

IBM

@server

iSeries

確定控制







@server

iSeries

確定控制



# 目錄

確定控制	1
V5R2 的新增功能	1
列印此主題	2
確定控制概念	2
確定控制如何運作	3
確定和回轉作業如何運作	3
確定作業	4
回轉作業	5
確定定義	5
確定定義的範圍	6
確定定義名稱	8
範例：工作與確定定義	9
確定控制如何使用物件	11
可確定資源的類型	12
本端與遠端可確定資源	13
可確定資源的存取意向	13
可確定資源的確定通信協定	14
登載的檔案與確定控制	14
在確定控制之下的一連串異動日誌登錄	15
確定循環 ID	16
記錄鎖定	17
確定控制和獨立磁碟儲存區	18
確定控制的注意事項和限制	19
批次應用程式的確定控制	20
兩階段確定控制	21
確定處理程序中的角色	21
兩階段確定控制的異動狀態	23
兩階段確定控制的確定定義	25
兩階段確定的確定定義：允許表決唯讀	25
兩階段確定的確定定義：不等待結果	27
兩階段確定的確定定義：指示確定可以省略	30
兩階段確定的確定定義：不選取最後代理程式	32
信賴表決對於確定處理流程的影響	32
確定控制的 XA 異動支援	34
確定控制的 SQL 伺服器模式和緒範圍異動	37
啓動確定控制	38
確定通知物件	39
確定鎖定層次	40
結束確定控制	42
系統起始的結束確定控制	43
啓動群組結束期間的確定控制	43
隱含的確定和回轉作業	44
正常遞送步驟結束期間的確定控制	46
於異常系統或工作結束期間的確定控制	47
更新通知物件	48
異常終止後起始程式載入期間的確定控制回復	49
管理異動和確定控制	50
顯示確定控制資訊	50
顯示異動已鎖定的物件	51

顯示與異動相關的工作	51
顯示異動的資源狀態	51
顯示異動內容	52
最佳化確定控制的效能	52
將鎖定數目降至最少	54
管理異動大小	55
實務和範例：確定控制	56
實務範例：確定控制	56
確定控制的練習問題	59
練習問題的邏輯流程	64
與練習程式的邏輯流程相關的步驟	66
範例：使用異動日誌檔來啟動應用程式	67
範例：使用通知物件啟動應用程式	71
範例：每一個程式的唯一通知物件	72
範例：適用全部程式的單一通知物件	76
範例：使用標準處理程式來啟動應用程式	76
範例：標準處理程式的程式碼	77
範例：標準確定處理程式的程式碼	78
範例：使用標準處理程式來決定是否重新啟動應用程式	80
疑難排解異動和確定控制	81
確定控制錯誤	81
錯誤狀況	82
無錯誤狀況	83
確定控制期間監督的錯誤訊息	83
監督 CALL 指令後的錯誤	86
一般確定或回轉處理程序的失敗	87
偵測死結	88
通信失敗後回復異動	88
何時強制確定和回轉，以及何時取消重新同步	89
確定控制的相關資訊	90

---

## 確定控制

確定控制是確保資料完整性的一種功能。它可讓您以異動來處理對資源 (例如資料庫檔或表格) 的一組變更。確定控制可確保在參與的所有系統上發生整組個別變更或不發生變更。DB2 Universal Database for iSeries™ 使用確定控制功能來確定及回轉使用 \*NONE (不確定) 以外之隔離層次執行的資料庫異動。

您可以使用確定控制來設計應用程式，讓系統在工作、工作內啟動群組或系統異常終止時能夠重新啟動應用程式。透過確定控制，您可以確保應用程式重新啟動時，資料庫中不會有因為先前失敗的不完整異動所導致的部份更新。

有關如何在您的 iSeries 伺服器上啟動及執行確定控制，請參閱下列資訊。

### **V5R2 的新增功能**

此主題列示確定控制的新資訊。

### **列印此主題**

列印整個資訊。

### **確定控制概念**

有關確定控制的運作方式，請參閱此資訊。

### **啟動確定控制**

有關啟動確定控制的相關資訊，請參閱此資訊

### **結束確定控制**

有關結束確定控制的必要先決條件及結束確定控制的方式，請參閱此資訊。

### **系統起始的結束確定控制**

請參閱系統起始結束確定控制時所需執行的作業。

### **管理異動和確定控制**

請參閱以確定控制管理系統所需執行的作業。

### **實務和範例：確定控制**

有關公司如何設定確定控制的資訊，請參閱這些實務和範例。請參閱使用確定控制的程式之程式碼範例。

### **疑難排解異動和確定控制**

如果您需要確定控制的疑難排解，請參閱此資訊。

### **確定控制的相關資訊**

請參閱與確定控制相關的主題、手冊、IBM 紅皮書及外部網站。


註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

---

## V5R2 的新增功能

對於 V5R2 來說，確定控制有一些改進和新增功能。下列項目包含這些改進和新增功能的摘要。

- **確定控制和獨立磁碟儲存區**  
具有獨立磁碟儲存區能力的新檔案庫提供確定控制的支援。
- **確定控制的 XA 異動支援**  
確定控制的 XA 異動支援更加遵循 XA 規格。
- **確定控制的「iSeries 領航員」支援**  
「iSeries 領航員」提供確定控制的支援。下列主題提供明細：
  - 顯示確定控制資訊
  - 確定控制的 XA 異動支援
  - 於通信失敗後回轉異動
  - 何時強制確定和回復以及何時取消重新同步

若要知道關於這個版次的新增功能或變更的其他資訊，請參閱使用者備忘錄 (Memo to Users) 。

---

## 列印此主題

欲檢視或下載 PDF 版本，請選取確定控制 (約 600 KB 或 112 頁)。


### 儲存 PDF 檔案

如欲在您的工作站上儲存 PDF，供檢視或列印之用：

1. 在瀏覽器中以滑鼠右鍵按一下 PDF (於上述鏈結上按一下滑鼠右鍵)。
2. 按一下**另存目標...**
3. 導覽到您想要儲存 PDF 之處。
4. 按一下**儲存**。

### 下載 Adobe Acrobat Reader

如果您需要 Adobe Acrobat Reader 來檢視或列印這些 PDF，您可以從 Adobe 網站

([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  下載。

---

## 確定控制概念

這些頁面資訊可協助您瞭解確定控制如何運作、如何與您的系統相互作用，以及如何與您網路中的其他系統相互作用：

- 確定控制如何運作
- 確定和回轉作業如何運作
- 確定定義
- 確定控制下的資源
- 確定控制和獨立磁碟儲存區
- 確定控制的注意事項和限制
- 批次應用程式的確定控制
- 兩階段確定控制
- 確定控制的 XA 異動支援
- SQL 伺服器模式和緒範圍限定的異動



## 確定控制如何運作

確定控制是一種功能，它可讓您以異動來定義及處理對資源 (例如資料庫檔案或表格) 的一組變更。確定控制可確保在參與的所有系統上發生整組個別變更或不發生變更。例如，當您將資金從儲蓄帳戶轉送到活儲帳戶時，多個變更被視為一個群組。對您而言，這個轉送就像是單一變更。不過，資料庫發生不止一個變更，因為儲蓄帳戶和活儲帳戶都會被更新。若要使這兩個帳戶保持正確，活存和儲蓄帳戶必須發生全部變更或不發生任何變更。

確定控制可讓您：

- 確定受影響的所有資源已完成異動內所有變更。
- 確定處理程序被岔斷時移除了異動內所有變更。
- 移除當應用程式判定異動有錯誤時在異動期間所做的變更。

您也可以設計應用程式，以便在工作、工作內啟動群組或系統異常終止時，確定控制能夠重新啟動應用程式。透過確定控制，您可以確保應用程式重新啟動時，資料庫中不會有因為先前失敗的不完整異動所導致的部份更新。

## 異動

異動是對系統上的物件的一組個別變更，看起來應該像使用者的單一變更。

註：

「iSeries 領航員」使用的術語是異動，而字元型介面使用的術語是「邏輯工作單元 (LUW)」。這兩個術語可以交換。除非有特別指示使用字元型介面，否則本主題使用的術語是異動。

異動可以是下列任何一項：

- 其中沒有發生資料庫檔案變更的查詢。
- 變更一個資料庫檔案的簡單異動。
- 變更一或多個資料庫檔案的複雜異動。
- 變更一或多個資料庫檔案的複雜異動，但這些變更僅代表異動的邏輯群組的一部份。
- 包含多個位置上資料庫檔案的簡單或複雜異動。資料庫檔案可以：
  - 位於單一遠端系統。
  - 位於本端系統及一或多個遠端系統。
  - 指定給本端系統上的多個異動日誌。每一個異動日誌可視為一個**本端位置**。
- 本端系統上包含資料庫檔案以外物件的簡單或複雜異動。

## 確定和回轉作業如何運作






有兩個作業會影響確定控制之下的變更：

- 確定作業  
確定作業會將前次確定或回轉作業之後在確定控制之下的所有變更變成永久。系統也會釋放此異動相關的所有鎖定。
- 回轉作業  
回轉作業會移除前次確定或回轉作業之後的所有變更。系統也會釋放此異動相關的所有鎖定。

下列程式設計語言和 API 支援確定和回轉作業：

語言或 API	確定	回轉
CL	COMMIT 指令	ROLLBACK 指令
ILE RPG/400	COMIT 作業碼	ROLBK 作業碼
ILE COBOL/400 <sup>(R)</sup>	COMMIT 動詞	ROLLBACK 動詞
ILE C/400 <sup>(R)</sup>	_Rcommit 函數	_Rrollbck 函數
PL/I	PLICOMMIT 次常式	PLIROLLBACK 次常式
SQL	COMMIT 陳述式	ROLLBACK 陳述式
SQL 呼叫層次介面 (CLI)	SQLTransact() 函數 (用來確定和回轉異動)	
XA API	db2xa_commit() API	db2xa_rollback() API

下列鏈結提供關於這些程式設計語言和 API 的詳細資訊：

- [COBOL/400 User's Guide](#) 
- [RPG/400 User's Guide](#) 
- [ILE C/C++ Programmer's Guide](#) 
- [CL Programming](#) 
- [System API Programming](#) 
- [DB2 UDB for iSeries SQL 呼叫層次介面](#)
- [DB2 UDB for iSeries 程式設計概念](#)

## 確定作業

確定作業可使上次確定或回轉作業後在確定控制下所做的全部變更成為永久變更。系統也會釋放所有與異動相關的鎖定。

系統在收到確定要求時會執行下列步驟：

- 系統會儲存供回復時使用的確定 ID (如果有提供的話)。
- 如果下列兩者屬實，則系統在執行確定作業之前會將記錄寫入檔案中：
  - 新增確定控制下之記錄至本端或遠端資料庫檔案。
  - 開啓檔案時指定了 SEQONLY(\*YES)，使系統使用區塊化 I/O 回饋且部份區塊記錄存在。

否則，I/O 回饋區域和 I/O 緩衝區不會變更。

- 系統針對確定定義中每一個 API 確定資源呼叫確定和回轉跳出程式。如果某個位置登記了多個跳出程式，則系統會依照在該位置的登記順序來呼叫跳出程式。
- 如果指定了對資源所做的任何記錄變更給異動日誌，系統會將 C CM 異動日誌登錄寫入與確定定義相關的每一個本端異動日誌。確定控制下異動日誌登錄的順序顯示通常在確定定義作用中時所寫入的登錄。
- 系統使擱置中的「物件層次」變更成為永久性變更。
- 系統將取得並保存供確定控制使用的記錄和物件鎖定解除。那些資源可供其他使用者使用。
- 系統變更確定定義中的資訊，指示現行異動已終止。

系統必須正確執行先前所有步驟，確定作業才能順利完成。

## 回轉作業

回轉作業會移除前次確定或回轉作業之後的所有變更。系統也會釋放此異動相關的所有鎖定。系統在接收回轉要求時會執行下列步驟：

- 如果下列兩者成立，系統會清除 I/O 緩衝區的記錄：
  - 如果記錄新增到在確定控制之下的本端或遠端資料庫檔案。
  - 開啓檔案時如果已指定 SEQONLY(\*YES)，使得系統使用區塊化 I/O，而且有部份記錄區塊尚未寫入資料庫。

否則，I/O 回饋區域和 I/O 緩衝區維持不變。

- 系統對確定定義中存在的每一個 API 確定資源，呼叫確定或回轉跳出程式。如果一個位置有登記一個以上的跳出程式，系統會以相反的登記順序來呼叫此位置的跳出程式。
- 如果從檔案中刪除一筆記錄，系統會將該記錄新增回到該檔案。
- 系統會移除此異動期間的任何記錄變更，將原始記錄 (未更新表格內容) 放回檔案中。
- 如果此異動期間有任何記錄新增到檔案，則它們會與被刪除的記錄一同保留在檔案中。
- 如果異動期間對於指派給日誌的資源有任何記錄變更，系統會新增一個異動日誌登錄 C RB 到日誌中，指出發生回轉作業。日誌也包含已回轉的記錄變更的影像。在要求回轉作業之前，已變更的記錄的更新後表格內容和未更新表格內容會放在日誌中。如果有任何可確定資源指派給日誌，則系統也會在預設日誌中寫入 C RB 登錄。
- 系統將確定控制之下開啓的檔案放在下列其中一個位置：
  - 前次異動中最後存取的記錄
  - 如果沒有使用此確定定義對檔案執行任何確定作業，則放在開啓位置

如果您執行循序處理，則這個注意事項非常重要。

- 系統不回轉資料庫檔案的不可確定變更。例如，不關閉已開啓的檔案，且不復置被清除的檔案。系統不重新開啓或重新定位此異動期間關閉的任何檔案。
- 系統會解除鎖定基於確定控制目的而取得的記錄鎖定，提供這些記錄給其他使用者使用。
- 系統目前儲存的確定 ID 與相同確定定義的最後確定作業所提供的確定 ID 維持相同。
- 系統反轉或回轉此異動期間的物件層次可確定變更。
- 基於確定控制目的而取得的物件鎖定會被解除鎖定，這些物件將可供其他使用者使用。
- 系統建立前一個確定界限做為現行確定界限。
- 系統變更確定定義中的資訊來顯示已結束現行異動。

系統必須正確地執行先前所有的步驟，回轉作業才能順利完成。

## 確定定義

當使用「啓動確定控制 (STRCMTCTL)」指令以於系統上啓動確定控制時，要建立確定定義。另外，當隔離層次不是「不確定」時，DB2 UDB for iSeries 會自動建立確定定義。確定定義包含在該工作內確定控制下變更的資源之相關資訊。系統將確定定義中的確定控制資訊作為確定資源變更維護，直到確定定義結束為止。系統上每一個作用中異動以一個確定定義代表。在作用中異動的每一個確定或回轉後，後續的異動可重覆使用確定定義。

確定定義通常包括：

- STRCMTCTL 指令上的參數。
- 確定定義的現行狀態。
- 有關包含現行異動期間所做變更的資料庫檔案及其他可確定資源的資訊。

對於具有工作範圍鎖定的確定定義，僅啟動確定定義的工作知道該確定定義。其他工作不會知道該確定定義。

程式可啟動及使用多重確定定義。工作的每一個確定定義可識別具有與其相關的可確定資源之個別異動。這些異動可單獨從該工作已啟動的其他確定定義的相關異動中確定或回轉。

下列提供有關確定定義的詳細資料：

- 確定定義的範圍
- 確定定義名稱
- 範例：工作與確定定義

有關確定定義和獨立磁碟儲存區的規則，請參閱確定控制和獨立磁碟儲存區。

## 確定定義的範圍

確定定義的範圍決定哪些程式將使用該確定定義，及在異動期間獲得的鎖定如何限定範圍。啟動確定定義的介面決定確定定義的範圍。對於確定定義來說有四個可能範圍，共成分兩大類：

### 含有「工作範圍」鎖定的確定定義

- 「啟動群組層次」確定定義
- 「工作層次」確定定義
- 明確命名的確定定義

### 含有「異動範圍」鎖定的確定定義。

- 異動範圍的確定定義

只有在啟動確定定義的工作中執行的程式才能使用含有「工作範圍」鎖定的確定定義。相較之下，一個以上的工作可以使用含有「異動範圍」鎖定的確定定義。

應用程式通常使用「啟動群組層次」或「工作層次」確定定義。這些確定定義是使用「啟動確定控制 (STRCMTCTL)」指令明確建立的，或當 SQL 應用程式以不是 \*NONE 的隔離層次執行時由系統隱含建立的。

### 「啟動群組層次」確定定義

最常見的範圍限定為啟動群組。當 STRCMTCTL 指令明確啟動確定定義時，或以非「不確定」的隔離層次執行的 SQL 應用程式隱含地啟動確定定義時，「啟動群組層次」確定定義是預設範圍。只有在該啟動群組內執行的程式才使用該確定定義。一個工作可以同時有許多作用中的「啟動群組層次」確定定義。然而，每一個「啟動群組層次」確定定義只能與單一啟動群組相關。在該啟動群組內執行的程式只能使它們的可確定變更與該「啟動群組層次」確定定義相關。

當「iSeries 領航員」、「使用確定定義 (WRKCMDFN)」指令、「顯示工作 (DSPJOB)」指令或「使用工作 (WRKJOB)」指令顯示「啟動群組層次」確定定義時，這些欄位會顯示下列內容：

- 確定定義欄位顯示啟動群組名稱。它顯示特殊值 \*DFACTGRP 以指出預設啟動群組。
- 啟動群組欄位顯示啟動群組號碼。
- 工作欄位顯示啟動確定定義的工作。
- 緒欄位顯示 \*NONE。

### 「工作層次」確定定義

確定定義只能以發出 `STRCMTCTL CMTSCOPE(*JOB)` 來限定其範圍為工作。在沒有「啟動群組層次」確定定義的啟動群組中執行的程式會使用「工作層次」確定定義，若工作的另一個程式已經啟動工作層次確定定義的話。一個工作只能啟動一個工作層次確定定義。

當「iSeries 領航員」、「使用確定定義 (WRKCMDFN)」指令、「顯示工作 (DSPJOB)」指令 或「使用工作 (WRKJOB)」指令顯示「工作層次」確定定義時，這些欄位會顯示下列內容：

- 確定定義欄位顯示特殊值 `*JOB`。
- 啟動群組欄位顯示空白。
- 工作欄位顯示啟動確定定義的工作。
- 緒欄位顯示 `*NONE`。

對於給定的啟動群組來說，在該啟動群組內執行的程式只能使用單一確定定義。因此在某啟動群組內執行的程式可以使用「工作層次」或「啟動群組層次」確定定義，但無法同時使用兩個定義。在沒有使用 `SQL` 伺服器模式的多緒工作中，會根據程式的啟動群組將程式的異動工作的範圍限定在適當的確定定義，而不考慮哪一個緒執行程式。如果多重緒使用相同啟動群組，它們必須合作才能執行異動工作並確保在正確時間發生確定和回轉。

即使工作有作用中的「工作層次」確定定義，如果在該啟動群組內沒有任何程式執行「工作層次」確定定義的確定控制要求或作業，程式仍可以啟動該「啟動群組層次」確定定義。否則，您必須先結束「工作層次」確定定義之後才能啟動該「啟動群組層次」確定定義。可防止「啟動群組層次」確定定義被啟動的「工作層次」確定定義的確定控制要求或作業包括：

- 在確定控制之下開啓 (完整或共用) 資料庫檔案。
- 使用「新增確定資源 (QTNADDCR)」API 來新增 API 確定資源。
- 確定異動。
- 回轉異動。
- 在確定控制之下新增遠端資源。
- 使用「變更確定選項 (QTNCHGCO)」API 變更確定選項。
- 使用「需要回轉 (QTNBRQD)」API 使確定定義處於需要回轉狀態。
- 搭配使用「傳送異動日誌登錄 (QJOSJRNE)」API 與「併入確定循環 ID」參數，來傳送一個包含現行確定循環 ID 的使用者異動日誌登錄。

同樣地，如果啟動群組內的程式目前正使用「啟動群組層次」確定定義，那麼必須先結束確定定義之後，在相同啟動群組內執行的程式才能使用「工作層次」確定定義。

開啓資料庫檔案時，已開啓檔案的開啓範圍可以限定為啟動群組或工作，但有一項限制：如果在確定控制之下某程式正在開啓檔案而且檔案的範圍限定為工作，那麼提出開啓要求的程式必須使用「工作層次」確定定義。

### 「明確命名」的確定定義

當系統需要執行自己的確定控制異動而不影響應用程式使用的異動時，系統會啟動「明確命名」的確定定義。啟動這些類型之確定定義的功能範例為問題日誌。應用程式無法啟動明確命名的確定定義。

當「iSeries 領航員」、「使用確定定義 (WRKCMDFN)」指令、「顯示工作 (DSPJOB)」指令或「使用工作 (WRKJOB)」指令顯示「明確命名」的確定定義時，這些欄位會顯示下列內容：

- 確定定義欄位顯示系統提供給它的名稱。
- 啟動群組欄位顯示空白。
- 工作欄位顯示啟動確定定義的工作。

- 緒欄位顯示 \*NONE。

### 「異動範圍」確定定義

由「異動範圍鎖定」的 XA API 啟動「異動範圍」確定定義。

這些 API 使用的確定控制通信協定是以緒或 SQL 連線為基礎，而不是以啟動群組為基礎。換句話說，當執行異動工作時 API 是用來使確定定義與某特定緒或 SQL 連線相關，以及使異動確定或回轉。系統根據 API 通信協定，將這些確定定義附加到執行異動工作的緒。它們可以給不同工作中的緒使用。

當「iSeries 領航員」、「使用確定定義 (WRKCMDFN)」指令、「顯示工作 (DSPJOB)」指令或「使用工作 (WRKJOB)」指令顯示「異動範圍」確定定義時，這些欄位會顯示下列內容：

- 確定定義欄位顯示特殊值 \*TNSOBJ。
- 啟動群組欄位顯示空白。
- 工作欄位顯示啟動確定定義的工作。
- 緒欄位顯示連接確定定義的緒 (或顯示 \*NONE，表示確定定義目前沒有連接到任何緒)。

### 確定定義名稱

系統會提供名稱給針對某工作啟動的所有確定定義。下表顯示用於特定工作的各種確定定義及其相關名稱。

啟動群組	確定範圍	確定定義名稱
任何	工作	*JOB
預設啟動群組	啟動群組	*DFACTGRP
使用者指名的啟動群組	啟動群組	啟動群組名稱 (例如，PAYROLL)
系統指名的啟動群組	啟動群組	啟動群組號碼 (例如，000000145)
無	明指名的命名	QDIR001 (僅供系統使用之系統定義確定定義範例)。以 Q 為開頭之系統定義的確定定義名稱。
無	異動	*TNSOBJ

僅「整合語言環境 (ILE)」已編譯的程式可啟動預設啟動群組以外的啟動群組之確定控制。因此，唯有工作執行一或多個 ILE 已編譯的程式，才能使用多重確定定義。有關「整合語言環境」<sup>(R)</sup> 的詳細資訊，請參閱 ILE

Concepts 。

「原始程式模型 (OPM)」程式在預設啟動群組中執行，依預設，它使用 \*DFACTGRP 確定定義。在混合 OPM 和 ILE 環境中，如果所有程式執行的所有可確定變更能夠一起確定或回轉，則工作應使用工作層次確定定義。

範圍限定為啟動群組的一個已開啓資料庫檔案可與「啟動群組層次」或「工作層次」確定定義產生關聯。範圍限定為工作的一個已開啓資料庫檔案只能與「工作層次」確定定義產生關聯。因此，在範圍限定為工作的確定控制下開啓資料庫檔案的任何程式、OPM 或 ILE，皆須使用「工作層次」確定定義。

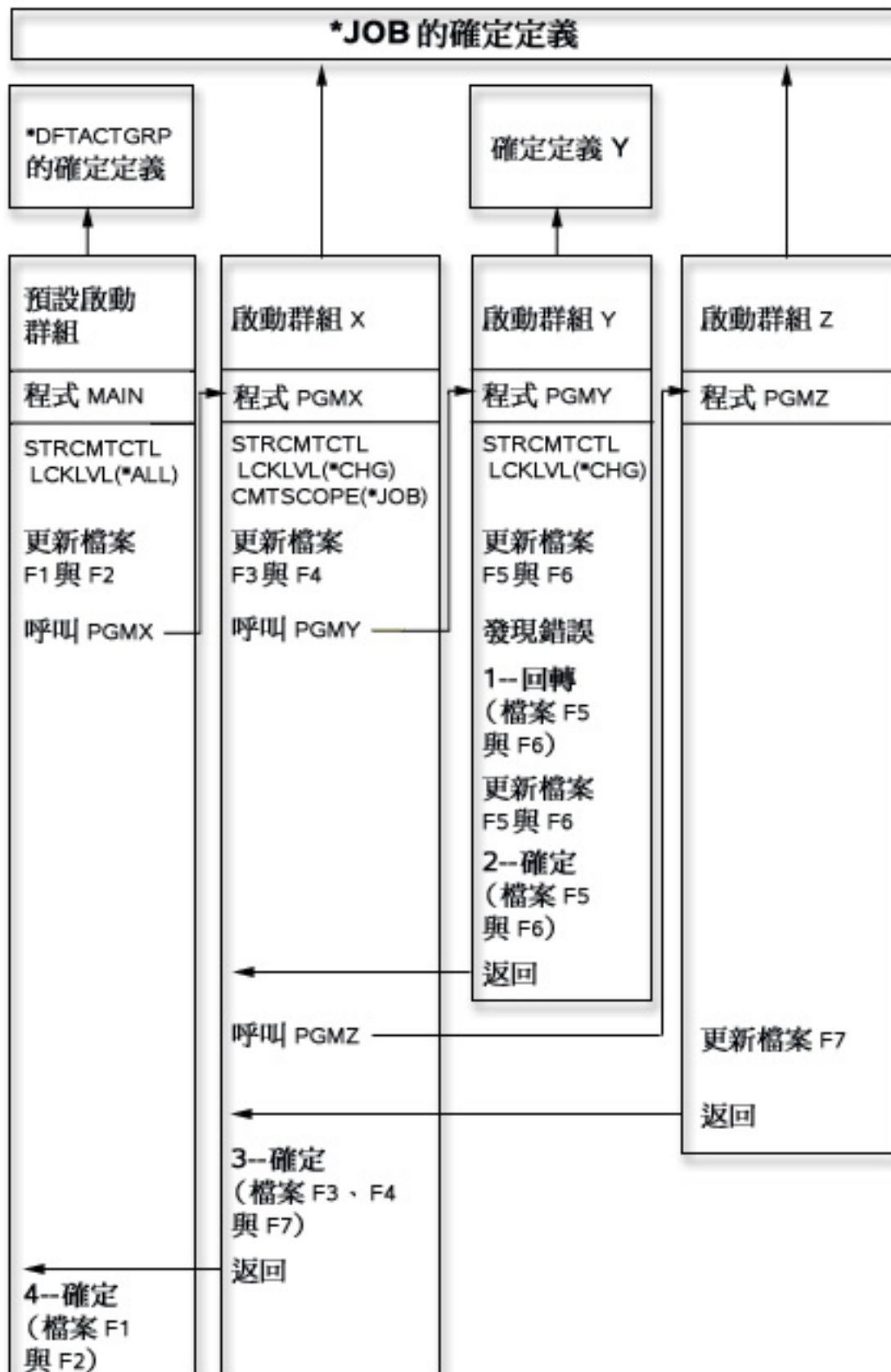
應用程式在建立確定控制要求時沒有使用確定定義名稱來識別特定的確定定義。確定定義名稱主要用於訊息中識別工作的特定確定定義。

以「啟動群組層次」確定定義而言，系統會根據要求程式在其中執行的啟動群組來決定使用哪一個確定定義。這可能是因為在啟動群組內，在任何時間點執行的程式只能使用單一確定定義。

以具有「異動限定範圍」鎖定的異動而言，XA API 和新增至 CLI 的異動相關屬性可決定呼叫緒使用哪一個確定定義。

### **範例：工作與確定定義**

下圖顯示使用多重確定定義的工作範例。它指示在每個啟動群組層次上哪些檔案更新被確定或回轉。此範例假設所有程式對資料庫檔案做的更新都在確定控制下完成。



\*



下表顯示上圖中實務範例變更時確定或回轉檔案的方式：

### 工作中多重確定定義的其他範例

實務範例中的變更	變更這些檔案的影響：			
	F1 與 F2	F3 與 F4	F5 與 F6	F7
PGMX 執行的是回轉作業而非確定作業 (3=COMMIT 變成 ROLLBACK)。	仍擱置中	回轉	已確定	回轉
PGMZ 傳回 PGMX 之前執行了確定作業。	仍擱置中	已由 PGMZ 確定	已確定	已確定
PGMZ 在更新檔案 F7 之後，嘗試指定 CMTSCOPE(*ACTGRP) 來啟動確定控制。由於使用工作層次確定定義的變更擱置中，所以嘗試失敗。	仍擱置中	仍擱置中	已確定	仍擱置中
PGMX 未啟動確定控制且未使用 COMMIT(*YES) 開啓檔案 F3 和 F4。PGMZ 嘗試使用 COMMIT(*YES) 開啓檔案 F7。	仍擱置中	不在確定控制下	已確定	無法開啓檔案 F7，因為 *JOB 確定定義不存在 (PGMX 沒有建立它)。

### 確定控制如何使用物件

當您將物件置於確定控制之下時，物件會變成可確定資源。並且登記在確定定義中。此物件參與該確定定義發生的每一個確定作業和回轉作業。

下列主題說明可確定資源的這些屬性：

- 資源類型
- 位置
- 確定通信協定
- 存取意向

下列鏈結提供確定控制下的資源的詳細資訊：


- 可確定資源的類型
- 本端和遠端可確定資源
- 可確定資源的存取意向
- 可確定資源的確定通信協定
- 日誌登載檔和確定控制
- 確定控制下的一連串異動日誌登錄
- 確定週期 ID
- 記錄鎖定

## 可確定資源的類型

下列表格顯示：

- 可確定資源的類型。
- 如何放置於確定控制之下。
- 如何由確定控制中移除。
- 套用於資源類型的限制。

資源類型	如何放置於確定控制之下	如何由確定控制中移除	何種變更是可確定的	限制
FILE- 區域資料庫檔案	在確定控制之下開啓 <sup>1</sup>	如果沒有擱置中的變更，則關閉檔案。  在執行下一個確定或回轉作業之後，如果關閉檔案時有擱置中的變更。	記錄層次變更	單一異動不可以鎖定 500 000 000 筆以上的記錄 <sup>2</sup> 。
DDL- 對區域 SQL 表格和 SQL 集合的「物件層次」變更。	在確定控制之下執行 SQL	在「物件層次」變更之後執行確定或回轉作業。	物件層次變更，例如： • 建立 SQL 資料包 • 建立 SQL 表格 • 捨棄 SQL 表格	只有使用 SQL 的「物件層次」變更是在確定控制之下。
DDM- 遠端分散式資料管理 (DDM) 檔案	在確定控制之下開啓。DDM 的確定控制支援提供確定控制和分散式資料管理的詳細資訊。	如果沒有擱置中的變更，則關閉檔案。  在執行下一個確定或回轉作業之後，如果關閉檔案時有擱置中的變更。	記錄層次變更	
LU 6.2- 受保護的交談	啓動交談 <sup>3</sup>	結束交談		
DRDA <sup>(R)</sup> - 分散式關聯資料庫	使用 SQL CONNECT 陳述式	結束連線		
API- 區域 API 確定資源	新增確定資源 (QTNADDCR) API	移除確定資源 (QTNRMVCR) API	由使用者程式決定。異動日誌登錄可由使用者程式使用傳送異動日誌登錄 (QJOSJRNE) API 來寫入，以協助追蹤這些變更。	應用程式必須提供一個跳出程式，供確定、回轉或重新同步作業期間呼叫。
TCP-TCP/IP 連線	對於定義來使用 TCP/IP 連線的 RDB 使用 SQL CONNECT 陳述式，或開啓以 TCP/IP 位置來定義的 DDM 檔案	結束 SQL 連線，或沒有擱置中變更時關閉 DDM 檔案。如果關閉 DDM 檔案時有擱置中變更，則在執行下一個確定或回轉作業之後關閉連線。		

資源類型	如何放置於確定控制之下	如何由確定控制中移除	何種變更是可確定的	限制
<p>註：</p> <p><sup>1</sup>關於資料庫檔案如何放置在確定控制之下的詳細資訊，請參閱適當的語言參考手冊。確定控制的相關資訊鏈結到您可以使用的語言手冊。</p> <p><sup>2</sup>您可以使用 QAQQINI 檔案來減少 500 000 000 的限制。請參閱管理異動大小中的指示。</p> <p><sup>3</sup>當啟動 DDM 連線時，DDM 檔案會指定 PTCCNV(*YES)，DDM 檔案會定義一個 SNA 遠端位置，LU6.2 資源會新增 DDM 資源。</p> <p>當啟動 DRDA 連線時，如果下列兩者成立，LU6.2 資源會新增 DRDA 資源：</p> <ul style="list-style-type: none"> <li>• 程式正在使用分散式工作單元連線通信協定。</li> <li>• 連線會連接到以 SNA 遠端位置來定義的 RDB。關於啟動受保護交談的詳細資訊，請參閱 APPC Programming</li> </ul> 				

## 本端與遠端可確定資源

可確定資源可為本端資源或遠端資源。

### 本端可確定資源

本端可確定資源與應用程式位於相同系統上。每一個與確定控制下的資源相關的異動日誌可視為一個本端位置。不是以異動日誌登記的所有資源 (DDL 資源和 API 資源為選用性) 可視為個別的本端位置。

如果可確定資源位於獨立磁碟儲存區，但確定定義位於另一個磁碟儲存區，則該資源不視為本端資源。有關可確定資源和獨立磁碟儲存區的詳細資訊，請參閱確定控制和獨立磁碟儲存區。

### 遠端可確定資源

遠端可確定資源與應用程式位於不同系統上。與遠端系統的每一個唯一交談都有一個遠端位置。在一或多個遠端系統上，一個確定定義可能有一或多個遠端位置。

當您將本端資源置於系統磁碟儲存區或任何獨立磁碟儲存區的確定控制下時，您必須使用 DRDA 來存取其他任何獨立磁碟儲存區中位於確定控制下的資源。

下列顯示可確定資源的類型及其位置：

資源類型	位置
FILE	本端
DDL	本端
API	本端
DDM	遠端
LU62	遠端
DRDA	本端或遠端
TCP	遠端

## 可確定資源的存取意向

當資源被置於確定控制下時，資源管理程式指示資源的存取方式：

- 更新

- 唯讀
- 待確定

存取意向決定資源參數異動的方式。下表顯示特定資源類型可用的存取意向，以及系統決定已登記資源的存取意向的方式：

資源類型	可用的存取意向	存取意向的決定方式
FILE	更新、唯讀	根據檔案開啓方式
DDL	更新	一律更新
API	更新	一律更新
DDM	更新、唯讀	根據檔案開啓方式
LU62	待確定	一律待確定
DRDA	更新、唯讀、待確定	以 DRDA 層次 1 而言，如果沒有登記其他遠端資源，則更新存取意向。否則，存取意向是唯讀。以 DRDA 層次 2 而言，存取意向一律待確定。
TCP	待確定	一律待確定

已登記的資源存取意向可決定是否登記新資源。下列規則套用至：

- 當下列任一項屬實時，存取意向為更新的一階段資源無法被登記：
  - 存取意向為更新的資源已在其他位置登記。
  - 存取意向為待確定的資源已在其他位置登記。
  - 存取意向為待確定的資源已在相同位置登記且資源已在現行異動期間變更。
- 當存取意向為更新的一階段資源已登記時，存取意向為更新的兩階段資源無法被登記。

## 可確定資源的確定通信協定

**確定通信協定**是資源能夠參與單階段或兩階段確定處理程序的能力。除了 API 可確定資源之外，區域資源一律為兩階段資源。

當可確定資源位於獨立的磁碟儲存區時，如果確定定義位於不同的磁碟儲存區上，則資源不被視為區域資源或兩階段資源。關於可確定資源和獨立磁碟儲存區的詳細資訊，請參閱確定控制和獨立的磁碟儲存區。

兩階段資源也稱為**受保護的資源**。遠端資源和 API 可確定資源在確定控制下時，必須登記為單階段資源或兩階段資源。下列表格顯示在確定定義中，何種可確定資源可以與單階段資源同時存在：

資源類型	可以同時存在
單階段 API 資源	其他區域資源。無遠端資源。
單階段遠端資源	相同位置的其他單階段資源。無區域資源。

## 登載的檔案與確定控制

在於確定控制下開啓資料庫檔案來輸出之前，或使用「不確定」以外之隔離層次的 SQL 應用程式參照資料庫檔案之前，您必須登載（記載）該資料庫檔案（資源類型 FILE 或 DDM）。若開啓檔案僅在確定控制下輸入，則不需要登載該檔案。下列情況會發生錯誤：

- 嘗試開啓資料庫檔案在確定控制下輸出，但該檔案目前未被登載。
- 未啓動可供確定控制下開啓的檔案使用之確定定義。

如果僅對確定控制下開啓的資料庫檔案登載更新後表格內容，則系統會自動啓動未更新表格內容及更新後表格內容的日誌登載。未更新表格內容僅針對在確定控制下發生檔案變更來撰寫。如果同時有不在確定控制下發生的其他檔案變更，則僅針對那些變更撰寫更新後表格內容。

系統自動將「記錄層次」可確定變更及「物件層次」可確定變更寫入異動日誌。若是「記錄層次」變更，則系統在必要時會將異動日誌登錄用於回復；系統不使用「物件層次」可確定變更的登錄作為回復用途。此外，系統不自動撰寫 API 確定資源的異動日誌登錄。不過，API 資源的跳出程式可使用「傳送異動日誌登錄 (QJOSJRNE)」API 來撰寫異動日誌登錄，以提供審核追蹤或回復協助。這些登錄內容由使用者跳出程式控制。

系統使用異動日誌以外的機制來執行「物件層次」確定資源回復。藉由呼叫與每一個特定 API 確定資源相關的確定和回轉跳出程式，可完成 API 確定資源的回復。跳出程式負責執行此狀況所需的實際回復。

有關日誌登載的詳細資訊，請參閱日誌管理主題。

## 在確定控制之下的一連串異動日誌登錄

下表顯示當確定定義為作用中時通常寫入的一連串登錄。您可以使用異動日誌碼搜尋器，取得關於異動日誌登錄內容的其他資訊。

唯有至少下列其中一項屬實，確定控制登錄才會寫入異動日誌 (本端或遠端)：

- 在「啓動確定控制 (STRCMTCTL)」指令上指定異動日誌為預設異動日誌。
- 在確定控制之下開啓至少一個指定給異動日誌的檔案。
- 在確定控制之下登記至少一個與異動日誌相關的 API 確定資源。

登錄類型	說明	寫入的位置	寫入的時間
C BC	開始確定控制	寫入至預設異動日誌，如果在 STRCMTCTL 指令上有指定它的話。	發出 STRCMTCTL 指令時
		寫入至每一個本端位置的異動日誌。	當開啓指定給異動日誌的第一個檔案時或在異動日誌中登記 API 資源時。
C SC	開始確定循環	寫入至每一個本端位置的異動日誌。	當指定給這個異動日誌 <sup>1</sup> 的檔案的異動發生第一筆記錄變更時。
		寫入至 API 資源的異動日誌。	當首先搭配使用 QJOSJRNE API 與併入確定循環ID 鍵時。
異動日誌碼 D 和 F	DDL 物件層次登錄	寫入至與被更新的物件相關的異動日誌。唯有包含確定循環 ID 的異動日誌登錄代表屬於異動一部份的 DDL 物件層次變更。	發生更新時。
異動日誌碼 R	記錄層次登錄	寫入至與被更新的檔案相關的異動日誌。	發生更新時。
異動日誌碼 U	使用者建立的登錄	寫入至與 API 資源相關的異動日誌。	如果應用程式使用 SNDJRNE 指令或 QJOSJRNE API。

登錄類型	說明	寫入的位置	寫入的時間
C CM	確定	寫入至每一個位置的異動日誌。	順利完成確定時。
		寫入至預設異動日誌。	如果任何可確定資源與異動日誌相關。
C RB	回轉	寫入至每一個本端位置的異動日誌。	完成回轉作業之後。
		寫入至預設異動日誌。	如果任何可確定資源與異動日誌相關。
C LW	結束異動	寫入至預設異動日誌，若在 STRCMTCTL 指令上已指定它的話。系統寫入 LW 標頭記錄和一或多筆詳細記錄。唯有在 STRCMTCTL 指令上指定 OMTJRNE(*NONE) 或發生系統錯誤時才會寫入這些登錄。	完成確定或回轉作業時。
C EC	結束確定控制	寫入至每一個本端位置的異動日誌。	當完成結束確定控制 (ENDCMTCTL) 指令時。
		寫入至非預設異動日誌的本端異動日誌。	當建立確定界限時，在從確定控制移除與該異動日誌相關的所有可確定資源之後。
<p>註：</p> <p><sup>1</sup> 您可以指定異動日誌登錄的固定長度部份包含異動資訊，方法是對「建立異動日誌 (CRTJRN)」或「變更異動日誌 (CHGJRN)」指令的「固定長度資料 (FIXLENDTA)」參數指定「邏輯工作單元 (*LUW)」值。藉由指定 FIXLENDTA (*LUW) 參數，每一個 C SC 異動日誌登錄的固定長度部份會包含現行異動的「邏輯工作單元 ID (LUWID)」。XA 異動也是一樣，如果指定 FIXLENDTA(*LUW) 參數，那麼每一個 C SC 異動日誌登錄的固定長度部份會包含現行異動的 XID。如果某特定異動涉及多個異動日誌或系統，那麼 LUWID 或 XID 可協助您找到該異動的所有確定循環。</p>			

## 確定循環 ID

**確定循環**是從一個確定界限到下一個確定界限的時間。系統指派一個**確定循環 ID**，將特定確定循環的所有異動日誌登錄結合在一起。參與異動的每一個異動日誌都有自己的確定循環及確定循環 ID。

確定循環 ID 是針對確定循環撰寫的 C SC 異動日誌登錄之日誌序號。確定循環 ID 位於確定循環期間撰寫的每一個異動日誌登錄中。若於確定循環期間使用了多個異動日誌，則每個異動日誌的確定循環 ID 不同。

您可藉由指定「邏輯工作單元 (\*LUW)」值給「建立異動日誌 (CRTJRN)」或「變更異動日誌 (CHGJRN)」指令的「固定長度資料 (FIXLENDTA)」參數，來指定異動日誌登錄的固定長度部份，包括異動資訊。藉由指定 FIXLENDTA (\*LUW) 參數，每一個 C SC 異動日誌登錄的固定長度部份將包含現行異動的「邏輯工作單元 ID (LUWID)」。同樣地，若是 XA 異動，如果您指定 FIXLENDTA(\*LUW) 參數，則每一個 C SC 異動日誌登錄的固定長度部份將包含現行異動的 XID。如果有多個異動日誌或系統涉及異動，LUWID 或 XID 可協助您尋找特定異動的所有確定循環。

您可以使用「傳送異動日誌登錄 (QJOSJRNE)」API 來撰寫 API 資源的異動日誌登錄。您可以選擇在那些異動日誌登錄上併入確定循環 ID。

您可以使用確定循環 ID，利用「套用登載的變更 (APYJRNCHG)」指令將登載的變更套用到確定界限，或利用「移除登載的變更 (RMVJRNCHG)」指令將登載的變更移除至確定界限。這些限制套用至下列所示：

- 在確定控制下所做的大部份「物件層次」變更寫入異動日誌中，但不使用 APYJRNCHG 和 RMVJRNCHG 指令來套用或移除。
- QJOSJRNE API 撰寫具有異動日誌碼 U 之使用者建立的異動日誌登錄。這些登錄無法使用 APYJRNCHG 和 RMVJRNCHG 指令來套用或移除。它們必須以使用者撰寫的程式來套用或移除。

## 記錄鎖定

當一個工作握有一個記錄鎖定且另一個工作嘗試擷取此記錄來更新時，要求端工作會等待並從作用中處理程序中移除，直到下列其中一個情形發生：

- 釋放記錄鎖定。
- 結束指定的等待時間。

一個以上的工作可以同時要求一筆被另一個工作鎖定的記錄。當釋放此記錄鎖定时，第一個要求此記錄的工作會接收此記錄。當等待一筆被鎖定的記錄時，請於下列建立、變更或置換指令的 WAITRCD 參數中指定等待時間：

- 建立實體檔案 (CRTPF)
- 建立邏輯檔案 (CRTLF)
- 建立來源實體檔 (CRTSRCPF)
- 變更實體檔案 (CHGPF)
- 變更邏輯檔案 (CHGLF)
- 變更來源實體檔 (CHGSRCPF)
- 置換資料庫檔案 (OVRDBF)

指定等待時間時，請考慮下列事項：

- 如果您不指定一個值，程式會等待處理的預設等待時間。
- 對於只有「異動範圍」鎖定的確定定義，工作預設等待時間可以被異動鎖定等待時間所置換，此時間可指定於：
  - xa\_open API。
  - JDBC 或 JTA 介面。分散式異動列出這些 API。
- 如果無法在指定的時間內配置記錄，則傳送通知訊息給高階語言程式。
- 如果超過記錄的等待時間，則傳送到工作日誌的訊息會指出保留鎖定記錄的工作名稱，它造成要求端工作等待。如果您遇到記錄鎖定異常，可以使用工作日誌來協助判斷要變更哪些程式，使之不要長時間保留鎖定。

程式基於下列其中一項理由而長時間保留記錄鎖定：

- 記錄仍然被鎖定的同時，工作站使用者想要進行變更。
- 記錄鎖定是一個長時間確定異動的一部份。請考慮縮小異動，使確定作業可以更頻繁地執行。
- 發生不想要的鎖定。例如，假設有一個檔案被定義成擁有唯一鍵的更新檔案，且程式在該檔案中更新和新增其他記錄。如果工作站使用者新增某記錄到該檔案中，程式可能嘗試存取該記錄來決定鍵是否已存在。如果存在，程式向工作站使用者通知要求無效。從檔案中擷取記錄時，記錄會被鎖定，直到被相同檔案的另一個讀取作業隱含地釋放，或直到被明確地釋放。

註：

關於如何使用每一個高階語言介面來釋放記錄鎖定的詳細資訊，請參閱適當的高階語言參考手冊。確定控制的相關資訊提供與高階語言手冊的鏈結，供您在使用確定控制時參考。

如果指定 LCKLVL(\*ALL)，則鎖定的持續時間會非常久，因為從檔案中擷取的記錄會被鎖定到下一個確定或回轉作業為止。不會被另一個讀取作業隱含地釋放，也無法被明確地釋放。

另一個可以在檔案上放置鎖定的功能是「當作用中時儲存」功能。當作用中時儲存您的伺服器主題提供「當作用中時儲存」功能的詳細資訊。

## 確定控制和獨立磁碟儲存區

獨立磁碟儲存區和獨立磁碟儲存區群組皆可擁有一個別的 OS/400 SQL 資料庫。您可以搭配確定控制來使用這些資料庫。不過，因為每一個獨立磁碟儲存區或獨立磁碟儲存區群組都有個別的 SQL 資料庫，所以您應該遵循這些注意事項。

### 確定定義的注意事項

當您啟動確定控制時，會在 QRECOVERY 檔案庫中建立確定定義。每一個獨立磁碟儲存區或獨立磁碟儲存區群組都有自己的 QRECOVERY 檔案庫版本。在獨立磁碟儲存區上，QRECOVERY 檔案庫的名稱為 QRCYxxxx，其中 xxxx 是獨立磁碟儲存區號碼。例如，獨立磁碟儲存區 39 的 QRECOVERY 檔案庫名稱為 QRCY00039。此外，如果獨立磁碟儲存區是某個磁碟儲存區群組的一部份，則僅主要磁碟儲存區具有 QRCYxxxx 檔案庫。

當您啟動確定控制時，會在與該工作相關的獨立磁碟儲存區之 QRECOVERY 檔案庫中建立確定定義，使確定控制在獨立磁碟儲存區上作用。

確定定義的其他注意事項如下：

- 當確定控制在獨立磁碟儲存區上作用時，使用「設定 ASP 群組 (SETASPGRP)」指令會有下列效果：
  - 如果您切換自獨立磁碟儲存區，而且磁碟儲存區上的資源是以確定控制登記，則 SETASPGRP 指令失敗，訊息 CPDB8EC，原因碼 2「緒具有未確定的異動」。此訊息後緊接訊息 CPF8E9。
  - 如果您切換自獨立磁碟儲存區但沒有資源以確定控制登記，則確定定義會移到您切換至的獨立磁碟儲存區。
  - 如果您切換自系統磁碟儲存區 (ASP 群組 \*NONE)，則確定控制不受影響。確定定義留在系統磁碟儲存區上。
  - 如果您使用通知物件，該通知物件必須和確定定義位於相同的獨立磁碟儲存區或獨立磁碟儲存區群組中。如果您將確定定義移至另一個獨立磁碟儲存區或獨立磁碟儲存區群組，通知物件也必須位於另一個獨立磁碟儲存區或獨立磁碟儲存區群組中。如果確定定義異常終止，則另一個獨立磁碟儲存區或獨立磁碟儲存區群組上的通知物件會被更新。如果在另一個獨立磁碟儲存區或獨立磁碟儲存區群組中找不到通知物件，則更新失敗，訊息 CPF8358。
- 於獨立磁碟儲存區轉接處理程序期間，會執行位於獨立磁碟儲存區的確定定義回復，此回復類似起始程式載入 (IPL) 回復。
- 於系統起始程式載入 (IPL) 期間，獨立磁碟儲存區中的確定定義不會被回復。
- 獨立磁碟儲存區的轉斷對確定定義具有下列影響：
  - 與獨立磁碟儲存區相關的工作結束。
  - 不允許在獨立磁碟儲存區上建立任何新的確定定義。
  - 位於獨立磁碟儲存區的確定定義變成無法使用。
  - 確定定義位於獨立磁碟儲存區上，但未連接至工作，請釋放異動範圍鎖定。
- 您無法使用 LU6.2 SNA 連線 (受保護的交談或分散式工作單元 (DUW)) 來從獨立磁碟儲存區資料庫連接至遠端資料庫。您可以使用不受保護的 SNA 交談來從獨立磁碟儲存區資料庫連接至遠端資料庫。



當確定控制對工作或緒是作用中時，存取確定定義所屬獨立磁碟儲存區或磁碟儲存區群組外的資料只能從遠端進行，就好像它是位於另一個系統上的資料一樣。當您發出 SQL CONNECT 陳述式來連接獨立磁碟儲存區上的關聯式資料庫 (RDB) 時，系統會建立一個遠端連線。

系統磁碟儲存區及基本磁碟儲存區不需要遠端連線就能夠取得獨立磁碟儲存區上資料的唯讀存取權。同樣地，獨立磁碟儲存區不需要遠端連線就能夠取得系統磁碟儲存區或基本磁碟儲存區上資料的唯讀存取權。

## XA 異動的注意事項

在 XA 環境中，每一個資料庫被視為個別的資源管理程式。當異動管理程式想要存取相同異動下的兩個資料庫時，它必須使用 XA 通信協定來對兩個資源管理程式執行兩階段確定。

因為每一個獨立磁碟儲存區都是個別的 SQL 資料庫，所以在 XA 環境中，每一個獨立磁碟儲存區也被視為個別的資源管理程式。要讓應用程式伺服器執行以兩個不同獨立磁碟儲存區為目標的異動，異動管理程式也必須使用兩階段確定通信協定。

有關獨立磁碟儲存區的詳細資訊，請參閱獨立磁碟儲存區主題。

## 確定控制的注意事項和限制

下列為確定控制的其他注意事項及限制：

### 資料庫檔案注意事項

- 如果您指定在確定控制下開啓共用檔，則該檔案的所有後續使用都必須在確定控制下開啓。
- 如果使用 LCKLVL(\*ALL) 對唯讀開啓的檔案指定 SEQONLY(\*YES) (明確透過高階語言程式或置換資料庫檔案 (OVRDBF) 指令)，則忽略 SEQONLY(\*YES) 而使用 SEQONLY(\*NO)。
- 在確定控制下所做的「記錄層次」變更會記錄於異動日誌中。這些變更可使用套用登載的變更 (APYJRNCHG) 指令套用到資料庫，或使用移除登載的變更 (RMVJRNCHG) 指令從資料庫移除。
- 檔案的未更新表格內容和更新後表格內容在確定控制下登載。如果僅指定登載檔案的更新後表格內容，系統也會自動登載在確定控制下發生的檔案變更之未更新表格內容。不過，由於並未擷取檔案的所有變更的未更新表格內容，所以您無法對這些檔案使用 RMVJRNCHG 指令。

### 物件和記錄層次變更的注意事項

- 在確定控制下所做的大部份物件層次變更會寫入異動日誌中，但不使用 APYJRNCHG 和 RMVJRNCHG 指令來套用或移除。不過，您可以使用傳送異動日誌登錄 (QJOSJRNE) API 來傳送其他事件的異動日誌登錄。在回復期間，您可以透過使用者撰寫的程式來擷取那些登錄並處理它們。
- 在確定控制下利用 SQL 所做的物件層次和記錄層次變更，將使用所要求程式在其中執行的啓動群組目前作用中的確定定義。如果工作層次和「啓動群組層次」確定定義皆非作用中，SQL 會啓動「啓動群組層次」確定定義。有關在確定控制下利用 SQL 執行變更的詳細資訊，請參閱 DB2 UDB for iSeries SQL 程式設計概念

### 一階段和兩階段確定注意事項

- 建立一階段遠端交談或連線時，其他位置的遠端交談或連線不被允許。如果建立了確定界限且所有資源已移除，則可變更位置。
- 如果您使用兩階段確定，則不需要使用提出遠端指令 (SBMRMTCMD) 指令來啓動確定控制，或在遠端位置執行其他任何確定控制作業。系統會為您執行這些功能。
- 若是一階段遠端位置，如果 SQL 位於呼叫堆疊中但遠端關聯式資料庫不在系統上，則 COMMIT 和 ROLLBACK CL 指令會失敗。若 SQL 不在呼叫堆疊中，則 COMMIT 和 ROLLBACK 指令不會失敗。

- 若是一階段遠端位置，則在對遠端資源做可確定變更之前，必須在來源系統上啟動確定控制。若 SQL 程式使用 \*NONE 以外的確定控制選項執行，系統會在連線時對來源系統的分散式資料庫 SQL 自動啟動確定控制。當第一個遠端資源被置於確定控制下時，系統會對目標系統啟動確定控制。

## 儲存注意事項

如果執行儲存的工作具有一或多個作用中確定定義，以及下列任何類型的可確定變更，則應避免儲存作業：

- 在要儲存之檔案庫中，檔案的記錄變更。若是邏輯檔案，則會檢查所有相關的實體檔案。
- 在要儲存的檔案庫內的任何「物件層次」變更。
- 已使用新增確定資源 (QTNADDCR) API 新增任何 API 資源，以及將「允許正常儲存程序」欄位設為預設值 N。

這可避免儲存作業儲存因部份異動而產生的儲存媒體變更。

物件鎖定和記錄鎖定可防止其他工作中確定定義的擱置中變更被儲存至儲存媒體。如果鎖定是在對 API 確定資源相關的一或多個物件做變更時取得的，這才適用 API 確定資源。

## 其他注意事項和限制

- 將您的系統升級成新版次之前，應先完成或取消所有擱置中重新同步。詳細資料，請參閱安裝軟體前確定兩階段確定整合性主題。
- 在確定或回轉期間，COMMIT 和 ROLLBACK 值顯示在 WRKACTJOB 函數欄位中。如果函數欄位中 COMMIT 或 ROLLBACK 存在了一段時間，則可能發生下列其中一個狀況：
  - 在確定或回轉需要重新同步期間資源失敗。除非重新同步完成或被取消，否則控制權不會傳回應用程式。
  - 此系統在唯讀期間已表決唯讀。除非起始確定的系統傳送資料給這個系統，否則控制權不會傳回應用程式。
  - 系統在確定期間表決「確定可以省略」。除非起始確定的系統傳送資料給這個系統，否則控制權不會傳回應用程式。

## 批次應用程式的確定控制

批次應用程式不一定需要確定控制。在某些情況下，批次應用程式可執行讀取輸入檔和更新主檔案的單一功能。不過，如果有必要於應用程式異常終止後重新啟動，您可以對此類型應用程式使用確定控制。

輸入檔是一種更新檔案，它使用記錄中的程式碼來指示記錄已被處理。此檔案及任何更新的檔案處於確定控制下。當程式碼出現在輸入檔時，它代表已完成的異動。程式會讀取輸入檔並略過具有完成碼的任何記錄。這可讓相同程式邏輯用於一般及重新啟動狀況。


如果批次應用程式含有與另一個批次應用程式相關的輸入記錄，而且含有切換或總計，則可使用通知物件來提供重新啟動的相關資訊。通知物件中的保留值可用來從輸入檔內最後一個確定異動中重新啟動處理程序。

如果輸入記錄與另一個輸入記錄相關，則可當作一個異動來處理。一個批次工作最多可鎖定 500 000 000 個記錄。您可以使用「查詢選項檔 (QAQQINI)」來減少此限制。使用變更查詢屬性 (CHGQRYA) 指令的 QRYOPTLIB 參數，對使用的工作指定「查詢選項檔」。使用「查詢選項檔」中的 COMMITMENT\_CONTROL\_LOCK\_LEVEL 值，作為工作的鎖定限制。

任何超出 2000 個鎖定的確定循環可能明顯降低系統效能。在其他方面，與交談式應用程式有相同的鎖定注意事項，但批次應用程式中鎖定的時間記錄長度可能不如交談式應用程式中重要。

## 兩階段確定控制

兩階段確定控制確保多個系統上的可確定資源保持同步。OS/400 根據 SNA LU 6.2 架構支援兩階段確定。關於系統用於兩階段確定的內部通信協定的詳細資訊，請參閱 *SNA Transaction Programmer's Reference for LU Type 6.2, GC30-3084-05*。OS/400 的所有支援版次都支援 SNA LU 6.2 的 Presumed Nothing 通信協定和 SNA LU 6.2 的 Presumed Abort 通信協定。

使用 TCP/IP 作為「分散式工作單元 (DUW)」DRDA 通信協定，也支援兩階段確定。若要使用 TCP/IP DUW 連線，那麼所有系統（應用要求程式和應用程式伺服器）必須為 V5R1M0 或更新版本。關於 DRDA 的其他資訊，請至 Open Group 網站  參閱 Open Group Technical Standard, *DRDA V2 Vol. 1: Distributed Relational Database Architecture*。

在兩階段確定之下，系統以兩階段執行確定作業：

- 在**準備階段**期間，資源管理程式向它的異動管理程式發出確定要求。異動管理程式通知它管理的其他資源以及其他異動管理程式，表示可以準備確定異動。所有資源管理程式必須回應表示它們已準備進行確定。這稱為**表決**。
- 在**確定階段**期間，起始確定要求的異動管理程式根據準備階段的結果決定執行什麼動作。如果順利完成準備階段而且所有參與者準備表決，異動管理程式會指示它管理的資源和其他異動管理程式確定異動。如果沒有順利完成準備階段，就會指示異動管理程式和資源管理程式回轉異動。

### 遠端資源的確定和回轉作業

當遠端資源在確定控制之下時，起始程式會將確定要求傳給所有遠端代理程式。此要求傳到整個異動程式網路。每一個代理程式以確定作業的結果作回應。

如果在準備階段期間發生錯誤，那麼起始程式會將回轉要求傳給所有代理程式。如果在確定階段期間發生錯誤，系統會嘗試儘量使最多位置處於確定狀態。這些嘗試可能導致試探混合狀態。關於可能狀態的其他資訊，請參閱兩階段確定控制的異動狀態。

錯誤會傳回起始程式來通知使用者。如果在啟動確定控制 (STRCMTCTL) 指令已指定預設異動日誌，就會寫入 C LW 登錄。如果發生錯誤會寫入這些登錄，即使有指定 OMTJRNE(\*LUWID) 也一樣。您可以使用這些登錄以及確定定義的錯誤訊息和狀態資訊，試著手動同步化可確定資源。

當遠端資源在確定控制之下時，起始程式會將回轉要求傳給所有遠端代理程式。此要求傳到整個異動程式網路。每一個代理程式以回轉作業的結果作回應。

關於其他資訊，請參閱下列項目：

- 在確定處理中的角色
- 兩階段確定控制的異動狀態
- 兩階段確定控制的確定定義

### 確定處理程序中的角色

如果異動的一個確定有一個以上的資源管理程式參與，則每一個資源管理程式在異動中扮演一個角色。資源管理程式負責確定或回轉異動期間的變更。依資源類型分類的資源管理程式如下：

#### FILE

資料庫管理程式

#### DDM

資料庫管理程式

**DDL**  
資料庫管理程式

**DRDA**  
通信異動程式

**LU62**  
通信異動程式

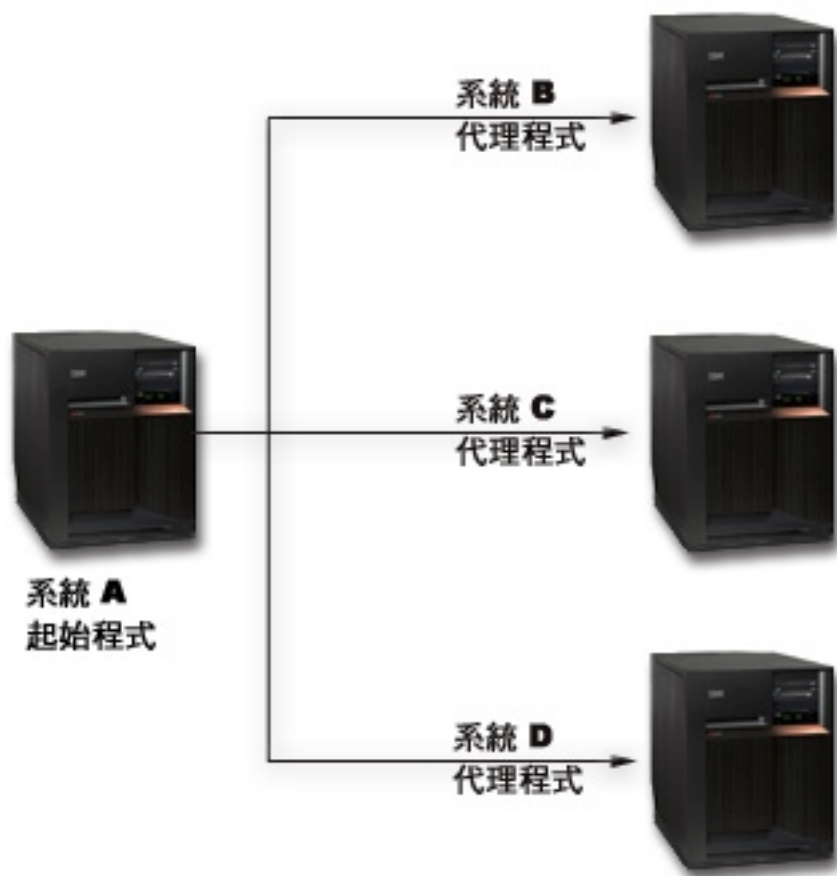
**API**  
API 跳出程式

下圖顯示異動中的基本角色。圖中顯示的結構稱為**異動程式網路**。結構可以是單層樹狀和多層樹狀。

#### 兩階段確定處理程序中的角色：單層樹狀

當「系統 A」上的一個應用程式發出確定要求時，「系統 A」上的資源管理程式會變成**起始程式**。對於透過 TCP/IP 的 DRDA 分散式工作單元，起始程式稱為**協調者**。

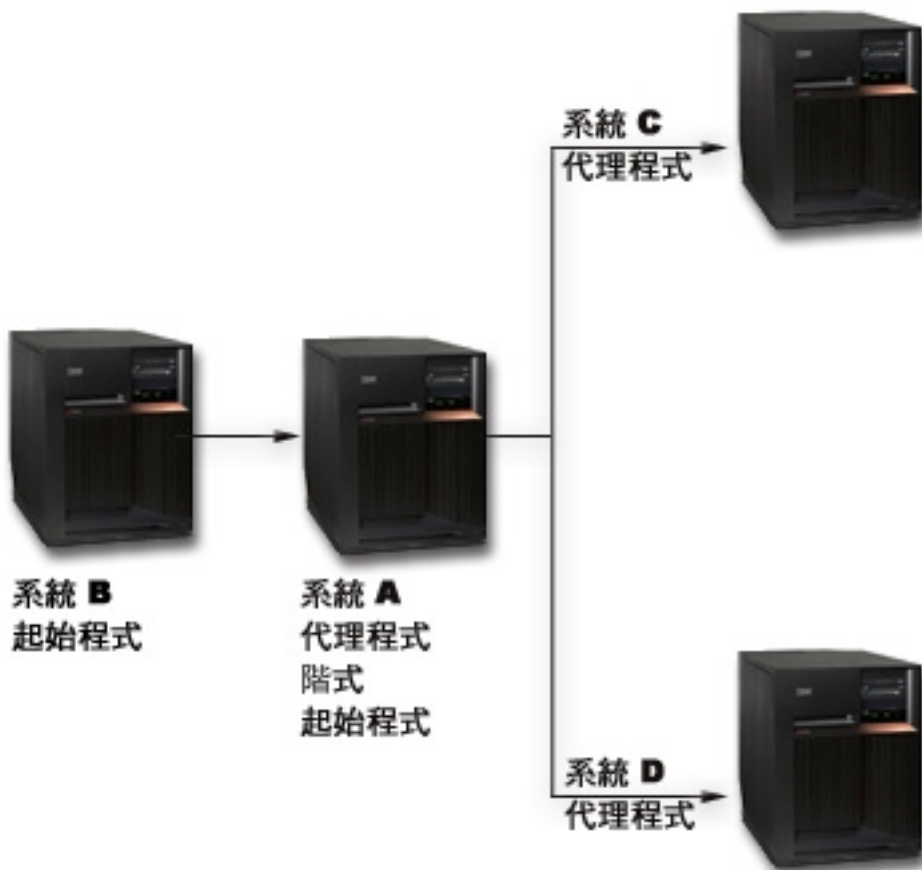
其他三個系統 (B、C 及 D) 的資源管理程式變成此異動的**代理程式**。對於透過 TCP/IP 的 DRDA 分散式工作單元，代理程式有時稱為**參與者**。



#### 兩階段確定處理程序中的角色：多層樹狀

如果應用程式使用 APPC 通信來執行兩階段確定，則在異動切換之間，系統之間的關係可能變動。下圖顯示相同的系統，「系統 B」的應用程式發出確定要求。此配置是多層樹狀。

此圖中的角色不適用於透過 TCP/IP 的 DRDA 分散式工作單元，因為不支援多層樹狀異動樹。



異動程式網路有另一個層次，因為「系統 B」不直接與「系統 C」和「系統 D」通信。「系統 A」中的資源管理程式現在扮演代理程式和階式排列起始程式的角色。

為了改進 LU6.2 兩階段異動的效能，起始程式可以指派**最後代理程式**角色給其中一個代理程式。最後代理程式不參與準備階段。在確定階段中，最後代理程式會先確定。若未順利確定最後代理程式，起始程式會指示其他代理程式回轉。

對於透過 TCP/IP 的 DRDA 分散式工作單元，協調者可以指派**重新同步伺服器**角色給一個參與者。當協調者發生通信失敗或系統失敗時，重新同步伺服器負責重新同步其他參與者。

### 兩階段確定控制的異動狀態

在屬於異動程式網路一部份的每一個位置建立確定定義。對於每一個確定定義，系統會記錄它的現行異動和先前異動的狀態。如果通信或系統失敗而岔斷異動，系統會使用此狀態來判斷要確定或回轉異動。如果有多個位置參與一個異動，可比較每一個位置的異動狀態來判斷正確動作（確定或回轉）。這種於位置之間通信以判斷正確動作的處理稱為**重新同步**。

下表顯示：

- 於異動期間可能發生的基本狀態。

- 可能發生的其他狀態。
- 若通信或系統失敗而岔斷異動的話，狀態是否需要重新同步。可能的值為：

**不需要**

每一個位置可以獨立做出正確決定。

**可能需要**

每一個位置可做出正確決定，但需要通知起始程序做了什麼決定。

**必要的**

必須判斷每一個位置的狀態之後才能做出正確決定。

- 通信或系統失敗所採取的動作。

狀態名稱	說明	若岔斷異動則重新同步	通信或系統失敗所採取的動作
<b>在兩階段確定處理期間的基本狀態：</b>			
重設 (RST)	從確定界限開始直到程式發出確定或回轉的要求。	不需要。	回轉擱置中的變更。
準備進度 (PIP)	起始程式已啟動準備階段。尚未表決所有位置。	可能需要。	回轉擱置中的變更。
備妥 (PRP)	已表決確定異動程式網路中的這個位置和下面的所有位置。這個位置尚未從起始程式收到確定通知。	必要的。	未完成的。根據重新同步處理的結果。
正在進行確定 (CIP)	已表決確定所有位置。起始程式已啟動確定階段。	必要的。	確定擱置中的變更。執行重新同步以確定所有位置已確定。如果另一個位置報告試探回轉，就會報告錯誤訊息。
已確定 (CMT)	所有代理程式已確定並將傳回對此節點的回答。	可能需要。	無。
<b>在兩階段確定處理期間的其他狀態：</b>			
最後代理程式擱置中 (LAP)	如果選取最後代理程式，那麼起始程式發生的這個狀態是介於 PIP 狀態與 CIP 狀態之間。起始程式已指示最後代理程式確定但尚未收到回應。	必要的	未完成的。根據重新同步處理的結果。
表決唯讀 (VRO)	此代理程式指出它沒有擱置中變更來回應準備階段。若允許表決唯讀狀態，那麼這個代理程式不併入已確定階段。	可能需要。	無。

狀態名稱	說明	若中斷異動則重新同步	通信或系統失敗所採取的動作
需要回轉 (RBR)	發生下列其中一項： <ul style="list-style-type: none"> <li>在執行確定作業之前代理程式發出回轉要求。</li> <li>發生異動失敗。</li> <li>使用 QTNRBRQD API 使異動處於需要回轉的狀態。</li> </ul> 在確定控制之下不允許異動程式執行其他變更。	可能需要。	回轉擱置中的變更。
<b>因為操作員動作或錯誤發生的狀態：</b>			
強制回轉	除了最後代理程式以外，經過操作員人工調整之後已經回轉異動程式網路中的這個位置和所有下面的位置。	可能需要	已回轉擱置中的變更。
強制確定	除了最後代理程式以外，經過操作員人工調整之後已經確定異動程式網路中的這個位置和下面的所有位置。	可能需要	已確定擱置中的變更。
試探混合 (HRM)	已確定部份資源管理程式。部份已回轉。使用操作員人工調整或發生系統錯誤。試探混合不會以狀態出現在確定定義畫面上。通知訊息會傳給操作員。	可能需要	操作員應在所有參與位置執行復置作業，使資料庫處於一致狀態。

## 兩階段確定控制的確定定義

啓動確定控制之後，您可以使用 QTNCHGCO (變更 確定選項) API 來變更異動的確定選項。根據您的環境和應用程式，變更確定選項可增進您的系統效能。

下列鏈結說明確定選項及使用它們的理由：

- 允許表決唯讀
- 不等待結果
- 指示可以忽略
- 不選取最後代理程式
- 信賴表決

如果您於 TCP/IP 連線上使用 DRDA 分散式工作單元，套用的唯一選項為「允許表決唯讀」。

**兩階段確定的確定定義：允許表決唯讀：** 一般而言，異動管理程式參與兩階段確定處理程序。若要增進確定處理程序效能，您可以設定異動中部份或全部位置來允許異動管理程式表決唯讀。如果異動期間的位置沒有可確定的變更，異動管理程式會在準備階段期間表決唯讀。位置不參與已確定階段。這可增進整體效能，因為在一個或多個遠端位置上沒有更新的異動期間，會省掉通常在已確定階段期間發生的通信流量。

於啓動確定控制之後，您可以使用變更確定選項 (QTNCHGCO) API 來變更允許表決唯讀選項為 Y。如果下列屬實，您就需要這麼做：

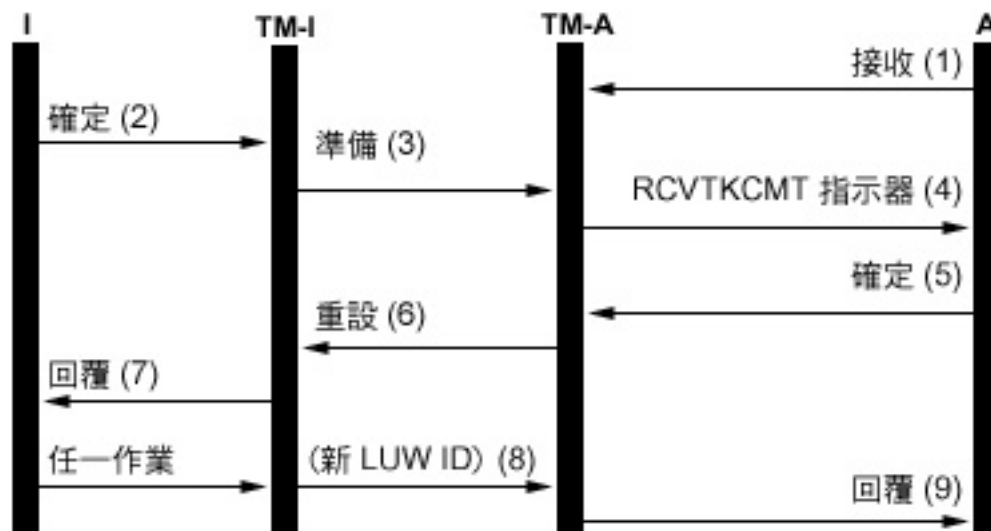
- 一或多個遠端系統沒有異動的任何可確定變更。
- 異動不根據上一個異動設定的檔案游標位置 (下一筆記錄)。當位置表決唯讀時，如果異動回轉的話，不會通知應用程式。由於位置已確定資料庫檔案的任何讀取作業，因此會移動游標位置。檔案游標位置通常只有在您執行循序處理時才重要。

如果您的確定定義設定為允許表決唯讀，則應用程式會等待另一個位置的下一個訊息流程。

允許表決唯讀選項是供主從架構型應用程式使用的。如果程式 A 的用途只是為滿足程式 I 的要求，並不要執行任何獨立的工作，建議接受程式 A 的表決唯讀選項。

### 代理程式表決唯讀時不含最後代理程式最佳化的確定處理流程

下圖顯示當代理程式表決唯讀時，應用程式發出不含最後代理程式最佳化的確定指示時，應用程式和異動管理程式之間的訊息流程。起始程式應用程式和代理程式應用程式兩者未察覺兩階段確定處理程序。在本圖中，括弧 ( ) 內的數字對應到下列說明中的編號項目。



#### 說明

- I** = 起始程式 (起始確定要求的應用程式)
- TM-I** = 起始程式的異動管理程式
- A** = 代理程式 (接收確定要求的應用程式)
- TM-A** = 代理程式的異動管理程式

下列為代理程式表決唯讀時不含最後代理程式最佳化的事件正常處理程序之說明。此說明基本流程。當異動程式網路具多重層次或發生錯誤時，這一連串的事件可能變得更加複雜。

1. 應用程式 A 發出一個接收要求，指示它已備妥可接收程式 I 發出的要求。
2. 起始程式 (I) 發出一個確定指示。
3. 起始程式的異動管理程式 (TM-I) 扮演這個異動的起始程式角色。它藉由傳送準備訊息給參與異動的其他所有位置來開始準備階段。
4. 其他每個位置的異動管理程式扮演代理程式 (TM-A) 的角色。TM-A 通知應用程式 A 已收到確定要求。若是 ICF 檔，通知的形式為將 Receive Take Commit (RCVTKCMT) ICF 指示器設定為開啓。
5. 應用程式 A 發出確定指示 (或回轉指示) 作為回應。此為應用程式的表決。



6. 若應用程式 A 使用「變更確定選項 API (QTNCHGCO)」將「允許表決唯讀」確定選項設為 Y，但異動期間代理程式上沒有任何變更，則代理程式 (TM-A) 回應一個重設訊息給起始程式 (TM-I)。代理程式將沒有已確定階段。
7. 傳送回覆給應用程式 (A)，以指示異動於代理程式 TM-A 上完成。
8. 下次起始程式 (TM-I) 發出任何訊息 (資料流程或確定指示) 給代理程式 (TM-A) 時，TM-I 會隨訊息送出其現行異動 ID。理由是如果在確定作業期間 TM-I 和另一系統之間發生通信失敗，則可能已在 TM-I 上產生新異動 ID。
9. 傳送回覆給應用程式 (A)，以指示異動在代理程式 TM-A 上完成。回覆會延遲至收到下一個訊息為止，因為在應用程式 A 啟動下一個異動之前，必須從 TM-I 接收新異動 ID。

有關兩階段確定控制的詳細資訊，請參閱確定處理程序中的角色和 兩階段確定控制的異動狀態。

**兩階段確定的確定定義：不等待結果：** 當確定作業期間發生通信或系統失敗而必須重新同步時，預設是在確定作業完成之前，先等待重新同步完成。

**註：** 如果您透過 TCP/IP 連線使用 DRDA 分散式工作單元，則不適用「不等待結果」選項。透過 TCP/IP 連線的 DRDA 分散式工作單元一律不等待結果。

如果下列條件成立，您應該考慮變更此行為：

- 參與的應用程式彼此獨立。
- 您的程式邏輯不需要先前異動的結果來確定資料庫檔案是否仍然同步。

在您啟動確定控制之後，您可以使用 QTNCHGCO (變更確定選項) API 來指定確定定義不等待重新同步的結果。如果您設定等待結果選項為 N (否)，則系統會使用資料庫伺服器工作 (QDBSRVnn)，以非同步方式來處理重新同步。

**註：** 這些資料庫伺服器工作是在起始程式載入 (IPL) 處理期間啟動的。如果您變更確定控制的選項，則這對系統啟動的工作數沒有影響。

本主題對於解析的等待結果選項僅參照兩個值，Y (是) 和 N (否)。實際上還有兩個值可以指定，L (「是」或「繼承自起始程式) 和 U (「否」或「繼承自起始程式」)。當您使用這些值時，每一個確定作業期間使用的實際值會被系統解析為「是」或「否」。QTNCHGCO (變更確定選項) API 主題詳細說明這些值。

**註：** 如果起始程式和代理程式皆支援擅自放棄，則起始程式的值只能被代理程式所繼承。

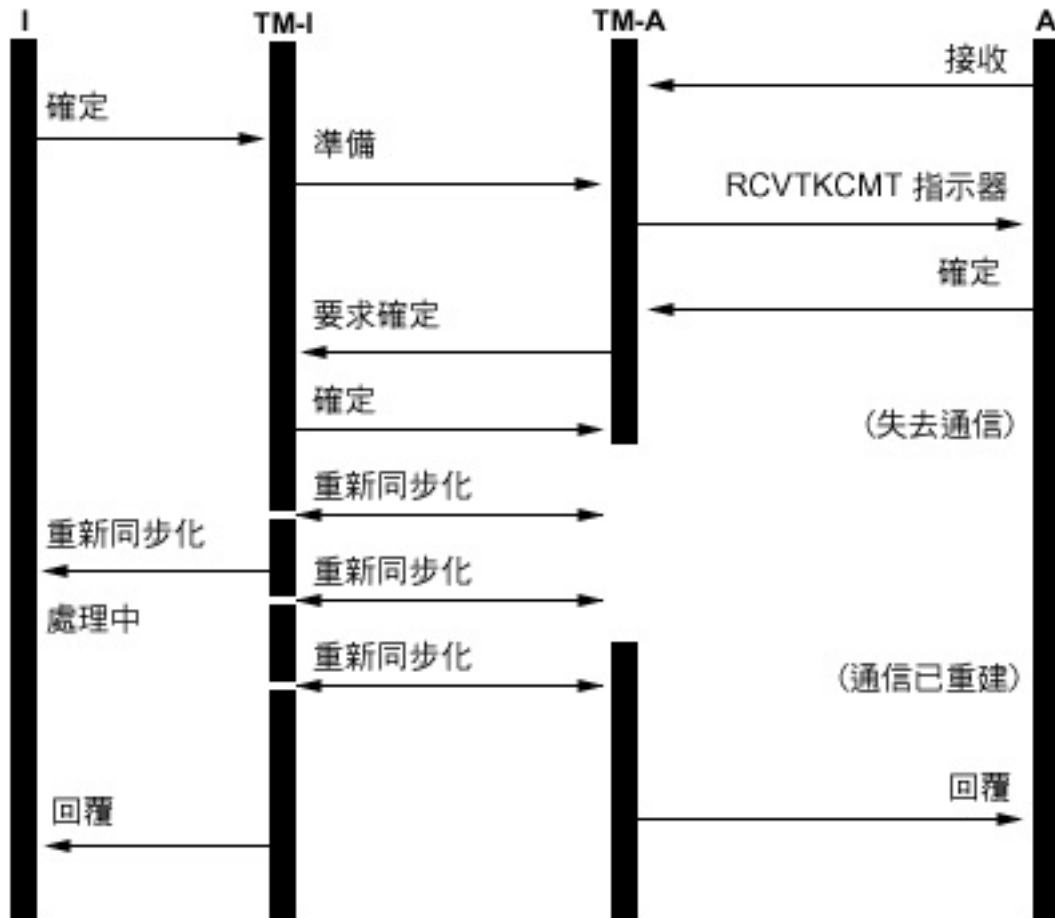
等待結果 (WFO) 選項不影響正常無誤的確定處理程序。如果發生錯誤，WFO 選項會根據下列狀況，決定應用程式是否等待重新同步：

- 如果解析的 WFO 選項是 Y (是)，則應用程式等待重新同步的結果。
- 如果解析的 WFO 選項是 N (否) 且支援擅自放棄通信協定的位置在準備信號或回轉期間發生通信失敗，則不執行重新同步，確定定義會回轉。
- 如果確定定義有疑問 (異動狀態為「備妥」或「最後代理程式擱置中」)，應用程式將等待重新同步的結果，不論解析的 WFO 值為何。關於有疑問的確定定義的進一步資訊，請參閱兩階段確定控制的異動狀態。
- 如果解析的 WFO 選項是 N，且條件 2 或 3 皆不成立，系統會嘗試重新同步化一次。如果未順利完成，系統會向應用程式發出信號 STATUS 訊息 CPF83E6，指出重新同步正在進行中。

因為 CPF83E6 是 STATUS 訊息，只有在應用程式監督它時才有作用。您的應用程式通常可以將此訊息視為參考訊息。在解決失敗之前，參與異動的系統會嘗試重新同步化異動。這些後續的重新同步嘗試是在資料庫伺服器工作中執行。如果資料庫伺服器工作中有一個後續執行的重新同步嘗試失敗，訊息 CPI83D0 會傳送到 QSYSOPR。

### 等待結果「是」

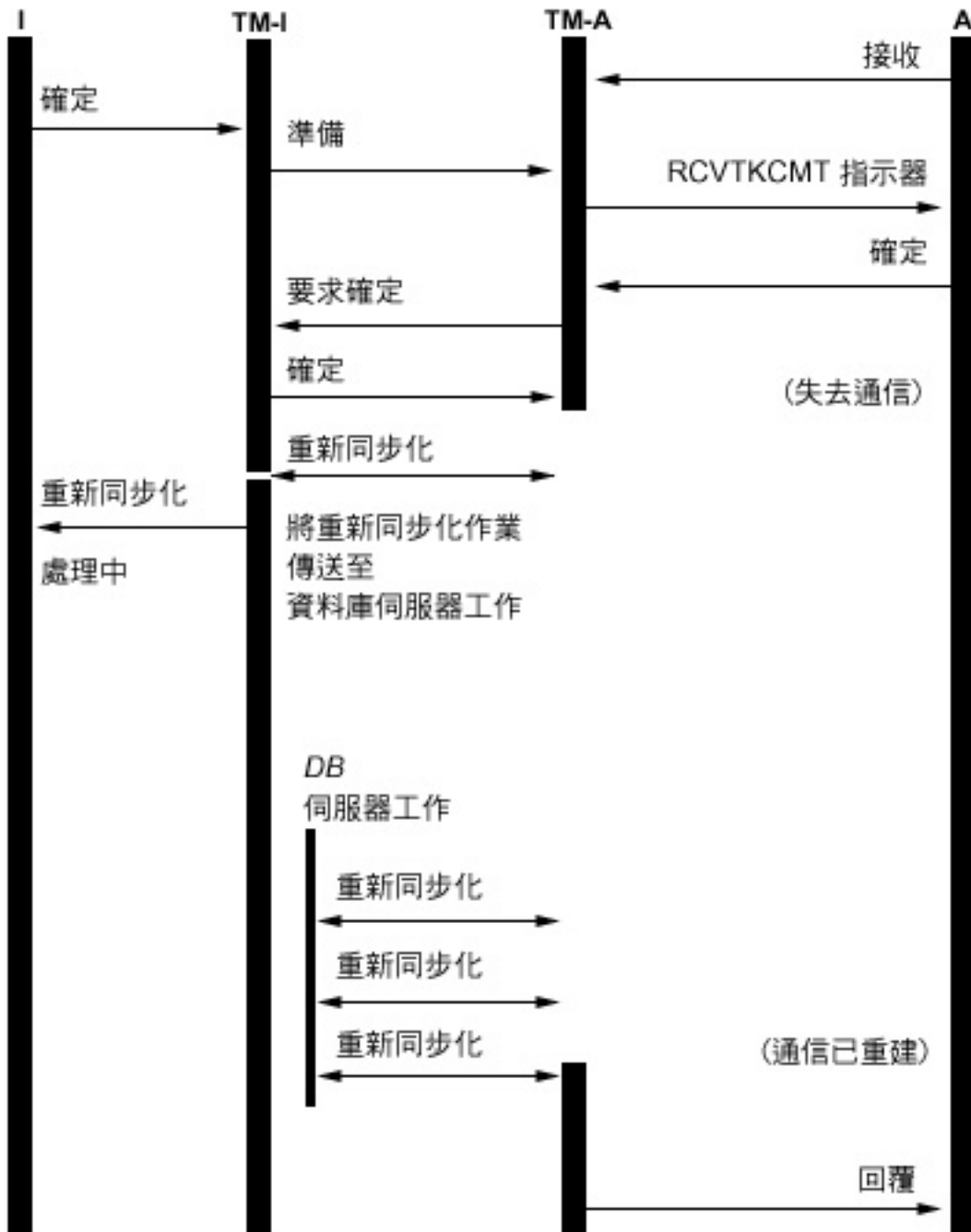
在下圖中，起始程式 (I) 的確定定義使用預設值 Y (是) 來設定等待結果選項。當 TM-I 和 TM-A 之間失去通信時，應用程式 A 和應用程式 I 會等待直到異動重新同步為止。



### 等待結果「否」

在下圖中，起始程式的確定定義將解析的 WFO 設定為 N (否)。TM-A 符合之前清單中的條件 3，TM-I 符合條件 4。在嘗試與 TM-A 重新同步化之後，控制權會傳回給應用程式 I。一個資料庫伺服器工作嘗試重新同步化。當確定要求已順利完成時，應用程式 I 不會收到傳回指示器。在重新建立通信之前，控制權不會傳回給代理應用程式 (A)。這要根據失敗的時間而定。在本例中，通信失敗是在收到起始程式的確定訊息前發生，所以 TM-A 無法決定是否確定或回轉。當異動管理程式有疑問時，則保留控制，直到完成重新同步，不論系統上解析的 WFO 值為何。

在完成重新同步前，如果您要讓所有系統上的應用程式繼續執行，您必須將所有系統上解析的 WFO 選項變更為 N (否)，或設定起始程式為 N，其餘系統設定為 U (「否」或「繼承自起始程式」)。切記，當異動管理程式無法決定確定或回轉時，解析的 WFO 選項會被忽略，在傳回控制權之前會一直等待，直到完成重新同步為止。



當連接到關聯式資料庫 (DB2) 且尚未啟動受保護的交談時，系統會隱含地將等待結果值變更為 N。理由是當等待結果值為 N 且遠端系統支援擅自放棄時，確定作業的效能會改進。等待結果值的隱含變更只對 DRDA 和 DDM 應用程式執行。APPC 應用程式使用預設等待結果值 Y，除非它們呼叫 QTNCHGCO API 變更它。

**兩階段確定的確定定義：指示確定可以省略：** 通常，異動程式網路中任何位置上的異動管理程式會參與每一個確定或回轉作業。為了改進效能，您可以設定異動中的部份或所有位置，讓異動管理程式指示可以省略。

註： 如果您透過 TCP/IP 連線使用 DRDA 分散式工作單元，則不適用「指示確定可以省略」選項。

如果異動期間沒有通信流量傳送到位置，則執行確定或回轉作業時會省略此位置。這可以改進整體效能，因為在沒有傳送資料到一或多個遠端位置的異動期間，確定或回轉時通常發生的通信流量會被省掉。

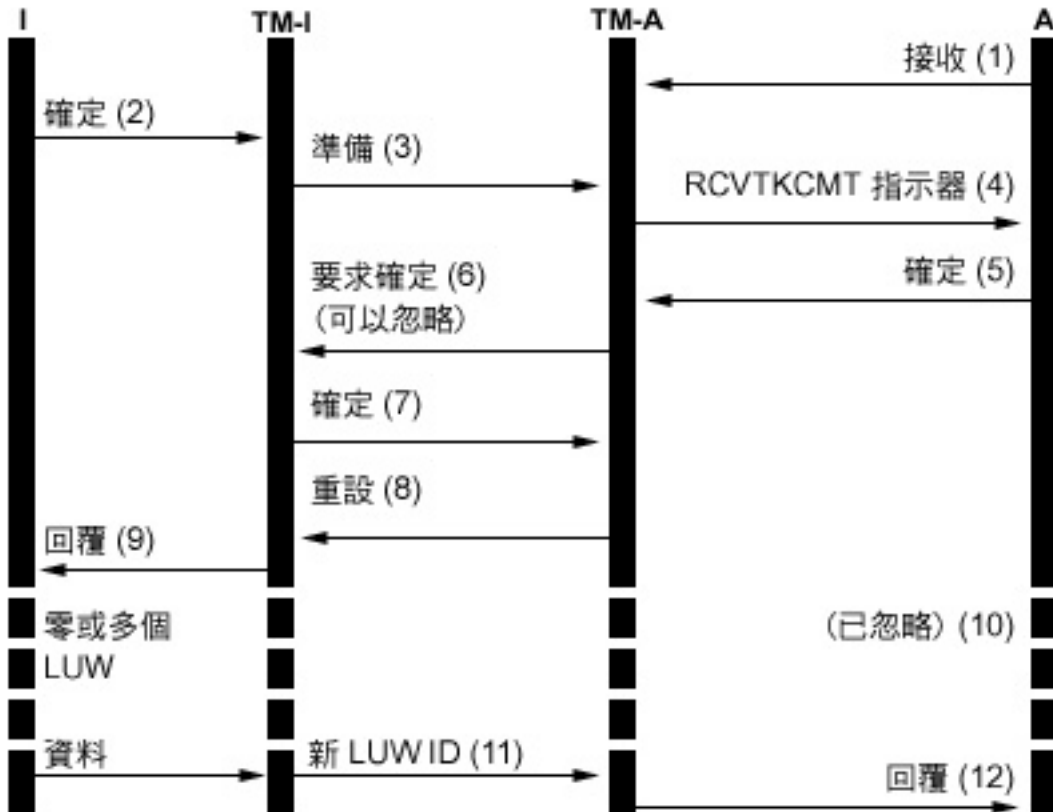
在您啟動確定控制之後，您可以使用「變更確定選項」(QTNCHGO) API 將「確定省略」選項變更為 Y (是)。如果一或多個遠端系統通常不參與異動，則您可以這樣做。

如果您的確定定義是設定為指示確定可以省略，應用程式會等待下一個來自另一個位置的訊息流程。

「確定可以省略」選項是針對主從架構性質的應用程式而提供的。如果程式 A 的唯一目的是滿足來自程式 I 的要求，則適合對程式 A 設定「確定可以省略」選項。

### **代理程式表決確定可以省略時，不執行最後代理程式最佳化的確定處理流程**

下圖顯示當應用程式發出確定指示但不執行最後代理程式最佳化且代理程式指示確定可以省略時，應用程式和異動管理程式之間的訊息流程。起始程式和代理程式兩者都沒有察覺兩階段確定處理程序。括弧 () 中的數字對應下列說明的編號項目。



**說明**

- I** = 起始程式 (起始確定要求的應用程式)
- TM-I** = 起始程式的異動管理程式
- A** = 代理程式 (接收確定要求的應用程式)
- TM-A** = 代理程式的異動管理程式

下列是當不執行最後代理程式最佳化且代理程式表決確定可以省略時，關於事件正常處理的說明。這說明一個基本流程。當異動程式網路有多重層次或發生錯誤時，這一連串的事件可能會非常複雜。

1. 應用程式 A 發出接收要求來指出它已準備好接收應用程式 I 的要求。
2. 起始程式 (I) 發出確定指示。
3. 起始程式的異動管理 (TM-I) 扮演此異動的起始程式之角色。它經由傳送準備訊息到參與異動的其他所有位置來啟動準備階段。
4. 其他每一個位置的異動管理程式扮演代理程式的角色 (TM-A)。TM-A 通知應用程式 A 已接收一個確定要求。對於 ICF 檔案，通知的形式為將的 Receive Take Commit (RCVTKCMT) ICF 指示器設定為開啓。
5. 應用程式 A 發出確定指示 (或回轉指示) 來回應。此為應用程式的表決。
6. 如果應用程式 A 已使用「變更確定選項」 API (QTNCHGCO) 來設定「確定可以省略」選項為 Y，則代理程式 (TM-A) 以要求確定訊息回應起始程式 (TM-I) 時會傳送一個指示器。

註： 在下一個順利完成確定作業之前，對「確定可以省略」選項所做的變更不會生效。

7. 當起始程式 (TM-I) 接收所有表決時，TM-I 會傳送確定訊息。這會啟動確定階段。
8. 每一個代理程式 (TM-A) 會以重設訊息來確定和回應。

9. 一個回覆會傳送到應用程式 (I) 來指出起始程式方面已完成異動。
10. TM-1 上可能發生任何數量的異動，但皆不需要變更 TM-A 或 TM-A 的資料。TM-A 不併入這些異動中。
11. 起始程式 (TM-I) 下次發出訊息到代理程式 (A) 時，新的異動 ID 會隨著訊息一併傳送。如果起始程式在傳送訊息給代理程式之前執行任何確定或回轉作業，則這些作業期間不會傳送訊息給代理程式 (這些確定或回轉會「省略」代理程式)。當代理程式被省略時，因為起始程式方面可能已確定或回轉一或多個異動，所以在下一個訊息傳送到代理程式時，起始程式必須傳達其現行異動 ID。
12. 一個回覆會傳送到應用程式 (A) 來指出原始確定已完成且它正在參與現行異動。

**兩階段確定的確定定義：不選取最後代理程式：** 依預設，起始程式的異動管理程式在確定作業期間任意選取任何代理程式作為最後代理程式。

**註：** 如果您透過 TCP/IP 連線使用 DRDA 分散式工作單元，則不套用「不選取最後代理程式」選項。

在多層次樹狀結構的情況中，任何被其起始程式選取為最後代理程式的代理程式也任意選取本身的最後代理程式。在確定作業期間選取了最後代理程式可增進效能，因為起始程式及其最後代理程式之間的兩個通信流程省略了 (這些系統的準備階段被省略了)。

不過，一旦起始程式傳送要求確定給它的最後代理程式，則必須等到接收最後代理程式的表決後才能繼續。這與確定定義的等待結果值無關。在正常無誤的確定處理程序期間，這不是問題。但是，如果在這個視窗期間發生錯誤，則在重新同步完成之前，起始程式無法繼續。如果起始應用程式在終端機上處理使用者要求，這可能要考量到可用性。

當發生這種錯誤時，您應該考慮到正常確定作業期間增進效能是否比可用性影響重要。請注意：如果傳送要求確定給最後代理程式之前發生錯誤，則 LUW 會立即回轉且起始程式將不等待。因此，錯誤造成起始程式等待的機率相當小，所以這種錯誤應不常發生。

如果您覺得可用性的影響比不上增進效能，您可以變更確定定義為不選取最後代理程式。啟動確定控制之後，您可以使用變更確定選項 (QTNCHGCO) API 來變更准許最後代理程式選項為 N。

**信賴表決對於確定處理流程的影響：** 信賴表決是一種增進效能的最佳化功能，能在確定作業之後較早返回起始程式應用程式，並且在確定作業期間刪除一則訊息。對於透過 TCP/IP 的 DRDA 分散式工作單元來說，並沒有明確的信賴表決最佳化功能。不過，OS/400 不會要求重設 (忘記) TCP/IP 連線的確認。因此對於 TCP/IP 連線來說，一律隱含重設 (忘記)。

啟動確定控制之後，您可以使用變更確定選項 (QTNCHGCO) API，將接受信賴表決選項變成 Y。

可以將信賴表決視為代理程式對它的起始程式的一項承諾，該承諾保證萬一發生通信失敗而代理程式有疑問時，代理程式不會採取任何試探法。在確定的準備階段，使用信賴表決最佳化功能的代理程式傳送指示符給起始程式。如果起始程式也使用信賴表決最佳化功能，那麼它會傳送指示符給代理程式表示不需要重設，來回應確定訊息。這個動作省掉重設訊息，並允許一旦傳送確定訊息，異動管理程式就返回起始程式上的應用程式。

如果下列狀況為真，那麼您應考慮使用信賴表決最佳化功能：

- 除非無法修復失敗，否則當發生系統或通信失敗時，未完成的代理程式不太可能採用試探法。
- 程式邏輯不需要先前異動的結果即可確保資料庫檔案仍保持同步。

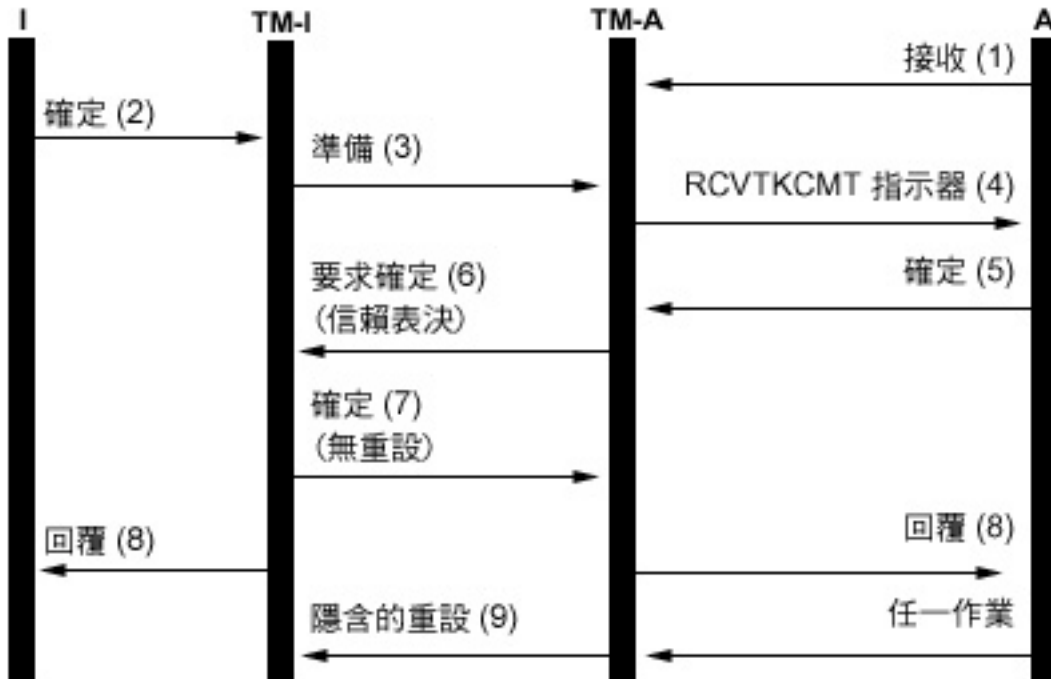
唯有下列狀況為真時，OS/400 才會使用信賴表決最佳化功能：

- 起始程式和代理程式位置支援確定控制的擅自放棄層次。
- 起始程式位置接受來自代理程式的信賴表決指示。在 OS/400 起始程式，這取決於兩個確定選項的值：

- 「等待結果」確定選項的值必須為「否」(「是」為預設值)。
- 「接受信賴表決」確定選項的值必須為「是」(「是」為預設值)。
- 在準備階段，代理程式位置做信賴表決。OS/400 代理程式一律做信賴表決。這是因為只能透過手動程序採用試探法，這警告採用試探法可能有反效果。

### 含有信賴表決最佳化功能的確定處理流程

下圖顯示當使用信賴表決最佳化功能時應用程式與異動管理程式之間的訊息流程。起始應用程式和代理應用程式兩者都不知道兩階段確定處理。圖中以括弧 ( ) 括住的數字對應下列說明中的編號項目。



#### 說明

- I** = 起始程式 (起始確定要求的應用程式)
- TM-I** = 起始程式的異動管理程式
- A** = 代理程式 (接收確定要求的應用程式)
- TM-A** = 代理程式的異動管理程式

下面是當代理程式做信賴表決時沒有最後代理程式最佳化情況下對事件正常處理的說明。此說明基本流程。當異動程式網路有多重層次或發生錯誤時，一連串的事件會變得更複雜。


1. 應用程式 A 執行接收要求來表示它已準備接收來自程式 I 的要求。
2. 起始程式 (I) 發出確定指示。
3. 起始程式異動管理 (TM-I) 擔任這個異動的起始程式角色。它傳送準備訊息給參與異動的所有其他位置來開始準備階段。
4. 其他所有位置的異動管理程式擔任代理程式角色 (TM-A)。TM-A 通知應用程式 A 已收到確定的要求。對於 ICF 檔案來說，通知的形式是將 Receive Take Commit (RCVTKCMT) ICF 指示符設定為開啓。
5. 應用程式 A 發出確定指示 (或回轉指示) 來回應。此為應用程式的表決。
6. 代理程式 (TM-A) 以要求確定訊息回應起始程式 (TM-I)。OS/400 系統傳送含有要求確定的信賴表決指示符。

7. 當起始程式 (TM-I) 接收所有表決時，TM-I 會傳送確定訊息。如果「等待」結果確定選項為 N (否) 而「接受」信賴表決確定選項為 Y (是)，那麼重設指示符不會隨確定訊息一起傳送。此告訴代理程式回應確定時不需要重設訊息。
8. 異動完成。會將回覆傳給應用程式 (I 和 A)。此回覆指出順利完成確定作業。如果因為在收到確定訊息之前採用試探法而使得系統 A 發生試探損壞，則不會通知應用程式 I。相反地，訊息會傳給 QSYSOPR 訊息佇列。不過，應用程式 A 會收到試探損壞指示。
9. 下次代理程式 (TM-A) 若傳送任何訊息給起始程式 (TM-I) (可能是資料流程或確定指示)，隱含的重設指示符會隨訊息一起傳送，以通知 TM-I 表示 TM-A 順利完成確定。這是因為在 TM-I 確認 TM-A 在步驟 7 順利收到確定訊息之前，它必須保留關於已完成異動的資訊。

## 確定控制的 XA 異動支援

DB2 UDB for iSeries 可參於 X/Open 廣域異動。Open Group 為異動工作定義一個業界標準模型，使對無關的資源所做的變更成為單一廣域異動的一部份。例如對兩家供應商提供的資料庫所做的變更。此模型稱為「X/Open 分散式異動處理」模型。下列出版品詳細說明「X/Open 分散式異動處理」模型：

- X/Open Guide, February 1996, Distributed Transaction Processing: Reference Model, Version 3 (ISBN:1-85912-170-5, G504), The Open Group.
- X/Open CAE Specification, December 1991, Distributed Transaction Processing: The XA Specification (ISBN:1-872630-24-3, C193 or XO/CAE/91/300), The Open Group.
- X/Open CAE Specification, April 1995, Distributed Transaction Processing: The TX (Transaction Demarcation) Specification (ISBN:1-85912-094-6, C504), The Open Group.

您應該熟悉這些書籍中的資訊之後 (尤其是 XA Specification)，再嘗試使用 DB2 UDB for iSeries 提供的 XA 異動支援。您可以在 Open Group 網站  找到這些書籍

DTP 模型有 5 個元件：

### 應用程式 (AP)

透過指定涉及資源 (例如資料庫) 的一連串作業，實施一般使用者想要的功能。它會定義廣域異動的開始和結束，存取異動界限內的資源，且通常會決定是要確定或回轉每一個異動。

### 異動管理程式 (TM)

管理廣域異動和協調啟動它們的決定，並確定它們或回轉它們以確保單一異動完成。在某元件失敗之後，TM 也以 RM 協調回復活動。

### 資源管理程式 (RM)

管理電腦的共用資源之已定義部份，例如資料庫管理系統。AP 使用每一個 RM 定義的介面以及 RM 提供的介面來完成異動。

### 通信資源管理程式 (CRM)

可讓某模型案例存取位於現行 TM 網域之內或之外的另一個案例。CRM 在 DB2 UDB for iSeries 範圍之外，此處不予討論。

### 通信協定

CRM 用來與彼此通信的通信協定。這是在 DB2 UDB for iSeries 範圍之外，此處不予討論。



「XA 規格」是 DTP 模型的一部份，它說明 DTP 模型的 TM 和 RM 元件使用的一組介面。DB2 UDB for iSeries 以一組 UNIX 樣式 API 和跳出程式來實作這些介面。有關這些 API 的詳細文件以及如何使用 DB2 UDB for iSeries 作為 RM 的其他資訊，請參閱 XA API。

## 「iSeries 領航員」和 XA 異動

「iSeries 領航員」支援以**廣域異動**來管理 XA 異動

廣域異動可能包含發生於 DB2 UDB for iSeries 內外的變更。外部「異動管理程式」使用 Open Group XA 架構或另一個類似架構來協調廣域異動。應用程式使用「異動管理程式」提供的介面確定或回轉廣域異動。「異動管理程式」使用 XA 架構或另一個架構定義的確定通信協定來完成異動。當參與廣域異動時，DB2 UDB for iSeries 是作為「XA 資源管理程式」。廣域異動有兩種類型：

- **異動範圍鎖定**：代表異動獲得的鎖定其範圍限於該異動。異動可以從一個工作或緒移到另一個工作或緒。
- **工作範圍鎖定**：代表異動獲得的鎖定其範圍限於該工作。異動無法從啟動它的工作移動。

如果正在對常駐在本端系統的資料庫執行 XA 異動，您應該對異動範圍鎖定使用 XA API。這些 API 對工作範圍鎖定的限制比 XA API 少，而且於下列狀況下提供更好的效能：

- 如果曾使用多個 SQL 連線在單一 XA 異動分支中工作。
- 如果使用單一 SQL 連線在多個並行 XA 異動分支中工作。

在這些狀況下，當您對「工作範圍鎖定」使用 XA API 時，必須啟動個別的工作來執行 XA 異動分支。

如果正在對常駐在遠端系統的資料庫執行異動，您必須對「工作範圍鎖定」使用 XA API。

## XA 異動的注意事項

應瞭解下列注意事項和限制之後再使用 DB2 UDB for iSeries 作為 RM。「緒」這個術語是指不具有緒功能的工作或是指具有緒功能之工作內的單緒。

除非另有指定，否則下列注意事項套用於具有異動範圍鎖定的異動以及具有工作範圍鎖定的異動。

## DB2 UDB for iSeries 注意事項

- 只能在那些於 SQL 伺服器模式中執行的工作中執行 XA 異動。此限制的其中一個影響是在 XA 異動期間，變更 DB2 UDB for iSeries 時該應用程式只能使用 SQL 介面。如果在一個已不於 SQL 伺服器模式中執行的工作中使用 db2xa\_open() API，就會隱含地啟動 SQL 伺服器模式。請參閱確定控制的 SQL 伺服器模式和緒範圍異動
- 任何在 XA API 呼叫期間為 DB2 UDB for iSeries 所偵測到的錯誤，都會根據 XA 規格透過回覆碼報告。當回覆碼包含的錯誤訊息意義不清楚時，工作日誌會提供診斷訊息。

## 內含的 SQL 注意事項

- 若要對 XA 異動使用「結構化查詢語言 (SQL)」連線，您必須使用 db2xa\_open() 應用程式介面 (API) 之後才能建立 SQL 連線。將連接至的關聯式資料庫必須透過 Xainfo 參數傳給 db2xa\_open() API。於遞送連線工作中所使用的使用者設定檔和密碼可以傳至 db2xa\_open() API。若沒有傳送它，那麼在嘗試連線時設定檔會預設成已指定的或預設的設定檔。
- 若使用內含的 SQL 執行 XA 異動，那麼為每一個連線執行的工作會遞送到不同工作，即使在相同緒中建立的連線也一樣。此與沒有 XA 的 SQL 伺服器模式不同，在該模式下為單緒中的所有連線執行的工作會遞送到相同工作。這是因為對於每一個資源管理程式案例來說，XA 規格需要個別的準備、確定或回轉呼叫。

註：下面注意事項只套用於具有工作範圍鎖定的異動。

- 如果使用內含的 SQL 執行 XA 異動，那麼每一個緒的每一個關聯式資料庫只能建立一個連線。每當緒與異動分支沒有主動產生關聯時，透過緒的其中一個連線要求的工作會使 RM 使用 TM 的 ax\_reg() 跳出程式，來決定工作是否要啟動、回復或結合異動分支。

若工作是要啟動異動分支，就會透過緒與對應關聯式資料庫的連線來執行工作。

如果工作是要結合異動分支，就會透過在啟動異動分支的緒中建立的對應關聯式資料庫連線重新遞送工作。請注意：系統不會強制該連線的使用者設定檔一定要與結合緒的連線的使用者設定檔相同。TM 負責保證此沒有安全上的顧慮。典型的 TM 對所有連線使用相同使用者設定檔。此使用者設定檔獲授權存取 TM 管理的所有資料。由 TM 或 AP 管理存取這個資料的進一步安全，而不使用標準 iSeries 安全機制。

- 如果工作是回復異動分支，就會根據是透過啟動或結合異動分支建立暫停的異動分支連結，來決定所使用的連線。

在使用 db2xa\_end() API 暫停或結束緒與該異動分支的連結之前，後續的工作是透過相同連線執行。

### CLI 注意事項

- 如果使用 CLI 執行 XA 異動，那麼在使用 db2xa\_open() API 之後可以在相同緒建立一個以上的連線。只要其他緒先使用含有相同 Xainfo 參數值的 db2xa\_open() API，這些其他緒就可以使用連線來執行 XA 異動。

註：下面注意事項只套用於具有工作範圍鎖定的異動。

- 若使用 CLI 執行 XA 異動，那麼用來啟動異動分支的連線必須用於該異動分支的所有工作。若另一個緒要結合異動分支，那麼用來啟動異動分支的連線的連線控點必須傳至結合緒，使得它能透過相同連線執行工作。同樣地，如果緒是要回復異動分支，就必須使用相同連線。

註：下列項目套用於具有「異動範圍」鎖定和「工作範圍」鎖定的異動。

因為 CLI 連線控點不能使用於不同工作，所以當使用 CLI 時結合功能僅限於在啟動異動分支的相同工作中正在執行的緒。

### 遠端關聯式資料庫注意事項

註：遠端關聯式資料庫的注意事項只適用於具有「工作範圍」鎖定的異動。

- 唯有關聯式資料庫常駐在支援「分散式工作單元 (DUW)」DRDA 連線的系統，才支援遠端關聯式資料庫的 XA 連線。這包含透過 SNA LU6.2 交談執行 DRDA 的系統。當使用 TCP/IP 連線執行 DRDA 時，此亦包含使用 V5R1 的系統。
- 使用 XA 結合功能之前，必須於結合緒中使用 db2xa\_open() API。必須在啟動異動分支的緒及結合緒中的 db2xa\_open() API 指定相同的關聯式資料庫名稱和 RMID。嘗試結合時如果異動分支為作用中，便會暫停執行結合緒。於暫停作用中緒或結束其與異動分支的連結前仍暫停執行結合緒。

### 回復注意事項

- 當異動分支處於備妥狀態時，如果需要強制異動分支確定或回轉，可使用提供給所有確定定義的手動試探確定和回轉支援。關於明細，請參閱何時強制確定和回轉以及何時取消重新同步。

### 異動分支注意事項

- 關於 XA 異動分支的資訊，請參閱為「iSeries 領航員」和「使用工作 (WRKJOB)」、「顯示工作 (DSPJOB)」以及「使用確定定義 (WRKCMDFN)」指令所顯示的確定控制資訊。TM 名稱、異動分支狀態、異動 ID 和分支限定元全部都會顯示出來。可以使用指令 WRKCMDFN JOB(\*ALL) STATUS(\*XOPEN) 或於「iSeries 領航員」中顯示**廣域異動**，來顯示與所有目前作用中 XA 異動相關的確定定義。

註：下列項目只適用於具有「工作範圍鎖定」的異動。

- 若使用 db2xa\_end() API 來暫停或結束緒和現存異動分支之間的連結，則緒可以啟動新的異動分支。若用來啟動新異動分支的連線先前已經用來啟動另一個異動分支，且該緒與該異動分支的連結已由 db2xa\_end() API

結束或暫停，則可以啟動新的 SQL 伺服器工作。唯有 db2xa\_commit() 或 db2xa\_rollback() API 尚未完成第一個異動分支，才需要新的 SQL 伺服器工作。於此狀況下，另一個完成訊息 SQL7908 會傳給識別新 SQL 伺服器工作的工作日誌，就像建立連線時會識別連線的原始 SQL 伺服器工作一樣。新異動分支的所有 SQL 要求會遞送到新的 SQL 伺服器工作。當 db2xa\_commit() 或 db2xa\_rollback() API 完成異動分支時，會重新循環新 SQL 伺服器工作並回到預先啟動工作儲存區。

- 發生下列狀況時，系統會將異動分支標示為「僅回轉」：
  - 當緒仍然與異動分支相關時緒結束。
  - 在一個與異動分支有作用中連結的緒中使用 db2xa\_close() API。
- 發生下列任何狀況時，如果有任何緒仍然與某個異動分支相關，那麼系統會回轉該異動分支：
  - 結束與異動分支相關的連線。
  - 結束啟動異動分支的工作。
  - 系統失敗。
- 有一種狀況是不論是否仍有相關的緒，系統都會回轉異動分支。當連線工作要遞送到的 SQL 伺服器工作結束時，會發生上述狀況。僅對該工作使用「結束工作 (ENDJOB)」CL 指令時才會發生上述狀況。

**註：**下列項目只適用於具有「工作範圍鎖定」的異動。

- 發生下列任何狀況時，若沒有緒與某個異動分支有作用中連結，就不會影響該異動分支。TM 可以從任何使用具有相同 Xainfo 參數值 (在啟動異動分支的緒指定此值) 的 db2xa\_open() API 緒確定或回轉異動分支。
  - 結束與異動分支相關的連線。
  - 為異動分支執行工作但不再與其有作用中連結的緒或工作，可使用 db2xa\_close() API。
  - 為異動分支執行工作但不再與其有作用中連結的緒和工作，可使用 db2xa\_close() API。
  - 系統失敗。於此狀況下，唯有異動分支處於備妥狀態下才不會影響異動分支。若它處於閒置狀態，則系統會將之回轉。

## 確定控制的 SQL 伺服器模式和緒範圍異動

含有「工作範圍」鎖定的確定定義通常把範圍限定為啟動群組。如果一個工作是多緒的工作，那麼工作中的所有緒可以存取確定定義，而且對特定異動所做的變更可以跨越多個緒。也就是說，在相同啟動群組中執行其程式的所有緒參與單一異動。

有時候適合將異動工作的範圍限定為緒而不是啟動群組。換句話說，每一個緒會有自己的確定定義，每一個確定定義的異動工作與在其他緒中執行的工作無關。

DB2 UDB for iSeries 使用變更工作 (QWTCGJJB) API，變更工作在 SQL 伺服器模式中執行，來支援上述目的。在 SQL 伺服器模式中要求 SQL 連接時，該連線會遞送到個別工作。為該連線執行的所有後續 SQL 作業也會遞送到該工作。產生連線時，完成訊息 SQL7908 會傳給 SQL 伺服器模式工作的工作日誌，指出 SQL 要求會遞送到哪一個工作。在這個訊息中指出的工作擁有確定定義。如果發生錯誤，可能需要在工作日誌中查看這兩個工作以瞭解問題來源，因為在執行 SQL 陳述式的工作中並沒有執行真實工作。

在 SQL 伺服器模式中執行時，只能使用 SQL 介面在確定控制之下執行工作。可以使用「內含的 SQL」或「呼叫層次介面 (CLI)」。透過單緒中內含的 SQL 產生的所有連線會遞送到相同的後處理工作。這個動作可讓單一確定要求確定所有連線的工作，就好像它在對於不是在 SQL 伺服器模式中執行的工作所做的一樣。透過 CLI 產生的每一個連線會遞送到個別工作。CLI 需要對每一個要獨立確定或回轉的連線執行工作。

在 SQL 伺服器模式中執行時您無法在確定控制之下執行下列作業：

- 記錄那些透過非 SQL 介面執行的變更

- 變更 DDM 檔
- 變更 API 確定資源

您無法直接在一個於 SQL 伺服器模式中執行的工作啟動確定控制。關於 SQL 伺服器模式的其他資訊，請參閱「資料庫」主題中的下面幾頁：

- 為什麼於 SQL 伺服器模式中執行 DB2 CLI (Why you would run DB2 CLI in SQL server mode)
- 於 SQL 伺服器模式中啟動 DB2 CLI (Starting DB2 CLI in SQL Server Mode)
- 於伺服器模式中執行 DB2 CLI 的限制 (Restrictions for running DB2 CLI in server mode)

---

## 啟動確定控制

若要啟動確定控制，請使用 STRCMTCTL (啟動確定控制) 指令。

註：SQL 應用程式不需要啟動確定控制。當 SQL 隔離層次不是 \*NONE 時，在連線時 SQL 會隱含地啟動確定控制。

使用 STRCMTCTL 指令時您可以指定下列項目：

下列主題說明這些參數。

### 確定「鎖定層次」

使用 STRCMTCTL 指令上的 LCKLVL 參數指定「鎖定層次」。您所指定的層次成為那些在確定義的確定控制之下開啓和放置資料庫檔案的記錄鎖定的預設層次。關於其他資訊，請參閱確定鎖定層次。

### 確定通知物件

使用 NTFY 參數指定通知物件。通知物件為訊息佇列、資料區或資料庫檔案，它包含的資訊識別針對特定確定義而完成的前次順利異動 (如果該確定義沒有正常結束的話)。關於其他資訊，請參閱確定通知物件。

### 確定範圍參數

使用 CMTSCOPE 參數指定確定範圍。啟動確定控制之後，系統會建立確定義。確定範圍參數識別確定義的範圍。預設是將確定義的範圍限定為提出啟動確定控制要求的程式的啟動群組。替代範圍限定為工作。

### 預設異動日誌參數

啟動確定控制時您可以指定預設異動日誌。您可以針對下列原因使用預設異動日誌：

- 您想要攫取異動日誌登錄。這些登錄可協助您分析什麼資源與某個異動相關的歷程。它們不用於套用和移除異動日誌變更。省略異動日誌登錄 (OMTJRNE) 參數決定系統是否寫入異動登錄。
- 您想要增進那些在遞送步驟內關閉檔案又重新開啓檔案的工作的效能。如果您關閉所有指定給非預設異動日誌的檔案，就會從遞送步驟中移除關於該異動日誌的所有系統資訊。如果以後開啓指定給該異動日誌的檔案，就必須重新建立關於該異動日誌的所有資訊。無論指定給異動日誌的資源是否是作用中，系統都會在確定義中保留關於預設異動日誌的資訊。

### 確定文字參數

當顯示關於為某個工作啟動的確定義的資訊時，使用 TEXT 文字參數來識別要與確定義相關的特定文字。如果沒有指定文字，系統會提供預設文字說明。

### 省略異動日誌登錄參數

如果指定預設異動日誌來增進效能，您可以使用 OMTJRNE 參數來防止系統寫入異動日誌登錄。讓系統寫入異動登錄會大幅增加異動日誌接收器大小，而使確定和回轉作業期間的效能降低。

設定和測試確定控制環境或新應用程式時異動登錄很有用。

在下列狀況下無論 OMTJRNE 參數的值是什麼，異動登錄會寫入預設異動日誌：

- 在確定或回轉作業期間發生系統錯誤。
- 手動變更了參與異動的資源，而此變更造成試探混合狀況。關於試探混合狀況的說明，請參閱兩階段確定控制的異動狀態。這種手動變更稱為試探法。

您可以使用關於什麼資源參與異動的資訊，來決定在這些狀況下要採取什麼動作。

在異動日誌登錄的可變長度部份主題，表格 15 到表格 21 顯示異動日誌登錄的登錄專用資料的佈置。

下列有關於啟動確定控制的其他資訊：

- 確定通知物件
- 確定鎖定層次

## 確定通知物件

**通知物件**為訊息佇列、資料區或資料庫檔案，內含的資訊可識別特定確定定義 (如果該確定定義未正常結束) 上次順利完成的異動。**確定 ID** 提供用來識別確定定義前次順利完成的異動，該確定 ID 可使確定作業與一組特定可確定資源變更產生關聯。

唯有確定定義未正常結束時，該確定定義前次順利完成異動的確定 ID 才會置於通知物件中。此資訊可協助判斷應用程式處理的結束位置，以便應用程式可重新啟動。

若是獨立磁碟儲存區，通知物件必須和確定定義位於相同的獨立磁碟儲存區或獨立磁碟儲存區群組中。如果您將確定定義移至另一個獨立磁碟儲存區或獨立磁碟儲存區群組，通知物件也必須位於另一個獨立磁碟儲存區或獨立磁碟儲存區群組中。如果確定定義異常終止，則另一個獨立磁碟儲存區或獨立磁碟儲存區群組上的通知物件會被更新。如果在另一個獨立磁碟儲存區或獨立磁碟儲存區群組中找不到通知物件，則更新失敗，訊息 CPF8358。

如果登載的資源參與現行異動且以確定 ID 執行確定作業，則確定 ID 置於確定異動日誌登錄中 (異動日誌碼和登錄類型 C CM)，它識別該特定異動為已確定。含有確定 ID 的確定異動日誌登錄傳送至與參與異動的資源相關的每個異動日誌中。

下表顯示如何指定確定 ID 及其最大值。如果確定 ID 超出其最大值，則在寫入通知物件時會被截斷。

語言	作業	確定 ID 的最大字元數
CL	COMMIT 指令	3000 <sup>1</sup>
ILE RPG*	COMIT 作業碼	4000 <sup>1</sup>
PLI°	PLICOMMIT 次常式	4000 <sup>1</sup>
ILE C*	_Rcommit 函數	4000 <sup>1</sup>
ILE COBOL*	COMMIT 動詞	不受支援
SQL	COMMIT 陳述式	不受支援
註：		
<sup>1</sup> 如果通知物件是資料區，則最大大小為 2000 個字元。		

當通知物件以確定 ID 更新後，它會被更新為如下所示：

## 資料庫檔案

如果使用資料庫檔案作為通知物件，則會新增確定 ID 至檔案結尾。任何現有的記錄將保留在檔案中。由於若干使用者或工作可同時變更記錄，所以檔案中每一個確定 ID 應包含使資料與失敗的工作和確定定義產生關聯之唯一資訊。可登載提供的檔案

## 資料區

如果使用資料區作為通知物件，則當確定 ID 置於資料區中時，整個資料區內容會被置換。如果有多個使用者或工作在使用相同程式，則僅未正常結束的最後確定定義中的確定 ID 會在資料區中。因此，單一資料區通知物件可能不會產生用於重新啟動應用程式的正確資訊。若要解決這個問題，請對每個工作站使用者或工作的每個確定定義使用個別的資料區。

## 訊息佇列

如果使用訊息佇列作為通知物件，則訊息 CPI8399 傳送至訊息佇列。確定 ID 置於訊息 CPI8399 的第二層文字中。當使用資料庫檔案作為通知物件時，每個確定 ID 的內容應僅識別一個工作的特定確定定義，以便重新啟動應用程式。

有關使用通知物件的範例，請參閱範例：使用通知物件來啟動應用程式。

## 確定鎖定層次

您對啟動確定控制 (STRCMTCTL) 指令上 LCKLVL 參數所指定的值，變成在確定定義的確定控制下所開啓及放置的資料庫檔案之記錄鎖定預設層次。當開啓本端資料庫檔案時，無法置換記錄鎖定的預設層次。不過，SQL 存取的資料庫檔案使用發出第一個 SQL 陳述式時生效的現行 SQL 隔離層次。確定控制的注意事項和限制已說明物件和記錄層次變更的注意事項。

應根據您的需求、允許的等待期間及最常用的釋放程序來指定鎖定層次。

下列說明僅套用至確定控制下所開啓的檔案：

### \*CHG 鎖定層次

若您要防止已變更的記錄被同時執行的其他工作所變更，請使用此值。對於確定控制下開啓的檔案，鎖定將保留於異動的持續時間內。對於未在確定控制下開啓的檔案，則記錄鎖定僅從讀取記錄開始保留到完成更新作業為止。

### \*CS 鎖定層次

如果您要防止已變更及擷取的記錄被同時執行的其他工作所變更，請使用此值。未變更之已擷取記錄僅保護到它們被釋放或擷取到另一個記錄為止。

\*CS 鎖定層次可確保其他工作無法讀取此工作已讀取的記錄來更新。此外，程式無法讀取另一個工作中使用記錄鎖定類型 \*UPDATE 鎖定的更新記錄，直到該工作存取另一個記錄為止。

### \*ALL 鎖定層次

如果您要防止確定控制下已變更及擷取的記錄被同時在確定控制下執行的其他工作所變更，請使用此值。已擷取或變更的記錄會被保護直到下一個確定或回轉作業為止。

\*ALL 鎖定層次可確保其他工作無法存取此工作已讀取來更新的記錄。此不同於一般鎖定通信協定。當鎖定層次被指定為 \*ALL 時，若於另一個工作中以記錄鎖定類型 \*UPDATE 鎖定，則即使並未讀取記錄來更新，也無法存取該記錄。

下表顯示在確定控制下和不在確定控制下的檔案之記錄鎖定持續時間。

要求	LCKLVL 參數	鎖定持續時間	鎖定類型
唯讀	非確定控制	非鎖定	無
	*CHG	非鎖定	無
	*CS	由讀取至下一個讀取、確定或回轉	*READ
	*ALL	由讀取至確定或回轉	*READ
讀取以更新，然後更新或刪除 <sup>1</sup>	非確定控制	由讀取至更新或刪除	*UPDATE
	*CHG	由讀取至更新或刪除	*UPDATE
		然後由更新或刪除至下一個確定或回轉 <sup>2</sup>	*UPDATE
	*CS	由讀取到更新或刪除	*UPDATE
		然後由更新或刪除至下一個確定或回轉 <sup>2</sup>	*UPDATE
	*ALL	由讀取到更新或刪除	*UPDATE
		然後由更新或刪除至下一個確定或回轉 <sup>2</sup>	
	讀取以更新，然後釋放 <sup>1</sup>	非確定控制	由讀取到釋放
*CHG		由讀取到釋放	*UPDATE
*CS		由讀取到釋放、確定或回轉	*UPDATE
		然後，由釋放到下一個讀取、確定或回轉	*UPDATE
由讀取到釋放、確定或回轉		*UPDATE	然後，由釋放到下一個確定或回轉
*UPDATE			
新增	非確定控制	非鎖定	無
	*CHG	由新增到確定或回轉	*UPDATE
	*CS	由新增到確定或回轉	*UPDATE
	*ALL	由新增到確定或回轉	*UPDATE
直接寫入	非確定控制	直接寫入的持續時間	*UPDATE
	*CHG	由直接寫入到確定或回轉	*UPDATE
	*CS	由直接寫入到確定或回轉	*UPDATE
	*ALL	由直接寫入到確定或回轉	*UPDATE

**備註：**

<sup>1</sup> 若確定或回轉作業執行於 read-for-update 作業之後，但之前記錄已被更新、刪除或釋放，則在確定或回轉作業期間記錄被解除鎖定。只要確定或回轉完成，便會結束記錄保護。

<sup>2</sup> 若已刪除了記錄，但尚未針對異動發出確定或回轉，則刪除的記錄不會保持鎖定。如果相同或不同的工作嘗試依索引鍵來讀取已刪除的記錄，則工作會收到找不到記錄的指示。不過，若檔案上的唯一索引存取路徑存在，則除非異動被確定，否則另一個工作無法使用與已刪除記錄相同的唯一索引鍵值來插入或更新記錄。若已刪除了記錄，但尚未針對異動發出確定或回轉，則刪除的記錄不會保持鎖定。若相同或不同的工作嘗試依索引鍵來讀取已刪除的記錄，則工作會收到找不到記錄的指示。不過，若檔案上的唯一索引存取路徑存在，則除非異動被確定，否則另一個工作無法使用與已刪除記錄相同的唯一索引鍵值來插入或更新記錄。

當鎖定層次為 \*CS 或 \*ALL 時，在不是讀取來更新的記錄上取得記錄層次類型 \*READ。此鎖定類型可防止其他工作讀取更新記錄，但不能防止唯讀作業存取記錄。

在已更新、刪除、新增或讀取來更新的記錄上取得記錄鎖定類型 \*UPDATE。此鎖定類型可防止其他工作讀取更新記錄，以及防止在確定控制下執行具有記錄鎖定類型 \*CS 或 \*ALL 的工作存取記錄，即使是唯讀作業的記錄亦然。

不是使用確定控制的程式可讀取另一工作鎖定的記錄，但無論對 LCKLVL 參數指定哪一值，都無法讀取更新記錄。

當對啟動群組或工作啟動確定控制時，指定給確定定義的鎖定層次僅套用到與該特定確定定義相關的已開啓檔案。

註： \*CS 和 \*ALL 鎖定層次值可防止您擷取目前有另一工作擱置中變更的記錄。不過，\*CS 和 \*ALL 鎖定層次值不阻止您使用一個啟動群組中執行的程式，來擷取目前在相同工作內另一個啟動群組中執行的程式之擱置中變更的記錄。

在相同工作內，程式可變更已於現行異動內變更的記錄，只要使用相同的確定定義重新存取該記錄即可。當使用工作層次確定定義時，在任何使用工作層次確定定義的啟動群組內執行的程式可存取已變更的記錄。

---

## 結束確定控制

您可以使用結束確定控制 (ENDCMTCTL) 指令來結束工作層次或啟動群組層次確定定義的確定控制。發出 ENDCMTCTL 指令，指示系統提出要求的程式正在使用的確定定義即將結束。ENDCMTCTL 指令僅結束工作的一個確定定義，而工作的其他確定定義則不會改變。

如果「啟動群組層次」確定定義結束，則該啟動群組內執行的程式無法再於確定控制下執行變更，除非已對工作啟動「工作層次」確定定義。如果「工作層次」確定定義作用中，則在剛結束確定控制的啟動群組內執行的程式可立即使用它。

如果「工作層次」確定定義結束，則在使用「工作層次」確定定義的工作內執行的任何程式，無法再於確定控制下執行變更，除非先使用 STRCMTCTL 指令重新啟動確定控制。

發出 ENDCMTCTL 指令之前，必須滿足下列情況才能結束確定定義：

- 在結束確定定義的確定控制下所開啓的全部檔案必須先關閉。當結束「工作層次」確定定義時，這包括在使用「工作層次」確定定義的任何啟動群組中所執行的任何程式在確定控制下開啓的全部檔案。
- 所要結束確定定義的所有 API 確定資源必須先使用 QTNRMVCR API 移除。當結束「工作層次」確定定義時，這包括在使用「工作層次」確定定義的任何啟動群組中所執行的任何程式新增的全部 API 確定資源。
- 與所要結束確定定義相關的遠端資料庫必須切斷連線。
- 所有與確定定義相關之受保護的交談必須使用正確同步化層次來正常結束。

如果確定控制已在交談式作業中結束，而且與確定定義相關的一或多個可確定資源具有擱置中變更，則會傳送查詢訊息 CPA8350 給使用者，詢問要確定擱置中變更、回轉擱置中變更或取消 ENDCMTCTL 要求。

如果確定控制已在批次工作中結束，而且與所要結束確定定義相關的一或多個已關閉檔案仍有擱置中變更，這些變更會被回轉並送出一則訊息：

- CPF8356 (如果僅登記本端資源)
- CPF835C (如果僅登記遠端資源)



- CPF83E4 (如果同時登記本端和遠端資源)

如果通知物件被定義給結束的確定定義，則它可能被更新。有關系統更新通知物件的詳細資訊，請參閱第 4-13 頁的「更新通知物件」。

當具有登記為最後代理程式的 API 之啟動群組結束時，會呼叫 API 的跳出程式來接收確定或回轉決策。在這個情形下，即使啟動群組是正常結束，API 跳出程式仍可能傳回回轉要求。因此，不會執行隱含的確定作業。

在確定定義順利結束後，表示所有必要的回復 (若有的話) 都已執行。不需要對結束的確定定義所相關的確定資源執行其他回復。

在結束確定定義後，可對啟動群組內執行的程式重新啟動「工作層次」或「啟動群組層次」確定定義。唯有尚未對工作啟動「工作層次」確定定義時才能啟動它。

雖然啟動群組內執行的程式可重複啟動及結束確定定義，但重複啟動和結束作業所需的系統資源數量可能導致工作效能及整體系統效能降低。因此，建議將確定定義保留在作用中，以供稍後呼叫的程式能夠使用它。

---

## 系統起始的結束確定控制

系統可以結束確定控制，或執行隱含確定或回轉作業。有時候系統起始結束確定控制是正常狀況。有時候會因為系統或工作異常結束而結束確定控制。

下列幾頁說明系統隱含地結束確定控制的狀況，以及您應該採取的動作：

- 在啟動群組結束期間的確定控制
- 隱含確定和回轉作業
- 在正常遞送步驟結束期間的確定控制
- 在異常系統或工作結束期間的確定控制
- 在確定控制結束之後更新通知物件
- 在起始程式載入期間回復確定控制

## 啟動群組結束期間的確定控制

當啟動群組結束時，系統會自動結束啟動群組層次確定定義。如果啟動群組層次確定定義有擱置中變更存在且啟動群組是正常結束，則系統會在確定定義結束前對其執行隱含的確定作業。否則，只要啟動群組異常終止，或系統在關閉啟動群組限定範圍內確定控制下所開啓的任何檔案時發生錯誤，便會在啟動群組層次確定定義結束前對其執行隱含的回轉作業。

註：

在 \*JOB 或 \*DFACTGRP 確定定義的啟動群組結束處理期間，絕不執行隱含的確定或回轉作業。原因是 \*JOB 和 \*DFACTGRP 確定定義絕不會因為啟動群組結束而結束。但是，這些確定定義可使用 ENDCMTCTL 指令來明確地結束，或在工作結束時由系統結束。

當啟動群組結束時，系統會自動關閉啟動群組限定範圍內的任何檔案。此包括在確定控制下所開啓啟動群組限定範圍內的任何資料庫檔案。任何此種檔案的關閉會於發生在對啟動群組層次確定定義執行任何隱含的確定作業之前。因此，在執行任何隱含的確定作業之前，任何位於 I/O 緩衝區內的記錄會先被強制執行至資料庫。

由於可能執行部份隱含的確定或回轉作業，所以會對每一個與啟動群組層次確定定義相關的 API 確定資源呼叫 API 確定和回轉跳出程式。跳出程式必須在 5 分鐘內完成其處理程序。於呼叫 API 確定和回轉跳出程式後，系統便會自動移除 API 確定資源。

若對由於啓動群組結束而結束的確定定義執行隱含的回轉作業，則會更新定義給確定定義的通知物件。有關系統更新通知物件之詳細資訊，請參閱更新通知物件。

## 隱含的確定和回轉作業

通常，確定或回轉作業起始自應用程式，它使用其中一個支援確定控制的可用程式設計語言。這些類型的確定和回轉作業稱為**明確的確定和回轉要求**。不過，於某些情況下，系統為確定定義起始確定或回轉作業。由系統所起始的確定和回轉作業稱為**隱含的確定和回轉要求**。

下列兩表格顯示當發生一些與具有擱置中變更的確定定義相關的事件時，系統所採取的動作。若下列任何一項屬實，表示確定定義具有擱置中變更：

- 已更新任何可確定資源。
- 已讀取在確定控制下開啓的資料庫檔案，因為讀取檔案會變更檔案位置。
- 確定定義具有 API 資源。由於使用者程式完成了 API 資源變更，所以系統必須假設所有 API 資源都有擱置中變更。

C CM (確定作業) 異動日誌登錄和 C RB (回轉作業) 異動日誌登錄，指示作業是為明確或隱含的。

下表顯示當工作正常或異常終止 (視下列各項而定) 時系統所採取的動作：

- 異動狀態。
- 確定定義的 action-if-end 工作值。
- API 資源是否為最後代理程式。

狀態	最後代理程式 API	Action if Endjob <sup>1</sup> 選項	確定或回轉作業
RST	N/A	N/A	<p>如果確定定義與 X/Open 整體異動無關，則會執行隱含的回轉。</p> <p>如果確定定義與 X/Open 整體異動相關，則會發生下列情況：</p> <ul style="list-style-type: none"> <li>• 如果異動分支狀態不是「作用中」(S1)，則不執行任何動作且異動分支保留在相同狀態中。</li> <li>• 如果異動分支狀態為「作用中」(S1)，則會執行隱含的回轉。</li> </ul>
PIP	N/A	N/A	<p>如果確定定義與 X/Open 整體異動無關，則會執行隱含的回轉。</p> <p>如果確定定義與 X/Open 整體異動相關，則異動分支處於「閒置」(S2) 狀態中，且保留於「閒置」(S2) 狀態中。</p>

狀態	最後代理程式 API	Action if Endjob <sup>1</sup> 選項	確定或回轉作業
PRP	N/A	WAIT	<p>如果確定定義與 X/Open<sup>2</sup> 整體異動無關，則會發生下列情況：</p> <ul style="list-style-type: none"> <li>• 啟動重新同步以接收確定作業的起始程式所做的決策。</li> <li>• 執行確定或回轉傳回的決策。此被視為明確的作業。</li> </ul>
PRP	N/A	C	如果確定定義與 X/Open <sup>2</sup> 整體異動無關，則會執行隱含的確定作業。
		R	<p>如果確定定義與 X/Open 整體異動無關，則會執行隱含的回轉作業。</p> <p>如果確定定義與 X/Open 整體異動相關，則會發生下列情況：</p> <ul style="list-style-type: none"> <li>• 如果啟動異動的工作終止，則異動會保持已備妥狀態，直到 XA TM 對它確定或回轉為止。於此情形下，XA 異動分支狀態將保持為「已備妥」(S3)。</li> <li>• 如果異動工作遞送到其中的 SQL 伺服器工作結束，則會隱含地執行強制回轉。在此情形下，XA 異動分支狀態將變更為「試探完成」(S5)。</li> </ul>
CIP	N/A	N/A	執行明確的確定作業。
LAP	NO	WAIT	1. 使用對最後代理程式的重新同步以擷取決策來確定或回轉。
			2. 執行傳回的決策確定或回轉。它被視為明確的作業。
LAP	YES	WAIT	1. 呼叫最後代理程式 API 來擷取確定或回轉決策。
			2. 執行確定或回轉作業。它被視為明確的作業。
LAP	N/A	C	執行隱含的確定作業。
		R	執行隱含的回轉作業。
CMT	N/A	N/A	已對此確定定義及任何下游位置完成確定作業。確定作業完成。

狀態	最後代理程式 API	Action if Endjob <sup>1</sup> 選項	確定或回轉作業
VRO	N/A	N/A	本端和遠端代理程式表決為唯讀。所有下游代理程式也必須表決為唯讀。不需要採取任何動作。
RBR	N/A	N/A	需要回轉作業。執行明確的回轉作業。

註：

<sup>1</sup> 您可以使用變更確定選項 (QTNCHGCO) API 來變更 Action if Endjob 選項。

<sup>2</sup> 如果確定定義與 X/Open 整體異動相關，則會發生下列情況：

- 如果啟動異動的工作終止，則異動會保持已備妥狀態，直到 XA TM 對它確定或回轉為止。於此情形下，XA 異動分支狀態將保持為「已備妥」(S3)。
- 僅針對「異動限定範圍」鎖定，若異動工作遞送到其中的 SQL 伺服器工作結束，則會隱含地執行強制回轉。於此情形下，XA 異動分支狀態將變更為「試探完成」(S5)。

下表顯示啟動群組結束並僅套用到具有「工作範圍」鎖定的異動時，系統所採取的動作。系統動作根據下列各項而定：

- 異動狀態。(當啟動群組結束時，一律重設 (RST) 之)。
- 啟動群組如何正常或異常結束。
- API 資源是否為最後代理程式。

註：

如果 API 資源被登記為最後代理程式，則會提供確定或回轉決策的控制權給最後代理程式。決策結果被視為明確的作業

狀態	最後代理程式 API	結束類型	確定或回轉作業
RST	否	正常	執行隱含的確定作業。如果受保護的交談存在，則確定定義會成為確定作業的根起始程式。
RST	否	異常	執行隱含的回轉。
RST	是	正常	呼叫 API 跳出程式。確定或回轉作業取決於 API。
RST	Yes	異常	呼叫 API 跳出程式。確定或回轉作業取決於 API。

## 正常遞送步驟結束期間的確定控制

當遞送步驟正常結束時，系統結束一個工作的所有確定定義。

註： 下列僅適用於「工作範圍」鎖定的確定定義。

下列其中一個情況會正常地結束一個遞送步驟：

- 批次作業正常結束。

- 交談式作業正常地登出。
- 「重新遞送工作 (RRTJOB)」、「傳送工作 (TFRJOB)」或「傳送批次作業 (TFRBCHJOB)」指令結束現行遞送步驟並啟動新的遞送步驟。

其他任何的遞送步驟結束皆視為不正常，在工作日誌中以工作完成訊息 CPF1164 的一個非零完成碼來辨識。

遞送步驟結束期間在結束一個確定定義之前，如果確定定義有擱置中變更，則系統會執行隱含的回轉作業。這包括對確定定義的每一個相關的 API 確定資源，呼叫 API 確定和回轉跳出程式。跳出程式必須在 5 分鐘之內完成其處理程序。在呼叫 API 確定和回轉跳出程式之後，系統會自動移除 API 確定資源。

如果確定定義有定義通知物件，則會一併更新。關於系統更新通知物件的詳細資訊，請參閱更新通知物件。

## 於異常系統或工作結束期間的確定控制

本主題僅套用至具有工作限定範圍鎖定的確定定義。當工作異常結束時，系統會結束該工作的所有確定定義。這些確定定義會在結束工作處理程序中結束。如果系統異常終止，則系統會結束在系統異常終止時所有已啟動並為所有作用中工作所使用的全部確定定義。結束這些確定定義是資料庫回復處理程序的一部份，而資料庫回復處理程序是在系統異常終止之後的下一個起始程式載入 (IPL) 期間執行。

### 注意：

確定定義的回復是指系統或工作因停電、硬體故障，或作業系統或授權內碼失敗而異常終止。您不應該使用「異常終止工作 (ENDJOBABN)」指令來強制工作異常終止。異常終止會導致您要結束的工作之作用中異動擱置變更確定或回轉一部份。下一個起始程式載入 (IPL) 會嘗試回復以 ENDJOBABN 指令結束的工作的部份異動。

系統在起始程式載入 (IPL) 期間針對您以 ENDJOBABN 指令結束的工作所執行的確定控制回復，其結果無法確定。此不確定性是由於確定資源的所有鎖定在工作異常終止時被釋放了。任何因部份異動而導致的擱置中變更已可供其他工作使用。這些擱置中變更會造成其他應用程式對資料庫產生其他錯誤的變更。同樣地，以後執行的任何起始程式載入 (IPL) 回復會對工作異常終止後應用程式所做的變更有不利的影響。例如，在起始程式載入 (IPL) 回復期間，SQL 表格可能因為擱置中建立表格的回轉動作而被捨棄。不過，其他應用程式在工作異常終止後可能已將若干列插入表格中。

系統會針對工作異常終止期間或系統異常終止後的下一個 IPL 期間結束的確定定義來執行下列作業：

- 於結束確定定義之前，若確定定義具有擱置中變更，則除非確定定義處理程序在確定作業中被岔斷，否則系統會執行一個隱含的回轉作業。若於確定作業中終止，則根據異動的狀態，異動可能被回轉、重新同步化或確定。請參閱隱含的確定及回轉作業。執行隱含的回轉作業或完成確定作業的處理程序包括對確定定義相關的每一個 API 確定資源呼叫 API 確定和回轉跳出程式。呼叫 API 確定和回轉跳出程式後，系統會自動移除 API 確定資源。

### 注意：

在異動未完成時 (異動狀態為 LAP 或 PRP) 結束工作，可能導致資料庫中產生不一致性 (變更可能在一或多個系統上確定但在其他系統上回轉)。

- 如果 *Action if Endjob* 確定選項為 COMMIT，則無論參與異動的其他系統變更被確定或回轉，只要工作結束，此系統的變更就會被確定。
- 如果 *Action if Endjob* 確定選項為 ROLLBACK，則無論參與異動的其他系統變更被確定或回轉，只要工作結束，此系統的變更就會被回轉。
- 如果 *Action if Endjob* 確定選項為 WAIT，則除非擁有確定或回轉決策的系統重新同步完成，否則工作不會結束。若要使工作在重新同步完成前結束，則必須執行試探法及取消重新同步。

不建議您在長期執行回轉期間異常終止工作或系統。此將導致因工作結束 (或於下一個 IPL 期間系統被終止) 而發生另一個回轉。後續的回轉會重複原始回轉所執行的工作，顯然要花更長的時間來執行。

- 如果通知物件被定義給確定定義，則它可能被更新。有關系統更新通知物件之詳細資訊，請參閱更新通知物件。

若程序在確定控制終止前結束且受保護的交談仍於作用中，則確定定義可能需要確定或回轉。此動作視確定定義的「狀態」選項和「若結束工作的動作」選項而定。

## 更新通知物件

在通知物件的用途方面，以下被視為未確定的變更：

- 在確定控制之下執行的記錄更新。
- 在確定控制之下刪除的記錄。
- 在確定控制之下對本端 DDL 物件所做的物件層次變更。
- 為在確定控制之下開啓的資料庫檔案執行的讀取作業。這是因為當執行回轉作業時檔案位置會回到前次確定界限。如果在確定控制之下執行讀取作業，就會變更檔案位置，因此確定定義有未確定的變更存在。
- 有新增下列其中一項資源的確定定義一律被視為含有未確定的變更：
  - API 確定資源
  - 遠端「分散式關聯資料庫架構 (DRDA\*)」資源
  - 「分散式資料庫管理架構 (DDM)」資源
  - LU 6.2 資源

這是因為系統不知道何時對此物件和與這些資源類型相關的物件做了真正變更。可確定資源的類型有更多關於如何新增和使用這些資源類型的資訊。

系統更新通知物件而且根據下列方法來結束確定定義：

- 如果工作異常結束而且沒有未確定的變更，那麼系統不會將前次順利確定作業的確定 ID 放在通知物件。
- 當啓動群組結束時如果為「啓動群組層次」確定定義執行了隱含的確定作業，那麼系統不會將前次順利完成的確定作業的確定 ID 放在通知物件中。

註：絕不會為 \*DFACTGRP 或 \*JOB 確定定義執行隱含的確定作業

- 如果在確定定義的第一個順利完成的確定作業之前，系統、工作或啓動群組異常結束，那麼系統不會更新通知物件，因為沒有前次確定 ID。若要區別這個狀況與正常程式完成，那麼程式必須使用特定登錄更新通知物件之後，才能完成確定定義第一個順利完成的確定作業。
- 如果在至少一個順利完成的確定作業之後發生異常工作結束或異常系統結束，那麼系統會將該確定作業的確定 ID 放在通知物件中。如果前次順利完成的確定作業沒有指定確定 ID，就不會更新通知物件。對於異常工作結束來說，會為該工作的每一個作用中確定定義執行這個通知物件處理。對於異常系統結束來說，會為該系統上所有工作的每一個作用中確定定義執行這個通知物件處理。
- 如果發生下列所有狀況，那麼系統會以該確定定義前次順利完成的確定作業的確定 ID 更新通知物件：
  - 非預設啓動群組結束。
  - 為「啓動群組層次」確定定義執行隱含的回轉作業。
  - 為該確定定義執行至少一個順利完成的確定作業。

如果前次順利完成的確定作業沒有指定確定 ID，就不會更新通知物件。如果異常結束啟動群組或在關閉那些在確定控制之下開啓的檔案時發生錯誤（這些檔案的範圍限定為該啟動群組），就會為「啟動群組層次」確定定義執行隱含的回轉作業。關於使資料庫檔案的範圍限定為啟動群組以及如何能夠結束啟動群組的其他資訊，請參閱您使用的 ILE 語言的參考書。

- 當工作異常結束時如果有未確定的變更而且已執行至少一個順利完成的確定作業，那麼前次順利完成的確定作業的確定 ID 會放在通知物件中，而且會回轉未確定的變更。如果前次順利完成的確定作業沒有指定確定 ID，就不會更新通知物件。當工作結束時，會為該工作的每一個作用中確定定義執行這個通知物件處理。關於在正常結束工作時執行的功能的其他資訊，請參閱在正常遞送步驟結束期間的確定控制主題。
- 當執行 ENDCMTCTL 指令時如果有未確定的變更，那麼唯有前次順利完成的確定作業有指定確定 ID 時才會更新通知物件：
  - 對於批次作業來說，會回轉未確定的變更而且前次順利完成之確定作業的確定 ID 會放在通知物件中。
  - 對於交談式作業來說，如果查詢訊息 CPA8350 的回應是回轉變更，就會回轉未確定的變更而且前次順利完成之確定作業的確定 ID 會放在通知物件中。
  - 對於交談式作業來說，如果查詢訊息 CPA8350 的回應是確定變更，那麼系統會提示您輸入要使用的確定 ID，然後確定變更。在提示畫面輸入的確定 ID 放在通知物件中。
  - 對於交談式作業來說，如果查詢訊息 CPA8350 的回應為取消 ENDCMTCTL 要求，就會保留擱置中變更而且不更新通知物件。

### 異常終止後起始程式載入期間的確定控制回復

當您在系統異常終止後執行起始程式載入 (IPL) 時，系統會嘗試回復其終止時作用中的所有確定定義。同樣地，當您轉接獨立磁碟儲存區時，系統會嘗試回復其轉斷或異常終止時作用中且與獨立磁碟儲存區相關的所有確定定義。起始程式載入 (IPL) 期間系統所啟動的資料庫伺服器工作執行此回復作業。系統所啟動的資料庫伺服器工作可處理其他工作無法或不應執行的作業。

資料庫伺服器工作名為 QDBSRVnn，其中 nn 是二位數。資料庫伺服器工作數目視您系統大小而定。同樣地，獨立磁碟儲存區或獨立磁碟儲存區群組的資料庫伺服器工作名稱為 QDBSxxxVnn，其中 xxx 是獨立磁碟儲存區數字，而 nn 為二位數。例如，QDBS035V02 可以是獨立磁碟儲存區 35 的資料庫伺服器工作名稱。

兩階段確定控制的異動狀態顯示系統根據失敗時的異動狀態所採取的動作。以 PRP 和 LAP 兩個狀態而言，系統動作有疑問。

註：

- 下列僅套用到具有「工作範圍」鎖定的確定定義。
- 異動管理程式利用 XA API 而非本主題中描述的重新同步處理程序來回復與 XA 異動相關的確定定義（無論其鎖定為「工作範圍」或「異動範圍」的鎖定）。

系統在對參與異動的其他位置執行重新同步之前，無法決定要採取什麼動作。此重新同步在起始程式載入 (IPL) 或轉接作業完成後執行。

系統使用資料庫伺服器工作來執行這個重新同步。所需回復的確定定義與資料庫伺服器工作相關。在起始程式載入 (IPL) 期間，系統會取得其終止前由確定定義所保留的所有記錄鎖定及其他物件鎖定。若要保護本端確定資源直到重新同步完成，以及讓資源可被確定或回轉，需要這些鎖定。

訊息傳送至資料庫伺服器工作的工作日誌，指示與遠端位置的重新同步狀態。如果異動不確定，則在確定或回轉本端資源之前，必須與擁有異動決策的位置完成重新同步。

當完成異動決策時，下列訊息可能傳送至資料庫伺服器工作的工作日誌。

### **CPI8351**

&1 擱置中變更已回轉

### **CPC8355**

工作 &19/&18/&17 的確定定義 &8 之後置 IPL 回復已完成。

### **CPD835F**

工作 &19/&18/&17 的確定定義 &8 之起始程式載入 (IPL) 回復失敗。

可能也會送出與回復相關的其他訊息。這些訊息傳送至歷程 (QHST) 日誌。如果發生錯誤，則訊息也會傳送至 QSYSOPR 訊息佇列。

您可以藉由使用「iSeries 領航員」、顯示資料庫伺服器工作的工作日誌，或使用「使用確定定義 (WRKCMDFN)」指令來判斷回復進度。雖然「iSeries 領航員」和「使用確定定義」顯示畫面可讓您強制系統確定或回轉，但只有當不得已時才應使用。如果您預期參與異動的所有位置最後都會返回作業，您應該讓系統將它們進行重新同步化。這可確保資料庫整合性。

---

## **管理異動和確定控制**

下列為您可執行來管理確定控制的一些作業。

### **顯示確定控制資訊**

此資訊具有一些作業，可用來顯示全部異動的相關資訊，以及顯示與異動相關的工作之相關資訊。

### **確定控制的最佳化效能**

此資訊具有一些作業，可用來將確定控制對系統效能的影響降至最低。

## **顯示確定控制資訊**

您可以使用「iSeries 領航員」來顯示系統上所有異動 (邏輯工作單元) 的相關資訊。您也可以尋找與異動相關的工作 (若有的話) 之相關資訊。

註： 這些顯示作業不顯示 SQL 應用程式的隔離層次。

顯示上述資訊，如下所示：

1. 在 **iSeries 領航員** 視窗中，展開您要使用的伺服器。
2. 展開**資料庫**。
3. 展開您要使用的系統。
4. 展開**異動**。

註： 若要檢視與 X/Open 整體異動相關的異動，請展開**整體異動**。若要檢視 DB2 UDB 管理的異動，請展開**資料庫異動**。

5. 展開**整體異動**或**資料庫異動**。

此顯示畫面顯示下列項目：

- 工作單元 ID
- 工作單元狀態
- 工作



- 使用者
- 數目
- 重新同步進度
- 確定定義

線上說明提供有關所有狀態顯示畫面及其欄位的資訊。

您也可以使用「iSeries 領航員」來顯示下列資訊：

- 顯示異動的已鎖定物件
- 顯示與異動相關的工作
- 顯示異動的資源狀態
- 顯示異動內容

### 顯示異動已鎖定的物件

您只能顯示具有「異動範圍」鎖定的整體異動之已鎖定物件。

顯示異動的已鎖定物件：

1. 在 **iSeries 領航員** 視窗中，展開您要使用的伺服器。
2. 展開**資料庫**。
3. 展開您要使用的系統
4. 展開**異動**。
5. 展開**整體異動**。
6. 以滑鼠右鍵按一下您要使用的異動，然後選取**已鎖定物件**。

### 顯示與異動相關的工作

顯示與異動相關的工作：

1. 在 **iSeries 領航員** 視窗中，展開您要使用的伺服器。
2. 展開**資料庫**。
3. 展開您要使用的系統
4. 展開**異動**。
5. 展開**整體異動**或**資料庫異動**。
6. 以滑鼠右鍵按一下您要使用的異動，然後選取**工作**。

若是具有工作範圍鎖定的資料庫異動和整體異動，則會顯示與異動相關的工作清單。

若是具有異動範圍鎖定的整體異動，則會顯示連接此異動物件或等待連接此異動物件的工作清單。

### 顯示異動的資源狀態

顯示異動的資源狀態：

1. 在 **iSeries 領航員** 視窗中，展開您要使用的伺服器。
2. 展開**資料庫**。
3. 展開您要使用的系統
4. 展開**異動**。
5. 展開**整體異動**或**資料庫異動**。
6. 以滑鼠右鍵按一下您要使用的異動，然後選取**資源狀態**。

## 顯示異動內容

顯示異動內容：

1. 在 **iSeries 領航員** 視窗中，展開您要使用的伺服器。
2. 展開**資料庫**。
3. 展開您要使用的系統
4. 展開**異動**。
5. 展開**整體異動**或**資料庫異動**。
6. 以滑鼠右鍵按一下您要使用的異動，然後選取**內容**。

## 最佳化確定控制的效能

使用確定控制來要求可能影響系統效能的資源。關於確定控制，有多個因素會影響系統效能。以下是不影響效能的因素、降低效能的因素及改進效能的因素。

### 不影響效能的因素

#### 開啓檔案

如果您開啓檔案但未指定確定開啓選項，即使已啓動確定定義，也不會使用額外的系統資源。關於指定確定開啓選項的詳細資訊，請參閱適當的高階語言參考手冊。

### 降低效能的因素

#### 日誌登載

登載檔案日誌需要系統資源。不過，大部份情況下，使用確定控制來執行日誌登載的效能優於不使用確定控制的效能。如果您僅指定已更新表格內容 (after-image)，則當確定控制生效時，確定控制會將它變更爲未更新表格內容 (before-image) 和已更新表格內容 (after-image)。此通常爲空間上的考量，無關效能。關於日誌登載的詳細資訊，請參閱日誌登載管理主題。

#### 確定作業

如果異動期間變更已登載的資源，則異動的每一個確定會新增兩個登錄至與這些資源相關的每一個日誌中。如果是大量的小異動，登錄數量會明顯地增加。您可以將異動日誌接收器置於不同於日誌的磁碟儲存區。

#### 回轉作業

因爲確定控制必須回轉資料庫內記錄的擱置中變更，所以發生回轉時需要額外的系統資源。而且，若記錄變更處於擱置中，回轉的作業會造成日誌新增額外的登錄。

#### 啓動確定控制 (STRCMTCTL) 和結束確定控制 (ENDCMTCTL) 指令

每次使用 STRCMTCTL 和 ENDCMTCTL 指令來分別啓動和結確定定義束時，系統會引起額外執行時間。請避免在每一個異動上使用 STRCMTCTL 和 ENDCMTCTL 指令。請視需要再使用。您可以在交談式作業開始時建立確定定義，然後在工作期間使用。

#### 在確定控制異動上使用一個以上的日誌

以兩階段確定，開啓於確定控制下的檔案可以登載到一個以上的日誌。不過，使用一個以上的日誌需要額外的系統資源來管理確定定義。使用一個以上的日誌也會使回復更複雜。

#### 記錄鎖定

記錄鎖定會影響其他應用程式。特定工作內鎖定的記錄數目會增加此工作所用的整體系統資源。需要存取相同記錄的應用程式必須等待異動結束。

### 要求 SEQONLY(\*YES)

如果您要求 SEQONLY(\*YES) 選項 (使用 OVRDBF 指令或應用程式隱含地嘗試使用 SEQONLY(\*YES)) 且在含有 LCKLVL(\*ALL) 的確定控制之下開啓僅供輸入的檔案，則選項會變更為 SEQONLY(\*NO)。此選項會影響輸入檔的效能，因為記錄不會區塊化。

### 當「作用中儲存」處理程序為作用中時對資料庫檔案要求「記錄層次」變更

如果確定定義是在確定界限上，且正在一個不同的工作中執行「作用中儲存」作業，則對資料庫檔案發出確定控制下的「記錄層次」變更可能會延遲。當檔案登載到與儲存要求上之部份物件相同的日誌時，就會發生這種情況。

註： 當一個工作由於「作用中儲存」核對點處理程序而被保留時，「使用作用中的工作」(WRKACTJOB 指令) 上的「狀態」直欄會顯示 CMTW (確定等待)。

### 當「作用中儲存」處理程序為作用中時，確定或回轉變更

當不同的工作正在執行「作用中儲存」作業時，確定或回轉作業可能會在確定界限上延遲。除非已使用 QTNADDCR API 來新增 API 資源，且容許正常儲存處理程序欄位的值是 Y，否則若 API 確定資源先前已新增到確定定義，就會發生這種情況。

因為工作是在確定或回轉要求期間被保留，且因為一次只能對一個確定定義執行確定或回轉要求，所以有一個以上的確定定義且已新增 API 確定資源的工作，一定使「作用中儲存」作業無法完成。

### 當「作用中儲存」處理程序為作用中時，要求「物件層次」的變更

如果確定定義是在確定界限上，且正在一個不同的工作中執行「作用中儲存」作業，則在確定控制下要求「物件層次」的變更可能會延遲。當做出「物件層次」變更時，若物件所在的檔案庫內正在執行「作用中儲存」作業，則會發生此情況。例如，對檔案庫 MYSQLLIB 執行「作用中儲存」作業時，在確定控制下對檔案庫 MYSQLLIB 中的表格 MYTBL 執行建立 SQL 表格作業可能延遲。

註： 如果等待時間超過 60 秒，則傳送查詢訊息 CPA8351 來詢問使用者是要繼續等待或取消作業。

### 使用 QTNADDCR API 來新增 API 資源

如果工作的所有確定定義是在確定界限上且不同的工作中正在執行「作用中儲存」作業，則要求使用 QTNADDCR API 來新增 API 確定資源可能會延遲。

註：

1. 如果等待時間超過 60 秒，則傳送查詢訊息 CPA8351 來詢問使用者是要繼續等待或取消作業。
2. 如果容許正常儲存處理程序欄位的值為 Y，則這不適用於已使用 QTNADDCR API 所新增的 API 資源。

## 改進效能的因素

### 使用預設日誌登載

當確定定義作用中時，如果您要關閉再重新開啓所有在確定控制下的檔案，則使用預設日誌登載可以改善效能。不過，同時使用預設日誌登載 OMTJRNE(\*NONE) 會降低確定和回轉作業的效能。

### 選取最後代理程式

選取一個最後代理程式可以強化效能，因為在確定期間，系統和最後代理程式之間需要較少的交談。不過，如果確定作業期間發生通信失敗，則不論等待結果選項的值為何，在重新同步完成之前，通信作業將不會完成。這種失敗較罕見，但若真的發生失敗，這個選項可讓應用程式作者考慮到負面影響，造成

使用者爲了重新同步而無限制等待。此應與順利完成確定作業期間時，由最後代理程式最佳化提供的效能加強做一比較衡量。此考量對於交談式作業的重要性通常大於批次作業。

預設爲允許由系統來選取最後代理程式，但使用者可以使用 QTNCHGCO API 來修改這個值。

#### 不使用等待結果選項

如果遠端資源在確定控制之下，則當「等待結果」選項設定爲 N (否) 且所有遠端系統皆支援擅自放棄時，效能會改進。當第一次連接到遠端系統時，對於 DRDA 和 DDM 應用程式，系統會將「等待結果」選項設定爲 N。APPC 應用程式必須明確地設定「等待結果」選項，否則將使用預設值 Y。

#### 選取「確定可以省略」選項

當選取「確定可以省略」選項時，效能會改進。有關此選項的進一步資訊，請參閱兩階段確定的確定定義：指示確定可以省略。

#### 選取「表決唯讀」選項

當選取「表決唯讀」選項時，效能會改進。有關此選項的進一步資訊，請參閱兩階段確定的確定定義：容許表決唯讀。

要改進效能，您可以執行下列作業：

- 將鎖定數縮至最小
- 管理異動大小

### 將鎖定數目降至最少

將記錄鎖定數目降至最少的典型方法就是釋放記錄鎖定。(如果指定了 LCKLVL(\*ALL)，則此方法沒有作用)。例如，單一檔案維護應用程式通常執行下列動作：

- 顯示變更記錄 ID 的提示。
- 擷取所要求的記錄。
- 顯示記錄。
- 允許工作站使用者做變更。
- 更新記錄。

大部份情況下，記錄被鎖定，而無法透過更新來存取所要求的記錄。對正在等待記錄的另一個工作而言，記錄等待時間可能超過。要避免在工作站使用者考慮變更時鎖定記錄，請從資料庫擷取該記錄後將它釋放 (在出現記錄顯示畫面之前)。然後，您需要在更新之前重新存取記錄。如果記錄在被釋放和重新存取的這段期間被變更，您應該通知工作站使用者。程式可判定記錄是否有變更，方法是在擷取記錄後保存原始記錄的一或多個欄位，並將它們與相同記錄中的欄位做比較，如下所示：

- 使用記錄中的更新計數欄位，在更新前對該欄位加 1。當重新擷取記錄時，程式會保存原始值並將它與欄位中的值做比較。如果有發生變更，則會通知工作站使用者並且重新顯示記錄。唯有發生更新時，更新計數欄位才會變更。此記錄在工作站使用者想要變更時釋放。如果您使用此方法，則必須在更新檔案的每個程式中使用它。
- 保存整個資料記錄內容並將它與下次擷取的記錄做比較。

在上述兩種情況中，這一連串的作業可防止僅使用 RPG 的外部說明資料，RPG 會將相同欄位名稱用於主要記錄和顯示檔案。使用相同欄位名稱 (在 RPG 中) 無效，因爲當重新擷取記錄時，工作站使用者的變更被覆蓋。

此問題的解決方法就是將記錄資料移至資料結構，如果您使用 DDS 關鍵字 RTNDTA，則繼續使用外部說明資料。RTNDTA 關鍵字可讓您的程式重讀顯示畫面上的記錄資料，作業系統不需要將資料從顯示畫面移至程式。這可讓程式執行下列動作：

1. 提示輸入記錄 ID。
2. 擷取資料庫中所要求的記錄。
3. 釋放記錄。
4. 保存一或多個欄位來判定記錄是否有變更。
5. 顯示記錄並等待工作站使用者回應。

如果工作站使用者變更了顯示畫面上的記錄，程式會使用下列順序：

1. 重新擷取資料庫中的記錄。
2. 比較保存的欄位以判定資料庫記錄是否有變更。如果有變更，則當記錄出現時，程式會釋放該記錄並傳送一則訊息。
3. 使用 RTNDTA 關鍵字執行讀取作業來擷取顯示畫面中的記錄，並更新資料庫記錄中的記錄。
4. 進行下一個邏輯提示，因為如果使用者取消了要求，就沒有其他記錄需要釋放。

將 LCKLVL(\*CHG) 和 LCKLVL(\*CS) 可使用於此狀況中。如果使用 LCKLVL(\*ALL)，您必須使用確定或回轉作業來釋放記錄鎖定。

有關鎖定的詳細資訊，請參閱偵測死結。

## 管理異動大小

這次討論的異動是交談式。(確定控制也可以用於批次應用程式，它們通常被視為一連串異動。許多相同注意事項適用於批次應用程式，批次應用程式的確定控制有加以討論)。

您可以在與異動相關的每一個異動日誌的異動期間鎖定最多 500 000 000 筆記錄。您可以使用「查詢選項檔 (QAQQINI)」減少這個限制。使用「變更查詢屬性 (CHGQRYA)」指令的 QRYOPTLIB 參數以指定工作使用的「查詢選項檔」。使用「查詢選項檔」中的 COMMITMENT\_CONTROL\_LOCK\_LEVEL 值作為工作的鎖定限制。

選擇記錄的鎖定層次時，請考慮異動的大小。此大小應決定異動結束之前鎖定記錄的時間長度。您必須決定確定控制的確定或回轉作業是否限於只使用一次 Enter 鍵，或異動是由多次使用 Enter 鍵組成來完成。

註：異動愈短，等待啓動「作用中儲存」核對點處理的工作愈早可以繼續和完成。

例如，對於訂單登錄應用程式來說，客戶可以在一張訂單訂購數個項目，需要更新訂單中每一個項目的訂購詳細記錄和庫存主要記錄。如果異動是定義成整張訂單，而每按一次 Enter 鍵就訂購一個項目，則整份訂單的持續時間會鎖定涉及訂單的所有記錄。因此，常用的記錄 (例如庫存主要記錄) 可長期鎖定，以防止進行其他處理。如果按一次 Enter 鍵使用子檔案輸入所有項目，那麼整份訂單的鎖定期間會縮至最短。

一般來說，應該將鎖定次數和持續時間縮至最小，使數個工作站使用者能夠存取相同資料而不必長時間等待。當使用者在顯示畫面輸入資料時您可以取消鎖定來達到上述目的。某些應用程式可能不需要多位工作站使用者存取相同資料。例如，在每位客戶有許多未結項目記錄的現金入帳應用程式中，一般的作法是鎖定全部記錄，並延遲到工作站使用者結清指定的款項為止。

如果工作站使用者針對某個異動按數次 Enter 鍵，就可能在數個區段中執行此異動。例如：

- 第一個區段是工作站使用者要求資訊的查詢。
- 第二個區段是工作站使用者想要完成整個異動的確認。
- 第三個區段是受影響的記錄的擷取和更新。

這個方法讓記錄鎖定限於按一次 Enter 鍵。

決策結果是來自顯示的資訊的應用程式中，通常使用這個查詢優先方法。例如，在航空公司訂位應用程式中，客戶可能想要知道有哪些班機時刻、轉機和座位安排之後才決定要搭乘哪一個班機。客戶一旦決定了之後，就輸入交易。如果交易失敗（班機目前客滿），就可以使用回轉功能並輸入另一個要求。如果從第一個查詢開始就鎖定記錄直到做出決定，那麼另一位訂票員必須等到另一個交易完成才行。

---

## 實務和範例：確定控制

下列為確定控制的實務和範例。實務以高階方式顯示 JKL 玩具公司如何實作確定控制來追蹤其區域資料庫的異動。

範例提供確定控制的程式碼範例。「練習問題」是一個實作確定控制的 RPG 程式。它包括邏輯流程，顯示每一個步驟的實際情形。

下列三個範例顯示在系統異常結束之後使用確定控制來啟動應用程式。

### 實務

- 實務：確定控制

### 範例

- 確定控制的練習問題
- 確定控制的練習問題之邏輯流程
- 範例：使用異動日誌檔來啟動應用程式
- 範例：使用通知物件來啟動應用程式
- 範例：使用標準處理程式來啟動應用程式

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

## 實務範例：確定控制

JKL Toy Company 使用確定控制來保護製造和庫存的資料庫記錄。此實務範例以高階顯示 JKL Toy Company 在零件從庫存部門轉送到製造部門時如何使用確定控制。

有關 JKL Toy Company 網路環境的說明，請參閱實務範例：日誌管理。下列實務範例顯示如何在生產伺服器 JKLPROD 上使用確定控制。

此實務範例說明在兩個範例中使用確定控制的優點。第一個範例顯示公司的庫存程式 Program A 如何在沒有確定控制的情況下運作，以及可能發生的問題。第二個範例顯示程式如何使用確定控制。

JKL Toy Company 在其伺服器 JKLPROD 上使用庫存應用程式 Program A。Program A 使用兩個記錄。一個記錄追蹤倉庫中儲存的項目。另一個記錄追蹤由倉庫移除並用於生產的項目。

### 不使用確定控制的 Program A

假設下列應用程式不使用確定控制。系統鎖定更新的記錄讀取。下列步驟說明應用程式如何追蹤從倉庫移除二極體並轉帳至活儲帳戶：

- Program A 鎖定並擷取倉庫記錄。(如果記錄被另一個程式鎖定，此動作可能需要等待)。
- Program A 鎖定並擷取生產記錄。(此動作也需要等待)。Program A 目前鎖定了兩個記錄，其他程式無法變更它們。

- Program A 更新倉庫記錄。這導致記錄被釋放，所以現在可讀取該記錄以供其他任何程式更新。
- Program A 更新生產記錄，這導致記錄被釋放，所以現在可讀取該記錄來供其他任何程式更新。

如果沒有使用確定控制的話，則需要解決問題來使程式能在所有情況下正常運作。例如，如果 Program A 因為工作或系統失敗而沒有更新這兩個記錄，則會發生問題。在這個情況下，兩個檔案不一致 — 二極體從倉庫記錄中移除了，但未新增到生產記錄中。使用確定控制可讓您確保涉及異動的所有變更能夠完成，或確保異動程序被岔斷時檔案能夠返回其原始狀態。

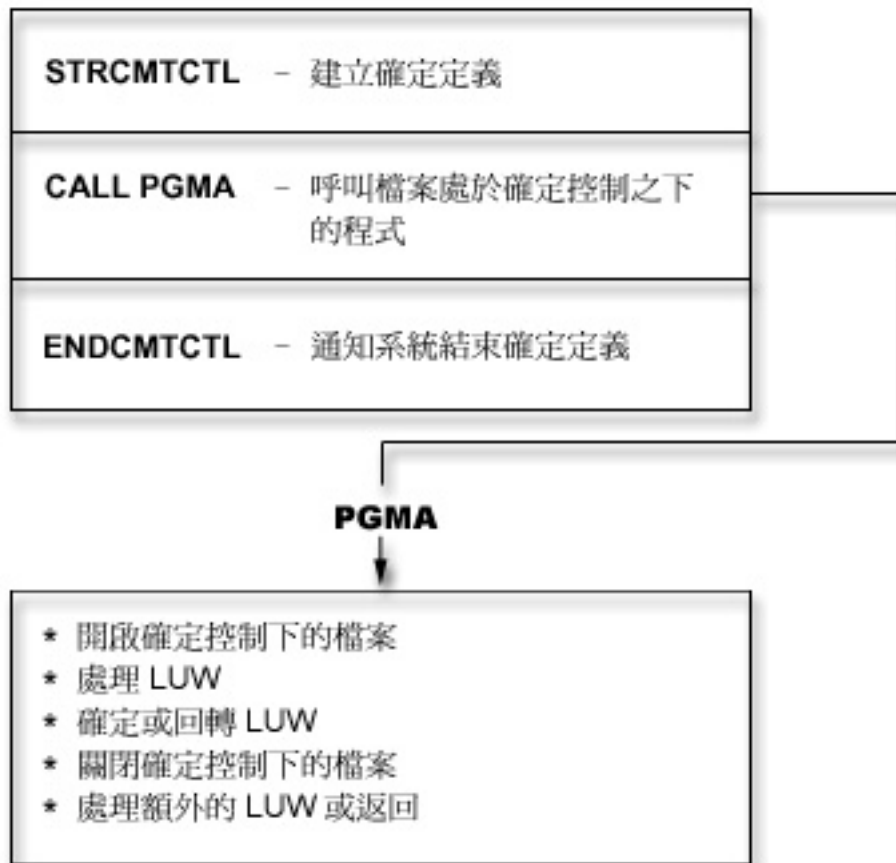
### 使用確定控制的 Program A

如果使用確定控制，則上述範例的變更如下所示：

1. 啟動確定控制。
2. Program A 鎖定並擷取倉庫記錄。(如果記錄被另一個程式鎖定，此動作可能需要等待)。
3. Program A 鎖定並擷取生產記錄。(此動作也需要等待)。Program A 目前鎖定了兩個記錄，其他程式無法變更它們。
4. Program A 更新倉庫記錄，確定控制對記錄保持鎖定。
5. Program A 更新生產記錄，確定控制對記錄保持鎖定。
6. Program A 確定異動。倉庫記錄和生產記錄的變更在檔案中成為永久變更。這些變更記錄於異動日誌中，並假設它們會出現於磁碟上。確定控制釋放兩記錄的鎖定。現在可讀取這些記錄以供其他任何程式更新。

由於兩記錄的鎖定由確定控制所保存，直到異動被確定為止，所以無法發生一個記錄被更新而另一個記錄未被更新的狀況。若於異動確定前發生路線步驟或系統失敗，系統會移除 (回轉) 所做的變更，使檔案更新至前次異動已確定的位置。

對於使檔案處於確定控制下的每一個路線步驟而言，會出現下圖所示的步驟：



在確定控制下執行的作業會被登載至異動日誌。啟動確定控制異動日誌登錄會出現於確定控制下的第一個檔案開啓登錄後面。此因為第一個檔案開啓登錄可決定確定控制使用的異動日誌。所以，第一個開啓作業的異動日誌登錄可用來檢查後續開啓作業，確定所有檔案使用相同異動日誌。

當發生工作失敗或系統失敗時，確定控制下的資源會更新至確定界限。如果異動啟動但未在路線步驟結束前完成，則該異動會被系統回轉，且在路線步驟結束後不出現於檔案中。如果系統在異動完成前異常終止，則該異動會被系統回轉，且在授權內碼的後續順利完成起始程式載入 (IPL) 後不出現於檔案中。只要發生回轉，反轉的登錄便會被存放到異動日誌中。

例如，假設 JKL 公司庫存中有 100 個二極體。製造部門從庫存中取出 20 個，新的結餘為 80。資料庫更新會產生未更新表格內容 (100) 和已更新表格內容 (80) 異動日誌登錄。

假設系統在登載這些登錄後異常終止，但之前到達了確定點或回轉點。在 IPL 之後，系統會讀取異動日誌登錄並更新對應的資料庫記錄。此更新會產生兩個反轉更新的異動日誌登錄：第一個登錄是未更新表格內容 (80)，第二個登錄是已更新表格內容 (100)。

當 IPL 在異常終止後順利完成，系統會移除 (或回轉) 任何不確定的資料庫變更。在前述範例中，系統從倉庫記錄中移除變更，因為確定作業不在該異動的異動日誌中。在此情況下，倉庫記錄的未更新表格內容會被置於檔案中。異動日誌含有回轉的變更，以及發生回轉作業的指示。



## 確定控制的練習問題

此練習問題將協助您瞭解確定控制及其基本要求。下列步驟假設您已熟悉 OS/400 授權程式和資料檔公用程式 (DFU)，且已閱讀本主題。邏輯流程將協助您進一步瞭解這個確定控制的練習程式。

註：請閱讀程式碼範例不保證聲明以了解重要的法律資訊。

於開始這個問題之前，請執行下列動作：

- 為此練習問題建立一個特殊檔案庫。於指示中，此檔案庫稱為 CMTLIB。在出現 CMTLIB 的地方，請置換成您的檔案庫名稱。
- 建立原始檔和工作說明。

執行下列步驟：

1. 建立一個實體檔案，叫作 ITMP (項目主檔案)。此檔案的資料說明規格 (DDS) 如下：

```
10  A   R ITMR
20  A   ITEM  2
30  A   ONHAND 5  0
40  A   K ITEM
```

2. 建立一個實體檔案，叫作 TRNP (異動檔)。此檔案用來做為異動日誌檔。此檔案的 DDS 如下：

```
10  A   R TRNR
20  A   QTY   5  0
30  A   ITEM  2
40  A   USER 10
```

3. 建立一個邏輯檔案，叫作 TRNL (異動邏輯)。此檔案用來協助重新啟動應用程式。USER 欄位為 LIFO 順序類型。此檔案的 DDS 如下：

```
10                                LIFO
20  A   R TRNR                    PFILE (TRNP)
30  A   K USER
```

4. 輸入 STRDFU 指令，並為 ITMP 檔案建立一個 DFU 應用程式，叫作 ITMU。在應用程式定義期間，接受 DFU 提供的預設值。
5. 鍵入指令 CHGDTA ITMU，為 ITMP 檔案輸入下列記錄：

項目	現有
AA	450
BB	375
CC	4000

6. 使用 F3 來結束程式。此登錄提供程式將操作的一些資料。
7. 建立 CL 程式「項目處理 (ITMPCSC)」如下：

```
PGM
DCL &USER *CHAR LEN(10)
RTVJOBA USER(&USER)
CALL ITMPCS PARM(&USER)
ENDPGM
```

此為呼叫 ITMPCS 程式的控制程式。它會擷取使用者名稱，然後傳送到處理程式。此應用程式假設使用唯一的使用者名稱。

8. 由 DDS 建立一個名為 ITMPCSD 的顯示檔案，如下所示。

有兩種格式，第一種是基本提示顯示，第二種可讓操作員複查已輸入的前次異動。此顯示檔案由 ITMPCS 程式所使用。

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      A          R PROMPT
2.00      A
3.00      A          CA03(93 'End of program')
4.00      A          CA04(94 'Review last')
5.00      A          SETOFF(64 'No rcd to rvw')
6.00      A          1 2'INVENTORY TRANSACTIONS'
7.00      A          3 2'Quantity'
8.00      A 61      QTY          5 0I  +1
9.00      A          ERRMSG('Invalid +
10.00     A          quantity' 61)
11.00     A          +5'ITEM'
12.00     A          2  I  +1
13.00     A 62      ITEM          2  I  +1
14.00     A          ERRMSG('Invalid +
15.00     A          Item number' 62)
16.00     A 63      ERRMSG('Rollback +
17.00     A          occurred' 63)
18.00     A 64      24 2'CF4 was pressed and +
19.00     A          there are no +
20.00     A          transactions for +
21.00     A          this user'
22.00     A          DSPATR(HI)
23.00     A          23 2'CF4 Review last +
24.00     A          transaction'
25.00     A          R REVW
26.00     A          1 2'INVENTORY TRANSACTIONS'
27.00     A          +5'REVIEW LAST TRANSACTION'
28.00     A          3 2'Quantity'
29.00     A          QTY          5 0  +1EDTCDE(Z)
30.00     A          +5'Item'
31.00     A          ITEM          2  +1

```

9. 研究確定控制的練習程式之邏輯流程中所提供的邏輯流程。

10. 輸入 STRSEU 指令，鍵入原始碼如下：

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      FITMP  UF  E          K          DISK
2.00      F*
3.00      FTRNP  0  E          DISK
4.00      F*
5.00      FTRNL  IF  E          K          DISK
6.00      F          TRNR
7.00      FITMPCSD CF E          WORKSTN
8.00      C* Enter parameter with User name for -TRNP- file
9.00      C          *ENTRY  PLIST
10.00     C          PARM          USER  10
11.00     C          LOOP      TAG
12.00     C          EXFMTPROMPT
13.00     C* Check for CF3 for end of program
14.00     C 93      DO
15.00     C          SETON          LR
16.00     C          RETRN
17.00     C          END
18.00     C* Check for CF4 for review last transaction
19.00     C 94      DO
20.00     C* Check for existence of a record for this user in -TRNL- file
21.00     C          USER      CHAINTRNR1          64  Not found
22.00     C 64      GOTO LOOP
23.00     C          EXFMTRREVW
24.00     C          GOTO LOOP
25.00     C          END
26.00     C* Access Item record
27.00     C          ITEM      CHAINITMR          62  Not found
28.00     C* Handle -not found- Condition
29.00     C 62      GOTO LOOP
30.00     C* Does sufficient quantity exist

```

```

31.00 C          ONHAND  SUB QTY      TEST  50  61 Minus
32.00 C* Handle insufficient quantity
33.00 C    61          DO
34.00 C* Release Item record which was locked by the CHAIN for update
35.00 C          EXCPTRLSITM
36.00 C          GOTO LOOP
37.00 C          END
38.00 C* Change ONHAND and update the Item record
39.00 C          Z-ADDTEST      ONHAND
40.00 C          UPDATITMR
41.00 C* Test for Special Simulation Conditions
42.00 C          ITEM      IFEQ 'CC'
43.00 C*          Simulate program need for rollback
44.00 C          QTY      IFEQ 100
45.00 C          SETON                      63      Simult Rlbck
46.00 C*          ROLBK
47.00 C          GOTO LOOP
48.00 C          END
49.00 C*          Simulate an abnormal program cancellation by Div by zero
50.00 C*          Operator Should respond -C- to inquiry message
51.00 C          QTY      IFEQ 101
52.00 C          Z-ADD0      ZERO  30
53.00 C          TESTZ      DIV ZERO      TESTZ  30      Msg occurs
54.00 C          END
55.00 C*          Simulate an abnormal job cancellation by DSPLY.
56.00 C*          Operator Should System Request to another job
57.00 C*          and cancel this one with OPTION(*IMMED)
58.00 C          QTY      IFEQ 102
59.00 C          'CC=102' DSPLY                      Msg occurs
60.00 C          END
61.00 C          END                      ITEM=CC
62.00 C* Write the -TRNP- file
63.00 C          WRITETRNR
64.00 C* Commit the update to -ITMP- and write to -TRNP-
65.00 C*          COMIT
66.00 C          GOTO LOOP
67.00 OITMR  E          RLSITM

```

11. 輸入 CRTRPGPMR 指令，使用上一個步驟中輸入的原始碼來建立程式 ITMPCS。
12. 鍵入指令 CALL ITMPCSC，按下 Enter 鍵，再按下 F4。應該出現訊息指出沒有這個操作員的登錄。
13. 輸入下列資料，查看程式是否正確操作：

數量	項目
3	AA
4	BB

14. 按下 F4。複查顯示畫面應該會出現且具有最後輸入的 BB 項目。輸入下列資料：

數量	項目
5	FF (應該發生無效項目數訊息)。
9000	BB (應該發生數量不足之錯誤訊息)。
100	CC (應該發生回轉訊息)。
102	CC (應該發生 RPG DSPLY 作業。請按 Enter 鍵)。
101	CC (程式應該顯示查詢訊息，根據工作屬性 INQMSGRPY 的設定，指出除以 0 狀況已發生或結束。如果出現了查詢訊息，請輸入 C 來取消 RPG 程式，然後在後續的查詢上輸入 C 來取消 CL 程式。此模擬異常錯誤狀況)。

15. 鍵入「顯示資料」指令 DSPDTA ITMP。

查看記錄 AA 和 BB 是否已正確地更新。值應該是 AA = 447、BB = 371 及 CC = 3697。注意，發生從 CC 扣除數量，但未寫入異動記錄。

16. 為確定控制建立異動日誌接收器。使用「建立異動日誌接收器」(CRTJRNRCV) 指令，在 CMRLIB 檔案庫中建立一個稱為 RCVR1 的異動日誌接收器。指定一個至少為 5000KB 的臨界值。如果系統有足夠的空間，建議使用較大的臨界值，以便將建立新異動日誌接收器之間的時間放至最大，將過於頻繁變更日誌的效能影響縮至最小。
17. 為確定控制建立日誌。使用「建立日誌 (CRTJRN)」指令，在 CMTLIB 檔案庫中建立一個稱為 JRNTEST 的日誌。因為此日誌僅使用於確定控制，請指定 MNGRCV(\*SYSTEM) DLTRCV(\*YES)。對於 JRNRCV 參數，請指定您在步驟 16 (請參閱 62) 中所建立的異動日誌接收器。
18. 使用「啟動日誌實體檔案 (STRJRNPF)」指令和參數 FILE(CMTLIB/ITMP CMTLIB/TRNP) JRN(CMTLIB/JRNTEST) 來登載要使用於確定控制的檔案。  
IMAGES 參數使用預設值 \*AFTER，表示只有記錄之已更新表格內容的變更才會出現於日誌中。檔案 ITMP 和 TRNP 現已啟動日誌登載。  
通常，您會於啟動日誌登載之後儲存檔案。對於沒有與異動日誌登錄相同 JID 的復置檔案，您無法套用登載的變更。因為此練習問題不需要您套用登載的變更，所以您可以省略儲存日誌登載檔案的動作。
19. 鍵入指令 CALL ITMPCSC，並輸入下列異動：

數量	項目
5	AA
6	BB

按一下 F3 以結束程式。

20. 鍵入「顯示日誌」指令：DSPJRN CMTLIB/JRNTEST。  
注意日誌中所顯示的登錄。日誌中的登錄順序 (「R UP = ITMP 的更新」緊接著「R PT = 新增到 TRNP 的記錄」) 與程式執行時相同。這是因為邏輯檔案定義於實體檔案 TRNP 之上，系統置換 RPG 預設值。如果不存在邏輯檔案，則使用 RPG 假設 SEQONLY(\*YES)，將出現一段 PT 登錄，因為區塊填滿之前，記錄會保存在 RPG 緩衝區中。
21. 變更 CL 程式 ITMPCSC 如下 (新的陳述式以星號顯示)：

```

PGM
DCL &USER *CHAR LEN(10)
RTVJOBA USER(&USER)
* STRCMTCTL LCKLVL(*CHG)
CALL ITMPCS PARM(&USER)
* MONMSG MSGID(RPG9001) EXEC(ROLLBACK)
* ENDCMTCTL
ENDPGM

```

STRCMTCTL 指令設定確定控制環境。LCKLVL 這個字指定為了更新而讀取但並未更新的記錄，可於異動期間釋放。MONMSG 指令處理任何 RPG 跳離訊息，且執行 ROLLBACK 以防 RPG 程式異常結束。ENDCMTCTL 指令結束確定控制環境。

22. 刪除現有的 ITMPCSC 程式，並重新建立它。
23. 變更 RPG 程式來移除陳述式 2.00、4.00、46.00 及 65.00 的註解符號。原始碼現在已備妥，可使用於確定控制。
24. 刪除現有的 ITMPCS 程式，並重新建立它。程式現已備妥，可於確定控制之下操作。
25. 鍵入指令 CALL ITMPCSC 和下列異動：

數量	項目
7	AA
8	BB

26. 使用「系統要求」，要求選項來顯示現行的工作。當「顯示工作」畫面出現時，選取選項 16 來要求顯示確定控制狀態。

注意顯示器上的值。應該有兩個確定，因為程式中已執行兩個確定陳述式。

27. 按一下 F9 來查看確定控制下的檔案清單及每一個檔案的活動數量。

28. 返回程式，並按一下 F3 來結束之。

29. 鍵入 DSPJRN CMTLIB/JRNTEST，並注意檔案的登錄及確定控制的特殊日誌登錄：

C BC	STRCMTCTL 指令發生。
C SC	啓動確定週期。當異動中的第一個資料庫作業造成於確定控制下插入、更新或刪除一筆記錄時，就會發生這種情況。
C CM	確定作業已發生。
C EC	ENDCMTCTL 指令發生。

即使您最初已對日誌要求 IMAGES(\*AFTER)，確定控制的未更新表格內容和已更新表格內容 (R UB 和 R UP 類型) 也會自動發生。

30. 鍵入指令 CALL ITMPCSC 和下列異動：

數量	項目
12	AA
100	CC (此為模擬應用程式需要回轉的狀況。會回轉 ITMP 檔案中被 RPG 陳述式 40.00 更新的 CC 記錄)。

31. 按一下 F4 來決定輸入的前次異動。

前次確定的異動為項目 AA 的登錄。

32. 使用「系統要求」，並要求「顯示現行工作」選項。當「顯示工作」畫面出現時，要求顯示確定控制狀態。

注意顯示器上的值以及回轉如何變更它們。

33. 返回程式。

34. 返回基本提示顯示畫面，並按一下 F3 來結束程式。

35. 鍵入指令 DSPJRN CMTLIB/JRNTEST。

注意爲了使用回轉登錄而出現在日誌中的額外登錄 (C RB 登錄)。當回轉 ITMP 記錄時，將有三個登錄會置於日誌中。這是因爲變更確定控制下的資料庫檔案會產生一個未更新 (R BR) 和已更新 (R UR) 的登錄。

36. 顯示有異動日誌碼 R 的登錄和這些登錄類型：UB、UP、BR 及 UR。使用選項 5 來顯示完整登錄。因爲數量欄位爲壓縮十進位數，請使用 F11 來要求顯示十六進位。注意下列各項：

- UB 記錄中的 ITMP 記錄的現有值
- UP 記錄如何減少現有值
- BR 記錄如何與 UP 記錄相同
- UR 記錄如何傳回 UB 原來顯示的值

最後登錄爲回轉結束的 RB 登錄。

37. 鍵入指令 CALL ITMPCSC，按一下 Enter 鍵，再按一下 F4。注意輸入的前次異動。

38. 鍵入下列異動：

數量	項目
13	AA

數量  
101

項目

CC (這是模擬異常錯誤狀況導致程式結束的狀況。此模擬是經由將一個欄位除以 0 而發生。程式會顯示查詢訊息或結束，視工作屬性 INQMSGRPY 的設定而定。如果查詢訊息出現，請輸入 C 來結束程式。因為 CL 程式已變更為監督 RPG 程式錯誤，之前發生的第二個查詢就不再發生)。

39. 鍵入指令 DSPJRN CMTLIB/JRNTEST。

相同類型的回轉處理已發生，但此次回轉是由 CL 程式中的 MONMSG 指令之 EXEC 參數所造成，不是由 RPG 程式造成。顯示兩個 RB 登錄來查看為哪一個程式所造成的。

40. 鍵入指令 WRKJOB，並記下完整的工作名稱，以供稍後使用。

41. 鍵入指令 CALL ITMPCSC，並輸入下列異動：

數量  
14  
102

項目

AA

CC (RPG DSPLY 作業應該發生在外部訊息佇列。使用「系統要求」鍵，並選取系統要求功能表上的選項 1 來轉送至輔助工作)。

42. 登入第二個工作，並重建您的環境。

43. 鍵入指令 ENDJOB，並指定先前識別的完整工作名稱和 OPTION(\*IMMED)。此模擬異常工作或系統結束。

44. 等待 30 秒，鍵入指令 CALL ITMPCSC，並按一下 F4。

注意前次確定的異動。它應該是先前輸入的 AA 項目。

45. 返回基本提示顯示畫面，並按一下 F3 以結束程式。

46. 鍵入指令 DSPJRN CMTLIB/JRNTEST。

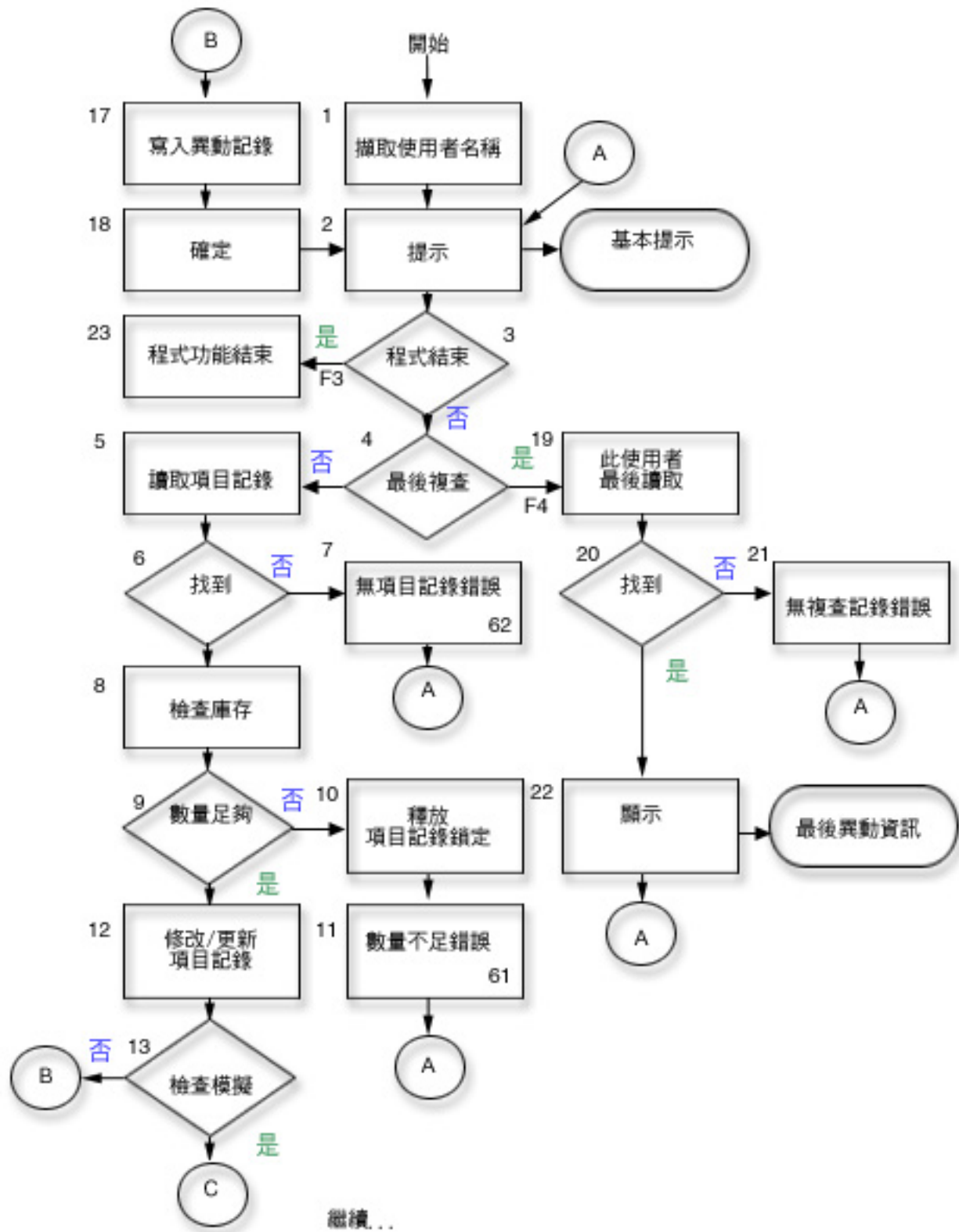
相同類型的回轉處理已發生，但這一次回轉是由系統所造成，不是由其中一個程式造成。RB 登錄由程式 QWTPITPP 寫入，此程式是工作管理異常終止程式。

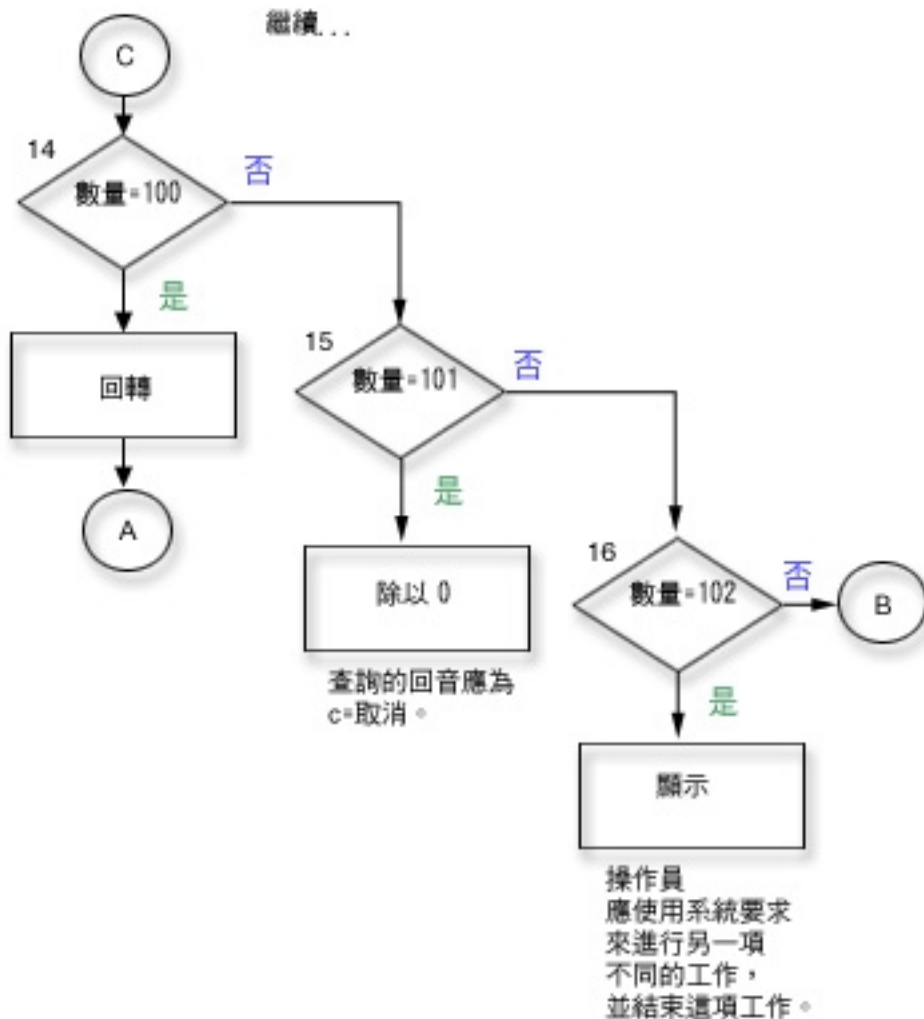
您現在已使用了確定控制的基本功能。您可以在應用程式上繼續進行確定控制，或嘗試其他功能，例如：

- 使用通知物件
- 使用 LCKLVL(\*ALL) 來鎖定唯讀的記錄
- 使用 LCKLVL(\*ALL) 來鎖定相同檔案中的多重記錄

## 練習問題的邏輯流程

下圖顯示確定控制的練習問題流程。在進行流程時，按一下有數字的形狀可查看與邏輯流程相關的每一個步驟。您也可以跳至與練習程式的邏輯流程相關的步驟來查看所有步驟。





### 與練習程式的邏輯流程相關的步驟

下列步驟與練習問題的邏輯流程相關。

1. 擷取以參數傳送的使用者名稱。這是用來寫入 TRNP 檔，也是用來擷取每一個操作員輸入的前次異動。這個應用程式假設每一個操作員有唯一的使用者名稱。
2. 使用格式名稱 PROMPT 提示基本顯示畫面。
3. 如果按 F3，則啟動結束程式功能。
4. 如果按 F4，則啟動常式以存取操作員輸入的前次異動。
5. 使用欄位 ITEM 讀取項目記錄。因為此檔案是更新檔案，所以這個要求鎖定記錄。
6. 在檔案 ITMP 檢查找不到的狀況。
7. 如果沒有 ITMP 記錄，開啓指示符 62 以產生錯誤訊息並返回步驟 2 (請參閱 66)。
8. 從現有庫存量 (ONHAND) 扣除所要求的數量 (QTY) 並放到工作區。
9. 檢查是否有足夠數量以滿足要求。
10. 如果數量不夠，請解除鎖定 ITMP 檔案中的記錄。需要這個步驟，因為數量不足。
11. 開啓指示符 61 以發出數量不足的顯示錯誤訊息，並返回步驟 2 (請參閱 66)。
12. 為的餘額衡變更 ONHAND 欄位並更新 ITMR 記錄。



13. 在 ITEM 欄位中檢查可用來模擬需要 ROLLBACK 的狀況的特殊登錄。
14. 檢查 QTY=100。發出 ROLLBACK 作業。這個動作模擬程式感覺需要回轉的狀況。
15. 檢查 QTY=101。使程式發生異常而產生查詢訊息。對這個函數使用除以 0。除非工作說明 INQMSGRPH 選項提供自動回答，否則操作員應輸入 C 來取消程式。這個動作模擬發生異常錯誤且操作員取消程式的狀況。
16. 檢查 QTY=102。發出含有查詢作業的顯示畫面。這個動作會在這個步驟停止程式並允許使用「系統要求鍵」移至另一個工作。取消更新工作。這個動作模擬在確定界限中間發生異常工作或系統結束的狀況。
17. 將異動記錄寫入 TRNP。
18. 確定異動的記錄並返回步驟 2 (請參閱 66)。
19. 讀取檔案 TRNL 的存取路徑上的第一筆記錄，使用 USER 作為鍵值。因為這個檔案是使用 LIFO 順序，所以這會是這位使用者輸入的前次異動記錄。
20. 在 TRNL 檔案中檢查找不到的狀況，若檔案不包含此使用者的登錄則會發生此狀況。
21. 如果沒有這位使用者的記錄，開啓指示符 64 以產生錯誤訊息並返回步驟 2 (請參閱 66)。
22. 顯示為這位使用者輸入的前次異動。如果操作員忘記先前輸入什麼內容或重新啓動異動的話，可以使用這個資訊。當操作員回應時，返回步驟 2 (請參閱 66)。
23. 執行任何結束程式功能。

### 範例：使用異動日誌檔來啓動應用程式

本範例提供如何在異常終止後使用異動日誌檔來啓動應用程式的程式碼範例及指示。

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

**異動日誌檔**用於未使用通知物件時，於系統或工作失敗後重新啓動應用程式。異動日誌檔通常在交談式應用程式中用來彙總異動影響。

例如，在訂單登錄應用程式中，所訂購每個項目的記錄通常寫入異動日誌檔中。此記錄包含訂購的項目、數量及價格。在應付帳款應用程式中，會在異動日誌檔中針對要收費的每個帳號寫入記錄。此記錄通常含有像帳戶號碼、消費金額及供應商等資訊。

在許多已有異動日誌檔的應用程式中，工作站使用者可要求輸入前次異動的相關資訊。藉由將確定控制新增到異動日誌檔已存在的應用程式中，您可以：

- 確定資料庫檔案更新至確定界限。
- 簡化異動的重新啓動。

如果您將異動日誌檔用於確定控制下重新啓動應用程式，您必須可以唯一地識別工作站使用者。如果系統上使用唯一的使用者設定檔名稱，該設定檔名稱可置於異動日誌記錄的某個欄位中。此欄位可作為檔案的索引鍵。

下列範例假設使用訂單庫存檔來執行異動且異動日誌檔已存在。程式執行下列動作：

1. 提示工作站使用者輸入數量和項目編號。
2. 更新生產主檔案 (PRDMSTP) 中的數量。
3. 將記錄寫入異動日誌檔 (ISSLOGL)。

如果現有的庫存數量不足，程式會拒絕異動。在寫入項目編號、說明、數量、使用者名稱及日期至異動日誌檔中後，工作站使用者可向程式詢問資料登錄的岔斷位置。

### 實體檔案 PRDMSTP 的 DDS

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          R PRDMSTR          TEXT('Master record')
2.00      A          PRODC T           3          COLHDG('Product' 'Number')
3.00      A          DESCRP           20         COLHDG('Description')
4.00      A          ONHAND            5 0        COLHDG('On Hand' 'Amount')
5.00      A                                     EDTCDE(Z)
6.00      A          K PRODC T

```

### ISSLOGP 所使用實體檔案 ISSLOGP 的 DDS

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          R ISSLOGR          TEXT('Product log record')
2.00      A          PRODC T           3          COLHDG('Product' 'Number')
3.00      A          DESCRP           20         COLHDG('Description')
4.00      A          QTY              3 0        COLHDG('Quantity')
5.00      A                                     EDTCDE(Z)
6.00      A          USER             10         COLHDG('User' 'Name')
7.00      A          DATE              6 0        EDTCDE(Y)
8.00      A          COLHDG('Date')

```

### 邏輯檔案 ISSLOGL 的 DDS

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          LIFO
2.00      A          R ISSLOGR          PFILE(ISSLOGP)
3.00      A          K USER

```

### 程式中所使用顯示檔案 PRDISSD 的 DDS

```

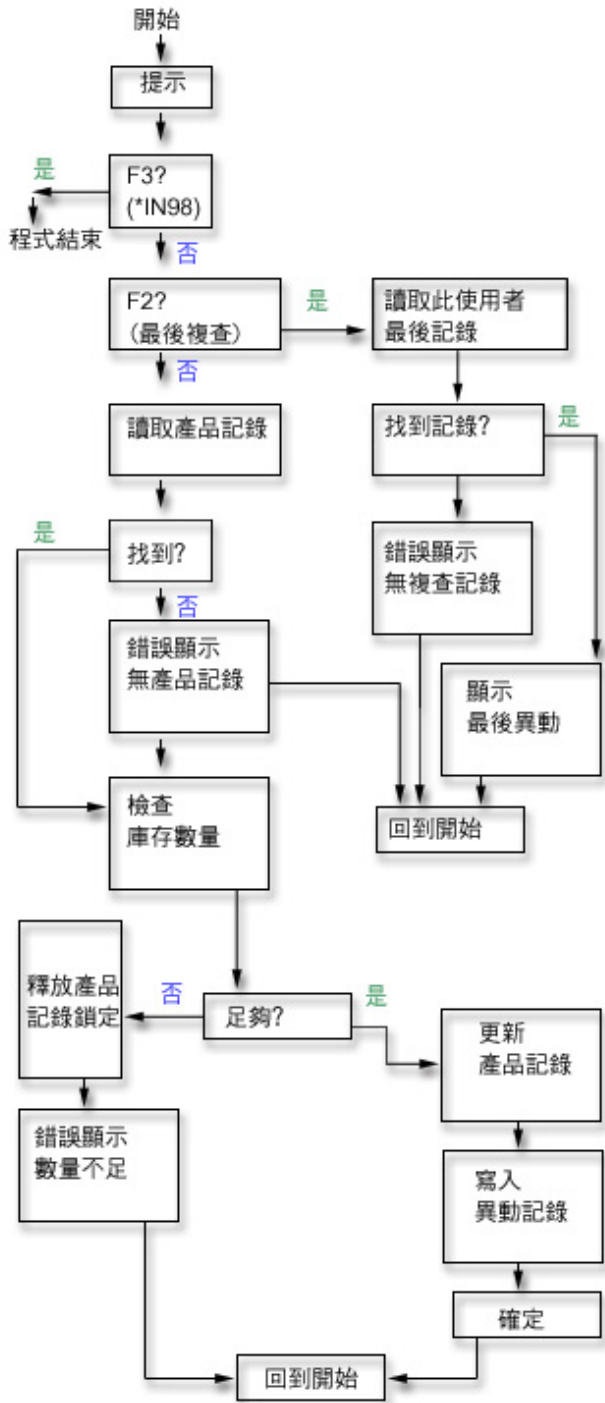
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      A          REF(ISSLOGP)
2.00      A          R PROMPT
3.00      A          CA03(98 'End of program')
4.00      A          CA02(97 'Where am I')
5.00      A          1 20'ISSUES PROCESSING'
6.00      A          3 2'Quantity'
7.00      A          QTY              R          I          +1
8.00      A 62          ERRMSG('Not enough +
9.00      A          Qty' 62)
10.00     A          +6'Product'
11.00     A          PRODC T           R          I          +1
12.00     A 61          ERRMSG('No Product +
13.00     A          record found' 62)
14.00     A 55          15 2'No Previous record exists'
15.00     A          24 2'CF2 Last transaction'
16.00     A          R RESTART
17.00     A          1 20'LAST TRANSACTION +
18.00     A          INFORMATION'
19.00     A          5 2'Product'
20.00     A          PRODC T           R          +1
21.00     A          7 2'Description'
22.00     A          DESCRP           R          +1
23.00     A          9 2'Qty'
24.00     A          QTY              R          +1REFFLD(QTY)

```

此程序描述程式流程。

### 程式流程



\*

更新 PRDMSTP 檔及寫入記錄到異動日誌檔中之後，指定 RPG COMMIT 作業碼。因為對操作員的每一個提示代表新異動的一個界限，所以異動被視為單一 Enter 異動。

當呼叫程式時，會將使用者名稱傳遞給程式。異動日誌檔的存取路徑以「後進先出 (LIFO)」順序定義，所以程式可輕易地存取輸入的上一筆記錄。

在系統或工作失敗之後，工作站使用者可利用指出資料登錄停止位置的相同功能來重新啟動程式。不需要新增額外程式碼至程式。若您目前使用異動日誌檔但未使用它來找出您所在的位置，請將使用者名稱新增至異動日誌檔中 (假設使用者名稱是唯一的)，並在程式中使用此方式。

下列顯示所使用的 RPG 程式。確定控制所需的陳述式以箭頭 (==>) 標示。

### RPG 程式

```

SEQNBR *... 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 ..
=>1.00  FPRDMSTP UP E          K          DISK          KCOMIT
=>2.00  FISSLOGL IF E          K          DISK          KCOMIT
3.00   PRDISSD CP E
4.00   *ENTRY          PLIST
5.00   PARM          USER 10
6.00   C*
7.00   C* Initialize fields used in Trans Log Rcd
8.00   C*
9.00   C          MOVE UPDATE DATE
10.00  C*
11.00  C* Basic processing loop
12.00  C*
13.00  C          LOOP TAG
14.00  C          EXFMTPROMPT
15.00  C 98          GOTO END          End of pgm
16.00  C 97          DO          Where am I
17.00  C          EXSR WHERE
18.00  C          GOTO LOOP
19.00  C          END
20.00  C          PRODC T CHAINPRDMSTR          61 Not found
21.00  C 61          GOTO LOOP
22.00  C          ONHAND SUB QTY TEST 50 62 Less than
23.00  C 62          DO          Not enough
24.00  C          EXCPTRLSMST          Release lock
25.00  C          GOTO LOOP
26.00  C          END
27.00  C*
28.00  C* Update master record and output the Transaction Log Record
29.00  C*
30.00  C          Z-ADDTEST ONHAND
31.00  C          UPDATPRDMSTR
32.00  C          WRITEISSLOGR
=>33.00 C          COMMIT
34.00  C          GOTO LOOP
35.00  C*
36.00  C* End of program processing
37.00  C*
38.00  C          END TAG
39.00  C          SETON          LR
40.00  C*
41.00  C* WHERE subroutine for "Where am I" requests
42.00  C*
43.00  C          WHERE BEGSR
44.00  C          USER CHAINISSLOGL          55 Not found
45.00  C N55          EXFMTRESTART
46.00  C          ENDSR
47.00  OPRDMSTR E          RLSMST

```

### 用來呼叫 RPG 程式 PRDISS 的 CL 程式

```

SEQNBR *... 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 ..
1.00   PGM
2.00   DCL          &USER *CHAR LEN(10)
3.00   STRCMTCTL  LCKLVL(*CHG)
4.00   RTVJOBA   USER(&USER)

```

5.00	CALL	PRDISS PARM(&USER)
6.00	MONMSG	MSGID(RPG9001) EXEC(ROLLBACK)
7.00	ENDCMTCTL	
8.00	ENDPGM	

若要在這個程式中使用確定控制，通常要指定鎖定層次 \*CHG。記錄會被變更鎖定至執行確定作業為止。請注意：如果庫存數量不足，記錄便會明確地被釋放。(如果記錄未在程式中明確地被釋放，則會在讀取檔案中下一筆記錄來更新時被釋放)。

在此範例中，使用鎖定層次 \*ALL 沒有其他優點。如果使用了 \*ALL，則當數量不足時，必須使用回轉或確定作業來釋放記錄。

上一個程式碼是呼叫 RPG 程式 PRDISS 的 CL 程式。注意 STRCMTCTL/ENDCMTCTL 指令用法。擷取唯一的使用者名稱 (RTVJOBA 指令) 並傳遞至程式。範例：使用標準處理程式來 啟動應用程式說明如何使用 MONMSG 指令來引起回轉。

### 範例：使用通知物件啟動應用程式

在異常終止之後啟動程式時，它可以在通知物件中尋找某登錄。如果有登錄存在，程式可以重新啟動異動。重新啟動異動之後，程式會清除通知物件以防止下次啟動相同異動。

下列是您可以使用通知物件的方法：

- 如果確定 ID 是放在資料庫檔案，請查詢這個檔案以判斷從何處重新啟動每一個應用程式或工作站工作。
- 如果確定 ID 是放在特定工作站的訊息佇列，那麼當工作站使用者登入時可傳送訊息給他們，通知他們已確定前次異動。
- 如果確定 ID 是放在有鍵值或使用者名稱的資料庫檔案，那麼當啟動程式時程式可以讀取這個檔案。如果檔案有記錄，請重新啟動程式。程式可以傳送訊息給工作站使用者來識別已確定的前次異動。程式執行任何回復。如果資料庫檔案有記錄存在，那麼程式會在結束程式時刪除該記錄。
- 對於批次應用程式來說，確定 ID 可以放在資料區，該區含有總計、切換設定和重新啟動應用程式需要的其他狀態資訊。啟動應用程式之後，它會存取資料區並驗證儲存在那裡的值。如果應用程式異常終止，那麼會設定資料區供下一次執行使用。
- 對於批次應用程式來說，確定 ID 可以傳給訊息佇列。當啟動應用程式時執行的程式可以從佇列擷取訊息並重新啟動程式。

根據您的應用程式需求，您可以使用數種方法重新啟動應用程式。在選擇方法時，請考慮下列事項：

- 同時有多位使用者使用一個程式時，無法使用單一資料區作為通知物件，因為在系統異常終止之後每一位使用者的確定 ID 會在資料區中彼此重疊。
- 您在設計刪除通知物件中的資訊時應該能夠處理在使用此資訊後立即發生失敗時的狀況：
  - 如果立即刪除資訊，那麼若在處理岔斷異動之前發生另一個失敗的話，該資訊就不存在了。
  - 應等到順利處理岔斷異動之後才刪除通知物件中的資訊。在這種狀況下，如果通知物件是資料庫檔案或訊息佇列，那麼通知物件中會有一個以上的登錄。
  - 如果有一個以上的登錄，那麼程式應存取最後一筆記錄。
- 無法使用通知物件提供工作站使用者已確定的前次異動，因為唯有發生系統或工作失敗或在正常結束工作時有未確定的變更時，才會更新通知物件。
- 如果工作站使用者看到資訊，那麼該資訊一定具有某種意義。若要完成這個動作可能需要程式將儲存在通知物件的程式碼轉換成會協助使用者重新啟動的資訊。
- 如果工作站使用者需要資訊時就應該重新顯示重新啟動的資訊。要防止不再有意義的資訊重新顯示，還需要程式的其他邏輯。

- 單一通知物件和標準處理程式可以提供重新啟動功能 (如果此通知物件是資料庫檔案的話)。需要重新啟動的能力的程式會呼叫這個標準處理程式，使每一個別程式的變更最小化。

關於使用通知物件的範例碼，請參閱下列各項：

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

- 每一個程式的唯一通知物件
- 所有程式的單一通知物件

### 範例：每一個程式的唯一通知物件

本主題提供範例程式碼和指示，說明使用唯一的通知物件來重新啟動每一個程式。

對每一個工作使用單一、唯一的通知物件，即使相同的程式有多個使用者，也可以使用外部說明確定 ID。在下列範例中，使用一個資料庫檔案當做通知物件，且只使用於此程式。

此程式有兩個資料庫檔案 (PRDMSTP 和 PRDLOCP)，必須加以更新才能收到庫存。程式所使用的顯示檔案稱為 PRDRCTD。以資料庫檔案 PRDRCTP 做為通知物件。此通知物件向程式定義為一個檔案，也用來做為通知功能的資料結構定義。

請參閱實體檔案 PRDMSTP 的 DDS (請參閱 67) 來查看實體檔案 PRDMSTP 的 DDS。

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

### 實體檔案 PRDLOCP 的 DDS

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7
1.00      A          R PRDLOCR          TEXT('Location record')
2.00      A          PRODC            3      COLHDG('Product' 'Number')
3.00      A          LOCATN           6      COLHDG('Location')
4.00      A          LOCAMT           5 0    COLHDG('Location' 'Amount')
5.00      A          EDTCDE(Z)
6.00      A          K PRODC
7.00      A          K LOCATN
```

### 顯示檔案 PRDRCTD 的 DDS

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..
1.00      A          REF(PRDMSTP)
2.00      A          R PROMPT
3.00      A          CA03(98 'End of program')
4.00      A          SETOFF(71 'RESTART')
5.00      A          1 20'PRODUCT RECEIPTS'
6.00      A          3  2'Quantity'
7.00      A          QTY              3 0I  +1
8.00      A          +6'Product'
9.00      A          PRODC            R      I  +1
10.00     A 61          ERRMSG('No record +
11.00     A          found in the +
12.00     A          master file' 62)
13.00     A          +6'Location'
14.00     A          LOCATN           R      I  +1REFFLD(LOCATN PRDLOCP)
15.00     A 62          ERRMSG('No record +
16.00     A          found in the +
17.00     A          location file' 62)
18.00     A          9  2'Last Transaction'
19.00     A 71          +6'This is restart +
20.00     A          information'
21.00     A          DSPATR(HI BL)
22.00     A          12 2'Quantity'
```

23.00	A			12	12'Product'
24.00	A			12	23'Location'
25.00	A			12	35'Description'
26.00	A	LSTPRD	R	14	15REFFLD(PRODCT)
27.00	A	LSTLOC	R	14	26REFFLD(LOCATN *SRC)
28.00	A	LSTQTY	R	14	5REFFLD(QTY *SRC)
29.00	A				EDTCDE(Z)
30.00	A	LSTDSC	R	14	35REFFLD(DESCRP)

### 通知物件及外部說明資料結構 (PRDRCTP) 的 DDS

SEQNBR \*... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00	A				LIFO
2.00	A				REF (PRDMSTP)
3.00	A	R	PRDRCTR		
4.00	A		USER	10	
5.00	A		PRODCT	R	
6.00	A		DESCRP	R	
7.00	A		QTY	3 0	
8.00	A		LOCATN	R	REFFLD(LOCATN PRDLOCP)
9.00	A	K	USER		

程式處理通知物件如下：

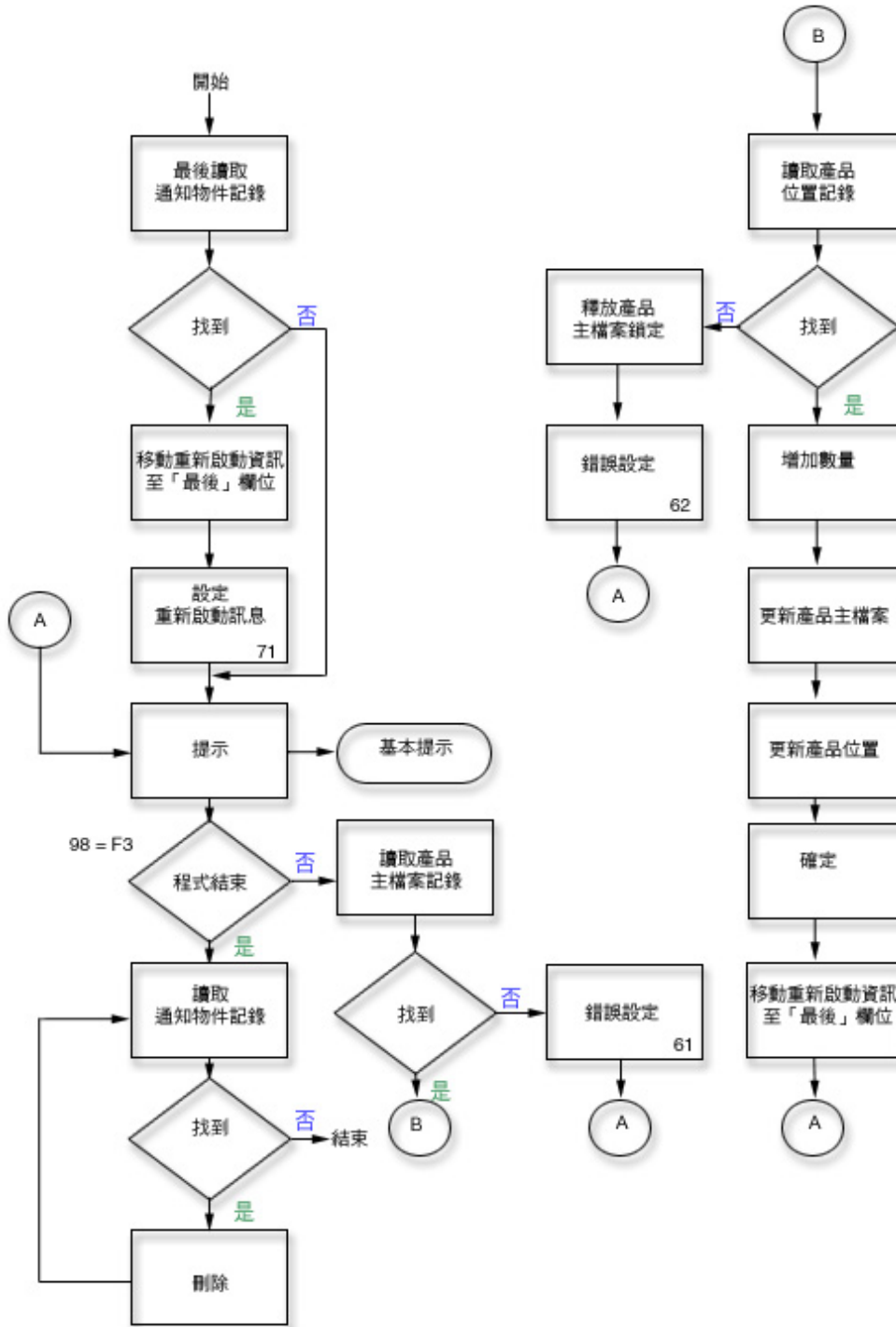
- 一開始，程式隨機處理通知物件，如果特定鍵有記錄存在，則顯示該記錄：
  - 如果存在多重記錄，則使用此鍵的最後一筆記錄，因為 PRDRCTP 檔案使用 LIFO 順序。
  - 如果沒有記錄，則不會岔斷異動，所以不必重新開始。
  - 如果程式在第一個順利確定作業之前失敗，則重新開始不是必要的。
- 清除通知物件的常式會在程式結束時執行：
  - 如果發生多個失敗，常式可以處理刪除通知物件中的多筆記錄。
  - 雖然系統會在資料庫檔案中放入確定 ID，但必須在 RPG 程式中指定確定 ID 為變數。
  - 因為 RPG 容許外部說明的資料結構，所以資料結構是指定確定 ID 的一種簡便方式。在本範例中，資料結構使用與資料庫檔案相同的外部說明做為通知物件。

此程式的處理程序提示使用者輸入產品編號、位置及數量：

- 必須更新兩個檔案：
  - 產品主檔案 (PRDMSTP)
  - 產品位置檔 (PRDLOCP)
- 更新每一個檔案或記錄前，每一個檔案中的記錄必須存在。
- 在順利輸入每一個異動之後，程式會將輸入欄位移動至幾個對應的最後欄位。在每一個提示上，這些最後欄位會顯示給操作員，代表對於最後輸入內容的意見。
- 如果有重新啟動的資訊，則移到這些最後的欄位，且顯示器上會出現特殊訊息。

下圖概述此處理。使用者名稱傳送到程式，以在通知物件中提供唯一的記錄。

### 程式流程



\*

下列是本範例的 RPG 原始程式碼。在程式開始和結束時，通知物件 (檔案 PRDRCTP) 做為正常檔案使用，在呼叫程式之前，也在 CL (STRCMTCTL 指令) 中指定為通知物件。

### RPG 原始碼



```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00   FPRDMSTP UF E           K           DISK           KCOMIT
2.00   FPRDLOCP UF E           K           DISK           KCOMIT
3.00   FPRDRCTD CF E           WORKSTN
4.00   F*
5.00   F* The following file is the specific notify object for this pgm.
6.00   F*       It is accessed only in a restart situation and at the
7.00   F*       end of the program to delete any records. The records
8.00   F*       are written to the notify object by Commitment Control.
9.00   F*
10.00  FPRDRCTP UF E           K           DISK
11.00  ICMTID      E DSPRDRCTP

12.00  C           *ENTRY   PLIST
13.00  C           PARM           USER10 10
14.00  C           MOVE USER10  USER
15.00  C*
16.00  C* Check for restart information - get last rcd per user
17.00  C*   PRDRCTP file access path is in LIFO sequence
18.00  C*
19.00  C           USER      CHAINPRDRCTR           20   Not found
20.00  C N20           DO           Restart
21.00  C           EXSR MOVLST           Move to last
22.00  C           SETON           71   Restart
23.00  C           END
24.00  C*
25.00  C* Basic processing loop
26.00  C*
27.00  C           LOOP      TAG
28.00  C           EXFMTPROMPT
29.00  C 98           GOTO END           End of pgm
30.00  C           PRODCY      CHAINPRDMSTR           61   Not found
31.00  C 61           GOTO LOOP
32.00  C           KEY        KLIST
33.00  C           KFLD           PRODCY
34.00  C           KFLD           LOCATN
35.00  C           KEY        CHAINPRDLOCR           62   Not found
36.00  C 62           DO
37.00  C           EXCPTRLMSST           Release lck
38.00  C           GOTO LOOP
39.00  C           END
40.00  C           ADD QTY      ONHAND           Add
41.00  C           ADD QTY      LOCAMT
42.00  C           UPDATPRDMSTR           Update
43.00  C           UPDATPRDLOCR           Update
44.00  C*

45.00  C* Commit and move to previous fields
46.00  C*
47.00  C           CMTID      COMIT
48.00  C           EXSR MOVLST           Move to last
49.00  C           GOTO LOOP
50.00  C*
51.00  C* End of program processing
52.00  C*
53.00  C           END      TAG
54.00  C           SETON           LR
55.00  C*56.00 C* Delete any records in the notify object
57.00  C*
58.00  C           DLTLPL      TAG
59.00  C           USER      CHAINPRDRCTR           20   Not found
60.00  C N20           DO
61.00  C           DELETPRDRCTR           Delete
62.00  C           GOTO DLTLPL
63.00  C           END
64.00  C*

```

```

65.00      C*  Move to -Last Used- fields for operator feedback
66.00      C*
67.00      C          MOVLST      BEGSR
68.00      C          MOVE  PRODC  LSTPRD
69.00      C          MOVE  LOCATN LSTLOC
70.00      C          MOVE  QTY   LSTQTY
71.00      C          MOVE  DESCRP LSTDSC
72.00      C          ENDSR
73.00      OPRDMSTR E          RLSMST

```

### 範例：適用全部程式的單一通知物件

將單一通知物件用於全部程式是有利的，因為重新啟動所需的全部資訊位於相同物件中，而且可在所有程式中使用通知物件的標準方式。在這個情況下，透過使用者與程式 ID 的唯一組合來確保程式重新啟動時存取的是正確資訊。

由於重新啟動所需的資訊可能視程式而有不同，所以不應使用確定 ID 的外部說明資料結構。如果使用單一通知物件，則上述程式說明的是程式內的資料結構，而非外部資料結構。例如：

```

1   10      USER
11  20      PGMNAM
21  23      PRODC  T
24  29      LOCATN
30  49      DESC
50  51  0   QTY
52  220     DUMMY

```

在使用此通知物件的每一個程式中，指定給確定 ID 的資訊對程式而言是唯一的（使用者和程式名稱不是唯一的）。通知物件必須夠大，足以包含任何程式將存放到確定 ID 中的最大資訊。

範例：適用每一個程式的唯一通知物件提供有關使用通知物件的其他範例。

### 範例：使用標準處理程式來啟動應用程式

標準處理程式為使用一個資料庫檔案作為所有應用程式的通知物件來重新啟動應用程式的一種方法。這個方法假設所有使用標準程式的應用程式中，使用者設定檔名稱都是唯一的。

在這個方法中，實體檔案 NPYOBNP 用來當作通知物件，且定義為：

```

唯一的使用者設定檔名稱      10 個字元
程式 ID                        10 個字元
使用於
    重新啟動的資訊            字元欄位
                                (這應該要足夠
                                包含最大數量的
                                重新啟動資訊，
                                以重新啟動
                                程式。
                                此欄位為應用
                                程式所必要的。
                                在本範例中，
                                假設其長度為 200)。

```

檔案是使用 SHARE(\*YES) 來建立的。檔案的前兩個欄位是檔案的鍵。(這個檔案也可以在 RPG 程式中定義成資料結構)。

註：請閱讀程式碼範例不保證聲明，以了解重要法律資訊。

下列提供標準處理程式的程式碼範例：

- 範例：標準處理程式的程式碼
- 範例：標準確定處理程式

- 範例：使用標準處理程式來決定重新啟動應用程式

### 範例：標準處理程式的程式碼

下列是使用標準處理程式的範例。下列程式碼範例中顯示的應用程式執行如下：

1. 應用程式在參數中接收使用者名稱，並且搭配程式名稱一起使用，在通知物件當做唯一的 ID。
2. 應用程式將要求碼 R 傳送到標準確定處理程式，由此程式決定通知物件中是否有記錄存在。
3. 如果標準確定處理程式傳送 1，表示找到一筆記錄，應用程式向使用者顯示需要重新啟動的資訊。
4. 應用程式繼續正常處理程序。
5. 異動完成時會儲存值以供參照，所以工作站使用者可以查看前次異動的情形。

儲存的資訊不是由通知物件提供，因為只有在工作或系統失敗發生時，通知物件才會更新。

關於這個程式的流程，請參閱處理流程。

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

### 應用程式範例

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      FPRDMSTP UF  E          K          DISK          KCOMIT
2.00      FPRDLOCP UF  E          K          DISK          KCOMIT
3.00      FPRDRCTD CF  E                      WORKSTN
4.00      F*
5.00      F* The following is a compile time array which contains the
6.00      F*   restart information used in the next example
7.00      F*
8.00      E                      RTXT   50  50  1                      Restart text
9.00      I*
10.00     I* Data structure used for info passed to notify object
11.00     I*
12.00     ICMTID      DS
13.00     I                      1  10  USER
14.00     I                      11  20  PGMNAM
15.00     I                      21  23  PRODCY
16.00     I                      24  29  LOCATN
17.00     I                      30  49  DESCRP
18.00     I                      P  50  510QTY
19.00     I                      52  170 DUMMY
20.00     I                      171 220 RSTART
21.00     C          *ENTRY  PLIST
22.00     C          PARM          USER10 10
23.00     C*
24.00     C* Initialize fields used to communicate with std program
25.00     C*
26.00     C          MOVE USER10  USER
27.00     C          MOVE 'PRDR2' PGMNAM
28.00     C          MOVE 'R'     RQSCOD          Read Rqs
29.00     C          CALL 'STDCMT'
30.00     C          PARM          RQSCOD  1
31.00     C          PARM          RTNCOD  1
32.00     C          PARM          CMTID 220          Data struct
33.00     C          RTNCOD  IFEQ '1'          Restart
34.00     C          EXSR MOVLST          Move to last
35.00     C SETON          71          Restart
36.00     C          END
37.00     C*
38.00     C* Initialize fields used in notify object
39.00     C*
40.00     C          MOVEARTXT,1  RSTART          Move text
41.00     C*
42.00     C* Basic processing loop

```

```

43.00      C*
44.00      C          LOOP      TAG
45.00      C          EXFMTPROMPT
46.00      C  98          GOTO END
47.00      C          PRODC   CHAINPRDMSTR          61  Not found
48.00      C  61          GOTO LOOP
49.00      C          KEY      KLIST
50.00      C          KFLD          PRODC
51.00      C          KFLD          LOCATN

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

52.00      C          KEY      CHAINPRDLOC   62  Not found
53.00      C  62          DO
54.00      C          EXCPTRLSMST          Release lck
55.00      C          GOTO LOOP
56.00      C          END
57.00      C          ADD QTY      ONHAND          Add
58.00      C          ADD QTY      LOCAMT
59.00      C          UPDATPRDMSTR          Update
60.00      C          UPDATPRDLOC   Update
61.00      C*
62.00      C* Commit and move to previous fields
63.00      C*
64.00      C          CMTID   COMIT
65.00      C          EXSR MOVLST          Move to last
66.00      C          GOTO LOOP
67.00      C* End of program processing
68.00      C*
69.00      C          END      TAG
70.00      C          MOVE 'D'      RQSCOD          Dlt Rqs
71.00      C          CALL 'STDCMT'
72.00      C          PARM          RQSCOD
73.00      C          PARM          RTNCOD
74.00      C          PARM          CMTID
75.00      C          SETON          LR
76.00      C*
77.00      C* Move to -Last Used- fields for operator feedback
78.00      C*
79.00      C          MOVLST   BEGSR
80.00      C          MOVE PRODC   LSTPRD
81.00      C          MOVE LOCATN  LSTLOC
82.00      C          MOVE DESCRP  LSTDSC
83.00      C          MOVE QTY     LSTQTY
84.00      C          ENDSR
85.00      OPRDMSTR E          RLSMST
86.00 ** RTXT          Restart Text
87.00 Inventory Menu - Receipts Option

```

## 範例：標準確定處理程式的程式碼

標準確定 (STDCMT) 處理程式執行與所有應用程式使用的單一通知物件通信所需要的功能。當確定控制功能將登錄自動寫入通知物件時，使用者撰寫的標準程式必須處理通知物件。標準程式簡化和標準化此方法。

撰寫程式驗證已傳送的參數並執行適當動作如下：

### O=開啓

呼叫程式要求在傳回時通知物件檔要保持開啓。因為 RPG 程式隱含地開啓通知物件，所以程式不可以關閉它。設定指示符 98 使得程式以 LR 的狀態返回以保留程式的工作區，並保持通知物件開啓，以便能夠重新呼叫它而不需要額外的執行時間。

### C=關閉

呼叫程式判斷不再需要通知物件並要求關閉。指示符 98 設定為關閉以允許完全關閉通知物件。

### R=讀取

呼叫程式要求讀取並傳回有符合鍵值欄位的記錄。程式使用傳送的鍵值欄位嘗試從 N FYO B J P 擷取記錄。如果有相同鍵的重複記錄，則傳回最後一筆記錄。會據此設定回覆碼，如果有此記錄，它會在資料結構 C M T I D 中傳回。

### W=寫入

呼叫程式要求將記錄寫入通知物件，讓呼叫程式能夠在下次呼叫它時重新啓動。程式把傳送的資料的內容撰寫成 N FYO B J P 中的記錄。

### D=刪除

呼叫程式要求刪除這個符合鍵值的記錄。通常在順利完成呼叫程式時會執行這個功能，以便在重新啓動時移除任何資訊。程式嘗試刪除傳送的鍵值欄位的記錄。如果沒有記錄，就會傳回另一個回覆碼。

### S=搜尋

呼叫程式要求搜尋特定使用者的記錄，不管哪一個程式寫入該記錄。在程式中使用這個功能來登入，以指出需要重新啓動。程式只使用使用者名稱作為鍵值來檢查是否有記錄存在。會適當地設定回覆碼，且讀取和傳回此鍵值的最後一筆記錄的內容 (如果有的話)。

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

下列項目顯示標準確定處理程式 STDCMT。

### 標準確定處理程式

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      FNFYOBJP UF  E      K      DISK      A
2.00      ICMTID      DS
3.00      I
4.00      I
5.00      I
6.00      C          *ENTRY  PLIST
7.00      C          PARM      RQSCOD 1
8.00      C          PARM      RTNCOD 1
9.00      C          PARM      CMTID 220
10.00     C          UNQUSR   CABEQ*BLANKS  BADEND      H1 Invalid
11.00     C          UNQPGM   CABEQ*BLANKS  BADEND      H2 Invalid
12.00     C*
13.00     C* 'O' for Open
14.00     C*
15.00     C          RQSCOD   IFEQ 'O'      Open
16.00     C          SETON      98      End LR
17.00     C          GOTO END
18.00     C          END
19.00     C*
20.00     C* 'C' for Close
21.00     C*
22.00     C          RQSCOD   IFEQ 'C'      Close
23.00     C          SETOF      98
24.00     C          GOTO END
25.00     C          END
26.00     C*
27.00     C* 'R' for Read - Get last record for the key
28.00     C*
29.00     C          RQSCOD   IFEQ 'R'      Read
30.00     C          KEY      KLIST
31.00     C          KFLD      UNQUSR
32.00     C          KFLD      UNQPGM
33.00     C          KEY      CHAINNFYOBJR   51      Not found
34.00     C 51      MOVE 'O'      RTNCOD
35.00     C 51      GOTO END

```

```

36.00 C          MOVE '1'      RTNCOD          Found
37.00 C          LOOP1      TAG
38.00 C          KEY       READENFYOBJR          20 EOF
39.00 C 20          GOTO END
40.00 C          GOTO LOOP1
41.00 C          END
42.00 C*
43.00 C* 'W' FOR Write
44.00 C*
45.00 C          RQSCOD     IFEQ 'W'          Write
46.00 C          WRITENFYOBJR
47.00 C          GOTO END
48.00 C          END
49.00 C*
50.00 C* 'D' for Delete - Delete all records for the key
51.00 C*
52.00 C          RQSCOD     IFEQ 'D'          Delete
53.00 C          KEY       CHAINNFYOBJR          51 Not found
54.00 C 51          MOVE '0'      RTNCOD
55.00 C 51          GOTO END
56.00 C          MOVE '1'      RTNCOD          Found
57.00 C          LOOP2      TAG
58.00 C          DELETNFYOBJR
59.00 C          KEY       READENFYOBJR          20 EOF
60.00 C N20         GOTO LOOP2
61.00 C          GOTO END
62.00 C          END
63.00 C*
64.00 C* 'S' for Search for the last record for this user
65.00 C*          (Ignore the -Program- portion of the key)
66.00 C*
67.00 C          RQSCOD     IFEQ 'S'          Search
68.00 C          UNQUSR     SETLLNFYOBJR          20 If equal
69.00 C N20         MOVE '0'      RTNCOD
70.00 C N20         GOTO END
71.00 C          MOVE '1'      RTNCOD          Found
72.00 C          LOOP3      TAG
73.00 C          UNQUSR     READENFYOBJR          20 EOF
74.00 C N20         GOTO LOOP3
75.00 C          GOTO END
76.00 C          END
77.00 C*
78.00 C* Invalid request code processing
79.00 C*
80.00 C          SETON          H2          Bad RQS code
81.00 C          GOTO BADEND
82.00 C*
83.00 C* End of program processing
84.00 C*
85.00 C          END          TAG
86.00 C N98          SETON          LR
87.00 C          RETRN
88.00 C* BADEND tag is used then fall thru to RPG cycle error return
89.00 C          BADEND      TAG

```

## 範例：使用標準處理程式來決定是否重新啓動應用程式

此主題提供範例 CL 程式碼，以使用標準處理程式來決定異常 IPL 後是否重新啓動應用程式。

起始程式可呼叫標準確定處理程式來決定是否需要重新啓動。而工作站使用者可決定是否重新啓動。

起始程式傳遞要求碼 S (搜尋) 給標準程式，搜尋使用者的任何記錄。如果有記錄存在的話，則會傳遞重新啓動的資訊給起始程式並對工作站使用者顯示。

通知物件中的確定 ID 應包含起始程式可顯示來識別哪些程式需要重新啟動的資訊。例如，可保留確定 ID 最後 50 個字元來包含此資訊。在應用程式中，此資訊可能位於編譯時間陣列中並在起始設定步驟中移至資料結構。範例：標準確定處理程式的程式碼顯示如何將此併入應用程式中。

下列為決定記錄是否存在於通知物件中的起始程式範例。

註：請閱讀程式碼範例不保證聲明以了解重要法律資訊。

### 起始程式範例

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      PGM
2.00      DCLF          CMTINLD
3.00      DCL          &RQSCOD *CHAR LEN(1) VALUE(S) /* Search */
4.00      DCL          &RTNCOD *CHAR LEN(1)
5.00      DCL          &CMTID *CHAR LEN(220)
6.00      DCL          &USER *CHAR LEN(10)
7.00      DCL          &INFO *CHAR LEN(50)
8.00      RTVJOBA      USER(&USER)
9.00      CHGVAR       &CMTID (&USER *CAT XX)
10.00     /* The XX is required to prevent a blank Pgm nam */
11.00     CALL         STDCMT PARM(&RQSCOD &RTNCOD &CMTID)
12.00     IF          (&RTNCOD *EQ '1') DO /* RESTART REQD */
13.00     CHGVAR       &INFO %SST(&CMTID 171 50)
14.00     SNDRCVF      RCD_FMT(RESTART)
15.00     ENDDO
16.00     /*
17.00     /* Enter normal initial program statements */
18.00     /* or -TFRCTL- to first menu program */
19.00     /*
20.00     ENDPGM

```

---

## 疑難排解異動和確定控制

下面幾頁提供您要對確定控制進行疑難排解所需要的一些作業：

### 確定控制錯誤

這個資訊說明產生錯誤的狀況、列出確定控制錯誤以及提供處理錯誤的方法。

### 偵測死結

這個作業的目標為尋找死結狀況。

### 在通信失敗之後回復異動

這個作業的目標是在與遠端系統通信失敗之後處理那些在遠端系統執行工作的異動。

### 何時強制確定和回轉以及何時取消重新同步

此資訊說明何時和如何強制回轉或確定，以及何時取消重新同步。

## 確定控制錯誤

當使用確定控制時，請務必瞭解哪些狀況會導致錯誤發生而哪些不會。一般而言，當確定控制功能的使用不一致時會發生錯誤，例如在使用確定定義的檔案仍開啓時執行「結束確定控制 (ENDCMTCTL)」指令。

### 確定處理程序期間發生錯誤

如果在確定作業期間發生通信或系統失敗，則需要執行重新同步來確保異動管理程式使涉及異動的所有系統保持資料一致。重新同步的操作方式及影響確定作業的程度視以下因素而定：

- 等待結果確定選項。詳細資料，請參閱兩階段確定的確定定義：不等待結果。
- 異動狀態。詳細資料，請參閱兩階段確定控制的異動狀態。

如果失敗情形嚴重，例如無法再被修復或無法以適當方式修復，則涉及異動的其他系統之系統操作員必須使用試探法。試探法可確定或回轉異動期間對該系統所做的變更。若於此種試探法後修復了失敗，但重新同步偵測到該試探法造成資料完整性問題，則會傳送訊息 CPD83D9 或 CPD83E9 至 QSYSOPR 訊息佇列。

下列資訊提供有關使用確定控制發生錯誤的詳細資料：

- 錯誤狀況
- 無錯誤狀況
- 於確定控制期間 監督的錯誤訊息
- 在 CALL 指令後 監督的錯誤訊息
- 一般確定或回轉處理程序的失敗

## 錯誤狀況

如果發生錯誤，則會送出一則供您在程式中監督的跳出訊息。下列是一些與確定控制相關的典型錯誤：

- 執行了連續 STRCMTCTL 指令，但沒有執行岔斷 ENDCMTCTL 指令。
- 檔案在確定控制下開啓，但未執行 STRCMTCTL 指令。

對於啓動群組內所執行使用「工作層次」確定定義的程式而言，這不是錯誤狀況。「工作層次」確定定義只能由單一程式啓動，一旦被程式啓動，則在非使用「啓動群組層次」確定定義的任何啓動群組中執行的任何程式，皆可使用工作層次確定定義。在啓動群組內執行使用「啓動群組層次」確定定義的程式，必須先使用 STRCMTCTL 指令啓動「啓動群組層次」確定定義。

- 在確定控制下開啓用於輸出的檔案未被登載。
- 共用檔的第一個開啓作業將檔案置於確定控制下，但相同共用檔的後續開啓作業沒有這麼做。
- 共用檔的第一個開啓作業未將檔案置於確定控制下，但相同共用檔的後續開啓作業有這麼做。
- 在單一異動中，已達工作的記錄鎖定限制。
- 程式發出讀取作業、確定作業及相同記錄的變更。讀取作業必須在確定作業後重新發出，因為確定作業釋放了記錄鎖定。
- 若是一階段位置，確定控制下的資源與已在確定定義的確定控制下之資源不位於相同位置。
- 發出 ENDCMTCTL 指令時有不確定的變更存在。

如果所有檔案已關閉、任何遠端資料庫已斷線，而且沒有 API 確定資源仍與所要結束的確定定義相關，則對於 ENDCMTCTL 指令而言，這不是錯誤狀況。

- 執行了確定、回轉或 ENDCMTCTL 指令，但未執行 STRCMTCTL 指令。

對於在啓動群組內執行且「工作層次」確定定義作用中的程式而言，這不是錯誤狀況。「工作層次」確定定義只能由單一程式啓動，一旦被程式啓動，則在未使用「啓動群組層次」確定定義的任何啓動群組中執行的任何程式，皆可使用工作層次確定定義。在啓動群組內執行使用「啓動群組層次」確定定義的程式，必須先使用 STRCMTCTL 指令來啓動「啓動群組層次」確定定義。

- 對仍在確定定義的確定控制下開啓的檔案執行 ENDCMTCTL 指令。
- 執行儲存作業的工作具有不在確定界限上的一或多個確定定義。
- 由於具有可確定資源的其他工作未在指定給 SAVACTWAIT 參數的時間內到達確定界限，所以結束了「作用中儲存」程序。
- 由於 API 可確定資源已新增至單一工作的多個確定定義中，所以無法繼續「作用中儲存」程序。
- 單一工作有超過 1023 個確定定義存在。



- 遠端位置的交談因資源失效而遺失。這會導致異動回轉。
- 開啓以便更新的一階段資源出現在未起始確定作業的節點上。您必須移除起始了確定要求的資源或節點。
- 當異動處於需要回轉 (RBR) 狀態時要求確定作業。回轉作業必須完成。
- API 跳出程式發出確定要求或回轉要求。
- 觸發程式對呼叫它時所依據的確定定義發出確定要求或回轉要求。

觸發程式可啓動個別的確定定義並對該定義發出確定或回轉要求。

## 無錯誤狀況

您可能期望這些狀況會導致錯誤訊息。不過，確定控制允許這些狀況。下列為確定控制中沒有發生錯誤的一些狀況：

- 確定或回轉作業在執行但資源不在確定控制下。這可讓您將確定或回轉作業併入程式中，不需要考慮是否有資源在確定控制下。它也可讓您在做任何可確定變更之前指定確定 ID。
- 確定或回轉作業在執行但沒有未確定的資源變更。這可讓您將確定或回轉作業併入程式中，不需要考慮是否有未確定的資源變更。
- 確定控制下的檔案關閉且未確定的記錄存在。此狀況允許呼叫另一個程式來執行確定或回轉作業。這無關檔案是否共用。此功能允許子程式執行屬於涉及多重程式異動的資料庫變更。
- 具有一或多個確定定義的未確定變更之工作正常或異常結束。所有確定定義的變更被回轉。
- 具有「啓動群組層次」確定定義的擱置中變更之啓動群組結束。如果啓動群組正常結束，且在結束的相同啓動群組限定範圍的確定控制下所開啓的任何檔案關閉時，沒有發生任何錯誤，系統會執行隱含的確定。否則，執行隱含的回轉。
- 程式重新存取尚未確定的已變更記錄。這可讓程式執行下列項目：
  - 新增記錄並在指定確定作業之前更新它。
  - 在指定確定作業之前更新相同記錄兩次。
  - 新增記錄並在指定確定作業之前刪除它。
  - 使用不同的邏輯檔案 (在確定控制下) 重新存取未確定的記錄。
- 您對 STRCMTCTL 指令指定 LCKLVL (\*CHG 或 \*CS)，並以確定作業開啓唯讀檔案。在此情況下，不會對要求產生鎖定。這視為確定控制無效，但檔案有出現在確定控制下檔案的 WRKJOB 功能表選項中。
- 您發出 STRCMTCTL 指令但沒有在確定控制下開啓任何檔案。在此狀況下，對檔案所做的任何記錄層次變更不在確定控制下執行。

## 確定控制期間監督的錯誤訊息

根據訊息類型及發生錯誤的時間，確定或回轉作業可傳回若干不同的錯誤訊息或傳送到工作日誌。

在下列期間會出現這些訊息：

- 一般確定或回轉處理程序
- 工作處理程序結束時的確定或回轉處理程序
- 啓動群組結束時的確定或回轉處理程序

您無法在啓動群組結束或工作處理程序結束期間監督下列任何訊息。而且，您只能監督 CPFxxxx 訊息。CPDxxxx 訊息一律以診斷訊息傳送，無法被監督。在啓動群組結束期間結束「啓動群組層次」確定定義時，或在工作結束期間結束任何確定定義時所發生的任何錯誤，會保留在工作日誌中作為診斷訊息。

與所要尋找的確定控制相關的錯誤訊息如下所示：

**CPD8351**

可能尚未確定變更。

**CPD8352**

未在遠端位置 &3 上確定變更。

**CPD8353**

可能尚未確定關聯式資料庫 &1 的變更。

**CPD8354**

可能尚未確定 DDM 檔案 &1 的變更。

**CPD8355**

可能尚未確定 DDL 物件 &1 的變更。

**CPD8356**

可能已確定回轉變更。

**CPD8358**

可能尚未回轉關聯式資料庫 &1 的變更。

**CPD8359**

可能尚未回轉 DDM 檔案 &1 的變更。

**CPD835A**

可能尚未回轉 DDL 物件 &3 的變更。

**CPD835C**

未更新 &2 中的通知物件 &1。

**CPD835D**

DRDA 資源不允許保留 SQL 游標。

**CPF835F**

確定或回轉作業失敗。

**CPD8360**

成員或檔案 (或兩者) 已被解除配置。

**CPD8361**

API 跳出程式 &1 在確定期間失敗。

**CPD8362**

API 跳出程式 &1 在回轉期間失敗。

**CPD8363**

API 跳出程式 &1 在確定期間 &4 分鐘後結束。

**CPD8364**

API 跳出程式 &1 在回轉期間 &4 分鐘後結束。

**CPD836F**

確定控制作業期間發生通信協定錯誤。

**CPD83D1**

API 資源 &4 不能為最後代理程式。

**CPD83D2**

資源與確定控制不相容。

**CPD83D7**

確定作業變更為回轉。

**CPD83D9**

發生試探混合狀況。

**CPF83DB**

確定作業導致回轉。

**CPD83DC**

「發生問題時使用的動作」決定確定或回轉作業；理由 &2。

**CPD83DD**

交談結束；理由 &1。

**CPD83DE**

傳回的資訊無效。

**CPD83EC**

API 跳出程式 &1 已表決回轉。

**CPD83EF**

對下一個邏輯工作單元啟動了回轉作業。

**CPF8350**

找不到確定定義。

**CPF8355**

不接受 ENDCMTCTL。擱置中變更作用中。

**CPF8356**

確定控制結束，但未確定 &1 本端變更。

**CPF8358**

未更新 &2 中的通知物件 &1。

**CPF8359**

回轉作業失敗。

**CPF835A**

已取消結束確定定義 &1。

**CPF835B**

結束確定控制時發生錯誤。

**CPF835C**

確定控制結束，但未確定遠端變更。

**CPF8363**

確定作業失敗。

**CPF8364**

確定控制參數值無效。原因碼 &3。

**CPF8367**

無法執行確定控制作業。

**CPF8369**

無法將 API 確定資源置於確定控制下；原因碼 &1。

**CPF83D0**

不接受確定作業。

**CPF83D2**

確定完成 == 已傳回「重新同步進行中」。

**CPF83D3**

確定完成 == 已傳回「試探混合」。

**CPF83D4**

未傳送邏輯工作單元異動日誌登錄。

**CPF83E1**

確定作業因限制違規而失敗。

**CPF83E2**

需要回轉作業。

**CPF83E3**

所要求的巢狀層次非作用中。

**CPF83E4**

確定控制結束，但未確定資源。

**CPF83E6**

確定控制作業完成，且重新同步進行中。

**CPF83E7**

不接受 X/Open 整體異動的確定或回轉。

## 監督 CALL 指令後的錯誤

當呼叫使用確定控制的程式時，您應該監督異常錯誤，而如果有錯誤發生，則執行回轉作業。例如，當程式發現異常錯誤 (如 RPG 除以 0 的錯誤) 時，會出現未確定的記錄。根據工作的查詢訊息回答 (INQMSGRPY) 參數狀態，程式會傳送查詢訊息或執行預設動作。如果操作員回應或預設動作結束了程式，則不確定記錄仍等待確定或回轉作業。

若呼叫另一個程式而產生確定作業，則上一個程式部份已完成的異動會被確定。

若要避免部份已完成的異動被確定，請監督 CALL 指令後的跳出訊息。例如，如果是 RPG 程式，則使用下列程式碼：

```
CALL RPGA
MONMSG MSGID(RPG9001)
EXEC(ROLLBACK) /*Rollback if pgm is canceled*/
```

如果是 COBOL 程式，則使用下列程式碼：

```
CALL COBOLA
MONMSG MSGID(CBE9001)
EXEC(ROLLBACK) /*Rollback if pgm is canceled*/
```

### 一般確定或回轉處理程序的失敗

在確定或回轉處理程序期間任何時候都可能發生錯誤。下表將這個處理程序分成四個狀況。中間的直欄說明系統在每個狀況中發現錯誤時所採取的動作。第三個直欄建議您或您的應用程式回應訊息時應採取什麼動作。這些建議與系統處理確定控制處理程序的方法一致。

狀況	確定或回轉處理程序	建議動作
「記錄層次」 I/O 確定失敗	<ul style="list-style-type: none"> <li>如果在準備階段期間發生錯誤，則會回轉異動並送出訊息 F83DB。</li> <li>如果在已確定階段期間發生錯誤，則確定處理程序繼續儘量確定剩餘資源。在確定處理程序結束時送出訊息 CPF8363。</li> </ul>	監督訊息；如預期處理
API 確定資源的「物件層次」或確定和回轉跳出程式在確定期間失敗	<ul style="list-style-type: none"> <li>如果在準備階段期間發生錯誤，則會回轉異動並送出訊息 F83DB。</li> <li>如果在已確定階段期間發生錯誤，則處理程序繼續儘量確定或回轉剩餘資源。根據確定資源類型，傳回下列其中一則訊息： <ul style="list-style-type: none"> <li>– CPD8353</li> <li>– CPD8354</li> <li>– CPD8355</li> <li>– CPD8361</li> </ul> </li> </ul> <p>在確定處理程序結束時送出訊息 CPF8363。</p>	監督訊息；如預期處理
「記錄層次」 I/O 回轉失敗	<ol style="list-style-type: none"> <li>傳回 CPD8356</li> <li>嘗試繼續處理回轉物件層次或 API 確定資源</li> <li>處理程序結束時傳回 CPF8359</li> </ol>	監督訊息；如預期處理
API 確定資源的「物件層次」或確定和回轉跳出程式在回轉期間失敗	<ol style="list-style-type: none"> <li>根據確定資源類型，傳回下列其中一則訊息： <ul style="list-style-type: none"> <li>• CPD8358</li> <li>• CPD8359</li> <li>• CPD835A</li> <li>• CPD8362</li> </ul> </li> <li>繼續處理</li> <li>處理程序結束時傳回 CPF8359</li> </ol>	監督訊息；如預期處理

### 工作結束期間的確定或回轉處理程序

上表描述的所有狀況也套用到工作結束時，但若傳送下列其中一則訊息則不適用：

- CPF8356 (如果僅登記本端資源)
- CPF835C (如果僅登記遠端資源)

- CPF83E4 (如果同時登記本端和遠端資源)

此外，如果呼叫了 API 可確定資源的確定及回轉跳出程式，則可能出現其中一個訊息來特別指示工作完成。如果確定及回轉跳出程式沒有在 5 分鐘內完成，則程式會被取消、送出診斷訊息 CPD8363 (針對確定) 或 CPD8364 (針對回轉)，然後繼續剩餘的確定或回轉處理程序。

### 起始程式載入 (IPL) 期間的確定或回轉處理程序

上表描述的所有狀況也套用到確定定義的起始程式載入 (IPL) 回復期間，不包括傳送訊息 CPF835F 而不是訊息 CPF8359 或 CPF8363。對特定確定定義送出的訊息可能在其中一個 QDBSRVxx 工作的工作日誌或 QHST 日誌中出現。在 QHST 日誌中，訊息 CPI8356 指示特定確定定義的起始程式載入 (IPL) 回復開始。訊息 CPC8351 指示特定確定定義的起始程式載入 (IPL) 回復結束，您可以在這兩訊息間找到有關確定定義回復的其他任何訊息。

如果呼叫了 API 可確定資源的確定及回轉跳出程式，則可能出現其中一個訊息來特別指示確定定義。如果確定及回轉跳出程式沒有在 5 分鐘內完成，則程式會被取消、送出診斷訊息 CPD8363 (針對確定) 或 CPD8364 (針對回轉)，然後繼續剩餘的確定或回轉處理程序。

### 偵測死結

當工作對某物件 (物件 A) 保留鎖定並等待取得另一物件 (物件 B) 的鎖定時，可能會發生死結狀況。如果同一時間，另一個工作或異動目前保留物件 B 的鎖定並等待取得物件 A 的鎖定。

請執行下列步驟以瞭解是否發生死結狀況並進行修正：

1. 尋找作用中工作清單中擱置的工作。查看工作狀態來決定工作是否擱置。
2. 尋找工作等待鎖定的物件。有關具有「異動範圍」鎖定的異動，請參閱顯示異動的已鎖定物件，以取得所需的步驟。有關具有工作範圍鎖定的異動，請參閱明細：作用中工作內容。
3. 對於工作等待鎖定的所有物件，查看鎖定持有者 (異動或工作) 清單並嘗試找出對應到擱置工作所要求層次的衝突鎖定。
4. 如果異動保留衝突的鎖定，顯示與此異動相關的工作，請查看其中之一是否在等待鎖定。
5. 決定此等待工作是否嘗試鎖定被起始擱置工作鎖定的其中一個物件。當您尋找嘗試鎖定被起始擱置工作鎖定的物件之工作時，您可以將有問題的物件視為易失敗處。
6. 請研究此異動以決定適當做法。
  - a. 尋找異動內容找出起始它的應用程式，然後尋找應用程式碼。
  - b. 或追蹤異動的動作至此位置，方法是在異動內容中尋找確定循環 ID，然後搜尋異動日誌中含有此 ID 的登錄。若要這麼做，您可以使用擷取異動日誌登錄 (RTVJRNE) 指令並指定 CMTCYCID 參數。
  - c. 在取得相關資訊後，使用者可選擇強制回轉或確定作業。

### 通信失敗後回復異動

萬一通信失敗時，系統通常會自動對任何遠端系統完成重新同步。不過，如果失敗情形嚴重，以致於無法再與遠端系統重建通信 (比方說，通信線路被切斷)，您必須親自取消重新同步和復置異動。異動也可能擁有需要釋放的鎖定。

1. 在「iSeries 領航員」中，顯示確定控制資訊來說明您正在使用的異動。
2. 找出正在嘗試與遠端系統重新同步化的異動。此異動的**重新同步進行中**欄位會設定為是。
3. 檢查個別異動的資源「狀態」來尋找與遠端系統有連線的異動。
4. 在找出異動之後，根據異動的狀態，您可能必須強制確定或強制回轉。
5. 在研究異動內容之後，您可以決定確定或回轉。

- 您可以使用**工作單元 ID** 在其他系統上尋找異動的其他部份。
  - 您也可以從異動的狀態來決定確定或回轉。例如，如果在通知失敗期間，資料庫異動正在執行兩階段確定，且失敗之後，狀態是「備妥」或「最後代理程式擱置中」，則您可選擇強制確定異動。
6. 在強制確定或回轉未完成的異動之後，請停止對識別的異動重新同步失敗的連線。

關於回復的詳細資訊，請參閱何時強制確定和回轉，何時取消重新同步。

## 何時強制確定和回轉，以及何時取消重新同步

強制確定、回轉或取消重新同步的決策稱為**試探法**。試探法就是您強制系統確定或回轉異動所採取的動作。當您執行試探法時，如果您的決策與異動中其他位置的結果不一致，則異動狀態變成試探混合。您必須負責決定參與異動的其他所有位置採取的動作並對資料庫記錄進行重新同步。

在執行試探法之前，請盡量收集有關異動的資訊。顯示與確定定義相關的工作並建立相關的異動日誌和檔案之記錄。稍後如果您需要顯示異動日誌登錄並手動套用或移除登載的變更，您可以使用此資訊。

找出異動相關資訊的最佳位置就是尋找該異動的起始程式所在的位置。不過，可能 API 資源或最後代理程式擁有確定或回轉的決定權。

如果 API 資源登記為最後代理程式資源，則 API 資源擁有確定或回轉的最終決定權。您必須參考應用程式的相關資訊及其使用 API 資源的方式，以決定要確定或回轉。

如果異動已選取最後代理程式，則該最後代理程式擁有確定或回轉的決定權。有關異動的資訊，請參考最後代理程式的狀態。

當您因為系統或通信失敗無法修復而必須執行試探法或取消重新同步時，您可以使用下列步驟來尋找所有不確定的異動：

1. 在「iSeries 領航員」中，展開您要使用的系統。
2. 展開**資料庫**和系統的本端資料庫。
3. 展開**異動**。
4. 展開**資料庫異動**或**整體異動**

在這個顯示畫面中，您可以看到每個異動的確定義、重新同步狀態、現行工作單元 ID 以及現行工作單元狀態。尋找具下列項目的異動：

- 具有**邏輯工作單元狀態**為「已備妥」或「最後代理程式擱置中」的異動。
- 顯示**重新同步進行中**狀態為「是」的異動。

若要使用此系統上參與異動的工作，請以滑鼠右鍵按一下異動並選取**工作**。

當您以滑鼠右鍵按一下異動時，您也可以選取**強制確定**、**強制回轉**或**取消重新同步**。

在執行試探法或取消重新同步之前，您可能要檢查其他系統上與異動相關的工作狀態。檢查遠端系統的工作可協助您避免造成系統間資料庫不一致的決策。

1. 以滑鼠右鍵按一下您要使用的異動。
2. 選取**資源狀態**
3. 在「資源狀態」對話框中，選取 SNA 連線的**交談標籤**；選取 TCP/IP 連線的**連線**。

每一個交談資源代表參與異動的一個遠端系統。在遠端系統上，您可以使用「iSeries 領航員」來查看與此異動相關的異動。

工作單元 ID 的基本部份在所有系統上是相同的。當您在遠端系統上顯示確定控制資訊時，請注意工作單元 ID 的基本部份在本端系統上是相同的。



例如，如果本端系統上的工作單元 ID 以 APPN.RCHASL97.X'112233445566 為開頭，請注意，遠端系統上的工作單元 ID 也是以 APPN.RCHASL97.X'112233445566 為開頭。

---





## 確定控制的相關資訊

以下是有關「確定控制」主題的 iSeries 手冊和 IBM 紅皮書<sup>(TM)</sup> (PDF 格式)、網站及「資訊中心」主題。您可以檢視或列印這些 PDF。

### 手冊

- COBOL/400 User's Guide  (約 425 頁)
- RPG/400 User's Guide  (約 580 頁)

### 紅皮書

- Connecting WebSphere to DB2 UDB Server  (約 458 頁)
- Advanced Functions and Administration for DB2 Universal Database for iSeries  (約 372 頁)
- Stored Procedures and Triggers on DB2 Universal Database for iSeries  (約 424 頁)
- Striving for Optimal Journal Performance on DB2 Universal Database for iSeries  (約 184 頁)

### 網站


The Open Group ([www.opengroup.org](http://www.opengroup.org)) 

### 其它資訊

- DB2 Universal Database for iSeries 資料庫 程式設計
- DB2 Universal Database for iSeries SQL 程式設計概念
- XA API
- 日誌管理

如欲在您的工作站上儲存 PDF，供檢視或列印之用：

1. 在您的瀏覽器中以滑鼠右鍵按一下 PDF (以滑鼠右鍵按一下上述的鏈結)。
2. 按一下**另存目標...**
3. 導覽到您想要儲存 PDF 之處。
4. 按一下**儲存**。

如果您需要 Adobe Acrobat Reader 來檢視或列印這些 PDF，您可以從 Adobe 網站 ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  下載之。





**IBM**