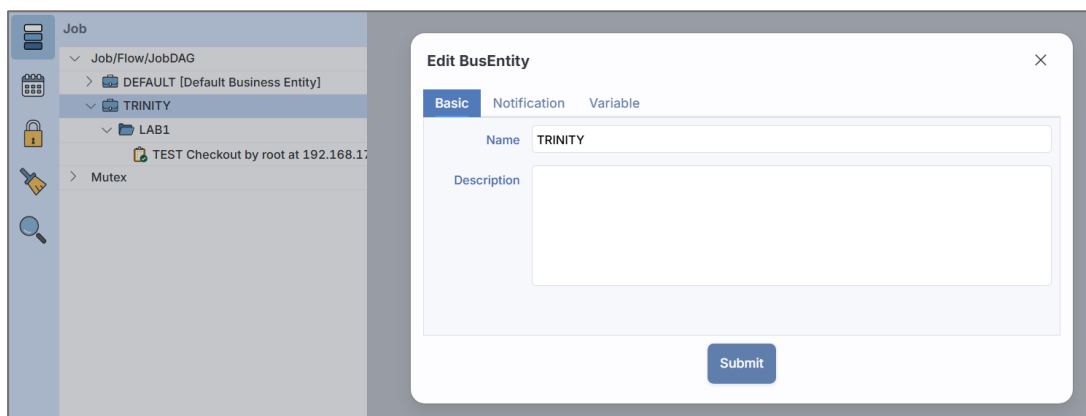
使用者可能會因為權限問題而無法編輯某個物件，例如下圖所顯示的即是對某個作業沒有修改的權限而提示的錯誤警告對話視窗。



當使用者擁有對物件進行修改的權限時，選擇「Property」會顯示關於此物件的詳細定義資訊，接著使用者可以對物件進行修改動作，如下圖的範例顯示：

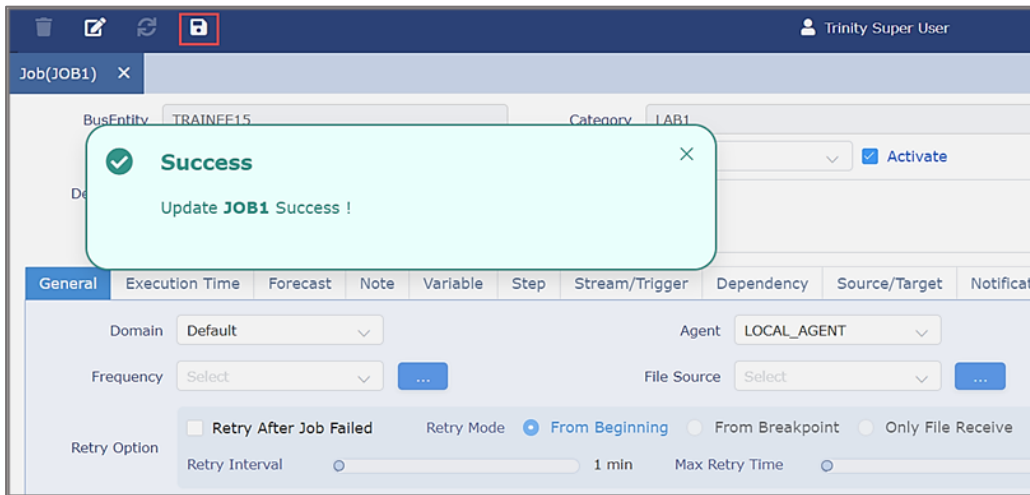


要注意的是只有 Job, Job Step, Flow, JobDAG 的定義才會顯示在右方的工作區中，所以使用者可以同時對多個不同的 Job, Job Step, Flow, JobDAG 進行編輯工作。而其他所有的物件會以對話視窗的方式呈現。以對話視窗呈現的物件編輯工作在同一時間只能進行一項，使用者必須按下「Submit」確認完成編輯工作後，或者按下關閉圖示放棄編輯動作後才能夠在 Web JFDesigner 中進行其他的操作。例如下圖就是開啟一項 Business Entity 物件的編輯對話視窗，使用者必須關閉此對話視窗後才能繼續進行其它的操作。



## 更新工作區中的編輯物件至數據庫

如果所編輯的物件是顯示在工作區中，例如是 Job, Job Step, Flow 或 JobDAG。那麼在完成所進行的編輯工作後要將輸入的資訊更新到 Trinity 的數據庫中的話，可選擇功能圖示中的「Update」圖示或是搭配鍵盤快捷鍵「Ctrl+S」來執行。當更新動作成功時會顯示一個對話視窗告訴使用者物件更新動作已經成功。例如下圖就是對一項 Job 物件執行更新動作後所顯示的畫面：

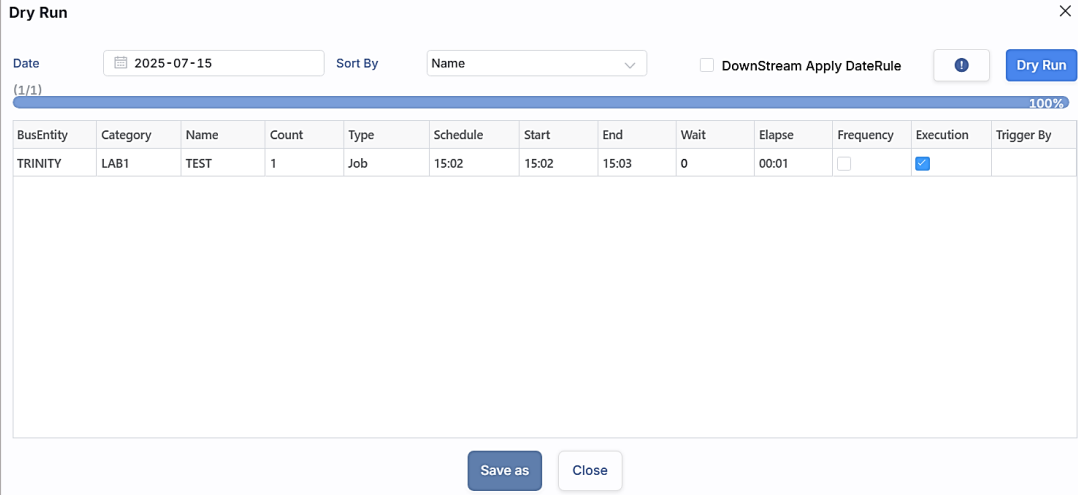


要注意的是當工作區中有多個工作頁面同時存在時，所執行的更新動作只會針對目前所選取的工作頁面進行更新而已，並不會對所有開啟中的工作頁面都進行更新動作。而如果要放棄已做的編輯動作不想更新的話，只要選擇每個工作頁面標題中的關閉按鈕就可以結束此物件的編輯動作而不影響到現存在數據庫中的物件定義。

# Web JFDesigner 的其他操作

## 作業演練(Dry Run)

利用 Dry Run 功能可指定日期演練作業，將當日會自動啟動的排程作業依指定排序列出明細並計算可能執行與完成時間。選擇功能選單的「Tools」->「Dry Run」，即會開啟下方視窗。(註：只有具備 root 和 administrator 權限帳號的使用者才可以執行，且不能授權給其他帳號的使用者去做執行)



The screenshot shows a 'Dry Run' dialog box with the following details:

- Date: 2025-07-15
- Sort By: Name
- DownStream Apply DateRule:
- Buttons: Dry Run, Save as, Close

BusEntity	Category	Name	Count	Type	Schedule	Start	End	Wait	Elapse	Frequency	Execution	Trigger By
TRINITY	LAB1	TEST	1	Job	15:02	15:02	15:03	0	00:01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

- Date：輸入欲演練的日期，圖示可開啟萬年曆
- Dry Run：依日期演練排程作業
- Sort By：Dry Run 結果明細的排序方式，包括名稱、開始執行時間或排程時間
- Save As：將列出明細另存為 EXCEL

## 搜尋作業(Search Job)

利用 Search Job 功能可在眾多作業中搜尋到欲搜尋的作業。選擇功能選單的「Tools」->「Search Job」，即會開啟下方視窗。

BusEntity	Category	Job Name	Job Activate	Step Name	Step Activate	Description
DEFAULT	FUND	TEST	<input checked="" type="checkbox"/>		<input type="checkbox"/>	

- Job Name：輸入欲搜尋的作業名稱
- Text：輸入欲搜尋作業中 Script 之內容或關鍵字
- Code Convert：欲搜尋的 Script 內容中包含特殊符號，例如：
- Search：列出搜尋到的作業項目
- Filter：篩選條件視窗

- Export：匯出搜尋作業項目
- Record Count：搜尋到的作業數量

## 新增變數(Edit/View Variable)

1. 新增 Business Entity Variable 及 Category Variable :  
於 Entity/Category 上點擊兩下或按右鍵點選「Property」, 上方選單選擇「Variable」頁籤, 於對話視窗中編輯內容即可。

**Edit BusEntity**

Basic Notification **Variable**

Variable Name	Variable Value

Submit

**Edit Category**

Basic Notification **Variable**

Variable Name	Variable Value

Submit

2. 新增 Job Variable :  
開啟 Job 設定頁面, 於 Variable 頁籤中設定變數

**\*Job(TEST)**

BusEntity: TRINITY      Category: LAB1

Name: TEST      Job Type: ETL Job       Activate

Description:

General Execution Time Forecast Note **Variable** Step Stream/Trigger Dependency Notification

Variable Name	Variable Value

3. 變數尋找的優先序為 Job 層級 > Category 層級 > Entity 層級 > Domain 層級, 所以當 Job 與 Domain 中都有相同名稱的變數時, 會以 Job 層的變數為主。

## 於 Step 中抓取資料文件名稱及控制檔名稱

在執行作業時，若設定 File Source 自動接收指定的資料文件，則後續在 External Command step 中可抓取資料文件名稱及控制檔名稱。

### External Command

- 透過 File Source 自動接收指定的資料文件後，在執行 External Command 時，會將此批次作業所接收的所有資料文件的目前路徑，記錄在一個清單檔案中當成執行時期的參數傳遞給所要執行的指令，此清單檔案會放在最後一個參數 (包含完整路徑及清單檔名)。
- 此清單預設只記錄資料文件的路徑及檔名，每一個資料文件名稱會單獨以一行表示。
- 如果要在清單檔案中同時包含資料文件名稱與控制文件名稱，則必須要將 jcsagent.conf 中的 FILESOURCE\_OUTPUTLIST\_APPEND 設為 true，如此才會將資料文件與控制文件同時放入清單檔案中

```
#=====
#Used for filesource with external command step
#-----
#FILESOURCE_OUTPUTLIST_APPEND_CONTROLFILE
# filesource output file list has append control file name
#exp:
# False : file1.txt
#         file2.txt
# True  : file1.txt controlfile.txt
#         file2.txt controlfile.txt
#=====
FILESOURCE_OUTPUTLIST_APPEND_CONTROLFILE=true
```

## 設定 Job Step 寫出的 Log 檔案上限

新增參數，可以設定 Job Step 寫出的 Log 檔案上限，於 jcsagent.conf 中增加下列參數，表示當 Log 檔一旦滿 100 MB，後面收到的 Log 則會直接忽略。

- 做法：
  - 修改 JCSAgent\cfg\jcsagent.conf，將預設的 200(MB) 更改為 100(MB)
  - JCS\_AGENT\_OUTPUTLOG\_FILE\_MAX\_SIZE=100
  - 存檔後重啟 JCSAgent 服務

# 作業階層設定(Business Entity 與 Category)

在本章，我們會詳述在 Web JFDesigner 中設定 Business Entity 與 Category 的工作方式。

在能夠進行 Job 的新增工作之前，使用者必須先將 Trinity 中所管理的 Business Entity 與 Category 先設定好，才能夠在 Category 下來新增要執行的 Job。

Trinity 在作業的階層管理上採用的三層架構，第一層為 Business Entity，第二層為 Category，第三層才是 Job 與 Job Flow。在 Trinity 中可以設定多個 Business Entity 來進行大類管理，而一項 Business Entity 下可以包含多個 Category，同樣地，一項 Category 項目下可以包含多個 Job 與 Job Flow。

在同一個 Business Entity 下的 Category 名稱必須是唯一而不能重複的，但不同的 Business Entity 中是可以擁有相同的 Category 名稱。

Job 名稱在同一個 Category 中也是不能重複的，不同 Category 下的 Job 才允許擁有同樣的名稱。

## Business Entity

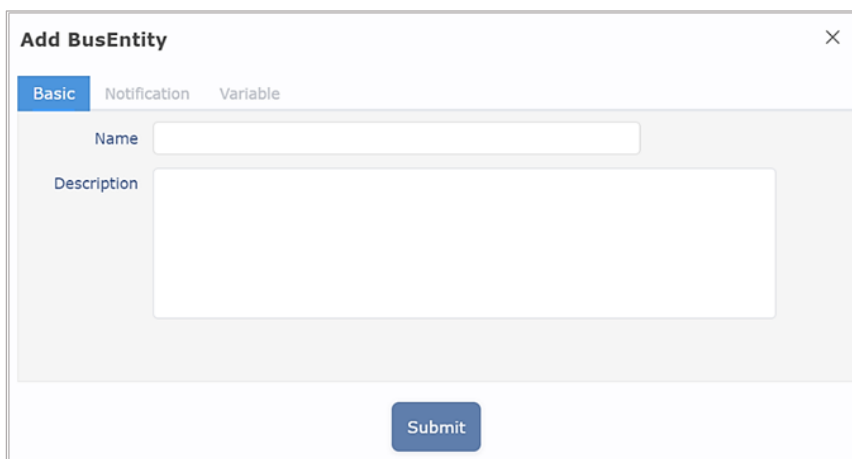
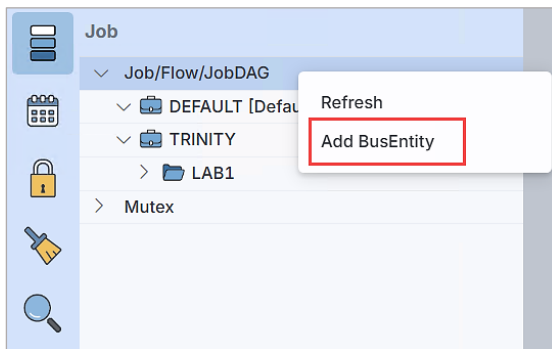
Business Entity 所代表的意義可以是一個公司或是所謂的事業單位，甚至也可以是一個部門，其目的是對於所執行的作業進行大類別的分類管理。當 Trinity 所管理的是整個集團中的所有工作時，那麼 Business Entity 代表的可能是集團底下的各個子公司。如果 Trinity 所管理的是只是一家公司中的作業，那麼 Business Entity 代表的可能就只是部門意義了。

只要能夠清楚地定義出 Business Entity 所代表的分類意義，Trinity 並不會強制要求 Business Entity 一定要是何種類型。

使用者必須要是 Trinity Super User(root)，或是具有 Administrator 角色的使用者才能對 Business Entity 進行新增/修改/刪除的動作。

### 新增 Business Entity

於樹狀視圖中選擇 Job/Flow/JobDAG 的項目，點選右鍵從 Popup Menu 中選取「AddBus Entity」，於出現的對話視窗當中輸入 Entity 名稱與說明，如下圖範例：

A screenshot of the 'Add BusEntity' dialog box. The dialog has a title bar with a close button. It features three tabs: 'Basic' (selected), 'Notification', and 'Variable'. Under the 'Basic' tab, there is a 'Name' text input field and a larger 'Description' text area. At the bottom center of the dialog is a blue 'Submit' button.

按下「Submit」按鈕會將所輸入的設定更新至 Trinity 數據庫中。

## 修改 Business Entity

於樹狀視圖中選擇要進行修改的物件子項目，滑鼠左鍵點擊兩下、點選上方編輯功能圖示或物件子項目上按下滑鼠右鍵從 Popup Menu 中選取「Property」。執行後會出現類似先前新增 Business Entity 的對話視窗可供使用者輸入修改的資訊。修改完畢後按下「Submit」按鈕會將所輸入的設定更新至 Trinity 數據庫中。

## 刪除 Business Entity

於樹狀視圖中選擇要進行刪除的物件子項目，點選上方刪除功能圖示或在物件子項目上按下滑鼠右鍵從 Popup Menu 中選取「Delete」。請參考基本操作中的關於刪除動作的說明。

注意：當 Business Entity 還包含有 Category 項目時，無法對它進行刪除的操作。

## Category

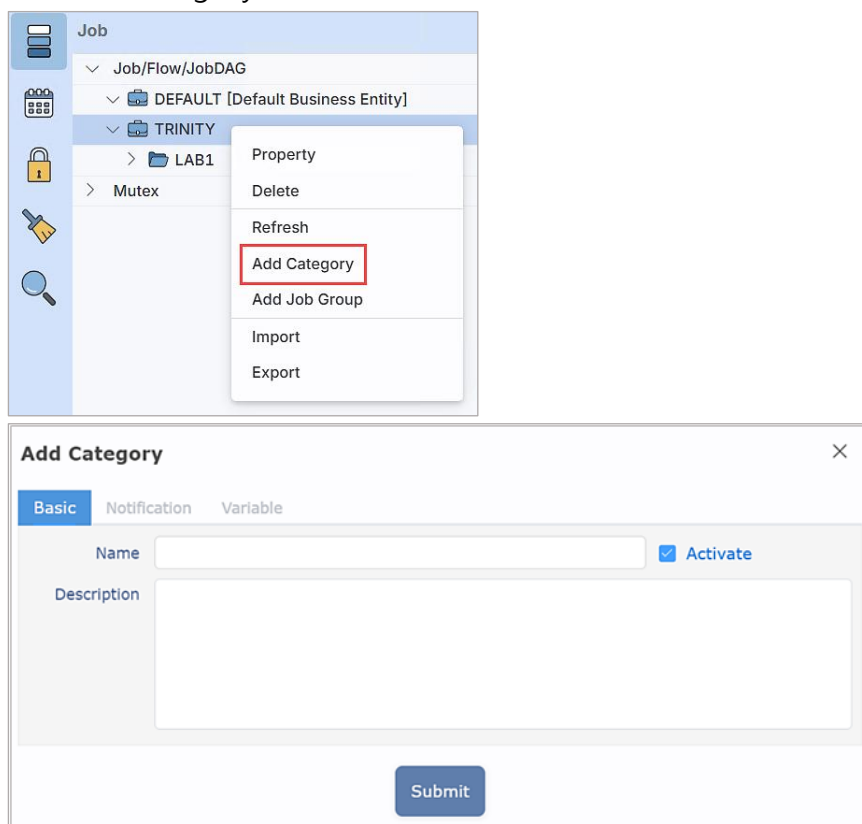
作業類別是用來對作業做進一步的分類管理。舉例來說使用者可能會以作業的類型來進行類別分類，像是「LOADING」用以存放數據載入的作業，「TRANSFORM」用以存放數據轉換的作業，「OLAP」是用已存放數據分析的作業，而「REPORT」則是用已存放資訊報表的作業。

或者使用者可以依照作業的相關性來對作業進行分類管理，例如以「FINANCE」類別來集中管理財務系統相關的作業，而以「SALES」類別來集中管理銷售系統相關的作業。

只要能具體定義出類別的管理意義，Trinity 並沒有強制對 Category 的命名規則做任何的限制。

### 新增 Category

於樹狀視圖中選擇要加入 Category 的 Business Entity 物件項目，點選右鍵從 Popup Menu 中選取「AddCategory」，輸入要新增的 Category 名稱、此 Category 的有效狀態與關於此 Category 的說明等，例如下圖中所呈現的範例：



按下「Submit」按鈕會將所輸入的設定更新至 Trinity 數據庫中。

## 修改 Category

於樹狀視圖中選擇要進行修改的物件子項目，滑鼠左鍵點擊兩下、點選上方編輯功能圖示或物件子項目上按下滑鼠右鍵從 Popup Menu 中選取「Property」。執行後會出現類似先前新增 Category 的對話視窗可供使用者輸入修改的資訊。修改完畢後按下「Submit」按鈕會將所輸入的設定更新至 Trinity 數據庫中。

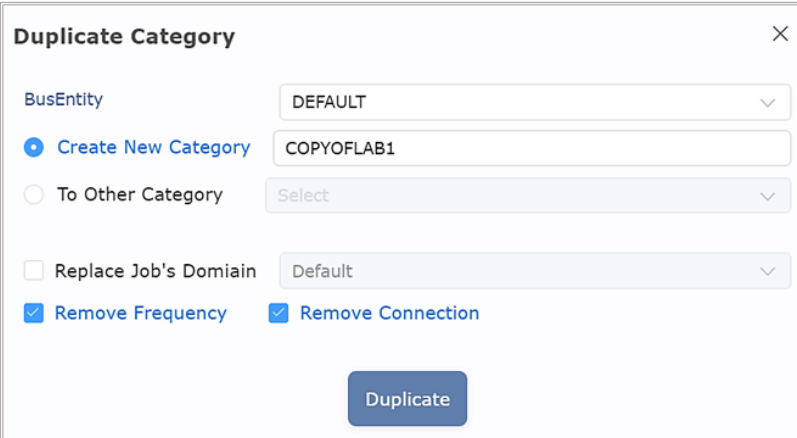
## 刪除 Category

於樹狀視圖中選擇要進行刪除的物件子項目，點選上方刪除功能圖示或在物件子項目上按下滑鼠右鍵從 Popup Menu 中選取「Delete」。請參考基本操作中的關於刪除動作的說明。

**注意：**當 Category 還包含有 Job 項目時，使用者是無法對它進行刪除的操作。

## 複製 Category

使用者可以透過複製 Category 的方式來節省設定作業的時間。於樹狀視圖中選擇要進行複製的 Category 物件子項目，再從功能選單中選擇「Design」->「Duplicate Category」功能來進行複製。執行後會出現一個對話視窗詢問使用者要進行複製的選項，例如下圖中所顯示的範例：



Duplicate Category

BusEntity: DEFAULT

Create New Category: COPYOFLAB1

To Other Category: Select

Replace Job's Domain: Default

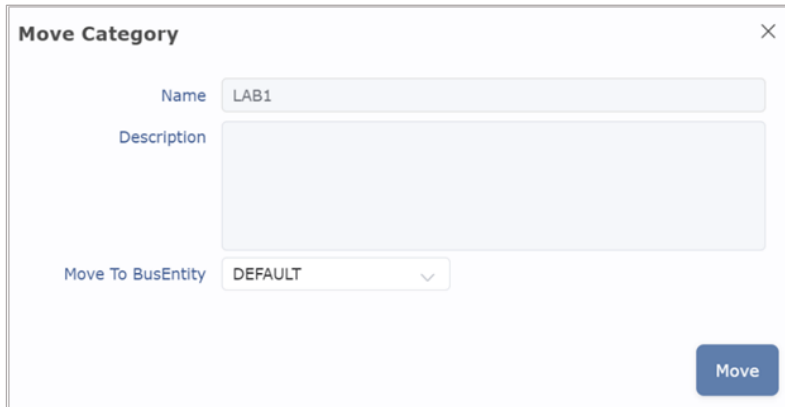
Remove Frequency  Remove Connection

Duplicate

此對話視窗中的「To Other Category」選項是用來指定是否要將作業複製到其他的 Category 中。如果勾選了此選項則需進一步的選擇要搬移過去的 Business Entity 與 Category。選擇完畢後可以按下「Duplicate」按鈕來執行複製的動作，Flow、Job Stream 及 Job Dependency 不會一併複製。

## 移動 Category 到其它的 Business Entity

有時候使用者可能會需要將一項 Category 從一個 Business Entity 中搬移到另外一個 Business Entity 中。於樹狀視圖中選擇要進行搬移的 Category 物件子項目按下滑鼠右鍵從 Popup Menu 中選取「Move Category」。執行後會出現一個對話視窗詢問使用者要將此 Category 搬移到那一個 Business Entity 中，例如下圖中所顯示的範例：



The screenshot shows a dialog box titled "Move Category" with a close button (X) in the top right corner. The dialog contains the following fields:

- Name:** LAB1
- Description:** (Empty text area)
- Move To BusEntity:** DEFAULT (Dropdown menu)
- Move:** (Blue button)

從「Move to Entity」的下拉選單中選擇要搬移過去的目的 Business Entity 之後，使用者可以按下「Move」按鈕會來執行搬移的動作。

# 作業階層權限設定

在這一章中，我們會詳述在 Web JFDesigner 中設定權限的工作方式。

## 權限的種類

在 Trinity 中可供設定的權限種類及說明如下：

權限	說明
View	是指使用者是否對此物件有查看的權限。
Add	是指使用者是否對此物件有新增 Job 及 Job Step 的權限。
Edit	是指使用者是否對此物件有進行修改的權限。
Delete	是指使用者是否對此物件有刪除的權限。
Run	是指使用者是否能在 TaskConsole 介面中對此物件有執行工作的權限。
Rerun	是指使用者是否能在 TaskConsole 介面中對此物件在執行工作失敗後有重新執行的權限。
Grant	是指使用者是否對此物件有再授權給其他使用者的權限。使用者必需對物件有 Grant 的權限後才能把自己擁有對物件的權限再授權給其他的使用者。
Import	是指使用者是否能把 Job 輸入
Export	是指使用者是否能把 Job 輸出

Trinity 中的使用者類型有兩種類型：

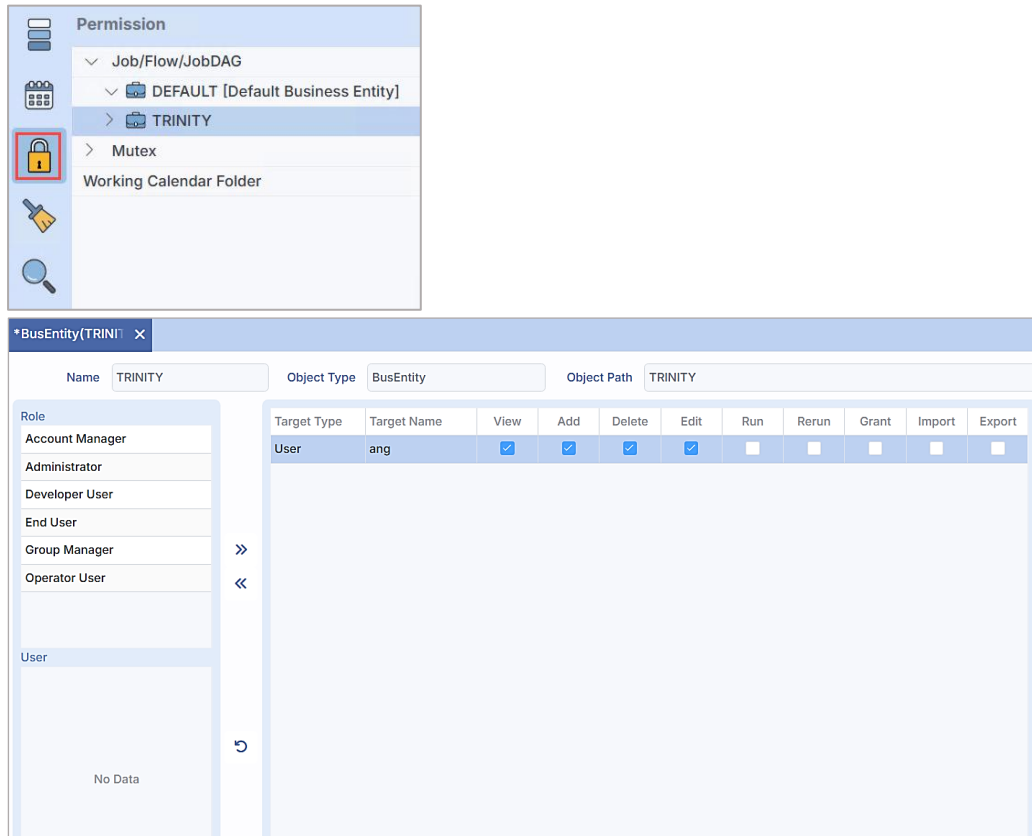
- 超級使用者(Super User)：超級使用者只會有一個，其帳號名稱為 root。超級使用者 root 擁有執行所有工作的權限，使用者不需要對 root 進行授權。
- 一般使用者：超級使用者之外的其他的使用者都是一般使用者。一般的使用者必須要授與適當的權限以便能夠對某個 Business Entity 或 Category，甚至單獨到特定的 Job 進行操作。

為了方便對權限的管理，在使用者的帳號之外，Trinity 也提供的角色(Role)的管理，系統內定會有六個預設的角色，「Administrator」、「Developer User」、「Operator User」、「End User」、「Account Manager」以及「Group Manager」。使用者也可以自行加入欲管理的角色。在實務上，通常會先定義一些角色，並事先對這些角色做好授權的動作，如此一來，在後續如有新增的使用者時，可以直接將使用者加入到角色中以取得適當的權限，而無需對使用者再次地進行繁瑣的授權動作。

一個使用者可以同時被加入到多個角色中，所擁有的權限會是所有角色權限的總合。有關於角色及使用者的建立與管理必須在 Admin UI 中進行，相關資訊請參考「Trinity V5.0 Admin UI User Guide」文件說明。

## 執行授權設定

在 Web JFDesigner 中可允許對物件進行授權動作，這些物件包含了 Business Entity、Category 以及 Job。於功能視圖中選擇權限圖示，滑鼠左鍵雙擊欲授權的項目，即可開啟 Grant Permission 頁面。



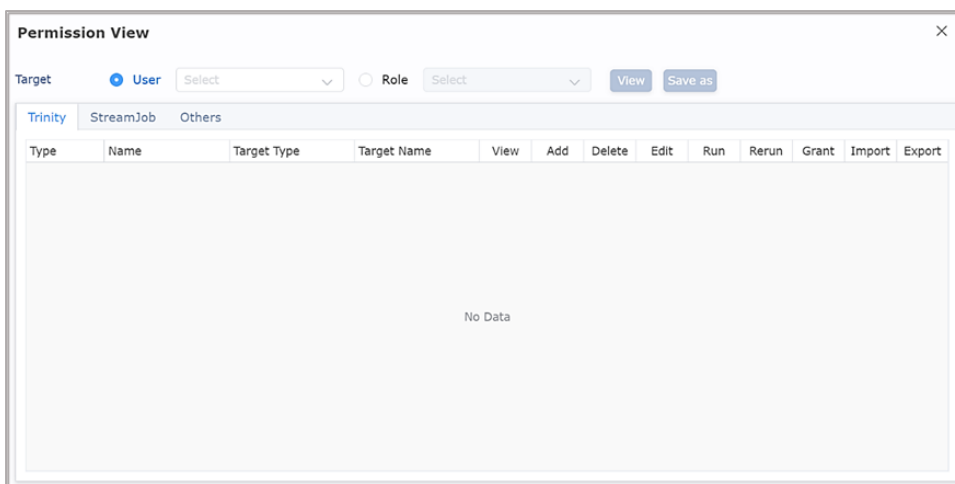
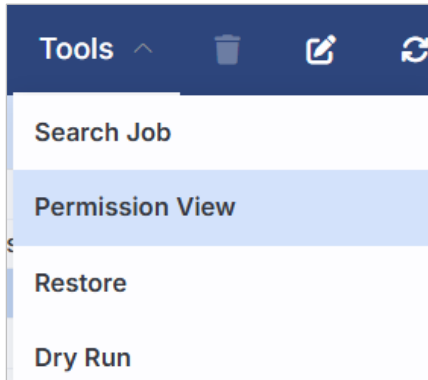
在此權限的設定視窗中有三個顯示區塊，左上方是顯示目前系統中有的角色，左下方則是顯示目前系統中有的使用者帳號，而在右方的區塊則是顯示目前此物件的授權狀況。使用者可以選擇要把角色或使用者帳號加入到權限設定中，或者從目前權限移除掉。

設定權限完畢後按下「Ctrl+S」或儲存功能圖示會將所輸入的設定更新至 Trinity 數據庫中。



## 權限查詢

在 Web JFDesigner 功能選單中「Tools」->「Permission View」可查詢使用者及角色的權限內容。



Target 選擇 User 則可查詢 User 權限，點選下拉選單選擇對象。Target 選擇 Role 則可查詢 Role 權限，於下拉選單中選擇對象。

Target 選擇完畢，按下「View」可將查詢結果顯示於下方表格。按下「Save as」可將查詢結果另存於本機端。

- Trinity 頁面：檢閱相關對應的作業 ( Job ) 使用權限。

**Permission View** ✕

Target  User  Role trainee15 Account Manager View Save as

**Trinity** StreamJob Others

Type	Name	Target Type	Target Name	View	Add	Delete	Edit	Run	Rerun	Grant	Import	Export
User	trainee15	BusEntity	RRR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User	trainee15	BusEntity	TEST1111111	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User	trainee15	BusEntity	TRAINEE1515	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User	trainee15	BusEntity	TRAINEE15TEST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User	trainee15	BusEntity	TRAINEE15帳號	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
User	trainee15	BusEntity	基礎	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
User	trainee15	BusEntity	測試JOB	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User	trainee15	Category	PM::MAIL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- 「Other」 頁面：檢視其他相關權限。

**Permission View** ✕

Target  User  Role trainee15 Account Manager View Save as

Trinity StreamJob **Others**

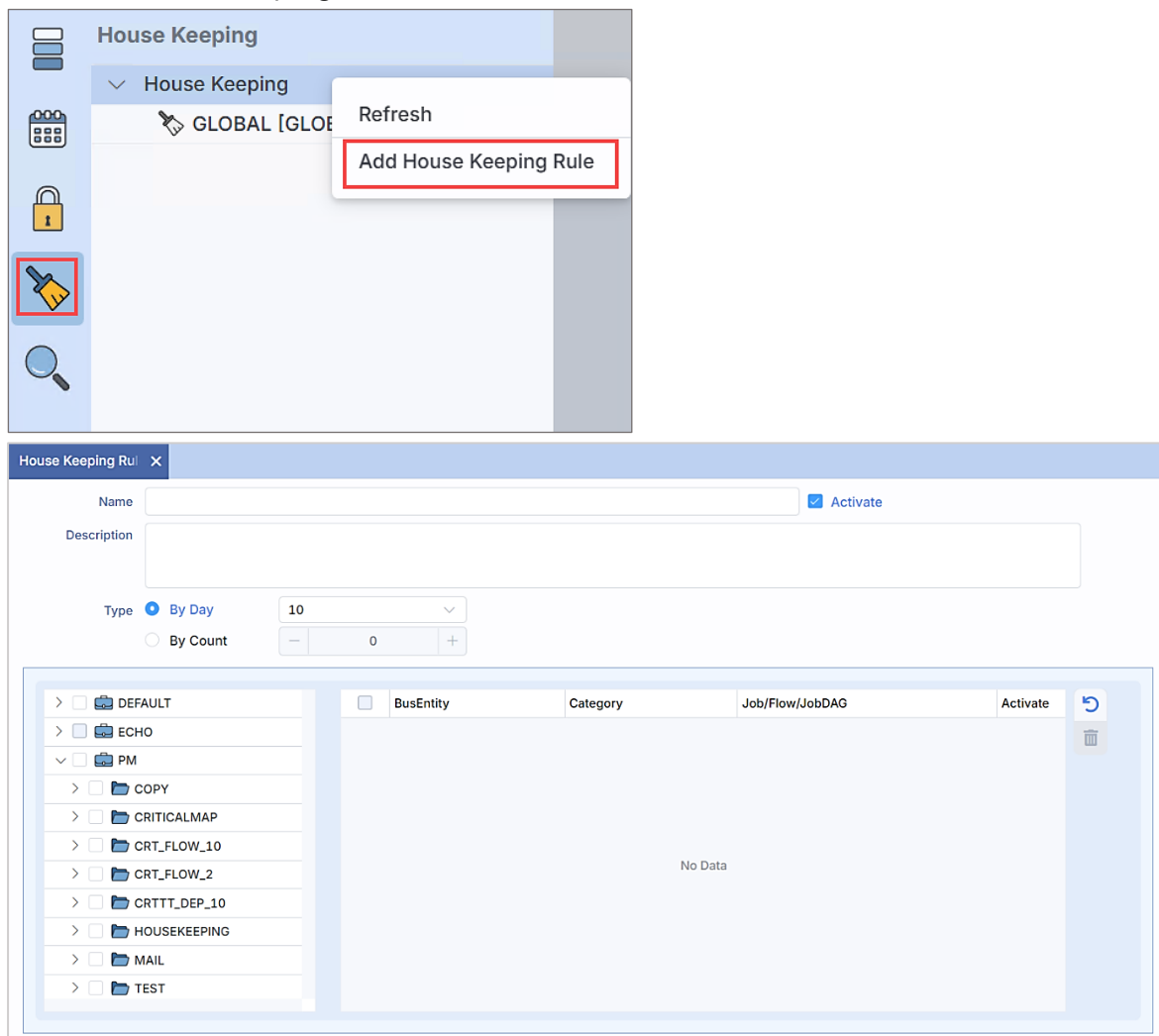
Type	Name	Target Type	Target Name	View	Add	Delete	Edit
Role	Administrator	Pluginlet	Pluginlets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	Alias	All Alias	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	Connection	All Connection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	Domain	All Domain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	BusEntity Variable	All Entity Variable	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	Extension Rule	All Extension Rule	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	File Source	All File Source	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Role	Administrator	Frequency	All Frequency	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

# House Keeping 設定


在 Task Console 中執行作業時會將任務的屬性和資訊記錄於 log 中。透過對特定的 Job 設定 House Keeping Rule，可以定期清除這些 log 資料。

## 新增 House Keeping Rule

功能視圖中點選 House Keeping 圖示，於 House Keeping Rule 上方點選滑鼠右鍵，點擊「Add House Keeping Rule」



- Name : House Keeping Rule 名稱
- Type :
  - By Day : log 保留天數
  - By Count : 同一 Job 的 task log 累計次數
- 新增：於左方樹狀圖勾選目標(Job, Flow, JobDAG,Category or Business Entity)，即可新增

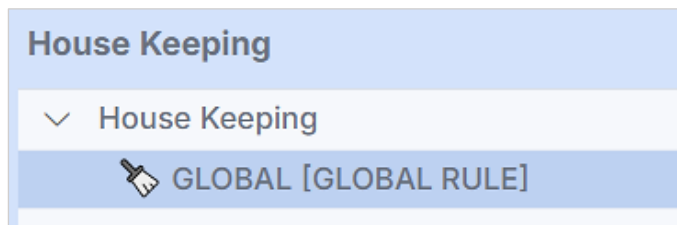
- 刪除：於右方欄位中，勾選欲刪除項目後點選刪除圖示  即可刪除
- House Keeping Rule 套用原則：
  - 同一目標(Job, Flow, JobDAG, Category or Business Entity)只可套用一種 rule。
  - 若 Job 包含在 Category 中，則以 Job 套用的規則執行。若 Category 包含在 Business Entity 中，則以 Category 套用的規則執行。

## 新增 Housekeeping Activity Log

當執行 Housekeeping 時，會在 JCSServer\log 目錄下產生一組 JCS\_HouseKeeping\_YYYYMMDD.log 記錄檔，以利維運人員知曉 HouseKeeping 的執行進度與內容（每日一份檔案，以執行作業開始時間計算）

## 設定 Global Rule

如沒有針對特定 Job 設定 House Keeping Rule，則可使用 Global rule。Global rule 套用於所有未設定 House Keeping rule 的 Job。在左方的樹狀視圖中選擇 House Keeping 項目中的 Global 項目，設定要保留的天數，將 Activate 打勾就可啟用 Global rule（預設為 Inactivate）。




The 'Set Global Rule' dialog box contains the following fields:

- Name:** GLOBAL  Activate
- Description:** GLOBAL RULE
- Type (By Day):** 10

## 修改 House Keeping Rule

於樹狀視圖中選擇要進行修改的物件子項目，在物件子項目上按下滑鼠右鍵從 Popup Menu 中選取「Property」或滑鼠雙擊兩下。執行後會出現類似先前新增 House Keeping Rule 的對話視窗可供使用者輸入修改的資訊。修改完畢後按下「OK」按鈕會將所輸入的設定更新至 Trinity 數據庫中。

## 刪除 House Keeping Rule

於樹狀視圖中選擇要進行刪除的物件子項目，在物件子項目上按下滑鼠右鍵後，從 Popup Menu 中選取「Delete」。請參考基本操作中的關於刪除動作的說明。

## House Keeping 時間

啟動 HouseKeeping 預設時間為 00:00，可於 jcserver.conf 中設定。(調整後須重啟 JCServer 服務)

```
#=====
# House Keeping execution time
#=====
# set time to start housekeeping job, it is hh:mm format.
HOUSEKEEPING_TIME=00:00
```

## House Keeping 結果

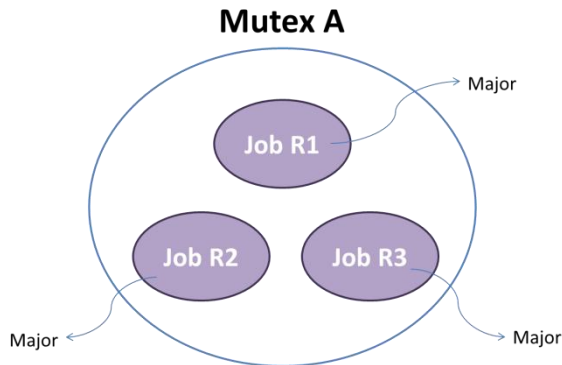
執行結果紀錄於 <JCServer>/log/ HouseKeeping\_yyyymmdd.log，示意如下：

```
[04/01 12:00:04.343] com.netpro.jcs.server.housekeeping.c : Start clean job/flow by time.
[04/01 12:00:04.346] com.netpro.jcs.server.housekeeping.a : Found job/flow by time rules (0) .
[04/01 12:00:04.351] com.netpro.jcs.server.housekeeping.a : Found category by time rules
job(0), flow(0) .
[04/01 12:00:04.356] com.netpro.jcs.server.housekeeping.a : Found busentity by time rules
job(0), flow(0) .
[04/01 12:00:04.357] com.netpro.jcs.server.housekeeping.a : Found need delete task(0).
[04/01 12:00:04.358] com.netpro.jcs.server.housekeeping.a : Found delete tasksteplog(0).
[04/01 12:00:04.358] com.netpro.jcs.server.housekeeping.c : HouseKeeping spent 14 miliseconds
to clean job/flow by time.
[04/01 12:00:04.359] com.netpro.jcs.server.housekeeping.c : Start clean job/flow by count.
[04/01 12:00:04.360] com.netpro.jcs.server.housekeeping.a : Found job/flow by count rules (0) .
[04/01 12:00:04.363] com.netpro.jcs.server.housekeeping.a : Found category by count rules
job(0), flow(0) .
[04/01 12:00:04.371] com.netpro.jcs.server.housekeeping.a : Found busentity by count rules
job(124), flow(4) .
[04/01 12:00:04.582] com.netpro.jcs.server.housekeeping.a : Found delete task(2032).
[04/01 12:00:06.034] com.netpro.jcs.server.housekeeping.a : Found delete tasksteplog(2257).
[04/01 12:00:06.530] com.netpro.jcs.server.housekeeping.a : Delete taskstepoutputlog(2253).
[04/01 12:00:06.889] com.netpro.jcs.server.housekeeping.a : Delete taskstepfulllog(2184).
[04/01 12:00:07.256] com.netpro.jcs.server.housekeeping.a : Delete tasksteplog(2257).
[04/01 12:00:07.660] com.netpro.jcs.server.housekeeping.a : Delete taskjoblog(2044).
[04/01 12:00:07.901] com.netpro.jcs.server.housekeeping.a : Delete receivefilelog(68).
[04/01 12:00:11.414] com.netpro.jcs.server.housekeeping.a : Delete
taskstatusreasonhistory(10316).
[04/01 12:00:11.761] com.netpro.jcs.server.housekeeping.a : Delete tasklog(2032).
[04/01 12:00:11.762] com.netpro.jcs.server.housekeeping.c : HouseKeeping spent 7403 miliseconds
to clean job/flow by count.
[04/01 12:00:11.762] com.netpro.jcs.server.housekeeping.c : Start clean global job/flow log.
```

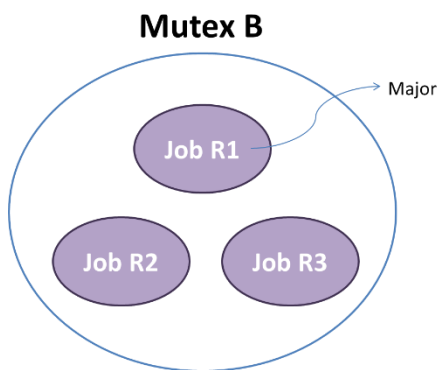
```
[04/01 12:00:11.766] com.netpro.jcs.server.housekeeping.a : Found global rules job(36),
flow(2) .
[04/01 12:00:11.794] com.netpro.jcs.server.housekeeping.a : Found need delete task(191).
[04/01 12:00:11.907] com.netpro.jcs.server.housekeeping.a : Found delete tasksteplog(1720).
[04/01 12:00:12.214] com.netpro.jcs.server.housekeeping.a : Delete taskstepoutputlog(1720).
[04/01 12:00:12.457] com.netpro.jcs.server.housekeeping.a : Delete taskstepfulllog(966).
[04/01 12:00:12.723] com.netpro.jcs.server.housekeeping.a : Delete tasksteplog(1720).
[04/01 12:00:12.771] com.netpro.jcs.server.housekeeping.a : Delete taskjoblog(936).
[04/01 12:00:12.796] com.netpro.jcs.server.housekeeping.a : Delete receivefilelog(19).
[04/01 12:00:12.831] com.netpro.jcs.server.housekeeping.a : Delete tasklog(191).
[04/01 12:00:13.057] com.netpro.jcs.server.housekeeping.a : Delete
taskstatusreasonhistory(694).
[04/01 12:00:13.058] com.netpro.jcs.server.housekeeping.c : HouseKeeping spent 1295 miliseconds
to clean global job/flow log.
```

# Mutex 設定

在 Task Console 中執行作業時會檢查同一 Mutex 中同一時間只會有一個 Job 執行 (預設為全勾選 Major) · 如下圖。

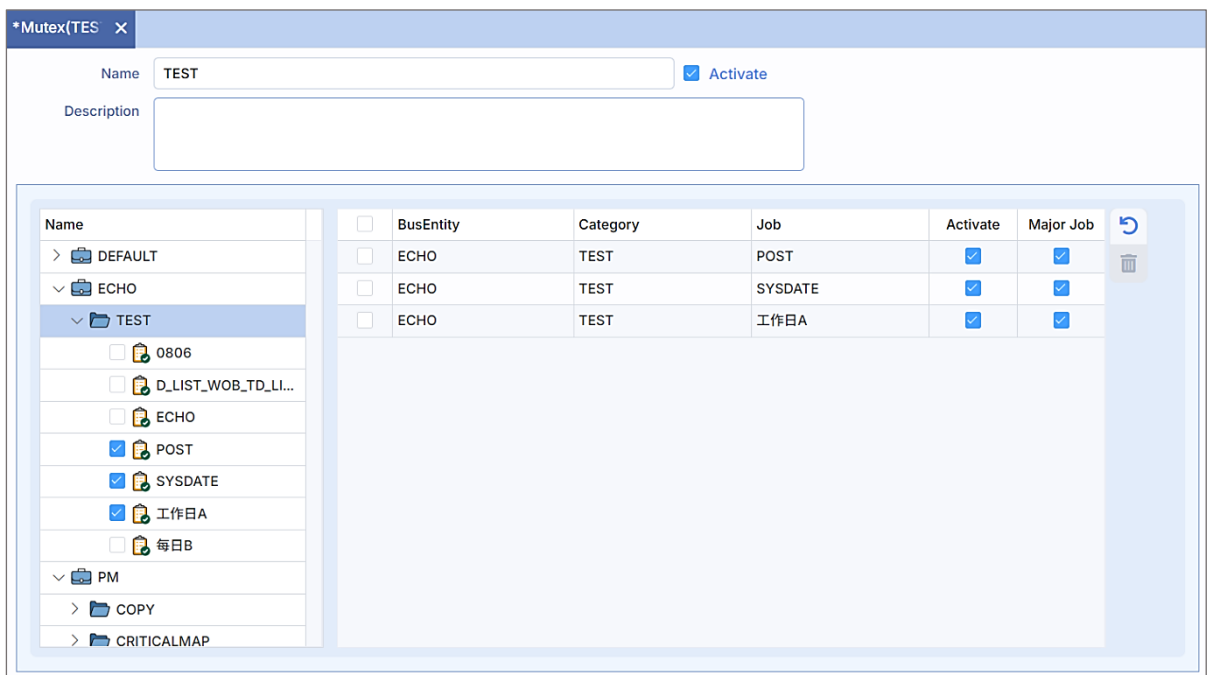
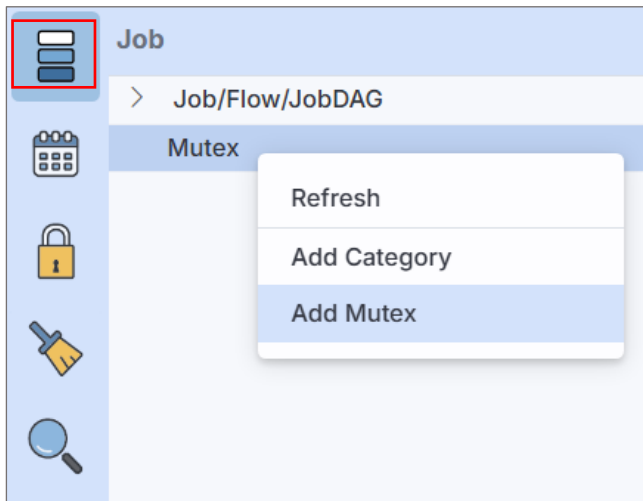



在 Task Console 中執行作業時會檢查同一 Mutex 中同一時間只有 Job R1 與其他 Job 互斥 (Job R1 有勾選 Major) · 而 Job R2 與 Job R3 則不會互斥 · 如下圖。



## 新增 Mutex

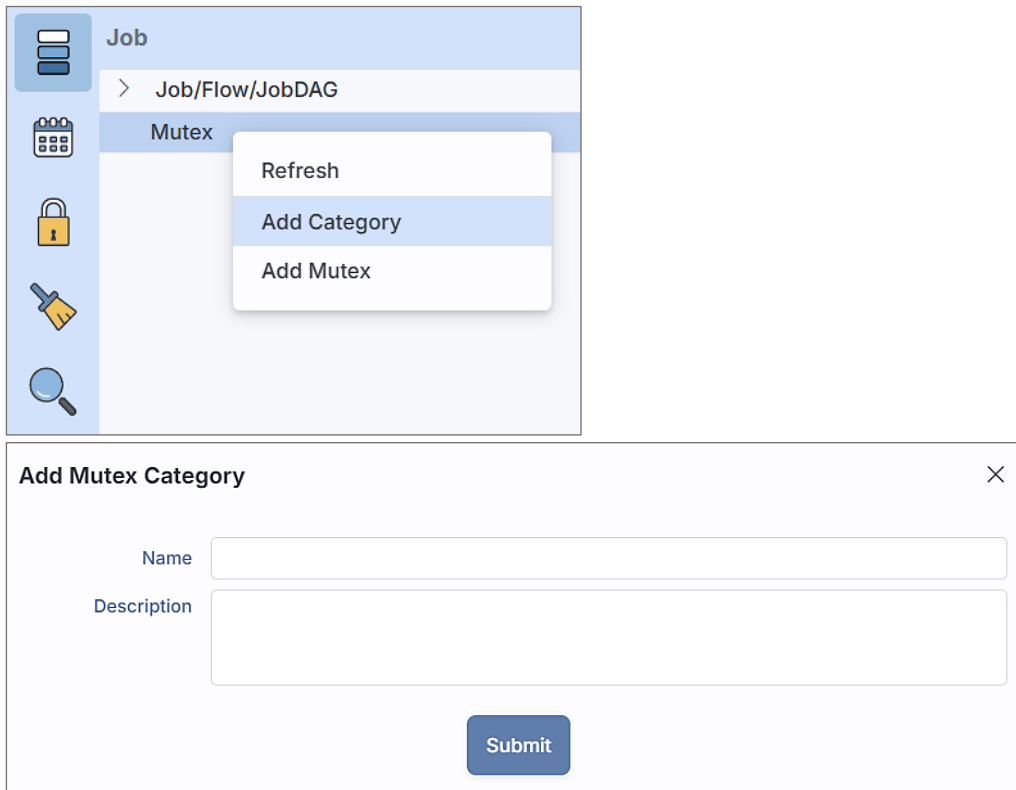
於功能視圖中點選 Job 項目，樹狀架構中選取 Mutex 點選滑鼠右鍵從 Popup Menu 中選取「Add Mutex」



- 新增 Job：於左方樹狀架構勾選 Job
- 刪除 Job：於右方欄位中，勾選欲刪除的 Job 後點選  即可刪除
- 新增完畢後在 Mutex 目錄下方會新增 Mutex 節點
- 若是選擇 Add Mutex Category，則會先在 Mutex 目錄下新增一個 Mutex Category，之後可在 Mutex Category 上按右鍵選擇 Add Mutex 新增 Mutex

## 新增 Mutex Category

於功能視圖中點選 Trinity 項目，樹狀架構中選取 Mutex 點選滑鼠右鍵從 Popup Menu 中選取「Add Category」，之後會出現一個對話視窗，要求輸入要新增的 Category 名稱，例如下圖中所呈現的範例：



## 修改 Mutex

於樹狀視圖中選擇要進行修改的物件子項目，在物件子項目上按下滑鼠右鍵從 Popup Menu 中選取「Property」。執行後會出現類似先前新增 Mutex 的對話視窗可供使用者輸入修改的資訊。修改完畢後按下「OK」按鈕會將所輸入的設定更新至 Trinity 數據庫中。

## 刪除 Mutex

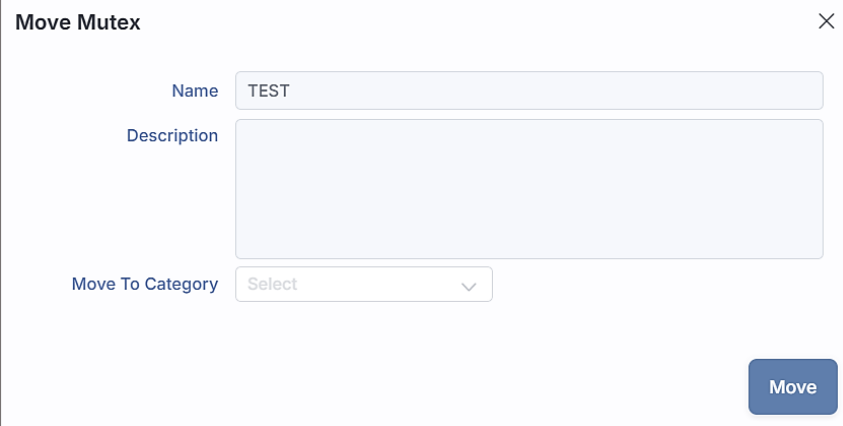
於樹狀視圖中選擇要進行刪除的物件子項目，在物件子項目上按下滑鼠右鍵後，從 Popup Menu 中選取「Delete」。

要注意的是當刪除 Mutex Category 時，如 Mutex Category 還包含有 Mutex 項目時，使用者是無法對它進行刪除的操作。

## 移動 Mutex 至其他 Mutex Category

有時候使用者可能會需要將一項 Mutex 從一個 Mutex Category 中搬移到另外一個 Mutex Category 中。於樹狀視圖中選擇要進行搬移的 Mutex 物件子項目，在物件子項目上按下滑鼠右鍵後，從 Popup Menu 中選取「MoveMutex」。

執行後會出現一個對話視窗詢問使用者要將此 Mutex 搬移到那一個 Mutex Category 中，例如下圖中所顯示的範例：



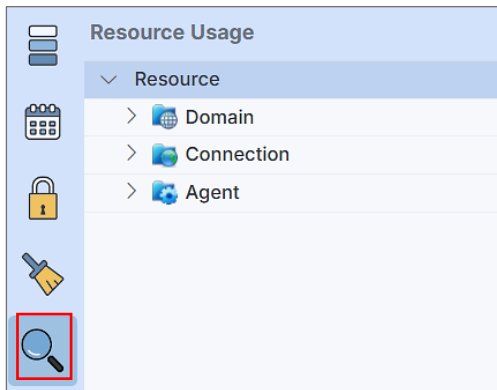
The screenshot shows a dialog box titled "Move Mutex" with a close button (X) in the top right corner. The dialog contains the following fields:

- Name:** A text input field containing the value "TEST".
- Description:** A large, empty text area.
- Move To Category:** A dropdown menu with the text "Select" and a downward arrow.

A blue button labeled "Move" is positioned at the bottom right of the dialog.

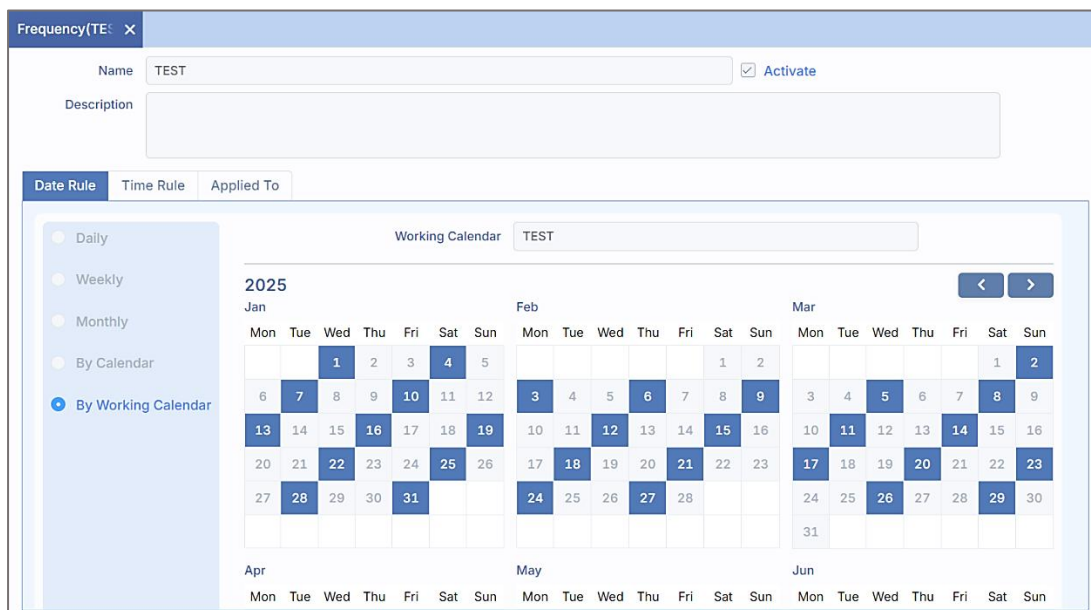
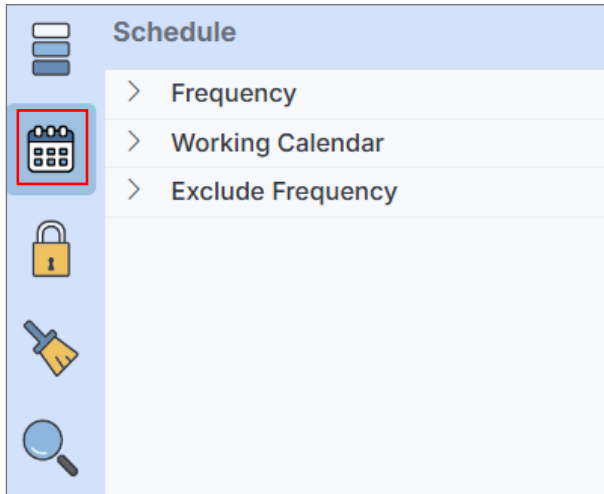
# Resource

Resource 的項目下包含了 Domain、Connection 及 Agent 項目。必需在 Admin UI 中才可進行設定。這些項目在 Web JFDesigner 中僅提供查詢及執行 Resource Export/Import 的功能。



# Schedule

Schedule 的項目下包含了 Frequency、Working Calendar 及 Exclude Frequency 項目。必需在 Admin UI 中才可進行設定，這些項目在 Web JFDesigner 中僅提供查詢功能。



# Import/Export

使用者可直接在此進行 Job、Flow 或 Resource 項目的匯出與匯入。

## Import/Export Job

預設只有 root 才可以使用，可授權其他使用者或角色使用。

匯入 Job/Flow 時，目的端的 Trinity 中，需存在與來源端相同的 Agent 及 Domain 名稱。故建議先進行 Resource 的匯入，再進行 Job/Flow 的匯入。

**操作 Import/Export 選項位置如下：**

- 於系統選單點選 File -> Import 或 Export
  - Export：需先於樹狀視圖區以滑鼠左鍵點選目標作業階層，此按鈕才會顯示為可選狀態。
  - Import：為可選狀態，但若未於樹狀視圖區以滑鼠左鍵選取目標作業階層再按此紐，則代表依照匯出檔的作業階層匯入；而若有先在樹狀視圖區以滑鼠左鍵點選目標作業階層再按此紐，則代表要匯入的目的地是所指定的作業階層下。
- 於樹狀視圖區對 Business Entity、Category、Job 或 Flow 以滑鼠右鍵開啟選單並點選 Import 或 Export。

**Export Job 的包含內容：**

- Job 基本資料(Execution Time, Note, Owner, Job Category, Business Entity, Stream Job(只包含設定，不包含目標 Job), Dependency Job(只包含設定，不包含目標 Job), Notification)
- Step 中所用到的 Connection
- Job 用到的 File Source,
- Job 用到的 Frequency
- 沒有包含 Domain, Agent

**Import Job 的成功條件(全部都滿足才可以算成功)：**

- 此 Job 所用到的 Domain 已經存在。
- 此 Job 所用到的 Agent 已經存在。
- 此 Job 所用到的全部 Alias 有存在 (Domain, File Source, Frequency, Agent, Connection)。
- 如果此 Job 存在，此 Job 不能被 Check out。
- 如果此 Job 存在，必須要有此 Job 的 Edit 權限。
- 如果此 Job 不存在，必須要有此 Category 的 Add 權限。
- 此 Job 如果有使用 Alias 的 Connection，來源端(Import File)的 Connection 類型跟目的端(資料庫)的類型一致。
- 此 Job 使用的 Connection，來源端(Import File)的 Connection 類型跟目的端(資料庫)的類型一致。(如果使用 Overwrite Connection 並不會檢查)

**Import Job 的失敗原因(如果有一項不滿足就算失敗)：**

- 此 Job 所用到的 Domain 不存在。
- 此 Job 所用到的 Agent 不存在。
- 此 Job 所用到的 Alias 不存在 (Domain, File Source, Frequency, Agent, Connection)。
- 如果此 Job 存在，此 Job 被某人 Check out。
- 如果此 Job 存在，沒有此 Job 的 Edit 權限。
- 如果此 Job 不存在，沒有此 Category 的 Add 權限。
- 此 Job 如果有使用 Alias 的 Connection，來源端(Export File)的 Connection 類型跟目的端(資料庫)的類型不一致。
- 此 Job 使用的 Connection，來源端(Import File)的 Connection 類型跟目的端(資料庫)的類型不一致。(如果使用 Overwrite Connection 並不會檢查)

**Import Job 結果出現 Warning 的情況：**

- 此 Job 的 Down Stream Job 不存在 (Job 還是會 Import 進去)。
- 此 Job 的 Dependency Job 不存在 (Job 還是會 Import 進去)。

**Export Flow 的包含內容：**

- Flow 基本資料(Execution Time, Job Category, Business Entity, Stream Job(只包含 Flow 的設計框架，不包含目標 Job), Notification)
- FlowMap

**Import Flow 成功條件(全部都滿足才可以算成功)：**

- 此 Flow 所用到的 Domain 已經存在。
- 此 Flow 所用到的全部 Alias 有存在 (Domain, Frequency)。
- 此 Flow 內的所有 Job 已經存在。
- 如果此 Flow 存在，此 Flow 不能被 Check out。
- 如果此 Flow 存在，必須要有此 Flow 的 Edit 權限。
- 如果此 Flow 不存在，必須要有此 Category 的 Add 權限。

**Import Flow 失敗原因(如果有一項不滿足就算失敗)：**

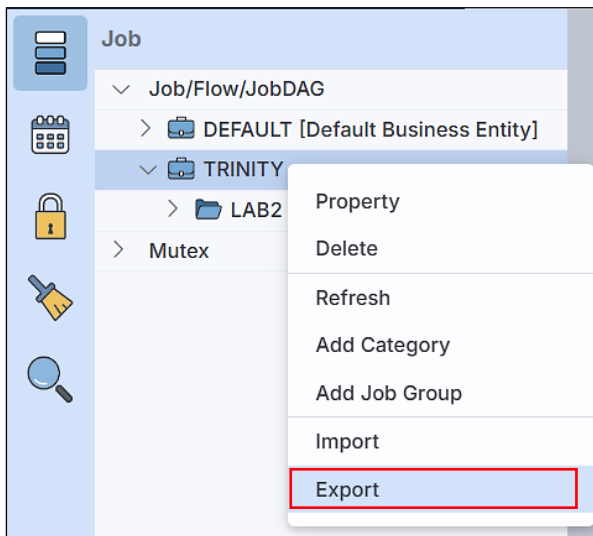
- 此 Flow 所用到的 Domain 不存在。
- 此 Flow 所用到的 Alias 不存在 (Domain, Frequency)。
- 此 Flow 內的任何一個 Job 不存在。
- 如果此 Flow 存在，此 Flow 被某人 Check out。
- 如果此 Flow 存在，沒有此 Flow 的 Edit 權限。
- 如果此 Flow 不存在，沒有此 Category 的 Add 權限。

**Import Flow 結果出現 Warning 的情況：**

- 此 Flow 的 Down Stream Job 不存在 (Flow 還是會 Import 進去)。

## Export Job

開啟 Web JFDesigner，對樹狀視圖區中目標項目以滑鼠右鍵開啟選單並點選「Export」，或於 Tool Bar 點選 File -> Export。



若目標項目為 Business Entity 作業階層，則開啟視窗畫面如下：

The 'Export BusEntity' dialog box contains the following elements:

- Checkboxes:  Dependency Job,  Down Stream Job,  Don't Export Related Job
- Radio buttons:  Current,  Last Check in Version,  Export To All,  Export To Individual
- Table:

Item	Action
<input type="checkbox"/> TEST	
<input type="checkbox"/> 0806	D/S
<input type="checkbox"/> D_LIST_WOB_TD_LISTS_	D/S
<input type="checkbox"/> ECHO	D/S
<input type="checkbox"/> POST	D/S
<input type="checkbox"/> SYSDATE	D/S
<input type="checkbox"/> 工作日A	D/S
<input type="checkbox"/> 每日B	D/S
<input type="checkbox"/> A	D/S
<input type="checkbox"/> B	D/S
<input type="checkbox"/> TEST	D/S

File Name:

若目標項目為 Category 作業階層，則開啟視窗畫面如下：

**Export Category** ✕

Dependency Job   
  Down Stream Job   
  Don't Export Related Job

Current   
  Last Check in Version  
 Export To All   
  Export To Individual

	Name	Type	Description
<input type="checkbox"/>	0806	Job	
<input type="checkbox"/>	D_LIST_WOB_TD_LISTS_	Job	
<input type="checkbox"/>	ECHO	Job	
<input type="checkbox"/>	POST	Job	
<input type="checkbox"/>	SYSDATE	Job	
<input type="checkbox"/>	工作日A	Job	
<input type="checkbox"/>	每日B	Job	
<input type="checkbox"/>	A	Flow	
<input type="checkbox"/>	B	Flow	
<input type="checkbox"/>	TEST	Flow	

File Name:

**Export**

若目標項目為 Job 作業階層，則開啟視窗畫面如下：

**Export Job** ✕

Dependency Job   
  Down Stream Job **D/S**   
  Don't Export Related Job

Current   
  Last Check in Version

File Name:

**Export**

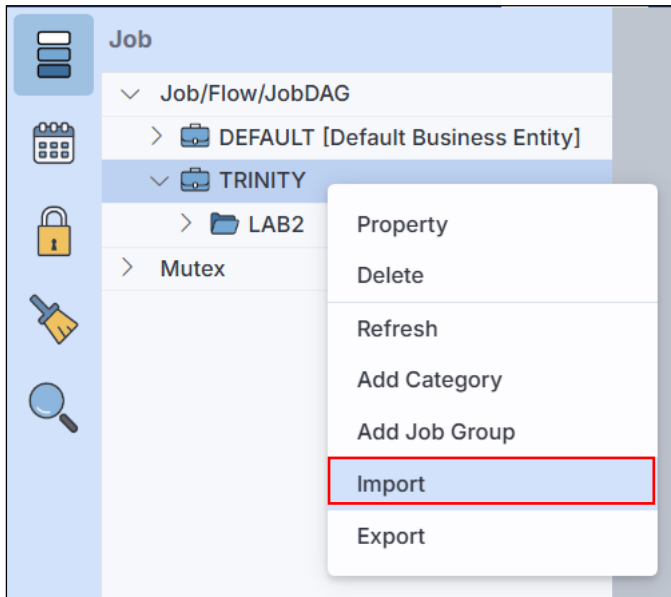
於開啟視窗，設定匯出目標：

- Dependency Job：勾選則會匯出 Job Dependency 設定及其相關的 Job。
- Down Stream Job：勾選則會匯出 Job Stream 設定及其相關的 Job。
- Don't Export Related Job：勾選則不會匯出與自身 Dependency/Job Stream 相關的 Job。
- Current：依照當時設計的 Job 作匯出。
- Last check in version：依照 Job 最新一版的 check in 內容匯出。
- Export to all：會將所有勾選的 Job 壓縮成一個 gz 格式檔案。

- **Export to individual** : 會將每個勾選的 Job 分別壓縮成獨立的 gz 格式檔案。
- 「D/S」按鈕：可檢視該筆 Job 的 Dependency/Job Stream 關係。
- 如欲將 Category 所開啟的頁面中所有 Job 一次全選，可點擊標題欄位當中第一格 (空白格)，則有全選效果。
- **File Name** : 可自行輸入檔名或使用預設檔名(副檔名需為 .gz)

## Import Job

開啟 Web JFDesigner，對樹狀視圖區中目標項目以滑鼠右鍵開啟選單並點選「Import」，或於 Tool Bar 點選 File -> Import。



開啟視窗畫面如下：

**Import** ✕

Import Target:

File Path:

Online Date Time:

Options:  Resource exist before they can import Job  Overwrite Dependency  
 Overwrite Stream  Only Depend/Stream  Use Original UUID

<input checked="" type="checkbox"/>	Mode	BusEntity	Category	Name	Type	Resource Option	File Name
<input checked="" type="checkbox"/>	Add	ECHO	COPY	COPY	Job		09-PM_CO
<input checked="" type="checkbox"/>		ECHO	TEST	D_LIST_WOB_TD_LI...	Job		09-ECHO_
<input checked="" type="checkbox"/>		ECHO	TEST	ECHO	Job		09-ECHO_

- Import Target :
  - 若未於樹狀視圖區以滑鼠左鍵點選目標作業階層就按下 Import，則此欄位顯示「Not Selected.」（最終會以匯出檔內容的作業階層匯入）
  - 若於樹狀視圖區以滑鼠左鍵點選目標作業階層為 Business Entity 再用任何方式點選 Import，則顯示 Business Entity 名稱；若點選目標作業階層為 Category 則顯示「Business Entity 名稱:: Category 名稱」。(最終會匯入至此欄位所顯示

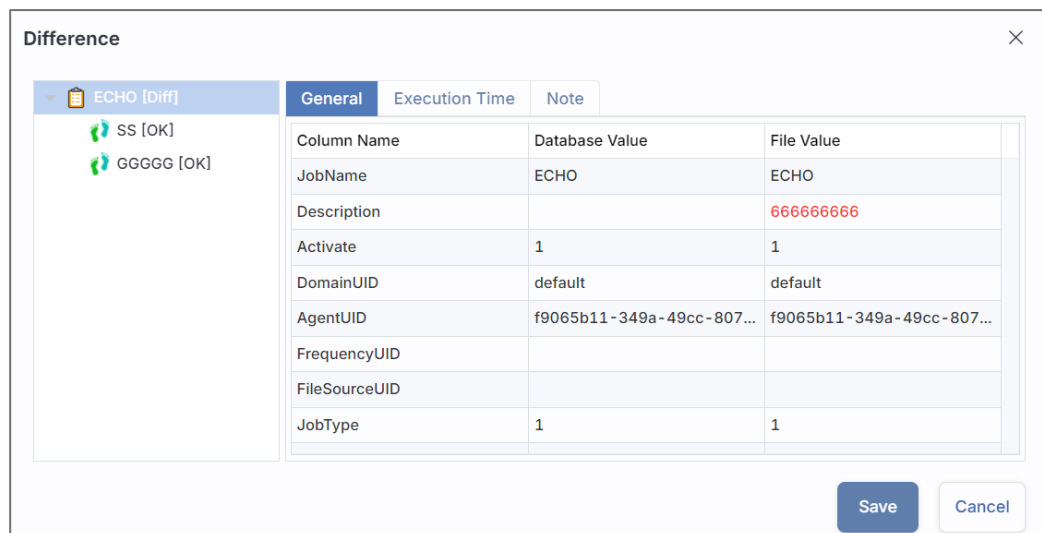
的作業階層下)

- File path :
  - 按下「Click To Upload」, 可自本地端選擇先前匯出的匯出檔。選擇之後於畫面下方會立即呈現匯出檔當中的項目。
  - 呈現後, 預設為全選, 也可僅選擇部分項目; 標題欄位當中第一格(空白格)則有全選效果。最後匯入時僅針對有勾選的目標進行匯入。
  - 也可指定到目錄, 代表要對目錄下所有匯出檔執行。執行時有固定的順序, 如下:

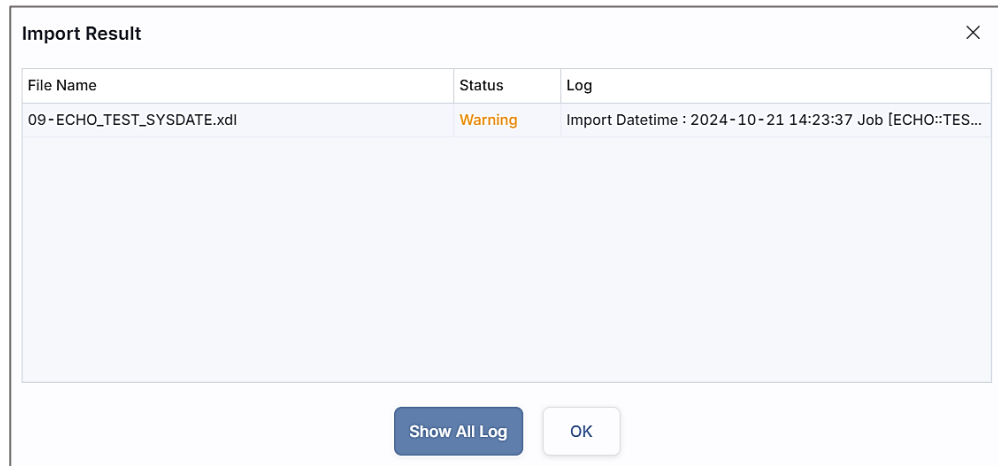
順序	Resource
01	Domain
02	Connection
03	File Source
04	Frequency
05	JCSAgent
06	Alias
07	User
08	User Group
09	Job
11	Flow
12	Business Entity Variable
13	Category Variable
14	Exclude Frequency
15	Mutex
16	Role
17	Permission

- Online Date Time : 若有勾選並設定 Online Date Time , 則此批匯入的作業其「Online Date Time」設定值將會一併調整, 可限制 Job 的啟用日期時間。
- Resource exist before they can import job : 預設為勾選, 可於匯入前檢查目的端是否具備來源端所需使用之相同名稱的 Resource , 若具備才可匯入, 否則顯示警示訊息且不匯入。若未勾選, 則不檢查且皆可匯入。
- Overwrite Dependency : 未勾選代表若目的端已存在該 Job , 則不會異動目的端 Job 的 Dependency 設定; 若勾選, 則會以來源端 Job 的 Dependency 設定進行異動。
- Overwrite Stream : 未勾選代表若目的端已存在該 Job , 則不會異動目的端 Job 的 Job Stream 設定; 若勾選, 則會以來源端 Job 的 Job Stream 設定進行異動。

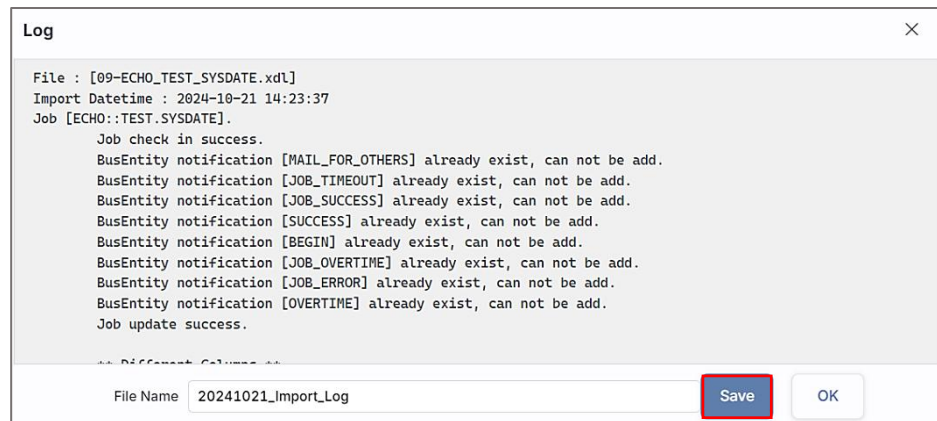
- Only Depend/Stream：勾選表示僅匯入 Dependency 及 Job Stream 相關項目。
- Use Original UUID：勾選表示匯入的作業會以來源端所匯出之 UUID 建立在目的端當中。若來源端所匯出之 UUID 已存在於目的端，則即使來源端與目的端的 Job Name 並不同，結果仍會將目的端該 UUID 的 Job 進行 Job 異動。
- 表格中的細項：
  - Mode：「Add」代表目的端不存在此項；「Update」代表目的端存在此項但與來源端內容不同；若未顯示則代表目的端存在此項且與來源端內容一致。
  - Resource Option：此欄供 Import Resource 時使用。Import Job 時不須理會此欄。
  - File Name：顯示來源端的檔案項目。
  - Type：Import 完畢後才會呈現，若顯示「Diff」代表 Import 結果有異動目的地端內容；若顯示「OK」代表目的端內容無異動。
  - Detail 按鈕：配合「Type」欄位結果為「Diff」時顯示，點即可查看異動細節資訊，舉例如下圖：



點擊 Import 後會跳出 Import Result 視窗



- Status 欄位：顯示 Import 結果狀態。
- Show Log：可針對單一項目點擊兩下或全部項目點選 Show All Log 並提供 Save 功能，舉例如下：



完成 Import 後回到 Import 畫面會自動重新 Check Import 項目的「Mode」及「Difference」狀態。

## Import/Export Resource

Resource 依使用特性可分為：

- Job 相關：Alias、Busentity Variable、Category Variable、Connection、Domain、Exclude Frequency、Frequency、File Source、Agent、Mutex
- 系統設定：Permission、Role、User、User Group

預設只有 root 及 Administrator 角色的使用者才可以使用此功能。

- 操作 Import/Export 選項位置如下：
- 於系統選單點選 File -> Import 或 Export
  - Export：需先於樹狀視圖區以滑鼠左鍵選取 Domain、Connection、Agent，此按鈕才會顯示為可選狀態。
  - Import：為可選狀態。
- 於樹狀視圖區對「Resource」項目以滑鼠右鍵開啟選單並點選 Import 或 Export。
- 說明：

Resource	Import/Export Resource 說明
Alias	<p><b>Export Resource 的包含內容</b></p> <p>1. Alias 基本資料</p> <p><b>Import Resource 成功條件：</b> (這邊的成功，是針對 Alias 的一筆資料)</p> <p>1. 此筆 Alias 資料，所對映的物件(Connection, Domain, Agent, Frequency, Filesource)必須存在</p> <p>2. 必須有此 Alias 所設定的 Business Entity 必須存在。</p> <p>3. 如果使用 Add Option，此 Alias Name 不存在。</p> <p><b>Import Resource 失敗原因：</b> (這邊的失敗，是針對 Alias 的一筆資料)</p> <p>1. 此筆 Alias 資料，所對映的物件(Connection, Domain, Agent, Frequency, Filesource)不存在。</p> <p>2. 此 Alias 所設定的 Business Entity 不存在</p> <p>3. 如果使用 Add Option，此 Alias Name 已經存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：只會新增名字不存在的 Alias。</p> <p>2. Overwrite：會把原本 Business Entity 上的 Alias 全部刪除，並匯入來源的 Alias。</p>

	<p>3. Update：會覆蓋跟來源上有相同名字的 Alias，不存在的會新增，原本已經存在的不會刪除。</p>
Business Entity Variable	<p><b>Export Resource 的包含內容：</b></p> <p>1. Business Entity Variable 基本資料</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 必須有此 Business Entity Variable 所設定的 Business Entity 必須存在。</p> <p>2. 如果使用 Add Option，此 Variable Name 不存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 此 Business Entity Variable 所設定的 Business Entity 不存在</p> <p>2. 如果使用 Add Option，此 Variable Name 已經存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：只會新增名字不存在的 Variable。</p> <p>2. Overwrite：會把原本 Business Entity 上的 Variable 全部刪除，並匯入來源的 Variable。</p> <p>3. Update：會覆蓋跟來源上有相同名字的 Variable，不存在的會新增，原本已經存在的不會刪除。</p>
Category Variable	<p><b>Export Resource 的包含內容：</b></p> <p>1. Category Variable 基本資料</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 必須有此 Category Variable 所設定的 Business Entity / Category 必須存在。</p> <p>2. 如果使用 Add Option，此 Variable Name 不存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 此 Category Variable 所設定的 Business Entity / Category 不存在</p> <p>2. 如果使用 Add Option，此 Variable Name 已經存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：只會新增名字不存在的 Variable。</p>

	<p>2. Overwrite：會把原本 Category 上的 Variable 全部刪除，並匯入來源的 Variable。</p> <p>3. Update：會覆蓋跟來源上有相同名字的 Variable，不存在的會新增，原本已經存在的不會刪除。</p>
Connection	<p><b>Export Resource 的包含內容：</b></p> <p>1. Connection 基本資料，如果此 Connection 在 Connection Category 底下，會一併匯出 Connection Category 資訊。</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 如果使用 Add Option，此 Connection 不存在。</p> <p>2. 如果使用 Overwrite Option，此 Connection 已經存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 如果使用 Add Option，此 Connection 已經存在。</p> <p>2. 如果使用 Overwrite Option，此 Connection 不存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：新增此筆 Connection 資訊</p> <p>2. Overwrite：會把原本 Connection 刪除，並匯入來源的 Connection。</p>
Domain	<p><b>Export Resource 的包含內容：</b></p> <p>1. Domain 基本資料。</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 如果使用 Add Option，此 Domain 不存在。</p> <p>2. 如果使用 Overwrite Option，此 Domain 已經存在。</p> <p>3. 如果使用 Update Option，此 Domain 已經存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 如果使用 Add Option，此 Domain 已經存在。</p> <p>2. 如果使用 Overwrite Option，此 Domain 不存在。</p> <p>3. 如果使用 Update Option，此 Domain 不存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：新增此筆 Domain 資訊</p>

	<p>2. Overwrite：會把原本 Domain 刪除(包含底下 Variable) · 並匯入來源的 Domain。</p> <p>3. Update：會覆蓋跟來源上有相同名字的 Variable · 不存在的會新增 · 原本已經存在的不會刪除。</p>
Exclude Frequency	<p><b>Export Resource 的包含內容：</b></p> <p>1. Exclude Frequency 基本資料 · 包含 Apply Job, Apply Flow, Apply Frequency 關係。</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 如果使用 Add Option · 此 Exclude Frequency 不存在。</p> <p>2. 如果使用 Overwrite Option · 此 Exclude Frequency 已經存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 如果使用 Add Option · 此 Exclude Frequency 已經存在。</p> <p>2. 如果使用 Overwrite Option · 此 Exclude Frequency 不存在。</p> <p><b>Import 結果出現 Warning 的情況：</b></p> <p>1. 此 Apply 的 Job 不存在。</p> <p>2. 此 Apply 的 Flow 不存在。</p> <p>3. 此 Apply 的 Frequency 不存在。</p> <p>4. 此 Apply 的 JobDAG 不存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：新增此筆 Exclude Frequency 資訊 · 包含 Apply Job, Apply Flow, Apply Frequency 關係。</p> <p>2. Overwrite：會把原本 Exclude Frequency 刪除 · 並匯入來源的 Exclude Frequency 包含 Apply Job, Apply Flow, Apply Frequency 關係。</p>
Filesource	<p><b>Export Resource 的包含內容：</b></p> <p>1. Filesource 基本資料 · 如果此 Filesource 在 Filesource Category 底下 · 會一併匯出 Filesource Category 資訊。</p> <p>2. 如果此 Filesource 有用到 FTP Connection 也會一併匯出</p> <p><b>Import Resource 成功條件：</b></p>

	<p>1. 如果使用 Add Option，此 Filesource 不存在。</p> <p>2. 如果使用 Overwrite Option，此 Filesource 已經存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 如果使用 Add Option，此 Filesource 已經存在。</p> <p>2. 如果使用 Overwrite Option，此 Filesource 不存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：新增此筆 Filesource 資訊</p> <p>2. Overwrite：會把原本 Filesource 刪除，並匯入來源的 Filesource。</p>
Frequency	<p><b>Export Resource 的包含內容：</b></p> <p>1. Frequency 基本資料，如果此 Frequency 在 Frequency Category 底下，會一併匯出 Frequency Category 資訊。</p> <p>2. 如果此 Frequency 有用到 Working Calendar 也會一併匯出</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 如果使用 Add Option，此 Frequency 不存在。</p> <p>2. 如果使用 Overwrite Option，此 Frequency 已經存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 如果使用 Add Option，此 Frequency 已經存在。</p> <p>2. 如果使用 Overwrite Option，此 Frequency 不存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：新增此筆 Frequency 資訊</p> <p>2. Overwrite：會把原本 Frequency 刪除，並匯入來源的 Frequency。</p>
Working Calendar (Export Frequency 選項內)	<p><b>Export Resource 的包含內容：</b></p> <p>1. Working Calendar 基本資料。</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 如果使用 Add Option，此 Working Calendar 不存在。</p> <p>2. 如果使用 Overwrite Option，此 Working Calendar 已經存在。</p>

	<p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Working Calendar 已經存在。</li> <li>2. 如果使用 Overwrite Option，此 Working Calendar 不存在。</li> </ol> <p><b>Import 的 Option 行為：</b></p> <ol style="list-style-type: none"> <li>1. Add：新增此筆 Working Calendar 資訊</li> <li>2. Overwrite：會把原本 Working Calendar 刪除，並匯入來源的 Working Calendar。</li> </ol>
JCS Agent	<p><b>Export Resource 的包含內容：</b></p> <ol style="list-style-type: none"> <li>1. Agent 基本資料。</li> </ol> <p><b>Import Resource 成功條件：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Agent 不存在。</li> <li>2. 如果使用 Overwrite Option，此 Agent 已經存在。</li> </ol> <p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Agent 已經存在。</li> <li>2. 如果使用 Overwrite Option，此 Agent 不存在。</li> </ol> <p><b>Import 的 Option 行為：</b></p> <ol style="list-style-type: none"> <li>1. Add：新增此筆 Agent 資訊</li> <li>2. Overwrite：會把原本 Agent 刪除，並匯入來源的 Agent。</li> </ol>
JCS Agent (Virtual)	<p><b>Export Resource 的包含內容：</b></p> <ol style="list-style-type: none"> <li>1. Virtual Agent 基本資料，包含與 Agent 的關係。</li> </ol> <p><b>Import Resource 成功條件：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Virtual Agent 不存在。</li> <li>2. 如果使用 Overwrite Option，此 Virtual Agent 已經存在。</li> <li>3. 此 Virtual Agent 的實際 Agent 必須存在。</li> </ol> <p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Virtual Agent 已經存在。</li> <li>2. 如果使用 Overwrite Option，此 Virtual Agent 不存在。</li> </ol>

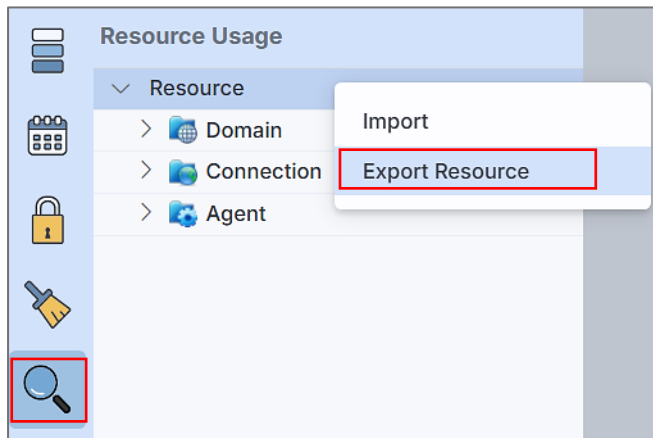
	<p>3. 此 Virtual Agent 的實際 Agent 任何一個不存在。</p> <p><b>Import 的 Option 行為：</b></p> <ol style="list-style-type: none"> <li>1. Add：新增此筆 Virtual Agent 資訊</li> <li>2. Overwrite：會把原本 Virtual Agent 刪除，並匯入來源的 Virtual Agent。</li> </ol>
Mutex	<p><b>Export Resource 的包含內容：</b></p> <ol style="list-style-type: none"> <li>1. Mutex 基本資料，包含與 Job 的關係，如果此 Mutex 在 Mutex Category 底下，會一併匯出 Mutex Category 資訊。</li> </ol> <p><b>Import Resource 成功條件：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Mutex 不存在。</li> <li>2. 如果使用 Overwrite Option，此 Mutex 已經存在。</li> <li>3. 此 Mutex 底下的 Job 必須存在。</li> </ol> <p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 Mutex 已經存在。</li> <li>2. 如果使用 Overwrite Option，此 Mutex 不存在。</li> <li>3. 此 Mutex 底下的 Job 任何一個不存在。</li> </ol> <p><b>Import 的 Option 行為：</b></p> <ol style="list-style-type: none"> <li>1. Add：新增此筆 Mutex 資訊</li> <li>2. Overwrite：會把原本 Mutex 刪除，並匯入來源的 Mutex。</li> </ol>
Permission	<p><b>Export Resource 的包含內容：</b></p> <ol style="list-style-type: none"> <li>1. Permission 基本資料</li> </ol> <p><b>Import Resource 成功條件：</b></p> <ol style="list-style-type: none"> <li>1. 此筆 Permission 的 Role or User 必須存在。</li> <li>2. 此筆 Permission 目標的物件必須存在(Business Entity, Category, Job, Flow...等)。</li> <li>3. 如果使用 Add Option，此筆 Permission 的資料不存在。</li> <li>4. 如果使用 Overwrite Option，此筆 Permission 的資料必須存在。</li> </ol> <p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 此筆 Permission 的 Role or User 不存在。</li> </ol>

	<p>2. 此筆 Permission 目標的物件不存在(Business Entity, Category, Job, Flow...等)。</p> <p>3. 如果使用 Add Option，此筆 Permission 的資料已經存在。</p> <p>4. 如果使用 Overwrite Option，此筆 Permission 的資料不存在。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：只會新增不存在的 Permission。</p> <p>2. Overwrite：會把原本此 Role or User 上的 Permission 全部刪除，並匯入來源的 Permission。</p> <p>3. Update：會覆蓋跟來源上有相物件的 Permission，不存在的會新增，原本已經存在的不會刪除。</p> <p><b>判斷是否為相同 Permission 的依據為，同個 Role or User 對同一個目標物件(Business Entity, Category, Job, Flow...等)。</b></p>
Role	<p><b>Export Resource 的包含內容：</b></p> <p>1. Role 基本資料，包含與 User 的關係。</p> <p><b>Import Resource 成功條件：</b></p> <p>1. 如果使用 Add Option，此 Role 不存在。</p> <p>2. 如果使用 Overwrite Option，此 Role 已經存在。</p> <p><b>Import Resource 失敗原因：</b></p> <p>1. 如果使用 Add Option，此 Role 已經存在。</p> <p>2. 如果使用 Overwrite Option，此 Role 不存在。</p> <p><b>Import 結果出現 Warning 的情況：</b></p> <p>1. 此 Role 的 Member user 不存在 (Role 還是會 Import 進去)。</p> <p><b>Import 的 Option 行為：</b></p> <p>1. Add：新增此筆 Role 資訊，包含與 User 的關係</p> <p>2. Overwrite：會把原本 Role 刪除，並匯入來源的 Role，包含與 User 的關係。</p> <p>3. Update：會把不存在的 User 關係新增，原本已經存在的不會刪除。</p>

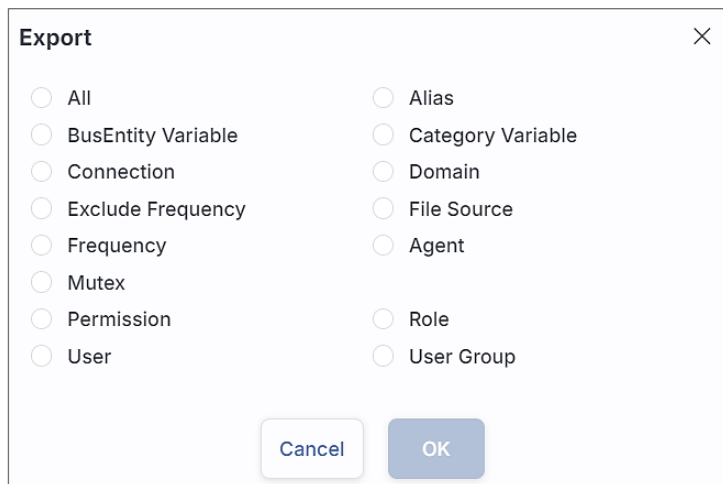
User	<p><b>Export Resource 的包含內容：</b></p> <ol style="list-style-type: none"> <li>1. User 基本資料。</li> </ol> <p><b>Import Resource 成功條件：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 User 不存在。</li> <li>2. 如果使用 Overwrite Option，此 User 已經存在。</li> </ol> <p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 User 已經存在。</li> <li>2. 如果使用 Overwrite Option，此 User 不存在。</li> </ol> <p><b>Import 的 Option 行為：</b></p> <ol style="list-style-type: none"> <li>1. Add：新增此筆 User 資訊</li> <li>2. Overwrite：會把原本 User 刪除，並匯入來源的 User。</li> </ol>
User Group	<p><b>Export Resource 的包含內容：</b></p> <ol style="list-style-type: none"> <li>1. User Group 基本資料，包含與 User 的關係。</li> </ol> <p><b>Import Resource 成功條件：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 User Group 不存在。</li> <li>2. 如果使用 Overwrite Option，此 User Group 已經存在。</li> </ol> <p><b>Import Resource 失敗原因：</b></p> <ol style="list-style-type: none"> <li>1. 如果使用 Add Option，此 User Group 已經存在。</li> <li>2. 如果使用 Overwrite Option，此 User Group 不存在。</li> </ol> <p><b>Import 結果出現 Warning 的情況：</b></p> <ol style="list-style-type: none"> <li>1. 此 User Group 的 Member user 不存在 (User Group 還是會 Import 進去)。</li> </ol> <p><b>Import 的 Option 行為：</b></p> <ol style="list-style-type: none"> <li>1. Add：新增此筆 User Group 資訊，包含與 User 的關係</li> <li>2. Overwrite：會把原本 User Group 刪除，並匯入來源的 User Group，包含與 User 的關係。</li> <li>3. Update：會把不存在的 User 關係新增，原本已經存在的不會刪除。</li> </ol>

## Export Resource

開啟 Web JFDesigner，點選功能視圖 Resource 圖示，在樹狀視圖區中對 Resource 項目點選滑鼠右鍵開啟選單並點選「Export」，或於 Tool Bar 點選 File -> Export。

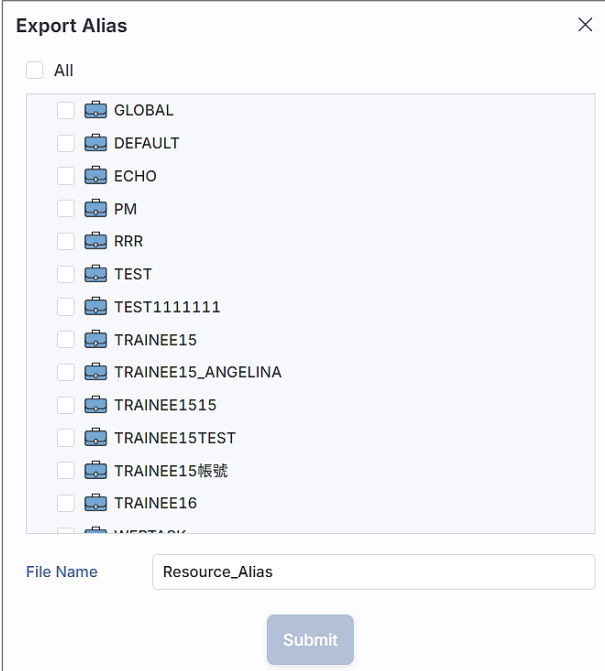


開啟視窗畫面如下：



- 「All」選項：僅供 Root 使用者登入使用，可將畫面上提及的所有 Resource 一併匯出。

- 「Alias」選項：選取後顯示下一個操作畫面，如下：



Export Alias

All

- GLOBAL
- DEFAULT
- ECHO
- PM
- RRR
- TEST
- TEST1111111
- TRAINEE15
- TRAINEE15\_ANGELINA
- TRAINEE1515
- TRAINEE15TEST
- TRAINEE15帳號
- TRAINEE16

File Name: Resource\_Alias

Submit

- 右上方「All」選項：可將所有項目一次選取。
- Alias 項目：列出所有 Alias Reference 項目(依據 Business Entity 區分)
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Busentity Variable」選項：選取後顯示下一個操作畫面，如下：

**Export BusEntity Variable** ×

All

- DEFAULT
- ECHO
- PM
- RRR
- TEST
- TEST1111111
- TRAINEE15
- TRAINEE15\_ANGELINA
- TRAINEE1515
- TRAINEE15TEST
- TRAINEE15帳號
- TRAINEE16
- WEBTASK

File Name:

Submit

- 右上方「All」選項：可將所有項目一次選取。
- Business Entity Variable 項目：列出所有 Business Entity Variable(依據 Business Entity 區分)。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Category Variable」選項：選取後顯示下一個操作畫面，如下：

Export Category Variable

All

- DEFAULT
- ECHO
- PM
- RRR
- TEST
- TEST1111111
- TRAINEE15
- TRAINEE15\_ANGELINA
- TRAINEE1515
- TRAINEE15TEST
- TRAINEE15帳號
- TRAINEE16
- WEBTASK

File Name: Resource\_CategoryVar

Submit

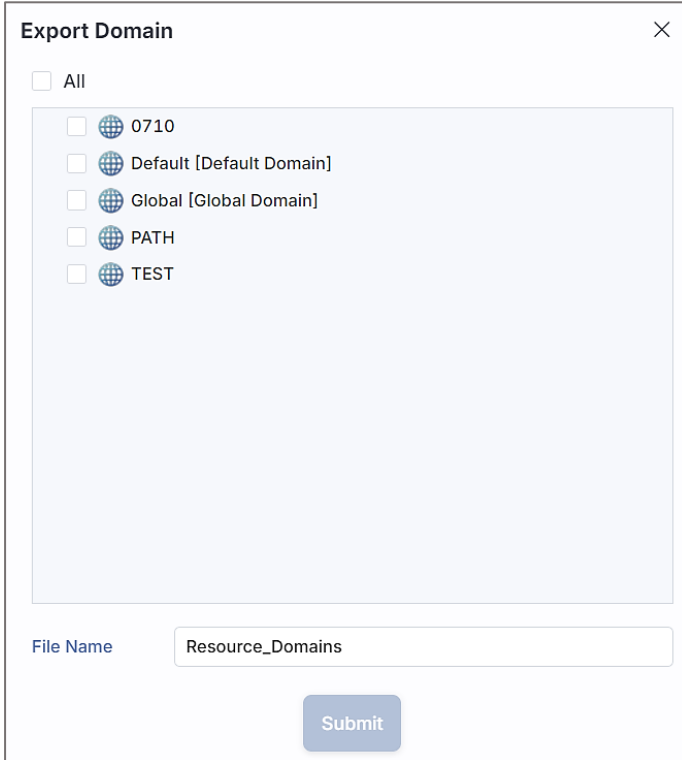
- 右上方「All」選項：可將所有項目一次選取。
- Category Variable 項目：列出所有 Category Variable 項目(依據 Category 區分)
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Connection」選項：選取後顯示下一個操作畫面，如下：



- 右上方「All」選項：可將所有項目一次選取。
- Connection 項目：列出所有 Connection 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Domain」選項：選取後顯示下一個操作畫面，如下：



The screenshot shows a dialog box titled "Export Domain" with a close button (X) in the top right corner. Inside the dialog, there is a list of domain options, each with a checkbox and a globe icon:

- All
- 0710
- Default [Default Domain]
- Global [Global Domain]
- PATH
- TEST

Below the list, there is a "File Name" label and an input field containing the text "Resource\_Domains". At the bottom center of the dialog is a "Submit" button.

- **注意：如果 Export 的 Resource 為 Domain – Default，Import 時限制一定要輸入 Resource Option 參數 overwrite(o) 或 update(u)。**  
**Ex: domain:o 或 domain:u**
- 右上方「All」選項：可將所有項目一次選取。
- Domain 項目：列出所有 Domain 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Exclude Frequency」選項：選取後顯示下一個操作畫面，如下：

Export Exclude Frequency

All

EX

- Apply Frequency
- Apply Job
- Apply Flow
- Apply JobDAG

File Name: Resource\_ExcludeFrequencies

Submit

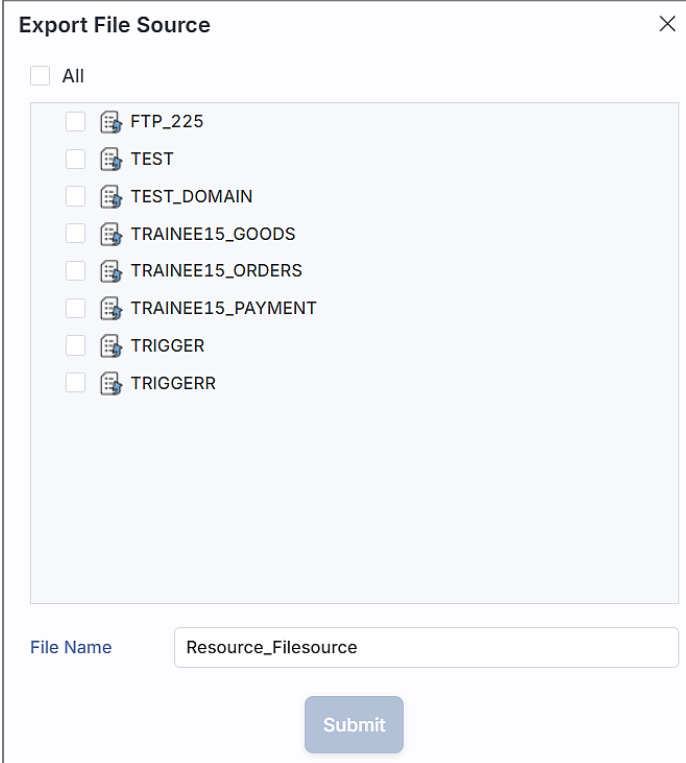
- **注意：如果 Export 的 Resource 為 Exclude Frequency – Global，Import 時限制一定要輸入 Resource Option 參數 overwrite(o)。**  
**Ex: exfq:o**
- 右上方「All」選項：可將所有項目一次選取。
- Exclude Frequency 項目：列出所有 Exclude Frequency 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Frequency」選項：選取後顯示下一個操作畫面，如下：

The screenshot shows a dialog box titled "Export Frequency" with a close button (X) in the top right corner. At the top left, there is an "All" checkbox. Below it is a list of items, each with a checkbox and a folder or file icon. The items are: TRAINEE15 (folder icon), DAILY 09:30 (calendar icon), FRE\_TEST0421 (calendar icon), ODS\_RES [04:00] (calendar icon), TEST (calendar icon), TEST\_0421\_2 (calendar icon), TRAINING\_15 (calendar icon), 工作日 (calendar icon), 每日 (calendar icon), RRRRRRR (calendar icon), SINOPAC\_ST\_WORKDAY [testtesttest] (calendar icon), and TEST (calendar icon). Below the list is a "File Name" label and an input field containing the text "Resource\_Frequencies". At the bottom center is a "Submit" button.

- 右上方「All」選項：可將所有項目一次選取。
- Frequency 項目：列出所有 Frequency 及 Working Calendar 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「File Source」選項：選取後顯示下一個操作畫面，如下：



**Export File Source** ×

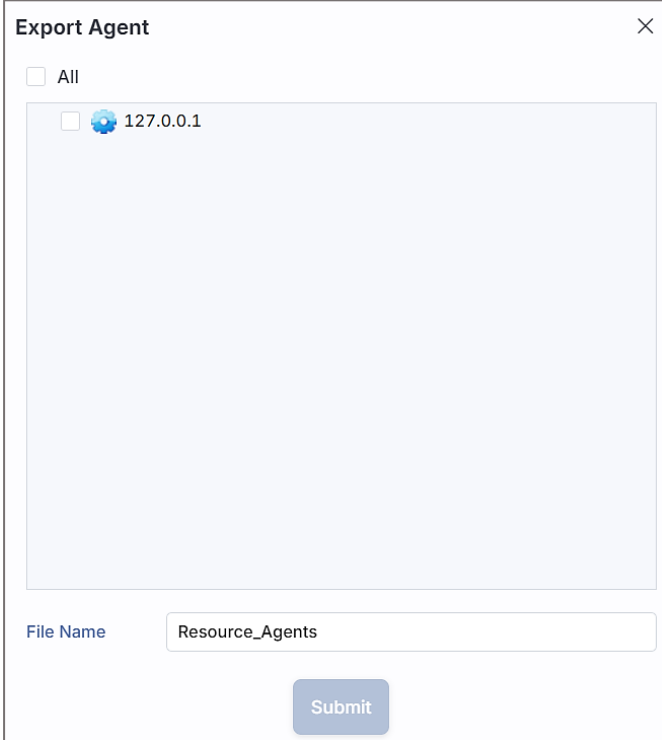
All

- FTP\_225
- TEST
- TEST\_DOMAIN
- TRAINEE15\_GOODS
- TRAINEE15\_ORDERS
- TRAINEE15\_PAYMENT
- TRIGGER
- TRIGGERR

File Name:

- 右上方「All」選項：可將所有項目一次選取。
- File Source 項目：列出所有 File Source 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Agent」選項：選取後顯示下一個操作畫面，如下：



Export Agent

All

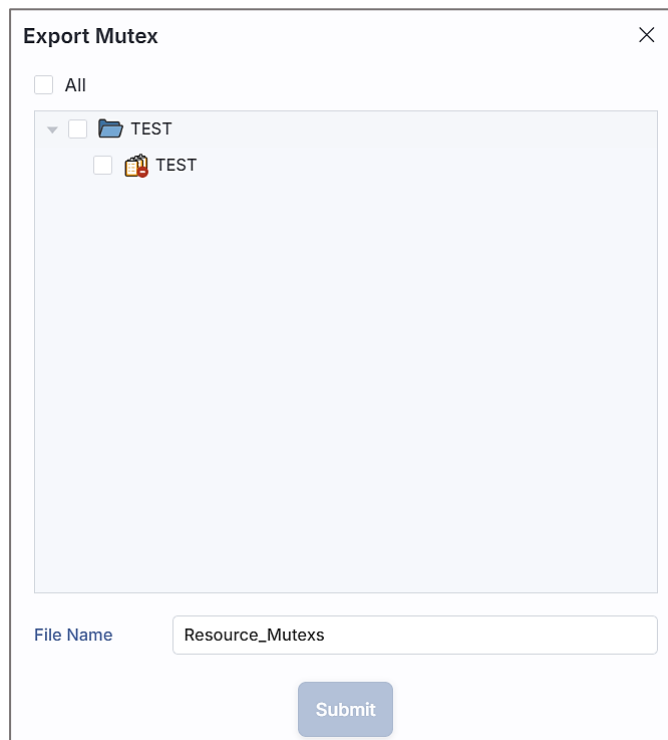
127.0.0.1

File Name: Resource\_Agents

Submit

- **注意：Virtual Agent 只會匯出定義，並不會把實體 Agent 一起匯出。**
- 右上方「All」選項：可將所有項目一次選取。
- Agent 項目：列出所有 Agent 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Mutex」選項：選取後顯示下一個操作畫面，如下：



- 右上方「All」選項：可將所有項目一次選取。
- Mutex 項目：列出所有 Mutex 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Permission」選項：選取後顯示下一個操作畫面，如下：

**Export Permission** ×

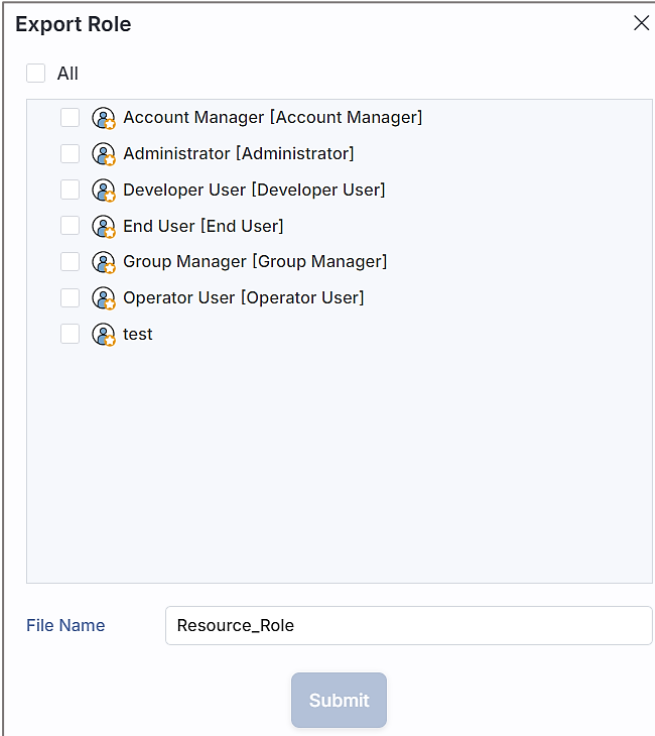
All

- Account Manager [Account Manager]
- Administrator [Administrator]
- Developer User [Developer User]
- End User [End User]
- Group Manager [Group Manager]
- Operator User [Operator User]
- test
- aaaa
- ang
- ray
- trainee10
- trainee11
- trainee12

File Name:

- 右上方「All」選項：可將所有項目一次選取。
- Permission 項目：列出所有 Permission 項目(依據 Role 或 User 區分)。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「Role」選項：選取後顯示下一個操作畫面，如下：



The screenshot shows a dialog box titled "Export Role" with a close button (X) in the top right corner. At the top left, there is a checkbox labeled "All". Below it is a list of roles, each with a checkbox and a user icon: "Account Manager [Account Manager]", "Administrator [Administrator]", "Developer User [Developer User]", "End User [End User]", "Group Manager [Group Manager]", "Operator User [Operator User]", and "test". At the bottom left, there is a "File Name" label and an input field containing the text "Resource\_Role". At the bottom center, there is a blue "Submit" button.

- **注意：如果 Export 的 Resource 為預設 Role – Administrator/Developer User/End User/Operator User, Import 時限制一定要輸入 Resource Option 參數 overwrite(o) 或 update(u)。**  
**Ex: role:o 或 role:u**
- 右上方「All」選項：可將所有項目一次選取。
- Role 項目：列出所有 Role 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「User」選項：選取後顯示下一個操作畫面，如下：

Export User

All

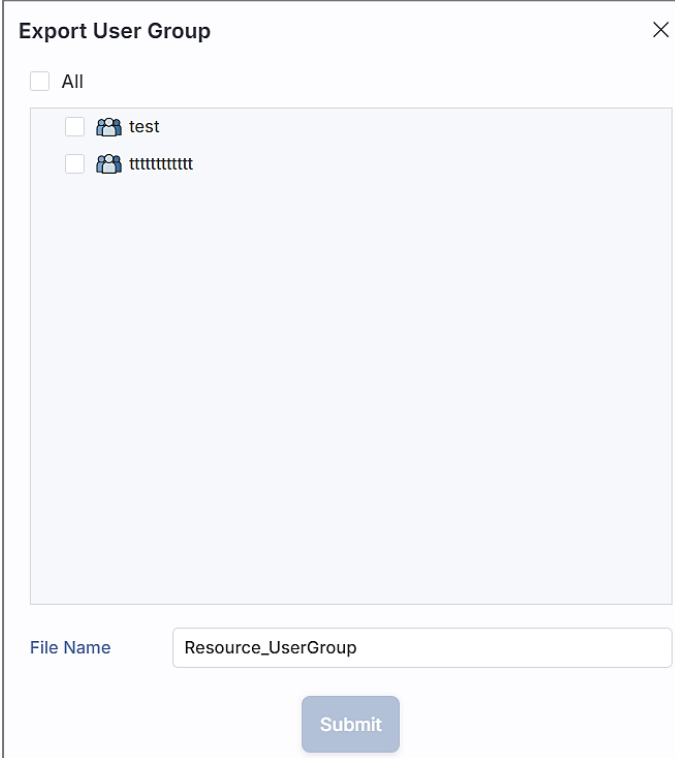
- aaaa
- ang
- ray
- trainee10
- trainee11
- trainee12
- trainee13
- trainee14
- trainee15
- trainee16
- trainee17
- trainee19
- trainee2

File Name: Resource\_User

Submit

- 右上方「All」選項：可將所有項目一次選取。
- User 項目：列出所有 User 項目。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

- 「User Group」選項：選取後顯示下一個操作畫面，如下：

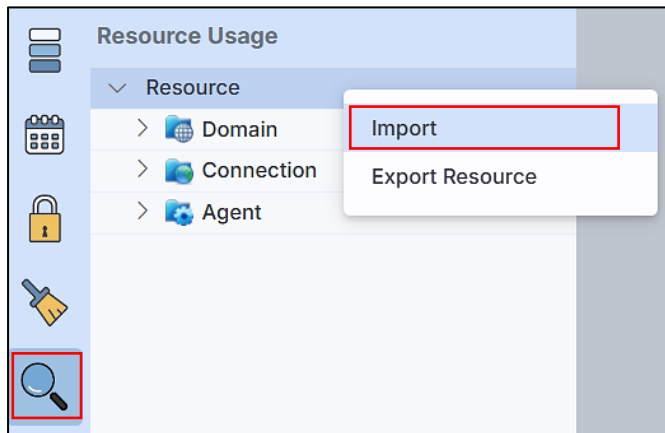


The screenshot shows a dialog box titled "Export User Group" with a close button in the top right corner. Below the title bar, there is a checkbox labeled "All". A list box contains two items: "test" and "tttttttttt", each with a user icon and a checkbox. Below the list box, there is a "File Name" label and a text input field containing "Resource\_UserGroup". At the bottom center, there is a "Submit" button.

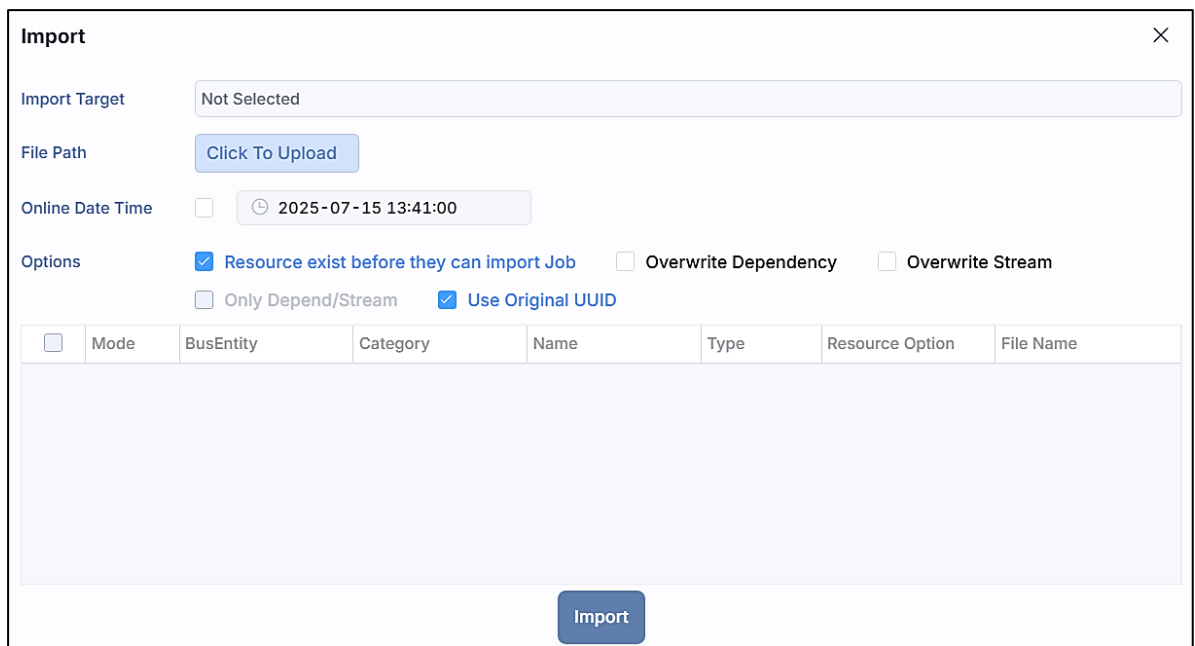
- **注意：User Group 只會匯出定義，並不會把 User 一起匯出。**
- 右上方「All」選項：可將所有項目一次選取。
- User Group 項目：列出所有 User Group。
- File name：可自行輸入檔名或使用預設檔名(副檔名須為 .gz)
- 「Submit」按鈕：按下開始進行匯出。

## Import Resource

開啟 Web JFDesigner，對樹狀視圖區中 Resource 項目以滑鼠右鍵開啟選單並點選「Import」，或於功能選單點選 File -> Import。



開啟視窗畫面如下：



- Import Target：固定顯示「Not Selected.」
- File or dir path：
- 按下「Click To Upload」，可自本地端選擇先前匯出的匯出檔。選擇之後於畫面下方會立即呈現匯出檔當中的項目。
- 呈現後，預設為全選，也可僅選擇部分項目；標題欄位當中第一格(空白格)則有全選效果。最後匯入時僅針對有勾選的目標進行匯入。
- 也可指定到目錄，代表要對目錄下所有匯出檔執行。執行時有固定的順序，如下：

順序	Resource
01	Domain

02	Connection
03	File Source
04	Frequency
05	JCSAgent
06	Alias
07	User
08	User Group
09	Job
11	Flow
12	Business Entity Variable
13	Category Variable
14	Exclude Frequency
15	Mutex
16	Role
17	Permission

- Online Date Time : 不須設定。
- Resource exist before they can import job : 不須設定。
- Overwrite Dependency : 不須設定。
- Overwrite Stream : 不須設定。
- Only Depend/Stream : 不須設定
- Use Original UUID : 不須設定。
- 表格中的細項：同本文件「Import/Export」章節 -> 「Import/Export Job」章節 -> 「Import Job」章節中同名欄位說明。

若有需要，可在 Resource Option 欄位裡面輸入參數設定，控制該筆項目的匯入行為（詳見本文件「Import/Export」章節之「Import 參數設定」章節說明），例如：conn:o

### Import

Import Target: Not Selected

File Path: Resource\_BusEntityVar.gz

Online Date Time:  2024-10-21 14:58:00

Options:  Resource exist before they can import Job  Overwrite Dependency  
 Overwrite Stream  Only Depend/Stream  Use Original UUID

<input checked="" type="checkbox"/>	Mode	BusEntity	Category	Name	Type	Resource Option	File Name
<input checked="" type="checkbox"/>				Resource	Res.		14-Resour

Import

點擊 Import 後會跳出 Import Result 視窗

- Status 欄位：顯示 Import 結果狀態。
- Show Log：可針對單一項目點擊兩下或全部項目點選 Show All Log 並提供 Save Log 功能。

## Import 參數設定

在進行 Import 時，可搭配下列參數使用於「Resource Option」欄位：

- o (overwrite)：會把原本存在的一律覆蓋。
- a (add)：只會新增，如果已存在則不會新增。
- n (no change)：不做任何變更。
- u (update and insert)：已存在的會更新，不存在的會新增，不會刪除原本既有的。

使用範例：

格式「Import 項目:參數」

- conn:o
- fq:a
- fs:a
- all:o
- conn:o,fq:q,fs:a

備註：

- u 只支援 Alias, Permission, Business Entity Variable, Domain, Role, Group。
- 如果未指定，預設都為 Add。
- u--Update 跟 o--overwrite 的差別在於 overwrite 會刪除原本既有的設定並全部使用檔案內的資料為主，而 Update 只會更新已存在、新增不存在的資料，而其他原本已經存在的資料並不會刪除。例如 Import Busentity Variable：

■ 目的端：

Variable name	value
V1	Value1
V2	Value2
V3	Value3

■ 來源端：

Variable name	value
V1	New_value1
V2	New_value2
V4	New_value4

- 使用 Overwrite 的結果，使目的端結果與來源端一致，變成只剩 V1、V2、V4：

Variable name	value
V1	New_value1
V2	New_value2
V4	New_value4

- 使用 Update 的結果，已存在的變數(V1、V2)進行異動，不存在的變數(V4)進行新增，來源端未包含的變數(V3)不受影響：

Variable name	value
V1	New_value1
V2	New_value2
V3	Value3
V4	New_value4

#### 使用情境範例：

- 在匯入 Connection Resource 時，只想要新增不存在的，可以使用 (conn:a) 或不輸入(預設為 Add)。
- 在匯入 Connection Resource 時，如果是此檔案的 Resource 都是必須更新或者新增，則可以輸入 (conn:o)。
- 在匯入 Group 時，如果要依照此檔案為主，則使用 (gp:o)
- 在匯入 Group 時，只想要新增檔案內的 Group User 原本已存在的保留，則使用(gp:u)

## Resource 項目參數對應表

Import 項目	設定使用名稱
alias	alias / al
business entity variable	busentityvar / bev
category variable	categoryvar / cv
connection	connection / conn
domain	domain / dm
exclude frequency	exfrequency / exfq
frequency	frequency / fq
filesource	filesource / fs
agent	agent / ag
mutex	mutex / mu
pluginlet	pluginlet / pl
permission	permission / pm
role	role / ro
user	user / us
user group	group / gp
virtual agent	vagent / vag
working calendar	workingcalendar / wc
notification	notification / nt

# Job/Job Step/JobDAG 設定

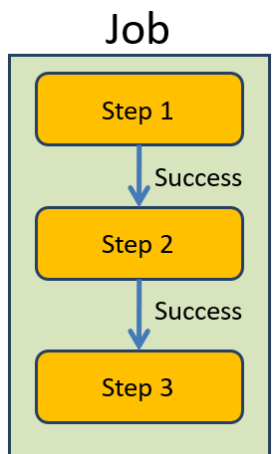
在這一章中，我們會詳述在 Web JFDesigner 中設定 Job、Job Step 與 JobDAG 的工作方式。

## 作業(Job)設定

在 Trinity 中，作業(Job)的定義代表了在某一台 JCS Agent 上所執行的工作。但是 Job 本身並不包含真正執行的指令，真正定義執行指令的是在作業步驟(Job Step)當中。

一個 Job 可以包含多個 Job Step，當一項 Job 開始執行時會從其中的第一個 Job Step 開始執行，必須在執行成功後才會往下執行下一個 Job Step，一直到最後一個 Job Step 執行完畢為止。

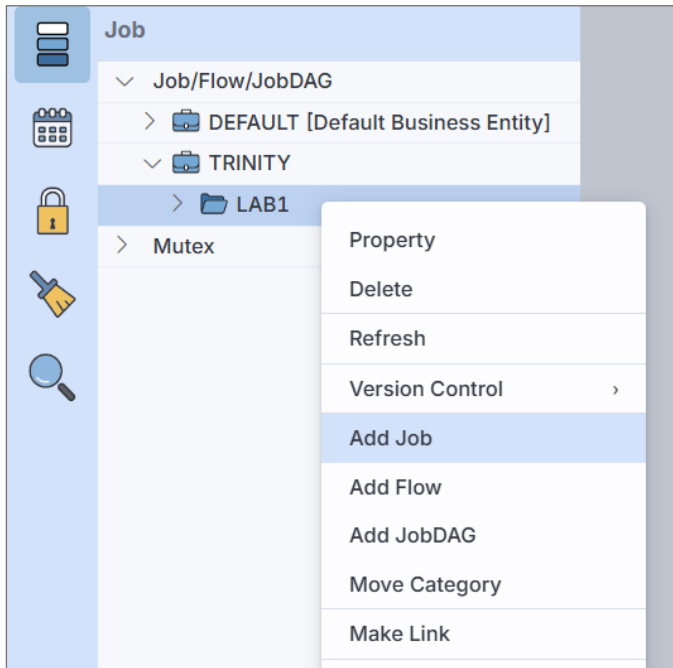
如果其中有任何一個 Job Step 執行失敗的話，此 Job 會被視為錯誤的狀態而停止下來。



## 新增 Job

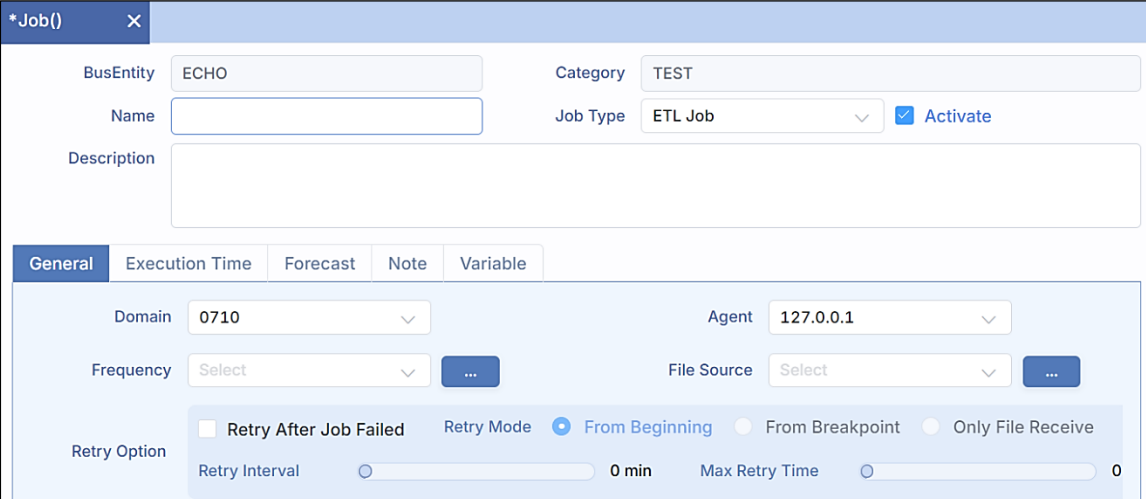
新增 Job 的動作必須要指定一個 Category，關於 Category 的建立工作的詳細操作請參考 Business Entity 與 Category 的操作說明。

於樹狀視圖中選擇 Category 物件項目後，在物件項目上按下滑鼠右鍵，從 Popup Menu 中選取「Add Job」。要注意的是使用者必須對要進行新增 Job 的 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此新增的動作。



之後會在右方的工作區中顯示一個新增 Job 的頁面。

此畫面中區分為 General、Execution Time、Forecast、Note、Variable 五個頁面：



\*Job() ×

BusEntity ECHO Category TEST

Name  Job Type ETL Job  Activate

Description

General Execution Time Forecast Note Variable

Domain 0710 Agent 127.0.0.1

Frequency Select File Source Select

Retry Option

Retry After Job Failed Retry Mode  From Beginning  From Breakpoint  Only File Receive

Retry Interval 0 min Max Retry Time 0

以下介紹 General 頁面內容，其餘頁面請見本文件「Job/Job Step 設定」章節中「Job 強化功能設定說明」章節。

## General 頁面各欄位的說明 -- 第一部分

BusEntity	ECHO	Category	TEST
Name		Job Type	ETL Job <input type="checkbox"/> Activate
Description			

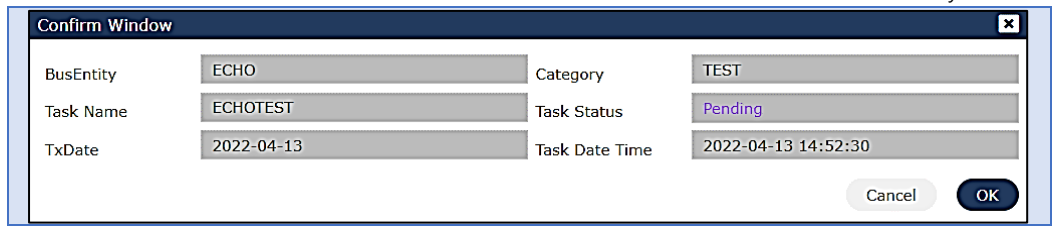
- Name：設定此作業的名稱
- Job Type：指定此作業的型別，選項如下：
  - ETL Job
  - OLAP Job
  - Online Batch Job：當選擇了此項目，表示當下不管有多少個 task 都會同時執行，而且即使前一個 Task 沒執行完，排程時間到了，下一個 Task 也會照常產生並執行。因此，如果有設定 Online Batch Job 且排程是設定每小時執行一次，但同時這個 Job 又設定了 time window，只有晚上 8 點到隔天早上 8 點能跑，這樣每個小時都會產生 Task，但只會在 time window 指定的時間才執行。
  - Others
  - Report Job
  - X1-Comfirm Job：在作業執行完成後，需要於 Task Console 將該作業進行人工確認，作業執行狀態才會變更成 Success，並觸發後續相關作業。

## Task Console 畫面示意圖：

Job 於 All View 頁面中呈現紫色，於 Current Task View (New) 頁面中的 Confirm Task 頁面呈現狀態為 Pending，需對其按滑鼠右鍵執行。按下後會顯示跳出視窗，確認 Job 資訊正確後按下 OK 即可。

Domain	%	BusEntity	ECHO
Job	%	Flow	%
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <input type="checkbox"/> + 100 <input type="checkbox"/> -         </div> <div style="display: flex; gap: 10px;"> <div style="background-color: green; color: white; padding: 5px 10px; border: 1px solid black;">ECHO</div> <div style="background-color: purple; color: white; padding: 5px 10px; border: 2px solid red;">ECHOTEST</div> </div> </div>			

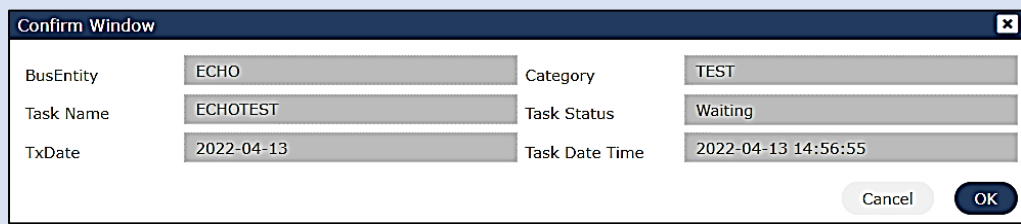
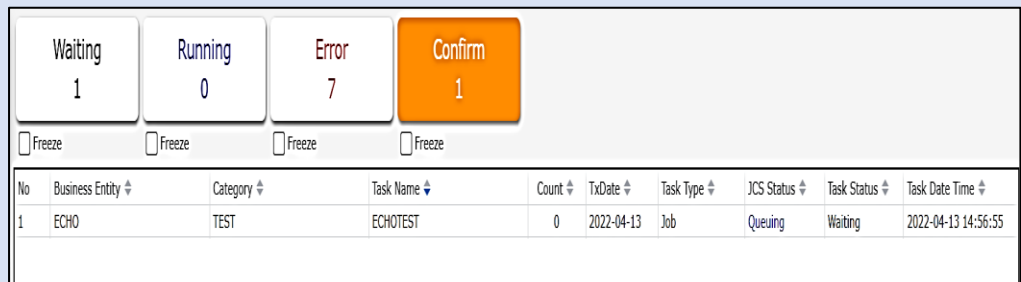
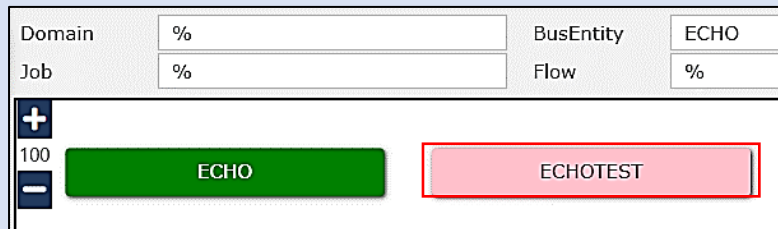
Waiting	Running	Error	Confirm						
0	1	7	1						
<input type="checkbox"/> Freeze	<input type="checkbox"/> Freeze	<input type="checkbox"/> Freeze	<input type="checkbox"/> Freeze						
No	Business Entity	Category	Task Name	Count	TxDate	Task Type	JCS Status	Task Status	Task Date Time
1	ECHO	TEST	ECHOTEST	1	2022-04-13	Job	Normal	Pending	2022-04-13 14:52:30



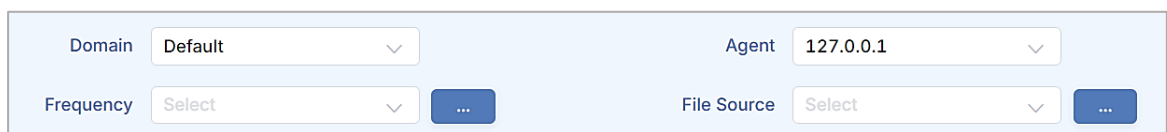
- X2-Confirm Job：在執行之前需要人工確認後，才會開始真正執行

**Task Console 畫面示意圖：**

Job 於 All View 頁面中呈現粉紅色，於 Current Task View (New) 頁面中的 Confirm Task 頁面呈現狀態為 Waiting，需對其按滑鼠右鍵執行。按下後會顯示跳出視窗，確認 Job 資訊正確後按下 OK 即可。



- Activate：勾選則代表此作業為有效狀態。作業必須是在有效狀態下才會被自動排程模組依照預先設定的時間來啟動執行。
- Description：對此作業的描述說明。



- Domain：選擇此作業所使用的 Domain。Domain 可以用來指定作業中會用的變數，Domain 的下拉選項需至 Admin UI 中進行設定。
- Agent：選擇此作業所使用的 Agent。Agent 的下拉選項需至 Admin UI 中進行設定。
- Frequency：選擇此作業所使用的 Frequency。Frequency 可用來指定該作業的排程執行時間，指定後即可在預定的時間點產生 Task 來執行。Frequency 的下拉選項需至 Admin UI 中進行設定。
- File Source：選擇此作業所使用的檔案來源。File Source 的下拉選項需至 Admin UI 中進行設定。

註：General 頁面中的 Domain、Agent、Frequency、File Source 下拉選項顯示內容包含顯示 Alias 項目（\$ 字號在前端的項目）（Alias 說明請見「Trinity V5.0 Admin UI User Guide」手冊）。

### General 頁面各欄位的說明 -- 第二部分

- Retry After Job Failed：作業執行失敗後是否要自動重新執行。
- Retry Mode：失敗作業重新執行的方式。
  - From Beginning：從作業中的第一個作業步驟開始重新執行。
  - From Breakpoint：從作業中失敗的作業步驟開始重新執行，此種方式會忽略前面已經執行成功的作業步驟。
  - Only File Receive：只在 File Receive 執行狀態為 fail 或 timeout 時才進行重新執行（若該 job 在 File Receive 執行狀態為 Success，後續 Step Error，則不滿足 retry 條件）
- Retry Interval：失敗作業重新執行的間隔時間，單位是分鐘，例如：設定值為 10，則會在作業執行失敗的 10 分鐘後，再重新執行失敗作業。
- Max Retry Time：重新執行失敗作業的最多次數，例如：設定值為 3，會嘗試重新執行失敗作業 3 次，如果已經失敗了 3 次，那麼此作業就會被放棄執行而標記成失敗的作業。

### General 頁面各欄位的說明 -- 第三部分

- Time Window Begin：允許作業在一天之中可執行的時間區間的起始時間。
- Time Window End：允許作業在一天之中可執行的時間區間的結束時間。

## General 頁面各欄位的說明 -- 第四部分

TxDate Setting	TxDate Rule	System Date	TxDate Offset	-	0	+	Select	View TxDate
----------------	-------------	-------------	---------------	---	---	---	--------	-------------

- TxDate Rule：設定此作業在排程產生工作時指定的 Transaction Date 的規則，可供選擇的選項有：
  - **Begining of Month (月初日期)：**  
TxDate Offset 設定範圍可為-99~999 的整數，-1 代表 Tx Date 為執行日所在月份的前一個月的第一天，0 代表 Tx Date 為與執行日同月的第一天，1 代表 Tx Date 為執行日所在月份的下一個月的第一天，2 代表 Tx Date 為執行日所在月份的下兩個月的第一天... 依次類推。所得到的日期格式為 YY-MM-DD。
  - **By Calendar-BOM (工作日曆月初日期)：**  
TxDate Offset 設定範圍可為-99~999 的整數，須配合右方下拉選單來選擇欲參照的工作日曆，當 TxDate Offset 設定-1 代表 Tx Date 為執行日所在月份的下一個月的第一天工作日，0 代表 Tx Date 為與執行日同月的第一天工作日，1 代表 Tx Date 為執行日所在月份的前一個月的第一天工作日，2 代表 Tx Date 為執行日所在月份的前兩個月的第一天工作日.. 依次類推 (若工作日曆當中該月無工作日，則預設為月份第一天)。所得到的日期格式為 YY-MM-DD。
  - **By Calendar – EOM (工作日曆月底日期)：**  
TxDate Offset 設定範圍可為-99~999 的整數，須配合右方下拉選單來選擇欲參照的工作日曆，當 TxDate Offset 設定-1 代表 Tx Date 為執行日所在月份的下一個月的最後一天工作日，0 代表 Tx Date 為與執行日同月的最後一天工作日，1 代表 Tx Date 為執行日所在月份的前一個月的最後一天工作日，2 代表 Tx Date 為執行日所在月份的前兩個月的最後一天工作日.. 依次類推 (若工作日曆當中該月無工作日，則預設為月份最後一天)。所得到的日期格式為 YY-MM-DD。
  - **By Calendar Offset (工作天)：**  
其中的 TxDate Offset 設定範圍可為-99~999 的整數，須配合右方下拉選單來選擇欲參照的工作日曆，-1 代表 Tx Date 為執行日的後一天工作天，0 代表 Tx Date 為執行日當天，1 代表 Tx Date 為執行日前一天工作天，2 代表 Tx Date 為執行日前二天工作天... 依次類推 (若推移後找不到工作日，則預設 Tx Date 為執行日當天)。所得到的日期格式為 YY-MM-DD。
  - **End of Month (月底日期)：**  
TxDate Offset 設定範圍可為-99~999 的整數，-1 代表 Tx Date 為執行日所在月份的前一個月的最後一天，0 代表 TxDate 為執行日當月的最後一天，1 代表 TxDate 為執行日所在月份的後一個月的最後一天，2 代表 TxDate 為執行日所在月份的後二個月的最後一天... 依次類推。所得到的日期格式為 YY-MM-DD。
  - **Last Job TxDate (前次執行 TxDate)：**

TxDate Offset 設定範圍可為-99~999 的整數，-1 代表 TxDate 為前次執行 TxDate 的前一天，0 代表 TxDate 為前次執行 TxDate 日期，1 代表 TxDate 為前次執行 TxDate 的後一天，2 代表 TxDate 為前次執行 TxDate 的後二天... 依次類推 (若無前次執行 TxDate，則以本次執行 TxDate 進行推移)。所得到的日期格式為 YY-MM-DD。

■ **Offset Day ( TxDate 推移之後的日期 ):**

其中的 TxDate Offset 設定範圍可為-99~999 的整數，-1 代表 TxDate 為執行日後一天，0 代表 TxDate 為執行日當天，1 代表 TxDate 為執行日前一天，2 代表 TxDate 為執行日前二天... 依次類推。所得到的日期格式為 YY-MM-DD。

■ **System Date ( 系統日期 ):**

以系統日期作為 TxDate，日期格式為 YY-MM-DD。為此欄位之預設選項。

■ **Week Day ( 一週起始日 ):**

一週為一個範圍，以星期日為起始日，TxDate Offset 設定範圍可為-99~999 的整數，-1 代表 TxDate 為執行日當週星期日的前一天，0 代表 TxDate 為執行日當週星期日的日期，1 代表 TxDate 為執行日當週星期日往後推算一天，2 代表 TxDate 為執行日當週星期日往後推算二天... 依次類推。所得到的日期格式為 YY-MM-DD。

範例：今天日期：18/04/20(五)

TxDate Offset	TxDate	Week
-2	18/04/13	星期五
-1	18/04/14	星期六
0	18/04/15	星期日
1	18/04/16	星期一
2	18/04/17	星期二
3	18/04/18	星期三

- TxDate Offset 輸入格：輸入 Transaction Date 的運算值。
- TxDate Offset 右方下拉式選單：此項依附 TxDate Rule，僅有 TxDate Rule 選擇 By Calendar-BOM、By Calendar-EOM 與 By Calendar Offset 時，此選單才可供選擇，選項則有預設之 SYSTEMDAY 與可自行於 Frequency 設定之選項可做選擇。
- View TxDate：推算 TxDate Rule 套用結果。

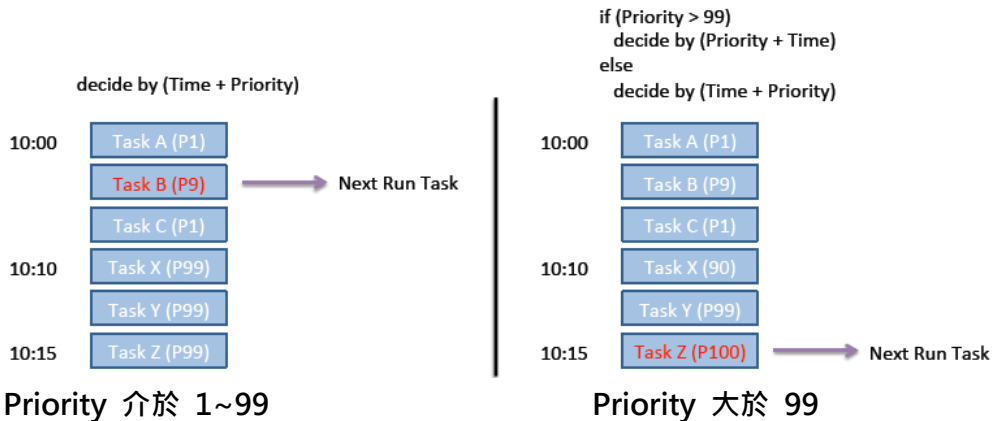
**View TxDate** ×

System Date

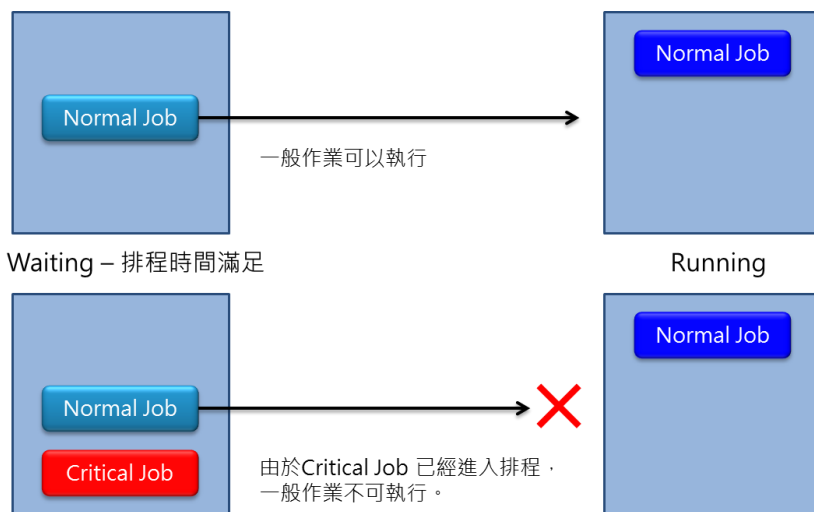
TxDate

General 頁面各欄位的說明 -- 第五部分

- **Priority**：執行作業的優先順序。若多個作業皆設定在同一個時間點執行，則此設定可排序作業的執行順序（若作業的執行時間皆不同，則此設定不影響執行）。執行機制會分成兩部分，Priority 大於 99（上限為 199）的 Task 會直接以優先序高低來決定下一個可執行的 Task，而 Priority 介於 1 ~ 99 的值則沿用目前的處理機制。



- **Bypass Error**：檢查若該作業前一次執行的執行結果為 Error，則本次是否可執行，勾選代表本次仍可執行，未勾選代表本次不可執行，須由維運人員排除前次執行問題，使前次執行成功後，才可執行本次。
- **Critical Job**：是否以最高優先層級來執行，並有以下的控制特性：
  - 特性(1) -- Critical Job 一旦在排程時間條件滿足後不會允許未開始執行的一般作業執行。如下：

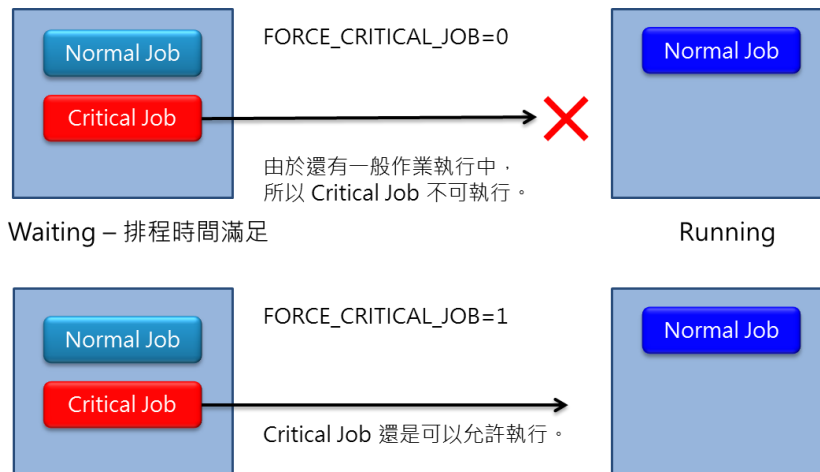


- 特性(2) -- Critical Job 在排程時間條件滿足後，會檢查當時是否有一般作業還在

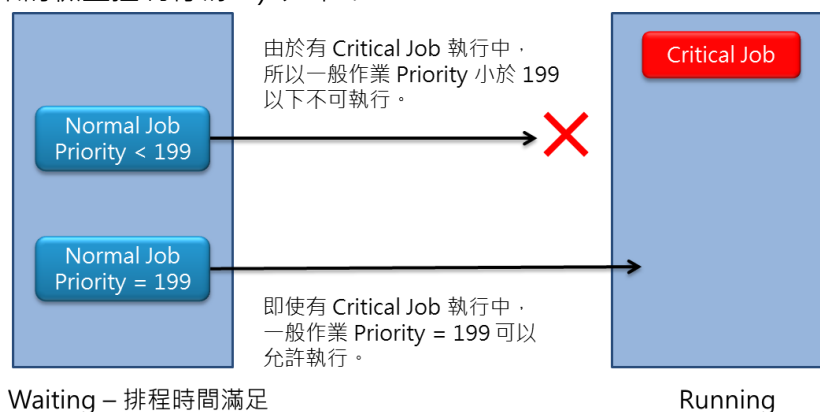
執行中。內定行為會等所有一般作業完成後才會執行，也可透過 `jcserver.conf` 中的設定值忽略等待執行中的作業：

在 `<Trinity>/JCSServer/cfg/jcserver.conf` 中增加一個參數

`FORCE_CRITICAL_JOB = [0,1]`。如果沒有此參數則內定值為 0。當設定值為 0 時則 Critical Job 必須等到沒有任何一般作業執行才能開始執行。當設定值為 1 時則 Critical Job 排程條件滿足後即使還有一般作業執行中也允許 Critical Job 開始執行。如下：



- 特性(3) --一般作業如果其 Priority 值為 199 的話，可允許在 Critical Job 執行時也可以執行。(在「特性(1)」情境時，Priority 199 作業也是可以允許執行；在「特性(2)」情境時，Priority 199 作業在執行中不會影響 Critical Job 對一般作業的檢查控制行為。) 如下：



- Created By：建立此作業的使用者。
- Use Step Postcondition：Job 中每個 Step 都可以設定各自的 Postcondition，這裡可以決定是否使用 Step Postcondition 功能。
- Waiting time：配合 Notification 中設定 When Task Waiting Long 時使用，在此設定等待時間，超過等待時間即寄送訊息通知。

## General 頁面各欄位的說明 -- 第六部分

Task Attribute	<input type="checkbox"/> Skip Missing Task	<input type="checkbox"/> Only Apply Completed Task	<input type="checkbox"/> Don't Save History	<input type="checkbox"/> Rerun From FileSource
Online Date	<input type="checkbox"/> Online Date Time	<input type="text" value=""/>	<input type="checkbox"/> Offline Date Time	<input type="text" value=""/>
Job Alert	<input type="checkbox"/> Start In Time	<input type="text" value="00:00"/>	<input type="checkbox"/> Finish In Time	<input type="text" value="00:00"/>
			Avg Time	<input type="text" value="1"/>

- Skip Missing Task：忽略過去時段中排程應該補產生的任務。
- Only Apply Completed Task：檢查若該作業前一次執行的執行結果為未完成（例如狀態為 等待中/執行中/暫停）則不會產生新的任務。
- Don't Save History：只保留當前的執行紀錄。
- Rerun From Filesource：當作業於 Task Console 手動執行 Rerun（重新執行）且 Rerun Method 為「From Beginning Step」時，會從 File Receiving 開始執行。若指定其他 Rerun Method，則不會從 File Receiving 開始。
- Online Date Time：是指定此作業的上線時間點，必需系統時間大於或等於這個時間，才會依照排程時間產生任務來執行。
- Offline Date Time：則是指定此作業的下線時間點，當系統時間超過此時間後就不再對此作業依照排程產生任務來執行。
- Job Alert 則是表示，當此 Job 執行時間超過設定的時間時，則會於 Task Console 中跳出提示訊息。
- 作業的設定值輸入完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。



## 修改 Job

於樹狀視圖中選擇要進行修改的 Job 物件子項目按下滑鼠右鍵從 Popup Menu 中選取「Property」或點選功能視圖中編輯圖示。執行後會在右方工作區中出現類似先前新增 Job 的工作頁面，可進行修改。修改完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。



要注意的是使用者必須對要進行修改的 Job，或 Category 項目或其所在的 Business Entity 項目擁有 Modify 的權限才能執行此修改 Job 的動作。

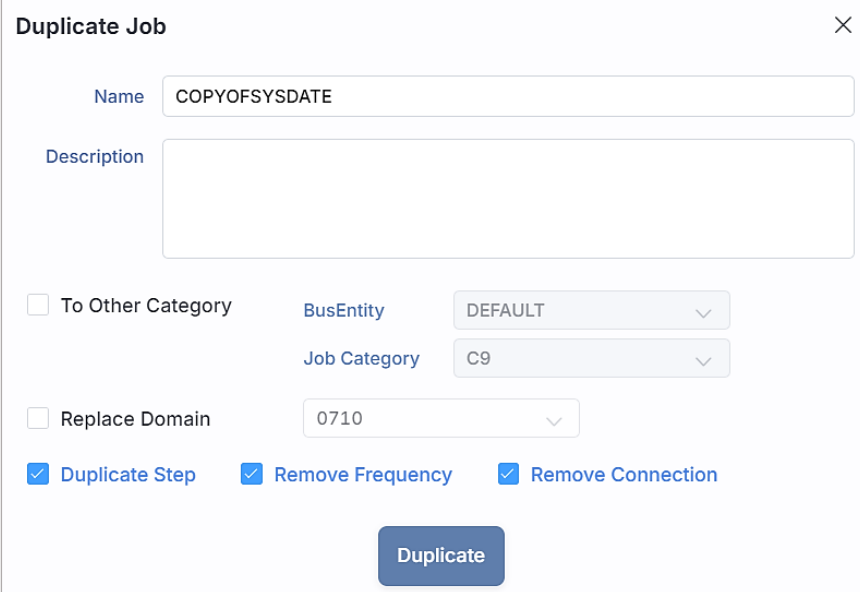
## 刪除 Job

於樹狀視圖中選擇要進行刪除的 Job 物件子項目，按下滑鼠右鍵，從 Popup Menu 中選取「Delete」。請參考基本操作中的關於刪除動作的說明。

要注意的是使用者必須對要進行刪除的 Job，或 Category 項目或其所在的 Business Entity 項目擁有 Delete 的權限才能執行此刪除 Job 的動作。

## 複製 Job

使用者可以透過複製 Job 的方式來節省布署作業的時間。於樹狀視圖中選擇要進行複製的 Job 物件子項目按下滑鼠右鍵，從 Popup Menu 中選取「Duplicate Job」。執行後會出現一個對話視窗詢問使用者要進行複製的選項，例如下圖中所顯示的範例：



The screenshot shows a 'Duplicate Job' dialog box with the following fields and options:

- Name:** COPYOFSYSDATE
- Description:** (Empty text area)
- To Other Category
- BusEntity:** DEFAULT
- Job Category:** C9
- Replace Domain
- Domain:** 0710
- Duplicate Step
- Remove Frequency
- Remove Connection
- Button:** Duplicate

此對話視窗中的「To Other Category」選項是用來指定是否要將作業複製到其他的 Category 中。如果勾選了此選項則需進一步的選擇要搬移過去的 Business Entity 與 Category。

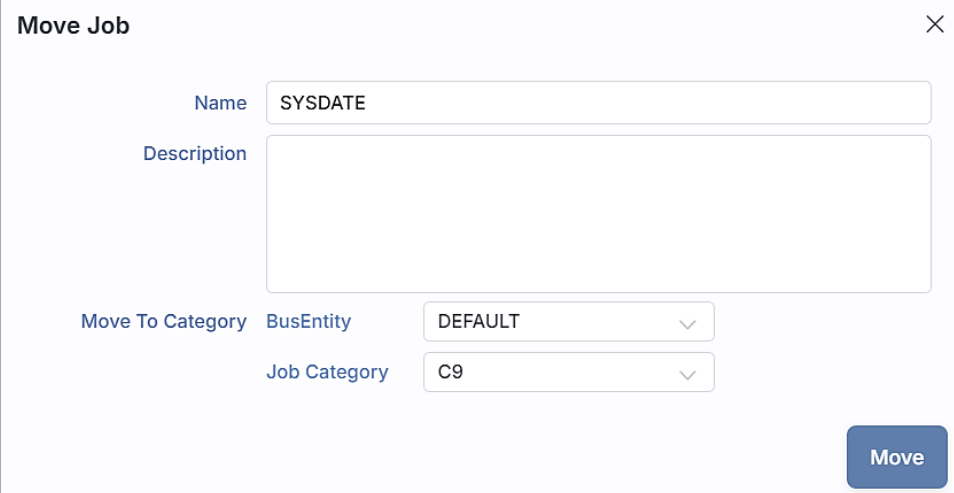
另外「Duplicate Step」的選項必需勾選後才會一併複製 Job 下的 Job Step 設定，同時也可選擇是否一併複製 Frequency 及 connection 設定。選擇完畢後可以按下

「Duplicate」按鈕來執行複製的動作，Job Stream、Job Dependency、Source Table、Target Table 及 Notification 不會一併複製。

要注意的是使用者必須對要複製過去的 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此複製 Job 的動作。

## 移動 Job 到其他 Category

已設定好的作業可以在不同的 Category 間進行移動以便進行分類管理。於樹狀視圖中選擇要進行移動的 Job 物件子項目按下滑鼠右鍵，從 Popup Menu 中選取「Move Job」。執行後會出現一個對話視窗詢問使用者要進行移動的選項，例如下圖中所顯示的範例：



The screenshot shows a dialog box titled "Move Job" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Name:** A text input field containing "SYSDATE".
- Description:** A large empty text area.
- Move To Category:** A section containing two dropdown menus:
  - BusEntity:** A dropdown menu currently showing "DEFAULT".
  - Job Category:** A dropdown menu currently showing "C9".
- Move:** A blue button located at the bottom right of the dialog.

選擇要搬移過去的 Business Entity 與 Category 後可以按下「Move」按鈕來執行移動 Job 的動作。

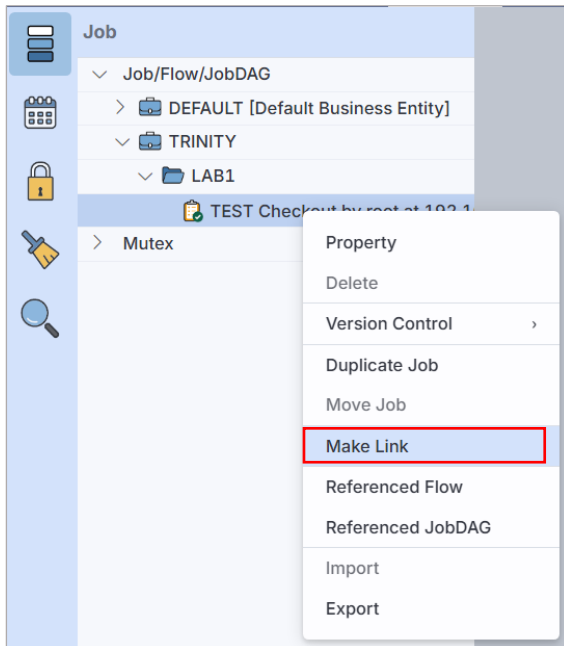
要注意的是使用者必須對要搬移過去的 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此搬移 Job 的動作。

## 新增 Link Job

可利用現存的 Job 來產生 Link Job，所產生的 Link Job 僅可調整其 Job 設定，不可新增 Step，並且會與原始 Job 共用原始 Job 的 Step 內容。

所產生的 Link Job 支援使用 Version Control 版本控管機制。

於樹狀視圖中選擇要建立 Link Job 的 Job 物件子項目/Category 項目，在 Job 物件子項目 / Category 項目上按下滑鼠右鍵後，從 Popup Menu 中選取「Make Link」



設定欲新增的 Link Job：

**Make Link Job** ✕

Auto Append [LINK]

<input checked="" type="checkbox"/>	Name	Description
<input checked="" type="checkbox"/>	ECHO[LINK]	

To Category BusEntity DEFAULT ▼

Category C9 ▼

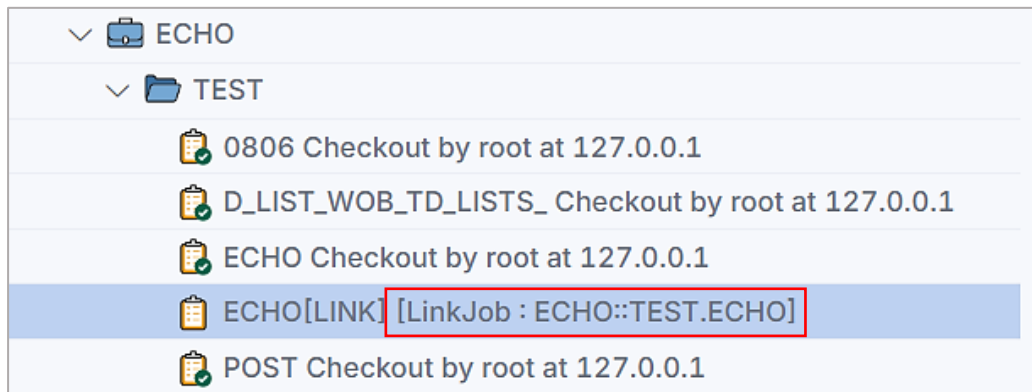
Close
Confirm

- 勾選欲連結的 Job (若前一動作是針對 Job 進行 Make Link，則此畫面中只會有該

Job 一項；若前一動作是針對 Category 進行 Make Link，則此畫面中會顯示該 Category 下的 Job )

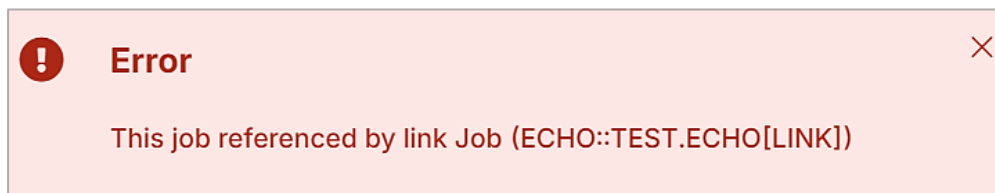
- 勾選 Auto Append 則可將右方輸入格當中的字串加到之後所新增的 Link Job 的 Job Name 上，輸入格的預設值為「 [LINK]」，可做修改。
- 選擇欲產生 Link Job 的 Business Entity 和 Job Category。

已建立的 Like Job，會在樹狀圖裡名稱後方顯示原始 Job 的資訊。



Link Job 無法新增 Step，Step 內容與原始 Job 共用，若原始 Job 修改 Step 內容，則 Link Job 內的 Step 內容也會同步。

刪除 Job 時，若此 Job 存在 Link Job，則會出現警告視窗無法刪除，必須先將 Link Job 刪除才可刪除原始 Job。

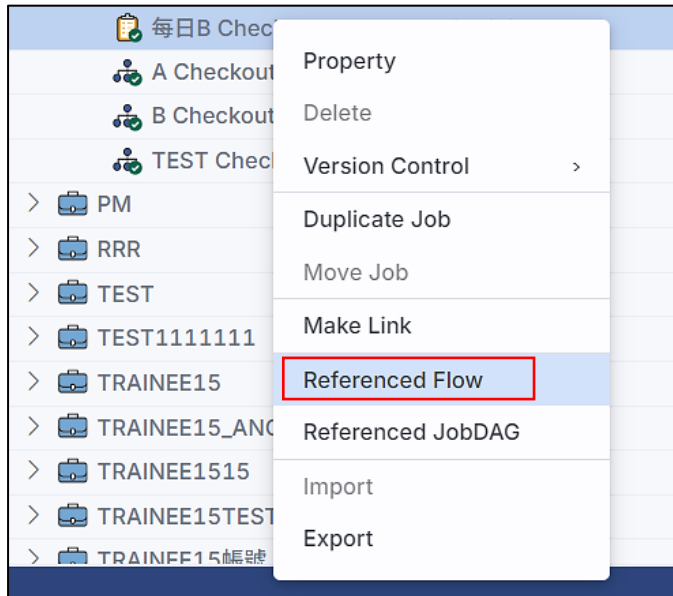


單對 Link Job 做 Export，會連同「原始 Job」和「Link Job 所在的 Category」所設的 Variable 一起匯出。

單對 Link Job 做 Import 的功能，則無法執行。

## Referenced Flow 顯示所有運用到自身 Job 的 Flow 資訊

可利用此功能得知該 Job 被哪些 Flow 所使用。



BusEntity	Category	Name	Description	Activate
ECHO	TEST	A		<input checked="" type="checkbox"/>
ECHO	TEST	B		<input checked="" type="checkbox"/>

[Close](#)

## Referenced JobDAG 顯示所有運用到自身 Job 的 JobDAG 資訊

可利用此功能得知該 Job 被哪些 JobDAG 所使用。

The screenshot shows a list of JobDAGs under the 'TEST1' folder. A context menu is open over the 'A Checkout by...' JobDAG, with 'Referenced JobDAG' highlighted. Below this, the 'Referenced JobDAG' dialog box is displayed, containing a table with the following data:

BusEntity	Category	Name	Description	Activate
TEST	TEST1	JOB DAG		<input checked="" type="checkbox"/>

The dialog box also features a 'Close' button at the bottom right.

## Job 分頁項目介紹

加入一項作業後，編輯頁面除了 General 還有其餘九個分頁項目，分別是 Execution Time, Forecast, Note, Variable, Step, Stream/Trigger, Dependency, Notification。

Job(TEST) ×

BusEntity TRINITY Category LAB1

Name TEST Job Type ETL Job  Activate

Description

General Execution Time Forecast Note Variable Step Stream/Trigger Dependency Notification

Domain Default Agent 192.168.17.53 Created By root

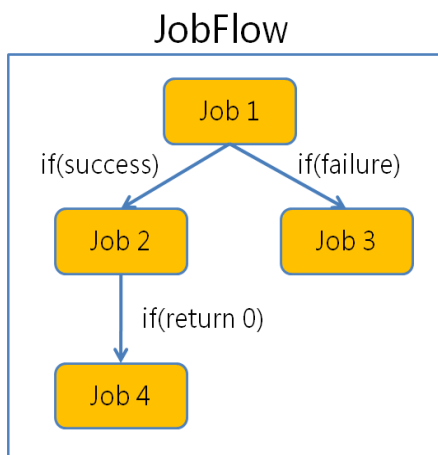
Frequency Select File Source Select

接下來因考量到各項目的說明篇幅，以下並未按照排列順序介紹，再請依照子項目名稱閱讀同名章節即可。

## 作業流程(JobFlow)

在 Trinity 中，作業流程(JobFlow)的定義代表控制 Job 成功、失敗或達其他條件時，下一步該執行哪一個 Job。在 Job 跟 Job 之間可以透過簡單設定，來決定此 JobFlow 的執行流程。但是 Flow 本身僅可定義基本執行設定，並不包含 Job 與 Job 之間的執行流程條件，真正定義執行流程條件的是在 FlowMap 當中。

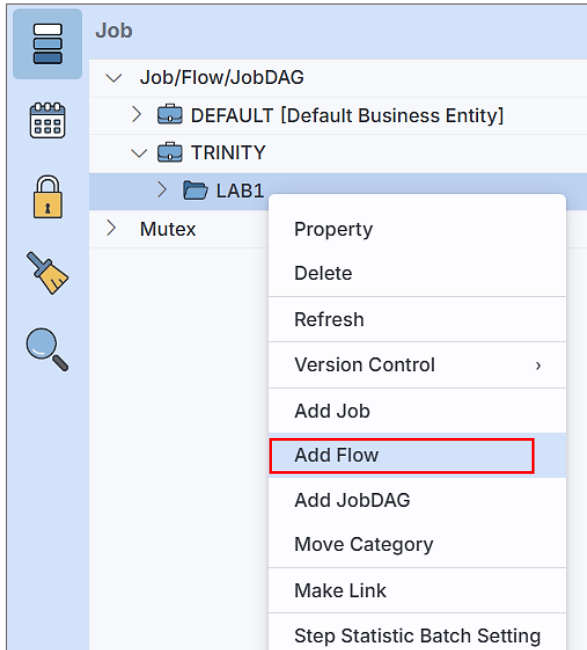
作業依據 FlowMap 的設計來執行，執行結果會檢查 Flow 當中所有實際已經執行的 Job 其末端點的狀態，若有任一端點失敗則回傳 Flow 為執行失敗的狀態。



## 新增 Flow

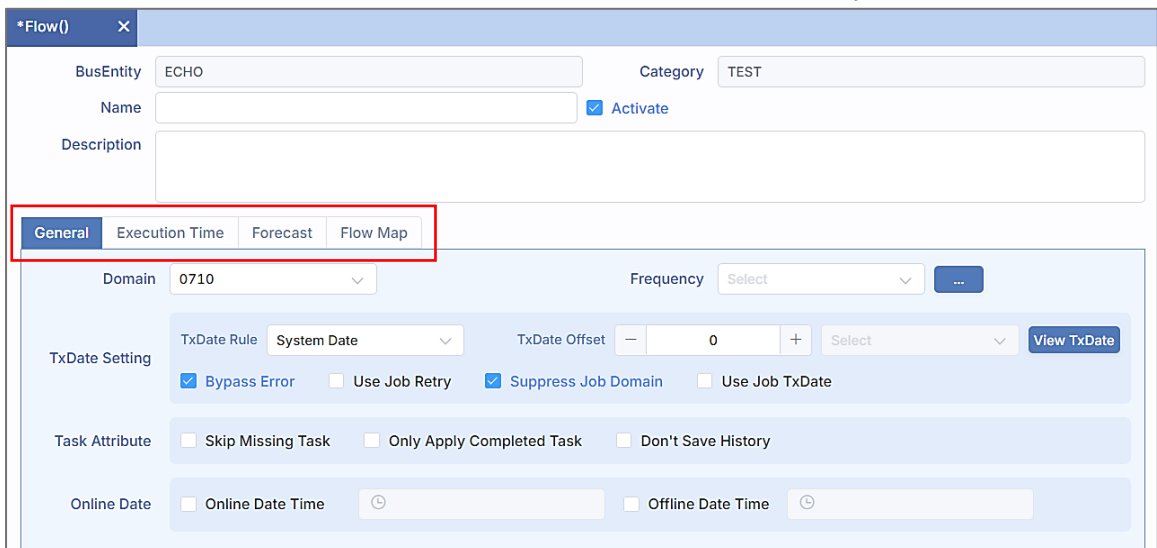
新增 Flow 的動作必須要指定一個 Category，關於 Category 的建立工作的詳細操作請參考 Business Entity 與 Category 的操作說明。

於樹狀視圖中選擇 Category 物件項目後按下滑鼠右鍵，從 Popup Menu 中選取「Add Flow」。要注意的是使用者必須對要進行新增 Job 的 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此新增的動作。



之後會在右方的工作區中顯示一個新增 Flow 的頁面

此畫面中區分為 General、Execution Time、Forecast、Flow Map 四個頁面，如下圖：



以下章節介紹 General 及 Flow Map 頁面內容，其餘畫面的使用方式與 Job 設定畫面中的同名功能相同，可參考本文件「Job/Job Step 設定」章節中「Job 強化功能設定說明」章節。

## General 頁面各欄位的說明 -- 第一部分

- Name：設定此 Flow 的名稱
- Activate：勾選則代表此作業為有效狀態。作業必須是在有效狀態下才會被自動排程模組依照預先設定的時間來啟動執行。
- Description：對此 Flow 的描述說明。
- Domain：選擇此 Flow 所使用的 Domain。Domain 可以用來指定作業中會用的變數，Domain 的下拉選項需至 Admin UI 中進行設定。
- Frequency：選擇此 Flow 所使用的 Frequency。Frequency 可用來指定該作業的排程執行時間，指定後即可在預定的時間點產生 Task 來執行。Frequency 的下拉選項需至 Admin UI 中進行設定。

## General 頁面各欄位的說明 -- 第二部分

- TxDate Rule：設定此作業在排程產生工作時指定的 Transaction Date 的規則，可供選擇的選項有：
  - Beging of Month：TxDate Offset 設定範圍-99~999 的整數，-1 代表 Tx Date 為前一個月的第一天，0 代表 Tx Date 為當月的第一天，1 代表 Tx Date 為後一個月的第一天，2 代表 Tx Date 為後二個月的第一天... 依次類推。所得到的日期格式為 YY-MM-DD。
  - By Calendar-BOM：TxDate Offset 設定範圍-99~999 的整數，-1 代表 TxDate 為 Calendar 設定之下一個月份的第一個可執行日期，0 代表 TxDate 為 Calendar 設定之當月份的第一個可執行日期，1 代表 TxDate 為 Calendar 設定之前一個月份的第一個可執行日期，2 代表 TxDate 為 Calendar 設定之前兩個月份的第一個可執行日期... 依次類推。所得到的日期格式為 YY-MM-DD。
  - By Calendar – EOM：TxDate Offset 設定範圍-99~999 的整數，-1 代表 TxDate 為 CALENDAR 設定之月份下一個月最後一個可執行日期，0 代表 TxDate 為 CALENDAR 設定之月份的最後一個可執行日期，1 代表 TxDate 為 CALENDAR 設定之月份前一個月最後一個可執行日期，2 代表後 TxDate 為

CALENDAR 設定之月份前兩個月最後一個可執行日期... 依次類推。所得到的日期格式為 YY-MM-DD。

- **By Calendar Offset** : TxDate Offset 設定範圍可-99~999 的整數，-1 代表 TxDate 為 CALENDAR 設定之日期後一天，0 代表 TxDate 為 CALENDAR 設定之日期，1 代表 TxDate 為 CALENDAR 設定之日期前一天，2 代表後 TxDate 為 CALENDAR 設定之日期前兩天... 為依次類推。所得到的日期格式為 YY-MM-DD。
- **End of Month** : TxDate Offset 可以從-99~999 的整數，-1 代表 TxDate 為前一個月的最後一天，0 代表 TxDate 為當月的最後一天，1 代表 TxDate 為後一個月的最後一天，2 代表 TxDate 為後二個月的最後一天... 依次類推。所得到的日期格式為 YY-MM-DD。
- **Last Flow TxDate** : TxDate Offset 可以從-99 到 999 的整數，-1 代表 TxDate 為上一次 TxDate 的前一天，0 代表 TxDate 為上一次執行的 TxDate 日期，1 代表 TxDate 為上一次執行 TxDate 的後一天，2 代表 TxDate 為上一次執行 TxDate 的後二天... 依次類推。所得到的日期格式為 YY-MM-DD。
- **Offset Day** : 取得 TxDate 之後的日期，其中的 TxDate Offset 可以從-99 到 999 的整數，-1 代表後一天，0 代表當天，1 代表前一天，2 代表前二天... 依次類推。所得到的日期格式為 YY-MM-DD。
- **System Date** : 取得系統日期，日期格式為 YY-MM-DD。新增 Job 時，系統預設為 System Date。
- **Week Day** : 一週為一個範圍，以星期日為起始日期，TxDate Offset 設定範圍-99~999 的整數，-1 代表星期日的前一天，0 代表當週星期日的日期，1 代表星期日往後推算一天，2 代表星期日往後推算二天... 依次類推。所得到的日期格式為 YY-MM-DD。

範例：今天日期：18/04/20(三)

TxDate Offset	TxDate	Week
-2	18/04/13	星期五
-1	18/04/14	星期六
0	18/04/15	星期日
1	18/04/16	星期一
2	18/04/17	星期二
3	18/04/18	星期三

- TxDate Offset : Transaction Date 的運算值，用來推算出所使用的 Transaction Date 日期。
- TxDate Offset 後方下拉式選單：此項依附 TxDate Rule，僅有 TxDate Rule 選擇 By

Calendar-BOM、By Calendar-EOM 與 By Calendar Offset 時，此選單才可供選擇，選項則有預設之 SYSTEMDAY 與可自行於 Frequency 設定之選項可做選擇。

- Bypass Error：檢查若該作業前一次執行的執行結果為 Error，則本次是否可執行，勾選代表本次仍可執行，未勾選代表本次不可執行，須由維運人員排除前次執行問題，使前次執行成功後，才可執行本次。
- Use Job Retry：勾選代表若 Flow 中的 Job 有設定 Retry Option，則當 Job 執行錯誤時會先執行 Job 各自的 Job Retry。若未勾選，則當 Job 執行錯誤時便會依照 Flow 的設定進行錯誤時的判斷。
- Suppress Job Domain：勾選代表使用 Flow 裡的 Domain 設定來取代 Job 裡的 Domain 設定
- Use Job TxDate：勾選代表 Flow 執行時，底下所有 Job 會以 Flow 的 Task DateTime 作為基準日，再依 Job Txdate Rule 設定推算 Txdate。未勾選代表 Flow 執行時，底下所有 Job 的 Txdate 皆套用 Flow 的 Txdate。

### General 頁面各欄位的說明 -- 第三部分

The screenshot shows a settings panel with two main sections. The 'Task Attribute' section contains three checkboxes: 'Skip Missing Task', 'Only Apply Completed Task', and 'Don't Save History'. The 'Online Date' section contains two checkboxes: 'Online Date Time' and 'Offline Date Time', each followed by a date-time input field with a clock icon.

- Skip Missing Task：忽略過去時段中排程應該補產生的任務。
- Only Apply Completed Task：檢查若該 Flow 前一次執行的執行結果為未完成（例如狀態為 等待中/執行中/暫停）則不會產生新的任務。
- Don't Save History：只保留當前的執行紀錄。
- Online Date Time：是指定此 Flow 的上線時間點，必需系統時間大於或等於這個時間，才會依照排程時間產生任務來執行。
- Offline Date Time：則是指定此 Flow 的下線時間點，當系統時間超過此時間後就不再對此作業依照排程產生任務來執行。

Flow 設定值輸入完畢後，請按存檔圖示，來將設定更新到數據庫中，或者使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。



## 修改 Flow

於樹狀視圖中選擇要進行修改的 Flow 物件子項目按下滑鼠右鍵從 Popup Menu 中選取「Property」或直接點擊兩下。執行後會在右方工作區中出現類似先前新增 Flow 的工作頁面讓使用者輸入修改的資訊。修改完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。



要注意的是使用者必須對要進行修改的 Flow，或 Category 項目或其所在的 Business Entity 項目擁有 Modify 的權限才能執行此修改 Flow 的動作。

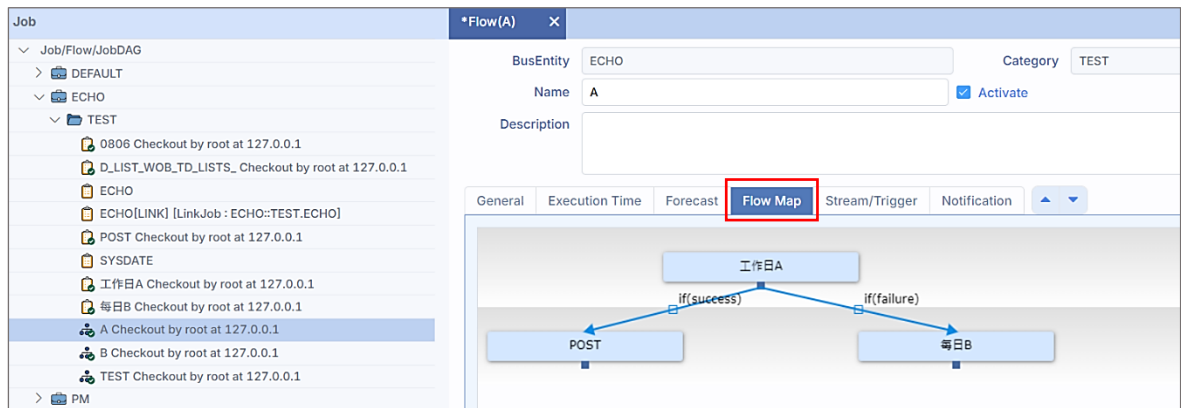
## 刪除 Flow

於樹狀視圖中選擇要進行刪除的 Flow 物件子項目按下滑鼠右鍵，從 Popup Menu 中選取「Delete」。請參考基本操作中的關於刪除動作的說明。

要注意的是使用者必須對要進行刪除的 Flow，或 Category 項目或其所在的 Business Entity 項目擁有 Delete 的權限才能執行此刪除 Flow 的動作。

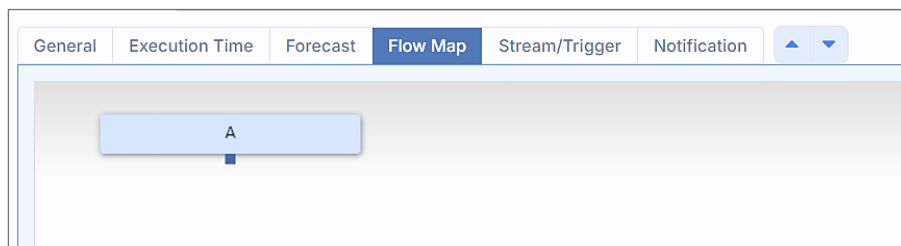
## 編輯 FlowMap

編輯 FlowMap 的動作必須要指定一個 Flow 按下滑鼠右鍵從 Popup Menu 中選取「Property」或直接點擊兩下，開啟編輯 Flow 頁籤，選取「Flow Map」頁面。

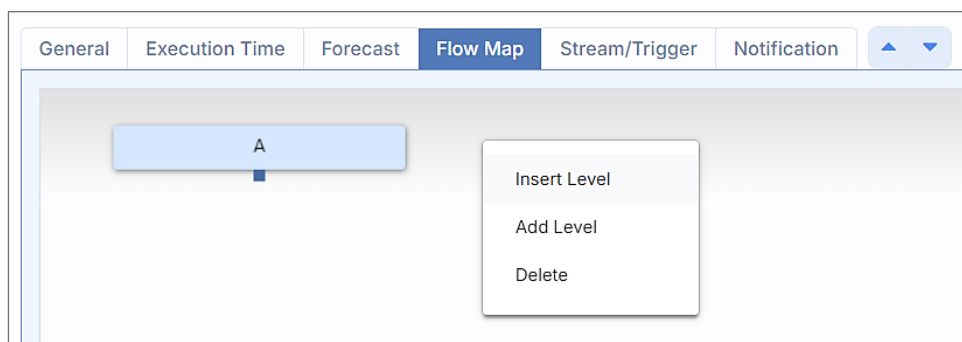


## 新增 Job 到 FlowMap

可以從左方樹狀選單當中，以滑鼠左鍵點擊目標 Job 項目不放並拖拉 Job 項目至右方灰色區塊（名為 Level）中，就會出現 Element 圖示代表此 Job。



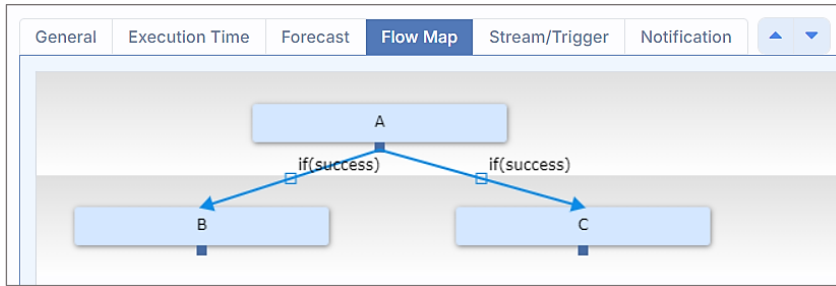
- 同一個 Level 當中可以加入多個 Job (上限為 64 個 Job)。
- 同一個 Level 中的 Job 無法建立條件關係。須為上層 Level 與下層 Level 之間(可跨單層或多層)的 Job Element 才可建立條件關係。
- 若欲新增下層 Level，可在畫面上空白處按滑鼠右鍵，便會出現下列選項：



- Insert Level：在所點選的 Level 區塊上方插入新的 Level 區塊。
- Add Level：在最下方插入新的 Level 區塊。
- Delete：刪除所點選的 Level 區塊或 Job Element。

對 Job Element 以滑鼠左鍵點擊圖下藍點不放，滑鼠拖拉至下層 level 中 Job Element

圖示上，再放開滑鼠左鍵，即可完成 Flow 中 Job 與 Job 之間的 Condition 串連。



### Job 與 Job 之間 Condition 關係

點選 Job 與 Job 之間線上的方塊，雙擊兩下會出現設定 Condition 的畫面

#### Condition Type 操作：

- if(success)：當上方 Job 執行成功時。
- if(failure)：當上方 Job 執行失敗時。
- if(timeout)：當上方 Job 執行超時時。
- if(completed)：當上方 Job 執行完成時
- if(return ?)：當上方 Job 執行後所回傳的值符合。
- else：當上方 Job 執行後並沒有符合其中一個 Condition 時。

#### Condition Grop 操作：

在 flowmap 中，對於 Job 與 Job 之間連線的 Element 邏輯設定加入 ConditionGroup (And、Or、Mandatory) 判斷：

- and：所有勾選為 and 的 condition 需同時成立。
- or：勾選為 or 的 condition 獨立成立即可。
- Mandatory：不管其他 condition 為 and 還是 or，有勾 Mandatory 的 condition 一

定要成立才行。

- 舉例說明：
  - 情境一：

假設目前有三個 Job，分別為 A,B,C

A->C(ConditionGroup=and) · B->C(ConditionGroup=and)

則當 A 與 B 皆執行成功時，才可執行 C。

Flow Element Setting

Condition Type  - 0 +

Condition Group   Mandatory

Flow Element Setting

Condition Type  - 0 +

Condition Group   Mandatory

- 情境二：

假設目前有三個 Job，分別為 A,B,C

A->C(ConditionGroup=or) · B->C(ConditionGroup=or)

則當 A 執行成功時 或 B 執行成功時，就可執行 C。

Flow Element Setting

Condition Type  - 0 +

Condition Group   Mandatory

Flow Element Setting

Condition Type  - 0 +

Condition Group   Mandatory

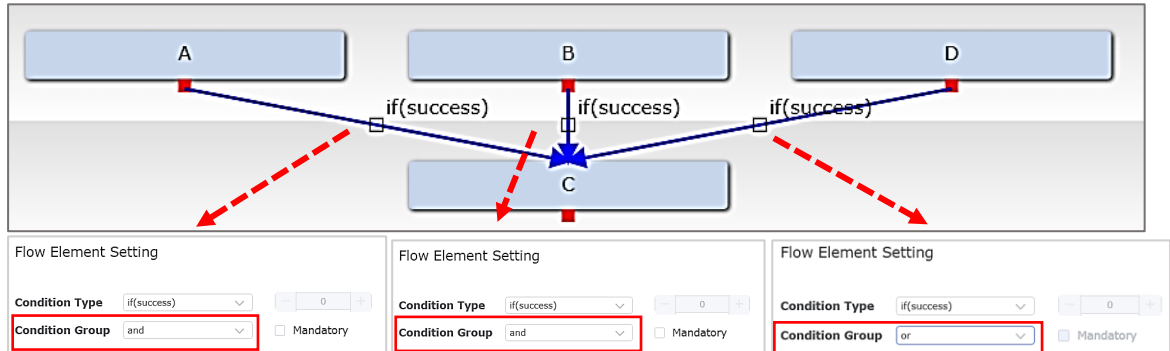
■ 情境三：

假設目前有四個 Job，分別為 A,B,C,D

A->C(ConditionGroup=and) · B->C(ConditionGroup=and) ·

D->C(ConditionGroup=or)

則當 A 與 B 及 D 皆執行成功時 或 D 獨自執行成功時，就可執行 C



■ 情境四：

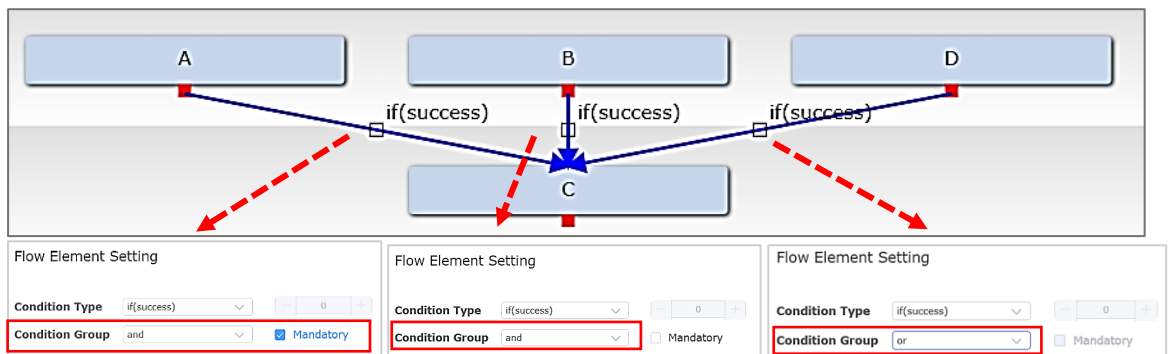
假設目前有四個 Job，分別為 A,B,C,D

A->C(ConditionGroup=and 並勾選 Mandatory) ·

B->C(ConditionGroup=and) ·

D->C(ConditionGroup=or)

一般而言，則當 A 與 B 及 D 皆執行成功時 或 D 獨自執行成功時，就可執行 Job C，但因為 A->C 的 condition 有勾選 Mandatory 的關係，所以就算 D 獨自執行成功時 邏輯上已經足以執行 C，但還是需要 A->C 的 condition 也為成立，才可執行 C。



■ 情境五：

假設目前有四個 Job，分別為 A,B,C,D

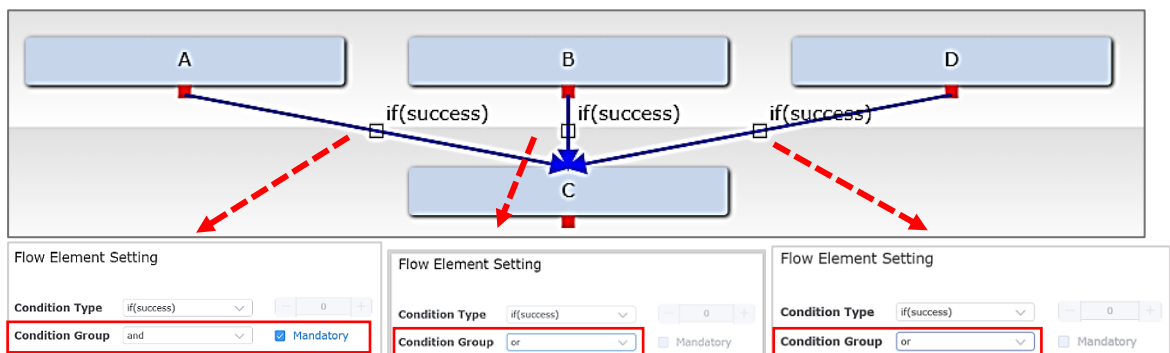
A->C(ConditionGroup=and 並勾選 Mandatory) ·

B->C(ConditionGroup=or) ·

D->C(ConditionGroup=or)

一般而言，當 A 與 B 及 D 執行成功時 或 B 獨自執行成功時 或 D 獨自執行成功時，就可執行 Job C，但因為 A->C 的 condition 有勾選 Mandatory 的關係，所以就算 B 獨自執行成功時 或 D 獨自執行成功時 在邏輯上已經足以執行 C，但還是需要搭配

A->C 的 condition 也為成立，才可執行 C。



■ 情境六：

假設目前有五個 Job，分別為 A,B,C,D,E

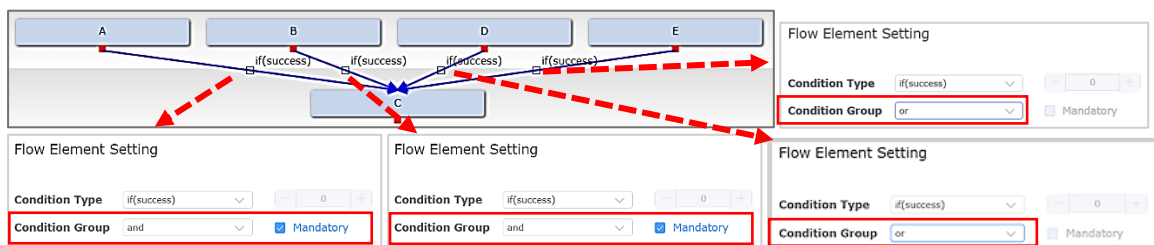
A->C(ConditionGroup=and 並勾選 Mandatory) ·

B->C(ConditionGroup= and 並勾選 Mandatory) ·

D->C(ConditionGroup= and) ·

E->C(ConditionGroup=or)

一般而言，當 A 與 B 與 D 及 E 皆執行成功時 或 E 獨自執行成功時，就可執行 Job C，但因為 A->C 的 condition 有勾選 Mandatory、B->C 的 condition 有勾選 Mandatory 的關係，所以就算 E 獨自執行成功時 在邏輯上已經足以執行 C，但還是需要搭配 A->C 的 condition 為成立 且 B->C 的 condition 為成立，才可執行 C。



※注意：

當 AND 及 OR 在同一 level 共同合使用時，AND 必會等待並判斷其他 Condition Group 的 Condition Type 條件是否成立。

舉例 1：AND 及 OR 共同使用時，OR 的 JOB 已先完成且符合 Condition Type 條件，OR 的 JOB 可直接往下一 level 執行。

但若 AND 的 JOB 先完成且符合 Condition Type 條件時，AND 會等待並判斷 OR 是否有完成且符合 Condition Type 條件，若成立，才能往下一 level 執行。

舉例 2：多個 AND 及 OR 共同使用時，若 AND 與其他 AND 條件皆已成立時，會持續等待判斷 OR 是否為成立，

才能繼續往下一個 level 執行。但此例若 OR 得條件已符合，OR 的 JOB 可直接往下一 level 執行。

## 作業條件式流程(JobDAG)

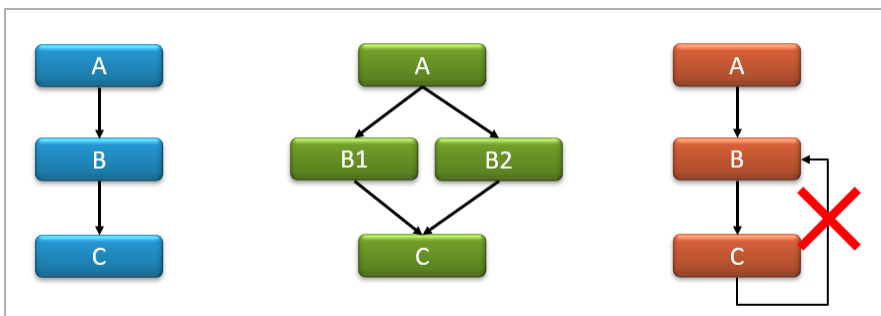
在 Trinity 中，JobDAG 用來控制群組作業一定的執行順序，Job 成功、失敗或達其他條件時，下一步該執行哪一個 Job。DAG 為 Directed Acyclic Graph 的縮寫，又稱「有向無環圖」，是由點 (Job) 與邊 (Condition) 構成的圖。

在 Job 跟 Job 之間可以透過簡單設定，來決定此 JobDAG 的執行流程，當中有起點與終點，且沒有階層的定義限制。

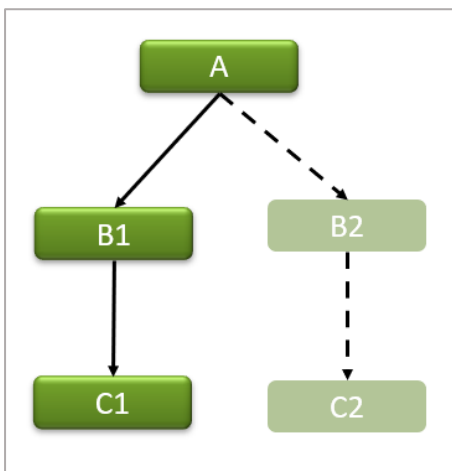
JobDAG 本身僅可定義基本執行設定，並不包含 Job 與 Job 之間的執行流程條件，真正定義執行流程條件的是在 JobDAG Map 當中，作業依據 JobDAG Map 的設計來執行。

以下分幾種情境，圖示說明：

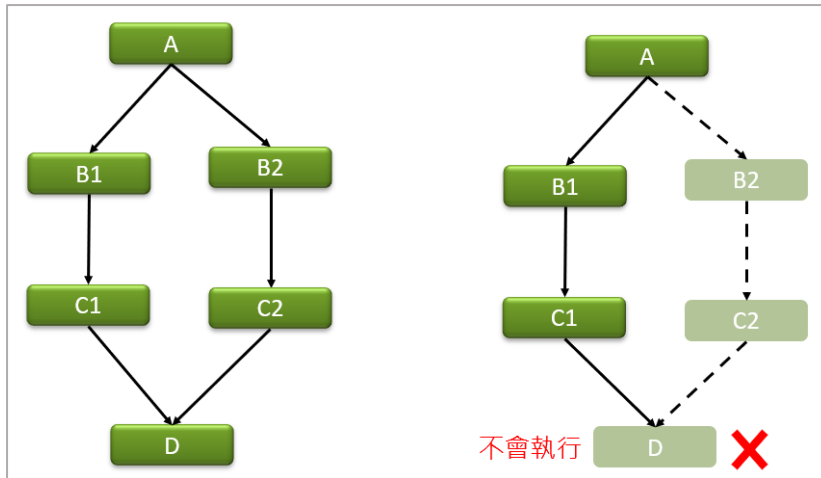
- JobDAG 不能走向先前經過的點造成循環。



- JobDAG 圖中的路徑會依據條件判斷是否需要執行。

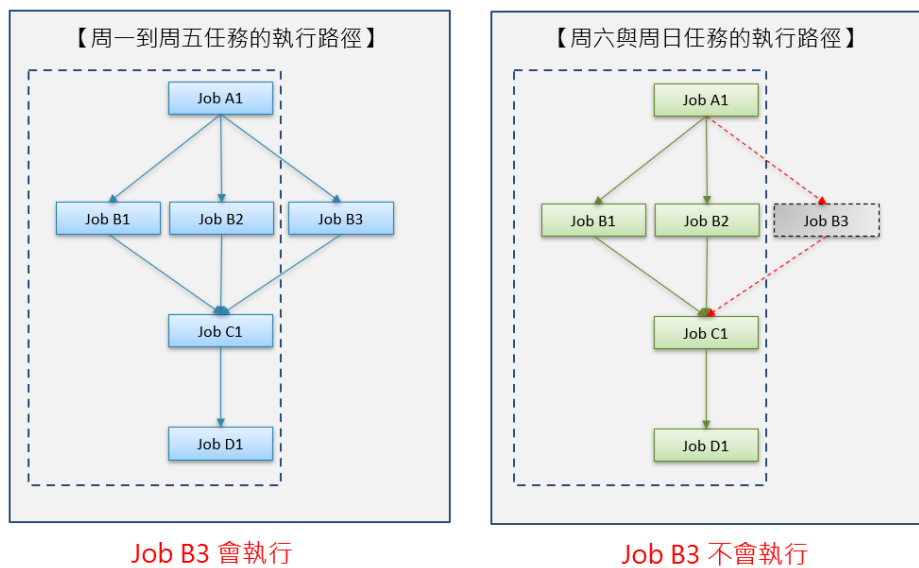


- 路徑條件若是成立的，下游作業將會執行；但路徑條件若有不成立的狀況，那麼不成立路徑其後的作業就會被忽略不執行。

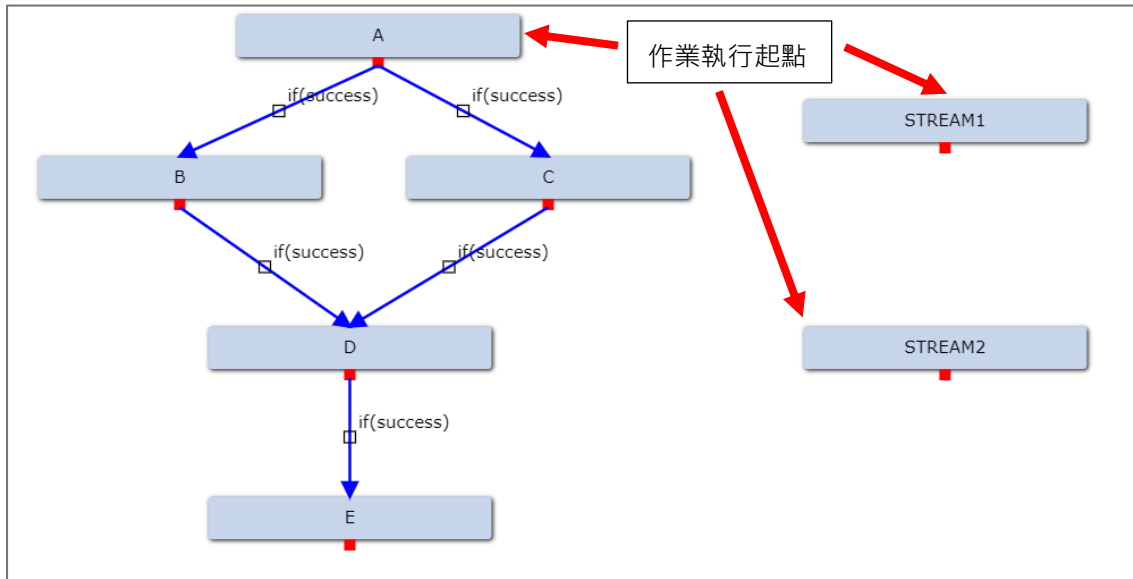


- JobDAG 中的作業將依照作業排程日期決定對應作業是否執行。

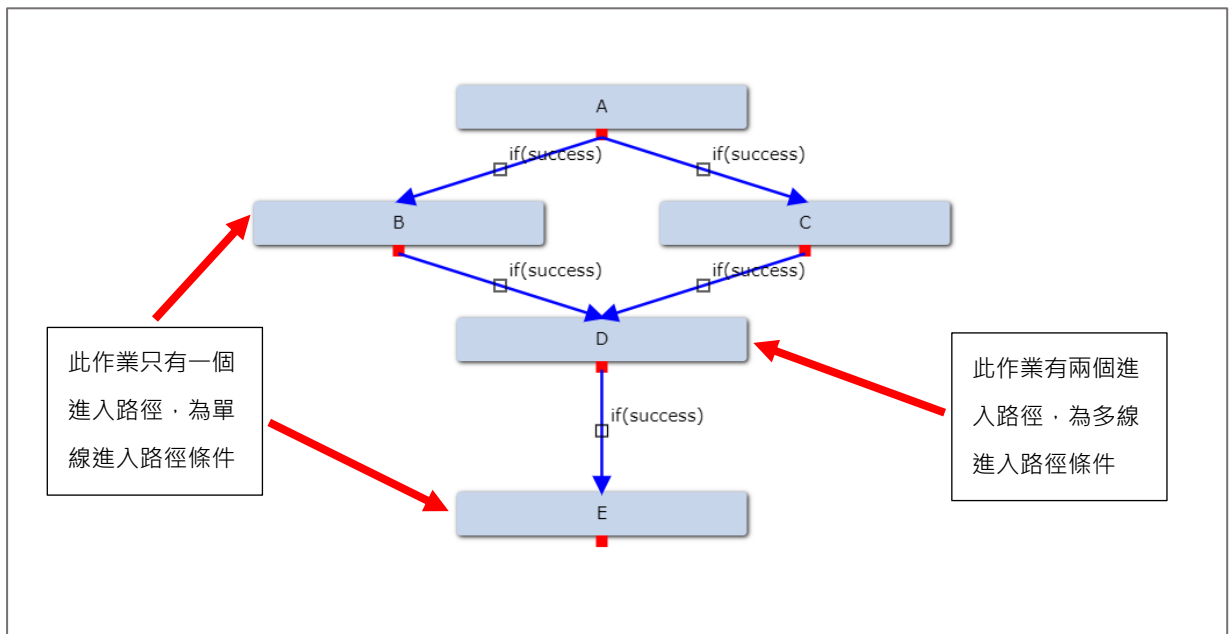
如下圖 JobDAG 設定每日執行，其中 Job B3 設置為周一至周五執行。



- JobDAG 中任何沒有進入路徑條件的作業都會被視為起點作業，故 JobDAG 可多起點執行。



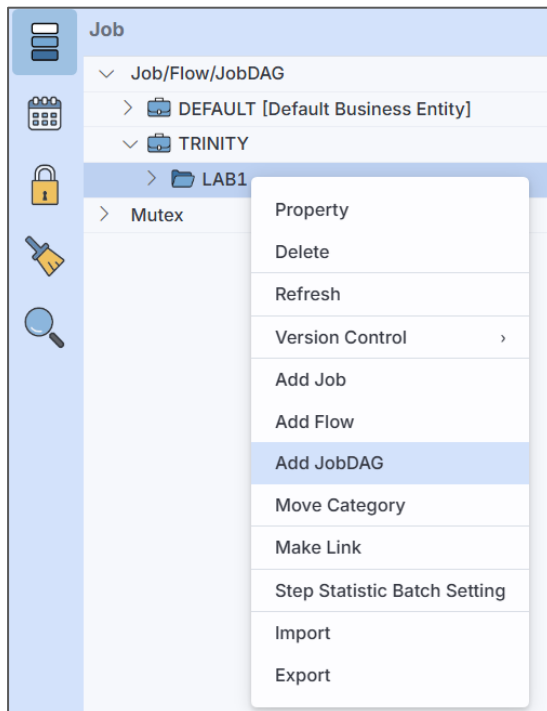
- JobDAG 單線進入路徑條件與多線進入路徑條件。



## 新增 JobDAG

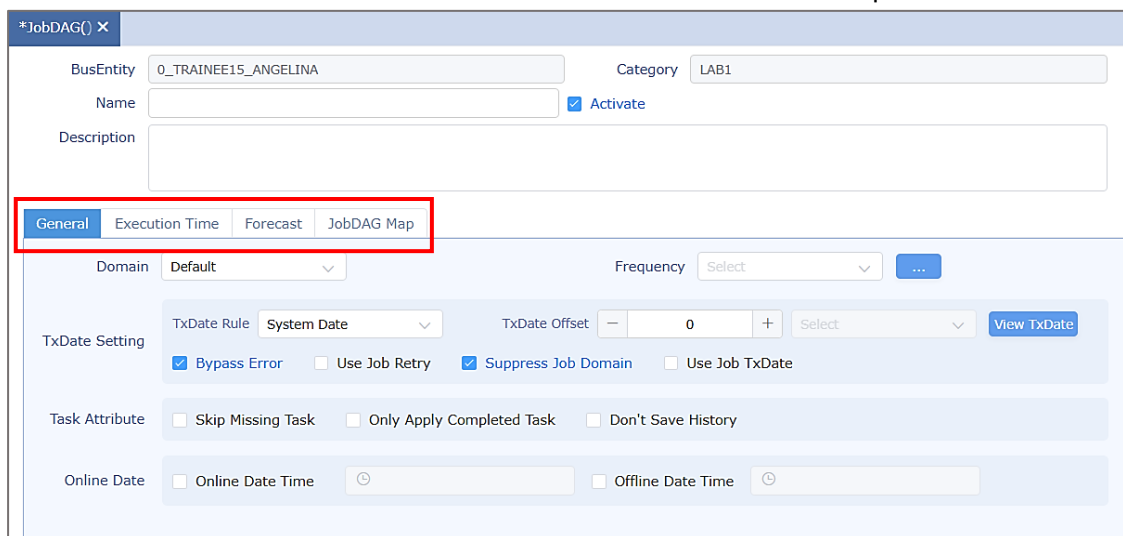
新增 JobDAG 的動作必須要指定一個 Category，關於 Category 的建立工作的詳細操作請參考 Business Entity 與 Category 的操作說明。

於樹狀視圖中選擇 Category 物件項目後按下滑鼠右鍵，從 Popup Menu 中選取「Add JobDAG」。要注意的是使用者必須對要進行新增 Job 的 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此新增的動作。



之後會在右方的工作區中顯示一個新增 JobDAG 的頁面

畫面中區分為 General、Execution Time、Forecast、JobDAG Map 四個頁面，如下圖：



以下介紹 General 頁面內容，其餘畫面的使用方式與 Job 設定畫面中的同名功能相同，可參考本文件「Job/Job Step 設定」章節中「Job 強化功能設定說明」章節。

### General 頁面各欄位的說明 -- 第一部分

- Name：設定此 JobDAG 的名稱
- Activate：勾選則代表此作業為有效狀態。作業必須是在有效狀態下才會被自動排程模組依照預先設定的時間來啟動執行。
- Description：對此 JobDAG 的描述說明。
- Domain：選擇此 JobDAG 所使用的 Domain。Domain 可以用來指定作業中會用的變數，Domain 的下拉選項需至 Admin UI 中進行設定。
- Frequency：選擇此 JobDAG 所使用的 Frequency。Frequency 可用來指定該作業的排程執行時間，指定後即可在預定的時間點產生 Task 來執行。Frequency 的下拉選項需至 Admin UI 中進行設定。

### General 頁面各欄位的說明 -- 第二部分

- TxDate Rule：設定此作業在排程產生工作時指定的 Transaction Date 的規則，可供選擇的選項有：
  - Beging of Month：TxDate Offset 設定範圍-99~999 的整數，-1 代表 Tx Date 為前一個月的第一天，0 代表 Tx Date 為當月的第一天，1 代表 Tx Date 為後一個月的第一天，2 代表 Tx Date 為後二個月的第一天... 依次類推。所得到的日期格式為 YY-MM-DD。
  - By Calendar-BOM：TxDate Offset 設定範圍-99~999 的整數，-1 代表 TxDate 為 Calendar 設定之下一個月份的第一個可執行日期，0 代表 TxDate 為 Calendar 設定之當月份的第一個可執行日期，1 代表 TxDate 為 Calendar 設定

之前一個月份的第一個可執行日期，2 代表 TxDate 為 Calendar 設定之前兩個月份的第一個可執行日期... 依次類推。所得到的日期格式為 YY-MM-DD。

- **By Calendar – EOM**：TxDate Offset 設定範圍-99~999 的整數，-1 代表 TxDate 為 CALENDAR 設定之月份下一個月最後一個可執行日期，0 代表 TxDate 為 CALENDAR 設定之月份的最後一個可執行日期，1 代表 TxDate 為 CALENDAR 設定之月份前一個月最後一個可執行日期，2 代表後 TxDate 為 CALENDAR 設定之月份前兩個月最後一個可執行日期... 依次類推。所得到的日期格式為 YY-MM-DD。
- **By Calendar Offset**：TxDate Offset 設定範圍可-99~999 的整數，-1 代表 TxDate 為 CALENDAR 設定之日期後一天，0 代表 TxDate 為 CALENDAR 設定之日期，1 代表 TxDate 為 CALENDAR 設定之日期前一天，2 代表後 TxDate 為 CALENDAR 設定之日期前兩天... 為依次類推。所得到的日期格式為 YY-MM-DD。
- **End of Month**：TxDate Offset 可以從-99~999 的整數，-1 代表 TxDate 為前一個月的最後一天，0 代表 TxDate 為當月的最後一天，1 代表 TxDate 為後一個月的最後一天，2 代表 TxDate 為後二個月的最後一天... 依次類推。所得到的日期格式為 YY-MM-DD。
- **Last JobDAG TxDate**：TxDate Offset 可以從-99 到 999 的整數，-1 代表 TxDate 為上一次 TxDate 的前一天，0 代表 TxDate 為上一次執行的 TxDate 日期，1 代表 TxDate 為上一次執行 TxDate 的後一天，2 代表 TxDate 為上一次執行 TxDate 的後二天... 依次類推。所得到的日期格式為 YY-MM-DD。
- **Offset Day**：取得 TxDate 之後的日期，其中的 TxDate Offset 可以從-99 到 999 的整數，-1 代表後一天，0 代表當天，1 代表前一天，2 代表前二天... 依次類推。所得到的日期格式為 YY-MM-DD。
- **System Date**：取得系統日期，日期格式為 YY-MM-DD。新增 Job 時，系統預設為 System Date。
- **Week Day**：一週為一個範圍，以星期日為起始日期，TxDate Offset 設定範圍-99~999 的整數，-1 代表星期日的前一天，0 代表當週星期日的日期，1 代表星期日往後推算一天，2 代表星期日往後推算二天... 依次類推。所得到的日期格式為 YY-MM-DD。

範例：今天日期：24/04/20(三)

TxDate Offset	TxDate	Week
-2	24/04/13	星期五
-1	24/04/14	星期六

0	24/04/15	星期日
1	24/04/16	星期一
2	24/04/17	星期二
3	24/04/18	星期三

- TxDate Offset：Transaction Date 的運算值，用來推算出所使用的 Transaction Date 日期。
- TxDate Offset 後方下拉式選單：此項依附 TxDate Rule，僅有 TxDate Rule 選擇 By Calendar-BOM、By Calendar-EOM 與 By Calendar Offset 時，此選單才可供選擇，選項則有預設之 SYSTEMDAY 與可自行於 Frequency 設定之選項可做選擇。
- Bypass Error：檢查若該作業前一次執行的執行結果為 Error，則本次是否可執行，勾選代表本次仍可執行，未勾選代表本次不可執行，須由維運人員排除前次執行問題，使前次執行成功後，才可執行本次。
- Use Job Retry：勾選代表若 JobDAG 中的 Job 有設定 Retry Option，則當 Job 執行錯誤時會先執行 Job 各自的 Job Retry。若未勾選，則當 Job 執行錯誤時便會依照 JobDAG 的設定進行錯誤時的判斷。
- Suppress Job Domain：勾選代表使用 JobDAG 裡的 Domain 設定來取代 Job 裡的 Domain 設定
- Use Job TxDate：勾選代表 JobDAG 執行時，底下所有 Job 會以 JobDAG 的 Task DateTime 作為基準日，再依 Job Txdate Rule 設定推算 Txdate。未勾選代表 JobDAG 執行時，底下所有 Job 的 Txdate 皆套用 JobDAG 的 Txdate。

### General 頁面各欄位的說明 -- 第三部分

Task Attribute  Skip Missing Task  Only Apply Completed Task  Don't Save History

Online Date  Online Date Time    Offline Date Time

- Skip Missing Task：忽略過去時段中排程應該補產生的任務。
- Only Apply Completed Task：檢查若該 JobDAG 前一次執行的執行結果為未完成（例如狀態為 等待中/執行中/暫停）則不會產生新的任務。
- Don't Save History：只保留當前的執行紀錄。
- Online Date Time：是指定此 JobDAG 的上線時間點，必需系統時間大於或等於這個時間，才會依照排程時間產生任務來執行。
- Offline Date Time：則是指定此 JobDAG 的下線時間點，當系統時間超過此時間後就不再對此作業依照排程產生任務來執行。

JobDAG 設定值輸入完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的

動作。



### 修改 JobDAG

於樹狀視圖中選擇要進行修改的 JobDAG 物件子項目按下滑鼠右鍵從 Popup Menu 中選取「Property」或直接點擊兩下。執行後會在右方工作區中出現類似先前新增 JobDAG 的工作頁面讓使用者輸入修改的資訊。修改完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。



要注意的是使用者必須對要進行修改的 JobDAG，或 Category 項目或其所在的 Business Entity 項目擁有 Modify 的權限才能執行此修改 Flow 的動作。

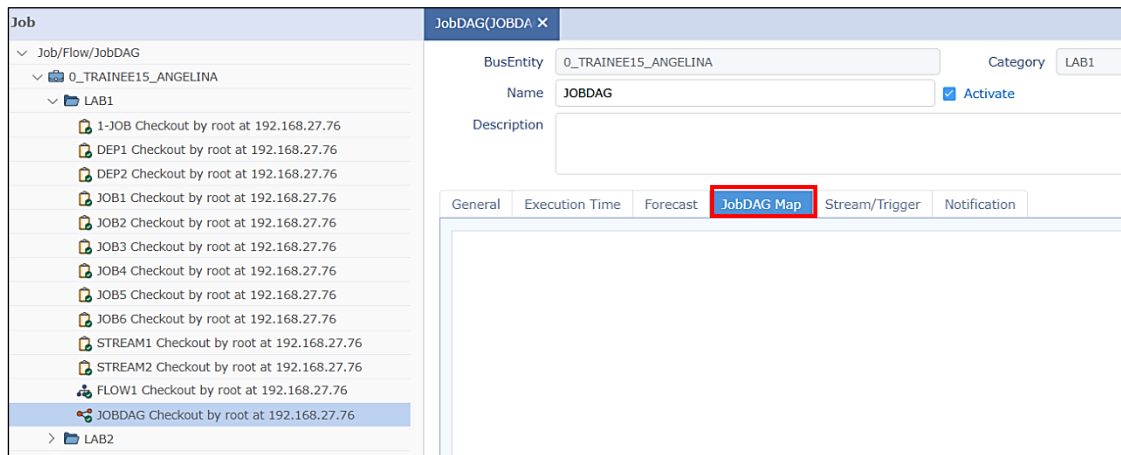
### 刪除 JobDAG

於樹狀視圖中選擇要進行刪除的 JobDAG 物件子項目按下滑鼠右鍵，從 Popup Menu 中選取「Delete」。請參考基本操作中的關於刪除動作的說明。

要注意的是使用者必須對要進行刪除的 JobDAG，或 Category 項目或其所在的 Business Entity 項目擁有 Delete 的權限才能執行此刪除 JobDAG 的動作。

## 編輯 JobDAG Map

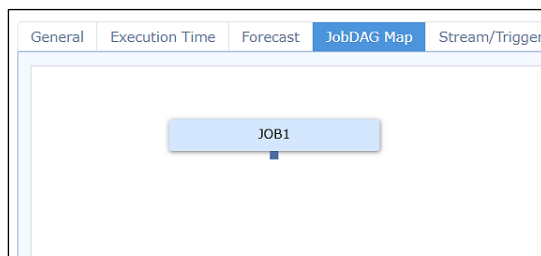
編輯 JobDAG Map 的動作必須要指定一個 JobDAG 按下滑鼠右鍵從 Popup Menu 中選取「Property」或直接點擊兩下，開啟編輯 JobDAG 頁籤，選取「JobDAG Map」頁面。



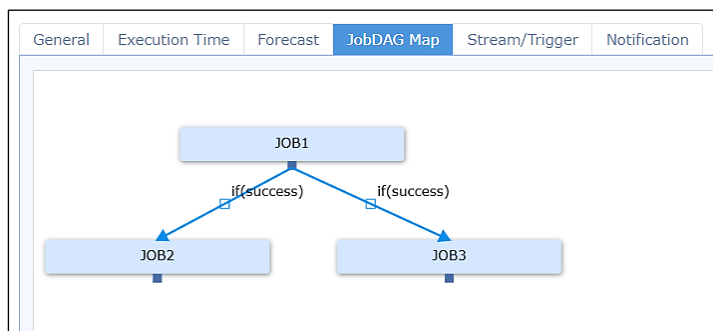
## 新增 Job 到 JobDAG Map

可以從左方樹狀選單當中，以滑鼠左鍵點擊目標 Job 項目不放並拖拉 Job 項目至右方區塊中，就會出現 Element 圖示代表此 Job。

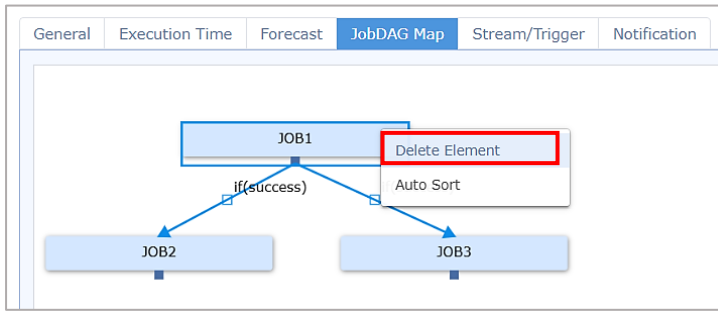
※同一支作業僅能在同一個 JobDAG 中出現一次。



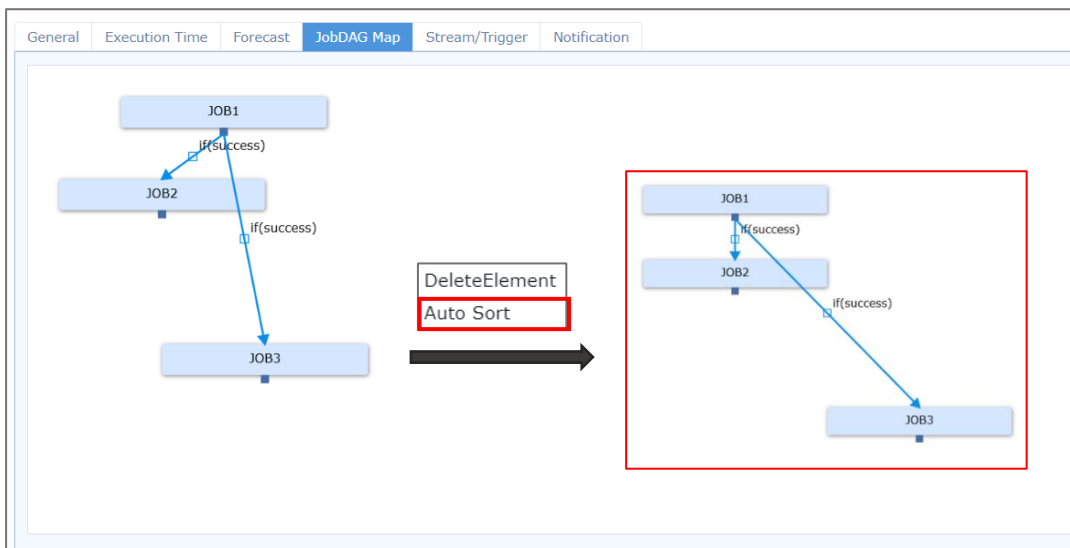
點選上游作業圖示下方紅色連接點，並以滑鼠拖拉，將連接線拖拉至下游作業圖示後放掉既可完成上下游作業條件路徑的連接。



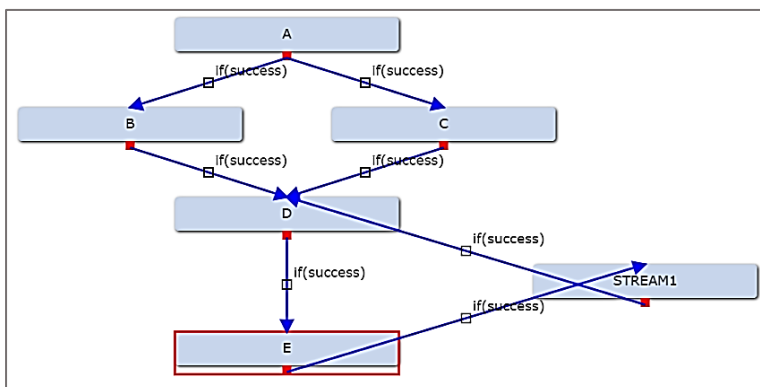
如要刪除作業項目或執行路徑，可針對欲刪除之作業或路徑點選右鍵選取 Delete Element 或點選欲刪除的項目後按下 Delete 鍵即可刪除



JobDAG 中可在空白區塊中按右鍵選擇 **Auto Sort** 功能來自動排列所有作業的位置，或以滑鼠點選作業項目後拖拉到指定的位置來自行編排在 JobDAG 中的位置



設計 JobDAG Map 時若出現循環現象，存檔時會跳出警告訊息，並指出那一個作業造成循環，必須進行修正後才允許存檔。

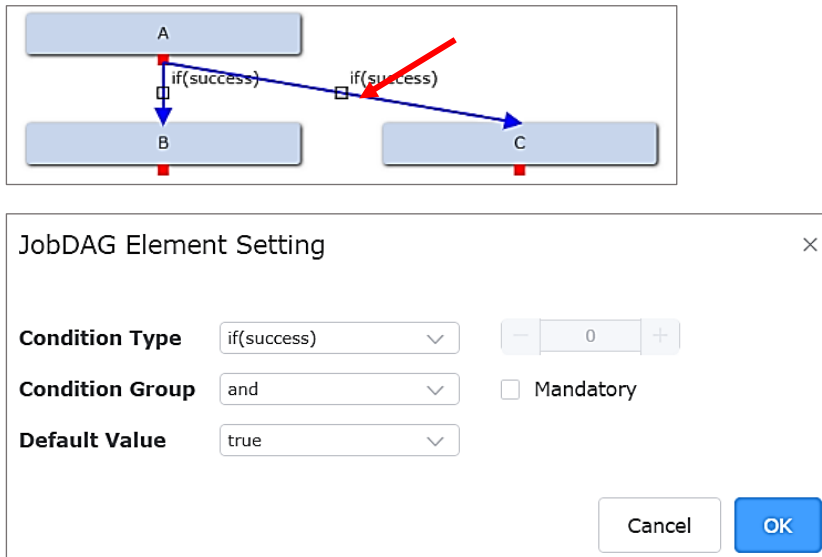


**Warning** ✕

JobDAG validate failed, cause by cyclic job.

## Job 與 Job 之間 Condition 關係

加入上下游作業間的條件路徑時預設會以 `if(success)` 的條件來使用，如要變更條件式內容，可直接雙擊條件連接線中間的方塊，就會開啟條件式的設定視窗。



### Condition Type 操作：

- `if(success)`：當上游作業執行結果成功時條件成立。
- `if(failure)`：當上游作業執行結果失敗時條件成立。
- `if(timeout)`：當上游作業執行結果 Timeout(未收取足夠來源檔案)時條件成立。
- `if(completed)`：當上游作業執行結束(不論結果為何)時條件成立。
- `if(return ?)`：當上游作業執行結果最後步驟的結束碼等於指定的值時條件成立。
- `else`：當上游作業有多個條件路徑時，`else` 條件為其他條件式都不符合的情況下才條件成立，否則條件不成立。

### Condition Grop 操作：

在 JobDAG Map 中，對於 Job 與 Job 之間連線的 Element 邏輯設定加入 ConditionGroup (And、Or、Mandatory) 判斷：

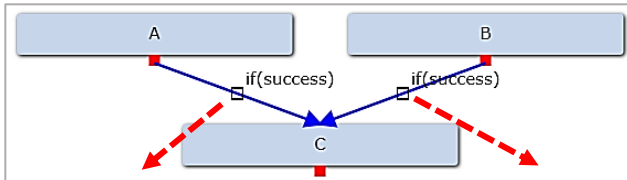
- `and`：所有勾選為 `and` 的 condition 需同時成立。
- `or`：勾選為 `or` 的 condition 獨立成立即可。
- `Mandatory`：不管其他 condition 為 `and` 還是 `or`，有勾 `Mandatory` 的 condition 一定要成立才行。

- 舉例說明：
  - 情境一：

假設目前有三個 Job，分別為 A,B,C

A->C(ConditionGroup=and) · B->C(ConditionGroup=and)

則當 A 與 B 皆執行成功時，才可執行 C。



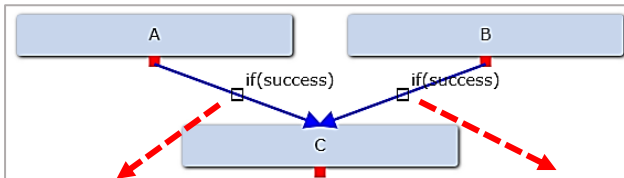
<p>JobDAG Element Setting</p> <p>Condition Type <input type="text" value="if(success)"/> <input type="button" value="-"/> <input type="text" value="0"/> <input type="button" value="+"/> <input type="checkbox"/> Mandatory</p> <p><b>Condition Group</b> <input type="text" value="and"/></p> <p>Default Value <input type="text" value="true"/></p>	<p>JobDAG Element Setting</p> <p>Condition Type <input type="text" value="if(success)"/> <input type="button" value="-"/> <input type="text" value="0"/> <input type="button" value="+"/> <input type="checkbox"/> Mandatory</p> <p><b>Condition Group</b> <input type="text" value="and"/></p> <p>Default Value <input type="text" value="true"/></p>
--	--

- 情境二：

假設目前有三個 Job，分別為 A,B,C

A->C(ConditionGroup=or) · B->C(ConditionGroup=or)

則當 A 執行成功時 或 B 執行成功時，就可執行 C。



<p>JobDAG Element Setting</p> <p>Condition Type <input type="text" value="if(success)"/> <input type="button" value="-"/> <input type="text" value="0"/> <input type="button" value="+"/> <input type="checkbox"/> Mandatory</p> <p><b>Condition Group</b> <input type="text" value="or"/></p> <p>Default Value <input type="text" value="true"/></p>	<p>JobDAG Element Setting</p> <p>Condition Type <input type="text" value="if(success)"/> <input type="button" value="-"/> <input type="text" value="0"/> <input type="button" value="+"/> <input type="checkbox"/> Mandatory</p> <p><b>Condition Group</b> <input type="text" value="or"/></p> <p>Default Value <input type="text" value="true"/></p>
---	---

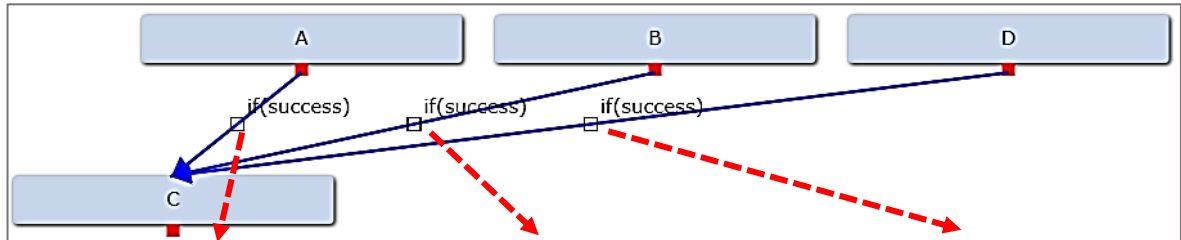
■ 情境三：

假設目前有四個 Job，分別為 A,B,C,D

A->C(ConditionGroup=and) · B->C(ConditionGroup=and) ·

D->C(ConditionGroup=or)

則當 A 與 B 及 D 皆執行成功時 或 D 獨自執行成功時，就可執行 C



<p>JobDAG Element Setting</p> <p>Condition Type: if(success)</p> <p>Condition Group: and</p> <p>Default Value: true</p>	<p>JobDAG Element Setting</p> <p>Condition Type: if(success)</p> <p>Condition Group: and</p> <p>Default Value: true</p>	<p>JobDAG Element Setting</p> <p>Condition Type: if(success)</p> <p>Condition Group: or</p> <p>Default Value: true</p>
---	---	--

■ 情境四：

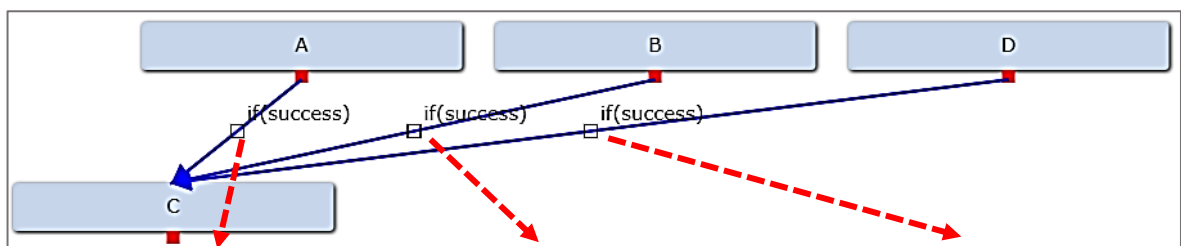
假設目前有四個 Job，分別為 A,B,C,D

A->C(ConditionGroup=and 並勾選 Mandatory) ·

B->C(ConditionGroup=and) ·

D->C(ConditionGroup=or)

一般而言，則當 A 與 B 及 D 皆執行成功時 或 D 獨自執行成功時，就可執行 Job C，但因為 A->C 的 condition 有勾選 Mandatory 的關係，所以就算 D 獨自執行成功時 邏輯上已經足以執行 C，但還是需要 A->C 的 condition 也為成立，才可執行 C。



<p>JobDAG Element Setting</p> <p>Condition Type: if(success)</p> <p>Condition Group: and</p> <p>Mandatory: <input checked="" type="checkbox"/></p> <p>Default Value: true</p>	<p>JobDAG Element Setting</p> <p>Condition Type: if(success)</p> <p>Condition Group: and</p> <p>Mandatory: <input type="checkbox"/></p> <p>Default Value: true</p>	<p>JobDAG Element Setting</p> <p>Condition Type: if(success)</p> <p>Condition Group: or</p> <p>Mandatory: <input type="checkbox"/></p> <p>Default Value: true</p>
---	--	---

■ 情境五：

假設目前有四個 Job，分別為 A,B,C,D

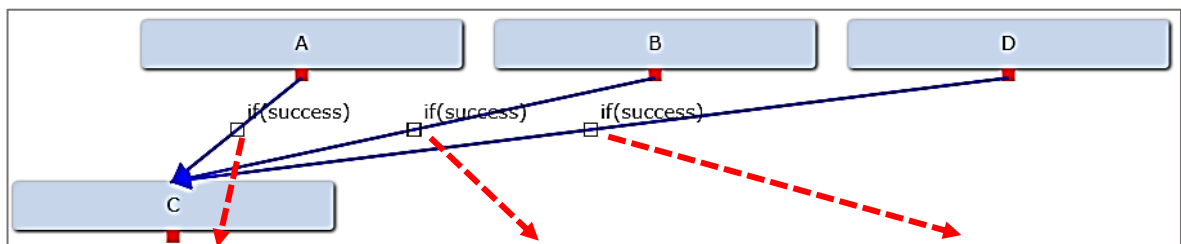
A->C(ConditionGroup=and 並勾選 Mandatory) ·

B->C(ConditionGroup=or) ·

D->C(ConditionGroup=or)

一般而言，當 A 與 B 及 D 執行成功時 或 B 獨自執行成功時 或 D 獨自執行成功時，就可執行 Job C，但因為 A->C 的 condition 有勾選 Mandatory 的關係，所以就算 B 獨自執行成功時 或 D 獨自執行成功時 在邏輯上已經足以執行 C，但還是需要搭配

A->C 的 condition 也為成立，才可執行 C。



JobDAG Element Setting	
Condition Type	if(success) 0
Condition Group	and <input checked="" type="checkbox"/> Mandatory
Default Value	true

JobDAG Element Setting	
Condition Type	if(success) 0
Condition Group	or <input type="checkbox"/> Mandatory
Default Value	true

JobDAG Element Setting	
Condition Type	if(success) 0
Condition Group	or <input type="checkbox"/> Mandatory
Default Value	true

■ 情境六：

假設目前有五個 Job，分別為 A,B,C,D,E

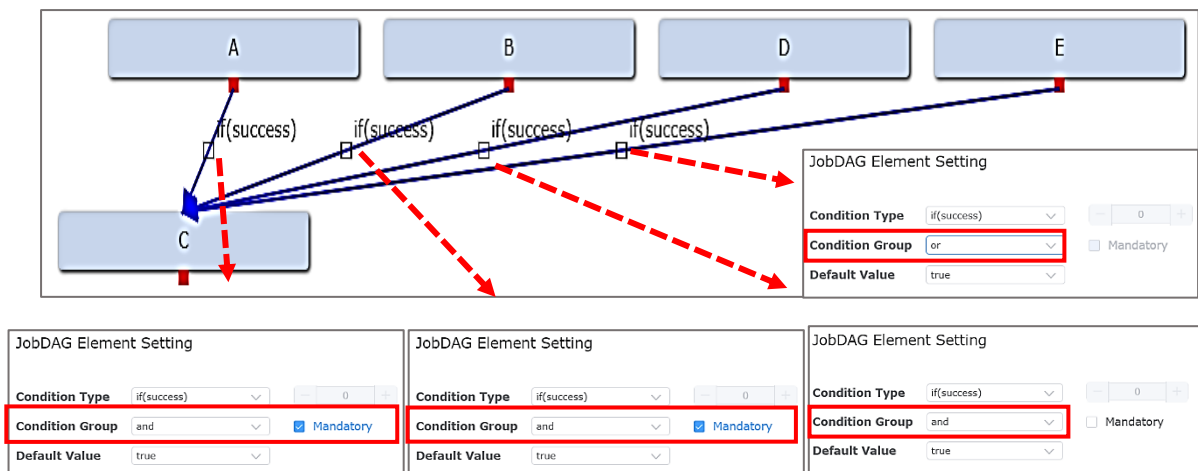
A->C(ConditionGroup=and 並勾選 Mandatory) ·

B->C(ConditionGroup= and 並勾選 Mandatory) ·

D->C(ConditionGroup= and) ·

E->C(ConditionGroup=or)

一般而言，當 A 與 B 與 D 及 E 皆執行成功時 或 E 獨自執行成功時，就可執行 Job C，但因為 A->C 的 condition 有勾選 Mandatory、B->C 的 condition 有勾選 Mandatory 的關係，所以就算 E 獨自執行成功時 在邏輯上已經足以執行 C，但還是需要搭配 A->C 的 condition 為成立 且 B->C 的 condition 為成立，才可執行 C。



※注意：

當 AND 及 OR 在同一 level 共同合使用時，AND 必會等待並判斷其他 Condition Group 的 Condition Type 條件是否成立。

舉例 1：

AND 及 OR 共同使用時，OR 的 JOB 已先完成且符合 Condition Type 條件，OR 的 JOB 可直接往下一 level 執行。但若 AND 的 JOB 先完成且符合 Condition Type 條件時，AND 會等待並判斷 OR 是否有完成且符合 Condition Type 條件，若成立，才能往下一 level 執行。

舉例 2：

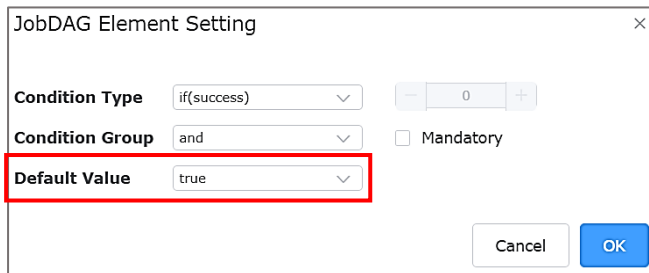
多個 AND 及 OR 共同使用時，若 AND 與其他 AND 條件皆已成立時，會持續等待判斷 OR 是否為成立，才能繼續往下一個 level 執行。但此例若 OR 得條件已符合，OR 的 JOB 可直接往下一 level 執行。

## Default Value 操作

用來指定當此條件的上游作業因為日期規則不成立未執行時此條件式狀態的預設是否成立：

- true：上游作業未執行時條件式也成立。
- false：條件不成立，此路徑以下的作業都會被忽略不執行。

※需注意如果下游作業有多重條件式，即使為 false 也會被視為成立進入到下游作業的執行條件的判斷。



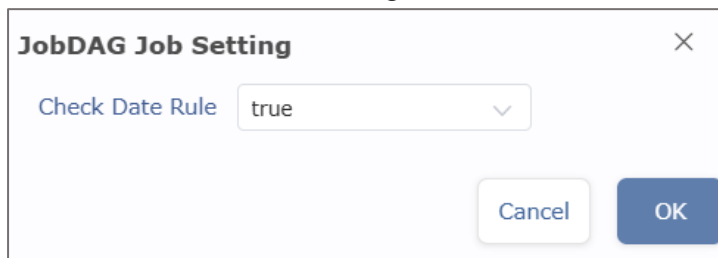
JobDAG Element Setting dialog box showing configuration options:

- Condition Type: if(success)
- Condition Group: and
- Mandatory:
- Default Value: true (highlighted with a red box)

Buttons: Cancel, OK

## Check Date Rule 操作

JobDAG 中的作業若要進一步以 Task Date 執行日期來決定是否執行，於作業圖示雙擊兩下，開啟 JobDAG Job Setting 視窗，勾選 Check Date Rule 即可。

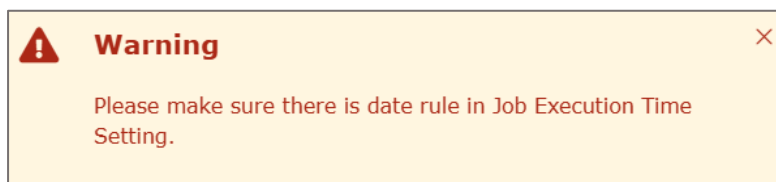


JobDAG Job Setting dialog box showing configuration options:

- Check Date Rule: true

Buttons: Cancel, OK

勾選 Check Date Rule 時會出現訊息視窗提醒要進一步確認此作業中的 Execution Time 是否有正確設定日期匹配規則。

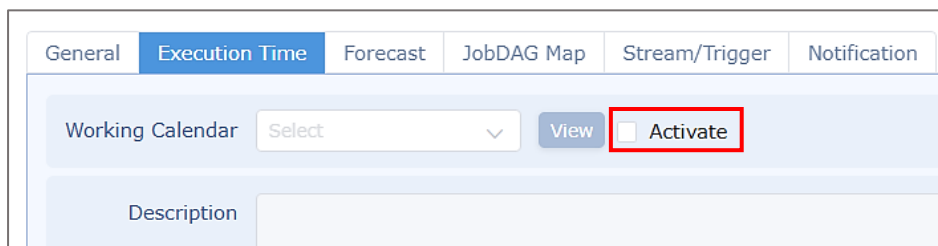


Warning message dialog box:

**Warning**

Please make sure there is date rule in Job Execution Time Setting.

需注意在 Execution Time 不要勾選 Activate 啟用此作業的排程，否則此作業也會產生各自的排程任務來執行



JobDAG Job Setting dialog box - Execution Time tab showing configuration options:

- Working Calendar: Select
- View:
- Activate:  (highlighted with a red box)

Description: [Empty text area]

## Job 強化功能設定說明

### Job Execution Time 頁面設定

Job Execution Time 是在原本的 Frequency 排程機制之外，提供另外一種更具彈性的作業層級的個別作業排程時間條件設定。要將作業的排程條件使用 Execution Time 的方式的話必須先在 General 中將 Frequency 條件清除，否則無法進行 Execution Time 的設定。也就是說在 Job 設定中，Frequency 的使用順序會比 Execution Time 來的優先。一旦 Job 有指定 Frequency 時會忽略 Execution Time 的條件設定。

在 Job 編輯畫面中切換到「Execution Time」頁面後會看到如下圖的畫面：

The screenshot shows the 'Execution Time' configuration page. It features several sections: 'Working Calendar' with a dropdown menu and an 'Activate' checkbox; a 'Description' text area; 'Apply Date Mode' with radio buttons for 'Non-Selected' and 'Selected'; 'Date Rule' with an 'All Day' checkbox and a table with columns 'No', 'Type', and 'Rule'; 'Holiday Calendar' with a dropdown menu and radio buttons for 'Skip', 'Forward', and 'Backward'; and 'Apply Time Mode' with a dropdown menu and a 'Selected Time' table with columns 'No' and 'Task Time'.

No	Task Time
1	0 : 0

在這個頁面中的各項選項設定說明如下：

- **Working Calendar**：工作日設定，Execution Time 的目標日期需藉由 Working Calendar 來定義，在此選擇此作業 Execution Time 要套用的 Working Calendar 物件。後方的「View」功能按鈕提供使用者查看所套用的 Working Calendar 中所選日期為何。
- **Activate**：勾選代表此作業的 Execution Time 設定為有效，必需是有效的狀態才會被系統的排程模組用來產生作業的任務執行，若是失效(Inactivate)的狀態則不會被用來產生任務。
- **Description**：是關於此 Execution Time 設定的描述說明。
- **Apply Date Mode**：設定 Working Calendar 中的日期使用何種套用模式。如果選擇的是「Selected」模式，那麼在 Working Calendar (工作日)中有被選擇的日期會符合要使用的日期條件。而如果選擇的是「Non-Selected」的話，那麼在 Working

Calendar (工作日)中沒有被選擇的日期才符合要使用的日期條件。

- **Date Rule**：在符合的日期條件中設定規則來控制要產生作業任務的日期批配規則，挑選出指定執行日。
- **All Day**：勾選代表所有符合的日期都會用來產生作業任務；未勾選代表需進一步設定日期匹配規則，操作方式請見本章節最後【**Date Rule 設定日期匹配規則**】說明。
- **Holiday Calendar**：休息日設定，可透過此項目來定義要排除的目標日期。定義方式是選擇 **Working Calendar** 物件（可事先建立一個 **Working Calendar** 物件並將要排除的目標日期定義在內）。當 **Working Calendar**(工作日) 由 **Date Rule** 推算出來的日期與 **Holiday Calendar**(休息日)相同，可以選擇 **skip**(忽略不執行)、**Forward**(前一個工作日執行)、**Backward**(下一個工作日執行)，若前一個工作日或下一個工作日已有排程，總共只會執行一次。（請見本段下方【**Working Calendar 工作日 與 Holiday Calendar 休息日 使用範例**】）
- 後方的「View」功能按鈕提供使用者查看所套用的 **Holiday Calendar** 中所選日期為何。

#### 【Working Calendar 工作日 與 Holiday Calendar 休息日 使用範例】

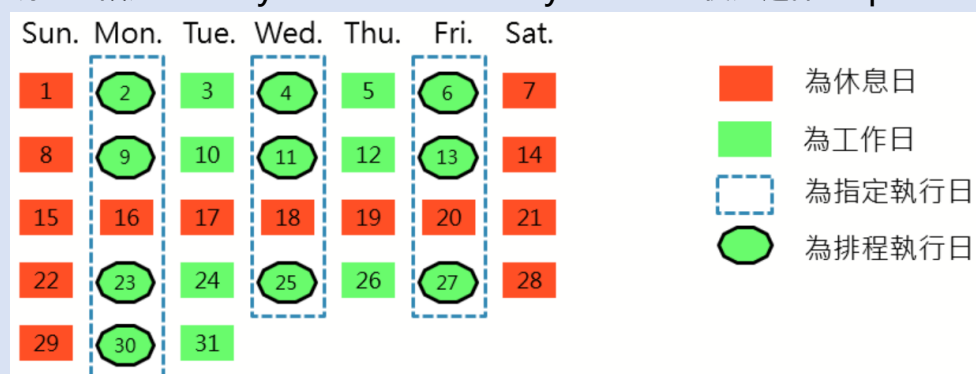
休息日：Holiday Calendar 當中所選日期

工作日：Working Calendar 當中所選日期（須配合 Apply Date Mode 選擇）

指定執行日：Holiday Calendar 當中所選日期配合 Date Rule 設定後所篩選日期

排程執行日：所有設定完成之後的執行結果

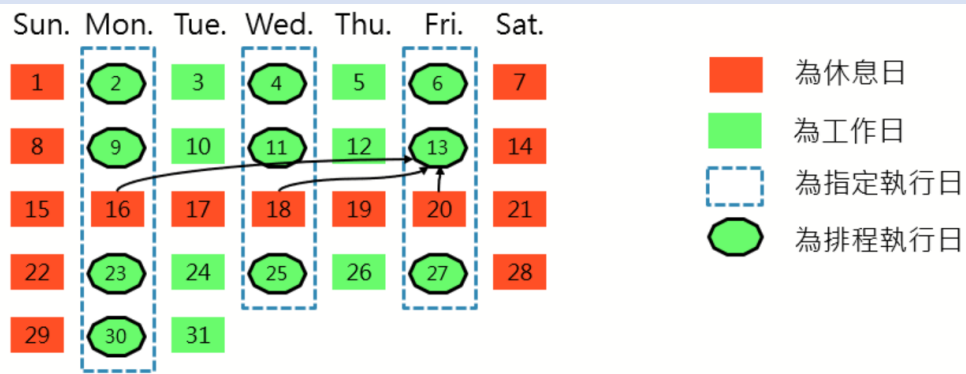
#### 例一：設定 Holiday Calendar 且 Holiday Calendar 後方選擇 Skip



依據 Date Rule 所挑選的指定執行日（藍色虛線）範圍當中包含休息日 16、18、20，故 skip 結果排程執行日期為 2, 4, 6, 9, 11, 13, 23, 25, 27, 30 號

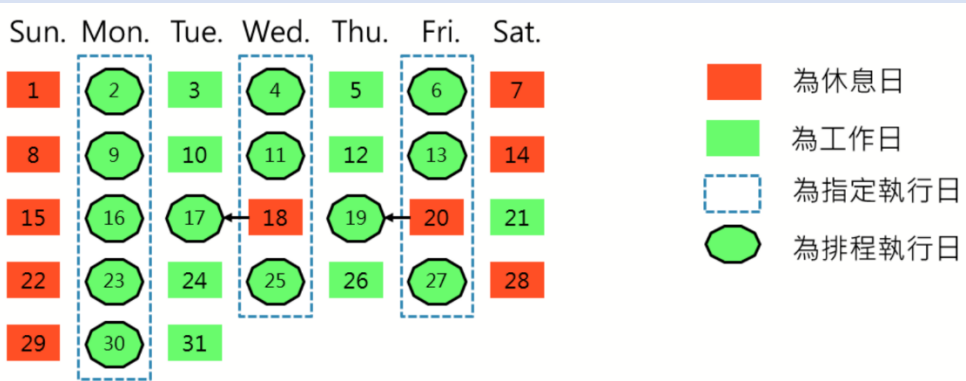
#### 例二：設定 Holiday Calendar 且 Holiday Calendar 後方選擇 Forward

(1)



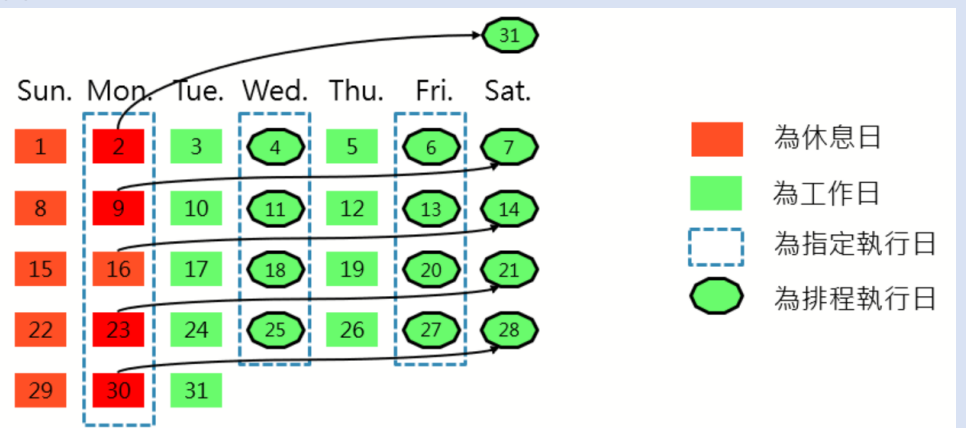
依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 16、18、20。三個休息日往前推一天都是 13。故排程執行日期為 2, 4, 6, 9, 11, 13, 23, 25, 27, 30 號。其中 13 當天此 Job 只會執行一次。

(2)



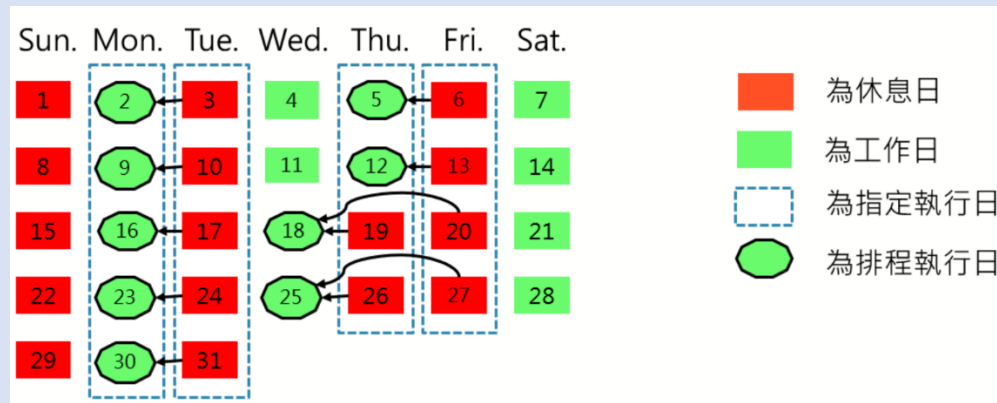
依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 18、20。休息日往前推一天工作日分別是 17、19 日。故排程執行日期為 2, 4, 6, 9, 11, 13, 16, 17, 19, 23, 25, 27, 30 號。其中 17、19 雖不是依據 Date Rule 所挑選的指定執行日，但會執行。

(3)



依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 2、9、16、23、30，往前推一天分別是 31(上個月)、7、14、21、28，故排程執行日期為 31(上個月), 4, 6, 7, 11, 13, 14, 18, 20, 21, 25, 27, 28。其中 31(上個月)、7、14、21、28 雖不是依據 Date Rule 所挑選的指定執行日，但會執行。

(4)



依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 3、6、10、13、17、19、20、24、26、27、31，往前推一天分別是 2、5、9、12、16、18、23、25、30，其中 19、20 都指到 18，26、27 都指到 25，故排程執行日期為 2, 5, 9, 12, 16, 18, 23, 25, 30 號，其中 18, 25 號皆只會執行一次。

- Apply Time Mode 則是用來指定要產生排程的時間條件模式，一共有三種模式，分別是 One Time Job、Multiple Time Job 以及 Continuous Job：
  - One Time Job：代表作業在每一個匹配的排程日期執行一次，所以只可以設定一個時間點，如下面的範例畫面僅定義在早上 6 點 00 分時執行作業任務。

No	Task Time
1	6 : 0

- Multiple Time Job：代表作業在每一個匹配的排程日期可以執行多次，所以可以設定多個時間點，如下面的範例畫面定義在 1:00/2:00/3:00 總共執行三次作業任務。(可逐筆新增，或區間方式自動產生)

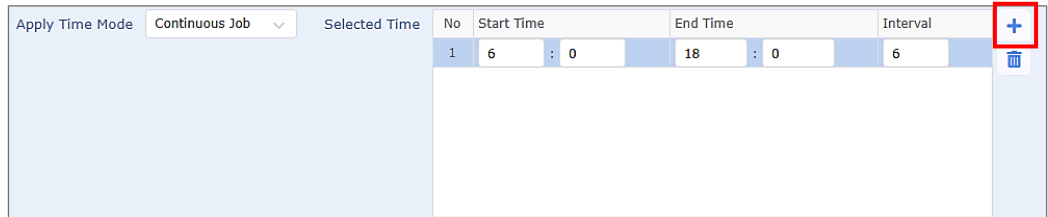
逐筆新增：

No	Task Time
1	1 : 0
2	2 : 0
3	3 : 0

區間方式自動產生：

No	Task Time
1	10 : 0
2	10 : 30
3	11 : 0
4	11 : 30
5	12 : 0

- **Continuous Job**：代表作業在每一個匹配的排程日期可以設定時間區間，在此時間區間內會持續多次執行作業任務，而兩次任務執行之間的等待時間則由 **Interval** 以分鐘為單位來設定（可設定多個區間），如下面的範例畫面定義在 6:00 到 18:00 之間持續性的以 6 分鐘的間隔等待時間來多次執行作業任務：



Apply Time Mode	Continuous Job	Selected Time	No	Start Time	End Time	Interval	
			1	6 : 0	18 : 0	6	+

- \* 要注意的是如果是 **Continuous Job** 的時間條件，實際的作業任務總執行次數是不一定的，因為間隔時間是以前一次作業任務執行完畢後的時間點才會開始計算等待的間隔時間，所以有可能會因為作業任務本身的執行時間較長而讓執行次數變的較少。

### 【Date Rule 設定日期匹配規則】

新增一個日期匹配規則時，會出現一個視窗用來選擇規則的屬性，如下圖：

Rule Type：年、月、週等多種區間選擇，選項會影響下方須設定的欄位

- Year：代表日期匹配規則在計算時是以每一年為區間單位來計算符合的日期天數。( First day、Last day、Day、Interval day 使用說明於本節 Rule type 為 Quarter 的說明之後統一介紹。 )

- Month：代表日期匹配規則在計算時是以每個月為區間單位來計算符合的日期天數，且會出現 Mask 的選項，可勾選特定月份執行，預設為全部月份勾選。( First day、Last day、Day、Interval day 使用說明於本節 Rule type 為 Quarter 的說明之後統一介紹。 )

- **Week**：代表日期匹配規則在計算時是以每一週為區間單位來計算符合的日期天數。( First day、Last day、Day、Interval day 使用說明於本節 Rule type 為 Quarter 的說明之後統一介紹。 )

- **Quarter**：代表日期匹配規則在計算時是以每一季為區間單位來計算符合的日期天數。( First day、Last day、Day、Interval day 使用說明於本節 Rule type 為 Quarter 的說明之後統一介紹。 )

First day、Last day、Day、Interval day 使用說明：

- 只有 Rule type 為 Year、Month、Week、Quarter 才可使用。
- First day：勾選代表在符合的日期區間中使用第一個符合的日期當作排程的日期。
- Last day：勾選代表在符合的區間中使用最後一個符合的日期當作排程的日期。
- Day：用來指定在符合的區間中使用第幾個符合的日期當作排程的日期；輸入負數可指定前一個符合的日期。
- Plus 選項則是以符合的日期為基準，再往後推算 N 天當作排程的日期。
- Interval day 選項是在第一個排程日期之後可重複每間隔 N 天當成下一個排程日期，例如設定值是 3 代表每隔 3 天執行下一次；設定值是 0 代表為單次作業、不會重複執行下一次。
- 舉例說明：

以 Month 為例，設定如下

The screenshot shows a 'Date Rule' configuration window. The 'Rule Type' is 'Month'. There are checkboxes for 'First Day' (checked) and 'Last Day' (unchecked). Below these are input fields for 'Day' (set to 1), a 'Plus' dropdown menu, and an 'Interval Day' field (set to 2). A 'Submit' button is located at the bottom center.

代表以每個月為區間單位，從第 1 個選擇的日期開始，每隔 2 天會當成匹配的排程日期來產生作業任務執行。設定後的畫面如下面的範例：

No	Type	Rule	
1	Month	First day of month, and interval [2] day(s).	+ ^ v ☒

- Calendar\_Week：可依照 Working Calendar(工作日)所勾選的日期，再選擇星期幾執行。( Not Match Action 使用說明於本節 Rule type 為 Calendar\_Year 的說明之後統一介紹。 )

**Date Rule** [X]

Rule Type:

Sunday     Monday     Tuesday     Wednesday  
 Thursday     Friday     Saturday

Not Match Action:  Skip     Forward     Backward

- **Calendar\_Month** : 可依照 Working Calendar(工作日)所勾選的日期，再選擇每月哪幾天執行 ( 可設定多日 )。( Not Match Action 使用說明於本節 Rule type 為 Calendar\_Year 的說明之後統一介紹。 )

**Date Rule** [X]

Rule Type:

1     2     3     4     5     6  
 7     8     9     10     11     12  
 13     14     15     16     17     18  
 19     20     21     22     23     24  
 25     26     27     28     29     30  
 31

Not Match Action:  Skip     Forward     Backward

- **Calendar\_Year** : 可依照 Working Calendar(工作日)所勾選的日期，再選擇每年幾月幾日執行 ( 可設定多日 )。( Not Match Action 使用說明於本節 Rule type 為 Calendar\_Year 的說明之後統一介紹。 )

**Date Rule** [X]

Rule Type:

Date:  /

Not Match Action:  Skip     Forward     Backward

### 【Not Match Action 使用說明】

指定若將 Working Calendar(工作日)依照前述所設定好的日期匹配規則挑選指定

執行日之後，指定執行日卻與 Holiday Calendar(休息日)有所重疊，則作業執行如何因應，此部份可選擇 Skip、Forward、Backward。

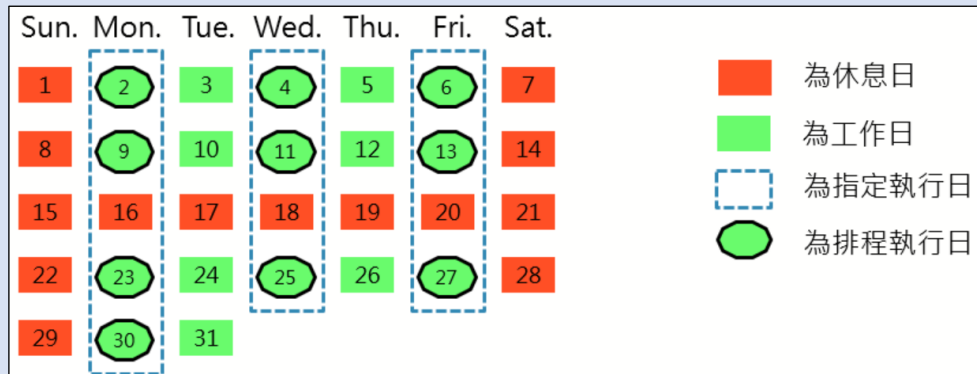
Skip：該日不產生作業執行

Forward：該日不產生作業執行，但提前一個工作天產生作業執行。

Backward：該日不產生作業執行，但往後順延一天工作天產生作業執行。

以 Forward 舉例說明：

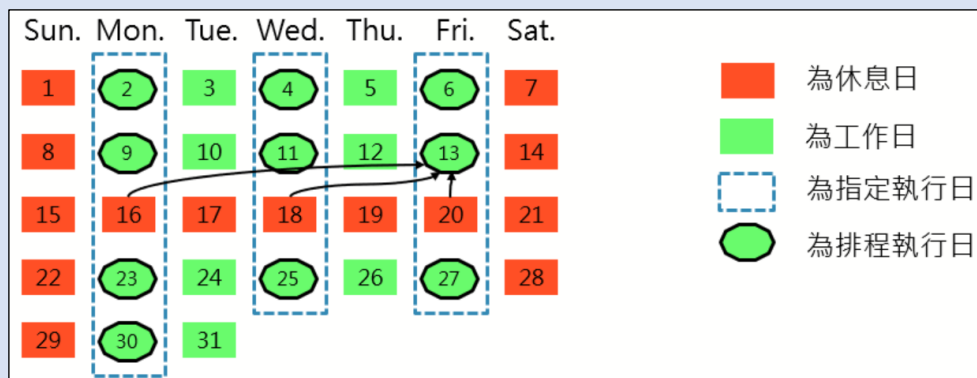
例一：選擇 Skip



依據 Date Rule 所挑選的指定執行日（藍色虛線）範圍當中包含休息日 16、18、20，故 skip 結果排程執行日期為 2, 4, 6, 9, 11, 13, 23, 25, 27, 30 號

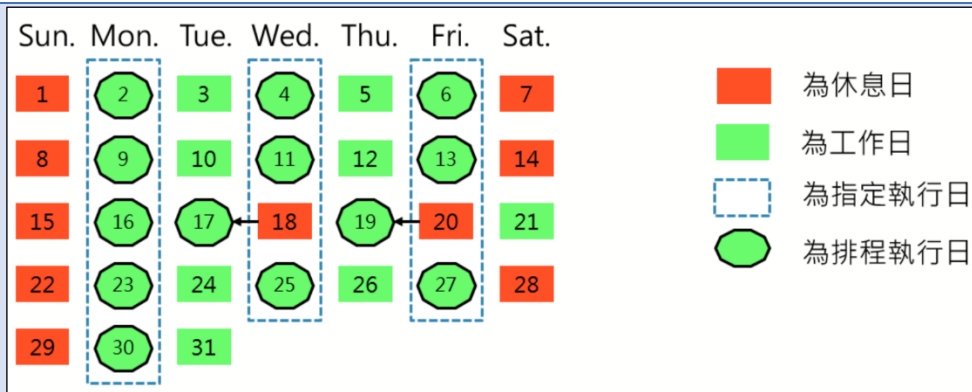
例二：選擇 Forward

(1)



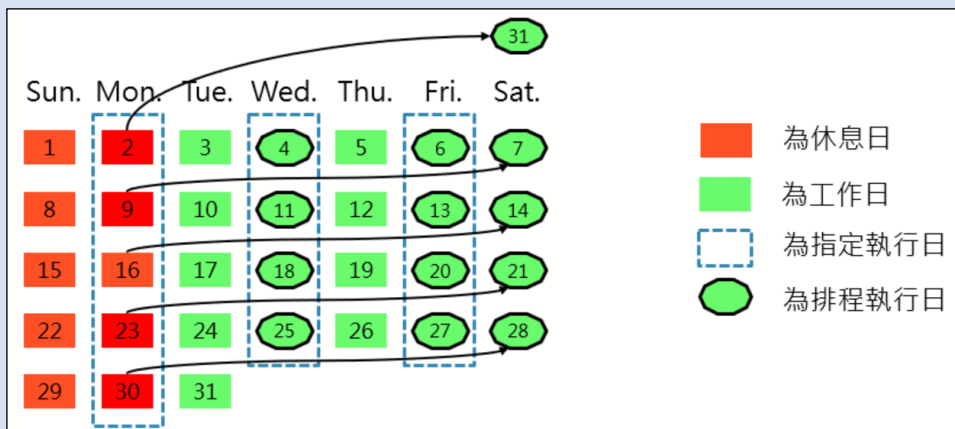
依據 Date Rule 所挑選的指定執行日（藍色虛線）範圍當中包含休息日 16、18、20，三個休息日往前推一天都是 13，故排程執行日期為 2, 4, 6, 9, 11, 13, 23, 25, 27, 30 號，其中 13 當天此 Job 只會執行一次

(2)



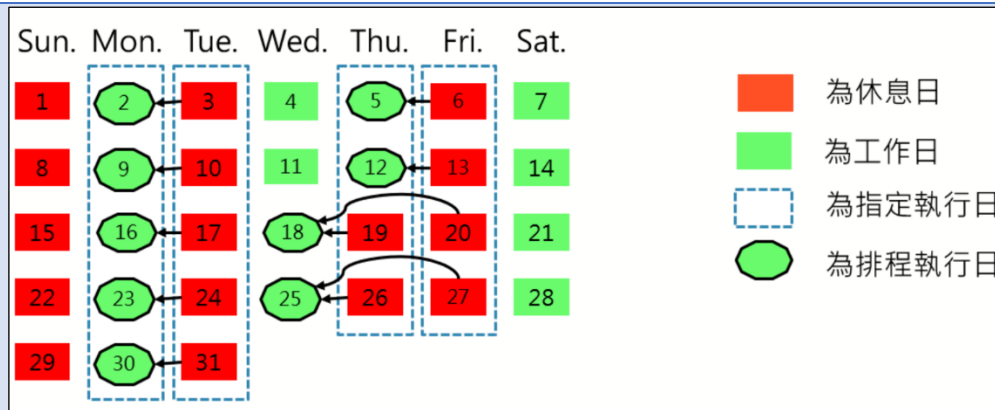
依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 18、20，休息日往前推一天工作日分別是 17、19 日，故排程執行日期為 2, 4, 6, 9, 11, 13, 16, 17, 19, 23, 25, 27, 30 號，其中 17、19 雖不是依據 Date Rule 所挑選的指定執行日，但會執行。

(3)



依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 2、9、16、23、30，往前推一天分別是 31(上個月)、7、14、21、28，故排程執行日期為 31(上個月), 4, 6, 7, 11, 13, 14, 18, 20, 21, 25, 27, 28，其中 31(上個月)、7、14、21、28 雖不是依據 Date Rule 所挑選的指定執行日，但會執行。

(4)

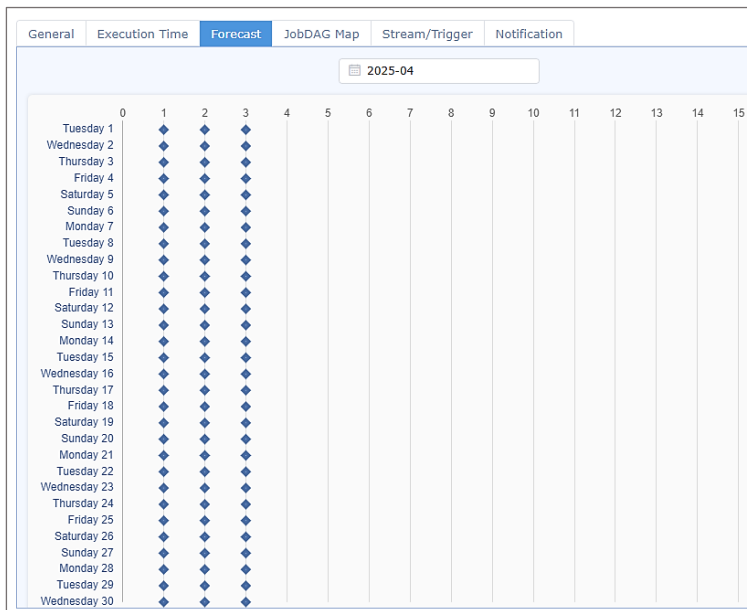


依據 Date Rule 所挑選的指定執行日 (藍色虛線) 範圍當中包含休息日 3、6、10、13、17、19、20、24、26、27、31) · 往前推一天分別是 2、5、9、12、16、18、23、25、30 · 其中 19、20 都指到 18 · 26、27 都指到 25 · 故排程執行日期為 2, 5, 9, 12, 16, 18, 23, 25, 30 號 · 其中 18, 25 號皆只會執行一次。

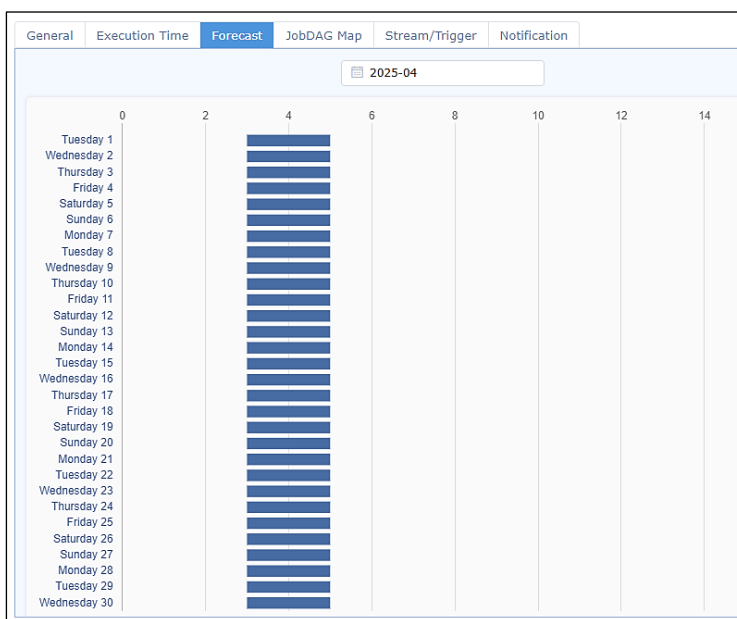
## Job Forecast 頁面

在作業的設定頁面中提供了 Forecast 功能，可以用來預覽此作業套用排程條件之後的預計排程狀況，方便事先檢核作業的排程條件是否設定正確。

在切換到「Forecast」頁面後會以當前此作業的排程條件來顯示當前這個月份中的每一天排程狀況。使用者可以往前或往後以每一個月份為區間來查看各月的排程狀況。例如下面的範例畫面就是以前面示範的 Multiple Time Job 類型來查看 2025 年 4 月的排程狀況。每個符合的排程時間會以藍色線條表示作業任務的預計啟動時間點。

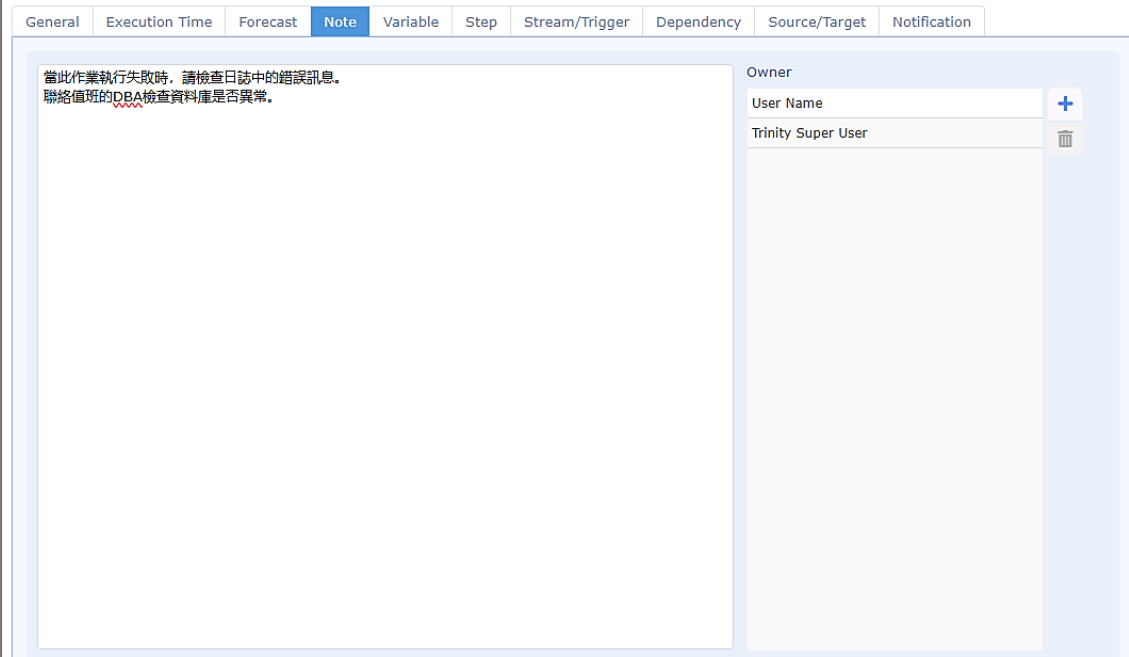


而下面的範例畫面就是以前面示範的 Continuous Job 類型來查看 2022 年 4 月的排程狀況。注意 Continuous Job 的排程預測會以各開始/結束的時間區間畫出長條圖來顯示。



## Job Note 功能

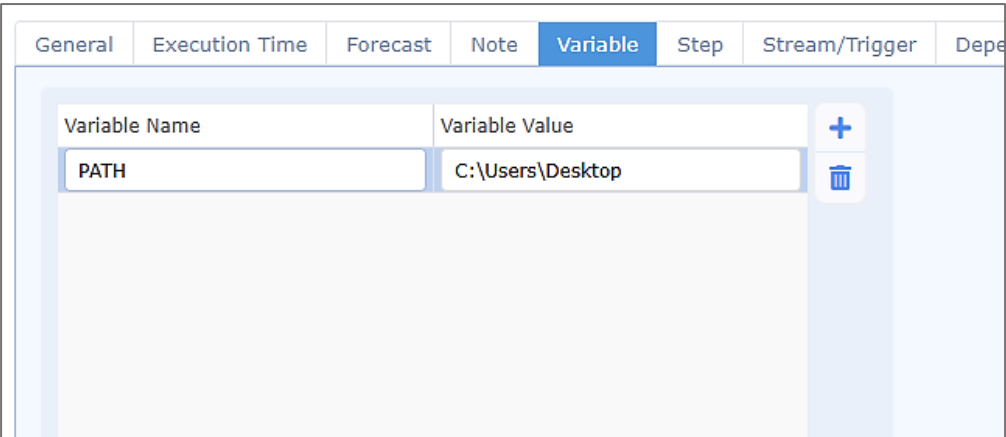
在作業的設定頁面中提供了 Note 註記功能，可以用來定義標準的作業程序(SOP)說明，例如當作業執行失敗時需要執行的人工檢核程序。以及此作業的負責人是誰。例如下面的範例畫面。



The screenshot shows the 'Note' configuration page. The 'Note' tab is active, displaying a text area with the following content: 當此作業執行失敗時，請檢查日誌中的錯誤訊息。聯絡值班的DBA檢查資料庫是否異常。 To the right, the 'Owner' section shows a list with 'Trinity Super User' and a trash icon.

## Job Variable 功能

可編輯單一 Job 專用變數。若此處變數與任何變數同名，則一律以此處設定的變數值為優先使用。



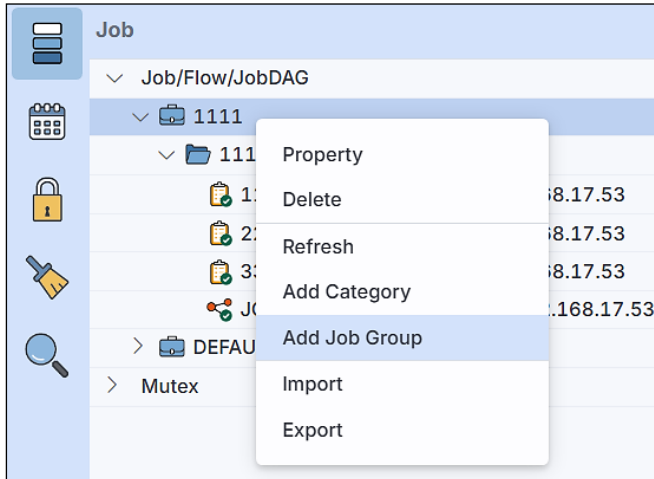
The screenshot shows the 'Variable' configuration page. The 'Variable' tab is active, displaying a table with the following content:

Variable Name	Variable Value	
PATH	C:\Users\Desktop	+ -

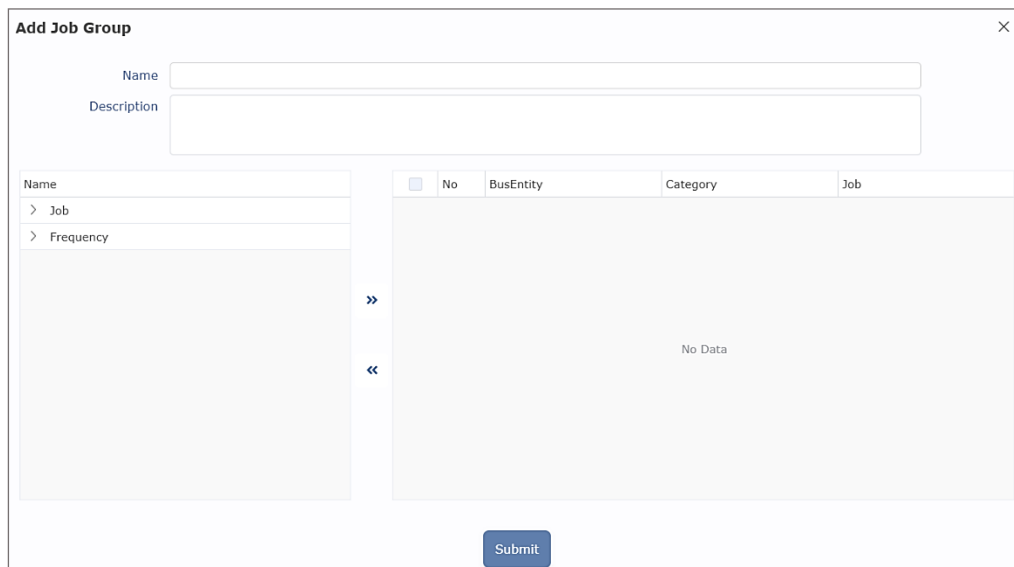
## Job Group

Job Group 是可以自己選擇 Job 組成 Group，在 Task Console Tree View 中則可以針對 Job Group 執行。作用在於，針對有需要的作業同時執行(如已失敗的作業)，為了提升方便性而新增此功能。

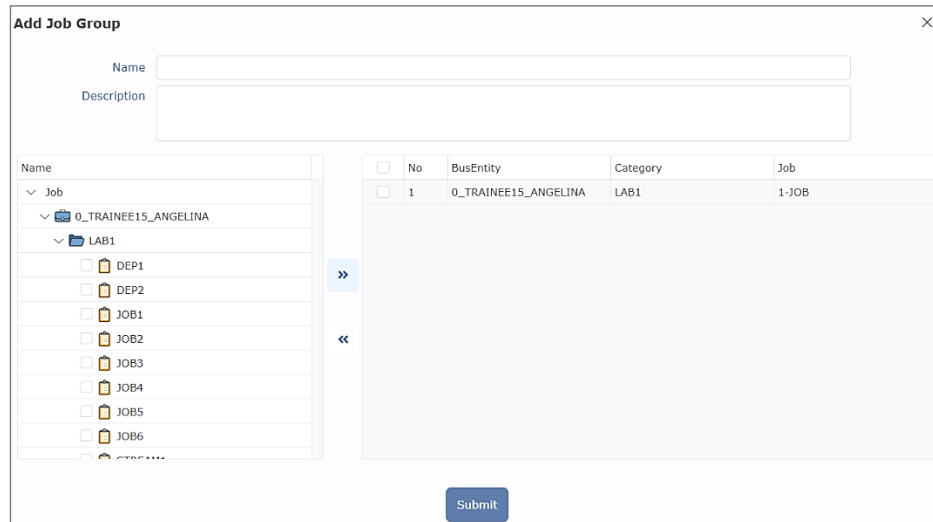
於樹狀視圖中選擇 Business Entity 物件項目，點選右鍵從 Popup Menu 中選取「Add JobGroup」



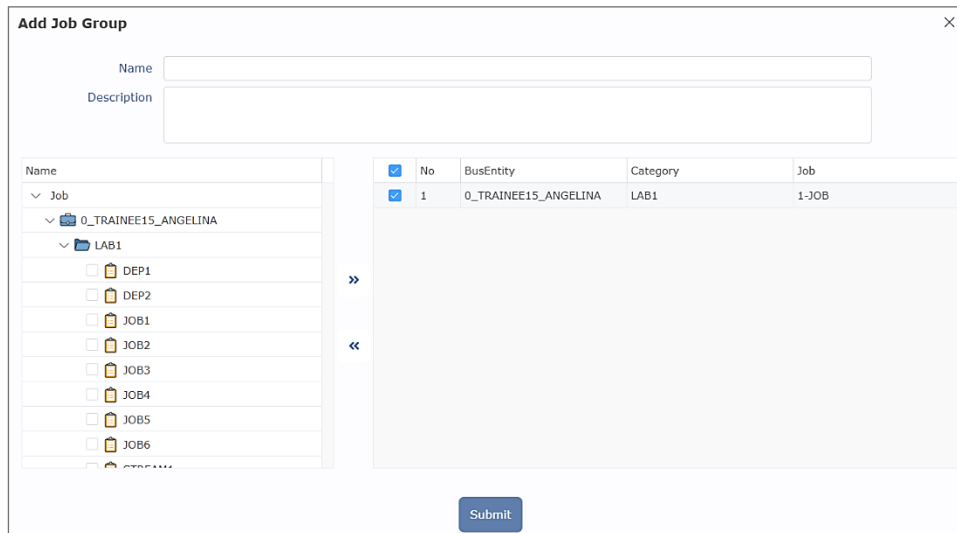
之後會出現一個對話視窗：



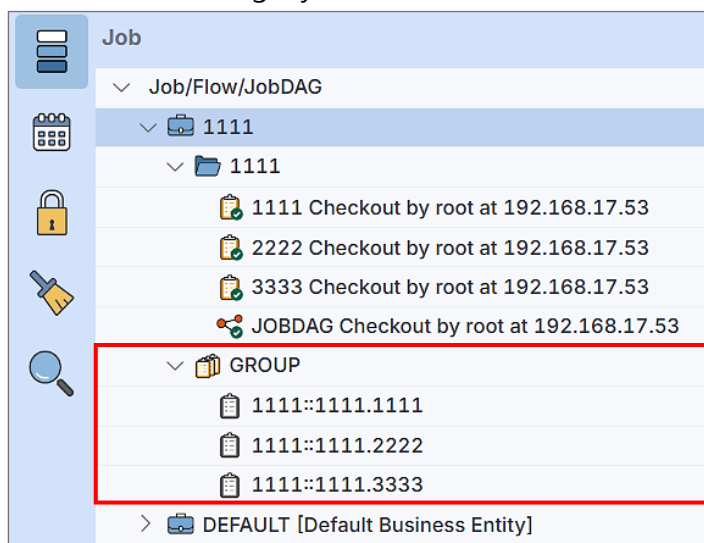
- 新增 Job：於左方勾選 Job，再點選中間  即可新增。



- 刪除 Job：於右方欄位中，勾選最前方 Check Box 後點選「<<」即可刪除。



新增完畢後在 Category 節點上，最後一項則是 Job Group 節點。

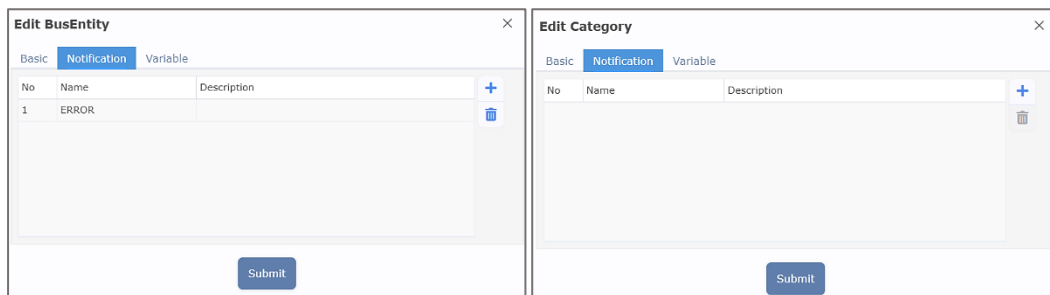


## 訊息通知(Notification)

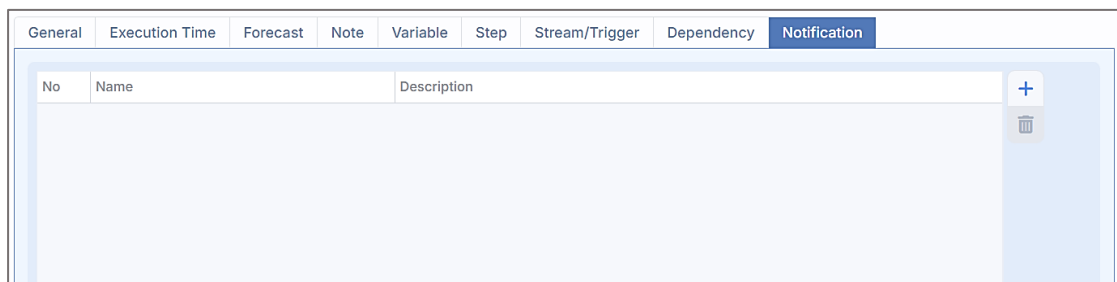
透過訊息通知的設定可以即時掌握各個作業執行的情況，使用者可以決定在作業執行的哪個時間點，採用電子郵件(Email)、行動短訊(SMS)、外部指令執行(External Command)及 HTTP 請求方式(HTTP Action)來通知群組或是使用者作業執行的情形。

### 新增 Notification 通知

選擇欲新增 Notification 的 Business Entity 或 Category 項目，滑鼠左鍵點擊兩下開啟編輯視窗，選擇 Notification 分頁點擊加號新增。



選擇 Job 編輯頁籤中的 Notification 分頁點擊加號也可開啟設定畫面。



採用訊息通知設定的優先序為 Job 層級 > Category 層級 > Entity 層級，若三個層級都沒有設定，則不會進行訊息通知。

**Notification**

Name

Description

Timing

**Email** SMS External Command HTTP Action

Activate

NTLM Domain  Auth Mechanisms

Message Subject  Attach Output Log

Message Content

Notification List

No	Activate	Name
No Data		

- Name：訊息通知名稱。
- Description：訊息通知描述說明。
- Timing：訊息通知發送時間點。
  - When Task Begin：當任務開始時。
  - When Task Completed：當任務結束時。
  - When Task Success：當任務成功時。
  - When Task Error：當任務失敗時。
  - When Task Timeout：當任務執行時超過接收檔案的時間。
  - When Task Overtime：當任務執行時超過設定時間，須配合 Step 進行設定，Step 設定位置如下圖：

Job(1-JOB) Step(TEST X)

BusEntity: 0\_TRAINEE15\_ANGELINA Category: LAB1

Step Name: TEST Step Type: DATA MANAGEMENT  Activate

Description:

Success Rule: Equal To Success Value 1: 0

General Design Debug **Statistic** Postcondition

Overtime Rule: Compare With Max Seconds

Difference Base: Seconds

Difference Value: 0

Min Seconds: 300

Max Seconds: 300

Avg Seconds: 300

Collect Statistic

( 詳細說明請見 Job Step 章節說明 )

- When Task Waiting Long : 當任務等待時間超過設定時間，須配合 Job 進行設定，Job 設定位置如下圖：

General Execution Time Forecast Note Variable Step Stream/Trigger Dependency Source/Target Notification

Domain: Default Agent: LOCAL\_AGENT

Frequency: Select File Source: Select

Retry Option:  Retry After Job Failed Retry Mode:  From Beginning  From Breakpoint  Only File Receive

Retry Interval: 1 min Max Retry Time: 1 time

Time Window: Time Window Begin: 00:00 Time Window End: 23:59

TxDate Setting: TxDate Rule: System Date TxDate Offset: 0 View TxDate

Priority: Priority: 1  Bypass Error  Critical Job Created By: root Use Step Postcondition

**Notification Setting: Waiting Time: 1 min**

Task Attribute:  Skip Missing Task  Only Apply Completed Task  Don't Save History  Rerun From FileSource

Online Date:  Online Date Time  Offline Date Time

Job Alert:  Start In Time 00:00  Finish In Time 00:00 Avg Time: 1

( 詳細說明請見 Job Step 章節說明 )

- When Task Retry : 當任務執行第一次 Retry 。
- When Task Retry Every Time : 任務執行每次 Retry 。
- When Task Has Error Record : 當有錯誤筆數時才會寄信 ( 註：不等同"error 時寄信"，因為系統允許設定容錯筆數，因此若有錯誤筆數但未達容錯上限時還是能夠使任務執行成功、不會 error。雖然執行狀態為成功，但 Error 筆數仍會如實紀錄於系統中；此選項會在有 error 筆數時生效並寄信 )
- No Further WCalendar Date Exist : 當任務所運用的 Working Calendar 已無後續的日期設定時。

- Email 頁面：

- Activate：是否啟用，勾選代表啟用。
- Ntlm Domain：Ntlm Domain。當此欄位有值，則「Auth Mechanisms」欄位為必填。
- Auth Mechanisms：選擇驗證通訊協定，選項有「NTLM」及空選項。當此欄位有值，則「Ntlm Domain」欄位為必填。
- Message Subject：發送訊息主旨。
- Message Content：發送訊息內容。
- Attach Output Log：勾選代表將任務執行 log 檔產製為信件中的附件檔案。
- Notification List：
  - ◆ Group：設定接收通知的群組
  - ◆ Account：設定接收通知的使用者

- SMS 頁面：

**Notification** [X]

Name

Description

Timing

Email **SMS** External Command HTTP Action

Activate

Message Subject

Notification List

Group		User
No	Activate	Name
No Data		

Submit

- Activate：是否啟用，勾選代表啟用。
- Message Subject：發送訊息內容。
- Notification List：
  - ◆ Group：設定接收通知的群組
  - ◆ Account：設定接收通知的使用者

- External Command 頁面：

Notification

Name

Description

Timing

Email SMS **External Command** HTTP Action

Activate

Command String

Submit

- Activate：是否啟用，勾選代表啟用。
- Command String：請輸入程式執行指令。

- HTTP Action 頁面：

- Activate：是否啟用，勾選代表啟用。
- Base URL\*：輸入 HTTP URL。
- HTTP Method\*：選擇以 GET 或 POST 方式。
- Encoding\*：選擇編碼，選項有「UTF-8」、「BIG5」、「GB2312」、「GBK」。
- Content-Type：
  - 選擇 Content-Type，選項有「application/x-www-form-urlencoded」、「application/json」、「text/html」及空選項。
  - 若「HTTP Method」選擇 POST 且「Parameters」有值，則此下拉選單只能不選或是選擇「application/x-www-form-urlencoded」。
  - 若已於下方「Header」區域新增一個名為「Content-Type」的 Key，且該筆 Header 的 Value 值為「charset=指定值」的話，則會以該筆 Header 的內容取代此下拉選單的選項值。
- Parameters：輸入 Parameters。當上方「HTTP Method」選擇 POST 時「Body」為可輸入狀態，須注意此時「Parameters」與「Body」不能同時有值。
- Headers：輸入 Headers。
- Cookies：輸入 Cookies。
- Body：「HTTP Method」選擇 POST 時才可輸入。須注意「Parameters」與「Body」不能同時有值。
- Test Conn.：點擊可測試送出值（會實際送出），並會在 JF Designer 當中以彈出視窗顯示送出結果。

■ 舉例：

檢視回應內容：

```

Tomcat
Process Do Get
----- Paramter -----
[Para0] : [abc]
[1Para] : [efg]
----- Paramter -----

----- Header -----
[cookie] : [Cookies0=C123]
[header0] : [H123]
[1header] : [H456]
[content-type] : [text/html;charset=UTF-8]
[host] : [127.0.0.1:9090]
[connection] : [Keep-Alive]
[user-agent] : [Apache-HttpClient/4.3.6 (java 1.5)]
----- Header -----

----- Cookie -----
[Cookies0] : [C123]
[1Cookies] : [C456]
----- Cookie -----

----- Body -----
----- Body -----
    
```

## 編輯通知

選擇要修改的通知子項目，滑鼠點擊兩下。會出現個別的編輯視窗。在確定完所進行的修改之後，按下「Submit」按鈕會將所做的修改更新至 Trinity 數據庫中。

## 刪除通知

選擇要刪除的通知子項目，點選畫面右方刪除圖示，確認後即可刪除。

## 便利性的變數替換規則

在 Notification 編輯視窗中的 Message Subject 或 Message Content 中，輸入變數標籤，則在您收到郵件時的 Subject 或 Content 之內容，這些變數標籤會被取代為真實的資訊，所有可用變數如下所示：

`${ENTITYNAME}`、`${CATEGORYNAME}`、`${JOBNAME}`、`${FLOWNAME}`、  
`${STEPNAME}`、`${JOBANDFLOWUID}`、`${FREQUENCYUID}`、`${TASKUID}`  
`${TASKDATE}`、`${TASKDATE1}`、`${TASKTIME}`、`${TASKTIME1}`  
`${STARTDATE}`、`${STARTTIME}`、`${ENDDATE}`、`${ENDTIME}`、`${TXDATE}`  
`${JCSSTATUS}`、`${TASKSTATUS}`、`${TASKNAME}`、`${TASKTYPE}`、`${TASKAGENTIP}`  
`${REASON}`、`${DESCRIPTION}`、`${DOWNSTREAM}`、`${RUNNINGCOUNT}`、  
`${RERUNSTEP}`、`${BYPASSERROR}`

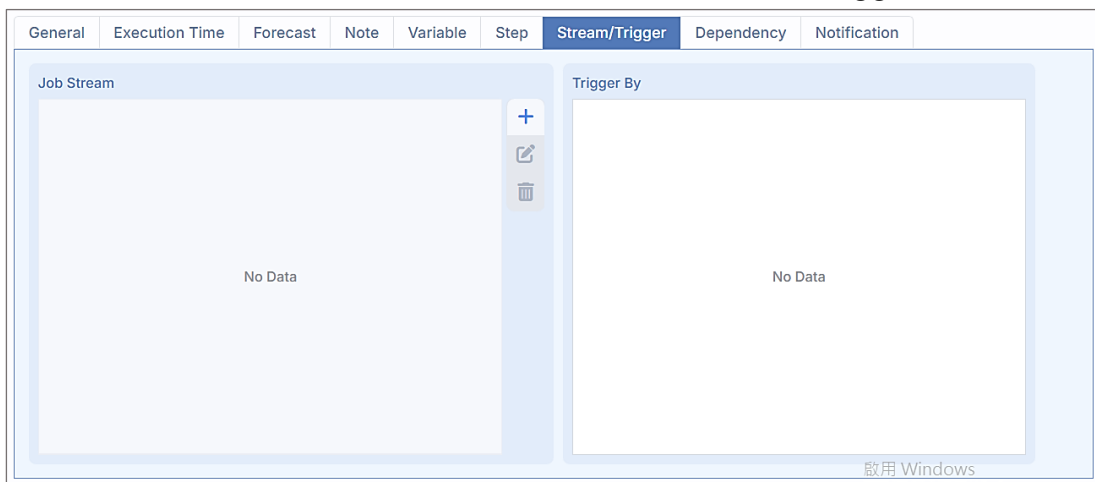
以上所有的變數之標籤，所替換的值都相當於您在 TaskConsole 中所見的各项值。

## Job Stream 設定

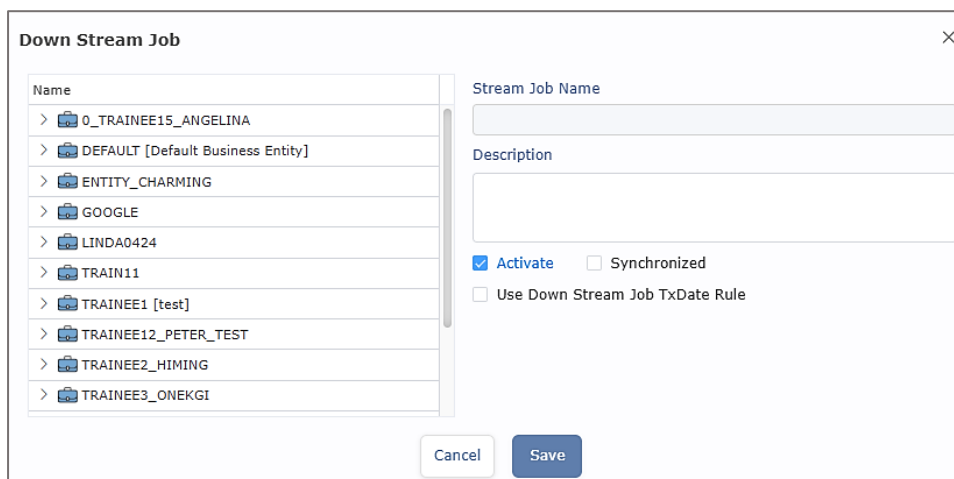
Job Stream 是用來設定作業的下游作業(Downstream Job)。所謂的下游作業是指在作業執行成功時會被自動觸發執行的作業。而一個擁有 Job Stream 的作業就被稱做是這些下游作業的上游作業(Upstream Job)。一個作業可以設定觸發多個下游作業，這些下游作業在被觸發後是同時執行的，作業彼此間並無先後順序。

### 新增 Job Stream

選擇要進行設定的作業項目開啟 Job 編輯畫面，點選「Stream/Trigger」頁面。



點擊加號圖示，開始設定下游作業視窗。



- Stream Job Name：使用者必需選擇一個欲被觸發的下游作業，選擇作業後此欄位中會自動顯示所選擇作業的全路徑名稱，格式是以「<Entity>::<Category>.<Job>」的方式呈現。
- Description：可輸入關於此 Job Stream 的說明。
- Activate：勾選代表此 Job Stream 為有效狀態。為有效狀態才可觸發產生下游作業。
- Use Down Stream Job TxDate Rule：勾選表示下游作業會用 TxDate Rule 來推算 TxDate，做法是以上游作業所執行的 TxDate，來推算下游作業的 TxDate。若未勾

選，就算下游作業有設定 TxDate Rule，還是會沿用上游作業的 TxDate。此功能只在排程執行時有效；若採手動執行，則系統會改以 Task console 的 Run task 介面上的「Virtual System Date」輸入值來推算下游作業的 TxDate。

■ 舉例說明：

建立兩個 Job：JobA 與 JobB

JobA 基本資訊設定當中將 TxDate Rule 設定為 Offset Day 1

The screenshot shows the configuration page for JobA. The 'TxDate Setting' section is highlighted with a red box, showing 'TxDate Rule' set to 'Offset Day' and 'TxDate Offset' set to '1'. Other visible settings include 'BusEntity' (ECHO), 'Name' (JOBA), 'Job Type' (ETL Job), 'Agent' (127.0.0.1), 'Priority' (1), and 'Waiting Time' (1 min).

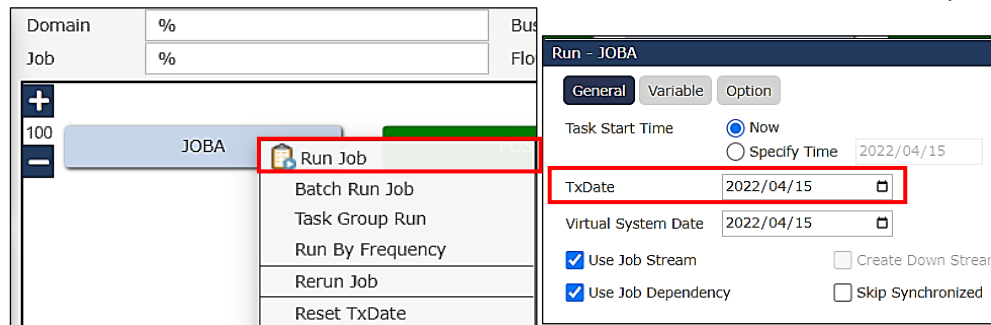
JobB 基本資訊設定當中將 TxDate Rule 設定為 Offset Day 2

The screenshot shows the configuration page for JobB. The 'TxDate Setting' section is highlighted with a red box, showing 'TxDate Rule' set to 'Offset Day' and 'TxDate Offset' set to '2'. Other visible settings include 'BusEntity' (ECHO), 'Name' (JOB B), 'Job Type' (ETL Job), 'Agent' (127.0.0.1), 'Priority' (1), and 'Waiting Time' (1 min).

JobA 設定 Stream 關聯 JobB (JobA 為 Up Stream，JobB 為 Down Stream)，勾選 Use Down Stream Job TxDate Rule

Use Down Stream Job TxDate Rule

並於 TaskConsole 手動執行 JobA 設定 Virtual System Date 為 2022-04-15：



執行時 JobB 會根據介面上所設定的 Virtual System Date (4/15) 為基準減去 JobB 的 TxDate Rule = Offset Day 2，故本例為 2022-04-15 減 2 天，結果以 2022-04-13 為 JobB 的 TxDate 執行作業。

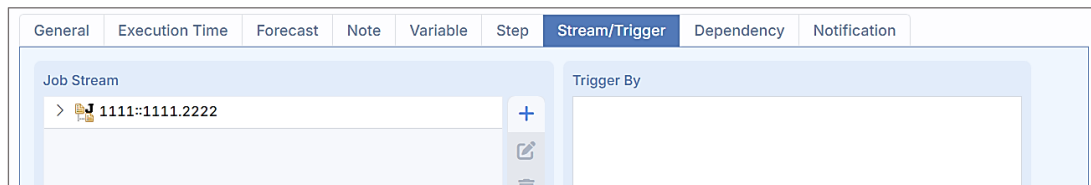
反之，若 JobA 設定 Stream 關聯 JobB ( JobA 為 Up Stream · JobB 為 Down Stream )，未勾選 Use Down Stream Job TxDate Rule

Use Down Stream Job TxDate Rule

則執行時 JobB 不會使用 JobB 基本設定當中的 TxDate Rule 設定，而是直接等同 JobA 的 TxDate 值。

- Synchronized：有勾選時，則上游作業要執行前，會先確認上游作業前一次執行時，其下游作業是否執行成功，若不成功則本次上游作業將不執行。

按下「Save」按鈕會將所輸入的 Job Stream 設定更新至 Trinity 數據庫中，並看到如下圖的畫面



## 修改 Job Stream

於「Stream/Trigger」頁面中選擇要修改的下游作業子項目，按右鍵從 Popup Menu 中選擇「Property」或是滑鼠雙擊兩下。執行之後會出現如前面新增 Job Stream 時的對話視窗，在確定完所進行的修改之後，可以按下「Save」按鈕將所做的修改更新至 Trinity 數據庫中，按下「Cancel」放棄操作。

## 刪除 Job Stream

於「Stream/Trigger」頁面中選擇要刪除的下游作業子項目，按右鍵從 Popup Menu 中選擇「Delete」或點擊下方刪除圖示。請參考基本操作中的關於刪除動作的說明。

## Job Dependency 設定

Job Dependency 是用來設定某個作業的檢核條件，此檢核條件會在準備執行作業之前用來檢查條件是否成立。檢核條件必須成立 Trinity 才會啟動作業的執行。而 Trinity 會定時重複執行檢核條件，直到條件成立便會開始作業的執行。

一項作業可以同時設定多個 Job Dependency，如果有多個 Job Dependency 時，那麼所有的檢核條件都必須成立後才能執行作業，只要有一個檢核條件不成立就無法執行作業。

兩個 Job 允許同時設定對方為 Dependency Job，例如 Job\_A & Job\_B，可設定 Job\_A depends on Job\_B，而 Job\_B 也 depends on Job\_A。

### 新增 Job Dependency

選擇要進行設定的作業項目開啟 Job 編輯畫面，點選「Dependency」頁籤。

BusEntity	Category	Job
No Data		

Dependent

Description

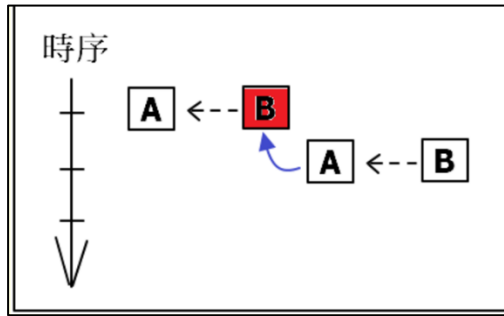
Rule

啟用 Windows

點擊加號圖示，開啟如下圖的編輯視窗設定檢核條件。

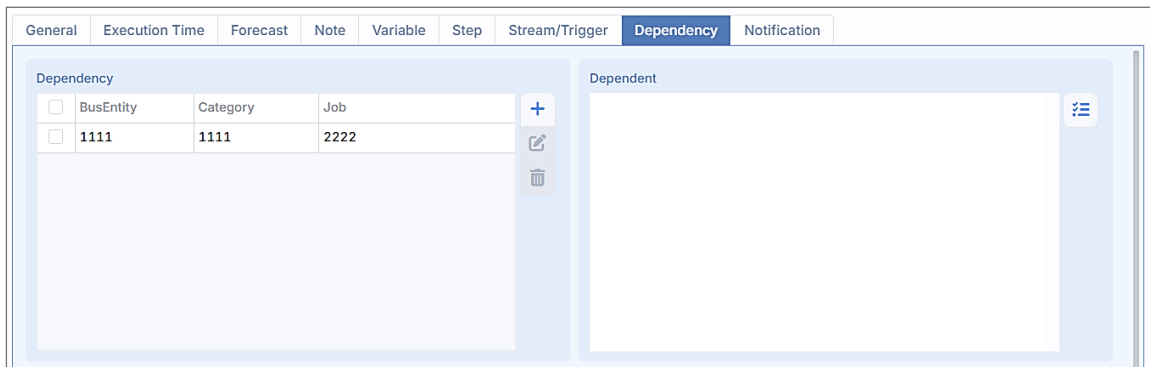
在 Job Dependency 對話視窗中使用者可選擇一個或多個要用來進行檢核的對象作業，所以使用者必須在畫面左方經由 Business Entity -> Category -> Job 的階層來勾選作業，打勾後會將所選的項目移至右方，若想移除只需要將欲移除的項目勾選的部份移除取消即可。點選右方的項目可進行該筆的設定。

- Description：可以輸入關於此 Job Dependency 的說明。
- Rule：用來設定要執行的檢核條件內容，設定多個作業時，每一項作業均須設定檢核條件，有關於檢核條件的用法請參考本文件後續章節的「Dependency Rule 用法說明」。
- Activate：設定此 Job Dependency 是否為有效狀態。此 Activate 選項必需勾選起來才會被用來進行檢核，如果沒有勾選此選項代表此 Job Dependency 關係是暫時失效的狀態而不會用來進行檢核。
- Major：勾選表示被倚賴的作業當天未產出 Task 時，目標作業產出的 Task 就不會產生 Dependency Rule 執行檢核，反之被倚賴的作業當天有產生 Task 時(判斷當下時間之前)，則會產生 Dependency Rule 執行檢核。(僅用於排程產生之工單)
- Synchronized：有勾選時，則 Dependency 當中被倚賴的 Job 在執行前，會先確認是否存在著「Task Date/Time 小於當前被倚賴的 Job 的 Task Date/Time」且執行狀態不為成功的目標 Job。
  - 舉例說明：建立兩個 Job：JobA 與 JobB  
JobB 設定 Dependency 關聯 JobA ( Job B 依賴 Job A，也就是用 Job A <--- JobB 來表示 )  
若 9/11 11:00 JobA 執行完成，則 JobB 12:00 符合可執行條件。  
隔天 9/12 11:00 JobA 執行前會去檢查 9/1112:00(也就是前一回) 所執行的 Job B 是否執行成功，成功才會執行 9/12 11:00 JobA。



- Excluded：有勾選時，則 Dependency 當中被倚賴的 Job 在執行前，會先確認是否存在未完成的目標 Job，若存在，則被倚賴的 Job 會維持等待狀態，直到目標 Job 執行完畢。

確認完成設定後按下「Save」 按鈕會將所輸入的 Job Dependency 設定更新至 Trinity 數據庫中，如下圖的畫面。按下「Cancel」放棄操作。



## Dependency Rule 用法說明

檢核條件函數分成兩類：

- 第一類函數的傳回值為布林值可直接用來判斷條件是否成立。
- 第二類函數的傳回值並不是布林值，而是傳回日期型態。此類函數無法直接用來判斷條件是否成立，而是必須搭配邏輯運算子才能判斷條件的成立與否。

函數說明：

- S() - 當依賴的作業為 Success 狀態時則傳回 True，否則傳回 False。
- LS() - 當依賴的作業最後一筆狀態為 Success 時則傳回 True，否則傳回 False。
- F() - 當依賴的作業為 Error 狀態時則傳回 True，否則傳回 False。
- D() - 當依賴的作業為 Success 或 Error 或 Timeout 狀態時則傳回 True，否則傳回 False。
- R() - 當依賴的作業為 Running 狀態時則傳回 True，否則傳回 False。
- W() - 當依賴的作業為 Waiting 狀態時則傳回 True，否則傳回 False。
- SW() - 當依賴的作業為 Success 或 Waiting 狀態時則傳回 True，否則傳回 False。
- DW() - 當依賴的作業為 Success 或 Error 或 Timeout 或 Waiting 狀態時則傳回 True，否則傳回 False。
- TX() - 用來取得依賴作業的目前資料日期(TXDATE)，當依賴的作業沒有資料日期時則傳回 1900-01-01。
- TD() - 用來取得依賴作業的排程日期(TASKDATE)。
- TT() - 用來取得依賴作業的排程日期(TASKTIME)。格式 HHmm。
- MONBEGIN(\$TXDATE, \$OFFSET) - 用來取得指定日期的月初日期。此函數需要傳進兩個參數，\$TXDATE 為指定要進行運算的基礎日期，\$OFFSET 為要推算月份的差距值，此函數會根據傳進的日期往前推算出某一個月份的月初日期。\$OFFSET 為 0 時會取得當月份的月初日期，\$OFFSET 為 -1 時會取得前一月的月初日期，\$OFFSET 為 -2 時會取得前兩月的月初日期，\$OFFSET 為 1 時會取得下一月的月初日期，\$OFFSET 為 2 時會取得下兩月的月初日期，以此類推。例如輸入 MONBEGIN(2010-5-8,-1) = THIS 則會取得 4 月的月初日期。
- MONEND(\$TXDATE, \$OFFSET) - 用來取得指定日期的月底日期。此函數需要傳進兩個參數，\$TXDATE 為指定要進行運算的基礎日期，\$OFFSET 為要推算月份的差距值，此函數會根據傳進的日期往前推算出某一個月份的月底日期。\$OFFSET 為 0 時會取得當月份的月底日期，\$OFFSET 為 -1 時會取得前一月的月底日期，\$OFFSET 為 -2 時會取得前兩月的月底日期，\$OFFSET 為 1 時會取得下一月的月底日期，\$OFFSET 為 2 時會取得下兩月的月底日期，以此類推。例如輸入 MONEND(2010-6-6,-1) = THIS 則會取得 5 月的月底日期。

- **WEEKBEGIN(\$TXDATE, \$OFFSET)** - 用來取得指定日期的當週第一天日期，當週第一天的定義是週日。此函數需要傳進兩個參數，\$TXDATE 為指定要進行運算的基礎日期，\$OFFSET 為要推算週數的差距值，此函數會根據傳進的日期往前推算出某一週的當週第一天日期。\$OFFSET 為 0 時會取得當週的星期日日期，\$OFFSET 為 -1 時會取得前一週的星期日日期，\$OFFSET 為 -2 時會取得前兩週的星期日日期，\$OFFSET 為 1 時會取得下一週的星期日日期，\$OFFSET 為 2 時會取得下兩週的星期日日期，以此類推。例如輸入 WEEKBEGIN (2010-6-6,-1) = THIS 則會取得 6 月 6 日前一周的星期日之日期。
- **WEEKEND(\$TXDATE, \$OFFSET)** - 用來取得指定日期的當週最後一天日期，當週最後一天的定義是週六。此函數需要傳進兩個參數，\$TXDATE 為指定要進行運算的基礎日期，\$OFFSET 為要推算週數的差距值，此函數會根據傳進的日期往前推算出某一週的當週最後一天日期。\$OFFSET 為 0 時會取得當週的星期六日期，\$OFFSET 為 -1 時會取得前一週的星期六日期，\$OFFSET 為 -2 時會取得前兩週的星期六日期，\$OFFSET 為 1 時會取得下一週的星期六日期，\$OFFSET 為 2 時會取得下兩週的星期六日期，以此類推。例如輸入 WEEKEND (2010-6-6,-1) = THIS 則會取得 6 月 6 日前一周的星期六之日期。
- **DATEDIFF(\$TXDATE, \$OFFSET)** - 用來取得指定日期的前後差距日期。此函數需要傳進兩個參數，\$TXDATE 為指定要進行運算的基礎日期，\$OFFSET 為要推算日期的差距值，\$OFFSET 為 0 時會取得與 \$TXDATE 相同的日期，\$OFFSET 為 -1 時會取得前一天的日期，\$OFFSET 為 1 時會取得後一天的日期，以此類推。例如輸入 DATEDIF (2010-6-6,-1) = THIS 則會取得 6 月 5 日之日期。
- **LW()** - 當依賴的作業最後一筆狀態為 Waiting 時則傳回 True，否則傳回 False。
- **LTX()** - 用來取得依賴作業最後一筆的資料日期(TXDATE)，當依賴的作業沒有資料日期時則傳回 1900-01-01。
- **LTD()** - 用來取得依賴作業最後一筆的排程日期(TASKDATE)，當依賴的作業沒有排程日期時則傳回 1900-01-01。
- **TDT()** - 代表依賴作業任務的 TASKDATE + TASKTIME，實際值內容格式為 yyyyMMddHHmm。
- **TXDT()** - 代表依賴作業任務的 TXDATE + TASKTIME，實際值內容格式為 yyyyMMddHHmm。
- **LTDT()** - 代表依賴作業最後一筆的 TASKDATE + TASKTIME，實際值內容格式為 yyyyMMddHHmm。
- **LTXDT()** - 代表依賴作業最後一筆的 TXDATE + TASKTIME，實際值內容格式為 yyyyMMddHHmm。

**THIS 關鍵字與邏輯運算子的函數介紹：**

- THIS 關鍵字：在作業相關條件的條件式中，THIS 是一個特殊的關鍵字，用來表示是要取得目前作業的任務中所指定的資料日期(TxDate)。
- THISTD：用來取得本身作業的排程日期(TASKDATE)。
- THISTT：用來取得本身作業的排程時間(TASKTIME)。
- THISTXDT：代表本身作業任務的 TXDATE + TASKTIME，實際值內容格式為 yyyyMMddHHmm。
- THISTDT：代表本身作業任務的 TASKDATE + TASKTIME，實際值內容格式為 yyyyMMddHHmm。
- 邏輯運算子：可以使用運算子來組合多個函數與條件或判斷條件是否成立。
- AND：進行 AND 運算，必須前後的條件都成立時，此運算子才會傳回 True，否則傳回 False。
- OR：進行 OR 運算，只要前後的任一條件成立時，此運算子就會傳回 True，必須在前後的條件都不成立時才傳回 False。
- =：用來對兩個日期型態參數進行等於比較，在左方的日期等於右方日期時，此運算子才會傳回 True，否則傳回 False。
- >：用來對兩個日期型態參數進行大於比較，在左方的日期大於右方日期時，此運算子才會傳回 True，否則傳回 False。
- <：用來對兩個日期型態參數進行小於比較，在左方的日期小於右方日期時，此運算子才會傳回 True，否則傳回 False。
- >=：用來對兩個日期型態參數進行大於比較，在左方的日期大於或等於右方日期時，此運算子才會傳回 True，否則傳回 False。
- <=：用來對兩個日期型態參數進行小於比較，在左方的日期小於或等於右方日期時，此運算子才會傳回 True，否則傳回 False。

### 應用範例

S()：

必須是所依賴作業的目前狀態為成功時，此條件的判斷結果才為成立。

TX() = THIS：

必須是所依賴作業的資料日期等於目前作業任務的資料日期時，此條件的判斷結果才為成立。

S() AND TX() = THIS：

必須是所依賴作業的目前狀態為成功同時其資料日期等於目前作業工作的資料日期時，此條件的判斷結果才為成立。

S() AND TX() = MONEND(THIS, -1)：

必須是所依賴作業的目前狀態為成功同時其資料日期等於目前作業工作的資料日期的前一個月的月底日時，此條件的判斷結果才為成立。

範例：假設此今日日期為 2010-6-6，藉由此函數的輸入則含意為 dependency rule 指定的 job 必須執行成功 並且 資料日期為 5 月 31 日時，才可執行本身的 job。

S() AND (TX())>=WEEKBEGIN(THIS,-1) AND TX()<=WEEKEND(THIS,-1)：

必須是所依賴作業的目前狀態為成功同時其資料日期等於目前作業工作的資料日期的前一週的範圍之內，此條件的判斷結果才為成立。

範例：假設今日日期為 2022-7-22，藉由此函數的輸入則含意為 dependency rule 指定的 job 必須執行成功 並且 資料日期為 7 月 11 日~7 月 17 日時，才可執行本身的 job。



## 應用情境

### 執行情境一：

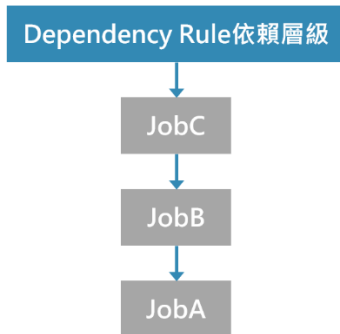
假設今日為 2022/7/7，則有一個 JobB 的執行與否 需要仰賴 JobA 在上一個月的月底被分派執行時是否為成功，則可於 Dependency Rule 描述如下：

Name	JOBA
Rule	S() AND TX() = MONEND(THIS,-1)
Description	0/255
<input checked="" type="checkbox"/> Activate <input type="checkbox"/> Major <input type="checkbox"/> Synchronized <input type="checkbox"/> Excluded	

### 執行情境二：

假設某百貨公司在 2022/08/30 因要結算年初至今日重大節日的銷售額度之資料移轉時，

進行 JobC 的分派工作，而 JobC 的執行條件為「JobB 在父親節當天執行資料移轉須成功」，而 JobB 的執行條件為「JobA 在母親節當天執行資料移轉須成功」



JobB 的 Dependency Rule 設定如下 (指定至母親節)

Name	JOBA
Rule	S() AND TX() = <u>WEEKBEGIN</u> ( 2022-5-31 , 0)
Description	0/255
<input checked="" type="checkbox"/> Activate <input type="checkbox"/> Major <input type="checkbox"/> Synchronized <input type="checkbox"/> Excluded	

JobC 的 Dependency Rule 設定如下 (指定至父親節)

Name	JOBB
Rule	S() AND TX() = <u>DATEDIFF</u> ( THIS , -22)
Description	0/255
<input checked="" type="checkbox"/> Activate <input type="checkbox"/> Major <input type="checkbox"/> Synchronized <input type="checkbox"/> Excluded	

### 修改 Job Dependency

於「Dependency」頁面中選擇要修改的檢核作業子項目滑鼠雙擊兩下，開啟編輯視窗。在確定完所進行的修改之後，可以按下「Save」按鈕會將所做的修改更新至 Trinity 數據庫中，按下「Cancel」放棄操作。

### 刪除 Job Dependency

於「Dependency」頁面中勾選要刪除的檢核作業子項目，點擊下方刪除圖示。請參考基本操作中的關於刪除動作的說明。

## Dependent Job 與 Trigger By

### Dependent Job 項目

若在其他 Job 中有設定 Dependency 於此 Job，則此 Job 中的「Dependency」頁籤 Dependent 區塊則會顯示是哪一個 Job Dependency 於此。

範例：

Job (每日 B) 設定 Dependency 到 Job (工作日 A) 上

BusEntity: 1111, Category: 1111, Name: 每日B, Job Type: ETL Job, Activate:

BusEntity	Category	Job
<input type="checkbox"/> 1111	1111	工作日A

則在 Job (工作日 A) 中的 Dependent 項目內就可以看到是哪一個 Job Dependency 於 Job (工作日 A) 上。

BusEntity: 1111, Category: 1111, Name: 工作日A, Job Type: ETL Job, Activate:

BusEntity	Category	Job
<input type="checkbox"/>		

Dependent: 1111::1111.每日B

於 Dependent 項目內作業按下滑鼠右鍵會顯示「Change Activate」，按下後可針對已存在的 dependency 關係，執行勾選來啟用 dependency 關係，或取消勾選來停用 dependency 關係。

Dependent: 1111::1111.每日B

Change Activate

## Trigger By 項目

若在其他 Job 中有設定 Job Stream 於此 Job，則此 Job 中的「Stream/Trigger」頁面，「Trigger By」區塊會顯示是哪一個 Job Stream 於此。

範例：

Job (A) 設定 Job Stream 到 Job (B) 上

The screenshot shows the configuration page for Job (A) in the 'Stream/Trigger' tab. The top section contains fields for 'BusEntity' (1111), 'Category' (1111), 'Name' (A), 'Job Type' (ETL Job), and an 'Activate' checkbox. Below this is a 'Description' field. The main area is divided into two panels: 'Job Stream' on the left, which contains a list with one item '1111:1111.B', and 'Trigger By' on the right, which is currently empty. The 'Job Stream' panel has a vertical toolbar with a plus sign, a pencil icon, and a trash icon.

則在 Job (B) 中的 Trigger By 項目內就可以看到是哪一個 Job。

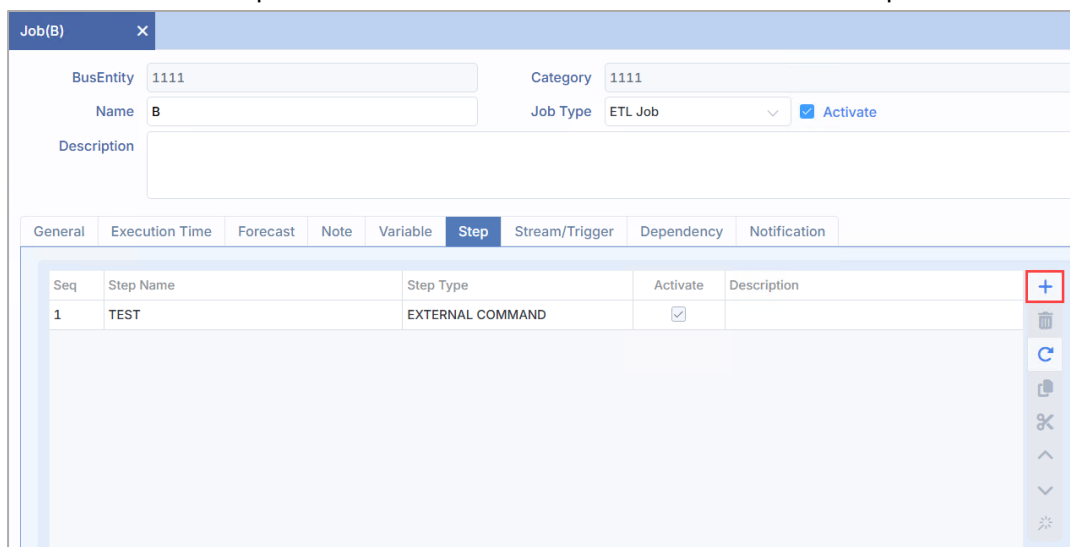
The screenshot shows the configuration page for Job (B) in the 'Stream/Trigger' tab. The top section contains tabs for 'General', 'Execution Time', 'Forecast', 'Note', 'Variable', 'Step', 'Stream/Trigger', 'Dependency', and 'Notification'. The main area is divided into two panels: 'Job Stream' on the left, which is currently empty, and 'Trigger By' on the right, which contains a list with one item '1111:1111.A'. The 'Job Stream' panel has a vertical toolbar with a plus sign, a pencil icon, and a trash icon.

## 作業步驟(Job Step)設定

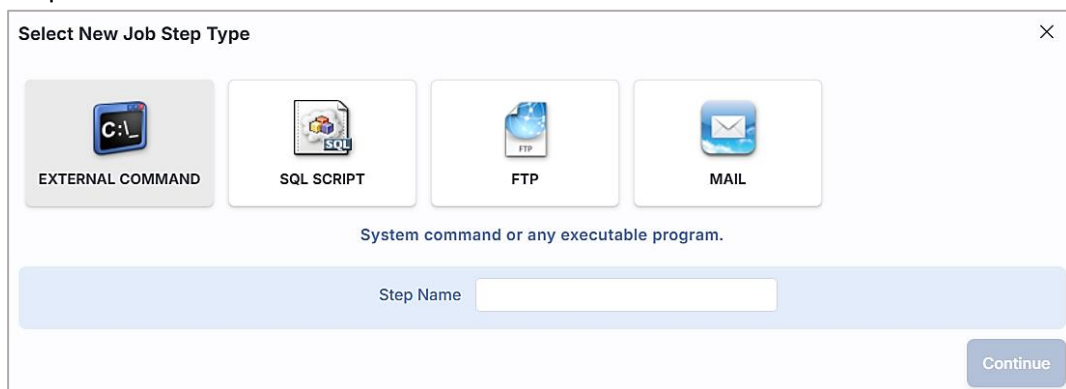
Job Step 必須存在某一個 Job 項目之下，Job Step 才是真正設定執行動作的地方，在 Trinity 中的 Job Step 一共有十種不同的類型，分別是 External Command、Data Management、Teradata Script、SQL Script、FTP、Mail、Big Data、Recovery DM、SQL Report 及 Rest Client。(Big Data Step 操作說明請參考 Trinity V5.0 DM Hadoop Adaptors User Guide)。

### 新增 Job Step

選擇要進行 Job step 設定的作業項目開啟 Job 編輯畫面，點選「Step」頁籤。



點選畫面中右側加號「Add Step」的圖示，會出現下圖視窗可供使用者選擇要新增的 Job Step 類型。



選擇作業步驟的類型並且輸入 Step Name 後，會顯示一頁新增作業步驟的工作頁面。而所顯示的工作頁面內容會隨著步驟類型的不同而出現不同的欄位。各種作業步驟類型的使用介紹請見後續說明。

下圖是選擇 External Command 類型之後所顯示的工作頁面：

Job(JOB1) \*Step(Add E X)

BusEntity TEST Category LAB1 Job JOB1

Step Name EX Step Type EXTERNAL COMMAND  Activate

Description

Success Rule Equal To Success Value 1 - 0 + Success Value 2 - 0 +

General Script Statistic Postcondition

Command

Argument

Encoding

Source Connection Select ...

Target Connection Select ...

Resource Control Select Usage Count 1

Resource Control-1 Select Usage Count 1

Resource Control-2 Select Usage Count 1

Resource Control-3 Select Usage Count 1

Don't Apply File Argument

下圖是選擇 SQL Script 類型之後所顯示的工作頁面：

Job(JOB1) \*Step(Add SQ X)

BusEntity TEST Category LAB1 Job JOB1

Step Name SQL Step Type SQL SCRIPT  Activate

Description

Success Rule Equal To Success Value 1 - 0 + Success Value 2 - 0 +

General Script Statistic Postcondition

Script Type SQL-DB2

Script Path

Script Name

Argument

Encoding

DataBase Select ...

Connection

Resource Control Select Usage Count 1

下圖是選擇 FTP 類型之後所顯示的工作頁面：

The screenshot shows the configuration page for a Job Step of type FTP. The page is titled "Job(JOB1) \*Step(Add FT X)". The "BusEntity" is set to "TEST", "Category" to "LAB1", and "Job" to "JOB1". The "Step Name" is "FTP" and "Step Type" is "FTP" with the "Activate" checkbox checked. The "Description" field is empty. The "Success Rule" is set to "Equal To", with "Success Value 1" and "Success Value 2" both set to "0". Below the main form, there are tabs for "General", "File", "Statistic", and "Postcondition". The "General" tab is active, showing "Target Connection" as "Select", "Error Handle" as "Break", "Protocol" as "SFTP" (unchecked), and "Option" with "Binary" checked, "Passive" unchecked, and "Check Size After Post" unchecked.

下圖是選擇 Mail 類型之後所顯示的工作頁面：

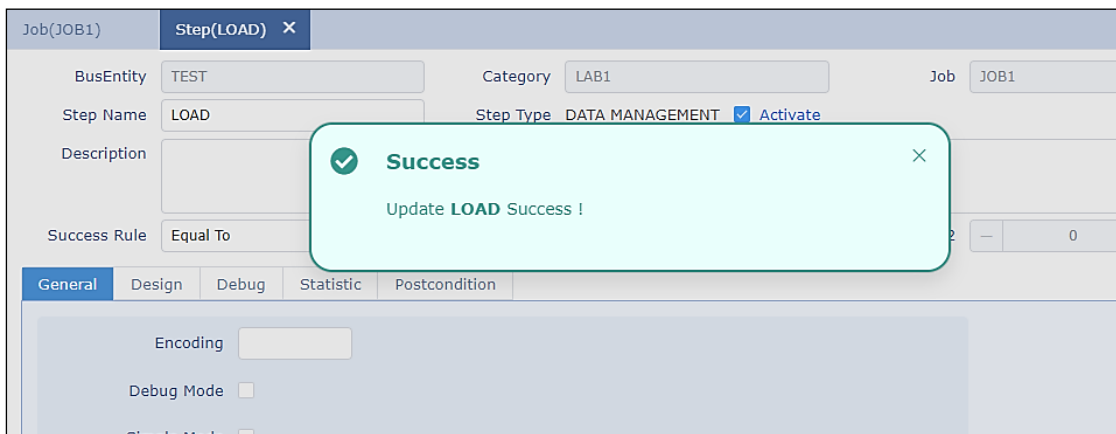
The screenshot shows the configuration page for a Job Step of type Mail. The page is titled "Job(JOB1) \*Step(Add MAI X)". The "BusEntity" is set to "TEST", "Category" to "LAB1", and "Job" to "JOB1". The "Step Name" is "MAIL" and "Step Type" is "MAIL" with the "Activate" checkbox checked. The "Description" field is empty. The "Success Rule" is set to "Equal To", with "Success Value 1" and "Success Value 2" both set to "0". Below the main form, there are tabs for "General", "File", "Statistic", and "Postcondition". The "General" tab is active, showing "Mail Server" as "Select", "Subject" as an empty text field, "Content" with "Delete File" selected (checked) and "Input Content File Path" unselected (unchecked), "Input Content" as an empty text field, and "Mail From" as an empty text field.

不同類型的作業步驟所需設定的項目不同，我們會在後面的章節中各自詳細說明不同類型的欄位項目所代表的意義。

作業的設定值輸入完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。



成功之後使用者會在剛才新增步驟的作業子項目的「Job Step」下看到成功加入的作業步驟，如下圖的示範畫面：



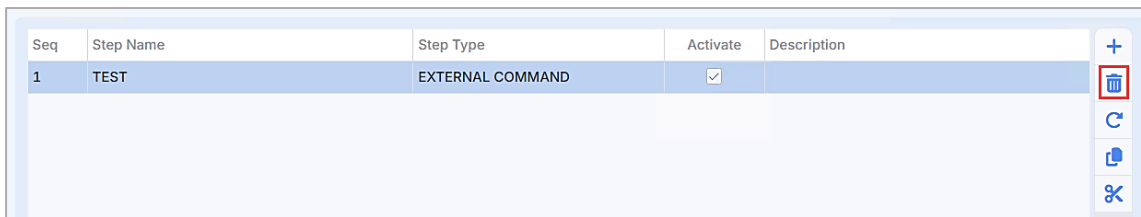
## 修改 Job Step

要修改一項 Job Step 時，於 Job 編輯畫面，點選「Step」頁籤，選擇要進行修改的 Job Step 物件子項目滑鼠點擊兩下。執行後會出現類似先前新增 Job Step 的工作頁面，可進行修改。修改完畢後，請按存檔圖示或使用快捷鍵「Ctrl+S」來執行更新數據庫的動作。

要注意的是使用者必須對要進行修改的 Job，或 Category 項目或其所在的 Business Entity 項目擁有 Edit 的權限才能執行此修改 Job Step 的動作。

## 刪除 Job Step

要刪除一項 Job Step 時，於 Job 編輯畫面，點選「Step」頁籤，選擇要進行刪除的 Job Step 物件子項目，點擊右側「Delete Step」圖示，確認後即可刪除。




Seq	Step Name	Step Type	Activate	Description
1	TEST	EXTERNAL COMMAND	<input checked="" type="checkbox"/>	

要注意的是使用者必須對要進行刪除的 Job，或 Category 項目或其所在的 Business Entity 項目擁有 Delete 的權限才能執行此刪除 Job Step 的動作。

## 複製 Job Step

使用者可以透過複製 Job Step 的方式來節省設定作業步驟的時間，透過複製 Job Step 可快速的進行 Job Step 的設定。要複製一項 Job Step 時，於「Step」頁籤中點選要進行複製的 Job Step 物件子項目，點擊右側「Duplicate」圖示來進行複製。

Seq	Step Name	Step Type	Activate	Description	
1	TEST	EXTERNAL COMMAND	<input checked="" type="checkbox"/>		

執行後會出現一個對話視窗詢問使用者要進行複製的選項，例如下圖中所顯示的範例：

**Duplicate Step** ×

Step Name

Step Description

To Other Job

BusEntity

Category


Job

此對話視窗中的「To Other Job」選項是用來指定是否要將作業複製到其他的 Job 中。如果勾選了此選項則需進一步的選擇要搬移過去的 Business Entity, Category 與 Job。選擇完畢後可以按下「Submit」按鈕來執行複製的動作。

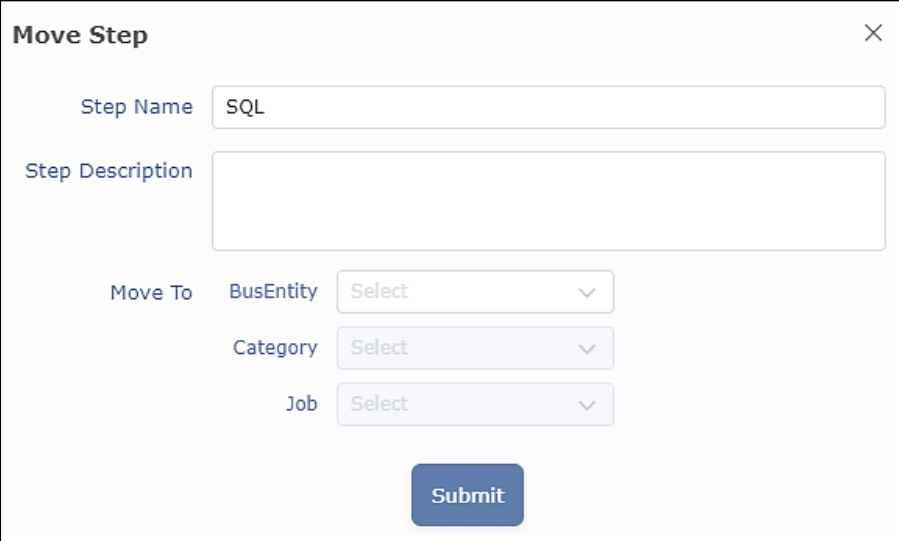
要注意的是使用者必須對要複製過去的 Job 項目或 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此複製 Job Step 的動作。

## 移動 Job Step

已設定好的作業步驟可以在不同的 Job 間進行移動以便進行分類管理。要移動一項 Job Step 時，於「Step」頁籤中點選要移動的 Job Step 物件子項目，點擊右側「Move」圖示。

Seq	Step Name	Step Type	Activate	Description	
1	TEST	EXTERNAL COMMAND	<input checked="" type="checkbox"/>		

執行後會出現一個對話視窗詢問使用者要進行移動的選項，例如下圖中所顯示的範例：



**Move Step** [X]

Step Name: SQL

Step Description: [Empty text area]

Move To: BusEntity [Select] [v]  
Category [Select] [v]  
Job [Select] [v]

[Submit]

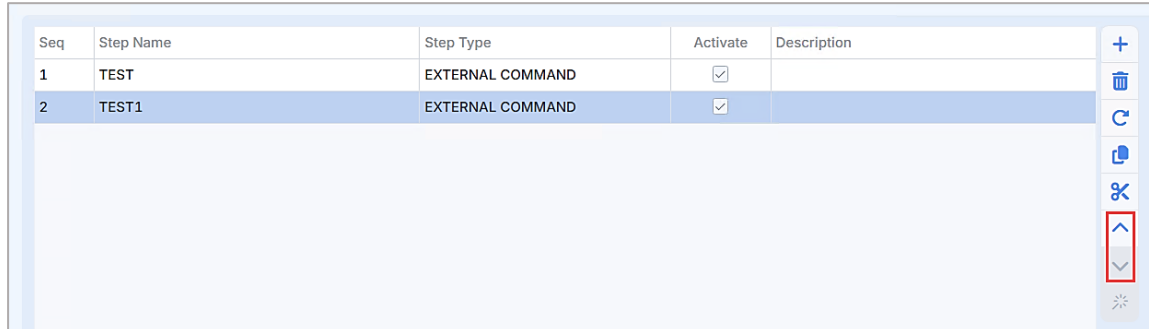
選擇要搬移過去的 Business Entity, Category 與 Job 後可以按下「Submit」按鈕來執行移動 Job Step 的動作。

要注意的是使用者必須對要搬移過去的 Job 項目或 Category 項目或其所在的 Business Entity 項目擁有 Add 的權限才能執行此搬移 Job Step 的動作。

## 調整 Job Step 的執行順序

當使用者想要調整作業底下多個步驟的執行順序時，於「Step」頁籤中點選要移動的 Job Step 物件子項目，點擊右側「Step Sequence Up」、「Step Sequence Down」圖示，即可透過上下箭頭的按鈕來調整步驟執行的順序。

Seq	Step Name	Step Type	Activate	Description
1	TEST	EXTERNAL COMMAND	<input checked="" type="checkbox"/>	
2	TEST1	EXTERNAL COMMAND	<input checked="" type="checkbox"/>	

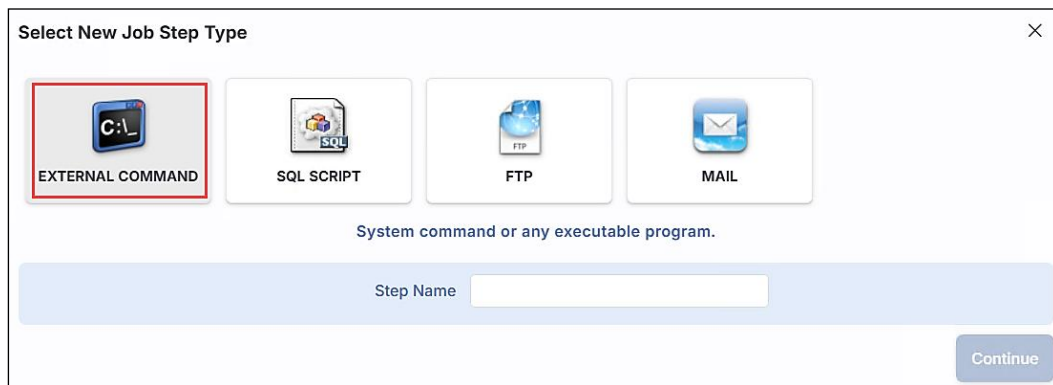


# 作業步驟(Job Step)類型介紹

以下介紹各種作業步驟類型。

## Job Step - External Command

External Command 類型的 Job Step 是用來執行一項外部命令的工作，所謂外部命令就是任何使用者可以在操作系統的命令列下以指令方式呼叫執行的程序。這些程序包含了可執行檔或特定的腳本文件。



## General 頁面說明

The screenshot shows the "General" tab of the Job Step configuration window. The window title is "Job(JOB1) \*Step(Add EX)". The "BusEntity" is "TEST", "Category" is "LAB1", and "Job" is "JOB1". The "Step Name" is "EXT", "Step Type" is "EXTERNAL COMMAND", and "Activate" is checked. The "Description" field is empty. The "Success Rule" is "Equal To", "Success Value 1" is "0", and "Success Value 2" is "0". The "General" tab is active, showing fields for "Command", "Argument", "Encoding", "Source Connection", "Target Connection", "Resource Control", "Resource Control-1", "Resource Control-2", "Resource Control-3", "Usage Count", "Don't Apply File Argument", "Run As", and "Load User's Profile".

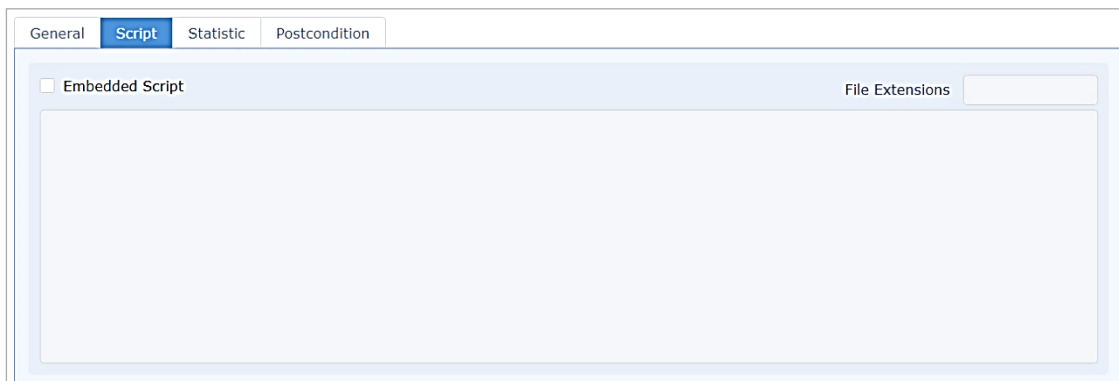
- Step Name：作業步驟的名稱。
- Activate：此作業步驟是否為有效狀態，作業狀態必須是有效狀態才會被執行
- Description：此作業步驟的描述

- Success Rule：判斷執行成功條件，有兩種選項：
  - Equal To：此步驟完成後的回傳值等於 Success Value 1 欄位輸入值時判斷執行成功
  - In Between：此步驟完成後的回傳值介於 Success Value 1 欄位輸入值與 Success Value 2 欄位輸入值判斷執行成功
- Command：設定執行外部程式的指令
  - 可直接輸入指令：例如「echo Trinity」；若使用 perl 則輸入「perl」即可(須配合 Script 頁面內容使用)，程式預設在指令最後自動加入實際檔案路徑，如果要自訂 command 中實際檔案路徑的指令位置(例如指令最後還有要輸入其他參數，所以實際檔案路徑並不是要在指令最後，因此想要自訂位置)，則輸入「perl \${EMBEDDED\_SCRIPT} -xxx」(-xxx 處請輸入參數)，程式會自動把 \${EMBEDDED\_SCRIPT}字串的位置替換成檔案路徑。
  - 可配合 Script 頁面使用：若要執行 Script 頁面所輸入的指令，Agent 為 Linux 作業系統時此欄位需輸入「sh \${EMBEDDED\_SCRIPT}」或「sh」，Agent 為 Windows 作業系統時此欄位需輸入「\${EMBEDDED\_SCRIPT}」且 Script 頁面當中右上角 File Extensions 需輸入附檔名，可參考本文下一段 Script 頁面介紹。
- Argument：執行外部指令時所需的參數
- Encoding：設定預計執行外部程式的編碼
- Source Connection：設定資料來源連結，透過變數(詳請見本文件「附錄 A. 工作變數的使用」介紹)取得資料來源連結中的連線資訊。(資料來源連結的種類須為 Database Type)
- Target Connection：設定資料目的連結，透過變數(詳請見本文件「附錄 A. 工作變數的使用」介紹)取得資料目的連結中的連線資訊。(資料來源連結的種類須為 Database Type)
- Resource Control、Resource Control - 1 ~ 3：設定資源使用項目，共可設定不同的四組。此項目的作用是為了避免同時連線至同一資料庫進行存取，因連線數過多造成 DB 執行效能降低，故透過資源項目及使用數目的設定來控制該 step 當中呼叫外部程式執行並同時連線至同一 DB 的數量。
  - 舉例說明：有一組資源項目 TD-BULKLOAD，可同時連線數設定為 5，此時分別有三個 Job 為 Job-A、Job-B、Job-C，連線數分別都為 2，且排程時間相同，當執行時間開始，這三個 Job 只會先有兩個 Job 可以執行，需等到其中一個執行完畢，第三個才能 Run。
  - 需特別注意資源項目名稱相同時 Resource control 連線數採用順位為 Global Domain 優先，接著才是自定義的 Domain。
  - 此資源項目需預先在 Domain 中的 Resource Attribute 中設定，詳情請參考

「Trinity V5.0 Admin UI User Guide」手冊說明。

- Usage Count：設定這個 step 在使用該資源時的使用數目(占用數目)。
- Don't Apply File Argument：勾選代表當此 Job 有使用 File Source 時，不會把 File 存放路徑當為參數之一傳入執行指令最後。預設此項目未勾選，代表當有設定 File Source 時，會將 File 存放路徑自動帶入執行指令最後。
- Run As：改用其他的 OS Account 去執行，必需先在 Connection 中設定 (Type 為 OS Account)。RUN AS 功能是讓 JCSAgent 執行一個 External Command 時以切換帳號的方式去執行(主要是切換權限)，如同使用 su 指令一樣，但 RUN AS 不會變更當前登入的帳號身份，當前的登入帳號一定是執行 JCSAgent 服務的帳號。詳情請參考「Trinity V5.0 Admin UI User Guide」手冊說明。
- Load User's Profile：若改用其他 OS Account 執行，且需用到此 OS Account 的環境變數，則可勾選此項。註：Windows 環境不需勾選 Load User's Profile(此功能僅提供給 Linux or Unix)，User 自有的環境變數不會帶入。External Command 中用 Run As 執行與 Windows Login 的情況不同，Agent 為 Windows 的一個服務。

## Script 頁面說明

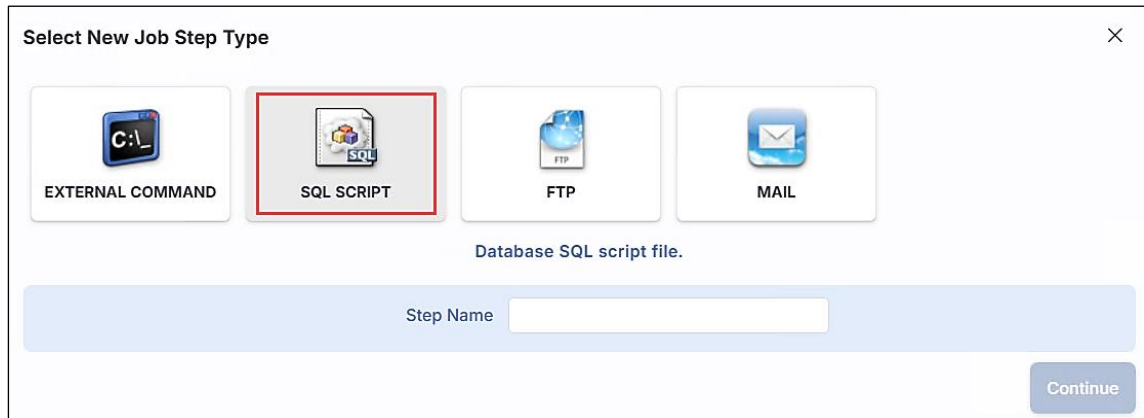


- Embedded Script：使用嵌入式方式執行，可以自動產生檔案，不需要先把檔案準備好
- File Extensions：可自行定義附檔名名稱，例：.perl, .bat
- Script：Script 內容

註：「Statistic」頁面和「Postcondition」頁面部分則是每種 Step 皆具備，於後續章節統一介紹。

## Job Step - SQL Script

SQL Script 類型的 Job Step 是用來執行一項數據庫的 SQL Script File，提供對不同數據庫的 SQL Script 一致的執行設定方式。



### General 頁面說明

The screenshot shows the "General" tab of the Job Step configuration interface. The "Step Name" is "SQL", "Step Type" is "SQL SCRIPT", and "Activate" is checked. The "Script Type" is "SQL-DB2". Other fields include "BusEntity" (TEST), "Category" (LAB1), "Job" (JOB1), "Description", "Success Rule" (Equal To), "Success Value 1" (0), "Success Value 2" (0), "Script Path", "Script Name", "Argument", "Encoding", "DataBase" (Select), "Connection", "Resource" (Select), and "Usage Count" (1).

在 SQL Script 的設定頁面中的各個輸入欄位的意義分別是：

- Step Name：作業步驟的名稱
- Activate：此作業步驟是否為有效狀態，作業狀態必須是有效狀態才會被執行
- Description：此作業步驟的描述說明
- Success Rule：判斷執行成功條件，有兩種選項：
  - Equal To：此步驟完成後的回傳值等於 Success Value 1 欄位輸入值時判斷執行成功
  - In Between：此步驟完成後的回傳值介於 Success Value 1 欄位輸入值與

Success Value 2 欄位輸入值判斷執行成功

- Script Type：設定 SQL Script 的類別
  - SQL-DB2：執行 DB2 數據庫的 SQL Script，不以 Auto-Commit 方式執行
  - SQL-DB2AC：執行 DB2 數據庫的 SQL Script，以 Auto-Commit 方式執行
  - SQL-MSSQL：執行 MS SQL Server 數據庫的 SQL Script
    - ◆ 使用前須先確定環境變數是否已設定完成。
    - ◆ Trinity/JCSAgent/cfg/jcsagent.conf 設定檔中修改：  
SCRIPTFILE\_SQLMSSQL\_PROGRAMPATH=SQLCMD
    - ◆ Connection 設定方式請參考「Trinity V5.0 Admin UI User Guide」手冊說明。設定範例：

The screenshot shows the configuration window for a script. The 'General' tab is selected. The 'Script Type' dropdown is set to 'SQL-MSSQL'. The 'Script Path' is 'C:\', 'Script Name' is 'TEST.SQL', and 'Argument' is empty. The 'Encoding' field is also empty. The 'DataBase' dropdown is set to 'SQL\_DB'. The 'Connection' field is empty. The 'Resource' dropdown is set to 'Select'. The 'Usage Count' is '1'. There are also 'Statistic' and 'Postcondition' tabs visible at the top.

- SQL-MYSQL：執行 MySQL 數據庫的 SQL Script
- SQL-Oracle：執行 Oracle 數據庫的 SQL Script
  - ◆ 使用前須先確定環境變數是否已設定完成
  - ◆ Trinity/JCSAgent/cfg/jcsagent.conf 設定檔中修改：  
SCRIPTFILE\_SQLORACLE\_PROGRAMPATH=sqlplus /nolog
  - ◆ Connection 設定方式請參考「Trinity V5.0 Admin UI User Guide」手冊說明。
  - ◆ 在執行的 SQL 檔中的 SQL Script 的語法應參照以下樣本：

```
WHENEVER SQLERROR EXIT SQL.SQLCODE;
CONNECT ${USERID}/${PASSWD}@${SERVER};

set echo on;
select 1
from dual;

QUIT 0;
```

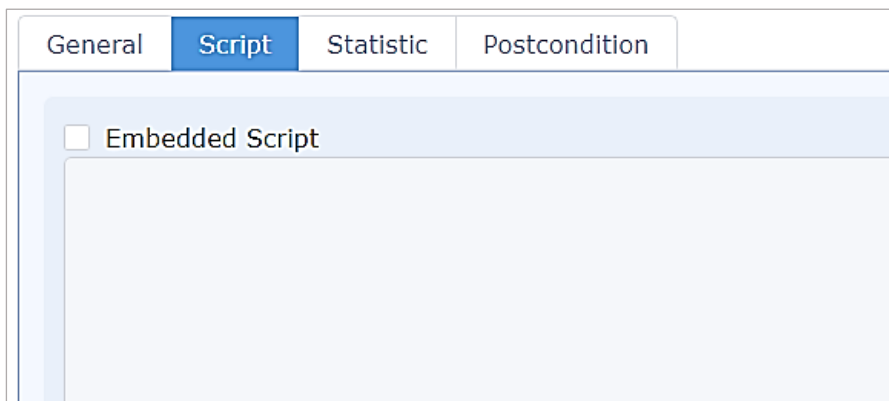
- ◆ 設定範例：

The screenshot shows the 'General' configuration tab for a script. The fields are as follows:

- Script Type: SQL-Oracle (dropdown)
- Script Path: C:\ (text input)
- Script Name: TEST.SQL (text input)
- Argument: (empty text input)
- Encoding: (empty text input)
- DataBase: SQL\_DB (dropdown with a browse button)
- Connection: (empty text input)
- Resource: Select (dropdown)
- Usage Count: 1 (text input)
- Control: (empty text input)

- SQL-PostgreSQL：執行 PostgreSQL 數據庫的 SQL Script
- SQL-SybaseIQ：執行 SybaseIQ 數據庫的 SQL Script
- SQL-Vertica：執行 Vertica 數據庫的 SQL Script
- Script Path：Script 放置的路徑
- Script Name：Script 名稱
- Argument：執行 Script 時所需的參數
- Encoding：設定編碼
- DB Connection：設定資料庫連結
- Resource Control：設定資源使用項目，共可設定不同的四組。此項目的作用是為了避免同時連線至同一資料庫進行存取，因連線數過多造成 DB 執行效能降低，故透過資源項目及使用數目的設定來控制該 step 當中呼叫外部程式執行並同時連線至同一 DB 的數量。
  - 舉例說明：有一組資源項目 TD-BULKLOAD，可同時連線數設定為 5，此時分別有三個 Job 為 Job-A、Job-B、Job-C，連線數分別都為 2，且排程時間相同，當執行時間開始，這三個 Job 只會先有兩個 Job 可以執行，需等到其中一個執行完畢，第三個才能 Run。
  - 需特別注意資源項目名稱相同時 Resource control 連線數採用順位為 Global Domain 優先，接著才是自定義的 Domain。
  - 此資源項目需預先在 Domain 中的 Resource Attribute 中設定，詳情請參考「Trinity V5.0 Admin UI User Guide」手冊說明。
- Usage Count：設定這個 step 在使用該資源時的使用數目(占用數目)。

## Script 頁面說明



- Embedded Script : 使用嵌入式方式執行，允許使用者在此編輯腳本內容，執行時 Trinity 可以自動產生檔案，不需要先把腳本準備好
- Script : Script 內容

註：「Statistic」頁面和「Postcondition」頁面部分則是每種 Step 皆具備，於後續章節統一介紹。

## Job Step – FTP

FTP Step 類型的 Job Step 是用來上傳檔案到指定的 FTP 目錄。

Select New Job Step Type

EXTERNAL COMMAND    SQL SCRIPT    **FTP**    MAIL

Upload file to FTP

Step Name

Continue

### General 頁面說明

Job(JOB1)    \*Step(Add FT X

BusEntity TEST    Category LAB1    Job JOB1

Step Name FTP    Step Type FTP  Activate

Description

Success Rule Equal To    Success Value 1 - 0 +    Success Value 2 - 0 +

General    File    Statistic    Postcondition

Target Connection Select ...

Error Handle Break

Protocol  SFTP

Option  Binary     Passive     Check Size After Post

在 FTP Step 的設定頁面中的各個輸入欄位的意義分別是：

- Step Name：作業步驟的名稱
- Activate：此作業步驟是否為有效狀態，作業狀態必須是有效狀態才會被執行
- Description：此作業步驟的描述說明
- Success Rule：判斷執行成功條件，有兩種選項：
  - Equal To：此步驟完成後的回傳值等於 Success Value 1 欄位輸入值時判斷執行成功
  - In Between：此步驟完成後的回傳值介於 Success Value 1 欄位輸入值與 Success Value 2 欄位輸入值判斷執行成功
- Target Connection：須先至 Connection 設定使用的 FTP 連結，設定方式請參考「Trinity V5.0 Admin UI User Guide」手冊說明。
- Error Handle：發生錯誤時的處理模式
  - Continue：跳過錯誤，繼續處理下一筆

- Break : 結束 · return value = 1
- Delete Uploaded & Break : 刪除之前上傳成功的檔案並結束 · return value = 2
- Protocol : 協定
  - SFTP : SSH FTP
- Option : 連接 FTP 和處理檔案的方式
  - Binary : 啟用 Binary Mode
  - Passive : 啟用 Passive Mode
  - Check size after post : 檔案上傳完成後檢查檔案大小是否相同

## File 頁面說明

- Upload File Path : 上傳檔案來源與目的地
  - File or dir Path : 檔案來源目錄
  - File Name :
    - 檔案來源名稱；支援處理多個符合 Pattern 的檔案，例如 File Name 輸入「FTP\_AP?????.list」，則目錄下為「FTP\_AP」開頭且四碼之後為「.list」結尾者（例如：FTP\_AP1202.list、FTP\_AP1203.list）皆會處理。
  - Remote Path : 遠端目錄
  - Rename : 覆寫檔案名稱
- File Type : 能直接上傳檔案，也能透過事先編輯好的檔案清單，依序將清單中的檔案上傳至目的端
  - Regular Data : 勾選則表示要上傳一般檔案
  - Control File : 勾選則表示利用檔案清單上傳
  - File List 檔 ( 檔案清單檔 ) 內容包含檔名、檔案大小，中間用空白字元分隔，檔案大小可以不輸入

- File List 內列出來的檔案，必須與 File List 檔案位在相同目錄下
  - ◆ 如果 File List 內容有包含 (檔名 + 檔案大小)，就會依照檔案大小檢查，若無則不檢查
  - ◆ File List 內列出來的檔案名稱；支援處理多個符合 Pattern 的檔案，例如輸入「FTP\_AP?????.list」，則目錄下為「FTP\_AP」開頭且四碼之後為「.list」結尾者 (例如：FTP\_AP1202.list、FTP\_AP1203.list) 皆會處理。
- File Validation：檔案驗證，須滿足條件才會進行上傳的動作。
  - 若不滿足 return value = 3
  - None：無
  - All File Exist：  
所有檔案必須存在；  
若 File Type 為 Control File，則執行時會檢查 File List (檔案清單檔) 是否存在，以及 File List 內所列出的檔案是否也存在，如果任一個檔案不存在，會在 Log 上顯示此檔案不存在，並且此 Step 為 Error 然後結束。(也就是說須整批檔案皆完整存在，才符合檢察)
  - All File Size(byte) >=：  
所有檔案大小必需大於等於後方欄位輸入值；  
若 File Type 為 Control File，則只會檢查 File List (檔案清單檔) 本身的 size，並不會檢查 File List 內所列出的檔案 size，因為 File List 內所列出的檔案，就可以包含檔案大小，故會依照各個檔案的設定再進行檢查。
  - All File Text Line Number >=：  
所有檔案行數必需大於等於後方欄位輸入值；  
若 File Type 為 Control File，則只會檢查 File List (檔案清單檔) 本身的 Text Line Number，並不會檢查 File List 內所列檔案的文字行數。
- Post Local Action：上傳後的動作
  - None：無
  - Delete Local File：刪除來源檔案
  - Move Local File To：移動來源檔案到指定目錄；若 File Type 為 Control File，會連同 File List (檔案清單檔) 一起移動。
  - Move Local File Overwrite To：移動來源檔案到指定目錄並覆寫；若 File Type 為 Control File，會連同 File List (檔案清單檔) 一起處理。

註：「Statistic」頁面和「Postcondition」頁面部分則是每種 Step 皆具備，於後續章節統一介紹。

## Job Step – Mail

Mail Step – 提供寄送電子郵件到指定的郵件帳號，並可以指定所需附加的檔案。

Select New Job Step Type

EXTERNAL COMMAND    SQL SCRIPT    FTP    MAIL

Send Mail

Step Name

Continue

### General 頁面說明

Job(JOB1)    \*Step(Add MAIL) X

BusEntity TEST    Category LAB1    Job JOB1

Step Name MAIL    Step Type MAIL  Activate

Description

Success Rule Equal To    Success Value 1 - 0 +    Success Value 2 - 0 +

General    File    Statistic    Postcondition

Mail Server Select ...

Subject

Content  Input Content File Path  Delete File

Input Content

Mail From

- Step Name：在加入新的 Step 時輸入，會帶到 Mail Step 設定畫面。
- Active：設定 Step 是否 Active。如果不打勾在執行 JOB 的時候會跳過這個 Step。
- Description：輸入 Step 功能的詳細描述。
- Success Rule：判斷執行成功條件，有兩種選項：
  - Equal To：此步驟完成後的回傳值等於 Success Value 1 欄位輸入值時判斷執行成功
  - In Between：此步驟完成後的回傳值介於 Success Value 1 欄位輸入值與 Success Value 2 欄位輸入值判斷執行成功
- Mail Server：須先至 connection 設定 Mail Server 的 host 或 IP 及驗證的帳號和密碼

碼，設定方式請參考「Trinity V5.0 Admin UI User Guide」手冊說明。

- Subject：主旨。
- Content：Mail 內容，有下列兩種編輯方式：
  - Input Content File Path：可先將內文部份編輯在文字檔中，然後直接指定此檔案路徑跟名稱，執行時就會將檔案內容帶入到內文中。若有勾選 Delete File，則在執行後會將此文字檔刪除。
  - Input Content：直接於此選項下方的輸入格當中編輯內文。
  - 內文支援以 html 的格式輸入
  - 可使用變數來指定附加檔案的內容是否加到內文當中。
    - ◆ 變數的格式為，第一個附加檔案\${attach1}，第二個\${attach2}...此類堆。
    - ◆ 如需以純文字輸出，於內文前後加上

```
<pre>...</pre>
```
- Mail From：寄件者的 E-Mail。這邊所代表的是當對方收到信時，看到的寄件者的 E-Mail，但並不是真的使用這個帳號寄出的，實際寄出的帳號是使用預先在 connection 中設定 Mail Connection 內的帳號。Mail Connection 的設定方式請參考「Trinity V5.0 Admin UI User Guide」手冊說明。

## File 頁面說明

- Mailing List File Path：存放著寄送目標 mail address 清單的檔案位置；收件者的 mail address 必須事先以文字檔的方式編輯，且此文字檔需位在所使用的 JCS Agent 所能讀取得到的位置，其內容格式包括：
- 斷行

舉例說明：

```
aaa@mail.com
bbb@mail.com
```

- 分號

舉例說明：

```
aaa@mail.com;bbb@mail.com;
```

- 逗號

舉例說明：

```
aaa@mail.com,bbb@mail.com
```

- Send mail if :

- File Exists：如果有附加檔案，當檔案不存在的時候，是否寄信，預設為不管是否存在都寄信，當選擇此項目時，則檔案必須存在才會寄送此信件。
- File Size(Bytes) >=：如果有附加檔案，當其中一個附件的檔案大小，大於等於後方輸入格設定值時則寄送此信件
- Text File Line Number >=：如果有附加檔案，當其中一個附件的檔案行數，大於等於後方輸入格設定值時則寄送此信件

- Action After Sending Mail：

- Do Nothing：如果有附加檔案，在寄信之後不做任何動作。
- Delete File：如果有附加檔案，刪除附件的來源檔案
- Move File To (Folder Path)：如果有附加檔案，移動來源檔案到指定目錄

- Attach File Path：可輸入多組附加檔案

- File Path：設定附加檔案的位置
- File Name：設定附加檔案的檔名，支援萬用字元(\*)
- Charset：
  - ◆ 可輸入所附加檔案的編碼，例：GB2312、UTF-8、BIG5。
  - ◆ Charset 未填，則預設值為 UTF-8。
  - ◆ 附加檔案為 binary 格式如 Excel，可輸入 binary，檔案將不做任何編碼處理。
- ZIP File：勾選代表將檔案以壓縮方式寄送。

## Statistic 頁面說明

透過 Notification 的通知機制及判斷執行時間的設定，即可讓使用者在作業執行過久時收到通知。(Notification 的通知機制需設定「Timing」為「When Task Overtime」)

The screenshot shows the 'Statistic' configuration panel with the following settings:

- Overtime Rule: Compare With Max Seconds
- Difference Base: Seconds
- Difference Value: 0
- Min Seconds: 300
- Max Seconds: 300
- Avg Seconds: 300
- Collect Statistic:

### 欄位說明

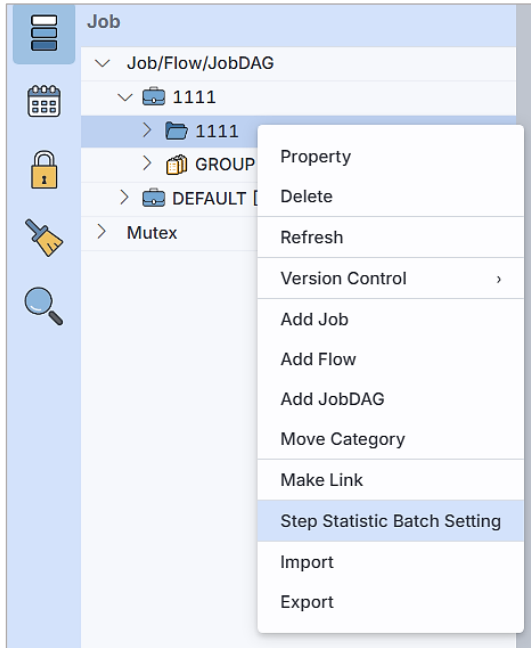
- Overtime Rule：設定此 Step 執行超時的規則，有三種選項：
  - None：沒有規則，即不通知使用者此 Step 的執行時間已超時
  - Compare With Max Seconds：與自己設定或系統統計的最大 Step 執行時間作比較
  - Compare With Avg Seconds：與自己設定或系統統計的平均 Step 執行時間作比較
- Difference Base：用來判斷過長時間的計算單位，有兩種選項：
  - Seconds：代表以時間來計算，單位為秒數
  - Percentage：代表以差距比率(Percentage)來計算，單位為百分比
- Difference Value：當 DifferentBase 為 'Seconds' 時，此欄位記錄比基準時間超過多少秒數即被視為執行過久，當 DifferentBase 為 'Percentage' 時，此欄位記錄比基準時間超過多少比例時間即被視為執行過久。
- Min Seconds / Max Seconds / Avg Seconds：呈現目前此 Step 的統計資料，也可供使用者自行設定值
- Collect Statistic：設定是否要對此 Step 進行執行時間的收集統計

## Step Statistic batch setting 說明

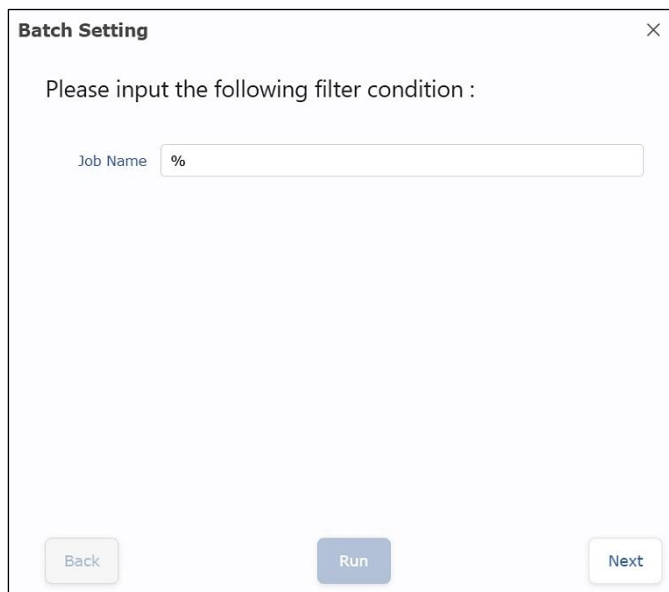
利用此方式，可以批次設定多個作業步驟(Step) Statistic 的設定。

首先，在 Category 階層上點選按右鍵，點選「Step statistic batch setting」

(使用者必須對 Category 項目或其所在的 Business Entity 項目擁有 Edit 的權限才能執行此動作。)



之後會出現新增 statistic 的對話窗，在此可輸入 Job 名稱，再點選「Next」來過濾篩選要設定的作業。



接著會出現一條件篩選的步驟 ( Step )，在此可勾選想要批次設定的步驟後，再點選「Next」繼續設定。

**Batch Setting** ×

Please select following Job steps:

<input checked="" type="checkbox"/>	Job Name	Step Name
<input checked="" type="checkbox"/>	DEP1	STEP1
<input checked="" type="checkbox"/>	DEP1	STEP2
<input checked="" type="checkbox"/>	DEP2	STEP1
<input checked="" type="checkbox"/>	DEP2	STEP2
<input checked="" type="checkbox"/>	JOB1	STEP1
<input checked="" type="checkbox"/>	JOB1	STEP2
<input checked="" type="checkbox"/>	JOB2	STEP1
<input checked="" type="checkbox"/>	JOB2	STEP2
<input checked="" type="checkbox"/>	JOB3	STEP1
<input checked="" type="checkbox"/>	JOB3	STEP2

接著會出現下方的視窗畫面，在此可以輸入設定的值，完成後再點選「Next」。

**Batch Setting** ×

Please set following step statistic properties:

Overtime Rule

Difference Base

Difference Value

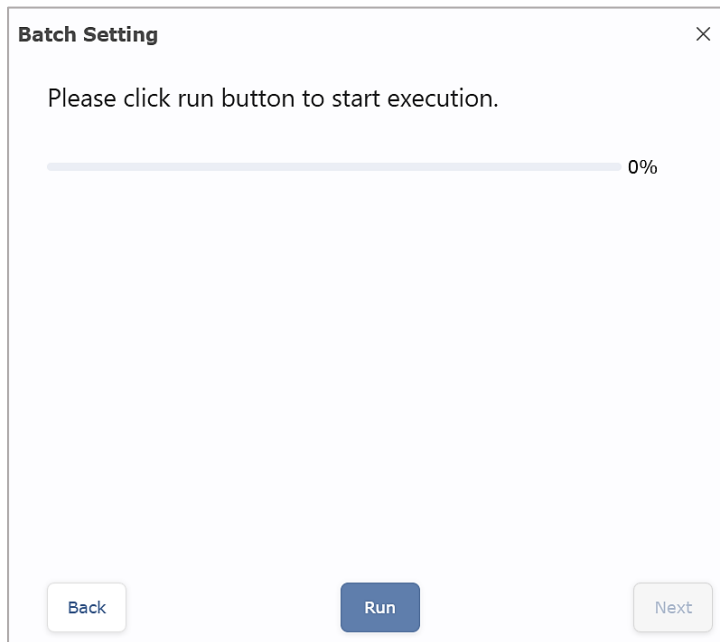
Min Seconds

Max Seconds

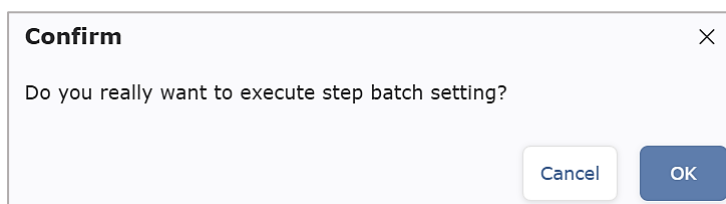
Avg Seconds

Collect Statistic

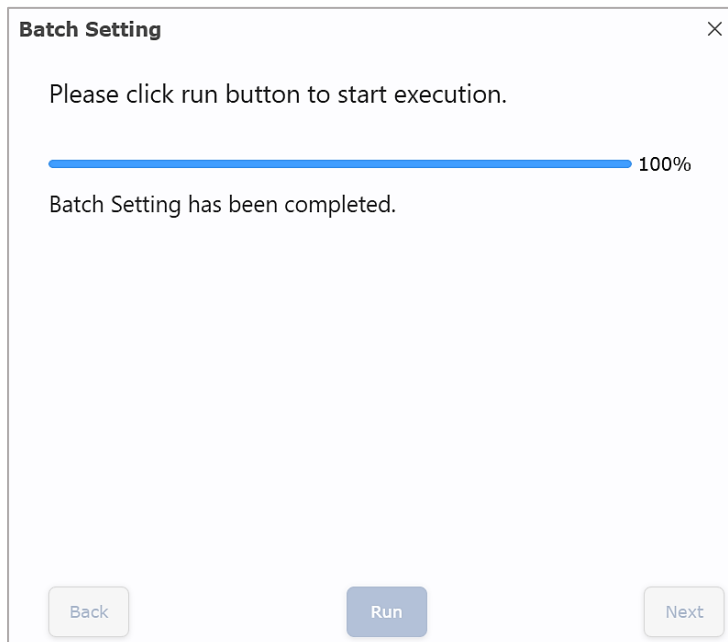
點選「Next」後，會出現下面視窗，確認是否要繼續執行。



點選「Run」執行時，系統會再次確認是否要執行 batch setting



點選「OK」後，系統會自動執行，到 100%後，即完成批次設定作業



## Step Postcondition

每一個 Step 都可以於「Postcondition」頁面，使用 Groovy 語法來編寫腳本，控制該 Step 執行結束後的動作(例如「執行下一個 Step」、「回到上一個 Step」、「執行指定 Step」、..... 等)。

General	Header	Body	Response	Statistic	Postcondition
<input checked="" type="checkbox"/> Apply Postcondition					
<pre> 1  if (READRECORD &lt; 1) 2  exportGlobal("VAR1", 700); 3  else if ((DEF_MARK == "13" ) &amp; ( ITEM_LIST_MARK &lt;= ITEM_LIST_COUNT)) 4  gotoStep("DEF_LIST"); </pre>					

若欲使用此功能，需在 Job 設定畫面勾選「Use Step Postcondition」。

General	Execution Time	Forecast	Note	Variable	Step	Stream/Trigger	Dependency	Source/Target	Notification
Domain		Default		Agent		127.0.0.1			
Frequency		Select		File Source		Select			
Retry Option		<input type="checkbox"/> Retry After Job Failed		Retry Mode <input checked="" type="radio"/> From Beginning <input type="radio"/> From Breakpoint <input type="radio"/> Only File Receive					
Retry Interval		1 min		Max Retry Time		1 time			
Time Window		Time Window Begin		00:00		Time Window End		23:59	
TxDate Setting		TxDate Rule		System Date		TxDate Offset		- 0 + Select View TxDate	
Priority		Priority		- 1 +		<input checked="" type="checkbox"/> Bypass Error <input type="checkbox"/> Critical Job		CreatedBy root <input type="checkbox"/> Use Step Postcondition	
Notification Setting		Waiting Time		1 min					
Task Attribute		<input type="checkbox"/> Skip Missing Task <input type="checkbox"/> Only Apply Completed Task <input type="checkbox"/> Don't Save History <input type="checkbox"/> Rerun From FileSource							
Online Date		<input type="checkbox"/> Online Date Time <input type="checkbox"/> Offline Date Time							
Job Alert		<input type="checkbox"/> Start In Time <input type="checkbox"/> Finish In Time <input type="checkbox"/> Avg Time           - 1 +							

以下列出內定的參數和函式

- STEPCOUNT : step 執行次數
- STEPRETURNCODE : step 回傳值
- STEPSTATUS : step 執行結果 ('success' or 'fail')
- STEPSTARTDATE : step 開始日期 (yyyyMMdd)
- STEPSTARTTIME : step 開始時間 (hhmmss)
- STEPENDDATE : step 結束日期 (yyyyMMdd)
- STEPENDTIME : step 結束時間 (hhmmss)
- STEPELAPSEDTIME : step 花費時間 (sec)
- READRECORD : read 筆數
- INSERTRECORD : insert 筆數

- UPDATERECORD : update 筆數
- DELETERECORD : delete 筆數
- ERROR1RECORD : error1 筆數
- ERROR2RECORD : error2 筆數
- ERROR3RECORD : error3 筆數
- first () : 跳至第一個 activate 的 Step
- last () : 跳至最後一個 activate 的 Step
- next () : 跳至下一個 activate 的 Step
- back () : 跳至上一個 activate 的 Step
- redo () : 重新執行此 Step
- gotoStep ("STEP 的名稱") : 跳至指定名稱的 Step · 例如 gotoStep("STEP3")
- exitS () : 結束 Job 且狀態為 success
- exitF () : 結束 Job 且狀態為 fail
- export ("變數名稱", 變數值) : 將變數匯出到工作變數 · 但不會寫入資料庫 · 也不會在該 Step 生效 · 而是須在下一個 Step 才會生效 · 例如 export("VAR1", 500)
- exportGlobal ("變數名稱", 變數值) : 將變數匯出到 Global 工作變數 · 會寫入資料庫 · 不會在該 Step 生效 · 在下一個 Step 會生效 · 例如 exportGlobal("VAR1", 500)

註：工作變數的說明請見本文件「附錄 A」介紹

另外腳本中也可以使用自訂的 Domain Variable、Entity Variable、Category Variable、Job Variable 以及 Step Variable。

請注意使用這些參數時，名稱只接受大寫的形態，並且 READRECORD 等筆數的值為

**Integer** 型態，其餘是以字串表示(使用時須以雙引號包住字串)，舉例：

```
if (READRECORD < 1)
    exportGlobal("VAR1", 700);
else if ((DEF_MARK == "13" ) & ( ITEM_LIST_MARK <= ITEM_LIST_COUNT))
    gotoStep("DEF_LIST");
```

如果沒有符合的函式需要執行，就按照原本的機制，成功就繼續執行下一個 Step，失敗就結束 Job，舉例：

```
if ( STEPRETURNCODE == 2 )
    last( );
else if (STEPRETURNCODE == 3 )
    exitF( );
```

這個例子中，若 STEPRETURNCODE 為 2，則跳至最後一個 activate 的 Step；若 STEPRETURNCODE 為 3，則結束 Job 且狀態為 fail；若皆不符合，則依照執行結果決定是否繼續進行下一個 Step。

## Step Variable

設計 Step 內容時，若透過指定格式並且宣告於行首，便可以自由定義變數，且此變數可被之後的 Step 所使用，但此變數只會存活在 Job 的層級。

### JCS\_STEP\_VAR

宣告變數的格式如下：`@@JCS_STEP_VAR %varname %varvalue`

例如 `@@JCS_STEP_VAR VAR1 1000` 代表宣告變數名稱為 VAR1 且預設值為 1000。

運用的方式是設法在 Agent 上執行這一段宣告，當 Trinity JCSServer 偵測到「@@JCS\_STEP\_VAR」指令，便會進行建立 Step Variable 的動作。可採取在 OS 上以指令執行，或製作 shell 檔、.bat 檔後執行。

運用方式舉例說明：

步驟一：建立 Job，且本範例為了後續驗證，設計上需勾選「Use Step Postcondition」。

步驟二：建立 STEP1，種類為 External command，編輯「General」頁面中的「command」欄位，輸入「`echo @@JCS_STEP_VAR VAR1 1000`」，使該 Job 所使用的 Agent 於作業系統中執行此 echo 語法。當 Trinity JCSServer 偵測到「@@JCS\_STEP\_VAR」tablen 指令，便會進行建立 Step Variable 的動作。

General	Script	Statistic	Postcondition
Command	echo @@JCS_STEP_VAR VAR1 1000		
Argument			
Encoding			

步驟三：建立 STEP2，種類為 External commnd，編輯「General」頁面中的「command」欄位，輸入「echo \${VAR1}」試圖印出前一個 Step (STEP1) 所建立的 Step Variable，來驗證 Step Variable 可存活於 Job 層級，供 Job 中的 Step 呼叫使用；編輯「Postcondition」頁面，勾選「Apply Postcondition」，輸入以下：

```
if ( VAR1 == '1000' )
    last( );
else
    exitF( );
```

代表若 STEP2 執行時「VAR1」為「1000」則執行「last()」跳至最後一個 activate 的 Step，否則執行「exitF()」結束 Job 且 Job 狀態為 fail。

General	Script	Statistic	Postcondition
Command	echo \${VAR1}		
Argument			

General	Script	Statistic	Postcondition
<input checked="" type="checkbox"/> Apply Postcondition			
1	if ( VAR1 == '1000' )		
2	last( );		
3	else		
4	exitF( );		
5			

步驟四：建立 STEP3，種類為 External commnd，編輯「General」頁面中的「command」欄位，輸入「echo "This is STEP3"」，做一個簡易的 step 執行；以相同內容建立 STEP4 及 STEP5。

General	Script	Statistic	Postcondition
Command	echo "This is STEP3"		
Argument			

步驟五：執行作業，結果是若 STEP1 執行完畢則會建立一個 Step Variable 為「VAR1」，STEP2 可使用 Step Variable 並印出其 value 值「1000」，且 STEP2 可用 VAR1 的變數值執行 if else 的判斷，判斷要接著執行 STEP5 還是要直接在 STEP2 執行完畢後就立刻結束 JOB 的執行。

ECHO::TEST.TEST_VAR					
Business Entity	Category	Task Name	Task Type	Count	Agent
ECHO	TEST	TEST_VAR	Job	1	127.0.0.1

Step Execution Log					
Task Name	Step Name	Count	Status	Start Date Time	End Date Time
TEST_VAR	STEP1	1	Success	2022-04-19 14:24:29	2022-04-19 14:24:31
TEST_VAR	STEP2	1	Success	2022-04-19 14:24:32	2022-04-19 14:24:34
TEST_VAR	STEP5	1	Success	2022-04-19 14:24:36	2022-04-19 14:24:38

Output Log	Received File Log	Script
1000		

## JCS\_GLOBAL\_VAR

宣告變數的格式如下：`@@JCS_GLOBAL_VAR %varname %varvalue`

例如 `@@JCS_GLOBAL_VAR VAR1 1000` 代表宣告全域變數名稱為 `VAR1` 且預設值為 `1000`。

運用的方式是設法在 Agent 上執行這一段宣告，當 Trinity JCSServer 偵測到「@@JCS\_GLOBAL\_VAR」指令，便會進行建立 Global Step Variable 的動作。可採取在 OS 上以指令執行，或製作 shell 檔、.bat 檔後執行。範例可參考本文件前一章節「JCS\_STEP\_VAR」的「運用方式舉例說明」敘述。

此變數不僅可以在 Job 內使用，還可以覆寫回 Job Variable、Category Variable、BusEntity Variable 或是 Domain Variable (於 Admin UI 中設定)，前提是原本的 Job Variable、Category Variable、BusEntity Variable 或是 Domain Variable 已存在此變數名稱，否則這個設定就無效。

如果 Job Variable、Category Variable、BusEntity Variable 和 Domain Variable 同時存在相同變數名稱，則覆寫的優先順位為 Job Variable 最高，其次為 Category Variable、BusEntity Variable，最後為 Domain Variable。

## JTextField & JTextArea & SQL Executor 控制字元的行為

在此章節中將介紹 在 Web JFDesigner JTextField 及 JTextArea 欄位，包含控制字元之說明。

JTextField：單行輸入之欄位（如下列圖示中的 Command、Argument、Encoding...等欄位）

JTextArea：多行輸入之區域（如下列圖示中的 Description）

JTextField 及 JTextArea 欄位，包含控制字元之行為：

項目	控制字元之行為
JTextField	控制字元均會被移除
JTextArea	控制字元均會被移除，除了下列三個字元不會被移除： ACII 碼值 9 為 tab ACII 碼值 10 為換行 ACII 碼值 13 為 carriage return

# 版本控管(VersionControl)介紹

## 功能目的

以往對於 Job(flow)執行編輯的情況下，單一使用者對單一 Job(Flow)在操作上是可無顧慮般地自由編輯的，但是若有第二位使用者對於相同的 Job(Flow)也可以執行編輯的話，就會產生對於資料庫存取不一致的情況。版本控管就是為了避免這個情況，使 Job(Flow)在新增、刪除、修改、查詢上不會發生衝突，且可排除相同 Job(Flow)設定資料存取不一致的情形。

如欲對既有的 Job(Flow)進行編輯，則需先將 Job(Flow)執行 **CheckOut**，當 Job(Flow)為 Check Out 狀態下，您才可以對其進行編輯。

如為新增 Job(Flow)，則新增時會自動視為新增者對這個新增的 Job(Flow)執行了 **CheckOut**。

Job(Flow)會因為時間的推演與實際需求的變動，而在各個時間進程對 Job(Flow)作不同的調整設定，在當下可對該 Job(Flow)進行 **儲存(update)**，或進行 **版本儲存(check in)**。執行 **儲存(update)** 之後該 Job(Flow)仍會維持在 CheckOut 狀態下；執行 **版本儲存(check in)** 之後該 Job(Flow)則會釋放 CheckOut 狀態，變成未 CheckOut 狀態，且將當下內容建立成一個正式版本儲存於系統中，此版本的內容後續可經由對該 Job(Flow)執行 **CheckOut** 取得，但作法是將該版內容複製一份出來成為新的版本，供您在此新版做編輯，此時的編輯並不會影響到系統中的版本。

若需取得舊版 Job(Flow)的內容時，則可進行 **版本回溯(roll back)**，指定到目標版本之後，配合後續執行 CheckOut 來將目標版本的內容取出。

## 功能操作相依關連

功能	功能執行目的	執行功能後，所衍生的功能	衍生功能的功用
Check Out	讓該使用者擁有唯一編輯權限。	Check In	詳見本表格 Check in 功能說明
		Release Lock	可釋放 CheckOut 狀態
		View Version History	提供使用者觀覽該 Job(Flow)之版本紀錄
		View Status	提供使用者觀覽該 Job(Flow)之版本訊息
		Force Release Lock	可強制釋放 CheckOut 狀態
		Compare	可對 Job 作版本差異比對
Check In	讓該使用者結束編輯狀態，並新增可回復版本。	Check In	詳見本表格 Check in 功能說明
		View Version History	提供使用者觀覽該 Job(Flow)之版本紀錄
		Roll Back	對存在 Job(Flow)作版本回復
		Compare	對 Job 作版本差異比對
		Restore	對已刪除 Job(Flow)作版本回復

### CheckOut 目的

運用 CheckOut 功能來限制單一使用者對此 Job(Flow)編輯權限，可避免發生多名使用者對相同 Job(Flow)同時存取資料造成存取衝突。若欲對此 Job(Flow)進行編輯，則須對該 Job(Flow)進行 CheckOut，當一個使用者 CheckOut，則其他使用者就不可 CheckOut。換句話說，當其他使用者想要對此 Job(Flow)執行編輯時，可透過觀察 Job(Flow)的 CheckOut 狀態來判斷是否已被編輯中。

### 因 CheckOut 狀態而衍生之相關操作

進入 CheckOut 狀態的 Job(Flow)，須被執行 ReleaseLock 或 ForceReleaseLock 才可釋放 CheckOut 狀態。

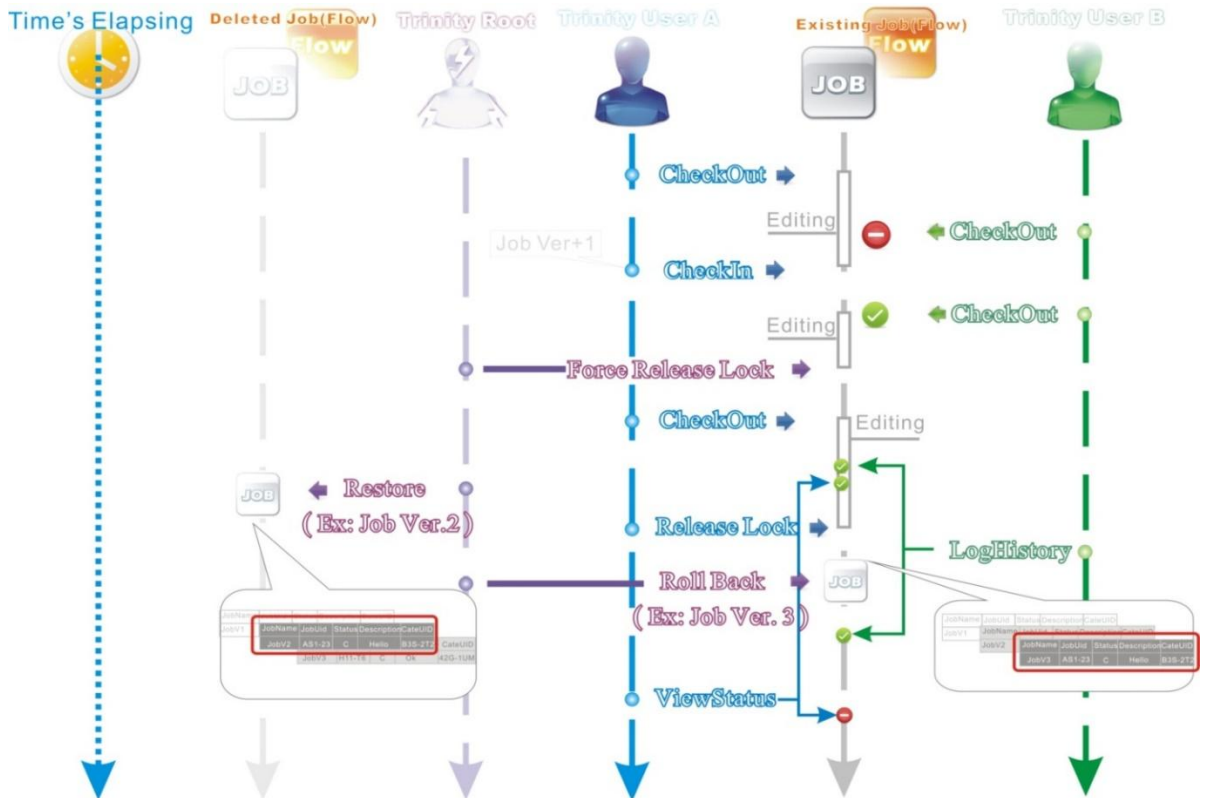
ReleaseLock 須由當初進行 CheckOut 的使用者執行；ForceReleaseLock 為強制執行，須由 root 使用者，或為 administrator 角色的使用者，或者為當初執行 CheckOut 的使用者但是當下 IP 與當初 CheckOut 時 IP 已不相同的情況下進行。

釋放 CheckOut 狀態的目的並不是為了新增下一版本的 Job(Flow)，而是為了提供編輯機會給其他使用者。

### 因 CheckIn 狀態而衍生之相關操作

當執行完 CheckIn 之後，會新增 Job(Flow)版本，有了版本資訊，此時 Job(Flow) 若依然存在於左方的樹狀結構，則可執行 RollBack 進行版本回復，回復到指定的版本；當情況是 Job(Flow)被刪掉、也就是不存在於左方的樹狀結構時，可點擊上方 Tool bar/Version Control/Restore 項目，依 BusEntity 和 Category 執行 Restore 版本回復。

功能圖解說明



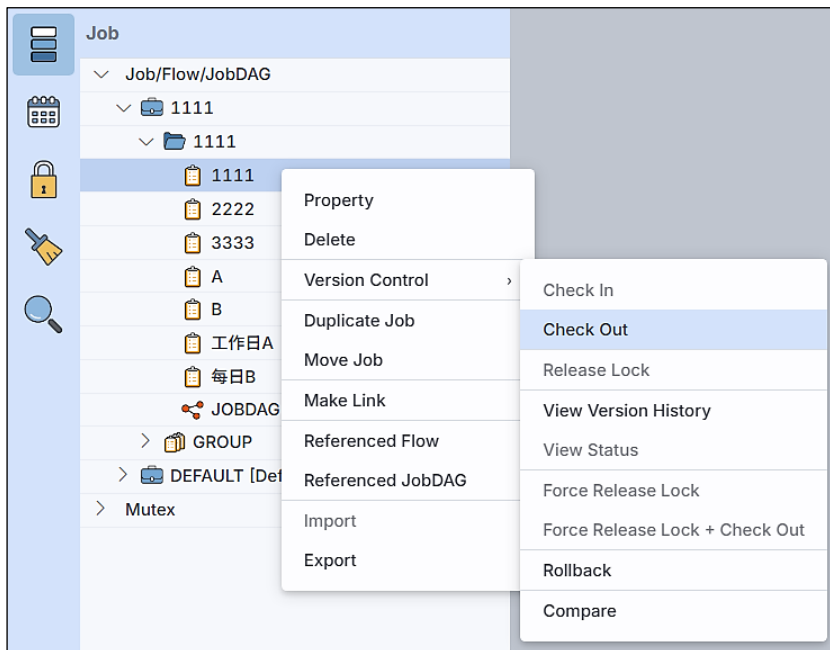
## 執行 CheckOut

### 功能簡介

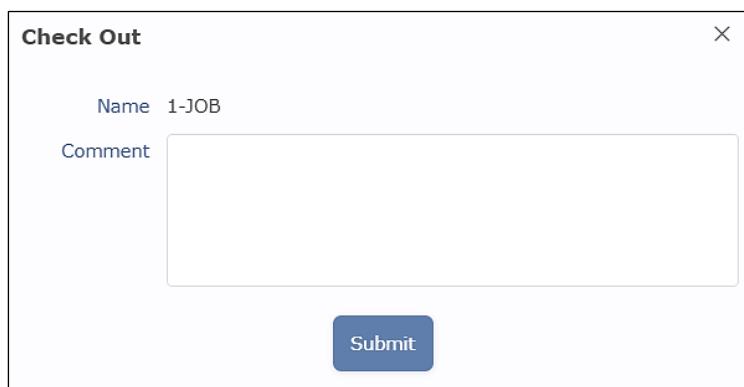
執行目的	可執行時機	影響
使 Job(Flow) 的編輯權限被限制為單一使用者	既有的 Job(Flow) · 且為未 CheckOut 狀態。	當該 Job(Flow)是未 CheckOut 狀態 · 此時其他使用者就可以對該 Job(Flow)執行 CheckOut 。
	既有的 Job · 對作業階層執行 Refresh 後 · 狀態改為未 CheckOut	若該 Job(Flow)原本被自己或者其他使用者使用中 · (也就是 CheckOut 狀態時) · 後來使用者本人對該 Job(Flow)執行 ReleaseLock · 或者由 root 使用者(或 administrator 角色的使用者)(或當初執行 CheckOut 的使用者但是當下 IP 與當初 CheckOut 時 IP 已不相同) 執行 ForceReleaseLock · 則 Job(Flow)會釋放 CheckOut 狀態 · 此時對作業階層執行 Refresh 後就會發現狀態已改變 · 改變為未 CheckOut 狀態 · 如此一來其他使用者就可以對該 Job(Flow)執行 CheckOut 。

## 執行畫面流程

於樹狀試圖區針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「CheckOut」。



CheckOut 彈出視窗介面，可輸入 Comment，藉此紀錄相關訊息。



CheckOut 後的狀態改變：



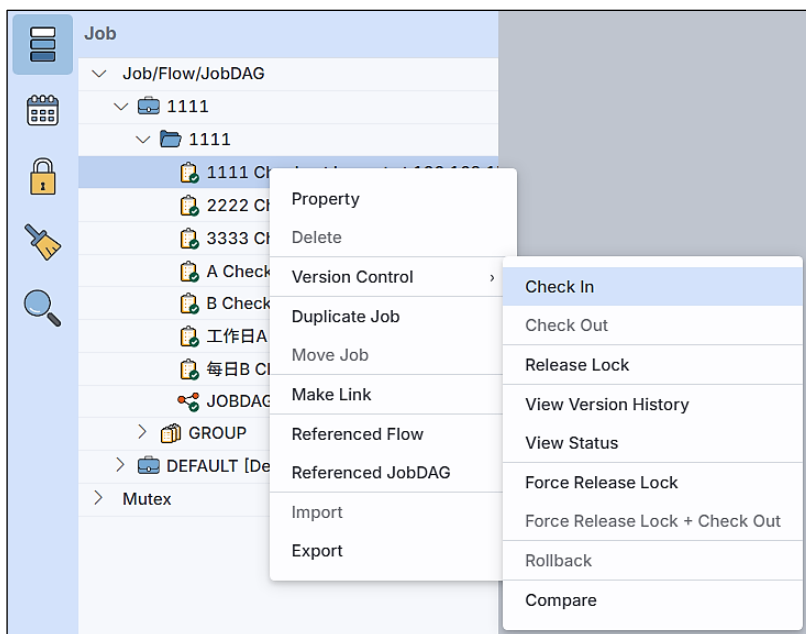
## 執行 CheckIn 新增 Job(Flow) 版本

### 功能簡介

執行目的	可執行時機	影響
將 Job(Flow)現有的內容保存為一個固定的版本，收存於系統中不可被異動，並將 Job(Flow)恢復成未 CheckOut 的狀態。	Job(Flow)狀態為 CheckOut 狀態，且是自身所執行的 CheckOut。	解除 CheckOut 狀態
		將 CheckOut 後到 CheckIn 之間的 Job(Flow)異動結果新增一個版本，並給予正式版號。

### 執行畫面流程

針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「CheckIn」。



CheckIn 彈出視窗介面，可輸入 Comment，藉此紀錄相關訊息。

 A screenshot of a dialog box titled 'Check In'. It has a close button (X) in the top right corner. The dialog contains a label 'Name' followed by the text '1-JOB'. Below this is a label 'Comment' followed by a large, empty text input area. At the bottom center of the dialog is a blue button labeled 'Submit'.

CheckIn 後的狀態改變：



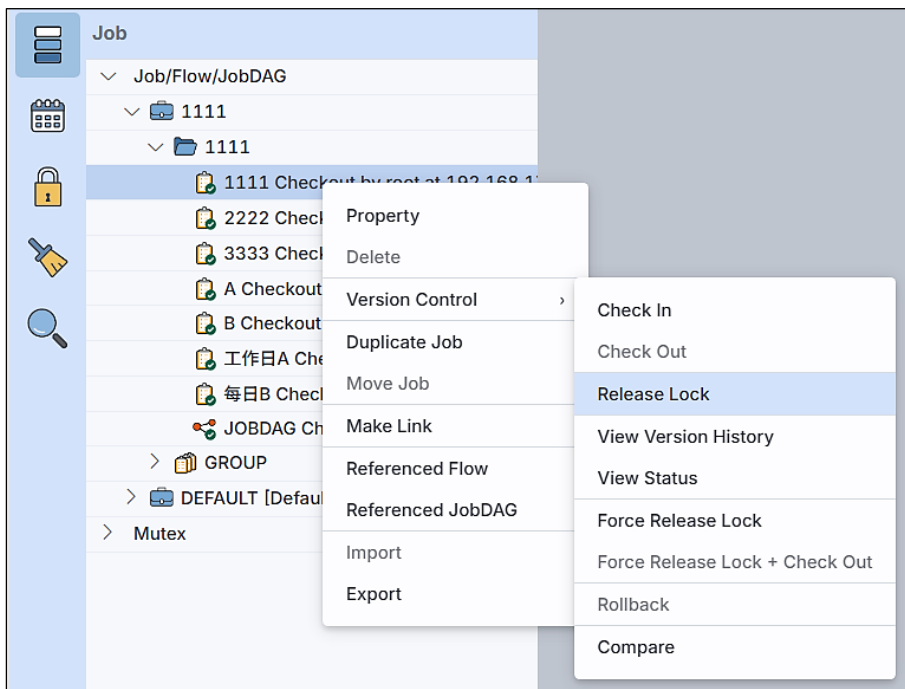
## 執行 ReleaseLock 以釋放 CheckOut 狀態

### 功能簡介

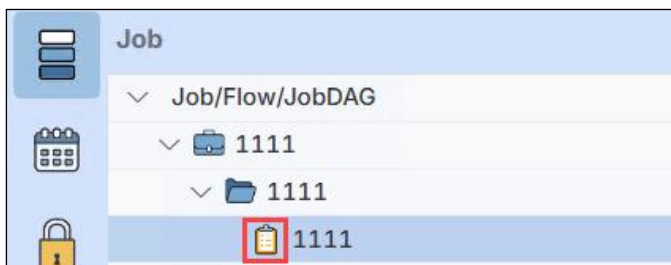
執行目的	可執行時機	影響
使得 Job(Flow)恢復成未 CheckOut 狀態。	Job(Flow)狀態為 CheckOut 狀態，且是自身所執行的 CheckOut。	執行後，Job(Flow)恢復成未 CheckOut 狀態，此時其他使用者(包含自身)可對其 Job(Flow)執行 CheckOut。

### 執行畫面流程

針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「ReleaseLock」。



Release Lock 後的狀態改變(如同 CheckIn)



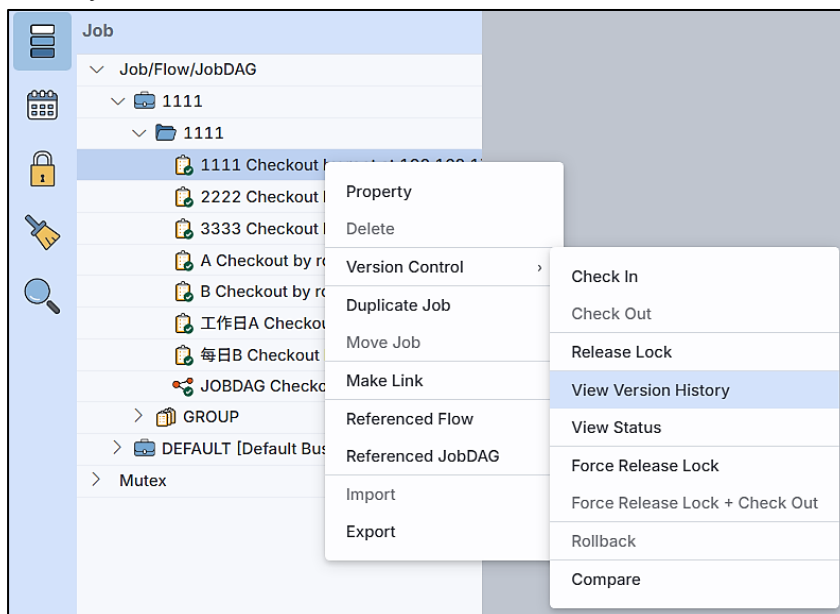
## 開啟 VersionHistory 觀覽歷屆 Job(Flow) 版本

### 功能簡介

執行目的	執行發動條件	功能執行影響所及
觀察此 Job(Flow)歷屆版本	隨時皆可點選觀覽	不產生任何影響；但可觀察到目前此 Job(Flow)已紀錄的歷屆版本

### 執行畫面流程

針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「View Version History」。



執行 VersionHistory 後，彈出歷屆版本資訊視窗。內容根據歷次 CheckIn 而存於 Job(Flow)VersionHistory 的資料紀錄所顯示。

Version	Check-In User	Check-In IP	Check-In Time	Comment
1	Trinity Super User	192.168.27.76	2025-04-28 14:05:36	

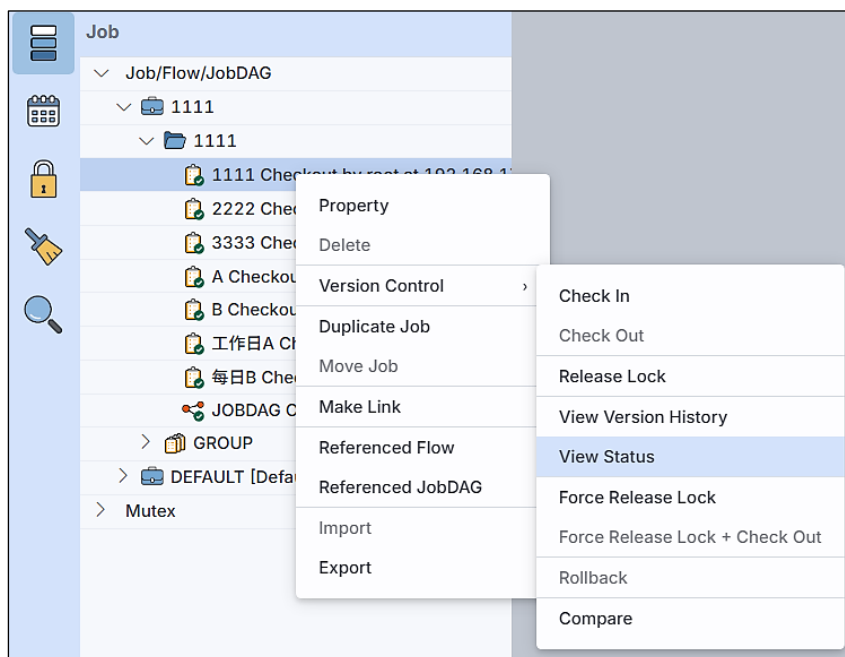
## 開啟 ViewStatus 查看 Job(Flow) 現階狀態

### 功能簡介

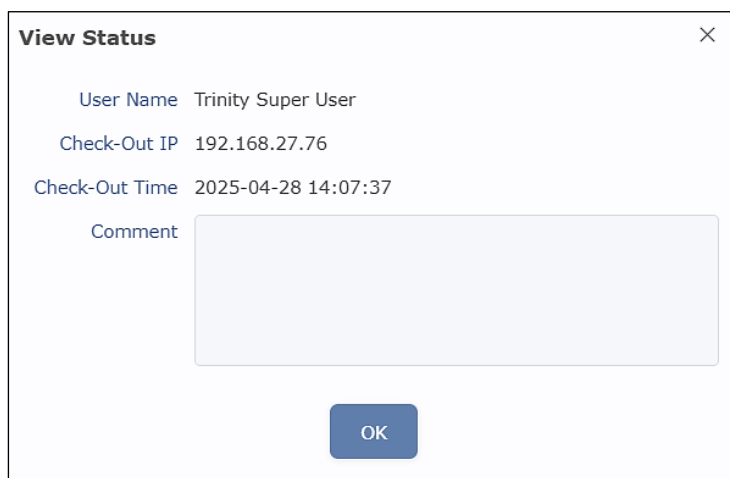
執行目的	執行發動條件	功能結束影響所及
觀察此 Job(Flow)目前被 CheckOut 的狀態	須被自身或別人所對此 Job(Flow)作 CheckOut	不產生任何影響；但可藉此判斷目前 Job(Flow)可否由自身執行 CheckOut

### 執行畫面流程

針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「ViewStatus」。



執行 ViewStatus 後，顯示目前 Job(Flow)的狀態資訊。內容列出 Job(Flow)的 CheckOut 使用者之 IP、Time、Name、Comment 訊息。



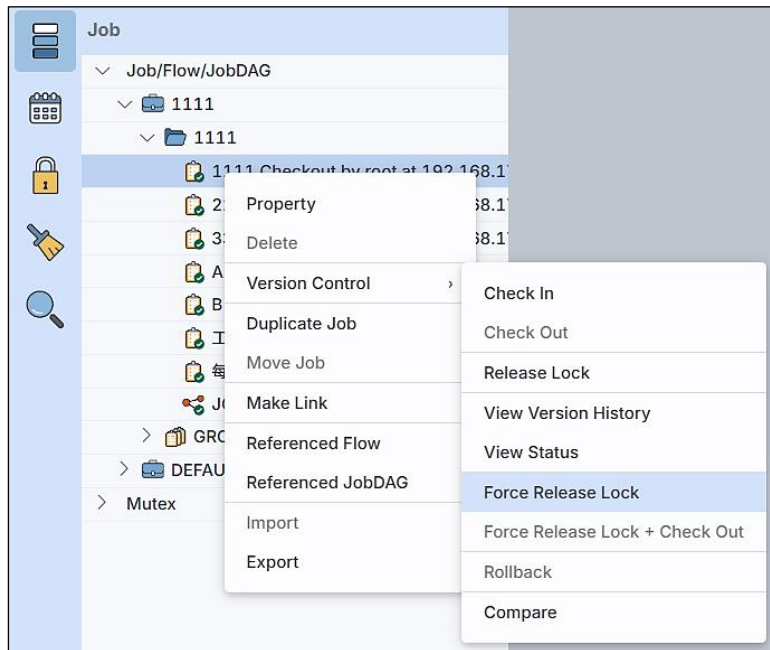
## ForceReleaseLock 強行釋放任一使用者對 Job(Flow) 之 CheckOut 狀態

### 功能簡介

執行目的	執行發動條件	功能結束影響所及
由 Root 使用者(或者是同一使用者，IP 不同時)強行介入，使得 Job(Flow)恢復成 Release Lock 狀態	現階 Job(Flow)狀態為 CheckOut	其他使用者(包含自身)可對其 Job(Flow)執行 CheckOut

### 執行畫面流程

針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「ForceReleaseLock」。



Force Release Lock 後的狀態改變(如同 CheckIn)：



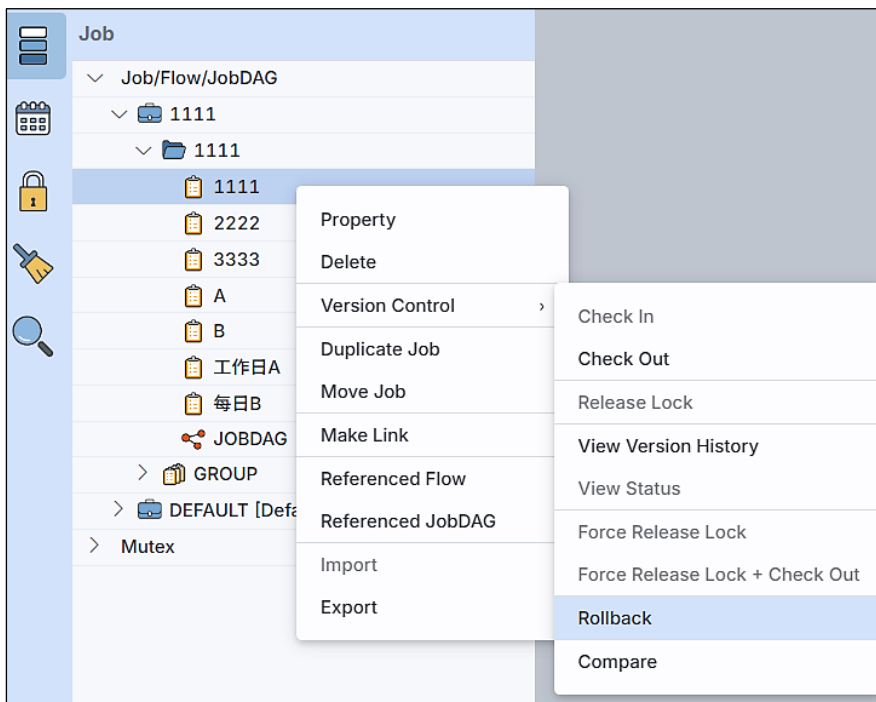
## 執行 RollBack 對已存在的 Job(Flow) 選擇版本回復

### 功能簡介

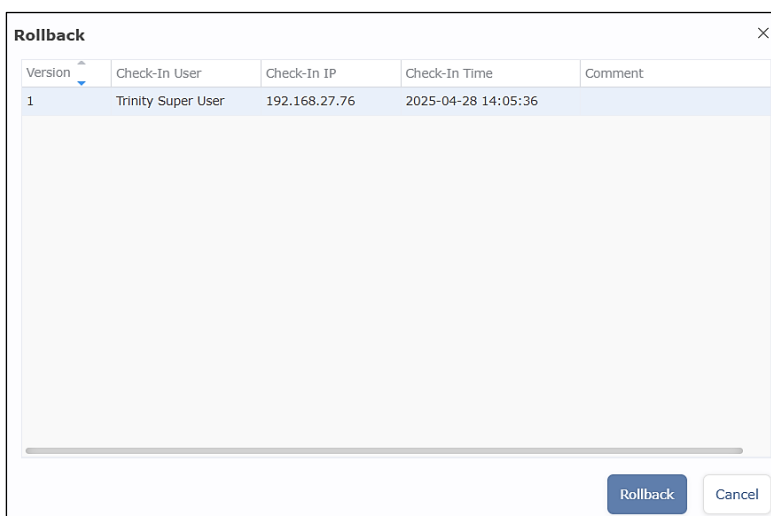
執行目的	執行發動條件	功能結束影響所及
將 Job(Flow)回復到以往版本	曾經作過 CheckIn · 且當下狀態不為 CheckOut 狀態。	依據版本的選擇而回溯到該版本時的相關設定狀態

### 執行畫面流程

針對目標 Job/Flow 右鍵點選「Version Control」展開選項選取「RollBack」。



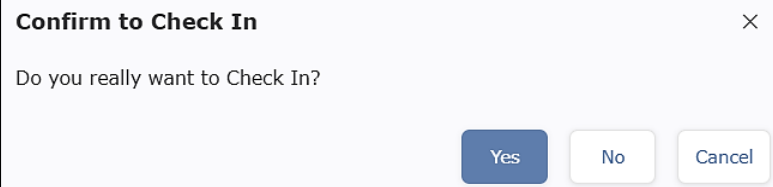
Roll Back 彈出視窗介面，供版本回復功能之作用。以滑鼠左鍵點選來指定目標版本，按下 Roll Back 便可執行版本回復。



Roll Back 執行之訊息確認視窗：

確認 JobName 與所選擇版本無誤

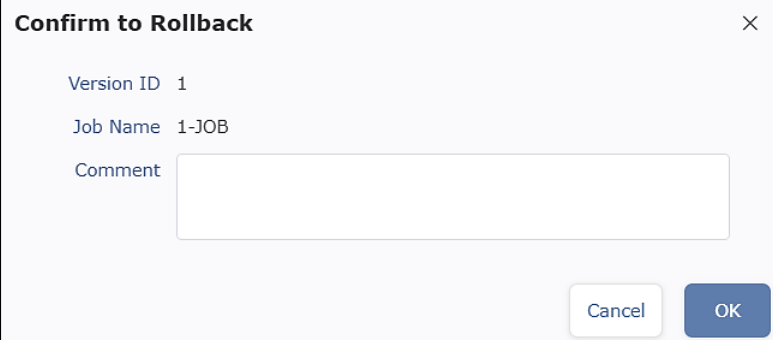
Comment 欄位可輸入如同 CheckIn 時的 Comment 以供註解



**Confirm to Check In** ×

Do you really want to Check In?

Yes No Cancel



**Confirm to Rollback** ×

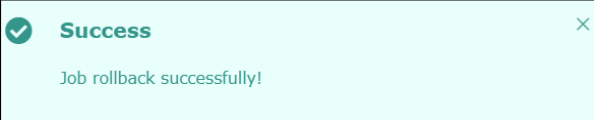
Version ID 1

Job Name 1-JOB

Comment

Cancel OK

執行 Roll Back 後之成功訊息視窗



✓ **Success** ×

Job rollback successfully!

Roll Back 後的狀態改變，後續可對 Step,Stream,Dependency,Source,Target 進行雙擊以查看其回復後的改變；亦可繼續執行 CheckIn 狀態時的相關操作。

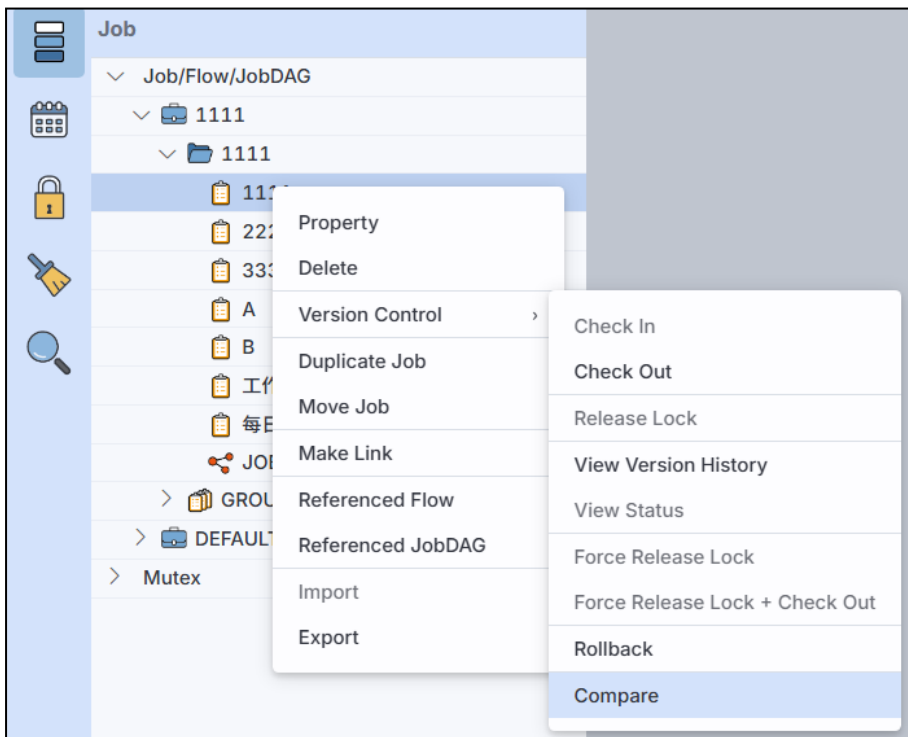
## 執行 Compare 對 Job 作版本差異比對

### 功能簡介

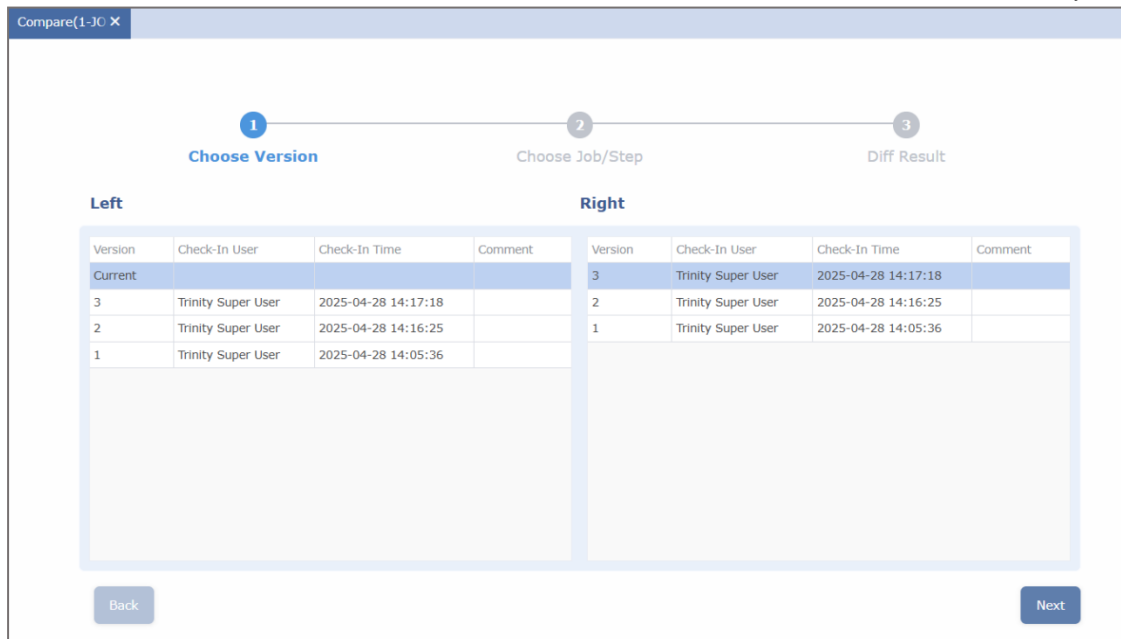
執行目的	執行發動條件	功能執行影響所及
比對 Job 版本與版本之間而後產生出差異比對	Job 狀態為 CheckIn/CheckOut	可產生出 Job 版本比對差異資訊

### 執行畫面流程

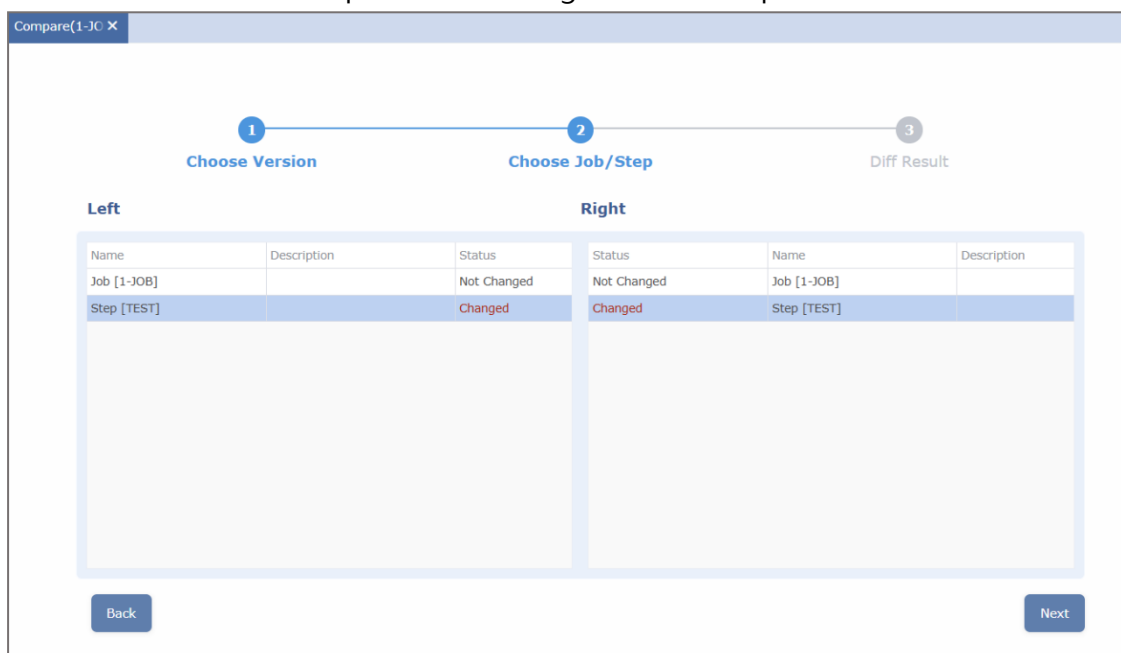
選擇要進行比對的 Job，右鍵選擇「Version Control」->「Compare」



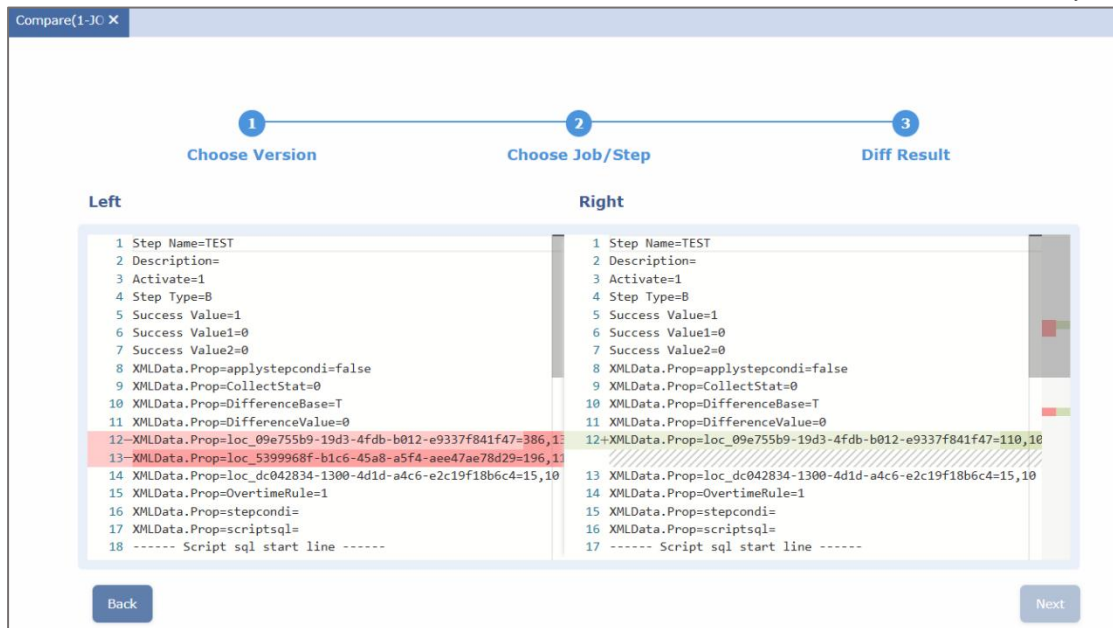
顯示 Compare 畫面，於第一步 Choose Version 選擇左右邊欲比對的版本，選擇 Next。



於第二步 Choose Job/Step，選擇有 Changed 的 Job/Step 選項，點選 Next。

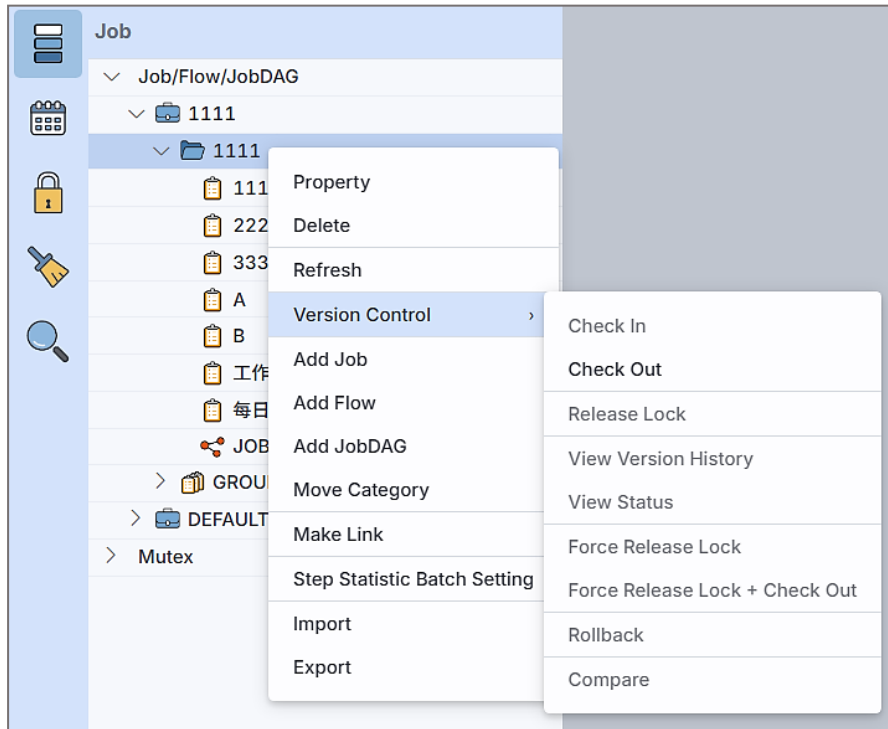


進入 Diff Result 畫面，可觀看版本間差異。



## Batch Version Control

可以對 Category 項目底下所有 Job 操作 Check in · Check out · Release lock · Force release lock。對 Category 項目按右鍵選項「Version Control」，這些按鈕都是按下去後才會檢查底下的 Job/Flow 是否有權限。



### Check In :

**Check In List** ✕

<input checked="" type="checkbox"/>	Name	Description	Type
<input checked="" type="checkbox"/>	1-JOB		Job
<input checked="" type="checkbox"/>	DEP1		Job
<input checked="" type="checkbox"/>	DEP2		Job
<input checked="" type="checkbox"/>	JOB1		Job
<input checked="" type="checkbox"/>	JOB2		Job
<input checked="" type="checkbox"/>	JOB3		Job
<input checked="" type="checkbox"/>	JOB4		Job
<input checked="" type="checkbox"/>	JOB5		Job
<input checked="" type="checkbox"/>	JOB6		Job
<input checked="" type="checkbox"/>	STREAM1		Job

Comment

在左方勾選需要 Check In 的 Job ，並填入 Comment ，按下 Submit 後即可。

**Check Out :**

**Check Out List** ×

<input checked="" type="checkbox"/>	Name	Description	Type
<input checked="" type="checkbox"/>	1-JOB		Job
<input checked="" type="checkbox"/>	DEP1		Job
<input checked="" type="checkbox"/>	DEP2		Job
<input checked="" type="checkbox"/>	JOB1		Job
<input checked="" type="checkbox"/>	JOB2		Job
<input checked="" type="checkbox"/>	JOB3		Job
<input checked="" type="checkbox"/>	JOB4		Job
<input checked="" type="checkbox"/>	JOB5		Job
<input checked="" type="checkbox"/>	JOB6		Job
<input checked="" type="checkbox"/>	STREAM1		Job

Comment

在左方勾選需要 Check Out 的 Job ，並填入 Comment ，按下 Submit 後即可。

**Release Lock :**

**Force Release Lock List** ×

<input checked="" type="checkbox"/>	Name	Description	Type
<input checked="" type="checkbox"/>	1-JOB		Job
<input checked="" type="checkbox"/>	DEP1		Job
<input checked="" type="checkbox"/>	DEP2		Job
<input checked="" type="checkbox"/>	JOB1		Job
<input checked="" type="checkbox"/>	JOB2		Job
<input checked="" type="checkbox"/>	JOB3		Job
<input checked="" type="checkbox"/>	JOB4		Job
<input checked="" type="checkbox"/>	JOB5		Job
<input checked="" type="checkbox"/>	JOB6		Job
<input checked="" type="checkbox"/>	STREAM1		Job

在左方勾選需要 Release Lock 的 Job ，按下 Submit 後即可。

## Force Release Lock :

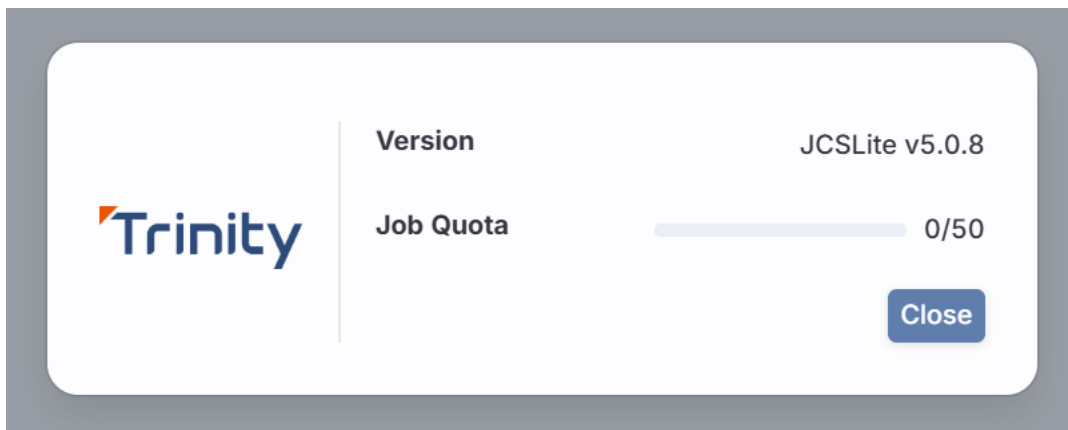
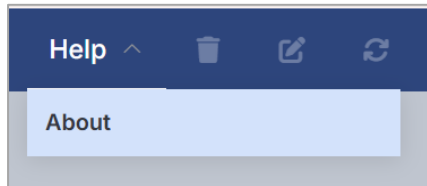
**Force Release Lock List** ×

<input checked="" type="checkbox"/>	Name	Description	Type
<input checked="" type="checkbox"/>	1-JOB		Job
<input checked="" type="checkbox"/>	DEP1		Job
<input checked="" type="checkbox"/>	DEP2		Job
<input checked="" type="checkbox"/>	JOB1		Job
<input checked="" type="checkbox"/>	JOB2		Job
<input checked="" type="checkbox"/>	JOB3		Job
<input checked="" type="checkbox"/>	JOB4		Job
<input checked="" type="checkbox"/>	JOB5		Job
<input checked="" type="checkbox"/>	JOB6		Job
<input checked="" type="checkbox"/>	STREAM1		Job

在左方勾選需要 Force Release Lock 的 Job，按下 Submit 後即可。

## 作業建立數量查詢

選擇功能選單的「Help」->「About」查看當前 Job 建立數



## 附錄 A. 工作變數的使用

我們可以在各個設定屬性中或某些類型的腳本檔(script file)中使用工作變數。使用工作變數的格式如下：\${varname}

工作變數必須是以\$為開頭並以左右大括號將變數名稱包含在內。Trinity 會在執行作業前將變數替換成實際的內容。

Trinity 中的工作變數可以分成兩類，第一類是 Job、Category、BusEntity、Domain 工作變數，第二類是系統工作變數。

Job、Category、BusEntity、Domain 工作變數是定義在 Job、Category、BusEntity、Domain 中的變數，這些變數除了可在設定屬性中引用外，還會在 External Command 的 Step 執行時，設定於環境變數中來使用。

系統工作變數則是由自動機制自動產生的變數。

變數名稱	說明
\${SERVER} \${TARGETHOST}	此變數的內容是用來記錄與資料庫連線的系統名稱，對應到 Target Connection 中的連線系統名稱。
\${USERID} \${TARGETUSERID}	此變數的內容是用來記錄與資料庫連線的使用者帳號，對應到 Target Connection 中的連線使用者帳號。
\${PASSWD} \${TARGETPASSWD}	此變數的內容是用來記錄與資料庫連線的使用者密碼，對應到 Target Connection 中的連線使用者密碼。
\${SERVER1} \${SRCHOST}	此變數的內容是用來記錄與資料庫連線的系統名稱，對應到 Source Connection 中的連線系統名稱。
\${USERID1} \${SRCUSERID}	此變數的內容是用來記錄與資料庫連線的使用者帳號，對應到 Source Connection 中的連線使用者帳號。
\${PASSWD1} \${SRCPASSWD}	此變數的內容是用來記錄與資料庫連線的使用者密碼，對應到 Source Connection 中的連線使用者密碼。
\${DOMAINNAME}	DOMAIN 名稱
\${ENTITYNAME}	ENTITY 名稱
\${CATEGORYNAME}	CATEGORY 名稱
\${JOBNAME}	JOB 名稱
\${STEPNAME}	STEP 名稱
\${DATAFILE}	此變數是只有當作業步驟類型是 Fastload 及 Multiload 時用來記錄要處理的資料檔案名稱。
\${TASKDATE} \${WODATE}	此變數是用來記錄當執行作業步驟時在工作聯單中所標記的工作日期，日期格式為 YYYY-MM-DD.

<code>\${TASKDATE1}</code>	此變數是用來記錄當執行作業步驟時在工作聯單中所標記的工作日期, 日期格式為 YYYYMMDD.
<code>\${WODATE1}</code>	

## 西元年工作變數

接下來的這些變數都是根據執行時的資料日期所計算出來的, 我們會在下列的表格中詳細的說明。

變數名稱	說明
<code>\${TXDATE}</code>	此變數是用來記錄當執行作業步驟時在作業任務中所標記的資料日期, 日期格式為 YYYY-MM-DD.
<code>\${TXDATE1}</code>	此變數是用來記錄當執行作業步驟時在作業任務中所標記的資料日期, 日期格式為 YYYYMMDD.
<code>\${LASTTXDATE}</code>	此變數是用來記錄所執行的作業上一次成功完成的日期, 日期格式為 YYYY-MM-DD.
<code>\${LASTTXDATE1}</code>	此變數是用來記錄所執行的作業上一次成功完成的日期, 日期格式為 YYYYMMDD.
<code>\${TXD}</code>	取得 TXDATE 的天數, 此天數指的是在當個月中的日期從 01 到 31。格式為 DD。
<code>\${TXM}</code>	取得 TXDATE 的月份數字, 格式為 MM, 從 01 到 12, 月份不足兩位數時補 0。
<code>\${TXM1}</code>	取得 TXDATE 的月份數字, 格式為 MM, 從 1 到 12, 月份不足兩位數時不補 0。
<code>\${TXMD}</code>	此變數是用來記錄當執行作業步驟時在作業任務中所標記的資料月份與日期, 日期格式為 MMDD.
<code>\${TXM-D}</code>	此變數是用來記錄當執行作業步驟時在作業任務中所標記的資料月份與日期, 日期格式為 MM-DD.
<code>\${TX2Y}</code>	取得 TXDATE 的年份數字, 格式為 YY。
<code>\${TX2YM}</code>	取得 TXDATE 的年份加月份數字, 格式為 YYMM。
<code>\${TX2Y-M}</code>	取得 TXDATE 的年份加月份數字, 格式為 YY-MM。
<code>\${TX2Y_M}</code>	取得 TXDATE 的年份加月份數字, 格式為 YY_MM。
<code>\${TX2YMB}</code>	取得 TXDATE 的當月第一天的日期, 格式為 YYMMDD。
<code>\${TX2YME}</code>	取得 TXDATE 的當月最後一天的日期, 格式為 YYMMDD。
<code>\${TX2Y-M-B}</code>	取得 TXDATE 的當月第一天的日期, 格式為 YY-MM-DD。

<code>#{TX2Y-M-E}</code>	取得 TXDATE 的當月最後一天的日期，格式為 YY-MM-DD。
<code>#{TX2Y_M_B}</code>	取得 TXDATE 的當月第一天的日期，格式為 YY_MM_DD。
<code>#{TX2Y_M_E}</code>	取得 TXDATE 的當月最後一天的日期，格式為 YY_MM_DD。
<code>#{TX2YWB}</code>	取得 TXDATE 的當星期第一天日期，格式為 YYMMDD。在這裡所得到的一个星期第一天是星期日。
<code>#{TX2YWE}</code>	取得 TXDATE 的當星期最後一天日期，格式為 YYMMDD。在這裡所得到的一个星期最後一天是星期六。
<code>#{TX2Y_W_B}</code>	取得 TXDATE 的當星期第一天日期，格式為 YY_MM_DD。在這裡所得到的一个星期第一天是星期日。
<code>#{TX2Y_W_E}</code>	取得 TXDATE 的當星期最後一天日期，格式為 YY_MM_DD。在這裡所得到的一个星期最後一天是星期六。
<code>#{TX2Y-W-B}</code>	取得 TXDATE 的當星期第一天日期，格式為 YY-MM-DD。在這裡所得到的一个星期第一天是星期日。
<code>#{TX2Y-W-E}</code>	取得 TXDATE 的當星期最後一天日期，格式為 YY-MM-DD。在這裡所得到的一个星期最後一天是星期六。
<code>#{TX4Y}</code>	取得 TXDATE 的年份數字，格式為 YYYY。
<code>#{TX4YM}</code>	取得 TXDATE 的年份加月份數字，格式為 YYYYMM。
<code>#{TX4Y-M}</code>	取得 TXDATE 的年份加月份數字，格式為 YYYY-MM。
<code>#{TX4Y_M}</code>	取得 TXDATE 的年份加月份數字，格式為 YYYY_MM。
<code>#{TX4YMB}</code>	取得 TXDATE 的當月第一天的日期，格式為 YYYYMMDD。
<code>#{TX4YME}</code>	取得 TXDATE 的當月最後一天的日期，格式為 YYYYMMDD。
<code>#{TX4Y-M-B}</code>	取得 TXDATE 的當月第一天的日期，格式為 YYYY-MM-DD。
<code>#{TX4Y-M-E}</code>	取得 TXDATE 的當月最後一天的日期，格式為 YYYY-MM-DD。
<code>#{TX4Y_M_B}</code>	取得 TXDATE 的當月第一天的日期，格式為 YYYY_MM_DD。
<code>#{TX4Y_M_E}</code>	取得 TXDATE 的當月最後一天的日期，格式為 YYYY_MM_DD。
<code>#{TX4YWB}</code>	取得 TXDATE 的當星期第一天日期，格式為 YYYYMMDD。在這裡所得到的一个星期第一天是星期日。
<code>#{TX4YWE}</code>	取得 TXDATE 的當星期最後一天日期，格式為 YYYYMMDD。在這裡所得到的一个星期最後一天是星期六。

<code>\${TX4Y-W-B}</code>	取得 TXDATE 的當星期第一天日期，格式為 YYYY-MM-DD。在這裡所得到的一個星期第一天是星期日。
<code>\${TX4Y-W-E}</code>	取得 TXDATE 的當星期最後一天日期，格式為 YYYY-MM-DD。在這裡所得到的一個星期最後一天是星期六。
<code>\${TX4Y_W_B}</code>	取得 TXDATE 的當星期第一天日期，格式為 YYYY_MM_DD。在這裡所得到的一個星期第一天是星期日。
<code>\${TX4Y_W_E}</code>	取得 TXDATE 的當星期最後一天日期，格式為 YYYY_MM_DD。在這裡所得到的一個星期最後一天是星期六。
<code>\${LASTTX2Y}</code>	取得 TXDATE 的前一年的年份，格式為兩個數字 YY。
<code>\${LASTTX4Y}</code>	取得 TXDATE 的前一年的年份，格式為四個數字 YYYY。
<code>\${LASTnnTXMD}</code>	取得 TXMD 之前的月日，其中的 nn 可以從 01 到 99，01 代表前一天，02 代表前二天... 依次類推。所得到的日期格式為 MMDD。
<code>\${LASTnnTXM}</code>	取得 TXDATE 之前的月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 MM，從 01 到 12 (月份不足兩位數時補 0)。
<code>\${LASTnnTXM1}</code>	取得 TXDATE 之前的月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 MM，從 1 到 12 (月份不足兩位數時不補 0)。
<code>\${LASTnnTXD}</code>	取得 TXDATE 的之前天數，其中的 nn 可以從 01 到 99，01 代表前一天，02 代表前二天... 依次類推。此天數指的是在當個月中的日期從 01 到 31。格式為 DD。
<code>\${LASTnnTXDATE}</code>	取得 TXDATE 之前的日期，其中的 nn 可以從 01 到 99，01 代表前一天，02 代表前二天... 依次類推。所得到的日期格式為 YYYY-MM-DD。
<code>\${LASTnnTXDATE1}</code>	取得 TXDATE 之前的日期，其中的 nn 可以從 01 到 99，01 代表前一天，02 代表前二天... 依次類推。所得到的日期格式為 YYYYMMDD。
<code>\${LASTnnTX2Y}</code>	取得 TXDATE 的之前年份，其中的 nn 可以從 01 到 99，01 代表前一年，02 代表前二年... 依次類推。格式為 YY。
<code>\${LASTnnTX2YM}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YYMM。

<code>\${LASTnnTX2Y-M}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YY-MM。
<code>\${LASTnnTX2Y_M}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YY_MM，月份不足兩位數時補 0。
<code>\${LASTnnTX2Y_M1}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YY_MM，月份不足兩位數時不補 0。
<code>\${LASTnnTX2YMB}</code>	取得 TXDATE 的之前月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月第一天，02 代表前二個月的當月第一天... 依次類推。格式為 YYMMDD。
<code>\${LASTnnTX2YME}</code>	取得 TXDATE 的之前月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月最後一天，02 代表前二個月的當月最後一天... 依次類推。格式為 YYMMDD。
<code>\${LASTnnTX2Y-M-B}</code>	取得 TXDATE 的之前月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月第一天，02 代表前二個月的當月第一天... 依次類推。格式為 YY-MM-DD。
<code>\${LASTnnTX2Y-M-E}</code>	取得 TXDATE 的之前月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月最後一天，02 代表前二個月的當月最後一天... 依次類推。格式為 YY-MM-DD。
<code>\${LASTnnTX2Y_M_B}</code>	取得 TXDATE 的之前月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月第一天，02 代表前二個月的當月第一天... 依次類推。格式為 YYYY_MM_DD。
<code>\${LASTnnTX2Y_M_E}</code>	取得 TXDATE 的之前月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月最後一天，02 代表前二個月的當月最後一天... 依次類推。格式為 YY_MM_DD。
<code>\${LASTnnTX2Y_W_B}</code>	取得 TXDATE 的之前星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期第一天，02 代表前二個星期的當星期第一天... 依次類推。格式為 YY_MM_DD。在這裡所得到的一個星期第一天是星期日。
<code>\${LASTnnTX2Y_W_E}</code>	取得 TXDATE 的之前星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期最後一天，02 代表前二個星期的當星期最後一天... 依次類推。格式為 YY_MM_DD。在這裡所得到的一個星期最後一天是星期六。

<code>\${LASTnnTX2YWB}</code>	取得 TXDATE 的之前星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期第一天，02 代表前二個星期的當星期第一天... 依次類推。格式為 YYMMDD。在這裡所得到的一個星期第一天是星期日。
<code>\${LASTnnTX2YWE}</code>	取得 TXDATE 的之前星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期最後一天，02 代表前二個星期的當星期最後一天... 依次類推。格式為 YYMMDD。在這裡所得到的一個星期最後一天是星期六。
<code>\${LASTnnTX2Y-W-B}</code>	取得 TXDATE 的之前星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期第一天，02 代表前二個星期的當星期第一天... 依次類推。格式為 YY-MM-DD。在這裡所得到的一個星期第一天是星期日。
<code>\${LASTnnTX2Y-W-E}</code>	取得 TXDATE 的之前星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期最後一天，02 代表前二個星期的當星期最後一天... 依次類推。格式為 YY-MM-DD。在這裡所得到的一個星期最後一天是星期六。
<code>\${LASTnnTX4Y}</code>	取得 TXDATE 的之前年份，其中的 nn 可以從 01 到 99，01 代表前一年，02 代表前二年... 依次類推。格式為 YYYY。
<code>\${LASTnnTX4YM}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YYYYMM。
<code>\${LASTnnTX4Y-M}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YYYY-MM。
<code>\${LASTnnTX4Y_M}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YYYY_MM，月份不足兩位數時補 0。
<code>\${LASTnnTX4Y_M1}</code>	取得 TXDATE 的之前月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表前一個月，02 代表前二個月... 依次類推。格式為 YYYY_MM，月份不足兩位數時不補 0。
<code>\${LASTnnTX4YMB}</code>	取得 TXDATE 的之前月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月第一天，02 代表前二個月的當月第一天... 依次類推。格式為 YYYYMMDD。

\${LASTnnTX4YME}	取得 TXDATE 的之前月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月最後一天，02 代表前二個月的當月最後一天... 依次類推。格式為 YYYYMMDD。
\${LASTnnTX4Y-M-B}	取得 TXDATE 的之前月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月第一天，02 代表前二個月的當月第一天... 依次類推。格式為 YYYY-MM-DD。
\${LASTnnTX4Y-M-E}	取得 TXDATE 的之前月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月最後一天，02 代表前二個月的當月最後一天... 依次類推。格式為 YYYY-MM-DD。
\${LASTnnTX4Y_M_B}	取得 TXDATE 的之前月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月第一天，02 代表前二個月的當月第一天... 依次類推。格式為 YYYY_MM_DD。
\${LASTnnTX4Y_M_E}	取得 TXDATE 的之前月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個月的當月最後一天，02 代表前二個月的當月最後一天... 依次類推。格式為 YYYY_MM_DD。
\${LASTnnTX4YWB}	取得 TXDATE 的之前星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期第一天，02 代表前二個星期的當星期第一天... 依次類推。格式為 YYYYMMDD。在這裡所得到的一個星期第一天是星期日。
\${LASTnnTX4YWE}	取得 TXDATE 的之前星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期最後一天，02 代表前二個星期的當星期最後一天... 依次類推。格式為 YYYYMMDD。在這裡所得到的一個星期最後一天是星期六。
\${LASTnnTX4Y-W-B}	取得 TXDATE 的之前星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期第一天，02 代表前二個星期的當星期第一天... 依次類推。格式為 YYYY-MM-DD。在這裡所得到的一個星期第一天是星期日。
\${LASTnnTX4Y-W-E}	取得 TXDATE 的之前星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期最後一天，02 代表前二個星期的當星期最後一天... 依次類推。格式為 YYYY-MM-DD。在這裡所得到的一個星期最後一天是星期六。
\${LASTnnTX4Y_W_B}	取得 TXDATE 的之前星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期第一天，02 代

	表前二個星期的當星期第一天... 依次類推。格式為 YYYY_MM_DD。在這裡所得到的一个星期第一天是星期日。
<code>\${LASTnnTX4Y_W_E}</code>	取得 TXDATE 的之前星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表前一個星期的當星期最後一天，02 代表前二個星期的當星期最後一天... 依次類推。格式為 YYYY_MM_DD。在這裡所得到的一个星期最後一天是星期六。
<code>\${NEXTnnTXDATE}</code>	取得 TXDATE 之後的日期，其中的 nn 可以從 01 到 99，01 代表後一天，02 代表後二天... 依次類推。所得到的日期格式為 YYYY-MM-DD。
<code>\${NEXTnnTXDATE1}</code>	取得 TXDATE 之後的日期，其中的 nn 可以從 01 到 99，01 代表後一天，02 代表後二天... 依次類推。所得到的日期格式為 YYYYMMDD。
<code>\${NEXTnnTX2Y}</code>	取得 TXDATE 的之後年份，其中的 nn 可以從 01 到 99，01 代表後一年，02 代表後二年... 依次類推，格式為 YY。
<code>\${NEXTnnTX4Y}</code>	取得 TXDATE 的之後年份，其中的 nn 可以從 01 到 99，01 代表後一年，02 代表後二年... 依次類推，格式為 YYYY。
<code>\${NEXTnnTXM}</code>	取得 TXDATE 之後的月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 MM，從 01 到 12。
<code>\${NEXTnnTXMD}</code>	取得 TXMD 之後的日期，其中的 nn 可以從 01 到 99，01 代表後一天，02 代表後二天... 依次類推。所得到的日期格式為 MMDD。
<code>\${NEXTnnTXD}</code>	取得 TXDATE 的之後天數，其中的 nn 可以從 01 到 99，01 代表後一天，02 代表後二天... 依次類推。此天數指的是在當個月中的日期從 01 到 31。格式為 DD。
<code>\${NEXTnnTX2YM}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YYMM。
<code>\${NEXTnnTX2Y-M}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YY-MM。
<code>\${NEXTnnTX2Y_M}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YY_MM，月份不足兩位數時補 0。

\${NEXTnnTX2Y_M1}	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YY_MM，月份不足兩位數時不補 0。
\${NEXTnnTX2YMB}	取得 TXDATE 的之後月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月第一天，02 代表後二個月的當月第一天... 依次類推。格式為 YYYYMMDD。
\${NEXTnnTX2YME}	取得 TXDATE 的之後月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月最後一天，02 代表後二個月的當月最後一天... 依次類推。格式為 YYYYMMDD。
\${NEXTnnTX2Y-M-B}	取得 TXDATE 的之後月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月第一天，02 代表後二個月的當月第一天... 依次類推。格式為 YY-MM-DD。
\${NEXTnnTX2Y-M-E}	取得 TXDATE 的之後月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月最後一天，02 代表後二個月的當月最後一天... 依次類推。格式為 YY-MM-DD。
\${NEXTnnTX2Y_M_B}	取得 TXDATE 的之後月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月第一天，02 代表後二個月的當月第一天... 依次類推。格式為 YY_MM_DD。
\${NEXTnnTX2Y_M_E}	取得 TXDATE 的之後月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月最後一天，02 代表後二個月的當月最後一天... 依次類推。格式為 YY_MM_DD。
\${NEXTnnTX2YWb}	取得 TXDATE 的之後星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期第一天，02 代表後二個星期的當星期第一天... 依次類推。格式為 YYYYMMDD。在這裡所得到的一個星期第一天是星期日。
\${NEXTnnTX2YWE}	取得 TXDATE 的之後星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期最後一天，02 代表後二個星期的當星期最後一天... 依次類推。格式為 YYYYMMDD。在這裡所得到的一個星期最後一天是星期六。
\${NEXTnnTX2Y-W-B}	取得 TXDATE 的之後星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期第一天，02 代表後二個星期的當星期第一天... 依次類推。格式為 YY-MM-DD。在這裡所得到的一個星期第一天是星期日。
\${NEXTnnTX2Y-W-E}	取得 TXDATE 的之後星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期最後一天，02

	代表後二個星期的當星期最後一天... 依次類推。格式為 YY-MM-DD。在這裡所得到的一个星期最後一天是星期六。
<code>\${NEXTnnTX2Y_W_B}</code>	取得 TXDATE 的之後星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期第一天，02 代表後二個星期的當星期第一天... 依次類推。格式為 YY_MM_DD。在這裡所得到的一个星期第一天是星期日。
<code>\${NEXTnnTX2Y_W_E}</code>	取得 TXDATE 的之後星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期最後一天，02 代表後二個星期的當星期最後一天... 依次類推。格式為 YY_MM_DD。在這裡所得到的一个星期最後一天是星期六。
<code>\${NEXTnnTX4YM}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YYYYMM。
<code>\${NEXTnnTX4Y-M}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YYYY-MM。
<code>\${NEXTnnTX4Y_M}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YYYY_MM，月份不足兩位數時補 0。
<code>\${NEXTnnTX4Y_M1}</code>	取得 TXDATE 的之後月份的年份加月份數字，其中的 nn 可以從 01 到 99，01 代表後一個月，02 代表後二個月... 依次類推。格式為 YYYY_MM，月份不足兩位數時不補 0。
<code>\${NEXTnnTX4YMB}</code>	取得 TXDATE 的之後月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月第一天，02 代表後二個月的當月第一天... 依次類推。格式為 YYYYMMDD。
<code>\${NEXTnnTX4YME}</code>	取得 TXDATE 的之後月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月最後一天，02 代表後二個月的當月最後一天... 依次類推。格式為 YYYYMMDD。
<code>\${NEXTnnTX4Y-M-B}</code>	取得 TXDATE 的之後月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月第一天，02 代表後二個月的當月第一天... 依次類推。格式為 YYYY-MM-DD。
<code>\${NEXTnnTX4Y-M-E}</code>	取得 TXDATE 的之後月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月最後一天，02 代表後二個月的當月最後一天... 依次類推。格式為 YYYY-MM-DD。

<code>\${NEXTnnTX4Y_M_B}</code>	取得 TXDATE 的之後月份的當月第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月第一天，02 代表後二個月的當月第一天... 依次類推。格式為 YYYY_MM_DD。
<code>\${NEXTnnTX4Y_M_E}</code>	取得 TXDATE 的之後月份的當月最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個月的當月最後一天，02 代表後二個月的當月最後一天... 依次類推。格式為 YYYY_MM_DD。
<code>\${NEXTnnTX4YWB}</code>	取得 TXDATE 的之後星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期第一天，02 代表後二個星期的當星期第一天... 依次類推。格式為 YYYYMMDD。在這裡所得到的一個星期第一天是星期日。
<code>\${NEXTnnTX4YWE}</code>	取得 TXDATE 的之後星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期最後一天，02 代表後二個星期的當星期最後一天... 依次類推。格式為 YYYYMMDD。在這裡所得到的一個星期最後一天是星期六。
<code>\${NEXTnnTX4Y-W-B}</code>	取得 TXDATE 的之後星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期第一天，02 代表後二個星期的當星期第一天... 依次類推。格式為 YYYY-MM-DD。在這裡所得到的一個星期第一天是星期日。
<code>\${NEXTnnTX4Y-W-E}</code>	取得 TXDATE 的之後星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期最後一天，02 代表後二個星期的當星期最後一天... 依次類推。格式為 YYYY-MM-DD。在這裡所得到的一個星期最後一天是星期六。
<code>\${NEXTnnTX4Y_W_B}</code>	取得 TXDATE 的之後星期的當星期第一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期第一天，02 代表後二個星期的當星期第一天... 依次類推。格式為 YYYY_MM_DD。在這裡所得到的一個星期第一天是星期日。
<code>\${NEXTnnTX4Y_W_E}</code>	取得 TXDATE 的之後星期的當星期最後一天日期，其中的 nn 可以從 01 到 99，01 代表後一個星期的當星期最後一天，02 代表後二個星期的當星期最後一天... 依次類推。格式為 YYYY_MM_DD。在這裡所得到的一個星期最後一天是星期六。

## 民國年工作變數

對應如下表

變數名稱(西元年)	變數名稱(民國年)	說明
\${TXDATE}	\${CYTXDATE}	功能與西元年變數相同，差異僅在民國年變數以 CYTXDATE 轉換，年份格式為 YYY。
\${TXDATE1}	\${CYTXDATE1}	
\${LASTTXDATE}	\${LASTCYTXDATE}	
\${LASTTXDATE1}	\${LASTCYTXDATE1}	
\${TXD}	\${CYTXD}	
\${TXM}	\${CYTXM}	
\${TXM1}	\${CYTXM1}	
\${TXMD}	\${CYTXMD}	
\${TXM-D}	\${CYTXM-D}	
\${TX2Y}	(無對應)	
\${TX2YM}	(無對應)	
\${TX2Y-M}	(無對應)	
\${TX2Y_M}	(無對應)	
\${TX2YMB}	(無對應)	
\${TX2YME}	(無對應)	
\${TX2Y-M-B}	(無對應)	
\${TX2Y-M-E}	(無對應)	
\${TX2Y_M_B}	(無對應)	
\${TX2Y_M_E}	(無對應)	
\${TX2YWB}	(無對應)	
\${TX2YWE}	(無對應)	
\${TX2Y_W_B}	(無對應)	
\${TX2Y_W_E}	(無對應)	
\${TX2Y-W-B}	(無對應)	
\${TX2Y-W-E}	(無對應)	
\${TX4Y}	\${CYTXY}	
\${TX4YM}	\${CYTXYM}	
\${TX4Y-M}	\${CYTXY-M}	
\${TX4Y_M}	\${CYTXY_M}	
\${TX4YMB}	\${CYTXYMB}	
\${TX4YME}	\${CYTXYME}	

\${TX4Y-M-B}	\${CYTXY-M-B}
\${TX4Y-M-E}	\${CYTXY-M-E}
\${TX4Y_M_B}	\${CYTXY_M_B}
\${TX4Y_M_E}	\${CYTXY_M_E}
\${TX4YWB}	\${CYTXYWB}
\${TX4YWE}	\${CYTXYWE}
\${TX4Y-W-B}	\${CYTXY-W-B}
\${TX4Y-W-E}	\${CYTXY-W-E}
\${TX4Y_W_B}	\${CYTXY_W_B}
\${TX4Y_W_E}	\${CYTXY_W_E}
\${LASTTX2Y}	(無對應)
\${LASTTX4Y}	\${LASTCYTXY}
\${LASTnnTXMD}	\${LASTnnCYTXMD}
\${LASTnnTXM}	\${LASTnnCYTXM}
\${LASTnnTXM1}	\${LASTnnCYTXM1}
\${LASTnnTXD}	\${LASTnnCYTXD}
\${LASTnnTXDATE}	\${LASTnnCYTXDATE}
\${LASTnnTXDATE1}	\${LASTnnCYTXDATE1}
\${LASTnnTX2Y}	(無對應)
\${LASTnnTX2YM}	(無對應)
\${LASTnnTX2Y-M}	(無對應)
\${LASTnnTX2Y_M}	(無對應)
\${LASTnnTX2Y_M1}	(無對應)
\${LASTnnTX2YMB}	(無對應)
\${LASTnnTX2YME}	(無對應)
\${LASTnnTX2Y-M-B}	(無對應)
\${LASTnnTX2Y-M-E}	(無對應)

$\${LASTnnTX2Y\_M\_B}$	(無對應)	
$\${LASTnnTX2Y\_M\_E}$	(無對應)	
$\${LASTnnTX2Y\_W\_B}$	(無對應)	
$\${LASTnnTX2Y\_W\_E}$	(無對應)	
$\${LASTnnTX2YWB}$	(無對應)	
$\${LASTnnTX2YWE}$	(無對應)	
$\${LASTnnTX2Y-W-B}$	(無對應)	
$\${LASTnnTX2Y-W-E}$	(無對應)	
$\${LASTnnTX4Y}$	$\${LASTnnCYTXY}$	
$\${LASTnnTX4YM}$	$\${LASTnnCYTXYM}$	
$\${LASTnnTX4Y-M}$	$\${LASTnnCYTXY-M}$	
$\${LASTnnTX4Y\_M}$	$\${LASTnnCYTXY\_M}$	
$\${LASTnnTX4Y\_M1}$	$\${LASTnnCYTXY\_M1}$	
$\${LASTnnTX4YMB}$	$\${LASTnnCYTXYMB}$	
$\${LASTnnTX4YME}$	$\${LASTnnCYTXYME}$	
$\${LASTnnTX4Y-M-B}$	$\${LASTnnCYTXY-M-B}$	
$\${LASTnnTX4Y-M-E}$	$\${LASTnnCYTXY-M-E}$	
$\${LASTnnTX4Y\_M\_B}$	$\${LASTnnCYTXY\_M\_B}$	
$\${LASTnnTX4Y\_M\_E}$	$\${LASTnnCYTXY\_M\_E}$	
$\${LASTnnTX4YWB}$	$\${LASTnnCYTXYWB}$	
$\${LASTnnTX4YWE}$	$\${LASTnnCYTXYWE}$	
$\${LASTnnTX4Y-W-B}$	$\${LASTnnCYTXY-W-B}$	

<code>\${LASTnnTX4Y-W-E}</code>	<code>\${LASTnnCYTXY-W-E}</code>
<code>\${LASTnnTX4Y_W_B}</code>	<code>\${LASTnnCYTXY_W_B}</code>
<code>\${LASTnnTX4Y_W_E}</code>	<code>\${LASTnnCYTXY_W_E}</code>
<code>\${NEXTnnTXDATE}</code>	<code>\${NEXTnnCYTXDATE}</code>
<code>\${NEXTnnTXDATE1}</code>	<code>\${NEXTnnCYTXDATE1}</code>
<code>\${NEXTnnTX2Y}</code>	(無對應)
<code>\${NEXTnnTX4Y}</code>	<code>\${NEXTnnCYTXY}</code>
<code>\${NEXTnnTXM}</code>	<code>\${NEXTnnCYTXM}</code>
<code>\${NEXTnnTXMD}</code>	<code>\${NEXTnnCYTXMD}</code>
<code>\${NEXTnnTXD}</code>	<code>\${NEXTnnCYTXD}</code>
<code>\${NEXTnnTX2YM}</code>	(無對應)
<code>\${NEXTnnTX2Y-M}</code>	(無對應)
<code>\${NEXTnnTX2Y_M}</code>	(無對應)
<code>\${NEXTnnTX2Y_M1}</code>	(無對應)
<code>\${NEXTnnTX2YMB}</code>	(無對應)
<code>\${NEXTnnTX2YME}</code>	(無對應)
<code>\${NEXTnnTX2Y-M-B}</code>	(無對應)
<code>\${NEXTnnTX2Y-M-E}</code>	(無對應)
<code>\${NEXTnnTX2Y_M_B}</code>	(無對應)
<code>\${NEXTnnTX2Y_M_E}</code>	(無對應)
<code>\${NEXTnnTX2YWB}</code>	(無對應)
<code>\${NEXTnnTX2YWE}</code>	(無對應)

$\{\text{NEXTnnTX2Y-W-B}\}$	(無對應)	
$\{\text{NEXTnnTX2Y-W-E}\}$	(無對應)	
$\{\text{NEXTnnTX2Y\_W\_B}\}$	(無對應)	
$\{\text{NEXTnnTX2Y\_W\_E}\}$	(無對應)	
$\{\text{NEXTnnTX4YM}\}$	$\{\text{NEXTnnCYTXYM}\}$	
$\{\text{NEXTnnTX4Y-M}\}$	$\{\text{NEXTnnCYTXY-M}\}$	
$\{\text{NEXTnnTX4Y\_M}\}$	$\{\text{NEXTnnCYTXY\_M}\}$	
$\{\text{NEXTnnTX4Y\_M1}\}$	$\{\text{NEXTnnCYTXY\_M1}\}$	
$\{\text{NEXTnnTX4YMB}\}$	$\{\text{NEXTnnCYTXYMB}\}$	
$\{\text{NEXTnnTX4YME}\}$	$\{\text{NEXTnnCYTXYME}\}$	
$\{\text{NEXTnnTX4Y-M-B}\}$	$\{\text{NEXTnnCYTXY-M-B}\}$	
$\{\text{NEXTnnTX4Y-M-E}\}$	$\{\text{NEXTnnCYTXY-M-E}\}$	
$\{\text{NEXTnnTX4Y\_M\_B}\}$	$\{\text{NEXTnnCYTXY\_M\_B}\}$	
$\{\text{NEXTnnTX4Y\_M\_E}\}$	$\{\text{NEXTnnCYTXY\_M\_E}\}$	
$\{\text{NEXTnnTX4YWB}\}$	$\{\text{NEXTnnCYTXYWB}\}$	
$\{\text{NEXTnnTX4YWE}\}$	$\{\text{NEXTnnCYTXYWE}\}$	
$\{\text{NEXTnnTX4Y-W-B}\}$	$\{\text{NEXTnnCYTXY-W-B}\}$	
$\{\text{NEXTnnTX4Y-W-E}\}$	$\{\text{NEXTnnCYTXY-W-E}\}$	
$\{\text{NEXTnnTX4Y\_W\_B}\}$	$\{\text{NEXTnnCYTXY\_W\_B}\}$	
$\{\text{NEXTnnTX4Y\_W\_E}\}$	$\{\text{NEXTnnCYTXY\_W\_E}\}$	

## 時間工作變數

對應如下表:

變數名稱	說明
<code>\${TASKTIME}</code>	此變數是用來記錄作業任務中所標記的任務開始執行時間, 格式為 HH:MM:00.
<code>\${TASKTIME1}</code>	此變數是用來記錄作業任務中所標記的任務開始執行時間, 格式為 HHMM00.