

單元4：

叢集(Cluster)

主題：

- a. 介紹叢集(Cluster)
- b. 使用叢集(Cluster)
- c. 錯誤叢集(Error Cluster)

叢集 Clusters

- 叢集用來集合不同的元件
- 可以讓人機介面簡潔
- 可以避免在程式區中造成雜亂的接線，並且減少subVI 需要的連接器嵌板接頭的數量
- 叢集若非控制器，就是指示器。叢集中不能混合容納控制器及指示器



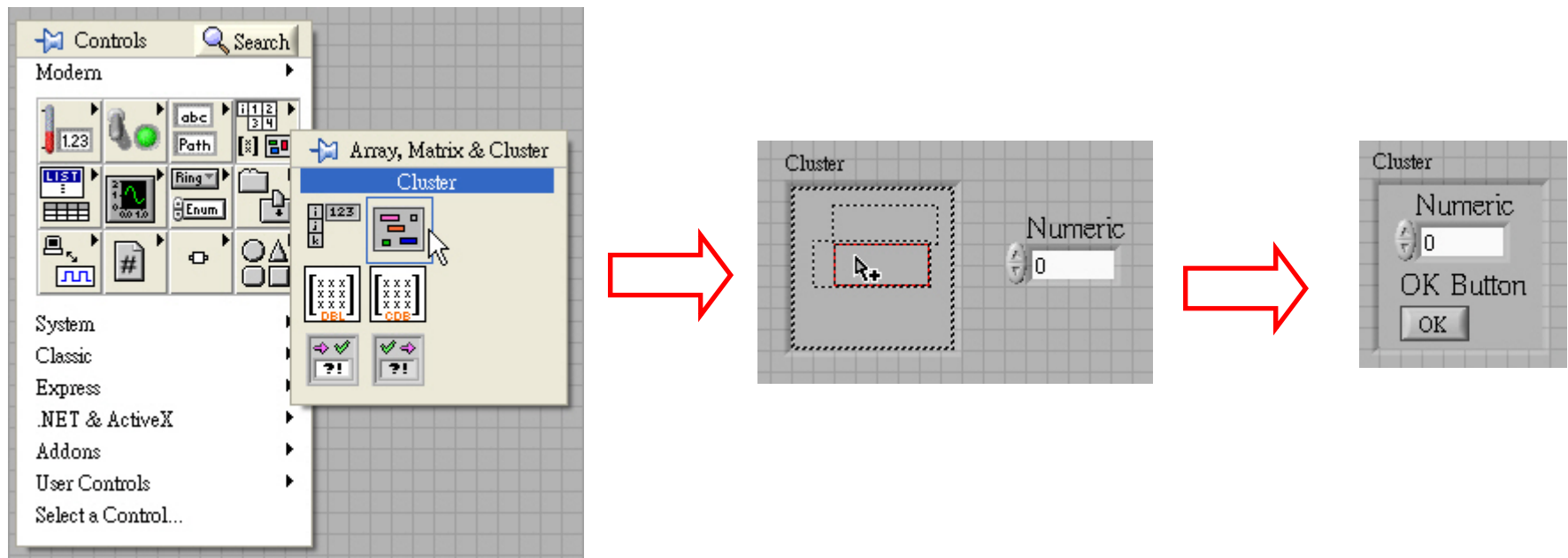
叢集 Cluster


- n. (名詞 noun)[C]
 - (葡萄等的)串, (花等的)束, 簇[(+of)]
 - (人, 動物等的)群, 組[(+of)]
- vt. (及物動詞 transitive verb)
 - 使成簇(或群)[(+together)]
 - 群集在
- vi. (不及物動詞 intransitive verb)
 - 叢生; 群集[(+around/round)]



建立Cluster的控制器及指示器

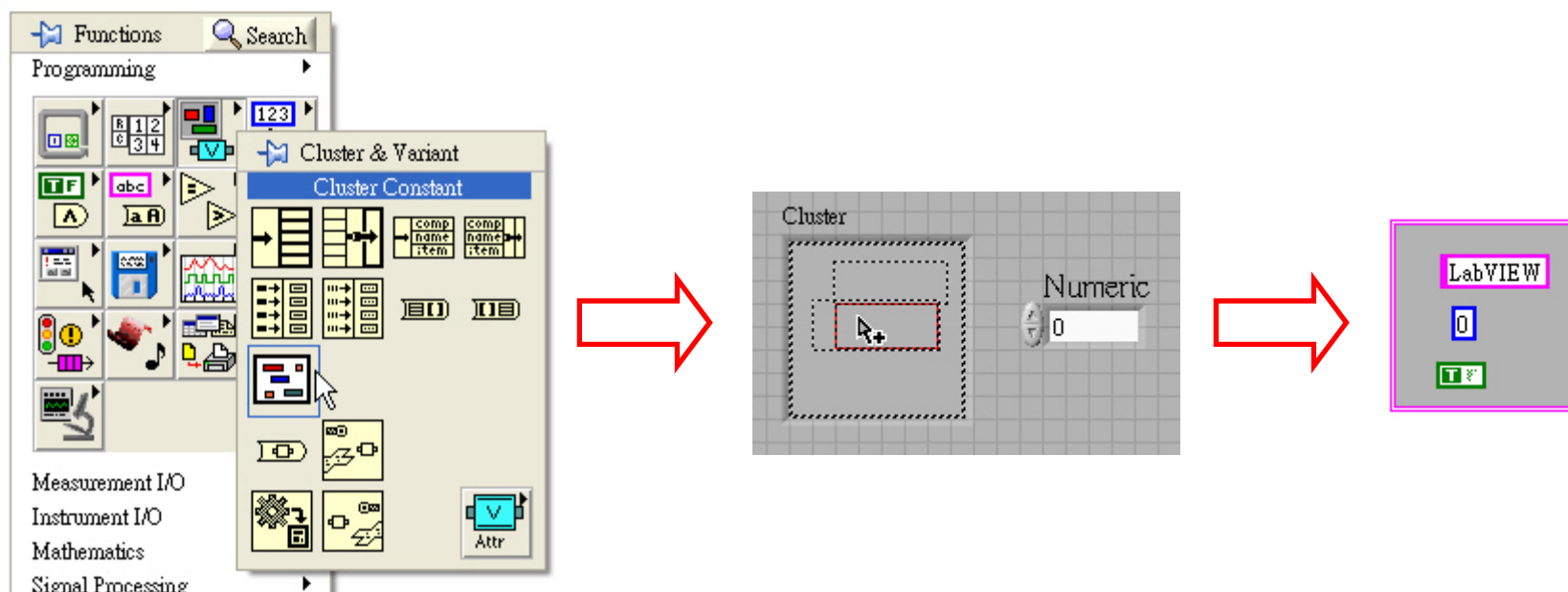
1. 從Array & Cluster面板，拖曳出Cluster
2. 把控制器拖曳進Cluster



 注意：Cluster內，仍可放置其他Cluster

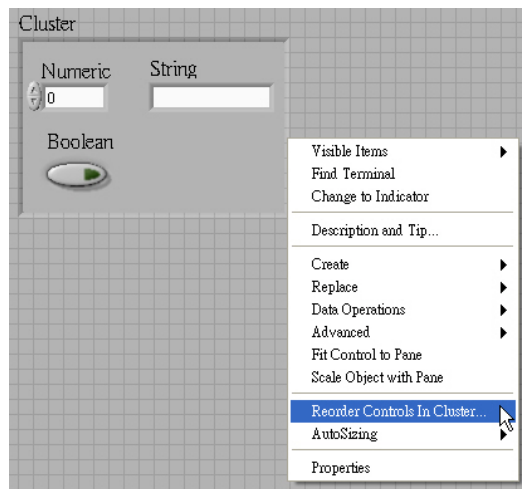
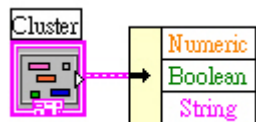
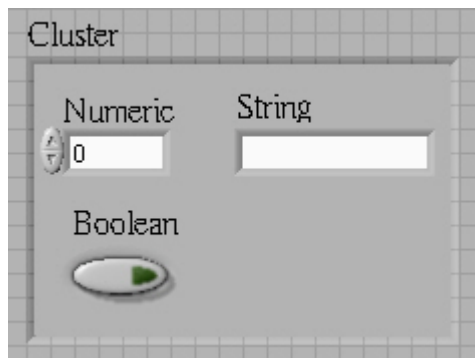
建立常數Cluster

1. 從「**Functions» Cluster & Variant» Cluster Constant**」拖曳出Cluster Constant
2. 把物件放在Cluster Constant裡面

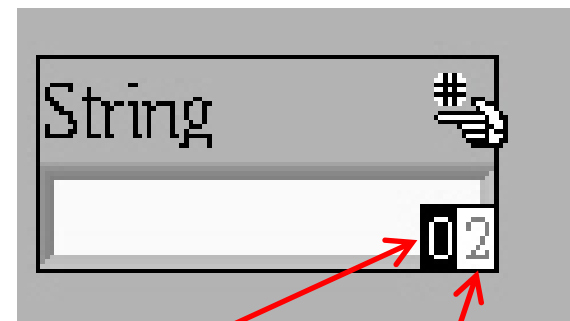
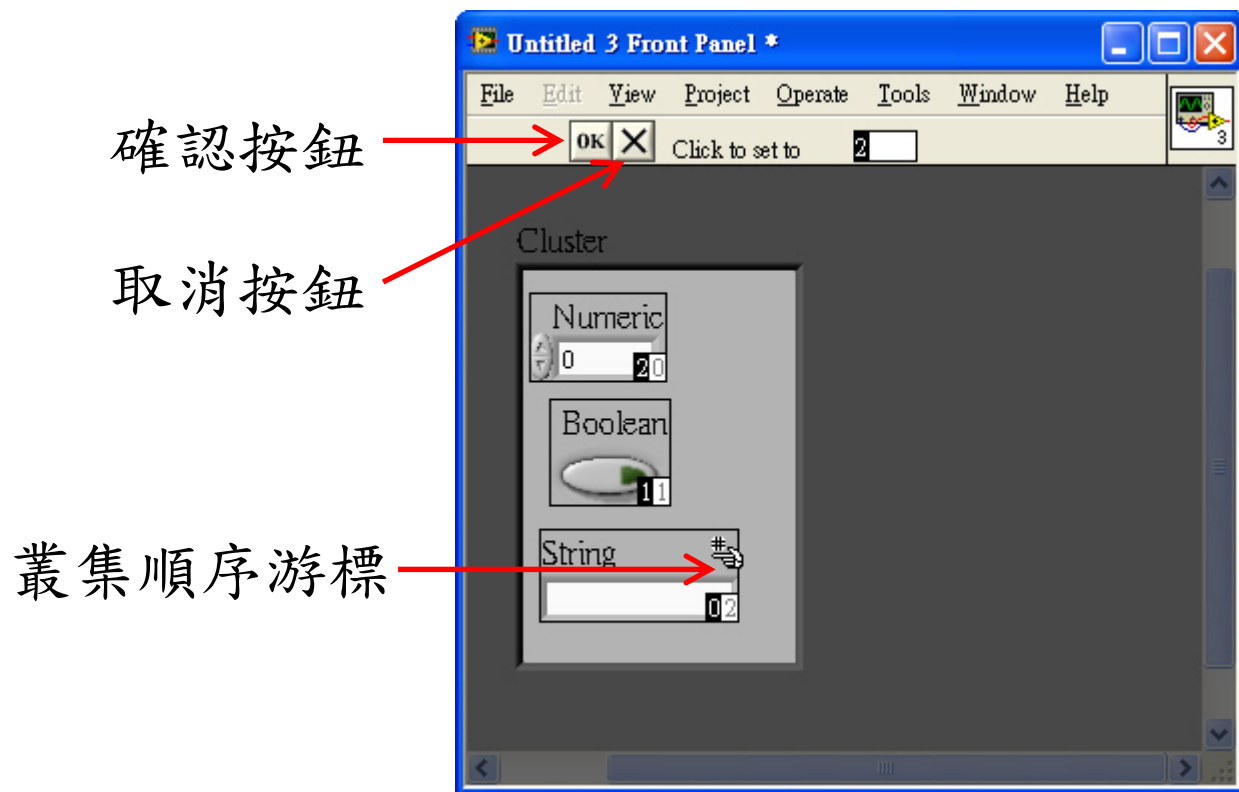


叢集順序

- 在Cluster內物件的順序，會隨著放入Cluster的順序依序排列。
- 若要改變Cluster內物件的順序，在Cluster上按滑鼠右鍵，選擇「**Reorder Controls in Cluster...**」



叢集順序



新順序

舊順序

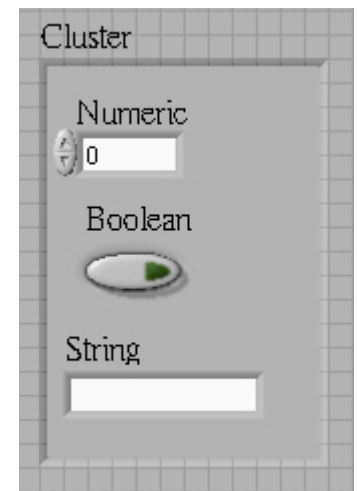
練習4.1 – 學習Cluster的使用

■ 目標：練習Cluster的使用

■ 任務：

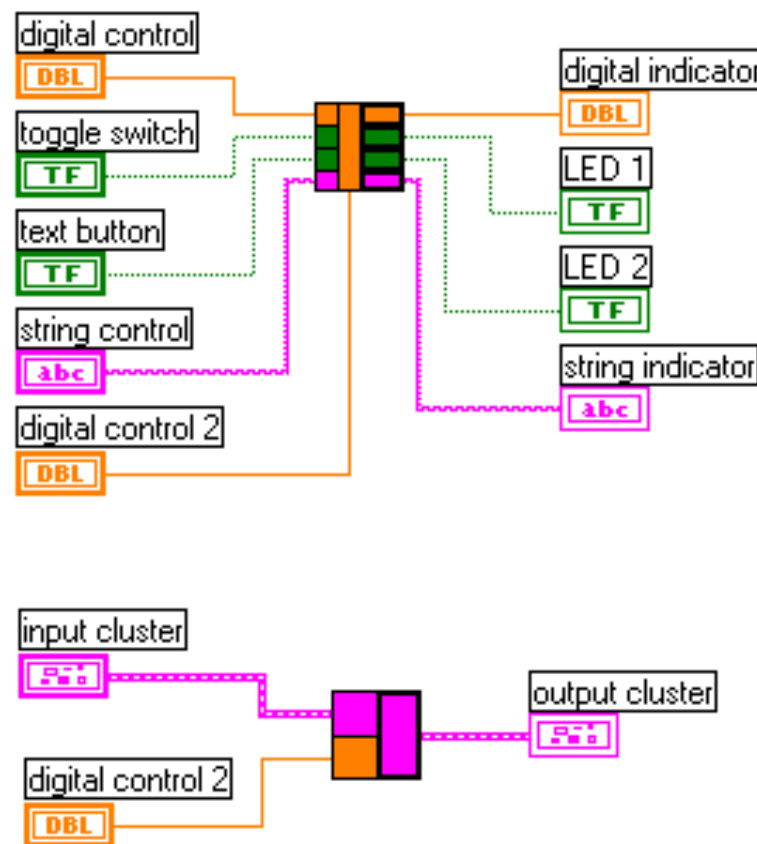
□ 做出右圖的Cluster

□ 改變Cluster內物件的順序



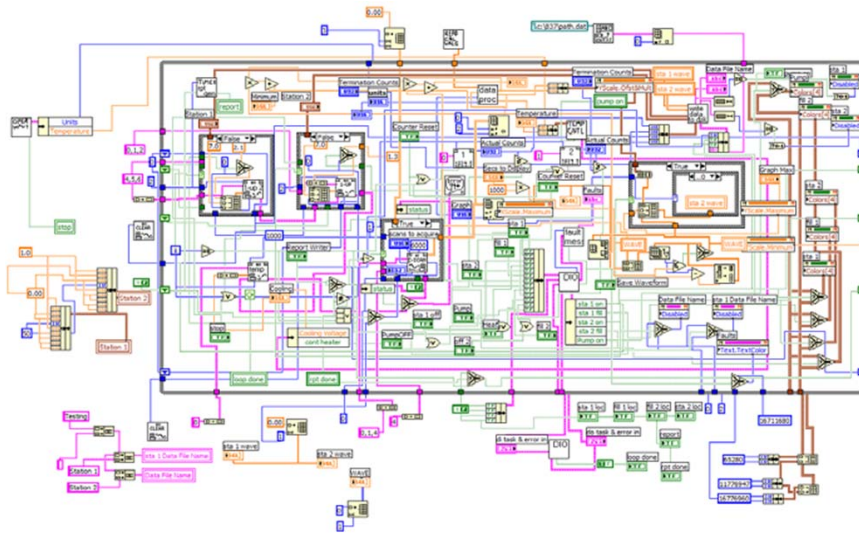
使用Cluster來傳遞資料到subVI

- 一個subVI最多只有28個資料接點。如果想從subVI傳遞超過28種資料，則可以使用Cluster，來減低資料接點的需求量
- 使用Cluster也可以簡化接線的複雜度。如右邊的兩個圖，上圖的揭露複雜，下圖的接線就相對簡單許多。

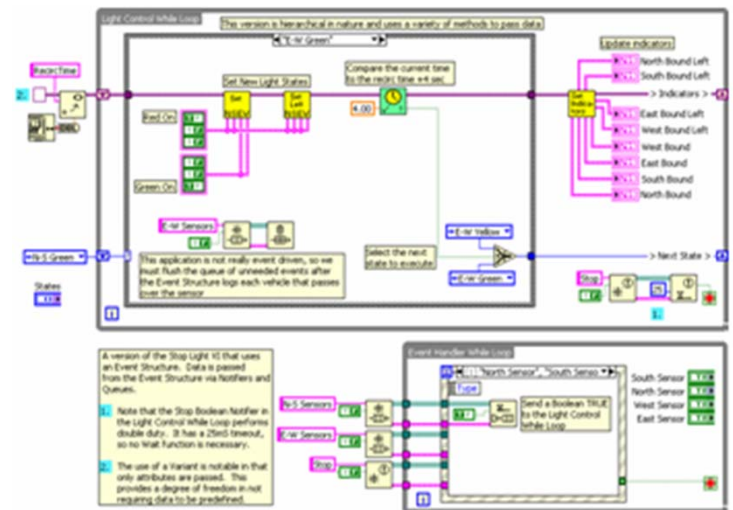


注意：避免在一個VI中指定16個以上的接頭

程式設計過程，接線簡化的效果



未經過簡化

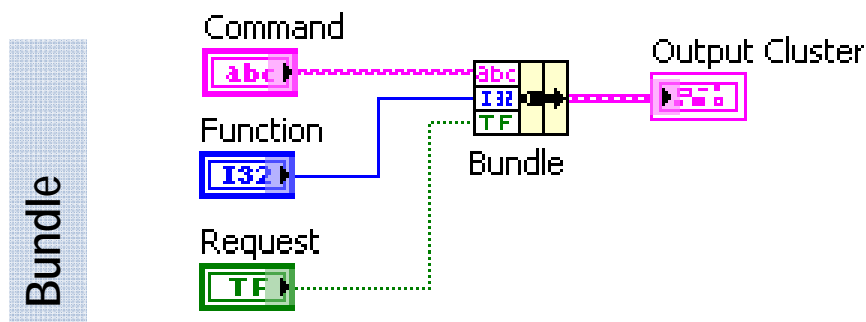


接線有經過簡化

Cluster常用功能 - Bundle



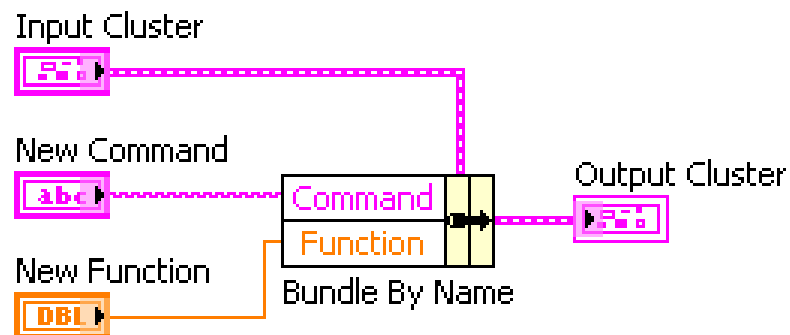
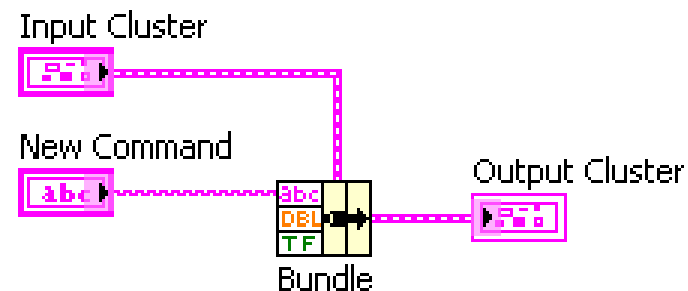
建立新的Cluster



Bundle
By Name

Must have an existing cluster to use this function.

修改現有的Cluster



Cluster常用功能 - Unbundle

Applicant Cluster

Name

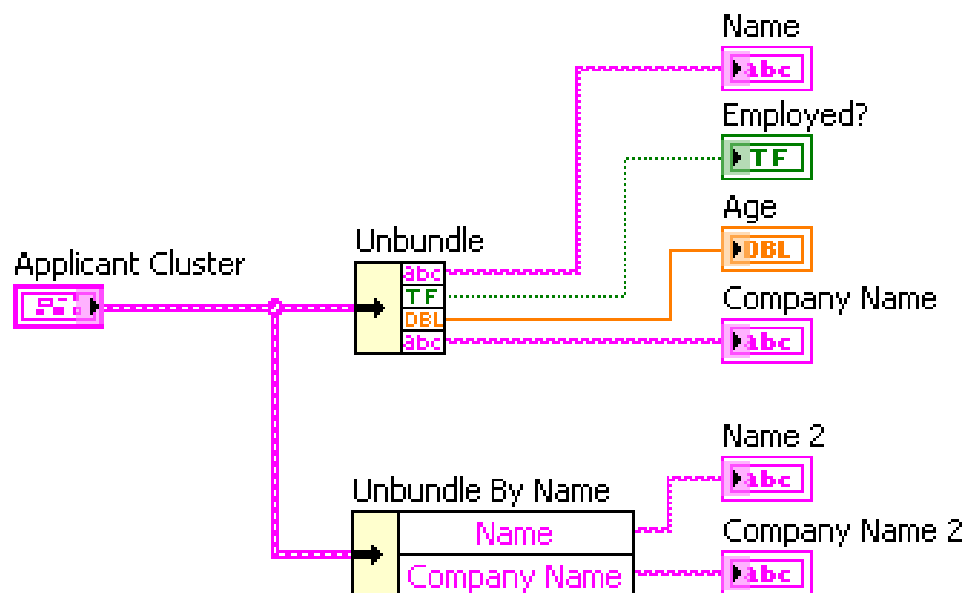
Age

Employed?

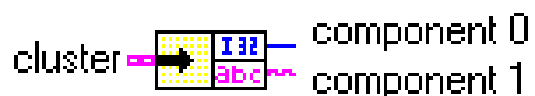
Yes ☐

No ☐

Company Name



Unbundle



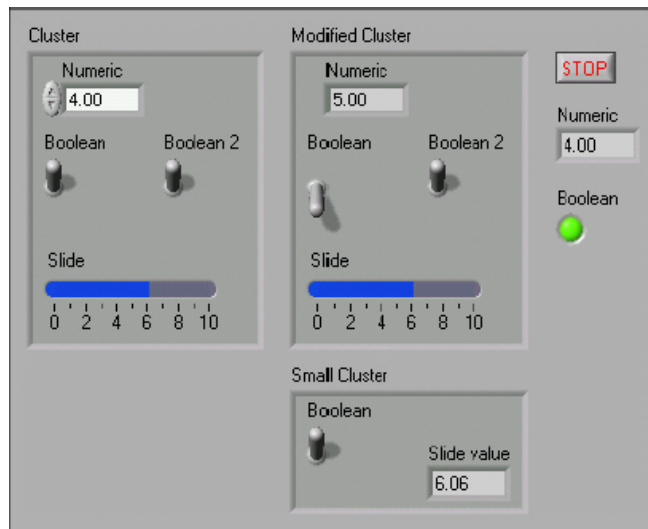
Unbundle By Name



練習4.2－叢集練習

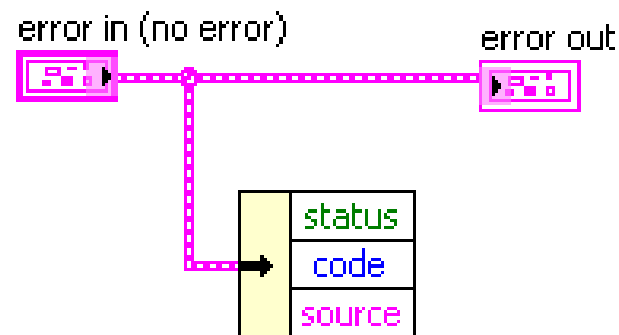
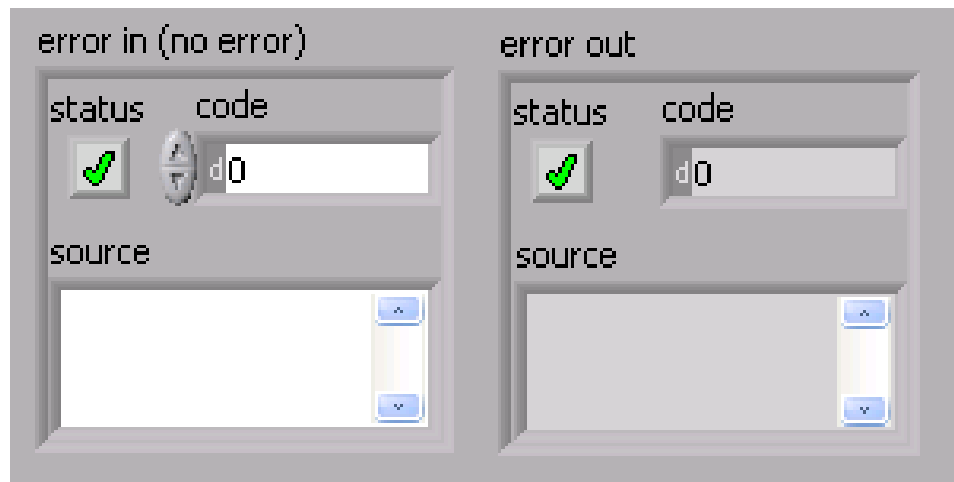
■任務：

- 建立以下的人機介面
- 確認每個元件的叢集順序

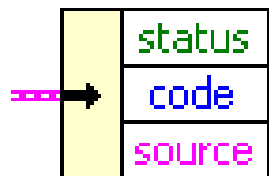


錯誤叢集 (Error Cluster)

- 習慣上，在subVI有兩個接點是負責傳入、傳出錯誤資訊的。
- 人機介面的錯誤叢集位於「**Controls»Array & Cluster**」



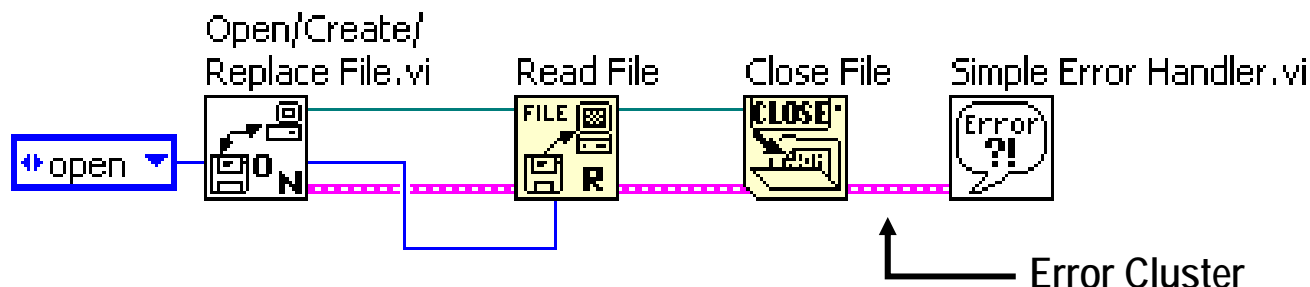
錯誤叢集的資料內容



- **Status**是布林。如果有錯誤就是True，沒有錯誤就是False。大部份接受布林值資料的VI、函數以及結構也會認得這個參數。
- **Code**是一個32 位元有正負記號整數，以數字的方式來辨識錯誤。非零代碼加上status 的值為FALSE，代表警訊，而非致命錯誤。
- **Source**是一個字串，用於辨識錯誤發生的位置。

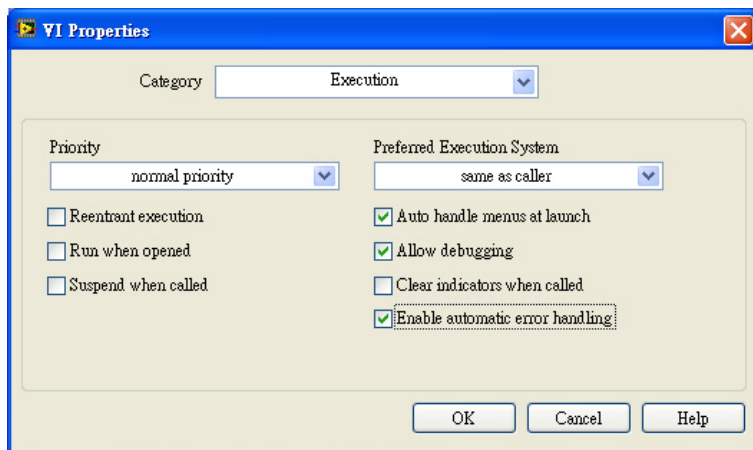
錯誤處理

- 在預設情況下，LabVIEW 會自動處理VI 執行時發生的任何錯誤，也就是暫停執行，並跳出錯誤訊息對話框
- 在LabVIEW 中，資料處理乃遵循資料流模式。正如資料流經VI 一樣，錯誤資訊也可以遵循資料流模式傳遞於subVI之中
- 當VI 執行時，LabVIEW 會在每個執行節點上測試是否有錯誤。如果LabVIEW 沒有發現任何錯誤，節點就會正常執行。如果LabVIEW 發現錯誤，該節點即將錯誤傳給下一個節點，但是不執行。

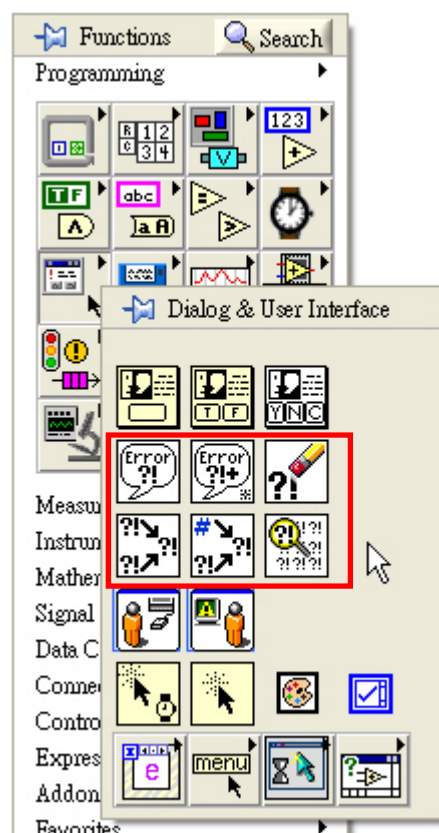


停止LabVIEW的自動錯誤處理

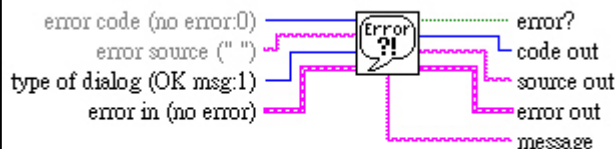
- 在預設的情況下，LabVIEW會自動處理錯誤，當Error產生，沒有資料流的導向時，LabVIEW會自動跳出視窗告訴你那邊有錯誤。
- 取消此功能：
 - ❑ **File»VI Properties**
 - ❑ 在Category下拉式選單中，選擇「Execution」
 - ❑ 把Enable automatic error handling核取方塊取消



常用的錯誤叢集函數

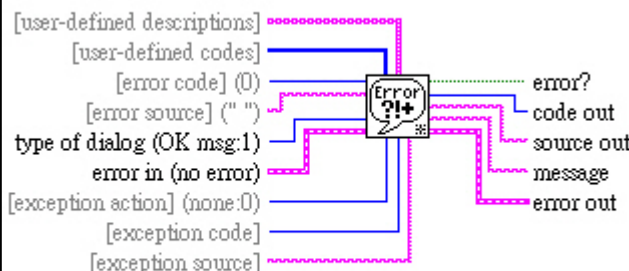


Simple Error Handler.vi



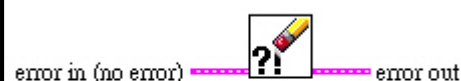
Indicates whether an error occurred. If an error occurred, this VI returns a description of the error and optionally displays a dialog box.

General Error Handler.vi



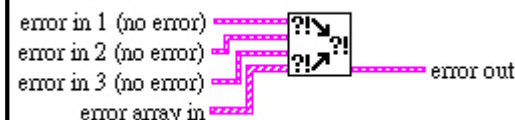
Indicates whether an error occurred. If an error occurred, this VI returns a description of the error and optionally displays a dialog box.

Clear Errors.vi



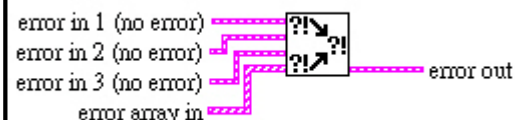
Resets the error **status** to no error, **code** to 0, and **source** to an empty string. Use this VI when you want to ignore an error.

Merge Errors.vi



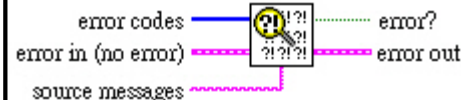
Merges error I/O clusters from different functions.

Merge Errors.vi



Merges error I/O clusters from different functions.

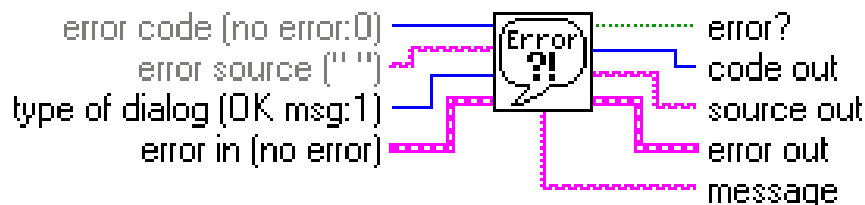
Find First Error.vi



Tests the error status of one or more low-level functions or subVIs that produce a numeric error code as output.

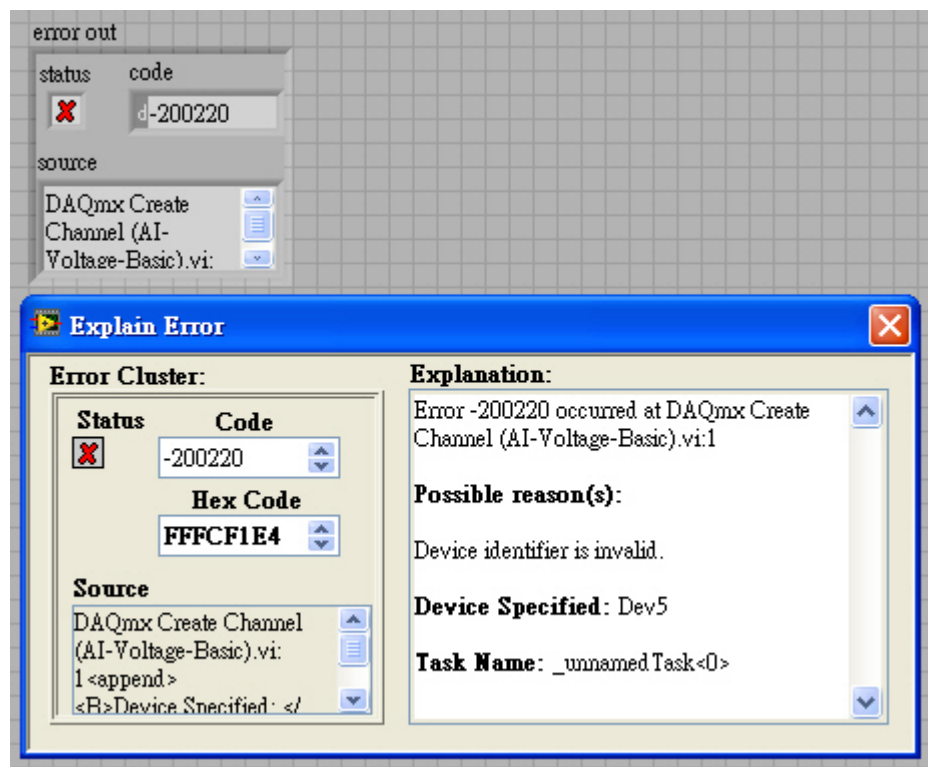
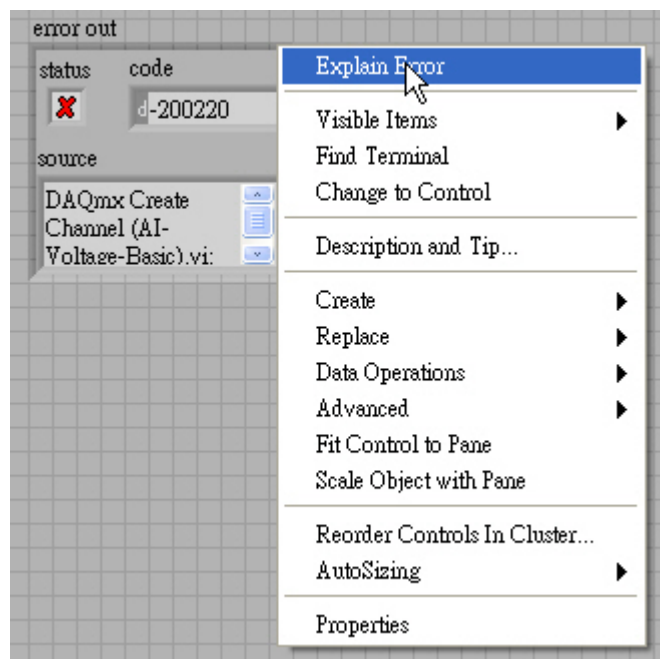
簡易錯誤處理器

- 這個簡易的錯誤處理函數位於「**Functions»All Functions»Time and Dialog**」 Wire the error cluster to the Error In (no error) input.
- 使用方法為把錯誤叢集接到這個簡易錯誤處理器的「Error In」處。



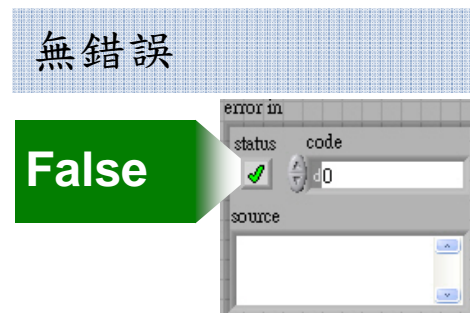
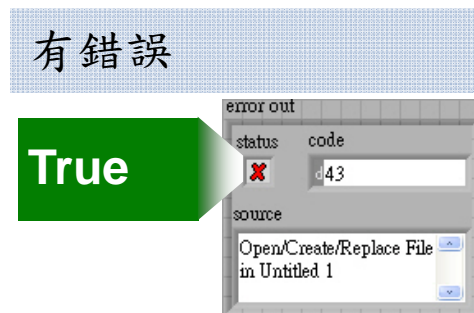
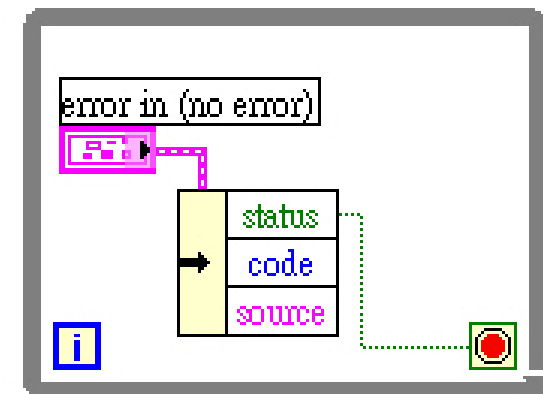
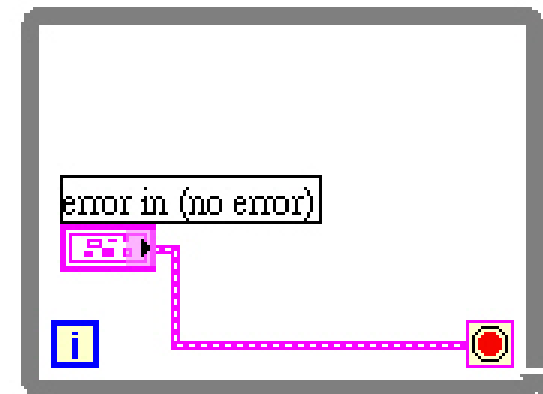
查看錯誤內容

■ 在錯誤叢集按右鍵，選「**Explain Error**」，就可以顯示錯誤訊息的內容。



使用 While Loop 做錯誤處理

- 你可以將錯誤叢集連接至 While Loop 的條件接頭(conditional terminal)，以停止 While Loop 的執行。當你將錯誤叢集連接至條件接頭時，只有錯誤叢集的status 參數的TRUE 或FALSE 值會傳送到接頭。當錯誤發生時，While Loop 就會停止
- 當錯誤叢集被連接至條件接頭時，捷徑選單中的Stop if True 及Continue if True 選項會改變為Stop on Error 及Continue while Error



本章重點回顧

- 叢集集合不同類型的資料元件。一個叢集中不能混合容納控制器及指示器。
- 要建立叢集控制器或指示器，請在Controls»All Controls»Array & Cluster 面板上選擇一個叢集，將它置於人機介面，再將控制器或指示器拖拉至叢集邊框內。
- 使用位於Functions»All Functions»Cluster 面板的Cluster 函數來建立及操作叢集。
- 錯誤查驗可以告訴你為何發生錯誤，以及錯誤發生的位置。
- 錯誤叢集會回報錯誤的status (狀態)、code (代碼)及source (資源)。
- 使用錯誤叢集控制器及指示器，在subVI 中建立錯誤輸入埠及輸出埠。