



IBM 系統 - iSeries

**系統管理
叢集**

版本 5 版次 4





IBM 系統 - iSeries

**系統管理
叢集**

版本 5 版次 4

請注意

使用此資訊及其支援的產品之前，請先閱讀第 143 頁的『注意事項』中的資訊。

第七版 (2006 年 2 月)

此版本適用於 IBM i5/OS (產品編號 5722-SS1) 版本 5 版次 4 修正層次 0，以及所有後續的版次和修訂版 (除非新版中另有指示)。此版本並非適用於所有的精簡指令集電腦 (RISC) 機型和 CISC 機型。

© Copyright International Business Machines Corporation 1998, 2006. All rights reserved.

目錄

叢集	1		建立叢集管理網域	103
V5R4 新增功能	1		新增受監視的資源項目	103
可列印的 PDF	2		監視叢集管理網域	104
叢集概念	2		監督叢集狀態	105
叢集的好處	3		叢集效能	106
叢集如何運作	3		結束叢集工作	107
叢集基礎	4		資源監視與控制 (RMC)	107
叢集的元素	6		工作結構和使用者的行列	108
叢集事件	16		維護所有節點上的使用者設定檔	109
叢集應用程式	26		叢集的備份及回復服務程式	110
叢集規劃	70		儲存叢集配置	110
配置與管理叢集的解決方案	70		範例：叢集配置	111
叢集的基本要求	77		範例：簡式雙節點叢集	111
設計叢集	78		範例：四節點叢集	112
叢集安全性	85		範例：使用獨立磁碟儲存區的切換式磁碟叢集	113
叢集配置核對清單	87		範例：管理對等節點資源的叢集管理網域	113
INETD 伺服器	89		範例：具有地理鏡映的獨立磁碟儲存區	115
可調整的叢集通訊參數	90		叢集的疑難排解	116
叢集解除配置核對清單	91		判斷叢集問題是否存在	116
叢集管理網域規劃	92		收集叢集的回復資訊	117
配置叢集	92		使用「傾出叢集追蹤 (DMPCLUTRC)」指令來調 查問題	117
建立叢集	93		使用 CLUSTERINFO 巨集來調查問題	121
管理叢集	94		一般叢集問題	127
將節點新增到叢集中	95		分割區錯誤	129
啟動叢集節點	96		叢集回復	133
結束叢集節點	96		「iSeries 領航員」叢集管理常見問題集	135
調整叢集的叢集版本	97		叢集支援的客服電話	141
刪除叢集	98		叢集的相關資訊	141
建立 CRG	98			
啟動 CRG	99			
變更叢集資源群組 (CRG) 的回復網域	99			
執行切換	100			
將節點新增到裝置網域	101			
從裝置網域中移除節點	102			
系統事件對叢集的影響	103			
			附錄. 注意事項	143
			程式設計介面資訊	144
			商標	144
			條款	145

叢集

叢集可讓您有效地將 iSeries 伺服器組合在一起，以設定一個環境，為重要的應用程式、裝置及資料提供近乎 100% 的可用性。

叢集也提供簡化的系統管理及增強的可調整性，在您的企業成長時，可以無縫地增加新元件。

使用程式碼範例，即表示您同意程式碼授權及免責聲明資訊的條款。

V5R4 新增功能

查看本版次的新增功能。

叢集管理網域支援

叢集管理網域可監視及同步化叢集內所選資源的變更。叢集管理網域可以更輕鬆地管理及同步化叢集內共用資源的屬性 (如環境變數或使用者設定檔)。如需叢集管理網域的詳細資訊，請參閱下列主題：

- 第 8 頁的『叢集管理網域』
- 第 92 頁的『叢集管理網域規劃』
- 第 92 頁的『叢集管理網域核對清單』
- 第 103 頁的『建立叢集管理網域』

對等節點叢集資源群組 (CRG) 支援

已增強所有的 CRG 介面，以支援對等節點叢集資源群組 (CRG)。對等節點叢集資源群組 (CRG) 是非切換式 CRG，其中回復網域中每個節點的角色，與在對等節點 CRG 之關聯資源的回復中的角色相同。如需詳細資訊，請參閱下列主題：

- 叢集資源群組
- 第 98 頁的『建立 CRG』
- 第 99 頁的『啟動 CRG』



叢集加強功能

已實作數個加強功能，來改進叢集環境內的關閉電源作業及解決問題的方式。這些改進包括：

- 採用系統的方式，當所有的作用中子系統結束，或系統結束或關閉電源時，結束叢集節點上的形成叢集作業。如需詳細資訊，請參閱第 103 頁的『系統事件對叢集的影響』。
- 能夠以作用中接管 IP 位址來配置新的應用程式 CRG。如需詳細資訊，請參閱第 99 頁的『以作用中接管 IP 位址來建立應用程式 CRG』。
- 能夠從作用中節點檢視整個叢集以及與它相關聯的 CRG，來疑難排解叢集問題。如需詳細資訊，請參閱第 117 頁的『收集叢集的回復資訊』。
- 新增除錯工具及其產生的結果之相關資訊。您可以使用這些工具及其結果來判定叢集內的問題之解決方式。如需詳細資訊，請參閱下列主題：
 - 第 117 頁的『使用「傾出叢集追蹤 (DMPCLUTRC)」指令來調查問題』
 - 第 121 頁的『使用 CLUSTERINFO 巨集來調查問題』

查看新增功能或變更項目的方式

為了協助您易於查閱技術變更之處，本資訊內容使用：

-  圖示，標示新增或變更資訊開始的位置。
-  圖示，標示新增或變更資訊結束的位置。




如需本版次新增功能或變更項目的其他資訊，請參閱使用者備忘錄。

可列印的 PDF

您可以檢視及列印本資訊的 PDF 格式。

若要檢視或下載本文件的 PDF 版本，請選取「叢集」（約 938 KB）。

Redbooks™

- Clustering and IASPs for Higher Availability  (約 6.4 MB) 此紅皮書顯示 iSeries™ 伺服器可用之叢集與可切換磁碟技術的概觀。
- iSeries Independent ASPs: A Guide to Moving Applications to IASPs  (約 3.4 MB) 此紅皮書以逐步方式呈現 iSeries 伺服器上的獨立 ASP。
- Roadmap to Availability on the iSeries 400®  (約 626 KB) 此紅皮書以逐步方式呈現 iSeries 伺服器上的獨立 ASP。

網站

- High Availability and Clusters  (www.ibm.com/servers/eserver/series/ha)


IBM® 網站，可取得「高可用性與叢集」的相關資訊

儲存 PDF 檔

若要在工作站上儲存 PDF 以供檢視或列印：

1. 以滑鼠右鍵按一下瀏覽器內的 PDF (以滑鼠右鍵按一下上述的鏈結)。
2. 如果是使用 Internet Explorer，請按一下**另存目標**。如果是使用 Netscape Communicator，請按一下**另存鏈結為**。
3. 瀏覽至要儲存 PDF 的目錄。
4. 按一下**儲存**。

下載 Adobe Acrobat Reader

您需要 Adobe Acrobat Reader 才可檢視或列印這些 PDF。您可以從 Adobe 網站 (www.adobe.com/products/acrobat/readstep.html)  下載副本。

叢集概念

這可讓您全盤地瞭解叢集如何運作。您可從中得知叢集的優點和它們對您的重要性，以及重要的叢集概念及它們如何整合在一起。

iSeries 叢集是作為單一系統一起運作之一或多個系統或邏輯分割區的集合或群組。叢集中的系統（稱為叢集節點）協力運作，以提供單一運算解決方案。在叢集中，iSeries 叢集最多支援 128 個節點。這可讓您將 iSeries 系統有效地組合在一起，以設定一個環境，為重要的應用程式及重要的資料提供近乎 100% 的可用性。這有助於確保您的重要系統及應用程式隨時都是可用的。叢集也提供簡化的系統管理及增強的可調整性，在您的企業成長時，可以無縫地增加新元件。

叢集的好處

若您的企業要求系統隨時可進行作業，叢集能提供解決方案。

- | 使用叢集，您便可以大幅減少非計劃性關機的次數與持續時間，以及計劃性關機的持續時間，從而確保您的系統、資料及應用程式持續可用。

叢集可為您的企業提供的主要好處有：

持續可用性

叢集可確保您的系統、資料及應用程式一直是持續可用的。

簡化的管理

- | 您可以用單一系統或單一資料庫的方式來管理系統群組，而不需要登入個別的系統。您可以使用叢集管理網域更輕鬆地管理在叢集內共用的資源。

增強的可調整性

當您的企業成長需要時，無障礙地新增新元件而無影響。

相關概念

第 16 頁的『錯誤後移轉』

當叢集中的伺服器在系統失效時自動切換為一或多個備份伺服器時，會發生錯誤後移轉。

相關工作

第 18 頁的『切換』

當您將資源存取權從某個伺服器手動切換至另一個伺服器時，會發生切換。

叢集如何運作

- | 叢集基礎架構是當作 i5/OS™ 的一部分（稱為叢集資源服務）來提供的，它可提供重要資源的回復。這些資源可包括資料、應用程式、裝置，以及由多個用戶端存取的其他資源。
- | 若發生系統故障或網站流失，則在叢集內的系統上所提供的功能，可透過已在叢集中所定義的其他系統來存取。可在兩種模型中存取此資料：主要備份模型與對等節點模型。如需根據這些模型可建立之叢集資源群組 (CRG) 的詳細資訊，請參閱「叢集資源群組」。

相關概念

第 16 頁的『錯誤後移轉』

當叢集中的伺服器在系統失效時自動切換為一或多個備份伺服器時，會發生錯誤後移轉。

第 23 頁的『抄寫』

抄寫可即時建立某些項目的副本。它表示將物件從叢集中的一個節點複製到叢集中的一或多個其他節點。

第 14 頁的『回復裝置』

回復裝置是實體資源，由配置物件（如裝置說明）來表示，可從叢集中一個以上的節點中存取。

第 14 頁的『回復資料』

回復資料是叢集中於一個以上的節點上的抄寫（複製）資料。

第 19 頁的『重新結合』

重新結合表示已成為未參加的成員之後，再成為叢集的作用中成員。

第 82 頁的『邏輯抄寫、切換式磁碟與跨站台鏡映的比較』

本主題提供不同資料回復技術的概觀，這些技術可與叢集搭配使用以增強高可用性。

相關工作

第 18 頁的『切換』

當您將資源存取權從某個伺服器手動切換至另一個伺服器時，會發生*切換*。

叢集基礎

在您開始設計及自訂滿足您需求的叢集之前，需要了解形成叢集的基本概念。

- 與叢集相關的基本概念有兩個：叢集節點及叢集資源群組。叢集節點是 iSeries 系統或邏輯分割區，它是叢集的成員。當您建立叢集時，指定您要作為節點併入叢集的系統或邏輯分割區。叢集資源群組 (CRG) 作為回復資源集合的控制物件。CRG 可包含叢集中的節點子集或所有節點。iSeries 叢集支援四種類型的 CRG：應用程式、資料、裝置及對等節點。在這些類型的 CRG 中有兩個通用元素：回復網域及跳出程式。

*回復網域*定義 CRG 中每個節點的角色。當您在叢集中建立 CRG 時，CRG 物件會建立在指定併入回復網域的所有節點上。但是，會提供 CRG 物件之單一系統影像 (您可從 CRG 的回復網域中之任何作用中節點存取)。也就是說，對 CRG 所作的任何變更，將在回復網域中之所有節點上進行。

在 CRG 的叢集相關事件期間會呼叫*跳出程式*。此類事件之一是將存取點從某個節點移至另一個節點。

- 在叢集中可以建立的 CRG 模型有兩種：主要備份模型及對等節點模型。在主要備份模型中，可定義 CRG 之回復網域中的節點，如下所示：

- 主要節點是回復叢集資源之主要存取點的叢集節點。
- 備份節點是在目前主要節點失敗或已起始手動切換時，接管主要存取角色的叢集節點。
- 複製節點是具有叢集資源副本，但無法擔任主要或備份節點角色的叢集節點。

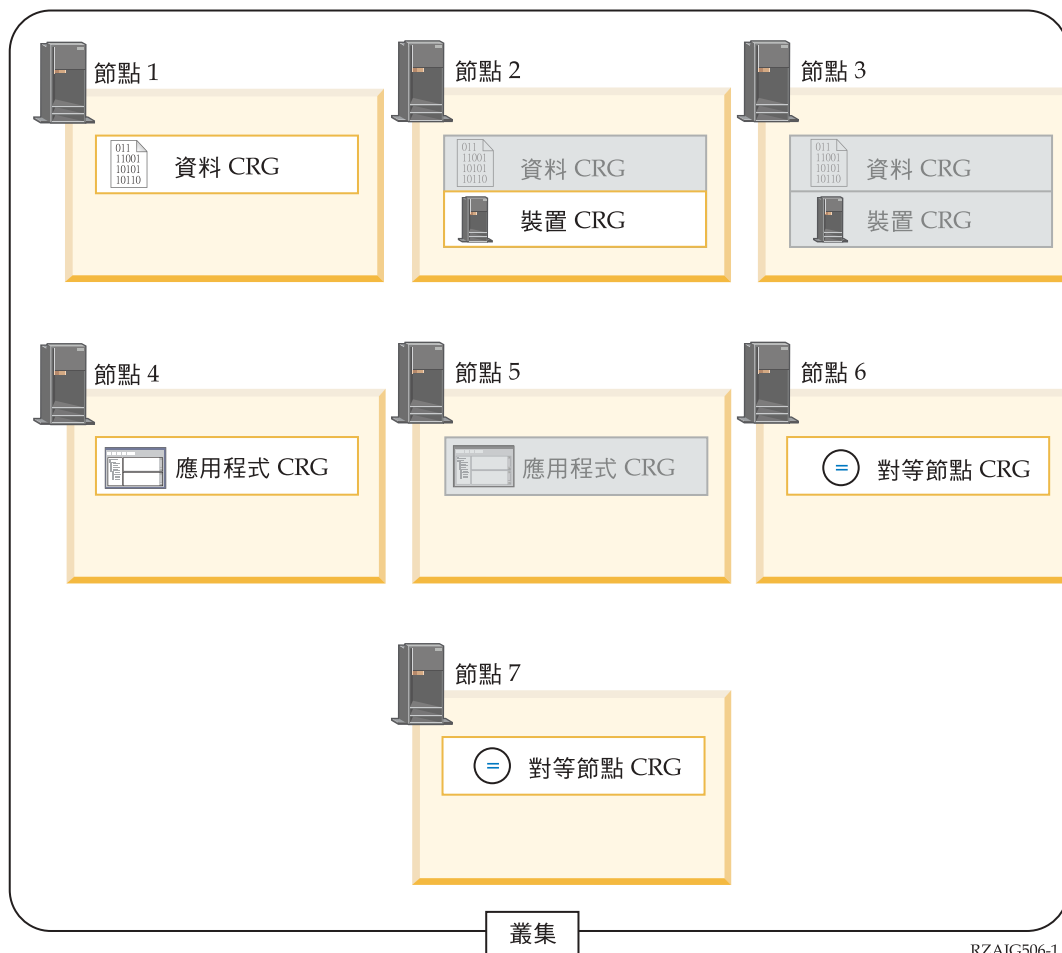
- 在對等節點模型中，對等節點 CRG 的回復網域定義節點間的對等關係。可定義對等節點 CRG 之回復網域中的節點，如下所示：

- 對等節點是可以作為叢集資源之作用中存取點的叢集節點。
- 複製節點是具有叢集資源副本的叢集節點。在對等節點 CRG 中定義為複製節點的節點代表叢集資源的非作用中存取點。

在對等節點 CRG 中，回復網域中的各節點在進行回復時所擔任的角色都是相等的。因為此對等節點 CRG 中的每個節點在本質上都具有相同的角色，因此不適用錯誤後移轉及切換的概念。這些節點具有對等關係，因此若其中一個節點失敗，其他對等節點將繼續運作。

- 您還可以建立由對等節點 CRG 代表的叢集管理網域。叢集管理網域中的節點在 CRG 回復網域中均為對等節點。不存在複製節點。

在以下範例中，將顯示每種類型的一個 CRG：



RZAIG506-1

資料 CRG

資料 CRG 是在「節點 1」、「節點 2」及「節點 3」上。這表示資料 CRG 的回復網域，已為「節點 1」(主要節點)、「節點 2」(第一個備份節點)及「節點 3」(第二個備份節點)指定角色。在此範例中，「節點 1」目前作為主要存取點。「節點 2」定義為回復網域中的第一個備份節點。這表示「節點 2」包含了透過邏輯抄寫而保持為最新的資源副本。萬一發生錯誤後移轉或切換，「節點 2」就會變成主要存取點。

應用程式 CRG

應用程式 CRG 是在「節點 4」及「節點 5」上。這表示應用程式 CRG 的回復網域已指定「節點 4」及「節點 5」。在此範例中，「節點 4」目前用作主要存取點。萬一發生錯誤後移轉或切換，「節點 5」就會變成應用程式的主要存取點。需要接管 IP 位址。

對等節點 CRG

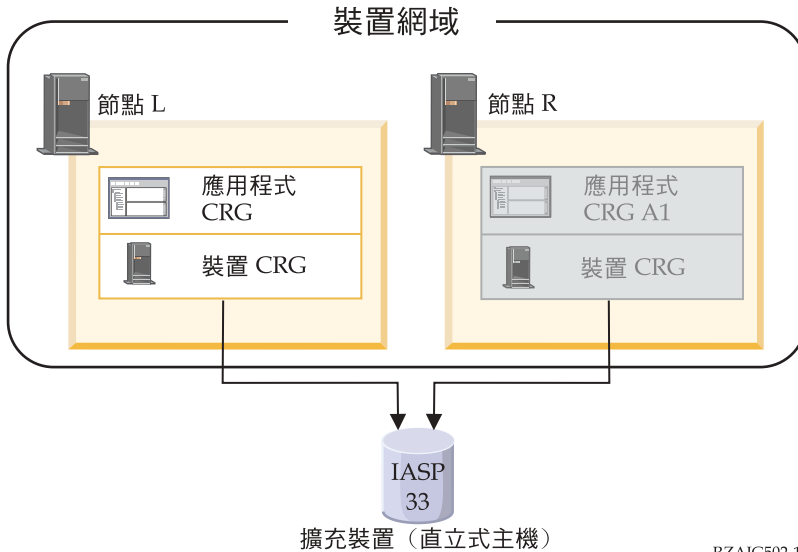
對等節點 CRG 是在「節點 6」及「節點 7」上。這表示對等節點 CRG 的回復網域已指定「節點 6」及「節點 7」。在此範例中，節點 6 及節點 7 可以是對等節點或複製節點。如果這是由對等節點 CRG 代表的叢集管理網域，則叢集管理網域所監視的資源將會使變更在節點 6 及節點 7 代表的網域間同步(無論變更最初是發生在哪一個節點上)。

裝置 CRG

裝置 CRG 是在「節點 2」及「節點 3」上。這表示裝置 CRG 的回復網域已指定「節點 2」及「節點 3」。在此範例中，「節點 2」目前用作主要存取點。這表示裝置 CRG 擁有的回復裝置目前可從「節點 2」存取。萬一發生錯誤後移轉或切換，「節點 3」就會變成裝置的主要存取點。

裝置 CRG 需要稱為獨立磁碟儲存區 (也稱為獨立輔助儲存體儲存區或獨立 ASP) 的回復裝置，以配置在外部裝置、擴充裝置 (直立式主機)、或邏輯分割區的 IOP 中。

裝置 CRG 的回復網域中之節點必須也是相同裝置網域的成員。以下範例說明有「節點 L」及「節點 R」之裝置 CRG 在其回復網域中。這兩個節點也是相同裝置網域的成員。



RZAIG502-1

相關概念

『叢集節點』

叢集節點是 iSeries 系統或邏輯分割區，它是叢集的成員。

第 7 頁的『叢集資源群組』

- | 叢集資源群組 (CRG) 是 iSeries 系統物件，它是一組或一套叢集資源，可用來管理在叢集環境中發生的事件。
- | 叢集資源群組說明回復網域，並提供某些叢集事件發生時所呼叫之叢集資源群組跳出程式的名稱。

第 10 頁的『回復網域』

- | 回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

第 10 頁的『叢集資源群組跳出程式』

發生 CRG 的叢集相關事件之後，會呼叫叢集資源群組跳出程式。

獨立磁碟儲存區

第 15 頁的『裝置網域』

裝置網域是共用裝置資源的叢集中之節點的子集。更特別的是，裝置網域中的節點可參與回復裝置資源之部分集合的切換動作。

叢集的元素

- | iSeries 叢集是作為單一系統一起運作之一或多個系統或分割區的集合。使用此資訊可了解元素及其彼此之間的關係。

叢集節點

叢集節點是 iSeries 系統或邏輯分割區，它是叢集的成員。

每一個叢集節點都是由八個字元的叢集節點名稱來識別，此名稱與代表 iSeries 系統的一或多個 IP 位址相關聯。當配置叢集時，對於叢集中的節點，您可以使用您想要的任何名稱。但是，建議節點名稱與主電腦名稱或系統名稱相同。

叢集通訊使用 TCP/IP 通訊協定套件，提供叢集中每一個節點上的叢集服務之間的通訊路徑。配置為叢集之一部分的叢集節點集被視為叢集成員清單。

叢集資源群組

- | 叢集資源群組 (CRG) 是 i5/OS 系統物件，它是一組或一套叢集資源，可用來管理在叢集環境中發生的事件。
- | 叢集資源群組說明回復網域，並提供某些叢集事件發生時所呼叫之叢集資源群組跳出程式的名稱。
- | 叢集提供兩種選擇來定義叢集內之節點間的關係：主要備份模型與對等節點模型。這兩種模型可根據環境的需要一起使用或單獨使用。

主要備份模型

- | 此種類的所有叢集資源群組都以特定角色來定義回復網域中的節點：主要、備份或複製。主要及備份節點是叢集資源可用的存取點。然而，在指定的時間點，只有一個節點是作用中的存取點。這便是主要節點。複製節點不能用來當作存取點。這可藉由為複製節點指派備份角色來改變。主要備份模型叢集資源群組定義為資料回復、應用程式回復或裝置回復。資料回復可讓多份資料副本維持在叢集的一個以上之節點上，及讓存取點變更為備份節點。應用程式回復可讓應用程式在叢集的相同節點或不同節點上重新啟動。裝置回復可讓裝置資源移動 (切換) 到備份節點。

每一個資料及應用程式叢集資源群組都有一個與其相關的叢集資源群組跳出程式。對於回復裝置叢集資源群組而言，跳出程式是可選用的。

在「iSeries 領航員」中，叢集資源群組是視為不同的。

- | • 裝置 CRG 被視為**切換式裝置**。
- | • 應用程式 CRG 被視為**切換式應用程式**。
- | • 資料 CRG 被視為**可切換的資料群組**。

對等節點模型

- | 此種類的所有叢集資源群組都以對等節點或複製節點的角色來定義回復網域中的節點。對等節點可用作叢集資源群組的存取點。啟動叢集資源群組時，定義為對等節點的所有節點都將是存取點。複製節點不能用來當作存取點。這可藉由為複製節點指派對等節點角色來改變。在對等節點 CRG 內，每個節點都包含每個節點上存在的抄寫資料。當對等節點 CRG 中的節點失敗時，會將失敗點傳送至叢集中的其他節點，那些節點便會從失敗點開始繼續作業。

叢集管理網域由具有僅包含對等節點之回復網域的對等節點 CRG 來代表。

相關概念

第 10 頁的『回復網域』

- | 回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

第 10 頁的『叢集資源群組跳出程式』

發生 CRG 的叢集相關事件之後，會呼叫叢集資源群組跳出程式。

管理叢集資源群組的處理程序

- | 發生節點失敗時，會進行錯誤後移轉。當發生 CRG 排序時，最先進行錯誤後移轉的是所有裝置 CRG，然後是所有資料 CRG，最後是應用程式 CRG。若為對等節點 CRG，則沒有順序問題，但在發生失敗時，每個節點都會收到通知。
- | 檢查 CRG 的狀態，可驗證叢集資源群組已完成錯誤後移轉或切換。

您也可以使用封鎖以保留應用程式，直到可處理資料為止。當處理資料回復叢集資源群組時，您可能想要封鎖存取由資料 CRG 所代表的資料。您可以使用「封鎖 EDRS 存取 (QxdaBlockEDRS) API」及「檢查 EDRS 封鎖狀態 (QxdaCheckEDRSBlock) API」來封鎖存取。若發生錯誤後移轉或切換，您可以使用這些 API，從叢集資源群組跳出程式中，封鎖存取及對存取解除封鎖。

相關概念

第 16 頁的『錯誤後移轉』

當叢集中的伺服器在系統失效時自動切換為一或多個備份伺服器時，會發生錯誤後移轉。

第 10 頁的『回復網域』

- | 回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

第 10 頁的『叢集資源群組跳出程式』

發生 CRG 的叢集相關事件之後，會呼叫叢集資源群組跳出程式。

相關工作

第 18 頁的『切換』

當您將資源存取權從某個伺服器手動切換至另一個伺服器時，會發生切換。

叢集管理網域

- | 叢集管理網域可用來管理需要在叢集環境中的節點間保持一致的資源。

- | 在作為回復資料、應用程式及裝置之存取點的每個節點上，需要定義特定作業或配置參數。如果在可能是存取點的任何節點上變更其中一個參數，則需要將變更傳送至可能會成為存取點的所有其他節點上。叢集管理網域提供識別需要在網域節點間保持一致之資源的功能。然後它會監視這些資源的變更，並在作用中網域同步化這些變更。叢集管理網域由對等節點 CRG 來代表。建立叢集管理網域時，系統會建立對等節點 CRG。叢集管理網域的名稱會成為對等節點 CRG 的名稱。對等節點 CRG 的回復網域會定義組成叢集管理網域的節點。所有節點均為對等節點。叢集管理網域中不允許有複製節點。叢集節點只能定義在叢集內的一個叢集管理網域中。如需叢集管理網域作業相關的詳細資訊，請參閱下列主題：

- | 1. 第 92 頁的『叢集管理網域規劃』
- | 2. 第 92 頁的『叢集管理網域核對清單』
- | 3. 第 103 頁的『建立叢集管理網域』
- | 4. 第 103 頁的『新增受監視的資源項目』
- | 5. 第 99 頁的『啟動 CRG』

- | 一旦建立叢集管理網域，就會使用一般 CRG 功能來管理叢集管理網域。例如，如果您要新增節點至管理網域，則需要將節點新增至節點角色為對等節點之 CRG 的回復網域。若要啟動叢集管理網域，請啟動該對等節點 CRG。

- | 藉由啟動及結束 CRG，可以控制同步化變更的程序。結束 CRG 時，對網域中任何節點上之受監視資源所做的變更將不會傳送到網域的其他部分。一旦啟動 CRG，在任何受監視資源處於非作用中時對其所做的變更，都將傳送到網域的其他部分。當 CRG 在作用中時，會動態傳送對任何作用中節點上的任何受監視資源所做的變更，以在管理網域保持資源的一致性。如需詳細資訊，請參閱第 104 頁的『監視叢集管理網域』。

為了將節點新增至叢集管理網域，您需要將叢集節點新增至對等節點 CRG 的回復網域中。將節點新增至網域後，會在新節點上建立所有受管理的資源 (如果它們不存在的話)，並會與管理網域的其他部分同步化。

如果刪除叢集管理網域，則會從網域的每個節點上移除在該叢集管理網域中定義的所有資源，但不會從系統中移除實際資源。如需詳細資訊，請參閱受監視資源。

受監視資源

受監視資源是可由叢集管理網域來管理的系統資源類型。這些資源在叢集管理網域中是以受監視資源項目 (MRE) 來代表。

叢集管理網域同步化的資源是以受監視資源項目 (MRE) 來代表。一旦將 MRE 新增至叢集管理網域，對叢集管理網域中之任何節點上的資源所做的變更都將傳送至作用中網域中的所有節點。您可以使用三個「整合作業環境 API」來管理叢集管理網域中的 MRE：

- 新增受監視資源項目 (QfpadAddMonitoredResourceEntry) API
- 移除受監視資源項目 (QfpadRmvMonitoredResourceEntry) API
- 擷取受監視資源項目 (QfpadRtvMonitoredResourceInfo) API

可將受監視資源項目新增至下列資源類型的叢集管理網域：

- 系統值
- 使用者設定檔
- 工作說明
- 類別
- 獨立磁碟儲存區裝置說明
- 網路屬性
- 系統環境變數
- TCP/IP 屬性

僅當叢集管理網域中的所有節點都處於作用中且參與群組時，才能將 MRE 新增至該網域。如果叢集管理網域已分割，則無法新增 MRE。一旦新增 MRE，在啟動對等節點 CRG 時，對 MRE 所代表之資源所做的變更就會傳送至網域中的所有作用中節點。如果 CRG 已結束，則在重新啟動 CRG 時，會將擱置中的變更傳送至作用中網域。

有一種與 MRE 相關聯的廣域狀態。如果 MRE 所代表的資源，與在作用中網域內所有節點上監視之所有屬性的值相同，則該資源的廣域狀態是一致的。如果叢集管理網域試圖更新一或多個節點上的資源，但更新失敗，資源的廣域狀態便不一致。當廣域狀態不一致時，管理者必須判定失敗的原因並加以更正。當管理者因更正導致更新失敗之問題而變更資源，或當重新啟動 CRG 時，叢集管理網域將在下次更新時嘗試重新同步化資源。

當叢集管理網域 CRG 結束時，所有 MRE 的廣域狀態都會設為不一致。這是因為在 CRG 結束時，會變更不同節點上的受監視資源，因此它們變得一致。

如果 MRE 所代表的資源為系統物件，若要將此資源刪除、重新命名或移至不同的檔案庫，應當先移除 MRE。如果將資源刪除、重新命名或移至不同的檔案庫，MRE 的廣域狀態將不一致，並且此後對任何節點上的資源所做的任何變更都不會傳送至叢集管理網域。

將節點新增至叢集管理網域時，會將作用中網域的所有 MRE 複製到新的節點。由 MRE 來代表且不在新節點上的任何資源都將被建立，且屬性的值將設為與作用中叢集管理相同的值。

如果叢集管理網域中有非作用中的節點，當這些節點重新結合作用中網域時，會將在作用中網域中所做的任何資源變更傳送至非作用中節點。如果已分割叢集管理網域，則會繼續在每個分割區的作用中節點中將變更同步化。當節點重新合併在一起時，叢集管理網域將會傳送在每個分割區上所做的變更，使作用中網域內的資源保持一致。如果在不同分割區中對同一個資源進行多項變更，即使順序不確定，在合併分割區之後，叢集管理網域也會處理每個變更。

相關概念

第 8 頁的『叢集管理網域』

叢集管理網域可用來管理需要在叢集環境中的節點間保持一致的資源。

叢集資源群組跳出程式

發生 CRG 的叢集相關事件之後，會呼叫叢集資源群組跳出程式。

對回復裝置 CRG 而言，跳出程式是可選用的，但對其他 CRG 類型而言則是必要的。使用叢集資源群組跳出程式時，會在全叢集事件發生時呼叫該程式，包括：

- 節點非預期地離開叢集時。
- 因為「結束叢集節點 (QcstEndClusterNode) API」或「移除叢集節點項目 (QcstRemoveClusterNodeEntry) API」而導致節點脫離叢集時。
- 因為「刪除叢集 (QcstDeleteCluster) API」而導致叢集遭到刪除時。
- 「啟動叢集節點 (QcstStartClusterNode) API」 啟動節點時。
- 重新建立具分段節點之通訊時。

跳出程式是由叢集中介軟體「IBM 商業夥伴」及具有叢集知識的應用程式提供者所撰寫及提供。

如需叢集資源群組跳出程式的詳細資訊 (包括傳送到每個動作碼之跳出程式的資訊)，請參閱「叢集 API」文件中的「叢集資源群組跳出程式」。

回復網域

回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

網域代表那些可從中存取叢集資源的叢集之節點。指定給特定叢集資源群組的叢集節點子集支援主要存取點、次要 (備份) 存取點、複製存取點或對等節點存取點。

在回復網域中，節點可具有四種角色，如下所示：

主要 回復叢集資源之主要存取點的叢集節點。

- 若為資料 CRG，主要節點包含資源的主要副本。
- 若為應用程式 CRG，主要節點是目前正在執行此應用程式的系統。
- 若為裝置 CRG，主要節點是裝置資源的現行擁有者。

註：使用地理鏡映時，裝置 CRG 之回復網域中的節點，也需要站台名稱與資料埠 IP 位址。如需詳細資訊，請參閱「站台名稱與資料埠 IP 位址」。

- 若為對等節點 CRG，不支援主要節點。

若 CRG 的主要節點失敗，或已起始手動切換，則該 CRG 的主要存取點會移至第一個備份節點

備份 若目前的主要節點失敗，或已起始手動切換時，將接管主要存取角色的叢集節點。

- 對於資料 CRG 而言，此叢集節點包含了目前以抄寫方式保留之資源的副本。
- 若為對等節點 CRG，不支援備份節點。

複製節點

有叢集資源副本，但無法擔任主要或備份角色的叢集節點。不允許錯誤後移轉或切換為複製節點。若您想讓複製節點變成主節點，您必須先將複製節點的角色變更為備份節點角色。

- 若為對等節點 CRG，定義為複製節點的節點代表叢集資源的非作用中存取點。

對等節點

未排序且可為叢集資源之作用中存取點的叢集節點。啟動 CRG 時，所有定義為對等節點的節點都將是作用中存取點。

- 若為對等節點 CRG，存取點完全由管理應用程式 (而不是系統) 控制。只有對等節點 CRG 才支援對等節點角色。

主要備份模型

若為加入主要備份模型的節點，回復網域中的每個節點都有一個與目前叢集作業環境有關的角色。在回復網域中，這稱為目前角色。由於叢集經歷了作業變更，如節點結束、節點啟動及節點失敗，節點的目前角色會因此而變更。回復網域中的每一個節點也有一個關係到喜好設定或理想叢集環境的角色。在回復網域中，這稱為喜好設定角色。喜好設定角色是靜態定義，當建立了叢集資源群組時，會起始設定該角色。儘管叢集環境變更，此角色也不會變更。只有從回復網域新增或移除節點，或從叢集移除節點時，才會變更喜好設定角色。您也可以手動變更喜好設定角色。

概念上，您可以檢視主要備份模型內的重復網域，如下所示：

表 1. 主要備份 CRG 的節點角色

節點	目前角色	喜好設定角色
A	備份 1	主要
B	備份 2	備份 1
C	主要	備份 2
D	複製節點	複製節點

在此範例中，節點 A、B、C 及 D 會提供屬於主要備份模型的 CRG 範例。「節點 C」擔任目前的主要節點。因為它有第二個備份節點的喜好設定角色，所以擔任主要節點之「節點 C」目前的角色是由於兩個錯誤後移轉或切換動作所造成的。針對第一個錯誤後移轉或切換動作，主節點角色會從「節點 A」移到「節點 B」，因為「節點 B」是定義為第一個備份節點。第二個錯誤後移轉或切換動作，觸發「節點 C」成為主要節點 (因為它定義為第二個備份節點)。「節點 D」目前及喜好設定角色是複製節點。在錯誤後移轉或切換期間，不能將複製節點視為存取點，除非將其角色手動變更為主要或備份節點。

註：回復網域中每一個節點的角色，也可以手動變更。此範例說明當發生切換或錯誤後移轉動作，且沒有手動變更回復網域中的角色指定時，回復網域中之角色的變更方式。

對等節點模型

若為對等節點模型，叢集資源群組內的節點可具有兩個角色中的一個：對等節點或複製節點。

表 2. 對等節點 CRG 的節點角色

節點	目前角色	喜好設定角色
A	對等節點	對等節點
B	對等節點	對等節點
C	對等節點	對等節點
D	複製節點	複製節點

節點 A、B、及 C 在回復網域中定義為對等節點。當「節點 A」上發生失敗時，無論目前角色為何，它都會傳送至回復網域中的所有節點。這些節點會從「節點 A」的失敗點開始繼續作業。「節點 D」包含資料，卻因定義為「複製節點」而無法繼續作業。

任何數目的節點都能指定為對等節點或複製節點。對等節點未排序，並且能成為叢集資源的作用中存取點。複製節點未排序，並且無法成為叢集資源的作用中存取點，除非使用變更叢集資源群組 (QcstChangeClusterResourceGroup) API，將其角色從複製節點變更為對等節點。

相關工作

第 99 頁的『變更叢集資源群組 (CRG) 的回復網域』

您可以變更節點在叢集資源群組之回復網域中的角色，以及在回復網域中新增或移除節點。針對裝置叢集資源群組，您也可以變更回復網域之節點的站台名稱與資料埠 IP 位址。

第 100 頁的『執行切換』

執行手動切換會造成現行主要節點切換到備份節點，亦即叢集資源群組之回復網域中定義的備份節點。

叢集版本

叢集版本代表叢集上可用功能的層次。

版本化技術可讓叢集包含多重版次的系統，並可判定要使用的通訊協定層次，讓系統可以完全互用。如果您使用將包含各種系統版次的叢集，請參閱「多重版次叢集」。

實際上有兩種叢集版本：

潛在叢集版本

代表可用於給定節點的最進階叢集功能層次。這是節點能夠與其他叢集節點通訊的版本。

現行叢集版本

代表目前使用於所有叢集作業的版本。這是叢集中節點之間通訊的版本。

潛在的叢集版本在每一個作業系統版次上都在增加，此版本具有舊版的叢集所沒有的重要新叢集功能。如果現行叢集版本低於潛在的叢集版本，則無法使用該功能，因為有些節點無法辨識或處理其要求。若要利用這種新功能，叢集中的每一個系統都必須是相同的潛在叢集版本，且現行叢集版本也必須設為該層次。

當節點嘗試結合叢集時，其潛伏的叢集版本會與現行叢集版本比較。如果潛伏叢集版本的值不同於現行版本 (N) 或不等於下一個版本層次 (N+)，則不容許節點加入叢集。請注意，現行叢集版本最初由叢集中定義的第一個節點使用建立叢集 API 或指令上指定的值所設定。

例如，如果您要讓 V5R3 節點與 V5R4 節點共存，可執行下列其中一項動作：

- 在 V5R3 系統上建立叢集，並在 V5R4 節點上新增它。
- 在 V5R4 系統上建立叢集，指定容許將先前的節點新增到叢集中，然後將 V5R3 系統新增至叢集。

在多重版次叢集中，一律以最低節點版次 (現行叢集版本) 執行叢集通訊協定。這是在最先建立叢集時所定義的。N 可設定為在產生「建立叢集要求」的節點上執行的潛在節點版本，或是發送端潛在節點版本之前的一個叢集版本。叢集中各節點最多只能有一個叢集版本層次的差異。

一旦叢集中的所有系統都已升級至下一個版次，就可以升級叢集版本，以便讓新功能發揮作用。這可以經由調整叢集版本來達成。

警告： 如果作業系統的新版本不同於現行叢集版本，或比現行叢集版本高一版，則重新啟動叢集節點時會失敗。若要從這種狀況中回復，必須刪除節點並使用正確的版本來重建節點。

- | **警告：** 在叢集中使用切換式獨立磁碟儲存區時，在版次之間執行切換有一些限制。您需要將前版次的獨立磁碟儲存區切換到執行現行版次之 i5/OS 的系統，並讓它可以被使用。在執行現行版次之 i5/OS 的系統上使它可以使用之後，其內部內容會變更，且無法重新在前版次的系統上使用。

詳細閱讀「叢集 API」文件中的叢集版本，其中包括限制及叢集版本如何對應到 i5/OS 版次的相關資訊。

相關概念

第 81 頁的『多版次叢集』

如果建立叢集時將併入多重叢集版本的節點，則在建立叢集時需要某些步驟。

第 127 頁的『一般叢集問題』

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

相關工作

第 93 頁的『建立叢集』

若要建立及配置叢集，您至少要將一個節點併入叢集中，且您必須具有將在叢集中的至少一個節點的存取權限。

第 97 頁的『調整叢集的叢集版本』

叢集版本定義叢集中所有節點正在以此相互通訊的層次。

回復資源

回復資源是系統資源 (如資料、裝置及應用程式)，如果您已在系統上使用叢集，則這些資源具有高可用性。

若一個叢集節點 (叢集中之特定回復資源集的主要存取點) 引起故障，則另一個叢集節點 (定義為該資源集的備份節點) 現在應該成為存取點。

可回復的系統資源類型為：

1. 在節點之間抄寫的資料。
2. 使用 IP 位址，可從一個節點切換為另一個節點的應用程式。
3. 可從一個節點切換為另一個節點的硬體裝置。
4. 由叢集管理網域支援的對等節點資源。

在叢集資源群組 (CRG) 物件中，可找到與回復資源集相關節點之間之關係的定義。叢集資源群組透過叢集資源服務，在叢集中的各節點之間抄寫及協調。

相關概念

第 7 頁的『叢集資源群組』

- | 叢集資源群組 (CRG) 是 i5/OS 系統物件，它是一組或一套叢集資源，可用來管理在叢集環境中發生的事件。
- | 叢集資源群組說明回復網域，並提供某些叢集事件發生時所呼叫之叢集資源群組跳出程式的名稱。

第 8 頁的『叢集管理網域』

叢集管理網域可用來管理需要在叢集環境中的節點間保持一致的資源。

回復應用程式：

回復應用程式是可在不同的叢集節點上重新啟動，無需重新配置用戶端的應用程式。

請參閱「使應用程式回復」，以了解使應用程式回復的性質。

回復應用程式需要辨識用戶端與伺服器間之「網際網路通訊協定 (IP)」連線的暫時流失之能力。用戶端應用程式必須知道 IP 連線將暫時無法使用，且必須重試存取，而非結束或起始錯誤後移轉。同樣地，若您正在執行切換，則伺服器應用程式需要知道 IP 連線不再可用。最後，將傳回錯誤狀況至伺服器應用程式。一旦接收了此錯誤狀況，則若伺服器應用程式辨識出此狀況並正常結束，會是最佳的結果。

IP 位址接管是高可用性功能，用來保護用戶端，避免應用程式伺服器故障。**應用程式接管 IP 位址**是與應用程式相關的浮動位址。這個概念是要使用別名化的 IP 位址，以定義與多重應用程式伺服器或主電腦連結的浮動 IP 位址。當叢集中的一個應用程式伺服器失敗時，另一個叢集節點會擔任應用程式伺服器的責任，不需要您重新配置用戶端。

另外也介紹 IP 位址接管的支援，這是應用程式叢集資源群組 (CRG) 的概念。應用程式 CRG 是包含應用程式接管 IP 位址資源及回復網域的叢集資源群組。回復網域包含了支援特定應用程式之叢集內的應用程式伺服器清單。若單一資源失敗，叢集資源服務會起始失敗資源所屬的群組上之錯誤後移轉。

相關概念

第 27 頁的『使應用程式回復』

了解如何使應用程式回復。

第 10 頁的『回復網域』

回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

相關工作

第 26 頁的『叢集應用程式』

應用程式回復是叢集環境中的其中一個主要元素。如果您計劃在叢集中撰寫並使用高可用性的應用程式，則應了解這些應用程式具有特定的可用性規格。

回復資料:

回復資料是叢集中於一個以上的節點上的抄寫 (複製) 資料。

回復網域中的每一個節點都包含透過部分抄寫機制維護之回復資料的副本。定義為回復網域中之備份的節點，可擔任回復資料的主要存取點之角色。定義為抄寫的節點也包含資料的副本，但無法擔任主節點的角色。通常，將資料從主節點複製到複製節點，是用來減輕工作負荷，例如備份或唯讀查詢。

相關概念

第 23 頁的『抄寫』

抄寫可即時建立某些項目的副本。它表示將物件從叢集中的一個節點複製到叢集中的一或多個其他節點。

回復裝置:

回復裝置是實體資源，由配置物件 (如裝置說明) 來表示，可從叢集中一個以上的節點中存取。

在故障事件中，資源的存取點會切換為叢集資源群組回復網域中的第一個備份節點。獨立磁碟儲存區或獨立 ASP 是回復裝置，其離線或上線不會影響其他系統儲存體。此外，您還可以使用地理鏡映，此為跨站台鏡映 (XSM) 的子功能，是 i5/OS「選項 41 (高可用可切換資源)」的一部分。地理鏡映功能可在兩個站台上對獨立磁碟儲存區製作兩個相同的副本，以提供高可用性及災難回復。主要節點擁有的副本是正式作業副本，而在另一個站台上的備份節點所擁有的副本是鏡映副本。使用者作業及應用程式可存取主要節點 (擁有正式作業副本的節點) 上的獨立磁碟儲存區。

回復裝置叢集資源群組可包含可切換的裝置之清單。清單中的每一個裝置會識別可切換的獨立磁碟儲存區。當發生故障時，整個裝置集合會切換為備份節點。裝置也可選用性地轉接為切換/錯誤後移轉處理程序的一部份。有與可切換的裝置之清單相關的實體配置之相關限制。如需如何為定義為要回復之獨立磁碟儲存區設定適當配置的詳細資訊，請參閱「獨立磁碟儲存區」。

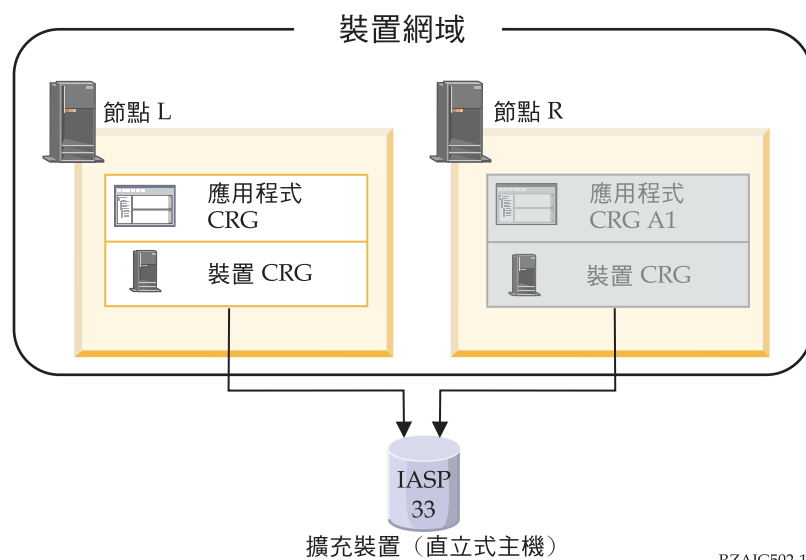
回復裝置 CRG 與 CRG 的其他類型非常類似。上述已提到其不同點為可切換裝置之清單。另一個不同點是跳出程式對裝置 CRG 而言是可選用的。若需要環境或資料特定處理程序，則跳出程式可使用於 CRG。如需此類 CRG 的詳細資訊，請參閱「建立叢集資源群組 (QcstCreateClusterResourceGroup) API」。

裝置網域

裝置網域是共用裝置資源的叢集中之節點的子集。更特別的是，裝置網域中的節點可參與回復裝置資源之部分集合的切換動作。

裝置網域是透過介面集來識別及管理，可讓您將節點新增到裝置網域，或從裝置網域移除節點。

裝置網域可用來管理某些必須的廣域資訊，以將回復裝置從一個節點切換到另一個節點。裝置網域中的所有節點都需要此資訊，以確定切換裝置時沒有發生衝突。例如，對於可切換的獨立磁碟儲存區之集合而言，在整個裝置網域中，獨立磁碟儲存區識別、硬碟機分派、及虛擬位址分派必須是唯一的。



一個叢集節點只能屬於一個裝置網域。在節點可新增到裝置 CRG 的回復網域之前，必須先將節點定義為裝置網域的成員。將在裝置 CRG 之回復網域中的所有節點，必須在相同的裝置網域中。

若要建立及管理裝置網域，您的系統上必須安裝「選項 41 (i5/OS - HA 可切換資源)」，並具備有效的授權碼。

相關概念

第 113 頁的『範例：使用獨立磁碟儲存區的切換式磁碟叢集』

使用切換式磁碟技術的叢集提供了抄寫資料的選擇方案。在切換式磁碟叢集中，資料實際上包含在獨立磁碟儲存區 (也稱為獨立 ASP) 中。

相關工作

第 101 頁的『將節點新增到裝置網域』

裝置網域是共用裝置資源的叢集中之節點的子集。

第 102 頁的『從裝置網域中移除節點』

裝置網域是共用裝置資源的叢集中之節點的子集。

選項 41 (HA 可切換資源):

若要建立及管理裝置網域，您的系統上必須安裝「選項 41 (i5/OS - HA 可切換資源)」，並具備有效的授權碼。

如果您要在叢集環境中執行下列任何動作，則需要安裝此功能：

- 使用「iSeries 領航員」叢集管理介面。
- 在系統之間切換獨立磁碟儲存區 (獨立 ASP)。
- 在分散於各處的系統之間進行跨站台鏡映

相關工作

第 101 頁的『將節點新增到裝置網域』

裝置網域是共用裝置資源的叢集中之節點的子集。

第 102 頁的『從裝置網域中移除節點』

裝置網域是共用裝置資源的叢集中之節點的子集。

叢集事件

叢集內可發生數種類型事件、動作及服務。

錯誤後移轉

當叢集中的伺服器在系統失效時自動切換為一或多個備份伺服器時，會發生錯誤後移轉。

將此與切換相對照，切換發生在您將存取權從某個伺服器手動切換至另一個伺服器時。一旦觸發了切換與錯誤後移轉，它們的功能是相同的。唯一的不同點是觸發事件的方式。

當發生錯誤後移轉時，存取會從目前作為叢集資源群組的回復網域中之主節點的叢集節點，切換為指定作為第一個備份的叢集節點。如需如何決定切換次序的詳細資訊，請參閱回復網域。

當多個叢集資源群組 (CRG) 與錯誤後移轉動作有關時，系統會先處理裝置 CRG (切換式裝置)，然後處理資料 CRG (切換式資料群組)，最後處理應用程式 CRG (切換式應用程式)。

錯誤後移轉訊息佇列會接收關於錯誤後移轉活動的訊息。您可以使用它來控制叢集資源群組的錯誤後移轉處理程序。

相關概念

第 10 頁的『回復網域』

回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

第 7 頁的『叢集資源群組』

叢集資源群組 (CRG) 是 i5/OS 系統物件，它是一組或一套叢集資源，可用來管理在叢集環境中發生的事件。叢集資源群組說明回復網域，並提供某些叢集事件發生時所呼叫之叢集資源群組跳出程式的名稱。

第 109 頁的『錯誤後移轉訊息佇列』

錯誤後移轉訊息佇列會接收關於錯誤後移轉活動的訊息。

第 77 頁的『叢集的硬體基本要求』

任何能夠執行 i5/OS V4R4M0 或更新版本的 iSeries 機型，在使用叢集上都是相容的。

相關工作

第 18 頁的『切換』

當您將資源存取權從某個伺服器手動切換至另一個伺服器時，會發生切換。

範例：失效：

通常，節點失效會產生錯誤後移轉，但其它原因也可能產生錯誤後移轉。

僅影響單一叢集資源群組的問題可能會對該叢集資源群組 (CRG) 產生錯誤後移轉，但不會對其它任何 CRG 產生錯誤後移轉。

下表顯示各種失效類型及其適合的種類：

失效	一般種類
CEC 硬體故障 (例如 CPU)	2
通訊配接卡、線路或路由器故障；或影響定義給節點的所有 IP 介面位址之 ENDTCPIFC	4
CEC 斷電	1
作業系統軟體機器檢查	2
發出 ENDTCP (*IMMED 或 *CNTRLD，有時間限制)	1
發出 ENDSBS QSYSWRK (*IMMED 或 *CNTRLD)	1
發出 ENDSBS (*ALL、*IMMED 或 *CNTRLD)	1
發出 ENDSYS (*IMMED 或 *CNTRLD)	1
發出 PWRDWN SYS (*IMMED 或 *CNTRLD)	1
叢集資源服務在系統中作用時按下了起始程式載入 (IPL) 按鈕	1
發出 QCSTCTL 工作的取消工作 (*IMMED 或 *CNTRLD，有時間限制)	1
發出 QCSTCRGM 工作的取消工作 (*IMMED 或 *CNTRLD，有時間限制)	1
發出叢集資源群組工作的取消工作 (*IMMED 或 *CNTRLD，有時間限制)	3
已呼叫結束叢集節點 API	1
已呼叫移除叢集節點 API	1
叢集資源群組工作發生了導致它異常結束的軟體錯誤	3
從控制面板輸入函數 8 或函數 3 來關閉系統電源	2
針對顯示的分割區關閉電源，請輸入功能 7	1
應用程式叢集資源群組的應用程式失效	3
一般種類：	
1. 節點上偵測為節點失效的所有叢集資源服務 (CRS) 失敗。節點實際上可能可以運作或節點已失效 (例如，斷電導致的系統失敗)。如果所有叢集資源服務失敗，則 CRS 所管理的資源將經歷錯誤後移轉程序。	
2. 所有 CRS 在偵測為叢集分割區的節點上失敗。該節點不一定在運作中。	
3. 個別叢集資源群組上發生失敗。這些狀況一律偵測為失敗。	
4. 發生失敗，但節點及叢集資源服務仍可運作且偵測為叢集分割區。	

當發生失敗時，叢集資源服務針對特定叢集資源群組所採取的動作，視失敗類型及叢集資源群組狀態而定。不過，在所有情況下，會呼叫跳出程式。錯誤後移轉必須使用失敗節點清單。當呼叫跳出程式時，它必須判斷是要處理單一節點失敗或處理失敗節點清單。

如果叢集資源群組非作用中，叢集資源群組回復網域中失敗節點的成員關係狀態變更為非作用中或分割區狀態。不過，節點角色不改變，而且備份節點不重新排序。呼叫「啟動叢集資源群組 (STRCRG)」指令或「啟動叢集資源群組 (QcstStartClusterResourceGroup) API」時，會在非作用中叢集資源群組中重新排序備份節點。但如果主節點非作用中，則「啟動叢集資源群組 API」會失敗。您必須發出「變更叢集資源群組 (CHGCRG)」指令或「變更叢集資源群組 (QcstChangeClusterResourceGroup) API」，將作用中節點指定為主要節點，然後重新呼叫「啟動叢集資源群組 API」。

如果叢集資源群組作用中且失敗節點不是主節點，錯誤後移轉會更新叢集資源群組回復網域中失敗回復網域成員的狀態。如果失敗節點是備份節點，則會重新排序備份節點清單，以便作用中節點位於清單開頭。

如果叢集資源群組處於作用中，且回復網域成員是主要節點，則會根據發生的失效類型來執行下列動作：

種類 1 失效

發生錯誤後移轉。每一個叢集資源群組中的主節點標示為非作用中且為上一個備份節點。第一個備份節點成為新主節點。首先，錯誤後移轉所有裝置叢集資源群組。然後，錯誤後移轉所有資料叢集資源群組。最後，錯誤後移轉所有應用程式叢集資源群組。如果任何 CRG 的錯誤後移轉偵測到沒有任何備份節點作用中，則 CRG 狀態設為未完成。

種類 2 失效

發生錯誤後移轉但不變更主節點。叢集分割區中主節點不是分割區成員的所有節點將結束作用中叢集資源群組。叢集資源群組回復網域中的節點狀態設為主要分割區中每個節點的分割區狀態。如果節點真的失敗但僅被偵測為分割區問題，而且失敗節點為主節點，您會失去該節點上所有資料及應用程式服務且不自動啟動錯誤後移轉。您必須宣告節點失敗，或重新啟用節點並在該節點上重新開始形成叢集。如需詳細資訊，請參閱將分割的節點變更為失敗。

種類 3 失效

如果僅單一叢集資源群組受到影響，則會個別地發生錯誤後移轉，因為叢集資源群組彼此獨立。如果某人取消了數個叢集資源工作，則可能發生數個叢集資源群組同時受到影響。不過，失敗類型是以逐一 CRG 的方式處理，且 CRG 之間不執行經過協調的錯誤後移轉。每一個叢集資源群組中的主節點標示為非作用中且成為最後一個備份節點。第一個備份節點成為新主節點。如果沒有作用中備份節點，則叢集資源群組狀態設為未完成。

種類 4 失效

這個種類與種類 2 類似。不過，當所有節點及節點上的叢集資源服務仍可運作時，並非所有節點都能互相通訊。叢集被分割，但主節點仍提供服務。不過，因為分割區的關係，您可能會遇到各種不同的問題。例如，如果主節點位於一個分割區中而所有備份節點或複製節點位於另一個分割區中，則您無法再複製資料，萬一主節點失敗也沒有保護。在包含主要節點的分割區中，錯誤後移轉程序會將叢集資源群組之回復網域中的節點狀態更新為另一個分割區中之所有節點的分割區。在不包含主要節點的分割區中，另一個分割區中的所有節點之叢集資源群組回復網域中的節點狀態會設為分割區。

相關概念

第 129 頁的『分割區錯誤』

某些叢集狀況很容易解決。如果發生叢集分割，您可以學習如何回復。此主題也告訴您如何避免叢集分割並舉例說明如何把分割區合併在一起。

切換

當您將資源存取權從某個伺服器手動切換至另一個伺服器時，會發生切換。

如果您要執行系統維護 (例如，套用暫時修訂程式 (PTF)、安裝新版次或升級您的系統)，通常會起始手動切換。將此與錯誤後移轉對照，錯誤後移轉會在主要節點上發生故障時自動發生。

當發生切換時，存取會從目前作為叢集資源群組之回復網域中之主節點的叢集節點，切換為指定作為第一個備份的叢集節點。如需如何決定切換次序的詳細資訊，請參閱回復網域。

若您正在執行多個 CRG 的管理切換，則您指定的次序應該考慮 CRG 之間的關係。例如，若您有一應用程式 CRG，它是以與裝置 CRG 相關的資料為依據，則為切換排序的步驟為：

1. 停止舊的主節點上之應用程式 (以停止對資料的變更)。
2. 將裝置 CRG 切換為新的主節點。
3. 將應用程式 CRG 切換為新的主節點。

| 4. 在新的主節點上重新啟動應用程式。

| **相關概念**

| 第 16 頁的『錯誤後移轉』

| 當叢集中的伺服器在系統失效時自動切換為一或多個備份伺服器時，會發生錯誤後移轉。

| 第 10 頁的『回復網域』

| 回復網域是叢集節點的子集，這些節點是因為共同的目的（例如執行回復動作或同步化事件）而一起組合在叢集資源群組（CRG）中。

| **相關工作**

| 第 100 頁的『執行切換』

| 執行手動切換會造成現行主要節點切換到備份節點，亦即叢集資源群組之回復網域中定義的備份節點。

| **重新結合**

| 重新結合表示已成為未參加的成員之後，再成為叢集的作用中成員。

| 例如，當節點成為非作用中之後，在節點上重新啟動叢集時，叢集節點會重新結合叢集。從已在叢集中作用的節點啟動叢集資源服務，就可在節點上啟動叢集資源服務。自叢集的第 3 版開始，節點可以啟動自己，且只要能夠在叢集中找到作用中節點，就能夠重新結合現行的作用中叢集。如需詳細資訊，請參閱「啟動叢集節點」。

| 假定節點 A、B、及 C 會組成叢集。節點 A 失敗。現在的作用中叢集是節點 B 及 C。一旦失敗的節點重新進行作業，則當節點從任何叢集節點（包括其本身）啟動時，它可以重新結合叢集。重新結合作業是以叢集資源群組的基礎來完成，該基礎表示每一個叢集資源群組（CRG）會獨立地結合叢集。

| 重新結合的主要功能會確定是在所有作用中回復網域節點上抄寫 CRG 物件。重新結合節點，以及所有現有的作用中叢集節點，必須有 CRG 物件的相同副本。此外，它們必須有部份內部資料的相同副本。

| 如果叢集管理網域中有非作用中的節點，當這些節點重新結合作用中網域時，會將在作用中網域中所做的任何資源變更傳送至非作用中節點。

| 當節點失敗時，叢集中之剩餘節點上的叢集資源服務之連續呼叫便能變更 CRG 物件中的資料。修改必須因為 API 或後續的節點之呼叫失敗而發生。對簡式叢集而言，是從目前在叢集中作用的部份節點，以 CRG 的副本來更新重新結合節點。但是，在所有例子中，這可能不正確。

| **相關工作**

| 第 96 頁的『啟動叢集節點』

| 啟動叢集節點會在叢集中的節點上啟動叢集資源服務。自叢集的第 3 版開始，節點可以啟動自己，且只要能夠在叢集中找到作用中節點，就能夠重新結合現行的作用中叢集。

| 第 131 頁的『變更分割的節點失敗』

| 有時，當節點故障真的發生時會報告分割的狀況。當叢集資源服務與一或多個節點失去通訊，但無法偵測節點是否仍然在運作時，就會發生這種狀況。當發生此狀況時，有一個簡式機制會指示節點已失效。

| **範例：重新結合：**

| 本主題說明節點重新結合叢集時所發生的動作。

| 下表說明當節點重新結合叢集時所採取的動作。此外，在 CRG 的回復網域之成員狀態欄位中，重新結合節點的狀態將從非作用中變更為作用中。會在 CRG 的回復網域中的所有節點上，呼叫跳出程式，並傳送「重新結合」的動作碼。

表 3. 重新結合作業

重新結合作業			
重新結合節點		叢集節點	
包含 CRG 的副本	不包含 CRG 的副本	包含 CRG 的副本	不包含 CRG 的副本
(1)	(2)	(3)	(4)

使用上表，可能有下列狀況：

1. 1 及 3
2. 1 及 4
3. 2 及 3
4. 2 及 4

若叢集中的節點有 CRG 的副本，則重新結合的一般規則是將 CRG 從叢集中的作用中節點，複製到重新結合節點。

重新結合狀況 1

來自叢集中的節點之 CRG 物件的副本，會傳送到結合節點。結果是：

- 以從叢集傳送的資料，在結合節點上更新 CRG 物件。
- 可能從結合節點刪除 CRG 物件。若結合節點不受叢集限制，而結合節點從 CRG 的回復網域移除時，就會發生這種狀況。

重新結合狀況 2

來自結合節點的 CRG 物件之副本，會傳送到所有叢集節點。結果是：

- 若沒有叢集節點在 CRG 的回復網域中，則不會變更。
- 可能在一或多個叢集節點上建立 CRG 物件。在下列實務範例中，就會發生這種狀況：
 - 節點 A、B、C 及 D 會組成叢集。
 - 這四個節點都在 CRG 的回復網域中。
 - 當節點 A 不受叢集限制時，會修改 CRG，以將 B 從回復網域移除。
 - 節點 C 及 D 失敗。
 - 叢集僅為沒有 CRG 副本的節點 B。
 - 節點 A 重新結合叢集。
 - 節點 A 有 CRG (即使目前是舊版)，而節點 B 沒有。CRG 建立在節點 B 上。當節點 C 及 D 重新結合叢集時，叢集中的 CRG 之副本會更新節點 C 及 D，且從回復網域移除節點 B 的前一個變更會遺失。

重新結合狀況 3

來自叢集中的節點之 CRG 物件的副本，會傳送到結合節點。結果是：

- 若結合節點不在 CRG 的回復網域中，則不會變更。
- 可能在結合節點上建立 CRG 物件。若叢集資源服務沒有在節點上作用，而在結合節點上刪除 CRG 時，就會發生這種狀況。

重新結合狀況 4

叢集中其中一個節點的部分內部資訊，可能會用來更新結合節點的相關資訊，但您看不見有發生任何狀況。

合併

合併作業與重新結合作業類似，不同處在於前者是在分割的節點重新開始通訊時發生。

分割區可能是仍在所有節點上作用的叢集資源服務之所在的真分割區。但是，因為通訊線路失敗，因此部分節點無法與其他節點通訊。或者，問題可能是節點已確實失敗，但未被偵測為失敗。

在第一個例子中，一旦通訊問題已修復，就會自動將分割區合併在一起。當兩個分割區定期地嘗試與分段的節點通訊，且最後重新建立相互的連接時，就會發生這種情況。在第二個例子中，必須在失敗的節點上重新啟動叢集資源服務，方法是從叢集中的任何節點啟動節點。

相關概念

第 19 頁的『重新結合』

重新結合表示已成為未參加的成員之後，再成為叢集的作用中成員。

第 129 頁的『分割區錯誤』

某些叢集狀況很容易解決。如果發生叢集分割，您可以學習如何回復。此主題也告訴您如何避免叢集分割並舉例說明如何把分割區合併在一起。

相關工作

第 96 頁的『啟動叢集節點』

啟動叢集節點會在叢集中的節點上啟動叢集資源服務。自叢集的第 3 版開始，節點可以啟動自己，且只要能夠在叢集中找到作用中節點，就能夠重新結合現行的作用中叢集。

第 131 頁的『變更分割的節點失敗』

有時，當節點故障真的發生時會報告分割的狀況。當叢集資源服務與一或多個節點失去通訊，但無法偵測節點是否仍然在運作時，就會發生這種狀況。當發生此狀況時，有一個簡式機制會指示節點已失效。

範例：合併：

合併作業可在數個不同的狀況下發生。

使用下列其中一個配置，可發生合併作業：

表 4. 合併主要分割區與次要分割區

合併	
主分割區	次要分割區

表 5. 合併兩個次要分割區

合併	
次要分割區	次要分割區

對叢集資源群組 (CRG) 而言，主分割區及次要分割區是唯一的。對主要備份 CRG 而言，將包含視為主要存取點之節點的分割區，定義為主要分割區。將不包含視為主要存取點之節點的分割區，定義為次要分割區。

若為對等節點 CRG，如果回復網域節點完全包含在一個分割區內，它將是主要分割區。如果回復網域節點跨越分割區，則沒有主要分割區。這兩個分割區均為次要分割區。

對叢集管理網域而言，如果將網域分割為兩個或更多個分割區，則每個分割區都將作為單一群組繼續作業。在每個分割區內，將繼續同步化資源的變更。當分割區又合併在一起時，系統將同步化每個分割區的變更。最後的結果就是受監視資源將在網域中保持一致。分割叢集管理網域後，便無法新增或移除 MRE。

表 6. 合併主要分割區與次要分割區

合併作業			
主分割區		次要分割區	
包含 CRG 的副本	「不」包含 CRG 的副本	包含 CRG 的副本	「不」包含 CRG 的副本
(1)	(2)	(3)	(4)

在主要次要合併期間 (如上表所顯示)，可能有下列狀況：

1. 1 及 3
2. 1 及 4
3. 2 及 3 (無法發生，因為大部分的分割區有作用中的主節點，且必須有 CRG 的副本。)
4. 2 及 4 (無法發生，因為大部分的分割區有作用中的主節點，且必須有 CRG 的副本。)

主要次要合併狀況

CRG 物件的副本傳送到次要分割區中的所有節點。下列動作可在次要分割區的節點上產生結果：

- 不執行動作，因為次要節點不在 CRG 的回復網域中。
- 以來自主分割區的資料，更新 CRG 的次要節點之副本。
- 從次要節點刪除 CRG 物件，因為次要節點不再存在於 CRG 的回復網域中。
- CRG 物件是建立在次要節點上，因為物件不存在。但是，節點是在從主分割區傳送之 CRG 副本的回復網域中。

表 7. 合併兩個次要分割區的實務範例

合併作業			
次要分割區		次要分割區	
包含 CRG 的副本	「不」包含 CRG 的副本	包含 CRG 的副本	「不」包含 CRG 的副本
(1)	(2)	(3)	(4)

在次要-次要合併期間 (如上表所顯示)，可能有下列狀況：

1. 1 及 3
2. 1 及 4
3. 2 及 3
4. 2 及 4

次要與次要合併狀況 1

若為主要備份 CRG，會選取具有 CRG 最新變更的節點，將 CRG 物件的副本傳送到其他分割區中的所有節點。若因為多個節點都顯示有最近的變更，而選取多個節點，則會使用回復網域次序來選取節點。

合併對等節點 CRG 的兩個次要分割區時，會將狀態為「作用中」之對等節點 CRG 的版本複製到其他分割區中的其他節點。如果對等節點 CRG 的兩個分割區具有相同的狀態，包含叢集資源群組回復網域中所列出之第一個節點的分割區，將複製到另一個分割區中的節點。

在主要備份 CRG 或對等節點 CRG 中的接收分割區節點上會發生下列動作：

- 不執行動作，因為節點不在 CRG 的回復網域中。
- CRG 是建立在節點上，因為節點是在其接收之 CRG 物件的副本之回復網域中。

- 從節點刪除 CRG，因為節點不在其接收之 CRG 物件的副本之回復網域中。

次要與次要合併狀況 2 及 3

- 選取有 CRG 物件之副本的分割區中之節點，以將物件資料傳送到其他分割區中的所有節點。若節點是在 CRG 的回復網域中，則 CRG 物件可能建立在接收分割區中的節點上。

次要與次要合併狀況 4

- 交換內部資料，以確定整個叢集的一致性。

- 隨後主分割區可分割為主分割區及次要分割區。若主要節點失敗，「叢集資源服務 (CRS)」會將其偵測為節點失敗。主分割區會變成次要分割區。若您結束使用「結束叢集節點 API」的主要節點，會發生相同的結果。若經由重新結合或合併作業，主節點變成在分割區中作用，則次要分割區會變更主分割區。

- 對於合併作業而言，不論節點在哪一個分割區中，都會在 CRG 的回復網域中的所有節點上，呼叫跳出程式。使用與重新結合相同的動作碼。沒有變更角色以作為合併的結果，但 CRG 的回復網域中之節點的成員狀態，會從分割變更為作用中。一旦所有分割區合併在一起，會清除分割區狀況，且可使用所有 CRG API 的分割區。

抄寫

- 抄寫可即時建立某些項目的副本。它表示將物件從叢集中的一個節點複製到叢集中的一或多個其他節點。

- 抄寫可保持您系統上的物件完全相同。如果您對叢集中一個節點上的物件做了某項變更，這個變更會抄寫到該叢集中其它節點上。

相關概念

- 第 14 頁的『回復資料』

- 回復資料是叢集中於一個以上的節點上的抄寫 (複製) 資料。

- 第 84 頁的『邏輯抄寫規劃』

- 使用邏輯抄寫維護資料的多份副本。從叢集中主節點抄寫或複製資料到回復網域中指定的備份節點。當主節點發生故障時，指定的備份節點接管為主要存取點，此時資料仍然可用。

活動訊號監督

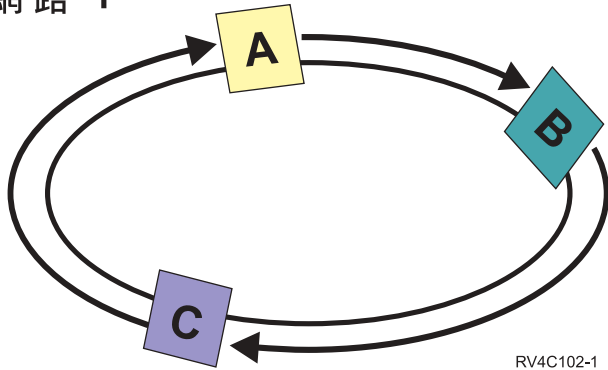
- 活動訊號監視是一種叢集資源服務功能，它可將信號從叢集中的每個節點，傳送到叢集中的所有其他節點，通知它們仍處於作用中，以確定每個節點都在作用中。

- 當節點的活動訊號失敗時，叢集資源服務會採取適當的動作。

- 請參考下列範例，以瞭解活動訊號監督的運作方式：

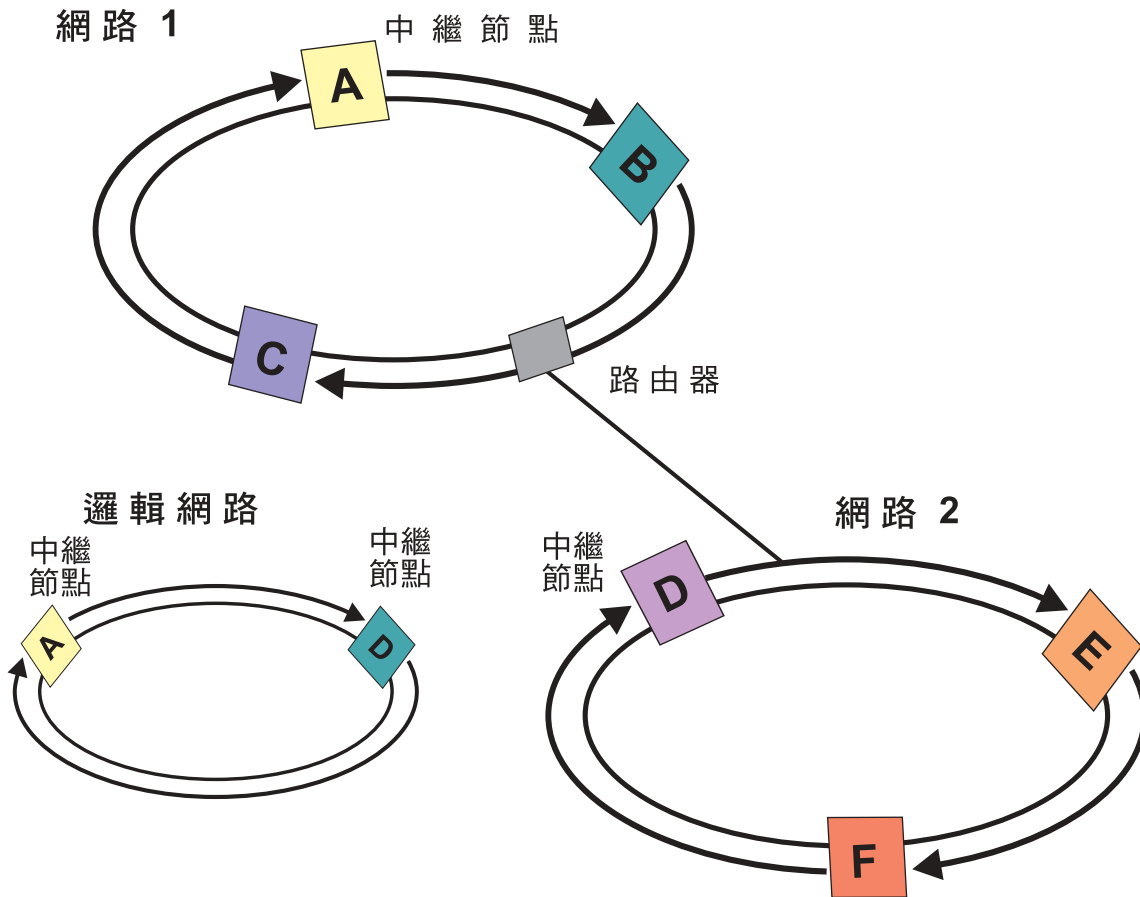
範例 1

網路 1



使用預設 (或正常) 設定值，會每隔 3 秒就將活動訊號訊息從叢集中的每一個節點，傳送到其上游鄰近點。例如，若您在「網路 1」上配置「節點 A」、「節點 B」及「節點 C」，則「節點 A」會傳送訊息到「節點 B」，「節點 B」會傳送訊息到「節點 C」，而「節點 C」會傳送訊息到「節點 A」。「節點 A」會要求確認來自「節點 B」的活動訊號，以及來自下游「節點 C」的送入活動訊號。事實上，活動訊號環會朝兩個方向進行。若「節點 A」沒有接收到來自「節點 C」的活動訊號，則「節點 A」及「節點 B」將持續每隔 3 秒傳送活動訊號。若「節點 C」遺失了四個連續活動訊號，則會發出活動訊號失敗的信號。

網路 1



範例 2

將另一個網路新增到此範例中，以顯示如何使用路由器及中繼節點。您在「網路 2」上配置「節點 D」、「節點 E」及「節點 F」。「網路 2」是使用路由器來連接「網路 1」。路由器可以是另一個 iSeries 伺服器或路由機箱，它可直接與其他地方的另一個路由器通訊。每一個區域網路都已指定中繼節點。此中繼節點是指定

給有網路中之最低節點 ID 的節點。將「節點 A」指定為「網路 1」上的中繼節點，將「節點 D」指定為「網路 2」上的中繼節點。然後會建立包含「節點 A」與「節點 D」的邏輯網路。使用路由器及中繼節點，這兩個網路上的節點可以互相監督，並發出任何節點失敗的信號。

相關概念

第 94 頁的『管理叢集』

本主題的資訊涵蓋一些涉及管理叢集的作業。

第 106 頁的『叢集效能』

變更叢集會影響管理叢集所需的額外執行時間。

相關工作

第 105 頁的『監督叢集狀態』

在必要時採取適當的動作，叢集資源服務便會使用可靠的訊息功能及活動訊號監視，來執行叢集及其元件的基本監視作業。

可靠的訊息功能

叢集資源服務的可靠的訊息功能會記錄叢集中之每一個節點，並確定所有節點都有關於叢集資源狀態的一致資訊。

可靠的傳訊使用重試及逾時值，這些值對形成叢集而言是唯一的。這些值會預先設定為適合大部份環境的值。但是，可透過「變更叢集資源服務設定值」介面來變更這些值。訊息重試及逾時值是用來決定失敗或發出分割區狀況信號之前，將訊息傳送到節點的次數。對於區域網路 (LAN) 而言，使用預設重試及逾時值，在失敗或發出分割區狀況信號之前，用盡重試次數所花費的時間量大約是 45 秒。針對遠端網路，允許以更多時間來判斷失敗或分割區狀況是否存在。對於遠端網路，您可以估算大約 4 分鐘又 15 秒。

相關概念

『變更叢集資源服務設定值』

影響訊息逾時和重試的預設值是為了說明大部份一般安裝而設定的。不過，可以變更這些值，使之更符合您的通訊環境。

第 94 頁的『管理叢集』

本主題的資訊涵蓋一些涉及管理叢集的作業。

相關工作

第 105 頁的『監督叢集狀態』

在必要時採取適當的動作，叢集資源服務便會使用可靠的訊息功能及活動訊號監視，來執行叢集及其元件的基本監視作業。

變更叢集資源服務設定值

影響訊息逾時和重試的預設值是為了說明大部份一般安裝而設定的。不過，可以變更這些值，使之更符合您的通訊環境。

可以透過下列其中一種方法來調整值：

- 設定符合環境的一般效能層次
- 對特定訊息調整參數設定值，以便做更特殊的調整。

在上述第一個方法中，訊息傳輸是調整到三種通訊層次的其中之一。一般層次是預設值，在「活動訊號監視」中有詳細說明。

第二種方法通常是在專家建議之下執行。

變更叢集資源服務 (QcstChgClusterResourceServices) API 詳細說明這兩種方法。

相關概念

第 23 頁的『活動訊號監督』

活動訊號監視是一種叢集資源服務功能，它可將信號從叢集中的每個節點，傳送到叢集中的所有其他節點，通知它們仍處於作用中，以確定每個節點都在作用中。

叢集分割區

叢集分割區是導因於通訊失敗之作用中叢集節點的子集。分割區中的成員會保持彼此之間的連通性。

只要當叢集中的一或多個節點之間的通訊遺失，及無法確定遺失節點的失敗時，叢集中就會出現叢集分割區。當偵測叢集分割區狀況時，叢集資源服務會限制您可在叢集分割區的節點上執行的動作類型。在分割期間會完成限制功能，如此一旦修正導致問題的原因後，叢集資源服務即可合併分割區。

分割叢集後，某些 CRG 作業會受限。如需針對每個類型的分割區限制哪些作業的詳細資訊，請參閱「叢集資源群組 API」。

如果已分割叢集管理網域，則會繼續在每個分割區的作用中節點中將變更同步化。當節點重新合併在一起時，叢集管理網域將會傳送在每個分割區上所做的變更，使作用中網域內的資源保持一致。

相關概念

第 80 頁的『避免叢集分割』

在叢集的所有節點之間配置備用通訊路徑，最能有效的避免一般網路相關的叢集分割。

第 129 頁的『分割區錯誤』

某些叢集狀況很容易解決。如果發生叢集分割，您可以學習如何回復。此主題也告訴您如何避免叢集分割並舉例說明如何把分割區合併在一起。

第 77 頁的『叢集的硬體基本要求』

任何能夠執行 i5/OS V4R4M0 或更新版本的 iSeries 機型，在使用叢集上都是相容的。

叢集應用程式

應用程式回復是叢集環境中的其中一個主要元素。如果您計劃在叢集中撰寫並使用高可用性的應用程式，則應了解這些應用程式具有特定的可用性規格。

藉由在叢集中利用回復應用程式，可在不同的叢集節點上重新啓動應用程式，而無需重新配置用戶端。此外，在切換或錯誤後移轉之後，將可使用與應用程式相關的資料。這表示應用程式的使用者可以體驗應用程式及其資料從主要節點切換成備份節點的最小程度岔斷，甚至不會感覺到岔斷。使用者無需知道後處理是否已經移動應用程式及資料。

爲了達到叢集中的應用程式回復，必須使用符合某些可用性規格的應用程式。爲了使應用程式可切換，以及使叢集中的應用程式使用者永遠可以使用應用程式，應用程式必須具有某些性質。如需這些應用程式特性的詳細資訊，請參閱「高可用性與叢集」。因爲這些基本要求的存在，下列選項可供您在叢集中使用切換式應用程式：

1. 購買具備叢集的軟體應用程式

具備叢集的軟體產品符合特定高可用性基本要求。

2. 撰寫或變更您的應用程式，以提高其可用性

獨立的軟體供應商及應用程式設計師可自訂應用程式，讓這些應用程式可在 iSeries 叢集環境中切換。

一旦您有回復應用程式，就必須在您的叢集內管理該回復應用程式。

相關概念

第 13 頁的『回復應用程式』

回復應用程式是可在不同的叢集節點上重新啟動，無需重新配置用戶端的應用程式。

具備叢集之應用程式的 i5/OS 架構

其他的一般使用者值是由高度可用的應用程式所提供，可辨識在故障、已規劃或未規劃事件中繼續使用的應用程式。

i5/OS 已提供應用程式回復架構，可支援各種程度上高度可用的應用程式。此架構的高遠目標是要以示範高可用性與高可用環境自動化的整合功能增強應用程式，並由叢集管理公用程式控制應用程式。

這些應用程式具有下列性質：

- 當主節點無法使用時，應用程式可切換為備份叢集節點。
- 應用程式在「回復定義」及「狀態資料區」中定義回復環境，由叢集管理應用程式來啟用應用程式的自動配置與啟動。
- 藉由讓應用程式 CRG 跳出程式利用 i5/OS 叢集資源服務的功能來處理叢集相關事件，應用程式提供了應用程式回復。
- 應用程式提供應用程式重新啟動功能，將使用者重新定位在應用程式功能表螢幕或其範圍之外。

示範更嚴謹可用性並重新啟動性質的應用程式具有下列性質：

- 藉由應用程式 CRG 跳出程式，透過更強的叢集事件之處理 (動作碼)，應用程式提供強化的應用程式回復。
- 應用程式提供應用程式重新啟動支援的更高層次。對於主電腦集中處理應用程式而言，確定控制或核對點功能會將使用者重新定位為異動界限。對於用戶端集中處理應用程式而言，使用者將體驗具最小服務中斷而的無縫式錯誤後移轉。

相關概念

iSeries High Availability and Clusters

撰寫高可用性的叢集應用程式

高可用性的應用程式，是在叢集環境中可由系統故障回復之應用程式。

有幾個可能的應用程式可用性層次：

1. 若發生應用程式錯誤，則應用程式會在相同節點上重新啟動其本身，並更正錯誤的任何潛伏原因 (例如毀損控制資料)。您可以檢視應用程式，就好像它是第一次啟動一樣。
2. 應用程式會執行部份核對點重新啟動處理程序。您可以檢視應用程式，就好像它接近失敗點一樣。
3. 若發生系統故障，則會在備份伺服器上重新啟動應用程式。您可以檢視應用程式，就好像它是第一次啟動一樣。
4. 若發生系統故障，則會在備份伺服器上重新啟動應用程式，並越過伺服器，執行部分核對點重新啟動處理程序。您可以檢視應用程式，就好像它接近失敗點一樣。
5. 若發生系統故障，則叢集中的其他節點會進行應用程式及其相關資料的已協調錯誤後移轉。您可以檢視應用程式，就好像它是第一次啟動一樣。
6. 若發生系統故障，則叢集中的其他節點會進行應用程式及其相關資料的已協調錯誤後移轉。應用程式會越過伺服器，執行部份核對點重新啟動處理程序。您可以檢視應用程式，就好像它接近失敗點一樣。

註：在上述例子 1 到 4 中，您負責回復資料。

使應用程式回復：

了解如何使應用程式回復。

預期回復應用程式具有下列性質：

- 可在此節點或另一個節點上重新啟動應用程式
- 用戶端可透過 IP 位址存取應用程式
- 應用程式是無狀態或瞭解狀態資訊
- 切換之後可使用與應用程式相關的資料

在叢集環境中，使應用程式能從系統故障回復中斷的三個必須元素為：

應用程式本身

應用程式對錯誤或系統故障的忍受程度為何，以及應用程式可重新啟動其本身的透通程度為何？

應用程式可以透過形成叢集功能來處理這個狀況。

相關資料

當故障發生時，會影響任何相關資料的可用性嗎？

利用形成叢集功能的叢集中介軟體「IBM 商業夥伴」抄寫產品，可處理此狀況。或者，資料可以儲存在切換式獨立磁碟儲存區 (切換式獨立 ASP) 中。

控制功能與管理

定義支援資料及應用程式可用性的環境之容易程度為何？

使用「形成叢集 API」，並以回復資料結合回復應用程式的協力廠商叢集管理解決方案，可處理此狀況。

重新啟動高可用性的叢集應用程式：

若要重新啟動應用程式，應用程式需要瞭解錯誤後移轉或切換時的狀態。

狀態資訊是應用程式特定的資訊；因此應用程式必須決定所需的資訊為何。若沒有任何狀態資訊，就可在您的 PC 上重新啟動應用程式。然而，您將必須重新建立您在應用程式中的定位。

有幾個方法可用來儲存備份系統的應用程式狀態資訊。每一個應用程式需決定其最佳方法。

- 應用程式可以將所有狀態資訊轉送到要求用戶端系統中。當發生切換或錯誤後移轉時，應用程式會使用儲存在用戶端上的狀態，在新伺服器中重新建立狀態。使用「分送資訊 API」或「叢集雜湊表 API」，可完成這個動作。
- 應用程式可依即時基礎來抄寫狀態資訊 (例如與應用程式相關的工作資訊及其他控制結構)。針對結構中的每一項變更，應用程式會將變更傳送至備份系統。
- 應用程式可將與應用程式相關的適當狀態資訊，儲存在該應用程式之叢集資源群組的跳出程式資料部分中。此方法假設需要少量的狀態資訊。您可以使用變更叢集資源群組 (QcstChangeClusterResourceGroup) API 來執行此動作。
- 應用程式可將狀態資訊及應用程式資料儲存在已抄寫至備份系統的資料物件中。
- 應用程式可將狀態資訊儲存在併入可切換 IASP 的資料物件 (也包含應用程式資料) 中。
- 應用程式可儲存用戶端的狀態資訊。
- 沒有要儲存的狀態資訊，您需要執行回復。

註：若應用程式使用某些核對點重新啟動處理程序的套表，則需要儲存的資訊數量會減少。狀態資訊僅儲存在預定的應用程式核對點中。而重新啟動會將您帶回到最新的已知核對點，類似於資料庫的確定控制處理程序的運作方式。

呼叫叢集資源群組 (CRG) 跳出程式：

在叢集環境的不同階段期間，會呼叫叢集資源群組跳出程式。

此程式會針對叢集中的資源建立環境必要的回復。對回復裝置 CRG 而言，跳出程式是可選用的，但對其他 CRG 類型而言則是必要的。使用叢集資源群組跳出程式時，會在全叢集事件發生時呼叫該程式，包括：

- 節點非預期地離開叢集時。
- 因為「結束叢集節點 (QcstEndClusterNode) API」或「移除叢集節點項目 (QcstRemoveClusterNodeEntry) API」而導致節點脫離叢集時。
- 因為「刪除叢集 (QcstDeleteCluster) API」而導致叢集遭到刪除時。
- 「啟動叢集節點 (QcstStartClusterNode) API」啟動節點時。
- 重新建立具分段節點之通訊時。

此跳出程式：

- 在已命名的啟動群組或呼叫程式的啟動群組 (*CALLER) 中執行。
- 若跳出程式有未處理的異常或已取消，則忽略重新啟動參數。
- 提供取消處理程式。

當執行叢集資源群組 API 時，會從具有在建立叢集資源群組 (QcstCreateClusterResourceGroup) API 中指定之使用者設定檔的個別工作中呼叫跳出程式。當呼叫跳出程式時，API 會自動建立個別工作。若資料 CRG 的跳出程式不成功或異常結束，會由具「還原」動作碼的回復網域中的所有作用中節點上，呼叫叢集資源群組跳出程式。此動作碼允許取消任何未完成的活動，並回復叢集資源群組的原始狀態。

若應用程式 CRG 的跳出程式不成功或異常結束，則若 CRG 的狀態為作用中，叢集資源服務將嘗試重新啟動應用程式。以「重新啟動」動作碼來呼叫叢集資源群組跳出程式。若無法在指定的最多嘗試次數內重新啟動應用程式，會以「錯誤後移轉」動作碼來呼叫叢集資源群組跳出程式。只有當以「啟動」動作碼呼叫跳出程式時，才會重設重新啟動計數，這可以是啟動 CRG、錯誤後移轉或切換的結果。

當啟動叢集資源群組時，在主節點上呼叫的應用程式 CRG 跳出程式，不會將控制返回叢集資源服務，直到應用程式本身結束或發生錯誤為止。應用程式 CRG 作用之後，若叢集資源服務必須向應用程式 CRG 跳出程式通知某些事件，則跳出程式的另一個案例會在不同的工作中啟動。預期會傳回「啟動」或「重新啟動」之外的其他任何動作碼。

當呼叫叢集資源群組跳出程式時，會傳送一組參數，其可識別處理的叢集事件、叢集資源的現行狀態及叢集資源的預期狀態。

如需叢集資源群組跳出程式的完整資訊 (包括傳送到每個動作碼之跳出程式的資訊)，請參閱「叢集 API」文件中的「叢集資源群組跳出程式」。QTSRTOOL 檔案庫中有提供原始程式碼範例，可用來作為撰寫跳出程式的基礎。請參閱 QATTSYSC 檔案中的 TCSTAPPEXT 成員。

應用程式 CRG 注意事項

應用程式叢集資源群組會管理應用程式回復。

管理應用程式 CRG 接管 IP 位址:

- 1 使用叢集資源服務管理應用程式 CRG 接管 IP 位址。您也可以手動管理這些位址。

有兩個方法可讓應用程式接管 IP 位址與管理的應用程式 CRG 連結。最簡單的方法 (此為預設值) 是讓叢集資源服務管理接管 IP 位址。此方法將指示叢集資源服務在回復網域的所有節點上建立接管 IP 位址，包括隨後新增至回復網域中的節點。選取此方法時，目前無法在回復網域中的任何節點上定義接管 IP 位址。

另一個方法是您自己管理接管 IP 位址。此方法會指示叢集資源服務不要執行配置接管 IP 位址的任何步驟；由使用者負責配置。在啟動叢集資源群組之前，您必須在回復網域的所有節點上 (除了複製節點之外)，新增接管 IP 位址。要新增至作用中 CRG 之回復網域的任何節點，在新增之前必須已配置接管 IP 位址。

多重子網路

雖然預設是讓所有回復網域節點在相同的子網路上，但有可能讓應用程式接管 IP 位址越過多重子網路來進行工作。請參閱啟用應用程式切換，以取得當回復網域中的節點跨子網路時，配置應用程式接管 IP 位址的步驟。

相關概念

第 31 頁的『範例：應用程式叢集資源群組 (CRG) 錯誤後移轉動作』

請參閱範例錯誤後移轉實務範例運作的方式。

第 99 頁的『以作用中接管 IP 位址來建立應用程式 CRG』

您可以指定在建立應用程式 CRG 時，允許使用作用中接管 IP 位址。只有在使用者配置接管 IP 位址時才允許這樣做。

越過子網路啟用應用程式切換:

通常，形成叢集需要應用程式叢集資源群組之回復網域中的所有叢集節點，都位於相同的 LAN 上 (使用相同的子網路位址)。

您可以用來達到將配置的應用程式接管 IP 位址，從回復網域中的一個節點，切換為另一個節點的基礎網路通訊協定，是「位址解析通訊協定 (ARP)」。

但是，它可以會擴展回復網域，併入常駐在其他 LAN (以商業路由器分隔) 上的叢集節點。

可能在網路的叢集節點及商業路由器上，透過「虛擬 IP 位址」支援的使用，以及使用「遞送資訊通訊協定 (RIP)」來擴展。如需詳細資料，請參閱『啟用應用程式切換』。

啟用應用程式切換:

- 1 當配置應用程式 CRG 時，叢集資源服務支援使用者配置的接管 IP 位址。

須以下列手動配置步驟來啟用切換環境。此指令集必須在回復網域的所有節點上執行，並對叢集中之其他節點 (將變成給定應用程式 CRG 的回復網域中之節點) 重複執行指令。

1. 選取應用程式 CRG 要使用的接管 IP 位址。
 - 為了避免混淆，此位址不應該與叢集節點或路由器使用的任何其他現有位址重疊。例如，若選擇 19.19.19.19，請確定 19.0.0.0 (19.19.0.0 或 ...) 不是系統遞送表所知的路徑。
 - 新增接管介面 (例如，19.19.19.19)；以線路說明 *VIRTUALIP、子網路遮罩 255.255.255.255 (主電腦路徑)、「最大傳輸單位」1500 (範圍自 576 至 16388 內的任何數字)、及「自動啟動」*NO，來建立接管介面。在下一個步驟中，將此接管位址 (例如，19.19.19.19) 識別為「相關本端介面」之前，它必須存在而且被當作 *VIRTUALIP 位址。然而，它不一定要處於作用中。
2. 當您建立叢集或將節點新增到叢集時，將預期的接管 IP 位址，以及指定給叢集通訊使用的一或兩個 IP 位址結合起來。
 - 例如，對形成叢集之本端使用的「乙太網路」匯流排上的叢集節點而言，表示使 19.19.19.19 接管位址成為 IP 位址上的「相關本端介面」。必須對每一個叢集節點上的每一個叢集位址完成此動作。

註：叢集位址將必須結束，以完成 CFGTCP 下的這種變更。

3. 建立叢集及建立任何 CRG。若為應用程式 CRG，請為「配置接管 IP 位址」欄位指定 QcstUserCfgsTakeoverIpAddr。請不要啟動任何應用程式 CRG。

4. 使用 CFGTCP 下的「配置 TCP/IP 應用程式」(選項 20)，再使用「配置 RouteD」(選項 2)，然後使用「變更 RouteD 屬性」(選項 1)，確定「提供」設定為 *YES。若不是，請將其設定為 *YES，並在每一個叢集節點上啟動或重新啟動 ROUTED (RIP 或 RIP-2)。
 - NETSTAT 選項 3 將使用「本端埠」(若目前執行中) 來顯示 ROUTED。ROUTED 必須在 CRG 回復網域中的每一個叢集節點上執行及通告路徑 (提供 = *YES)。
5. 確定與回復網域 LAN 交互連接的網路中之所有商業路由器，接受並通告 RIP 的主電腦路徑。
 - 不需要路由器的預設設定。語言將隨著路由器製造商而有不同，但在「RIP 介面」下，預期傳送「主電腦路徑」及接收動態主電腦。
 - 此亦套用至指向 iSeries 伺服器的路由器介面，以及路由器對路由器介面。

註: 在此實務範例中，不要使用 iSeries 伺服器作為路由器。請使用設計用來進行遞送的商業路由器 (IBM 或其他)。無法配置 iSeries 遞送來處理此功能。
6. 現在您可以在其中一個叢集節點上手動啟動接管位址，允許 RIP 最多使用 5 分鐘來傳送路徑，以及從 CRG 回復網域中的所有節點及將使用此位址之 LAN 上的已選取用戶端，連通測試 (ping) 接管位址。
 - 在此驗證測試之後，請再次確定接管位址已結束。
 - 當已啟動 CRG 時，形成叢集在指定的主節點上啟動位址。
7. 啟動應用程式 CRG。
 - 現在於指定的喜好節點上形成叢集，將啟動接管位址，且 RIP 將通告整個回復網域的路徑。RIP 可能花費多達 5 分鐘，以越過網域來更新路徑。RIP 功能是從啟動 CRG 功能獨立出來。

重要注意事項:

- 若應用程式 CRG 回復網域中的所有叢集節點沒有遵循上述的程序，則切換程序期間，叢集將擱置。
- 即使我們沒有對抄本節點進行錯誤後移轉，則在抄本節點往後遲早會變更為備份節點的事件中，在抄本節點上執行程序是不錯的方法。
- 如果要使用多重虛擬 IP 位址，則每個位址將需要個別的應用程式 CRG，以及要關聯的個別 IP 位址。此位址可能是相同實體配接卡上的另一個邏輯 IP 位址，或可能完全是另一個實體配接卡。此外，必須注意防止遞送表中有模稜兩可的狀況。執行下列動作，是達到目的的最佳方法：
 - 針對每一個虛擬 IP 位址，新增 *DFTRROUTE 至遞送表。
 - 這可以在 CFGTCP (選項 2) 下完成。
 - 將所有參數 (包括下一個跳躍點) 設定為完全相同，以達到選擇的路由器，但喜好的連結介面，應該設定為與虛擬 IP 位址 (將由此路徑表示) 相關的本端系統 IP 位址。

範例：應用程式叢集資源群組 (CRG) 錯誤後移轉動作:

請參閱範例錯誤後移轉實務範例運作的方式。

當因為超過重試限制或若工作被取消，使回復應用程式的叢集資源群組失敗時，會發生下列狀況：

- 以錯誤後移轉動作碼，在 CRG 之回復網域中的所有作用中節點上，呼叫叢集資源群組跳出程式。這指出叢集資源服務正準備錯誤後移轉應用程式之第一個備份的存取點。
- 叢集資源服務會結束接管主節點上的「網際網路通訊協定 (IP)」連線。如需接管 IP 位址的詳細資訊，請參閱「管理應用程式 CRG IP 位址」。
- 叢集資源服務會在第一個備份節點 (新的主節點) 上啟動接管 IP 位址。

- 僅在有「啟動」動作碼的新主節點上時，叢集資源服務才會提出呼叫叢集資源群組跳出程式的工作。此動作會重新啟動應用程式。

上述範例顯示一種錯誤後移轉實務的運作方式。其他錯誤後移轉實務可以不同方式運作。

範例：應用程式跳出程式：

此程式碼範例包含範例應用程式叢集資源群組跳出程式的程式碼。

您可以在 QUSRTOOL 檔案庫中找到此程式碼範例。

使用程式碼範例，即表示您同意程式碼授權及免責聲明資訊的條款。

```

/*****/
/*
/* Library:   QUSRTOOL
/* File:     QATTSYSC
/* Member:   TCSTAPPEXT
/* Type:     ILE C
/*
/* Description:
/* This is an example application CRG exit program which gets called for
/* various cluster events or cluster APIs. The bulk of the logic must
/* still be added because that logic is really dependent upon the unique
/* things that need to be done for a particular application.
/*
/* The intent of this example to to provide a shell which contains the
/* basics for building a CRG exit program. Comments throughout the example
/* highlight the kinds of issues that need to be addressed by the real
/* exit program implementation.
/*
/* Every action code that applies to an application CRG is handled in this
/* example.
/*
/* The tcstdtaara.h include is also shipped in the QUSRTOOL library. See
/* the TCSTDARA member in the QATTSYSC file.
/*
/* Change log:
/* Flag Reason   Ver   Date   User Id   Description
/* -----
/* ...   D98332   v5r1m0  000509  ROCH      Initial creation.
/* $A1   P9950070 v5r2m0  010710  ROCH      Dataarea fixes
/* $A2   D99055   v5r2m0  010913  ROCH      Added CancelFailover action code
/* $A3   D98854   v5r2m0  010913  ROCH      Added VerificationPhase action code
/* $A4   P9A10488 v5r3m0  020524  ROCH      Added example code to wait for data
/* CRGs on switchover action code
/*
/*
/*****/

/*-----*/
/*
/* Header files
/*
/*-----*/
#include          /* Useful when debugging          */
#include          /* offsetof macro          */
#include          /* system function          */
#include          /* String functions          */
#include          /* Exception handling constants/structures */
#include          /* Various cluster constants          */
#include          /* Structure of CRG information          */
#include "qusrtool/qattsysc/tcstdtaara" /* QCSTHAAPPI/QCSTHAAPPO data areas*/
#include          /* API to Retrieve contents of a data area */
#include          /* API error code type definition          */

```

```

#include          /* mitime builtin          */
#include          /* waittime builtin          */

/*-----*/
/*          */
/* Constants          */
/*          */
/*-----*/
#define UnknownRole -999
#define DependCrgDataArea "QCSTHAAPPO"
#define ApplCrgDataArea "QCSTHAAPPI"
#define Nulls 0x000000000000000000000000

/*-----*/
/*          */
/* The following constants are used in the checkDependCrgDataArea()
/* function. The first defines how long to sleep before checking the data
/* area. The second defines that maximum time to wait for the data area
/* to become ready before failing to start the application when the Start
/* CRG function is being run. The third defines the maximum wait time for
/* the Initiate Switchover or failover functions.          */
/*          */
/*-----*/
#define WaitSecondsIncrement 30
#define MaxStartCrgWaitSeconds 0
#define MaxWaitSeconds 900

/*-----*/
/*          */
/* As this exit program is updated to handle new action codes, change the
/* define below to the value of the highest numbered action code that is
/* handled.          */
/*          */
/*-----*/
#define MaxAc 21

/*-----*/
/*          */
/* If the exit program data in the CRG has a particular structure to it,
/* include the header file for that structure definition and change the
/* define below to use that structure name rather than char.          */
/*          */
/*-----*/
#define EpData char

/*-----*/
/*          */
/* Change the following define to the library the application resides in
/* and thus where the QCSTHAAPPO and QCSTHAAPPI data areas will be found.
/*          */
/*-----*/
#define ApplLib "QGPL"

/*-----*/
/*          */
/* Prototypes for internal functions.          */
/*          */
/*-----*/
static int getMyRole(Qcst_EXTP0100_t *, int, int);
#pragma argopt(getMyRole)
static int doAction(int, int, int, Qcst_EXTP0100_t *, EpData *);
#pragma argopt(doAction)
static int createCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int startCrg(int, int, Qcst_EXTP0100_t *, EpData *);

```

```

static int restartCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int endCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int verifyPhase(int, int, Qcst_EXTP0100_t *, EpData *);
static int deleteCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int memberIsJoining(int, int, Qcst_EXTP0100_t *, EpData *);
static int memberIsLeaving(int, int, Qcst_EXTP0100_t *, EpData *);
static int switchPrimary(int, int, Qcst_EXTP0100_t *, EpData *);
static int addNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int rmvNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int chgCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int deleteCrgWithCmd(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoPriorAction(int, int, Qcst_EXTP0100_t *, EpData *);
static int endNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int chgNodeStatus(int, int, Qcst_EXTP0100_t *, EpData *);
static int cancelFailover(int, int, Qcst_EXTP0100_t *, EpData *);
static int newActionCode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoCreateCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoStartCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoEndCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoMemberIsJoining(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoMemberIsLeaving(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoSwitchPrimary(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoAddNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoRmvNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoChgCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoCancelFailover(int, int, Qcst_EXTP0100_t *, EpData *);
static void bldDataAreaName(char *, char *, char *);
#pragma argopt(bldDataAreaName)
static int checkDependCrgDataArea(unsigned int);
#pragma argopt(checkDependCrgDataArea)
static void setApp1CrgDataArea(char *);
#pragma argopt(setApp1CrgDataArea)
static void cancelHandler(_CNL_Hndlr_Parms_T *);
static void unexpectedExceptionHandler(_INTRPT_Hndlr_Parms_T *);
static void endApplication(unsigned int, int, int, Qcst_EXTP0100_t *, EpData *);
#pragma argopt(endApplication)

/*-----*/
/*
/* Some debug routines
/*
/*-----*/
static void printParms(int, int, int, Qcst_EXTP0100_t *, EpData *);
static void printActionCode(unsigned int);
static void printCrgStatus(int);
static void printRcvyDomain(char *,
                           unsigned int,
                           Qcst_Rcvy_Domain_Array1_t *);
static void printStr(char *, char *, unsigned int);

/*-----*/
/*
/* Type definitions
/*
/*-----*/

/*-----*/
/*
/* This structure defines data that will be passed to the exception and
/* cancel handlers. Extend it with information unique to your application.*/
/*
/*-----*/
typedef struct {
    int *retCode;          /* Pointer to return code
    EpData *epData;       /* Exit program data from the CRG
    Qcst_EXTP0100_t *crgData; /* CRG data

```



```

    unsigned int actionCode; /* The action code */
    int role; /* This node's recovery domain role */
    int priorRole; /* This node's prior recovery domainrole */
} volatile HandlerDataT;

/*-----*/
/*
/* Function pointer array for handling action codes. When the exit program*/
/* is updated to handle new action codes, add the new function names to */
/* this function pointer array. */
/*
/*-----*/
static int (*fcn[MaxAc+1]) (int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) = {
    newActionCode, /* 0 - currently reserved */
    createCrg, /* 1 */
    startCrg, /* 2 */
    restartCrg, /* 3 */
    endCrg, /* 4 */
    verifyPhase, /* 5 - currently reserved */
    newActionCode, /* 6 - currently reserved */
    deleteCrg, /* 7 */
    memberIsJoining, /* 8 */
    memberIsLeaving, /* 9 */
    switchPrimary, /* 10 */
    addNode, /* 11 */
    rmvNode, /* 12 */
    chgCrg, /* 13 */
    deleteCrgWithCmd, /* 14 */
    undoPriorAction, /* 15 */
    endNode, /* 16 */
    newActionCode, /* 17 - applies only to a device CRG */
    newActionCode, /* 18 - applies only to a device CRG */
    newActionCode, /* 19 - applies only to a device CRG */
    chgNodeStatus, /* 20 */
    cancelFailover /* 21 */
};

/*-----*/
/*
/* Function pointer array for handling prior action codes when called with */
/* the Undo action code. When the exit program is updated to handle */
/* Undo for new action codes, add the new function names to this function */
/* pointer array. */
/*
/*-----*/
static int (*undoFcn[MaxAc+1]) (int role,
                                int priorRole,
                                Qcst_EXTP0100_t *crgData,
                                EpData *epData) = {
    newActionCode, /* 0 - currently reserved */
    undoCreateCrg, /* 1 */
    undoStartCrg, /* 2 */
    newActionCode, /* 3 */
    undoEndCrg, /* 4 */
    newActionCode, /* 5 - no undo for this action code */
    newActionCode, /* 6 - currently reserved */
    newActionCode, /* 7 */
    undoMemberIsJoining, /* 8 */
    undoMemberIsLeaving, /* 9 */
    undoSwitchPrimary, /* 10 */
    undoAddNode, /* 11 */
    undoRmvNode, /* 12 */
};

```

```

undoChgCrg,          /* 13 */
newActionCode,      /* 14 */
newActionCode,      /* 15 */
newActionCode,      /* 16 */
newActionCode,      /* 17 - applies only to a device CRG */
newActionCode,      /* 18 - applies only to a device CRG */
newActionCode,      /* 19 - applies only to a device CRG */
newActionCode,      /* 20 */
undoCancelFailover /* 21 */
};

/*****
/*
/* This is the entry point for the exit program.
/*
/*
*****/
void main(int argc, char *argv[]) {

    HandlerDataT hdldata;

/*-----*/
/*
/* Take each of the arguments passed in the argv array and cast it to
/* the correct data type.
/*
/*
/*-----*/
    int *retCode      = (int *)argv[1];
    unsigned int *actionCode = (unsigned int *)argv[2];
    EpData *epData    = (EpData *)argv[3];
    Qcst_EXTP0100_t *crgData = (Qcst_EXTP0100_t *)argv[4];
    char *formatName   = (char *)argv[5];

/*-----*/
/*
/* Ensure the format of the data being passed is what we are expecting.
/* If not, a change has been made and this exit program needs to be
/* updated to accommodate the change. Add appropriate error logging for
/* your application design.
/*
/*
/*-----*/
    if (0 != memcmp(formatName, "EXTP0100", 8))
        abort();

/*-----*/
/*
/* Set up the data that will be passed to the exception and cancel
/* handlers.
/*
/*
/*-----*/
    hdldata.retCode      = retCode;
    hdldata.epData      = epData;
    hdldata.crgData     = crgData;
    hdldata.actionCode  = *actionCode;
    hdldata.role        = UnknownRole;
    hdldata.priorRole   = UnknownRole;
    _VBDY(); /* force changed variables to home storage location */

/*-----*/

```

```

/*
/* Enable an exception handler for any and all exceptions.
/*
/*-----*/
#pragma exception_handler(unexpectedExceptionHandler, hdldata, \
                          _C1_ALL, _C2_ALL, _CTLA_INVOKE )

/*-----*/
/*
/* Enable a cancel handler to recover if this job is canceled.
/*
/*-----*/
#pragma cancel_handler(cancelHandler, hdldata)

/*-----*/
/*
/* Extract the role and prior role of the node this exit program is
/* running on. If the cluster API or event changes the recovery domain
/* (node role or membership status), the new recovery domain's offset is
/* passed in Offset_Rcvy_Domain_Array and the offset of the recovery
/* domain as it looked prior to the API or cluster event is passed in
/* Offset_Prior_Rcvy_Domain_Array. If the recovery domain isn't changed,
/* only Offset_Rcvy_Domain_Array can be used to address the recovery
/* domain.
/*
/*-----*/
hdldata.role = getMyRole(crgData,
                        crgData->Offset_Rcvy_Domain_Array,
                        crgData->Number_Nodes_Rcvy_Domain);
if (crgData->Offset_Prior_Rcvy_Domain_Array)
    hdldata.priorRole =
        getMyRole(crgData,
                  crgData->Offset_Prior_Rcvy_Domain_Array,
                  crgData->Number_Nodes_Prior_Rcvy_Domain);
else
    hdldata.priorRole = hdldata.role;
_VBDY(); /* force changed variables to home storage location */

/*-----*/
/*
/* Enable the following to print out debug information.
/*
/*-----*/
/*
printParms(*actionCode, hdldata.role, hdldata.priorRole, crgData,
epData);
*/

/*-----*/
/*
/* Do the correct thing based upon the action code. The return code
/* is set to the function result of doAction().
/*
/*-----*/
*retCode = doAction(*actionCode,
                    hdldata.role,

```

```

        hdldata.priorRole,
        crgData,
        epData);

/*-----*/
/*
/* The exit program job will end when control returns to the operating
/* system at this point.
/*
/*
/*-----*/
return;

#pragma disable_handler /* unexpectedExceptionHandler */
#pragma disable_handler /* cancelHandler */
} /* end main()

/*****
/*
/* Get the role of this particular node from one of the views of the
/* recovery domain.
/*
/* APIs and cluster events which pass the updated and prior recovery domain*/
/* to the exit program are:
/* QcstAddNodeToRcvyDomain
/* QcstChangeClusterNodeEntry
/* QcstChangeClusterResourceGroup
/* QcstEndClusterNode (ending node does not get the prior domain)
/* QcstInitiateSwitchOver
/* QcstRemoveClusterNodeEntry (removed node does not get the prior domain)
/* QcstRemoveNodeFromRcvyDomain
/* QcstStartClusterResourceGroup (only if inactive backup nodes are
/* reordered)
/*
/* a failure causing failover
/* a node rejoining the cluster
/* cluster partitions merging
/*
/* All other APIs pass only the updated recovery domain.
/*
/*****
static int getMyRole(Qcst_EXTP0100_t *crgData, int offset, int
count) {

    Qcst_Rcvy_Domain_Array1_t *nodeData;
    unsigned int iter = 0;

/*-----*/
/*
/* Under some circumstances, the operating system may not be able to
/* determine the ID of this node and passes *NONE. An example of such a
/* circumstance is when cluster resource services is not active on a
/* node and the DLTCRG CL command is used.
/*
/*-----*/
if (0 == memcmp(crgData->This_Nodes_ID, QcstNone,
sizeof(Qcst_Node_Id_t)))
return UnknownRole;

/*-----*/
/*
/* Compute a pointer to the first element of the recovery domain array.
/*
/*

```

```

/*-----*/
nodeData = (Qcst_Rcvy_Domain_Array1_t *)((char *)crgData +
offset);

/*-----*/
/*
/* Find my node in the recovery domain array. I will not be in the
/* prior recovery domain if I am being added by the Add Node to Recovery
/* Domain API.
/*
/*
/*-----*/
while ( 0 != memcmp(crgData->This_Nodes_ID,
nodeData->Node_ID,
sizeof(Qcst_Node_Id_t))
&&
iter < count
) {
nodeData++;
iter++;
}

if (iter < count)
return nodeData->Node_Role;
else
return UnknownRole;
} /* end getMyRole() */

/*****/
/*
/* Call the correct function based upon the cluster action code. The
/* doAction() function was split out from main() in order to clarify the
/* example. See the function prologues for each called function for
/* information about a particular cluster action.
/*
/*
/* Each action code is split out into a separate function only to help
/* clarify this example. For a particular exit program, some action codes
/* may perform the same function in which case multiple action codes could
/* be handled by the same function.
/*
/*
/*****/
static int doAction(int actionCode,
int role,
int priorRole,
Qcst_EXTPO100_t *crgData,
EpData *epData) {

/*-----*/
/*
/* For action codes this exit program knows about, call a function to
/* do the work for that action code.
/*
/*
/*-----*/

if (actionCode <= MaxAc )
return (*fcn[actionCode]) (role, priorRole, crgData, epData);
else

/*-----*/
/*
/* IBM has defined a new action code in a new operating system release
/* and this exit program has not yet been updated to handle it. Take a

```

```

    /* default action for now. */
    /* */

/*-----*/
    return newActionCode(role, priorRole, crgData, epData);
} /* end doAction() */

/*****/
/* */
/* Action code = QcstCrgAcInitialize */
/* */
/* The QcstCreateClusterResourceGroup API was called. A new cluster */
/* resource group object is being created. */
/* */
/* Things to consider: */
/* - Check that the application program and all associated objects are on */
/* the primary and backup nodes. If the objects are not there, */
/* consider sending error/warning messages or return a failure return */
/* code. */
/* - Check that required data or device CRGs are on all nodes in the */
/* recovery domain. */
/* - Perform any necessary setup that is required to run the */
/* the application on the primary or backup nodes. */
/* - If this CRG is enabled to use the QcstDistributeInformation API, */
/* the user queue needed by that API could be created at this time. */
/* */
/*****/
static int createCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

    return QcstSuccessful;
} /* end createCrg() */

/*****/
/* */
/* Action code = QcstCrgAcStart */
/* */
/* The QcstStartClusterResourceGroup API was called. A cluster resource */
/* group is being started. */
/* The QcstInitiateSwitchOver API was called and this is the second action */
/* code being passed to the exit program. */
/* The fail over event occurred and this is the second action code being */
/* passed to the exit program. */
/* */
/* A maximum wait time is used when checking to see if all dependent CRGs */
/* are active. This is a short time if the CRG is being started because of */
/* the QcstStartClusterResourceGroup API. It is a longer time if it is */
/* because of a failover or switchover. When failover or switchover are */
/* being done, it make take a while for data or device CRGs to become */
/* ready so the wait time is long. If the Start CRG API is being used, the */
/* dependent CRGs should already be started or some error occurred, the */
/* CRGs were started out of order, etc. and there is no need for a long */
/* wait. */
/* */
/* Things to consider: */
/* - If this node's role is primary, the application should be started. */
/* This exit program should either call the application so that it runs */
/* in this same job or it should monitor any job started by this */
/* exit program so the exit program knows when the application job */
/* ends. By far, the simplest approach is run the application in this */
/* job by calling it. */
/* Cluster Resource Services is not expecting this exit program to */
/* return until the application finishes running. */

```

```

/* - If necessary, start any associated subsystems, server jobs, etc. */
/* - Ensure that required data CRGs have a status of active on all nodes */
/*   in the recovery domain. */
/* */
/*****/
static int startCrg(int role,
                   int doesNotApply,
                   Qcst_EXTP0100_t *crgData,
                   EpData *epData) {

    unsigned int maxWaitTime;

    /* Start the application if this node is the primary */
    if (role == QcstPrimaryNodeRole) {

/*-----*/
        /*
        /* Determine if all CRGs that this application CRG is dependent upon
        /* are ready. If the check fails, return from the Start action code.
        /* Cluster Resource Services will change the state of the CRG to
        /* Inactive.
        /* */
/*-----*/
        if (crgData->Cluster_Resource_Group_Status ==
QcstCrgStartCrgPending)
            maxWaitTime = MaxStartCrgWaitSeconds;
        else
            maxWaitTime = MaxWaitSeconds;
        if (QcstSuccessful != checkDependCrgDataArea(maxWaitTime))
            return QcstSuccessful;

/*-----*/
        /*
        /* Just before starting the application, update the data area to
        /* indicate the application is running.
        /* */
/*-----*/
        setApp1CrgDataArea(App1_Running);

/*-----*/
        /*
        /* Add logic to call application here. It is expected that control
        /* will not return until something causes the application to end: a
        /* normal return from the exit program, the job is canceled, or an
        /* unhandled exception occurs. See the cancelHandler() function for
        /* some common ways this job could be canceled.
        /* */
/*-----*/

/*-----*/
        /*
        /* After the application has ended normally, update the data area to
        /* indicate the application is no longer running.
        /* */
/*-----*/
        setApp1CrgDataArea(App1_Ended);
    }
}

```

```

else

/*-----*/
/*
/* On backup or replicate nodes, mark the status of the application in
/* the data area as not running.
/*
/*
/*-----*/
    setApp1CrgDataArea(App1_Ended);

    return QcstSuccessful;
} /* end startCrg()
    */

/*****
/*
/* Action code = QcstCrgAcRestart
/*
/* The previous call of the exit program failed and set the return
/* code to QcstFailWithRestart or it failed due to an exception and the
/* exception was allowed to percolate up the call stack. In either
/* case, the maximum number of times for restarting the exit program has
/* not been reached yet.
/*
/* This action code is passed only to application CRG exit programs which
/* had been called with the Start action code.
/*
/*
*****/
static int restartCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

/*-----*/
/*
/* Perform any unique logic that may be necessary when restarting the
/* application after a failure and then call the startCrg() function to
/* do the start functions.
/*
/*
/*-----*/

    return startCrg(role, doesNotApply, crgData, epData);
} /* end restartCrg()
    */

/*****
/*
/* Action code = QcstCrgAcEnd
/*
/* The end action code is used for one of the following reasons:
/* - The QcstEndClusterResourceGroup API was called.
/* - The cluster has become partitioned and this node is in the secondary
/* partition. The End action code is used regardless of whether the
/* CRG was active or inactive. Action code dependent data of
/* QcstPartitionFailure will also be passed.
/* - The application ended. Action code dependent data of
/* QcstResourceEnd will also be passed. All nodes in the recovery
/* domain will see the same action code (including the primary).
/* - The CRG job has been canceled. The exit program on this node will
/* be called with the End action code. QcstMemberFailure will be
/* passed as action code dependent data.
*****/

```



```

/* */
/* */
/* */
/* Things to consider: */
/* - If the CRG is active, the job running the application is canceled */
/* and the IP takeover address is ended AFTER the exit program is */
/* called. */
/* - If subsystems or server jobs were started as a result of the */
/* QcstCrgAcStart action code, end them here or consolidate all logic */
/* to end the application in the cancelHandler() since it will be */
/* invoked for all Cluster Resource Services APIs which must end the */
/* application on the current primary. */
/* */
/*****/
static int endCrg(int role,
                 int priorRole,
                 Qcst_EXTP0100_t *crgData,
                 EpData *epData) {

/*-----*/
/* */
/* End the application if it is running on this node. */
/* */
/*-----*/
endApplication(QcstCrgAcRemoveNode, role, priorRole, crgData,
epData);

return QcstSuccessful;
} /* end endCrg() */

/*****/
/* */
/* Action code = QcstCrgAcVerificationPhase */
/* */
/* The verification phase action code is used to allow the exit program to */
/* do some verification before proceeding with the requested function */
/* identified by the action code depended data. If the exit program */
/* determines that the requested function cannot proceed it should return */
/* QcstFailWithOutRestart. */
/* */
/* */
/* NOTE: The exit program will NOT be called with Undo action code. */
/* */
/*****/
static int verifyPhase(int role,
                     int doesNotApply,
                     Qcst_EXTP0100_t *crgData,
                     EpData *epData) {

/*-----*/
/* */
/* Do verification */
/* */
/*-----*/
if (crgData->Action_Code_Dependent_Data == QcstDltCrg) {
    /* do verification */
    /* if ( fail ) */
    /* return QcstFailWithOutRestart */
}

return QcstSuccessful;
} /* end verifyPhase() */

```

```

/*****/
/*
/* Action code = QcstCrgAcDelete
/*
/* The QcstDeleteClusterResourceGroup or QcstDeleteCluster API was called.
/* A cluster resource group is being deleted while Cluster Resource
/* Services is active.
/* If the QcstDeleteCluster API was used, action code dependent data of
/* QcstDltCluster is passed.
/* If the QcstDeleteCluster API was used and the CRG is active, the exit
/* program job which is still active for the Start action code is canceled*/
/* after the Delete action code is processed.
/*
/* Things to consider:
/* - Delete application programs and objects from nodes where they are
/* no longer needed such as backup nodes. Care needs to be exercised
/* when deleting application objects just because a CRG is being
/* deleted since a particular scenario may want to leave the
/* application objects on all nodes.
/*
/*****/
static int deleteCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

    return QcstSuccessful;
} /* end deleteCrg()
/*****/
/*
/* Action code = QcstCrgAcReJoin
/*
/* One of three things is occurring-
/* 1. The problem which caused the cluster to become partitioned has been
/* corrected and the 2 partitions are merging back together to become
/* a single cluster. Action code dependent data of QcstMerge will be
/* passed.
/* 2. A node which either previously failed or which was ended has had
/* cluster resource services started again and the node is joining the
/* cluster. Action code dependent data of QcstJoin will be passed.
/* 3. The CRG job on a particular node which may have been canceled or
/* ended has been restarted. Action code dependent data of QcstJoin
/* will be passed.
/*
/* Things to consider:
/* - If the application replicates application state information to other*/
/* nodes when the application is running, this state information will
/* need to be resynchronized with the joining nodes if the CRG is
/* active.
/* - Check for missing application objects on the joining nodes.
/* - Ensure the required data CRGs are on the joining nodes.
/* - If the application CRG is active, ensure the required data CRGs are
/* active.
/*
/*****/
static int memberIsJoining(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

/*-----*/
/*
/* Ensure the data area status on this node starts out indicating
/* the application is not running if this node is not the primary.
*/

```

```

/*                                                                 */
/*-----*/
if (role != QcstPrimaryNodeRole) {
    setApp1CrgDataArea(App1_Ended);
}

/*-----*/
/*                                                                 */
/* If a single node is rejoining the cluster, you may do a certain set of */
/* actions. Whereas if the nodes in a cluster which became partitioned */
/* are merging back together, you may have a different set of actions. */
/*                                                                 */
/*-----*/
if (crgData->Action_Code_Dependent_Data == QcstJoin) {
    /* Do actions for a node joining. */
}
else {
    /* Do actions for partitions merging. */
}

return QcstSuccessful;
} /* end memberIsJoining() */

/*****
/*                                                                 */
/* Action code = QcstCrgAcFailover */
/*                                                                 */
/* Cluster resource services on a particular node(s) has failed or ended */
/* for this cluster resource group. The Failover action code is passed */
/* regardless of whether the CRG is active or inactive. Failover can */
/* happen for a number of reasons: */
/*                                                                 */
/* - an operator canceled the CRG job on a node. Action code dependent */
/* data of QcstMemberFailure will be passed. */
/* - cluster resource services was ended on the node (for example, the */
/* QSYSWRK subsystem was ended with CRS still active). Action code */
/* dependent data of QcstNodeFailure will be passed. */
/* - the application for an application CRG has failed on the primary */
/* node and could not be restarted there. The CRG is Active. */
/* Action code dependent data of QcstApp1Failure will be passed. */
/* - the node failed (such as a power failure). Action code dependent */
/* data of QcstNodeFailure will be passed. */
/* - The cluster has become partitioned due to some communication failure */
/* such as a communication line or LAN failure. The Failover action */
/* code is passed to recovery domain nodes in the majority partition. */
/* Nodes in the minority partition see the End action code. Action */
/* code dependent data of QcstPartitionFailure will be passed. */
/* - A node in the CRG's recovery domain is being ended with the */
/* QcstEndClusterNode API. The node being ended will see the End Node */
/* action code. All other nodes in the recovery domain will see the */
/* Failover action code. Action code dependent data of QcstEndNode */
/* will be passed for the Failover action code. */
/* - An active recovery domain node for an active CRG is being removed */
/* from the cluster with the QcstRemoveClusterNodeEntry API. Action */
/* code dependent data of QcstRemoveNode will be passed. If an */
/* inactive node is removed for an active CRG, or if the CRG is */
/* inactive, an action code of Remove Node is passed. */
/*                                                                 */
/* The exit program is called regardless of whether or not the CRG is */
/* active. The exit program may have nothing to do if the CRG is not */
/* active. */
/*                                                                 */
/* If the CRG is active and the leaving member was the primary node, */

```

```

/* perform the functions necessary for failover to a new primary. */
/* */
/* The Action_Code_Dependent_Data field can be used to determine if: */
/* - the failure was due to a problem that caused the cluster to become */
/*   partitioned (all CRGs which had the partitioned nodes in the */
/*   recovery domain are affected) */
/* - a node failed or had cluster resource services ended on the node (all */
/*   CRGs which had the failed/ended node in the recovery domain are */
/*   affected) */
/* - only a single CRG was affected (for example a single CRG job was */
/*   canceled on a node or a single application failed) */
/* */
/* */
/* Things to consider: */
/* - Prepare the new primary node so the application can be started. */
/* - The application should NOT be started at this time. The exit */
/*   program will be called again with the QcstCrgAcStart action code if */
/*   the CRG was active when the failure occurred. */
/* - If the application CRG is active, ensure the required data CRGs are */
/* active. */
/* */
/*****
static int memberIsLeaving(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

/*-----*/
/* */
/* If the CRG is active, perform failover. Otherwise, nothing to do. */
/* */
/*-----*/
if (crgData->Original_Cluster_Res_Grp_Stat == QcstCrgActive) {

/*-----*/
/* */
/* The CRG is active. Determine if my role has changed and I am now */
/* the new primary. */
/* */
/*-----*/

if (priorRole != role && role == QcstPrimaryNodeRole) {

/*-----*/
/* */
/* I was not the primary but am now. Do failover actions but don't */
/* start the application at this time because this exit program will */
/* be called again with the Start action code. */
/* */
/*-----*/

/*-----*/

/*-----*/
/* */
/* Ensure the data area status on this node starts out indicating */
/* the application is not running. */
/* */
/*-----*/
setApplCrgDataArea(Appl_Ended);

```

```

/*-----*/
/*
/* If the application has no actions to do on the Start action code */
/* and will become active as soon as the takeover IP address is */
/* activated, then this code should be uncommented. This code will */
/* determine if all CRGs that this application CRG is dependent upon */
/* are ready. If this check fails, return failure from the action */
/* code. */
/*
/*
/*-----*/
/* if (QcstSuccessful != checkDependCrgDataArea(MaxWaitSeconds)) */
/* return QcstFailWithOutRestart; */

}
}

return QcstSuccessful;
} /* end memberIsLeaving() */

/*****
/*
/* Action code = QcstCrgAcSwitchover */
/*
/* The QcstInitiateSwitchOver API was called. The first backup node in */
/* the cluster resource group's recovery domain is taking over as the */
/* primary node and the current primary node is being made the last backup.*/
/*
/* Things to consider: */
/* - Prepare the new primary node so the application can be started. */
/* - The application should NOT be started at this time. The exit */
/* program will be called again with the QcstCrgAcStart action code. */
/* - The job running the application is canceled and the IP takeover */
/* address is ended prior to the exit program being called on the */
/* current primary. */
/* - Ensure required data or device CRGs have switched over and are */
/* active. */
/*
/*****
static int switchPrimary(int role,
                        int priorRole,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

/*-----*/
/*
/* See if I am the old primary. */
/*
/*-----*/
if (priorRole == QcstPrimaryNodeRole) {

/*-----*/
/*
/* Do what ever needs to be done to cleanup the old primary before the */
/* switch. Remember that that job which was running the exit program */
/* which started the application was canceled already. */
/*
/* One example may be to clean up any processes holding locks on the */
/* database. This may have been done by the application cancel */
/* handler if one was invoked. */
/*
/*-----*/
}
}

```

```

/*-----*/
/*
/* I'm not the old primary. See if I'm the new primary.
/*
/*
/*-----*/
else if (role == QcstPrimaryNodeRole) {

/*-----*/
/*
/* Do what ever needs to be done on the new primary before the
/* application is started with the QcstCrgAcStart action code.
/*
/*
/*-----*/

/*-----*/
/*
/* Ensure the data area status on this nodes starts out indicating
/* the application is not running.
/*
/*
/*-----*/
setApp1CrgDataArea(App1_Ended);

/*-----*/
/*
/* If the application has no actions to do on the Start action code
/* and will become active as soon as the takeover IP address is
/* activated, then this code should be uncommented. This code will
/* determine if all CRGs that this application CRG is dependent upon
/* are ready. If this check fails, return failure from the action
/* code.
/*
/*
/*-----*/
/*
/* if (QcstSuccessful != checkDependCrgDataArea(MaxWaitSeconds))
/* return QcstFailWithOutRestart;
/*
}
else {

/*-----*/
/*
/* This node is one of the other backup nodes or it is a replicate
/* node. If there is anything those nodes must do, do it here. If
/* not, remove this else block.
/*
/*
/*-----*/

/*-----*/
/*
/* Ensure the data area status on this nodes starts out indicating
/* the application is not running.
/*
/*
/*-----*/
setApp1CrgDataArea(App1_Ended);
}

return QcstSuccessful;

```

```

} /* end switchPrimary() */

/*****
/*
/* Action code = QcstCrgAcAddNode
/*
/* The QcstAddNodeToRcvyDomain API was called. A new node is being added
/* to the recovery domain of a cluster resource group.
/*
/* Things to consider:
/* - A new node is being added to the recovery domain. See the
/* considerations in the createCrg() function.
/* - If this CRG is enabled to use the QcstDistributeInformation API,
/* the user queue needed by that API could be created at this time.
/*
*****/
static int addNode(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/*-----*/

/*
/* Determine if I am the node being added.
/*
/*-----*/

if (0 == memcmp(&crgData->This_Nodes_ID,
               &crgData->Changing_Node_ID,
               sizeof(Qcst_Node_Id_t)))
{
/*-----*/

/*
/* Set the status of the data area on this new node.
/*
/*-----*/

setApp1CrgDataArea(App1_Ended);

/*-----*/

/*
/* Create the queue needed by the Distribute Information API.
/*
/*-----*/

if (0 == memcmp(&crgData->DI_Queue_Name,
               Nulls,
               sizeof(crgData->DI_Queue_Name)))
{
}

return QcstSuccessful;
} /* end addNode()
*/

/*****

```

```

/* */
/* Action code = QcstCrgAcRemoveNode */
/* */
/* The QcstRemoveNodeFromRcvyDomain or the QcstRemoveClusterNodeEntry */
/* API was called. A node is being removed from the recovery domain of */
/* a cluster resource group or it is being removed entirely from the */
/* cluster. */
/* */
/* This action code is seen by: */
/* For the QcstRemoveClusterNodeEntry API: */
/* - If the removed node is active and the CRG is Inactive, all nodes in */
/* the recovery domain including the node being removed see this */
/* action code. The nodes NOT being removed see action code dependent */
/* data of QcstNodeFailure. */
/* - If the removed node is active and the CRG is Active, the node being */
/* removed sees the Remove Node action code. All other nodes in the */
/* recovery domain see an action code of Failover and action code */
/* dependent data of QcstNodeFailure. */
/* - If the node being removed is not active in the cluster, all nodes */
/* in the recovery domain will see this action code. */
/* For the QcstRemoveNodeFromRcvyDomain API: */
/* - All nodes see the Remove Node action code regardless of whether or */
/* not the CRG is Active. Action code dependent data of */
/* QcstRmvRcvyDmnNode will also be passed. */
/* */
/* Things to consider: */
/* - You may want to cleanup the removed node by deleting objects no */
/* longer needed there. */
/* - The job running the application is canceled and the IP takeover */
/* address is ended after the exit program is called if this is the */
/* primary node and the CRG is active. */
/* - If subsystems or server jobs were started as a result of the */
/* QcstCrgAcStart action code, end them here or consolidate all logic */
/* to end the application in the cancelHandler() since it will be */
/* invoked for all Cluster Resource Services APIs which must end the */
/* application on the current primary. */
/* */
/*****/
static int rmvNode(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/*-----*/

/* */
/* Determine if I am the node being removed. */
/* */

/*-----*/

    if (0 == memcmp(&crgData->This_Nodes_ID,
                   &crgData->Changing_Node_ID,
                   sizeof(Qcst_Node_Id_t)))
    {

/*-----*/
        /* */
        /* End the application if it is running on this node. */
        /* */

/*-----*/
        endApplication(QcstCrgAcRemoveNode, role, priorRole, crgData,
epData);

    }

```



```

    return QcstSuccessful;
} /* end rmvNode */

/*****
*/
/* Action code = QcstCrgAcChange */
/* */
/* The QcstChangeClusterResourceGroup API was called. Some attribute */
/* or information stored in the cluster resource group object is being */
/* changed. Note that not all changes to the CRG object cause the exit */
/* program to be called. As of V5R1M0, only these changes will cause the */
/* exit program to be called- */
/* - the current recovery domain is being changed */
/* - the preferred recovery domain is being changed */
/* */
/* If any of the above changes are being made but additionally the exit */
/* program is being changed to *NONE, the exit program is not called. */
/* */
/* Things to consider: */
/* - None unless changing the recovery domain affects information or */
/* processes for this cluster resource group. Note that the primary */
/* node cannot be changed with the QcstChangeClusterResourceGroup API */
/* if the CRG is active. */
/* */
/*****
static int chgCrg(int role,
                 int priorRole,
                 Qcst_EXTP0100_t *crgData,
                 EpData *epData) {

    return QcstSuccessful;
} /* end chgCrg() */

/*****
*/
/* Action code = QcstCrgAcDeleteCommand */
/* */
/* The Delete Cluster Resource Group (DLTCRG) CL command has been called */
/* to delete a cluster resource group object, the QcstDeleteCluster API */
/* has been called, or the QcstRemoveClusterNodeEntry API has been called. */
/* In each case, cluster resource services is not active on the cluster */
/* node where the command or API was called. Thus, this function is not */
/* distributed cluster wide but occurs only on the node where the CL */
/* command or API was called. */
/* */
/* If the QcstDeleteCluster API was used, action code dependent data of */
/* QcstDltCluster is passed. */
/* */
/* See the considerations in the deleteCrg() function */
/* */
/*****
static int deleteCrgWithCmd(int role,
                           int doesNotApply,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {

    return QcstSuccessful;
} /* end deleteCrgWithCmd() */

/*****
*/
/* Action code = QcstCrgEndNode */
/* */
/* The QcstEndClusterNode API was called or a CRG job was canceled. */

```

```

/* */
/* The QcstCrgEndNode action code is passed to the exit program only on the*/
/* node being ended or where the CRG job was canceled. On the node where */
/* a Cluster Resource Services job is canceled, action code dependent data*/
/* of QcstMemberFailure will be passed. */
/* When Cluster Resource Services ends on this node or the CRG job ends, it*/
/* will cause all other nodes in the cluster to go through failover */
/* processing. The action code passed to all other nodes will be */
/* QcstCrgAcFailover. Those nodes will see action code dependent data of */
/* QcstMemberFailure if a CRG job is canceled or QcstNodeFailure if the */
/* node is ended. */
/* */
/* Things to consider: */
/* - The job running the application is canceled and the IP takeover */
/* address is ended after the exit program is called if this is the */
/* primary node and the CRG is active. */
/* - If subsystems or server jobs were started as a result of the */
/* QcstCrgAcStart action code, end them here. */
/* */
/*****/
static int endNode(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/*-----*/
/* */
/* End the application if it is running on this node. */
/* */
/*-----*/

endApplication(QcstCrgEndNode, role, priorRole, crgData, epData);

return QcstSuccessful;
} /* end endNode() */

/*****/
/* */
/* Action code = QcstCrgAcChgNodeStatus */
/* */
/* The QcstChangeClusterNodeEntry API was called. The status of a node */
/* is being changed to failed. This API is used to inform cluster resource*/
/* services that the node did not partition but really failed. */
/* */
/* Things to consider: */
/* - The exit program was called previously with an action code of */
/* QcstCrgAcEnd if the CRG was active or an action code of */
/* QcstCrgAcFailover if the CRG was inactive because cluster resource */
/* services thought the cluster had become partitioned. The user is */
/* now telling cluster resource services that the node really failed */
/* instead of partitioned. The exit program has something to do only */
/* if it performed some action previously that needs to be changed now */
/* that node failure can be confirmed. */
/* */
/*****/
static int chgNodeStatus(int role,
                        int priorRole,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

return QcstSuccessful;
} /* end chgNodeStatus() */

/*****/

```

```

/* */
/* Action code = QcstCrgAcCancelFailover */
/* */
/* Cluster resource services on the primary node has failed or ended */
/* for this cluster resource group. A message was sent to the failover */
/* message queue specified for the CRG, and the result of that message */
/* was to cancel the failover. This will change the status of the CRG to */
/* inactive and leave the primary node as primary. */
/* */
/* Things to consider: */
/* - The primary node is no longer participating in cluster activities. */
/* The problem which caused the primary node to fail should be fixed */
/* so that the CRG may be started again. */
/* */
/*****/
static int cancelFailover(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

    return QcstSuccessful;
} /* end cancelFailover() */

/*****/
/* */
/* Action code = exit program does not know it yet */
/* */
/* A new action code has been passed to this exit program. This can occur */
/* after a new i5/OS release has been installed and some new cluster API */
/* was called or some new cluster event occurred. The logic in this exit */
/* program has not yet been updated to understand the new action code. */
/* */
/* Two different strategies could be used for the new action code. The */
/* correct strategy is dependent upon the kinds of things this particular */
/* exit program does for the application. */
/* */
/* One strategy is to not do anything and return a successful return code. */
/* This allows the new cluster API or event to run to completion. It */
/* allows the function to be performed even though this exit program */
/* did not understand the new action code. The risk, though, is that the */
/* exit program should have done something and it did not. At a minimum, */
/* you may want to log some kind of error message about what happened so */
/* that programming can investigate and get the exit program updated. */
/* */
/* The opposite strategy is to return an error return code such as */
/* QcstFailWithRestart. Of course doing this means that the new cluster */
/* API or event cannot be used until the exit program is updated for the */
/* new action code. Again, logging some kind of error message for */
/* programming to investigate would be worthwhile. */
/* */
/* Only the designer of the exit program can really decide which is the */
/* better course of action. */
/* */
/*****/
static int newActionCode(int role,
                          int doesNotApply,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

/*-----*/
/* */
/* Add logic to log an error somewhere - operator message queue, job */
/* log, application specific error log, etc. so that the exit program */
/* gets updated to properly handle the new action code. */
/* */

```

```

/* Note that if this is left coded as it is, this is the "don't do */
/* anything" strategy described in the prologue above. */
/* */
/*-----*/

return QcstSuccessful;
} /* end newActionCode() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Note: The exit program is never called with an undo action code for */
/* any of these prior action codes: */
/* QcstCrgAcChgNodeStatus */
/* QcstCrgAcDelete */
/* QcstCrgAcDeleteCommand */
/* QcstCrgEndNode */
/* QcstCrgAcRemoveNode (If the node being removed is active in the */
/* cluster and the API is Remove Cluster Node. */
/* The Remove Node From Recovery Domain will call */
/* with Undo and the Remove Cluster Node API will */
/* call with Undo if the node being removed is */
/* inactive. */
/* QcstCrgAcRestart */
/* QcstCrgAcUndo */
/* */
/* APIs that call an exit program do things in 3 steps. */
/* 1. Logic which must be done prior to calling the exit program. */
/* 2. Call the exit program. */
/* 3. Logic which must be done after calling the exit program. */
/* */
/* Any errors that occur during steps 2 or 3 result in the exit program */
/* being called again with the undo action code. This gives the exit */
/* program an opportunity to back out any work performed when it was first */
/* called by the API. The API will also be backing out any work it */
/* performed trying to return the state of the cluster and cluster objects */
/* to what it was before the API was called. */
/* */
/* It is suggested that the following return codes be returned for the */
/* specified action code as that return code will result in the most */
/* appropriate action being taken. */
/* */
/* QcstCrgAcInitialize: QcstSuccessful; The CRG is not created. */
/* QcstCrgAcStart: QcstSuccessful; The CRG is not started. */
/* QcstCrgAcEnd: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/* The cause of the failure needs to*/
/* investigated. */
/* QcstCrgAcReJoin: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/* The cause of the failure needs to*/
/* investigated. */
/* QcstCrgAcFailover: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/* The cause of the failure needs to*/
/* investigated. */
/* QcstCrgAcSwitchover: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/* The cause of the failure needs to*/
/* investigated. */
/* QcstCrgAcAddNode: QcstSuccessful; The node is not added. */
/* QcstCrgAcRemoveNode: QcstFailWithOutRestart; The CRG is set to Indoubt*/
/* The cause of the failure needs to*/
/* investigated. */
/* QcstCrgAcChange: QcstSuccessful; The recovery domain is not */
/* changed. */
/* */
/*****/

```

```

static int undoPriorAction(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

/*-----*/
/*
/* The prior action code defines what the exit program was doing when
/* it failed, was canceled, or returned a non successful return code.
/*
/*
/*-----*/
    if (crgData->Prior_Action_Code &lt;= MaxAc )
        return (*undoFcn[crgData-&lt;Prior_Action_Code])
                (role, priorRole, crgData,
epData);
    else

/*-----*/
/*
/* IBM has defined a new action code in a new operating system release
/* and this exit program has not yet been updated to handle it. Take a
/* default action for now.
/*
/*
/*-----*/
    return newActionCode(role, priorRole, crgData, epData);
} /* end undoPriorAction()

/*****
/*
/* Action code = QcstCrgAcUndo
/*
/* Prior action code = QcstCrgAcInitialize
/*
/* Things to consider:
/* The CRG will not be created. Objects that might have been created
/* on nodes in the recovery domain should be deleted since a subsequent
/* create could fail if those objects already exist.
/*
/*
/*****
static int undoCreateCrg(int role,
                        int doesNotApply,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

    return QcstSuccessful;
} /* end undoCreateCrg()

/*****
/*
/* Action code = QcstCrgAcUndo
/*
/* Prior action code = QcstCrgAcStart
/*
/* Things to consider:
/* Cluster Resource Services failed when it was finishing the Start CRG
/* API after it had already called the exit program with the Start
/* Action code.
/*
/* On the primary node, the exit program job which is running the
/* application will be canceled. The exit program will then be called
/* with the Undo action code.
/*
/*

```

```

/* All other nodes in the recovery domain will be called with the Undo */
/* action code. */
/* */
/*****/
static int undoStartCrg(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

    return QcstSuccessful;
} /* end undoStartCrg() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcEnd */
/* */
/* Things to consider: */
/* The CRG will not be ended. If the exit program did anything to bring */
/* down the application it can either restart the application or it can */
/* decide to not restart the application. If the application is not */
/* restarted, the return code should be set to QcstFailWithOutRestart so */
/* the status of the CRG is set to Indoubt. */
/* */
/*****/
static int undoEndCrg(int role,
                     int doesNotApply,
                     Qcst_EXTP0100_t *crgData,
                     EpData *epData) {

    return QcstFailWithOutRestart;
} /* end undoEndCrg() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcReJoin */
/* */
/* Things to consider: */
/* An error occurred which won't allow the member to join this CRG */
/* group. Anything done for the Join action code needs to be looked at */
/* to see if something must be undone if this member is not an active */
/* member of the CRG group. */
/* */
/*****/
static int undoMemberIsJoining(int role,
                               int doesNotApply,
                               Qcst_EXTP0100_t *crgData,
                               EpData *epData) {

    return QcstFailWithOutRestart;
} /* end undoMemberIsJoining() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcFailover */
/* */
/* Things to consider: */
/* This does not mean that the node failure or failing member is being */
/* undone. That failure is irreversible. What it does mean is that the */

```

```

/* exit program returned an error from the Failover action code or */
/* Cluster Resource Services ran into a problem after it called the exit */
/* program. If the CRG was active when Failover was attempted, it is */
/* not at this point. End the resilient resource and expect a human to */
/* look into the failure. After the failure is corrected, the CRG will */
/* must be started with the Start CRG API. */
/* */
/* */
/*****/
static int undoMemberIsLeaving(int role,
                              int doesNotApply,
                              Qcst_EXTP0100_t *crgData,
                              EpData *epData) {

    return QcstFailWithOutRestart;
} /* end undoMemberIsLeaving() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcSwitchover */
/* */
/* Things to consider: */
/* Some error occurred after the point of access was moved from the */
/* original primary and before it could be brought up on the new primary.*/
/* The IP address was ended on the original primary before moving the */
/* point of access but is started on the original primary again. Cluster*/
/* Resource Services will now attempt to move the point of access back */
/* to the original primary. The application exit program and IP takeover*/
/* address will be started on the original primary. */
/* */
/* */
/*****/
static int undoSwitchPrimary(int role,
                             int doesNotApply,
                             Qcst_EXTP0100_t *crgData,
                             EpData *epData) {

    return QcstFailWithOutRestart;
} /* end undoSwitchPrimary() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcAddNode */
/* */
/* Things to consider: */
/* If objects were created on the new node, they should be removed so */
/* that a subsequent Add Node to aRecovery Domain does not fail if it */
/* attempts to create objects again. */
/* */
/* */
/*****/
static int undoAddNode(int role,
                      int doesNotApply,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

    return QcstSuccessful;
} /* end undoAddNode() */

/*****/

```

```

/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcRemoveNode */
/* */
/* Things to consider: */
/* The node is still in the recovery domain. If objects were removed */
/* from the node, they should be added back. */
/* */
/*****/
static int undoRmvNode(int role,
                      int doesNotApply,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

    return QcstFailWithoutRestart;
} /* end undoRmvNode() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcChange */
/* */
/* Things to consider: */
/* Changes to the CRG will be backed out so that the CRG and its */
/* recovery domain look just like it did prior to the attempted change. */
/* Any changes the exit program made should also be backed out. */
/* */
/*****/
static int undoChgCrg(int role,
                     int doesNotApply,
                     Qcst_EXTP0100_t *crgData,
                     EpData *epData) {

    return QcstSuccessful;
} /* end undoChgCrg() */

/*****/
/* */
/* Action code = QcstCrgAcUndo */
/* */
/* Prior action code = QcstCrgAcCancelFailover */
/* */
/* Things to consider: */
/* This does not mean that the node failure or failing member is being */
/* undone. That failure is irreversible. What it does mean is that */
/* Cluster Resource Services ran into a problem after it called the exit */
/* program. The CRG will be InDoubt regardless of what is returned from */
/* this exit program call. Someone will need to manually look into the */
/* the failure. After the failure is corrected, the CRG will must be */
/* started with the Start CRG API. */
/* */
/* */
/*****/
static int undoCancelFailover(int role,
                              int doesNotApply,
                              Qcst_EXTP0100_t *crgData,
                              EpData *epData) {

    return QcstSuccessful;
} /* end undoCancelFailover() */

/*****/

```



```

/*
/* A simple routine to take a null terminated object name and a null
/* terminated library name and build a 20 character non-null terminated
/* qualified name.
/*
/*****
static void bldDataAreaName(char *objName, char* libName, char *qualName) {

    memset(qualName, 0x40, 20);
    memcpy(qualName, objName, strlen(objName));
    qualName += 10;
    memcpy(qualName, libName, strlen(libName));
    return;
} /* end bldDataAreaName

/*****
/*
/* The data area is checked to see if all the CRGs that this application
/* is dependent upon are ready. If they are not ready, a wait for a
/* certain amount of time is performed and the data area is checked again.
/* This check, wait loop continues until all dependent CRGs become ready or
/* until the maximum wait time has been reached.
/* The length of the wait can be changed to some other value if a
/* particular situation would be better with shorter or longer wait times.
/*
/*
/*****
static int checkDependCrgDataArea(unsigned int maxWaitTime) {

    Qus_EC_t errCode = { sizeof(Qus_EC_t), 0 };
    char dataAreaName[20];
    struct {
        Qwc_Rdtaa_Data_Returned_t stuff;
        char ready;
    } data;

/*-----*/
/*
/* This is an accumulation of the time waited for the dependent CRGs to
/* become ready.
/*
/*-----*/
    unsigned int timeWaited = 0;

/*-----*/
/*
/* Build definition of the amount of time to wait.
/*
/*-----*/
    _MI_Time    timeToWait;
    int hours   = 0;
    int minutes = 0;
    int seconds = WaitSecondsIncrement;
    int hundreths = 0;
    short int options = _WAIT_NORMAL;
    mitime( &timeToWait, hours, minutes, seconds, hundreths );

/*-----*/
/*
/* Build the qualified name of the data area.
/*
/*

```

```

/*-----*/
    bldDataAreaName(DependCrgDataArea, ApplLib, dataAreaName);

/*-----*/
/*
/* Get the data from the data area that indicates whether or not the
/* CRGs are all ready. This data area is updated by the High
/* Availability Business Partners when it is ok for the application to
/* proceed.
/*
/*
/*-----*/
    QWCRDTAA(&data,
            sizeof(data),
            dataAreaName,
            offsetof(Qcst_HAAPPO_t,Data_Status)+1, /* API wants a 1 origin */
            sizeof(data.ready),
            &errCode);

/*-----*/
/*
/* If the dependent CRGs are not ready, wait for a bit and check again.
/*
/*
/*-----*/
    while (data.ready != Data_Available) {

/*-----*/
/*
/* If the dependent CRGs have not become ready by the time we have
/* waited our maximum wait time, return an error. Consider logging
/* some message to describe why the application did not start so that
/* the problem can be looked into.
/*
/*
/*-----*/
        if (timeWaited >= maxWaitTime)
            return QcstFailWithOutRestart;

/*-----*/
/*
/* Wait to allow the data CRGs to become ready.
/*
/*
/*-----*/
        waittime(&timeToWait, options);
        timeWaited += WaitSecondsIncrement;

/*-----*/
/*
/* Get information from the data area again to see if the data CRGs are
/* ready.
/*
/*
/*-----*/
        QWCRDTAA(&data,
                sizeof(data),
                dataAreaName,
                offsetof(Qcst_HAAPPO_t,Data_Status)+1, /* API wants a 1 origin */
                sizeof(data.ready),
                &errCode);

```

```

}

return QcstSuccessful;
} /* end checkDependCrgDataArea */

/*****
/*
/* The application CRG data area is updated to indicate that the
/* application is running or to indicate it is not running. This data area
/* information is used by the High Availability Business Partners to
/* coordinate the switchover activities between CRGs that have dependencies
/* on each other.
/*
/*
*****/
static void setApp1CrgDataArea(char status) {

    char cmd[54];
    char cmdEnd[3] = {0x00, ' ', 0x00};

/*-----*/
/*
/* Set up the CL command string with the data area library name, the data
/* area name, and the character to put into the data area. Then run the
/* CL command.
/*
/*
/*-----*/
memcpy(cmd, "CHGDTAARA DTAARA(", strlen("CHGDTAARA DTAARA")+1);
strcat(cmd, ApplLib);
strcat(cmd, "/");
strcat(cmd, App1CrgDataArea);
strcat(cmd, " (425 1) VALUE("); /* @A1C */
cmdEnd[0] = status;
strcat(cmd, cmdEnd);

system(cmd);

return;
} /* end setApp1CrgDataArea */

/*****
/*
/* This function is called any time the exit program receives an exception
/* not specifically monitored for by some other exception handler. Add
/* appropriate logic to perform cleanup functions that may be required.
/* A failure return code is then set and control returns to the operating
/* system. The job this exit program is running in will then end.
/*
/*
/* When this function gets called, myData->role may still contain the
/* UnknownRole value if an exception occurred before this node's role
/* value was set. To be completely correct, the role should be tested
/* for UnknownRole before making any decisions based upon the value of
/* role.
/*
/*
*****/
static void unexpectedExceptionHandler(_INTRPT_Hndlr_Parms_T
*exData) {

/*-----*/
/*
/* Get a pointer to the structure containing data we passed to the
/* exception handler.
/*
/*
/*-----*/

```

```

/*-----*/
HandlerDataT *myData = (HandlerDataT *)exData->Com_Area;

/*-----*/
/*
/* Perform as much cleanup function as necessary. Some global state */
/* information may must be kept so the exception handler knows what */
/* steps were completed before the failure occurred and thus knows what */
/* cleanup steps must be performed. This state information could be */
/* kept in the HandlerDataT structure or it could be kept in some other */
/* location that this function can address. */
/*
/*-----*/

/*-----*/
/*
/* If this is the primary node and the application was started, end it. */
/* The application is ended because the exit program will be called again*/
/* with the Restart action code and we want the restartCrg() function to */
/* always work the same way. In addition, ending the application may */
/* clear up the condition that caused the exception that got us here. */
/* If possible, warn users and have them stop using the application so */
/* things are done in an orderly manner. */
/*
/*-----*/
endApplication(myData->actionCode,
               myData->role,
               myData->priorRole,
               myData->crgData,
               myData->epData);

/*-----*/
/*
/* Set the exit program return code. */
/*
/*-----*/
*myData->retCode = QcstFailWithRestart;

/*-----*/
/*
/* Let the exception percolate up the call stack. */
/*
/*-----*/
return;
} /* end unexpectedExceptionHandler */

/*****
/*
/* This function is called any time the job this exit program is running in*/
/* is canceled. The job could be canceled due to any of the following */
/* (the list is not intended to be all inclusive)- */
/* - an API cancels an active application CRG. The End CRG, Initiate */
/*   Switchover, End Cluster Node, Remove Cluster Node or Delete Cluster */
/*   API cancels the job which was submitted when the exit program was */
/*   called with a Start action code. */
/* - operator cancels the job from some operating system display such as */
/*   Work with Active Jobs */
*****/

```

```

/* - the subsystem this job is running in is ended */
/* - all subsystems are ended */
/* - the system is powered down */
/* - an operating system machine check occurred */
/*
/* When this function gets called, myData->role may still contain the
/* UnknownRole value if cancelling occurred before this node's role
/* value was set. To be completely correct, the role should be tested
/* for UnknownRole before making any decisions based upon the value of
/* role.
*/
*/
/*****
static void cancelHandler(_CNL_Hndlr_Parms_T *cnlData) {

/*-----*/
/*
/* Get a pointer to the structure containing data we passed to the
/* cancel handler.
*/
*/

/*-----*/
HandlerDataT *myData = (HandlerDataT *)cnlData->Com_Area;

/*-----*/
/*
/* Perform as much cleanup function as necessary. Some global state
/* information may must be kept so the cancel handler knows what
/* steps were completed before the job was canceled and thus knows if
/* the function had really completed successfully or was only partially
/* complete and thus needs some cleanup to be done. This state
/* information could be kept in the HandlerDataT structure or it could
/* be kept in some other location that this function can address.
*/
*/

/*-----*/

/*-----*/
/*
/* This job is being canceled. If I was running the application as a
/* result of the Start or Restart action codes, end the application now.
/* This job is being canceled because a Switch Over or some other
/* Cluster Resource Services API was used which affects the primary node
/* or someone did a cancel job with a CL command, from a system display,
/* etc.
*/
*/

/*-----*/
endApplication(myData->actionCode,
               myData->role,
               myData->priorRole,
               myData->crgData,
               myData->epData);

/*-----*/
/*
/* Set the exit program return code.
*/
*/

/*-----*/
*myData->retCode = QcstSuccessful;

```

```

/*-----*/
/*
/* Return to the operating system for final ending of the job.
/*
/*
/*-----*/
return;
} /* end cancelHandler */

/*****
/*
/* A common routine used to end the application by various action code
/* functions, the exception handler, and the cancel handler.
/*
/*
/*****
static void endApplication(unsigned int actionCode,
                          int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

    if ( role == QcstPrimaryNodeRole
        &&
            crgData->Original_Cluster_Res_Grp_Stat == QcstCrgActive)
    {
/*-----*/
/*
/* Add logic to end the application here. You may need to add logic
/* to determine if the application is still running because this
/* function could be called once for an action code and again from
/* the cancel handler (End CRG is an example).
/*
/*
/*-----*/

/*-----*/
/*
/* After the application has ended, update the data area to indicate
/* the application is no longer running.
/*
/*
/*-----*/
    setApp1CrgDataArea(App1_Ended);
    }

    return;
} /* end endApplication */

/*****
/*
/* Print out the data passed to this program.
/*
/*
/*****
static void printParms(int actionCode,
                      int role,
                      int priorRole,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

    unsigned int i;
    char *str;

```

```

/* Print the action code. */
printf("%s", "Action_Code = ");
printActionCode(actionCode);

/* Print the action code dependent data. */
printf("%s", " Action_Code_Dependent_Data = ");
switch (crgData->Action_Code_Dependent_Data) {
    case QcstNoDependentData: str = "QcstNoDependentData";
        break;
    case QcstMerge: str = "QcstMerge";
        break;
    case QcstJoin: str = "QcstJoin";
        break;
    case QcstPartitionFailure: str = "QcstPartitionFailure";
        break;
    case QcstNodeFailure: str = "QcstNodeFailure";
        break;
    case QcstMemberFailure: str = "QcstMemberFailure";
        break;
    case QcstEndNode: str = "QcstEndNode";
        break;
    case QcstRemoveNode: str = "QcstRemoveNode";
        break;
    case QcstApplFailure: str = "QcstApplFailure";
        break;
    case QcstResourceEnd: str = "QcstResourceEnd";
        break;
    case QcstDltCluster: str = "QcstDltCluster";
        break;
    case QcstRmvRcvyDmnNode: str = "QcstRmvRcvyDmnNode";
        break;
    case QcstDltCrg: str = "QcstDltCrg";
        break;
    default: str = "unknown action code dependent data";
}
printf("%s \n", str);

/* Print the prior action code. */
printf("%s", " Prior_Action_Code = ");
if (crgData->Prior_Action_Code)
    printActionCode(crgData->Prior_Action_Code);
printf("\n");

/* Print the cluster name. */
printStr(" Cluster_Name = ",
        crgData->Cluster_Name, sizeof(Qcst_Cluster_Name_t));

/* Print the CRG name. */
printStr(" Cluster_Resource_Group_Name = ",
        crgData->Cluster_Resource_Group_Name,
sizeof(Qcst_Crg_Name_t));

/* Print the CRG type. */
printf("%s \n", " Cluster_Resource_Group_Type =
QcstCrgApplResiliency");

/* Print the CRG status. */
printf("%s", " Cluster_Resource_Group_Status = ");
printCrgStatus(crgData->Cluster_Resource_Group_Status);

/* Print the CRG original status. */
printf("%s", " Original_Cluster_Res_Grp_Stat = ");
printCrgStatus(crgData->Original_Cluster_Res_Grp_Stat);

/* Print the Distribute Information queue name. */

```

```

printStr(" DI_Queue_Name = ",
        crgData->DI_Queue_Name,
sizeof(crgData->DI_Queue_Name));
printStr(" DI_Queue_Library_Name = ",
        crgData->DI_Queue_Library_Name,
sizeof(crgData->DI_Queue_Library_Name));

/* Print the CRG attributes. */
printf("%s", " Cluster_Resource_Group_Attr = ");
if (crgData->Cluster_Resource_Group_Attr &
QcstTcpConfigByUsr)
    printf("%s", "User Configures IP Takeover Address");
printf("\n");

/* Print the ID of this node. */
printStr(" This_Nodes_ID = ",
        crgData->This_Nodes_ID, sizeof(Qcst_Node_Id_t));

/* Print the role of this node. */
printf("%s %d \n", " this node's role = ", role);

/* Print the prior role of this node. */
printf("%s %d \n", " this node's prior role = ", priorRole);

/* Print which recovery domain this role comes from. */
printf("%s", " Node_Role_Type = ");
if (crgData->Node_Role_Type == QcstCurrentRcvyDmn)
    printf("%s \n", "QcstCurrentRcvyDmn");
else
    printf("%s \n", "QcstPreferredRcvyDmn");

/* Print the ID of the changing node (if any). */
printStr(" Changing_Node_ID = ",
        crgData->Changing_Node_ID, sizeof(Qcst_Node_Id_t));

/* Print the role of the changing node (if any). */
printf("%s", " Changing_Node_Role = ");
if (crgData->Changing_Node_Role == -3)
    printf("%s \n", "*LIST");
else if (crgData->Changing_Node_Role == -2)
    printf("%s \n", "does not apply");
else
    printf("%d \n", crgData->Changing_Node_Role);

/* Print the takeover IP address. */
printStr(" Takeover_IP_Address = ",
        crgData->Takeover_IP_Address,
sizeof(Qcst_TakeOver_IP_Address_t));

/* Print the job name. */
printStr(" Job_Name = ", crgData->Job_Name, 10);

/* Print the CRG changes. */
printf("%s \n", " Cluster_Resource_Group_Changes = ");
if (crgData->Cluster_Resource_Group_Changes &
QcstRcvyDomainChange)
    printf(" %s \n", "Recovery domain changed");
if (crgData->Cluster_Resource_Group_Changes &
QcstTakeOverIpAddrChange)
    printf(" %s \n", "Takeover IP address changed");

/* Print the failover wait time. */
printf("%s", "Failover_Wait_Time = ");
if (crgData->Failover_Wait_Time == QcstFailoverWaitForever)
    printf("%d %s \n", crgData->Failover_Wait_Time, "Wait
forever");
else if (crgData->Failover_Wait_Time == QcstFailoverNoWait)

```



```

    printf("%d %s \n", crgData->Failover_Wait_Time, "No wait");
else
    printf("%d %s \n", crgData->Failover_Wait_Time, "minutes");

/* Print the failover default action. */
printf("%s", "Failover_Default_Action = ");
if (crgData->Failover_Default_Action == QcstFailoverProceed)
    printf("%d %s \n", crgData->Failover_Default_Action,
"Proceed");
else
    printf("%d %s \n", crgData->Failover_Default_Action,
"Cancel");

/* Print the failover message queue name. */
printStr(" Failover_Msg_Queue = ",
        crgData->Failover_Msg_Queue,
sizeof(crgData->Failover_Msg_Queue));
printStr(" Failover_Msg_Queue_Lib = ",
        crgData->Failover_Msg_Queue_Lib,
sizeof(crgData->Failover_Msg_Queue_Lib));

/* Print the cluster version. */
printf("%s %d \n",
        " Cluster_Version = ", crgData->Cluster_Version);

/* Print the cluster version mod level */
printf("%s %d \n",
        " Cluster_Version_Mod_Level = ",
        crgData->Cluster_Version_Mod_Level);

/* Print the requesting user profile. */
printStr(" Req_User_Profile = ",
        crgData->Req_User_Profile,
sizeof(crgData->Req_User_Profile));

/* Print the length of the data in the structure. */
printf("%s %d \n",
        " Length_Info_Returned = ",
crgData->Length_Info_Returned);

/* Print the offset to the recovery domain array. */
printf("%s %d \n",
        " Offset_Rcvy_Domain_Array = ",
crgData->Offset_Rcvy_Domain_Array);

/* Print the number of nodes in the recovery domain array. */
printf("%s %d \n",
        " Number_Nodes_Rcvy_Domain = ",
crgData->Number_Nodes_Rcvy_Domain);

/* Print the current/new recovery domain. */
printRcvyDomain(" The recovery domain:",
        crgData->Number_Nodes_Rcvy_Domain,
(Qcst_Rcvy_Domain_Array1_t *)
        ((char *)crgData +
crgData->Offset_Rcvy_Domain_Array));

/* Print the offset to the prior recovery domain array. */
printf("%s %d \n",
        " Offset_Prior_Rcvy_Domain_Array = ",
crgData->Offset_Prior_Rcvy_Domain_Array);

/* Print the number of nodes in the prior recovery domain array. */
printf("%s %d \n",
        " Number_Nodes_Prior_Rcvy_Domain = ",
crgData->Number_Nodes_Prior_Rcvy_Domain);

```

```

/* Print the prior recovery domain if one was passed.          */
if (crgData->Offset_Prior_Rcvy_Domain_Array) {
    printRcvyDomain("  The prior recovery domain:",
                    crgData->Number_Nodes_Prior_Rcvy_Domain,
                    (Qcst_Rcvy_Domain_Array1_t *)
                    ((char *)crgData +
crgData->Offset_Prior_Rcvy_Domain_Array));
}

return;
} /* end printParms                                          */

/*****
/*
/* Print a string for the action code.
/*
/*
*****/
static void printActionCode(unsigned int ac) {

char *code;
switch (ac) {
    case QcstCrgAcInitialize: code = "QcstCrgAcInitialize";
                              break;
    case QcstCrgAcStart:      code = "QcstCrgAcStart";
                              break;
    case QcstCrgAcRestart:   code = "QcstCrgAcRestart";
                              break;
    case QcstCrgAcEnd:       code = "QcstCrgAcEnd";
                              break;
    case QcstCrgAcDelete:    code = "QcstCrgAcDelete";
                              break;
    case QcstCrgAcReJoin:    code = "QcstCrgAcReJoin";
                              break;
    case QcstCrgAcFailover:  code = "QcstCrgAcFailover";
                              break;
    case QcstCrgAcSwitchover: code = "QcstCrgAcSwitchover";
                              break;
    case QcstCrgAcAddNode:   code = "QcstCrgAcAddNode";
                              break;
    case QcstCrgAcRemoveNode: code = "QcstCrgAcRemoveNode";
                              break;
    case QcstCrgAcChange:    code = "QcstCrgAcChange";
                              break;
    case QcstCrgAcDeleteCommand: code = "QcstCrgAcDeleteCommand";
                              break;
    case QcstCrgAcUndo:      code = "QcstCrgAcUndo";
                              break;
    case QcstCrgAcEndNode:   code = "QcstCrgAcEndNode";
                              break;
    case QcstCrgAcAddDevEnt: code = "QcstCrgAcAddDevEnt";
                              break;
    case QcstCrgAcRmvDevEnt: code = "QcstCrgAcRmvDevEnt";
                              break;
    case QcstCrgAcChgDevEnt: code = "QcstCrgAcChgDevEnt";
                              break;
    case QcstCrgAcChgNodeStatus: code = "QcstCrgAcChgNodeStatus";
                              break;
    case QcstCrgAcCancelFailover: code = "QcstCrgAcCancelFailover";
                              break;
    case QcstCrgAcVerificationPhase: code =
"QcstCrgAcVerificationPhase";
                              break;
    default:                  code = "unknown action code";
                              break;
}
printf("%s", code);

```

```

    return;
} /* end printActionCode */

/*****/
/*
/* Print the CRG status.
/*
/*
/*****/
static void printCrgStatus(int status) {

    char * str;
    switch (status) {
        case QcstCrgActive:          str = "QcstCrgActive";
                                    break;
        case QcstCrgInactive:        str= "QcstCrgInactive";
                                    break;
        case QcstCrgIndoubt:         str = "QcstCrgIndoubt";
                                    break;
        case QcstCrgRestored:        str = "QcstCrgRestored";
                                    break;
        case QcstCrgAddnodePending:  str =
"QcstCrgAddnodePending";
                                    break;
        case QcstCrgDeletePending:   str = "QcstCrgDeletePending";
                                    break;
        case QcstCrgChangePending:   str = "QcstCrgChangePending";
                                    break;
        case QcstCrgEndCrgPending:   str = "QcstCrgEndCrgPending";
                                    break;
        case QcstCrgInitializePending: str =
"QcstCrgInitializePending";
                                    break;
        case QcstCrgRemovenodePending: str =
"QcstCrgRemovenodePending";
                                    break;
        case QcstCrgStartCrgPending: str =
"QcstCrgStartCrgPending";
                                    break;
        case QcstCrgSwitchOverPending: str =
"QcstCrgSwitchOverPending";
                                    break;
        case QcstCrgDeleteCmdPending: str =
"QcstCrgDeleteCmdPending";
                                    break;
        case QcstCrgAddDevEntPending: str =
"QcstCrgAddDevEntPending";
                                    break;
        case QcstCrgRmvDevEntPending: str =
"QcstCrgRmvDevEntPending";
                                    break;
        case QcstCrgChgDevEntPending: str =
"QcstCrgChgDevEntPending";
                                    break;
        case QcstCrgChgNodeStatusPending: str =
"QcstCrgChgNodeStatusPending";
                                    break;
        default: str = "unknown CRG status";
    }
    printf("%s \n", str);

    return;
} /* end printCrgStatus */

/*****/

```

```

/*                                                                    */
/* Print the recovery domain.                                          */
/*                                                                    */
/*****
static void printRcvyDomain(char *str,
                           unsigned int count,
                           Qcst_Rcvy_Domain_Array1_t *rd) {

    unsigned int i;
    printf("\n %s \n", str);
    for (i=1; i<=count; i++) {
        printStr("    Node_ID = ", rd->Node_ID,
sizeof(Qcst_Node_Id_t));
        printf("%s %d \n", "    Node_Role = ", rd->Node_Role);
        printf("%s", "    Membership_Status = ");
        switch (rd->Membership_Status) {
            case 0: str = "Active";
                break;
            case 1: str = "Inactive";
                break;
            case 2: str = "Partition";
                break;
            default: str = "unknown node status";
        }
        printf("%s \n", str);
        rd++;
    }
    return;
} /* end printRcvyDomain                                             */

/*****
/*                                                                    */
/* Concatenate a null terminated string and a non null terminated string */
/* and print it.                                                      */
/*                                                                    */
/*****
static void printStr(char *s1, char *s2, unsigned int len) {

    char buffer[132];
    memset(buffer, 0x00, sizeof(buffer));
    memcpy(buffer, s1, strlen(s1));
    strcat(buffer, s2, len);
    printf("%s \n", buffer);
    return;
} /* end printStr                                                  */

```

叢集規劃

- 1 在您的 iSeries 伺服器上設定叢集之前，了解您需要執行的動作。了解叢集的先決條件及設計叢集的提示。 最後，閱讀網路設定秘訣和叢集的一些效能提示。

本主題涵蓋使用叢集之前的需求。下列主題提供設計叢集解決方案的一般概念、需求及注意事項。

配置與管理叢集的解決方案

叢集資源服務提供基本叢集基礎架構。有數種方法可讓您利用叢集資源服務所提供的形成叢集功能。

iSeries 上的 i5/OS 叢集資源服務提供可讓您使用叢集的基礎架構。叢集資源服務提供一組整合服務，用來維護叢集拓樸、執行活動訊息，以及建立與管理叢集配置和叢集資源群組。叢集資源服務也提供錯誤控制傳訊功能，可追蹤叢集中每一個節點並確保所有節點的叢集資源狀態資訊一致。

在叢集資源服務提供基本叢集架構的同時，有一些方法可讓您使用這些形成叢集功能。每一個方法都有不同的優點及功能。

重要: 僅單獨使用其中一個解決方案。如果您嘗試使用多個解決方案來建立及管理叢集，可能會產生衝突、問題及無法預測的情況。您在「iSeries 資訊中心」中找到的資訊，記載「iSeries 領航員」與叢集資源服務 CL 指令及 API 特定的程序。如果您使用叢集中介軟體「IBM 商業夥伴」解決方案，請參閱產品附帶提供的文件來取得執行作業的程序資訊。

「iSeries 領航員」叢集管理

IBM 提供的叢集管理介面可透過「iSeries 領航員」使用，並透過「選項 41 (i5/OS - HA 可切換資源)」存取。

- | 此介面可讓您建立及管理使用可切換的獨立磁碟儲存區 (可切換的獨立 ASP) 之叢集，以確定資料的可用性。
- | 它還可以讓您建立及管理叢集、CRG、叢集管理網域及資源。
- | **重要:** 「iSeries 領航員」叢集管理介面不包含叢集資源服務提供的所有功能。當「iSeries 領航員」提供許多必要的功能來配置及管理叢集時，請注意有部分功能僅可透過叢集指令及 API，或可能透過叢集中介軟體「IBM 商業夥伴」應用程式來使用，視特定的應用程式而定。例如，iSeries 叢集架構最多支援叢集中的 128 個節點，但是「iSeries 領航員」介面最多僅支援叢集中的 4 個節點。「iSeries 領航員」叢集管理擁有精靈的特性，可逐步引導您建立簡式且具有四個節點的叢集。如果您的叢集需要超過此數量，您應該考慮使用 IBM 叢集指令及 API 或叢集中介軟體「IBM 商業夥伴」產品。
- | 您還可以使用「iSeries 領航員」執行其他叢集相關的作業。部分作業包括：
 - 將節點新增到現有的叢集
 - 將切換式裝置新增至叢集
 - 將切換式應用程式新增至叢集
 - 將可切換的資料群組新增到叢集
 - 變更回復網域中的節點角色
 - 變更叢集說明
 - 刪除叢集
 - 啟動形成叢集
 - 停止形成叢集
 - 檢視關於叢集活動的訊息
- | 建立叢集管理網域
- | 新增受監視資源項目
- | 如需「iSeries 領航員」中可用之叢集相關作業的完整清單，請參閱叢集的線上說明。

註: 「iSeries 領航員」叢集管理介面不支援邏輯物件抄寫。若要抄寫，您應考慮使用高可用性「IBM 商業夥伴」的叢集產品。

相關概念

iSeries 領航員

第 72 頁的『叢集指令及 API』

i5/OS 叢集資源服務提供一組控制語言 (CL) 指令、應用程式介面 (API) 及機能，供 iSeries 應用程式提供者或客戶增強其應用程式可用性。

第 76 頁的『叢集中介軟體「IBM 商業夥伴」與可用的叢集產品』

您可以向 IBM 叢集中介軟體「IBM 商業夥伴」，購買提供形成叢集所需的邏輯抄寫功能及簡化叢集建立與管理的產品。

第 127 頁的『一般叢集問題』

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

相關參考

第 135 頁的『「iSeries 領航員」叢集管理常見問題集』

有關用於建立及管理叢集之「iSeries 領航員」圖形式使用者介面的問題和回答。

叢集指令及 API

i5/OS 叢集資源服務提供一組控制語言 (CL) 指令、應用程式介面 (API) 及機能，供 iSeries 應用程式提供者或客戶增強其應用程式可用性。

您可以利用叢集控制語言 (CL) 指令及應用程式設計介面 (API)，來撰寫您自訂的應用程式，以配置並管理您的叢集。這些指令與 API 會利用當作 i5/OS 的一部分來提供的叢集資源服務所提供的技術。

QUSRTOOL

叢集資源服務也在 QUSRTOOL 檔案庫中提供一組指令範例，對映到沒有支援指令介面的 API。在部份環境中，QUSRTOOL 指令可能會非常有用。例如，您可以變更活動訊號，或將資訊傳送到叢集各處。如需這些範例指令的詳細資訊，請參閱檔案 QUSRTOOL/QATTINFO 中的成員 TCSTINFO。應用程式 CRG 跳出程式的範例也包括在 QUSRTOOL 檔案庫中。此原始程式碼範例可用來作為撰寫跳出程式的基礎。檔案 QATTSYSC 中的來源檔範例 TCSTDTAEXT 包含了一個來源檔，供程式建立 QCSTHAAPPI 和 QCSTHAAPP0 資料區，以及 QACSTOSDS (物件指定元) 檔案。

相關工作

第 95 頁的『將節點新增到叢集中』

您可以使用「iSeries 領航員」或指令，新增節點至叢集。

叢集 CL 指令及 API 說明:

有許多 API 及 CL 指令可用於配置、啟動及管理叢集、叢集節點及叢集資源群組。

下列表格顯示叢集控制的名稱與簡短說明，以及可用的叢集資源群組控制語言 (CL) 指令及 API。只能在 OS/400® V5R2M0 或更新版本上使用叢集 CL 指令。

- 1 第一個表格包含用於在叢集中配置、啟動及管理叢集及節點的指令及 API。第二個表格包含用於在叢集中配置、
- 1 啟動及管理叢集資源群組的指令及 API。第三個表格包含用於配置及管理叢集管理網域的指令。第四個表格包
- 1 含可用於新增及移除叢集管理網域中的受監視資源項目之「整合作業系統 API」的說明。

表 8. 叢集控制 CL 指令及 API 說明

說明	叢集控制 CL 指令	叢集控制 API 名稱
新增叢集節點項目 將節點新增到現有叢集的成員清單中。亦指派叢集通訊要使用的 IP 介面位址。	ADDCLUNODE	新增叢集節點項目 (QcstAddClusterNodeEntry)
新增裝置網域項目 將節點新增到裝置網域成員清單中，使它可參與回復裝置的回復動作。第一個新增到裝置網域的節點有建立該裝置網域的效果。	ADDDEVDMNE	新增裝置網域項目 (QcstAddDeviceDomainEntry)
調整叢集版本，變更叢集版本 例如，將現行叢集版本調整為下一個版本，使得可在叢集內使用新功能。	CHGCLUVER	調整叢集版本 (QcstAdjustClusterVersion)

表 8. 叢集控制 CL 指令及 API 說明 (繼續)

說明	叢集控制 CL 指令	叢集控制 API 名稱
變更叢集節點項目 變更叢集節點項目中的欄位。例如，可變更使用於叢集通訊的 IP 介面位址。	CHGCLUNODE	變更叢集節點項目 (QcstChangeClusterNodeEntry)
變更叢集資源服務，變更叢集配置 調整叢集效能及配置調整參數，以符合使用於通訊的網路之通訊環境。	CHGCLUCFG	變更叢集資源服務 (QcstChgClusterResourceServices)
建立叢集 建立一或多個節點的新叢集。	CRTCLU	建立叢集 (QcstCreateCluster)
刪除叢集 刪除現有的叢集。叢集資源服務在所有作用中叢集節點上結束，並從叢集移除。	DLTCLU	刪除叢集 (QcstDeleteCluster)
結束叢集節點 在現有的叢集之成員清單中的一或所有節點上，結束叢集資源服務。叢集將無法使用該節點，直到使用「啓動叢集節點」介面來重新啓動它為止。	ENDCLUNOD	結束叢集節點 (QcstEndClusterNode)
列出叢集資訊，顯示叢集資訊 擷取關於叢集的資訊。例如，可傳回完整的叢集成員清單。	DSPCLUINF	列出叢集資訊 (QcstListClusterInfo)
列出裝置網域資訊，顯示叢集資訊 列出叢集的裝置網域資訊。例如，可傳回目前定義的裝置網域之清單。	DSPCLUINF	列出裝置網域資訊 (QcstListDeviceDomainInfo)
移除叢集節點項目 從叢集的成員清單移除節點。從任何回復網域移除節點，在節點上結束叢集作業，並從節點刪除所有叢集資源服務物件。	RMVCLUNODE	移除叢集節點項目 (QcstRemoveClusterNodeEntry)
移除裝置網域項目 從裝置網域成員清單移除節點。若這是裝置網域中的最後一個節點，則這也會有從叢集刪除裝置網域的效果。	RMVDEVDMNE	移除裝置網域項目 (QcstRemoveDeviceDomainEntry)
擷取叢集資訊，顯示叢集資訊 擷取關於叢集的資訊。例如，傳回叢集名稱及目前的叢集版本。	DSPCLUINF	擷取叢集資訊 (QcstRetrieveClusterInfo)
擷取叢集資源服務資訊，顯示叢集資訊 擷取關於叢集資源服務效能調整及配置參數的資訊。	DSPCLUINF	擷取叢集資源服務資訊 (QcstRetrieveCRSInfo)
啓動叢集節點 在屬於叢集之一部份，但目前非作用中的節點上，啓動叢集資源服務。	STRCLUNOD	啓動叢集節點 (QcstStartClusterNode)

表 8. 叢集控制 CL 指令及 API 說明 (繼續)

說明	叢集控制 CL 指令	叢集控制 API 名稱
使用叢集 顯示及使用叢集節點及物件。	WRKCLU	無

表 9. 叢集資源群組 CL 指令及 API 說明

說明	叢集資源群組 CL 指令	叢集資源群組 API
新增叢集資源群組裝置項目 將新的裝置項目新增到叢集資源群組中。裝置會成為可切換裝置之群組的成員。	ADDCRGDEVE	新增叢集資源群組裝置項目 (QcstAddClusterResourceGroupDevice)
新增節點至回復網域，新增叢集資源群組節點項目 將新的節點新增到現有的叢集資源群組之回復網域中。	ADDCRGNODE	新增節點至回復網域 (QcstAddNodeToRcvyDomain)
變更叢集資源群組 變更叢集資源群組的屬性。例如，可修改應用程式 CRG 的接管 IP 位址。	CHGCRG	變更叢集資源群組 (QcstChangeClusterResourceGroup)
變更叢集資源群組裝置項目 變更叢集資源群組中的裝置項目。例如，可修改切換或錯誤後移轉中，用來線上改變配置物件的選項。	CHGCRGDEVE	變更叢集資源群組裝置項目 (QcstChangeClusterResourceGroupDev)
建立叢集資源群組 建立叢集資源群組物件。叢集資源群組物件會定義回復網域，它是將在回復中扮演角色的叢集中之節點集。	CRTCRG	建立叢集資源群組 (QcstCreateClusterResourceGroup)
刪除叢集資源群組 僅刪除本端節點上的叢集資源群組 (CRG)。刪除本端叢集資源群組需要「叢集資源服務」為非作用中。	DLTCRG	無
刪除叢集資源群組，刪除 CRG 叢集 從叢集刪除叢集資源群組。將從回復網域中的所有作用中節點刪除 CRG 物件。	DLTCRGCLU	刪除叢集資源群組 (QcstDeleteClusterResourceGroup)
分送資訊 從 CRG 的回復網域中之節點，將資訊分送到該 CRG 之回復網域中的其他節點。	無	分送資訊 (QcstDistributeInformation)
結束叢集資源群組 停用指定的叢集資源群組之回復。在此 API 順利完成時，叢集資源群組狀態會設定為非作用中。	ENDCRG	結束叢集資源群組 (QcstEndClusterResourceGroup)

表 9. 叢集資源群組 CL 指令及 API 說明 (繼續)

說明	叢集資源群組 CL 指令	叢集資源群組 API
起始切換，變更叢集資源群組主要節點 引起叢集資源群組執行管理切換。會變更回復網域，如此目前的主節點會變成最後的備份節點，而目前的第一個備份節點會變成新的主節點。	CHGCRGPRI	起始切換 (QcstInitiateSwitchover)
列出叢集資源群組，顯示叢集資源群組資訊 產生叢集資源群組的清單，及關於叢集中之叢集資源群組的一些資訊。	DSPCRGINF	列出叢集資源群組 (QcstListClusterResourceGroups)
列出叢集資源群組資訊，顯示叢集資源群組資訊 傳回叢集資源群組物件的內容。例如，可傳回回復網域及目前的節點角色。	DSPCRGINF	列出叢集資源群組資訊 (QcstListClusterResourceGroupInf)
移除叢集資源群組裝置項目 從叢集資源群組移除裝置項目。此裝置將不再是可切換的資源。	RMVCRGDEVE	移除叢集資源群組裝置項目 (QcstRemoveClusterResourceGroupDev)
移除回復網域中的節點，移除叢集資源群組節點項目 從現有的叢集資源群組之回復網域移除節點。節點將不再參與該資源群組的回復動作。	RMVCRGNODE	從回復網域中移除節點 (QcstRemoveNodeFromRcvyDomain)
啓動叢集資源群組 啓用指定的叢集資源群組之回復。叢集內的叢集資源群組會變成作用中。	STRCRG	啓動叢集資源群組 (QcstStartClusterResourceGroup)

註：叢集資源服務也在 QUSRTOOL 檔案庫中提供一組指令範例，與上述的 CL 指令與 API 對映。在部份環境中，QUSRTOOL 指令可能會非常有用。例如，其中一指令可輕易地設定叢集，以供測試叢集啓用的應用程式。如需這些範例指令的詳細資訊，請參閱檔案 QUSRTOOL/QATTINFO 中的成員 TCSTINFO。

表 10. 管理網域 CL 指令說明

說明	管理網域 CL 指令	管理網域 API
建立管理網域 建立代表叢集管理網域的對等節點 CRG。一旦建立叢集管理網域，就可將受監視資源項目 (MRE) 新增至網域，以同步化資源變更。 註：一旦建立叢集管理網域，就可使用第 74 頁的表 9 中的 CRG 指令來管理它。	CRTADMDMN	無

表 10. 管理網域 CL 指令說明 (繼續)

說明	管理網域 CL 指令	管理網域 API
刪除管理網域 刪除代表叢集管理網域的對等節點 CRG。一旦完成，就會從網域中移除所有 MRE，且不會再傳送受監視資源的變更。	DLTADMDMN	無

表 11. 整合作業系統 API 說明. 除了這些叢集管理網域 CL 指令以外，還有數個「整合作業系統」應用程式可程式化介面 (API) 可用於新增及移除受監視資源項目。

說明	CL 指令 ¹	整合作業系統 API
新增受監視資源項目 為系統資源及其屬性新增受監視資源項目。	無	新增受監視資源項目 (QfpadAddMonitoredResourceEntry)
移除受監視資源項目 從受監視資源目錄中移除受監視資源項目 (MRE)。	無	移除受監視資源項目 (QfpadRmvMonitoredResourceEntry)
擷取受監視資源資訊 傳回受監視資源的相關資訊。	無	擷取受監視資源資訊 (QfpadRtvMonitoredResourceInfo)

註:

1. 沒有相等於此功能的受支援 CL 指令。QUSRTOOL 檔案庫中提供了不受支援之指令的來源及呼叫處理程式 (CPP)。若要了解此指令來源及 CPP，請查看檔案 QATTINFO 中的成員 QFPADINFO。

相關參考

叢集 API

叢集中介軟體「IBM 商業夥伴」與可用的叢集產品

您可以向 IBM 叢集中介軟體「IBM 商業夥伴」，購買提供形成叢集所需的邏輯抄寫功能及簡化叢集建立與管理的產品。

IBM 叢集中介軟體「IBM 商業夥伴」提供專用抄寫及叢集管理功能的軟體解決方案。如果您要購買提供形成叢集所需的邏輯抄寫功能及簡化叢集建立與管理的產品，請聯絡 IBM 業務代表或商業夥伴。他們可提供 IBM 叢集中介軟體「IBM 商業夥伴」所提供之啓用叢集產品的完整清單。

叢集中介軟體「IBM 商業夥伴」叢集管理產品：

- 提供使用者介面來定義及維護叢集配置。
- 提供使用者介面來定義及管理裝置、資料和應用程式叢集資源群組。
- 透過使用叢集 API 來了解叢集中定義的叢集資源群組及必要的關係。
- 建立裝置、資料及應用程式叢集資源群組。

叢集中介軟體「IBM 商業夥伴」抄寫產品：

- 建置中介軟體的控制結構，以識別具回復所需的資料及物件。
- 建立重要資料的叢集資源群組並使該物件與其控制結構產生關聯。

- 提供跳出程式給資料叢集資源群組。

相關工作

第 95 頁的『將節點新增到叢集中』

您可以使用「iSeries 領航員」或指令，新增節點至叢集。

叢集的基本要求

說明使用叢集的硬體、軟體及通訊基本要求。

使用叢集的基本要求會根據您選擇使用的叢集功能而有所不同。例如，您可以選擇使用簡式雙節點叢集來使用邏輯抄寫。或者，您也可以選擇使用設計成利用切換式磁碟及切換式獨立磁碟儲存區的叢集。

相關概念

第 111 頁的『範例：叢集配置』

使用這些典型的叢集實作範例來了解何時、為何及如何使用叢集最有利。

叢集的硬體基本要求

任何能夠執行 i5/OS V4R4M0 或更新版本的 iSeries 機型，在使用叢集上都是相容的。

此外，您應該透過外部不斷電系統或同等設備來提供斷電保護。否則，叢集節點發生意外斷電時，會導致叢集分割狀態，而非錯誤後移轉。

形成叢集作業使用「網際網路通訊協定 (IP)」多點傳播功能。多點傳播並不會完全對映至所有實體媒體類型。如需可套用至特定硬體之多點傳播限制的詳細資訊，請參閱 TCP/IP 設定。

如果您計劃在叢集中使用獨立磁碟儲存區，請參閱獨立磁碟儲存區的硬體基本要求主題。您也可以使用鏡映保護或裝置同位檢查保護來保護您的磁碟。在主系統上使用這些解決方案可防止因為受保護的磁碟失效而發生的錯誤後移轉。萬一發生錯誤後移轉，則在您的備份系統上使用這些解決方案也是一個好方法。如需詳細資訊，請參閱「磁碟保護」。

- | **註：**如需配置叢集之前所需之其他基本要求的詳細資訊，請參閱第 87 頁的『叢集配置核對清單』。

相關概念

不斷電系統

第 26 頁的『叢集分割區』

叢集分割區是導因於通訊失敗之作用中叢集節點的子集。分割區中的成員會保持彼此之間的連通性。

第 16 頁的『錯誤後移轉』

當叢集中的伺服器在系統失效時自動切換為一或多個備份伺服器時，會發生錯誤後移轉。

叢集的軟體與授權基本要求

為了使用叢集，您必須具有正確的軟體與授權。

1. 配置了 TCP/IP (TCP/IP 連通性公用程式) 的 OS/400 V4R4M0 或更新版本。
2. 叢集配置與管理軟體解決方案。這可以是下列任何一項：
 - 「iSeries 領航員」叢集管理
 - 叢集中介軟體「IBM 商業夥伴」解決方案
 - 您使用叢集資源服務指令及 API 撰寫的專屬叢集管理應用程式
- | 3. 請參閱第 87 頁的『叢集配置核對清單』

重要：如果您計劃使用獨立磁碟儲存區來利用切換式裝置，則有額外的基本要求。如需詳細資訊，請參閱獨立磁碟儲存區規劃。

相關概念

第 70 頁的『配置與管理叢集的解決方案』

叢集資源服務提供基本叢集基礎架構。有數種方法可讓您利用叢集資源服務所提供的形成叢集功能。

第 12 頁的『叢集版本』

叢集版本代表叢集上可用功能的層次。

叢集的通訊需求

您可以在叢集環境中使用任何通訊媒體類型，只要它支援「網際網路通訊協定 (IP)」即可。

叢集資源服務只使用 TCP/IP 通訊協定在節點之間通訊。支援區域網路 (LAN)、廣域網路 (WAN)、OptiConnect 系統區域網路 (SAN) 或這些連通性裝置的任何組合。您的選擇應基於下列因素：

- 交易量
- 回應時間需求
- 節點距離
- 成本考量

在決定使用連線媒體來連接主要和備份資源位置時，您可以使用這些相同的注意事項。在規劃叢集時，建議您在遠端位置指定一或多個備份節點，以便在發生網站損失災害時可以挽救。

為了避免容量不足而導致的效能問題，您需要評估用來處理節點之間傳送資訊量的通訊媒體。您可以選擇偏好使用的實體媒體，例如記號環、乙太網路、非同步轉送模式 (ATM)、SPD OptiConnect、高速線路鏈接 (HSL) OptiConnect 或虛擬 OptiConnect (一種邏輯分割區之間的高速內部連線)。

HSL OptiConnect 是 OptiConnect for i5/OS 軟體 (i5/OS 選項 23 - i5/OS OptiConnect) 提供的技術。它可用來建構高可用性的解決方案。HSL OptiConnect 是一種系統區域網路，利用「高速線路鏈接」(HSL) 迴路技術，提供叢集節點之間高速的點對點連通性。HSL OptiConnect 需要標準 HSL 電纜，但不需要額外的硬體。

關於可切換硬體，亦稱為回復裝置 CRG，您的環境中需要一個可切換獨立磁碟儲存區。在邏輯分割區環境中，這是指邏輯分割區共用之匯流排上的硬碟機集合，或是連接至已指定給 I/O 儲存區之輸入/輸出處理器的硬碟機集合。在多重系統環境中，這是指 HSL 迴路上適當配置的一或多個可切換擴充裝置 (直立式主機)，迴路中也包含回復網域中的系統。可切換直立式主機也可以使用於 LPAR 環境中。如需切換式硬體及獨立磁碟儲存區的詳細規劃資訊，請參閱獨立磁碟儲存區規劃。

註: 如果您僅透過 TCP/IP 使用 2810 LAN 配接卡，而不使用「系統網路架構 (SNA)」或 IPX，則可透過使用線路說明 (WRKLIND) 指令，為特定線路說明指定「僅對 TCP 啟用 (*YES)」，以增加 V4R5M0 伺服器上的配接卡效能。V5R1M0 與以上的版本會自動設定「僅對 TCP 啟用 (*YES)」。

相關資訊

OptiConnect for i5/OS

設計叢集

識別您的需求來決定如何設計叢集。

由於根據您所希望達到的目標，叢集的使用方式有很多種，所以您務必要花點時間來了解自己的需求，以決定如何設計您的叢集。

設計您的叢集網路

在配置用於形成叢集的網路之前，您必須小心規劃及執行一些涉及 TCP/IP 的起始預設叢集配置。

在配置叢集之前，請務必閱讀這些主題。這些主題提供您執行下列事項的時機或方法：

- 設定 IP 位址
- 設定 TCP/IP 配置屬性
- 避免叢集分割

如需設定備用通訊路徑及是否需要專用網路來形成叢集的詳細資訊，請參閱「叢集專用網路」。

設定 IP 位址:

由於叢集資源服務僅使用 IP 來與其它叢集節點通訊，因此所有叢集節點必須可抵達 IP。

這表示您必須配置 IP 介面來連接您叢集中的節點。這些 IP 位址必須由每個叢集節點上 TCP/IP 遞送表中的網路管理者手動設定，或由網路中路由器上執行的遞送通訊協定產生。這個 TCP/IP 遞送表是形成叢集用來尋找每個節點的對映；因此，每個節點必須有自己的唯一 IP 位址。每一個節點最多可指定兩個 IP 位址。這些位址不可以被其它網路通訊應用程式以任何方式變更。在指定每一個位址時，必須考慮到哪一個位址使用哪一種通訊線路。如果您偏好使用特定類型的通訊媒體，請務必使用您偏好的媒體來配置第一個 IP 位址。第一個 IP 位址就是可靠的訊息功能及活動訊號監視優先處理的 IP 位址。節點上的所有 IP 位址都必須能夠連上叢集中的所有其他 IP 位址。如果可以連通測試 (ping) 並雙向使用 UDP 訊息追蹤路徑，一個位址就能連上另一個位址。

註: 您必須確定形成叢集的迴路位址 (127.0.0.1) 是在作用中。根據預設值，此位址通常是作用中，它用於將任何訊息傳回本端節點。不過，如果此位址錯誤結束，則除非重新啟動此位址，否則叢集傳訊無法運作。

相關概念

第 25 頁的『可靠的訊息功能』

叢集資源服務的可靠的訊息功能會記錄叢集中之每一個節點，並確定所有節點都有關於叢集資源狀態的一致資訊。

第 23 頁的『活動訊號監督』

活動訊號監視是一種叢集資源服務功能，它可將信號從叢集中的每個節點，傳送到叢集中的所有其他節點，通知它們仍處於作用中，以確定每個節點都在作用中。

設定 TCP/IP 配置屬性:

若要啟用叢集資源服務，則在您網路的 TCP/IP 配置中需要一些屬性設定。

在新增任何節點至叢集之前，您必須設定這些屬性：

- 如果您計劃使用 iSeries 伺服器作為路由器與其他網路通訊，而且您沒有在該伺服器上執行其他遞送通訊協定，請使用 CHGTCPA (變更 TCP/IP 屬性) 指令，將 IP 資料包轉遞設為 *YES。
- 將 INETD 伺服器設為 START。如需啟動 INETD 伺服器的詳細資訊，請參閱 INETD 伺服器。
- 使用 CHGTCPA (變更 TCP/IP 屬性) 指令，將「使用者資料包通訊協定 (UDP)」CHECKSUM 設為 *YES。
- 如果您使用橋接器來連接記號環網路，請將 MCAST 轉遞設為 *YES。
- 如果您使用 OptiConnect for i5/OS 在叢集節點之間通訊，請指定 STRSBS(QSOC/QSOC) 來啟動 QSOC 子系統。

叢集通訊要訣:

設定通訊路徑時，請考量下列要訣。

- 請確定您的通訊線路上有足夠的頻寬，以處理非叢集活動，以及形成叢集活動訊號功能，並繼續監督增加的活動。
- 爲了最佳可靠性，請不要配置連結一或多個節點的單一通訊路徑。
- 請不要使負責確定您仍與節點通訊之線路負荷過重。
- 儘量除去失敗的單一點，例如讓兩條通訊線路進入單一配接卡、相同的輸入-輸出處理器或相同的直立式主機。
- 如果有極高的資料容量透過您的通訊線路來傳送，您可能要考慮將資料抄寫及活動訊號監視置於個別的網路上。
- 如果您正在使用「網際網路通訊協定 (IP)」多點傳播，您應該參閱 TCP/IP Configuration and Reference，以取得可套用至不同實體媒體類型的多點傳播限制。
- 「使用者資料包通訊協定 (UDP)」多點傳播是喜好的通訊協定，叢集通訊基礎結構用它來傳送叢集中的節點之間的叢集管理資訊。當實體媒體支援多點傳播功能時，叢集通訊會利用 UDP 多點傳播，將管理訊息從給定的節點傳送到支援相同子網路位址的所有區域叢集節點。傳送到遠端網路上之節點的訊息，一律使用 UDP 點對點功能來傳送。叢集通訊不依賴多點傳播訊息的遞送功能。
- 支援叢集管理訊息的多點傳播傳輸趨向於自然變動。根據給定的 LAN (支援共用子網路位址) 上的節點數，以及叢集管理者所選擇的叢集管理結構之複雜程度，叢集相關的多點傳播封包每秒可輕易地超過 40 個封包。這種自然變動會對較舊的網路設備造成負面影響。LAN 之裝置上的壅塞問題就是一個例子，這些裝置是作爲「簡易網路管理通訊協定 (SNMP)」代理程式，需要評估每一個 UDP 多點傳播封包。部份較早的網路設備沒有足夠的頻寬可跟得上這類傳輸。您需要確定您或網路管理者已複查要處理 UDP 多點傳播傳輸的網路容量，確保叢集沒有對網路效能造成負面影響。

相關概念

第 84 頁的『邏輯抄寫規劃』

使用邏輯抄寫維護資料的多份副本。從叢集中主節點抄寫或複製資料到回復網域中指定的備份節點。當主節點發生故障時，指定的備份節點接管爲主要存取點，此時資料仍然可用。

第 25 頁的『可靠的訊息功能』

叢集資源服務的可靠的訊息功能會記錄叢集中之每一個節點，並確定所有節點都有關於叢集資源狀態的一致資訊。

相關資訊

TCP/IP 設定

避免叢集分割:

在叢集的所有節點之間配置備用通訊路徑，最能有效的避免一般網路相關的叢集分割。

備用通訊路徑表示在叢集中的兩個節點之間配置兩條路線。如果第一條通訊路徑發生失敗，第二條通訊路徑可以接管來維持節點之間的通訊，藉此，將叢集中的一或多個節點置於一個叢集分割區的情況能減至最少。配置這些路徑時要考慮的一件事，就是如果您的兩條通訊線路連接到系統上的相同配接卡，一旦這個單一配接卡故障，這些線路仍然有失效的風險。

- 1 您應該瞭解並不能完全避免叢集分割。如果您的系統斷電，或發生硬體故障，還是會發生叢集分割。

相關概念

第 26 頁的『叢集分割區』

叢集分割區是導因於通訊失敗之作用中叢集節點的子集。分割區中的成員會保持彼此之間的連通性。

第 79 頁的『叢集通訊要訣』

設定通訊路徑時，請考量下列要訣。

第 129 頁的『分割區錯誤』

某些叢集狀況很容易解決。如果發生叢集分割，您可以學習如何回復。此主題也告訴您如何避免叢集分割並舉例說明如何把分割區合併在一起。

叢集專用網路:

在正常作業期間，基本叢集通訊流量最小。不過，強烈建議您對叢集中每個節點配置備用通訊路徑。

藉由配置兩條線路，您可以提供一條線路專供形成叢集傳輸使用，另一條線路可處理一般傳輸，如果用於形成叢集的專用線路失效，另一條線路也可以成為備用線路。

相關概念

第 80 頁的『避免叢集分割』

在叢集的所有節點之間配置備用通訊路徑，最能有效的避免一般網路相關的叢集分割。

多版次叢集

如果建立叢集時將併入多重叢集版本的節點，則在建立叢集時需要某些步驟。

根據預設值，現行叢集版本會設為加入叢集中第一個節點的潛在叢集版本。如果此節點位於叢集中最低版本層次，則這是一個適當的方式。不過，如果此節點位於較新的版本層次，則以後您將無法新增具有較低版本層次的節點。替代方案是使用建立叢集的目標叢集版本值，將現行叢集版本設為小於加入叢集中第一個節點之潛在叢集版本的版本。

例如，以要建立雙節點叢集為例。此叢集的節點如下：

節點 ID	版次	潛在叢集版本
節點 A	V5R3	4
節點 B	V5R4	5

如果要從「節點 B」建立叢集，則必須謹慎指出此叢集將是混合版次叢集。必須設定目標叢集版本，來指示叢集節點將在小於要求節點的潛在節點版本之版本上通訊。

相關概念

第 12 頁的『叢集版本』

叢集版本代表叢集上可用功能的層次。

識別要併入叢集中的伺服器

為識別您要併入叢集中的伺服器，您必須決定哪些伺服器能夠對您執行業務所需的資料及應用程式提供足夠的備份。

請做下列判定：

- 哪些伺服器含有您的重要資料及應用程式？
- 哪些伺服器是那些系統的備份？

一旦確定之後，這些伺服器就是您要併入叢集中的伺服器。

比較主要備份模型與對等節點模型

主要備份 CRG 與對等節點 CRG 都能為叢集內的資源提供回復；不過，了解這兩者之間的差異及用法是很重要的。

叢集支援兩種模型來定義環境中的 CRG。主要備份及對等節點模型都會定義角色。在主要備份模型中，需要為角色定義次序。節點失敗時，定義為備份節點的節點便能存取主要節點上的資源。在對等節點模型中，每個節點的角色都相同，並可提供對資源的存取，但卻沒有次序的概念。

主要備份模型

在主要備份模型中，使用者必須將節點定義為主要、備份或複製角色。這些角色可在回復網域中定義及管理。如果節點已定義為資源的主要存取點，則在主要節點失敗時，其他節點便能提供備份。

對等節點模型

對等節點模型 CRG 不需定義有次序的回復網域。若為對等節點模型，可將節點定義為對等節點或複製節點。如果將節點定義為對等節點，則回復網域中的所有節點都相等，並可提供資源的存取點。

識別要併入叢集中的應用程式

並非每一個應用程式都有提供形成叢集的可用性優點。

應用程式必須具有回復才能使用形成叢集所提供的切換及錯誤後移轉功能。應用程式回復可讓應用程式在備份節點上重新啟動，而不需要使用應用程式來重新配置用戶端。因此，您的應用程式必須符合某些基本要求才能完全使用到形成叢集所提供的功能。如需回復應用程式的詳細資訊，請參閱叢集應用程式主題。

資料回復規劃

當使用者或應用程式可隨時取得資料時，就算是達到資料回復了。您可以經由使用邏輯抄寫或切換式獨立磁碟儲存區來達到資料回復。

決定哪些資料應具有回復：

瞭解應考慮哪些資料類型具有回復。

決定哪些資料需要有回復，就像決定您為系統準備備份及回復策略時需要備份及儲存哪種資料一樣。您必須判斷環境中哪些資料對於維持企業營運很重要。

例如，如果您在網路上執行業務，以下是您的重要資料：

- 當日訂單
- 庫存
- 客戶記錄

一般而言，不經常變更或不需要每天使用的資訊可能就不必有回復。

相關概念

規劃備份及回復策略

邏輯抄寫、切換式磁碟與跨站台鏡映的比較：

本主題提供不同資料回復技術的概觀，這些技術可與叢集搭配使用以增強高可用性。

資料回復可讓資料即使在原始裝載資料的系統失敗時，仍可供應用程式及使用者使用。在您的整體商業連續性策略環境定義中選擇一組正確的資料回復技術是相當複雜和困難的。了解可用於加強多重系統環境中之可用性的不同資料回復解決方案很重要。您可以選擇單一解決方案，也可以使用這些技術的組合來滿足您的需要。

如需這些解決方案的詳細資訊，請參閱 [Data Resilience Solutions for IBM i5/OS High Availability Clusters](#)。 [Data Resilience Solutions for IBM i5/OS High Availability Clusters](#) 一節中包含每一種技術之屬性的詳細比較。

邏輯抄寫

邏輯抄寫是將物件從叢集中的某個節點複製到叢集中的一或多個其他節點的程序，這可使所有系統上的物件都相同。

抄寫的資源可讓物件 (例如應用程式及其資料) 從叢集中的一個節點，複製到叢集中的一或多個其他節點。這項處理會讓資源的回復網域中之所有伺服器上的物件保持一致。若您對叢集中之一個節點上的物件進行變更，則會將變更抄寫到叢集的其他節點。然後，發生錯誤後移轉切換時，備份節點可無縫地扮演主節點的角色。作為備份的伺服器定義於回復網域。當定義為回復網域中之主節點的伺服器發生故障，且已起始切換或錯誤後移轉時，指定為回復網域中的節點會變成資源的主要存取點。

抄寫需要使用自訂撰寫應用程式，或由叢集中介軟體企業夥伴撰寫的軟體應用程式。如需詳細資訊，請參閱邏輯抄寫規劃。

切換式磁碟

切換式磁碟可讓位於擴充裝置或邏輯分割區之共用匯流排或 I/O 儲存區上的輸入/輸出處理器 (IOP) 上的資源 (如資料及應用程式)，在叢集的主要節點與備份節點之間切換。當目前使用硬碟機的伺服器遇到故障，且發生錯誤後移轉及切換時，會容許從第二個伺服器 (定義為叢集資源群組的回復網域中之備份節點) 存取硬碟機集。

在您的叢集中利用可切換的資源，需要使用獨立磁碟儲存區。如需詳細資訊，請參閱獨立磁碟儲存區規劃。

跨站台鏡映

跨站台鏡映與地理鏡映功能結合，可讓您在站台的磁碟上鏡映資料，而這些站台之間的地理距離可能非常遙遠。可使用此技術延伸裝置叢集資源群組 (CRG) 的功能，可超過實體元件連接的限制。地理鏡映提供的功能可將獨立磁碟儲存區之正式作業副本的變更，抄寫到該獨立磁碟儲存區的鏡映副本。當資料寫入獨立磁碟儲存區的正式作業副本時，作業系統會透過另一個系統，將該資料鏡映到獨立磁碟儲存區的第二個副本。這個處理程序會保留多個相同的資料副本。

透過裝置 CRG 發生錯誤後移轉或切換時，備份節點可以完整地扮演主要節點的角色。作為備份的伺服器定義於回復網域。備份節點可以與主要節點位於相同或不同的實體位置。當定義為回復網域之主要節點的伺服器發生故障，且已起始切換或錯誤後移轉時，指定為回復網域之備份的節點會成為資源的主要存取點，接著會擁有獨立磁碟儲存區的正式作業副本。因此，您可以從與可切換資源相關的單一失敗點取得保護。

表 12. 比較可與叢集搭配使用的資料回復技術。了解不同資料回復技術的性質可幫助您決定叢集的最佳解決方案。

因素	抄寫	切換式磁碟	跨站台鏡映
彈性	10 台伺服器	2 台伺服器	4 台伺服器
單一的失敗點	無	磁碟子系統	無
成本	<ul style="list-style-type: none">需要額外的磁碟容量。商業夥伴抄寫軟體。複製磁碟。	<ul style="list-style-type: none">切換式 I/O 擴充直立式主機或 IOP選項 41	<ul style="list-style-type: none">獨立磁碟儲存區鏡映副本的其他磁碟選擇性地切換 I/O 擴充裝置選項 41
效能	抄寫額外執行時間	少許影響	地理鏡映額外執行時間
即時涵蓋面	記載的物件	併入獨立磁碟儲存區的物件	併入獨立磁碟儲存區的物件
地理上的散佈程度	效能注意事項的限制	當伺服器及擴充裝置必的限制連接距離須連接到 HSL OptiConnect 迴圈時 (最大為 250 公里)	效能考量的限制 (系統沒有強制限制。不過，所選取通訊線路的回應時間與產能可要求一些特定限制。)

表 12. 比較可與叢集搭配使用的資料回復技術 (繼續)。了解不同資料回復技術的性質可幫助您決定叢集的最佳解決方案。

因素	抄寫	切換式磁碟	跨站台鏡映
災難回復保護	是	否	是
並行備份	是	否	否
設定	<ul style="list-style-type: none"> 抄寫環境。 決定要抄寫的內容。 	<ul style="list-style-type: none"> 獨立磁碟儲存區環境。 大量輸入資料至獨立磁碟儲存區。 	<ul style="list-style-type: none"> 獨立磁碟儲存區環境 (包括設定地理鏡映) 大量輸入資料至獨立磁碟儲存區。

相關概念

『邏輯抄寫規劃』

使用邏輯抄寫維護資料的多份副本。從叢集中主節點抄寫或複製資料到回復網域中指定的備份節點。當主節點發生故障時，指定的備份節點接管為主要存取點，此時資料仍然可用。

第 85 頁的『切換式獨立磁碟儲存區與跨站台鏡映 (XSM) 的規劃』

資料的單一拷貝在可切換硬體上維護；指邏輯分割區環境中的擴充裝置 (直立式主機) 或 IOP。

邏輯抄寫規劃:

使用邏輯抄寫維護資料的多份副本。從叢集中主節點抄寫或複製資料到回復網域中指定的備份節點。當主節點發生故障時，指定的備份節點接管為主要存取點，此時資料仍然可用。

抄寫可即時建立某些項目的副本。它是一種將物件從叢集中一個節點複製到該叢集中一或多個其它節點的程序。抄寫可保持您系統上的物件完全相同。如果您對叢集中一個節點上的物件做了某項變更，這個變更會抄寫到該叢集中其它節點上。

您必須決定一種用於邏輯抄寫的軟體技術。下列解決方案可在您的叢集中達到邏輯抄寫目的：

- 「IBM 商業夥伴」產品

來自已驗證之叢集「IBM 商業夥伴」的資料抄寫軟體，可讓您在多重節點之間抄寫物件。如需詳細資訊，請參閱第 76 頁的『叢集中軟體「IBM 商業夥伴」與可用的叢集產品』。

- 自訂撰寫的抄寫應用程式

IBM 異動日誌管理提供一種方法，可讓您記錄系統上的物件活動。您可以利用異動日誌管理撰寫應用程式來達到邏輯抄寫目的。如需異動日誌管理如何運作的詳細資訊，請參閱 iSeries 異動日誌管理。

相關概念

異動日誌管理

判定哪些系統用於邏輯抄寫:

在判定將哪些系統用於邏輯抄寫時，有幾點重要的注意事項。

這些注意事項是：

- 效能產能
- 磁碟產能
- 重要資料
- 災害預防

- | 如果系統失效，您必須知道主系統及備份系統上所執行的資料及應用程式。您想要將重要資料存放在系統失效時最有能力應付工作量的系統上。您不希望磁碟空間用盡。如果您的主系統用盡了空間而失效，很可能您的備份系統也因為磁碟空間不足而失效。為確保您的資料中心不會因為天災（如洪水、颱風或暴風雨）而完全損毀，您應該在遠端位置設置備份系統。

切換式獨立磁碟儲存區與跨站台鏡映 (XSM) 的規劃:

資料的單一拷貝在可切換硬體上維護；指邏輯分割區環境中的擴充裝置 (直立式主機) 或 IOP。

當主節點發生故障時，可切換硬體上的資料存取切換至指定的備份節點。此外，獨立磁碟儲存區可在跨站台鏡映 (XSM) 環境中使用。這可讓獨立磁碟儲存區的鏡映副本，基於可用性或保護目的，在位置遠離原來站台的系統 (選擇性地) 上維護。

如果您計劃使用切換式獨立磁碟儲存區或跨站台鏡映 (XSM) 上的可切換資源，請小心規劃。

相關概念

獨立磁碟儲存區規劃

叢集安全性

考慮當您在系統上規劃使用叢集時所需注意的一些安全性問題。

啓用要新增到叢集的節點

在您可將節點新增到叢集之前，您必須為「允許新增至叢集 (ALWADDCLU)」網路屬性設定一個值。

在您要設定為叢集節點的任何伺服器上，使用「變更網路屬性 (CHGNETA)」指令。「變更網路屬性 (CHGNETA)」指令會變更系統的網路屬性。ALWADDCLU 網路屬性會指定節點是否允許另一個系統，將其新增為叢集中的節點。

註: 您必須具有 *IOSYSCFG 權限，才可變更網路屬性 ALWADDCLU。

您可以挑選下列其中一個值：

***SAME**

值不會變更。系統的出廠預設值為 *NONE。

***NONE**

沒有其他系統可將此系統新增為叢集中的節點。

***ANY** 任何其他系統都可將此系統新增為叢集中的節點。

***RQSAUT**

在已鑑別叢集新增要求之後，任何其他系統都可將此系統新增為叢集中的節點。

檢查 ALWADDCLU 網路屬性，以查看是否允許已新增的節點成為叢集的一部份，以及是否要透過 X.509 數位憑證的使用，來驗證叢集要求。**數位憑證**是個人身份證明的套表，可以電子方式來驗證。若驗證是必要的，則要求節點及已新增的節點必須讓下列各項安裝在系統上：

- i5/OS 選項 34 (數位憑證管理程式)
- 密碼存取提供者

選取 *RQSAUT 時，必須適當地設定 i5/OS 叢集安全伺服器應用程式的認證中心信任清單。伺服器應用程式 ID 是 QIBM_QCST_CLUSTER_SECURITY。至少，新增容許結合叢集的那些節點之認證權限。

相關概念

數位憑證管理

第 127 頁的『一般叢集問題』

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

相關參考

變更網路屬性 (CHGNETA) 指令

分送全叢集資訊

- 了解使用及管理全叢集資訊的安全性含意。

分送資訊 (QcstDistributeInformation) API 可用於將訊息從叢集資源群組回復網域中的一個節點，傳送至該回復網域中的其他節點。這在跳出程式處理中很有用。但是，應該注意該資訊沒有加密。不應該使用此機制來傳送安全資訊，除非您使用安全網路。

可使用「叢集雜湊表 API」，在叢集節點之間共用及抄寫非持續性資料。資料儲存在非持續性儲存體中。這表示資料只會保留到叢集節點不再是叢集雜湊表之一部份為止。僅可從定義於叢集雜湊表網域中之叢集節點中，使用這些 API。叢集節點必須在叢集中作用。

透過叢集傳訊所分送的其他資訊同樣不安全。這包括低層次的叢集傳訊。這樣一來，當對跳出程式資料進行變更時，不會有包含該資料之訊息的加密。

相關參考

分送資訊 (QcstDistributeInformation) API

叢集雜湊表 API

維護所有節點上的使用者設定檔

- 您可以使用兩種機制來維護叢集內所有節點上的使用者設定檔。

- 一種機制是建立叢集管理網域，以監視叢集中各節點上的共用資源。除了使用者設定檔之外，叢集管理網域還可以監視數種類型的資源，因此能輕鬆地管理在各節點上共用的資源。如需這些資源的詳細資訊，請參閱「受監視資源」。更新使用者設定檔時，如果叢集管理網域處於作用中，則會自動將變更傳送至其他節點。如果叢集管理網域不在作用中，則將在啟動叢集管理網域後立即傳送變更。

- 註：**如果您計劃共用叢集中使用密碼同步化的使用者設定檔，則必須將「保持伺服器安全性 (QRETSVRSEC)」系統值設為 1。

- 利用第二種機制，管理者還可以使用「iSeries 領航員」中的「管理中心」，跨多個系統及系統群組執行功能。這項支援包含一些通用的使用者管理作業，是操作員在其叢集內的多重系統上需要執行的作業。「管理中心」可以對系統群組執行使用者設定檔功能。建立使用者設定檔時，管理者可以指定在目標系統上執行一個後置傳遞指令。

相關概念

第 108 頁的『工作結構和使用者佇列』

在管理叢集時，您需要了解工作結構及使用者佇列。

第 8 頁的『叢集管理網域』

叢集管理網域可用來管理需要在叢集環境中的節點間保持一致的資源。

搭配使用防火牆及叢集的考量

- 如果您在使用防火牆的網路上使用叢集，則應該了解其一些限制及基本要求。

- 如果您搭配使用防火牆及叢集，則需要讓每一個節點都能夠對其他叢集節點傳送離埠訊息，以及接收來自其他叢集節點的入埠訊息。在防火牆上必須有個出入口，能讓每個節點上的每個叢集位址與每個其他節點上的每個

叢集位址進行通訊。透過網路傳送的 IP 封包可以是各種類型的資料傳輸。叢集使用連通測試 (ping)，其類型為 ICMP，也使用 UDP 及 TCP。當使用者配置防火牆時，他們可以根據類型來過濾資料傳輸。為了讓叢集運作，防火牆必須允許 ICMP、UDP 及 TCP 的資料傳輸。可在任何埠上傳送離埠資料傳輸，而入埠資料傳輸則可在埠 5550 及 5551 上接收。

叢集配置核對清單

在您開始配置叢集之前，請先完成叢集配置核對清單，來確定您已備妥適當的環境。

表 13. 叢集的 TCP/IP 配置核對清單

TCP/IP 基本要求	
—	使用「啟動 TCP/IP (STRTCP)」指令，讓您計劃併入叢集的每一個節點上啟動 TCP/IP。
—	配置 TCP 迴路位址 (127.0.0.1)，並驗證狀態顯示為作用中。在叢集中的每一個節點上，使用「TCP/IP 網路狀態 (WRKTCPSTS)」指令來驗證。
—	在主旨節點上，「使用 TCP/IP 網路狀態 (WRKTCPSTS)」指令來驗證用於形成指定節點之叢集的 IP 位址，其狀態必須顯示為作用中。
—	使用 STRTCPMSVR *INETD 指令，或透過「iSeries 領航員」完成下列步驟，以驗證 INETD 在叢集中的所有節點上都處於作用中。 <ol style="list-style-type: none"> 1. 在「iSeries 領航員」中，展開網路。 2. 展開伺服器。 3. 展開 TCP/IP。 4. 在 INETD 上按一下滑鼠右鍵，然後選取啟動。 這可由主旨節點上的「作用中工作」中之 QTOGINTD (使用者 QTCP) 工作的存在來驗證。
—	驗證 INETD 的使用者設定檔 (在 /QIBM/ProdData/OS400/INETD/inetd.config 中指定) 只擁有最小權限。如果此使用者設定檔的權限大於最小權限，STRCLUNOD (啟動叢集節點) 指令便會失敗。依預設，會將 INETD 的使用者設定檔指定為 QUSER。
—	驗證叢集中的每個 IP 位址均可遞送至叢集中的所有其他 IP 位址，並可傳送 UDP 資料包給它們。使用 PING 指令指定本機 IP 位址，使用 TRACEROUTE 指令指定 UDP 訊息。
—	驗證沒有其他應用程式使用埠 5550 與 5551。這些埠保留給 IBM 叢集。可用「使用 TCP/IP 網路狀態 (WRKTCPSTS)」指令來檢視埠的用法。一旦開啓了 INETD，叢集將開啓埠 5550，並處於「接收」狀態。

如果您計劃在叢集中使用切換式裝置，則必須滿足下列基本要求：

表 14. 叢集的回復裝置配置核對清單

回復裝置基本要求	
—	驗證已安裝選項 41 (HA 可切換資源)，且在將位於裝置網域中的所有叢集節點上都存在有效的授權碼。請注意，使用「iSeries 領航員」叢集管理介面需要此選項。
—	若要存取「iSeries 領航員」中的磁碟管理功能，請以 DST 存取權及使用者設定檔配置服務工具伺服器 (STS)。如需詳細資訊，請參閱「設定通訊」。
—	如果您正在系統的邏輯分割區之間切換回復裝置，並使用 HMC 以外的項目來管理邏輯分割區，請為分割區啓用「虛擬 OptiConnect」。在專用服務工具 (DST) 登入時，已完成此動作。如需詳細資訊，請參閱虛擬 OptiConnect。 如果您正在使用「硬體管理主控台」管理您的分割區，請在 OptiConnect 標籤上變更您的分割區設定檔內容，為切換式配置中的每個分割區啓用「虛擬 OptiConnect」。您必須啟動分割區設定檔，才能反映變更。

表 14. 叢集的回復裝置配置核對清單 (繼續)

回復裝置基本要求	
—	<p>若 HSL OptiConnect 迴圈上的直立式主機可在兩個系統之間切換，而其中一個系統具有邏輯分割區，請為分割區啟用 HSL OptiConnect。如果您正在使用 HMC 以外的項目來管理邏輯分割區，則在專用服務工具 (DST) 登入時，已完成此動作。</p> <p>如果您正在使用「硬體管理主控台」管理您的分割區，請在 OptiConnect 標籤上變更您的分割區設定檔內容，為切換式配置中的每個分割區啟用 HSL OptiConnect。您必須啟動分割區設定檔，才能反映變更。</p>
—	<p>如果您正在邏輯分割區之間切換回復裝置，並使用 HMC 以外的項目來管理邏輯分割區，則必須配置要在分割區之間共用的匯流排，或配置「I/O 儲存區」。匯流排必須由一個分割區配置為「自有共用匯流排」，且必須將參與裝置切換的所有其他分割區配置為「使用共用匯流排」。</p> <p>如果您正在使用「硬體管理主控台」管理您的邏輯分割區，則必須配置 I/O 儲存區 (包含 I/O 處理器、I/O 配接卡及所有連接的資源)，讓獨立磁碟儲存區可在分割區之間切換。每一個分割區都必須擁有對 I/O 儲存區的存取權限。如需詳細資訊，請參閱「將硬體設定為可切換」。如需切換式裝置之實體規劃基本要求的詳細資訊，請參閱「實體規劃基本要求」。</p>
—	<p>當切換兩個不同系統之間的 HSL 迴圈上之直立式主機時，直立式主機必須配置為可切換。如需詳細資訊，請參閱「將硬體設定為可切換」。</p>
—	<p>直立式主機加到現有的 HSL 迴圈時，請重新啟動相同迴圈上的所有伺服器。</p>
—	<p>您的通訊路徑之最大傳輸單位 (MTU) 必須大於叢集通訊可調整的參數 (訊息片段大小)。在主旨節點上，可用「使用 TCP/IP 網路狀態 (WRKTCPSSTS)」指令來驗證叢集 IP 位址的 MTU。在整個通訊路徑的每一個步驟中，也必須驗證 MTU。一旦建立了叢集，它可輕易地降低訊息片段大小參數，而非為通訊路徑提升 MTU。如需訊息片段大小的詳細資訊，請參閱「可調整的叢集通訊參數」。您可以使用「擷取叢集資源服務資訊 (QcstRetrieveCRSInfo) API」來檢視調整參數的現行設定值，以及使用「變更叢集資源服務 (QcstChgClusterResourceServices) API」來變更設定值。</p>

表 15. 叢集的安全配置核對清單

安全基本要求	
—	<p>若嘗試啟動遠端節點，則 ALWADDCLU (允許新增到叢集) 網路屬性必須適當地設定在目標節點上。這應該設定為 *ANY 或 *RQSAUT，視您的環境而定。如果設為 *RQSAUT，則必須安裝 i5/OS 選項 34 (數位憑證管理程式) 及「密碼存取提供的產品」。如需設定 ALWADDCLU 網路屬性的詳細資訊，請參閱「啓用要新增到叢集的節點」。</p>
—	<p>為在 /QIBM/ProdData/OS400/INETD/inetd.config 中指定的 INETD 啓用使用者設定檔的狀態。必須具有 *SECADM 或 *ALLOBJ 特殊權限。依預設，會將 INETD 的使用者設定檔指定為 QUSER。</p>
—	<p>驗證呼叫叢集資源服務 API 的使用者設定檔存在於所有叢集節點上，且具有 *IOSYSCFG 權限。</p>
—	<p>驗證針對叢集資源群組 (CRG) 執行跳出程式的使用者設定檔，已存在於所有的回復網域節點上。</p>

表 16. 叢集的工作配置核對清單

工作注意事項	
—	<p>工作可由叢集資源服務 API 提出，以處理要求。工作將在建立叢集資源群組時，於要執行指定的跳出程式之使用者設定檔下執行，或在要求 API 的使用者設定檔下執行 (僅供回復裝置 CRG 中的轉接裝置使用)。使用者必須確定與使用者設定檔相關之服務工作佇列的子系統，是針對它可從該工作佇列執行的工作數，而配置為：*NOMAX。</p>
—	<p>工作將提出至工作說明指定的工作佇列中，為 CRG 定義的使用者設定檔中可取得工作說明。預設工作說明將使工作傳送到 QBATCH 工作佇列中。因為此工作佇列是使用於許多使用者工作，因此可能無法以及時的方式來執行跳出程式工作。使用者應該以唯一的使用者佇列來考慮唯一的工作說明。</p>

表 16. 叢集的工作配置核對清單 (繼續)

工作注意事項	
—	當跳出程式工作執行時，它們將使用來自工作說明的遞送資料，以選擇將使用哪一個主記憶體儲存體及執行時間屬性。預設值將導致工作在具有工作優先順序 50 的其他指批次工作之儲存區中執行。它們都無法為跳出程式工作產生需要的效能。起始跳出程式工作的子系統 (使用唯一工作佇列的相同子系統)，應該將跳出程式工作指派給儲存區，而相同子系統或其他子系統起始的其他工作，不使用該儲存區。此外，應該為跳出程式工作指定執行優先順序 15，如此它們將在大部分所有其他使用者工作之前執行。
—	QMLTTHDACN 系統值必須設為 1 或 2。

有幾個軟體解決方案，可用來配置及管理您的叢集。其中一個解決方案是「iSeries 領航員」叢集管理。如果您選擇使用「iSeries 領航員」，則必須滿足下列基本要求：

表 17. 叢集的「iSeries 領航員」配置核對清單

「iSeries 領航員」叢集管理注意事項	
—	必須安裝選項 41 (i5/OS - HA 可切換資源)，且在將位於裝置網域中的所有叢集節點上都存在有效的授權碼。
—	使用 STRHOSTSVR (啟動主電腦伺服器) 指令：STRHOSTSVR SERVER(*ALL)，驗證已啟動所有的主電腦伺服器。
—	使用 STRTCPSVR (啟動 TCP/IP 伺服器) 指令：STRTCPSVR SERVER(*MGTC)，驗證已啟動「管理中心」伺服器。

相關概念

第 71 頁的『「iSeries 領航員」叢集管理』

IBM 提供的叢集管理介面可透過「iSeries 領航員」使用，並透過「選項 41 (i5/OS - HA 可切換資源)」存取。

『INETD 伺服器』

必須已啟動網際網路常駐程式 (INETD) 伺服器，以便新增或啟動節點及合併分割區處理程序。

第 90 頁的『可調整的叢集通訊參數』

「變更叢集資源服務 (QcstChgClusterResourceServices) API」可以調整部分叢集拓樸服務及叢集通訊效能和配置參數，以充份適用於形成叢集的許多獨特應用程式及網路環境。這個 API 適用於執行叢集版本 2 或更高版本的任何叢集。

相關參考

第 92 頁的『叢集管理網域核對清單』

涵蓋建立叢集管理網域之前必須完成的所有必備項目。

INETD 伺服器

必須已啟動網際網路常駐程式 (INETD) 伺服器，以便新增或啟動節點及合併分割區處理程序。

建議 INETD 伺服器一直在您的叢集中執行。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (i5/OS - HA 可切換資源)」。

若要啟動 INETD 伺服器，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開網路。
2. 展開伺服器。
3. 展開 TCP/IP。

4. 在 **INETD** 上按一下滑鼠右鍵，然後選取**啓動**。

使用 **CL** 指令及 **API**

使用 **STRTCPSVR** (啓動 TCP/IP 伺服器) 指令並指定 ***INETD** 參數，亦可啓動 **INETD** 伺服器。當已啓動 **INETD** 伺服器時，**QTOGINTD** (使用者 **QTCP**) 工作將出現在主旨節點上的「作用中工作」清單中。

相關概念

第 127 頁的『一般叢集問題』

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

相關參考

STRTCPSVR (啓動 TCP/IP 伺服器) 指令

可調整的叢集通訊參數

「變更叢集資源服務 (**QcstChgClusterResourceServices**) **API**」可以調整部分叢集拓樸服務及叢集通訊效能和配置參數，以充份適用於形成叢集的許多獨特應用程式及網路環境。這個 **API** 適用於執行叢集版本 2 或更高版本的任何叢集。

「變更叢集配置調整 (**CHGCLUCFG**)」指令提供基本層次的調整，而 **QcstChgClusterResourceServices** **API** 則提供基本及進階兩種層次的調整。

QcstChgClusterResourceServices **API** 及 **CHGCLUCFG** 指令可用來調整叢集效能和配置。此 **API** 和指令提供基本的調整支援層次，可將叢集調整到一組預先定義的值，包含以高、低、一般來表示的逾時和傳訊間隔值。若想要進行進階層次的調整 (通常有 **IBM** 支援人員協助)，還可以在預先定義的值範圍上，使用 **API** 來調整個別參數。不適當地變更個別參數很容易導致叢集效能降低。

何時及如何調整叢集參數？

CHGCLUCFG 指令及 **QcstChgClusterResourceServices** **API** 提供快捷的方法來設定叢集效能和配置參數，您並不需要了解細節。這個基本調整層次主要是影響活動訊號靈敏度和叢集訊息逾時值。基本調整支援層次的有效值如下：

1 (高逾時值/較不頻繁活動訊號)

2 (預設值)

叢集通訊效能和配置參數上使用正常的預設值。這個設定可讓所有參數返回原始預設值。

3 (低逾時值/較頻繁活動訊號)

調整叢集通訊來減少活動訊號間隔和降低各種訊息逾時值。活動訊號愈頻繁和逾時值愈短，叢集對通訊失敗反應愈快 (高敏感度)。

下列表格顯示活動訊號失敗造成節點分割的回應時間範例：

	1 (低敏感度)			2 (預設值)			3 (高敏感度)		
	偵測活動訊號問題	分析	總計	偵測活動訊號問題	分析	總計	偵測活動訊號問題	分析	總計
單一子網路	00:24	01:02	01:26	00:12	00:30	00:42	00:04	00:14	00:18
多重子網路	00:24	08:30	08:54	00:12	04:14	04:26	00:04	02:02	02:06

註：指定的時間格式為「分:秒」。

根據一般網路負載和所用的特定實體媒體，叢集管理者可選擇調整活動訊號靈敏度和訊息逾時層次。例如，如果是高速度高可靠性的傳輸 (如 OptiConnect)，且所有系統都在一個通用 OptiConnect 匯流排上的叢集中，則您可能希望建立一個較敏感的環境，以確保能快速偵測來達成更快速的錯誤後移轉。選擇選項 3。如果在高負載的 10Mbps 乙太網路匯流排上執行，且設定值會因為網路尖峰負荷造成偶而的分割，則可以選擇選項 1 來降低叢集對尖峰負荷的靈敏度。

「變更叢集資源服務 API」也可以針對特定的網路環境需求來調整特定的個別參數。例如，再假設一個叢集的所有節點都在一個 OptiConnect 匯流排上。將「訊息片段大小」參數設為最大值 32,500 個位元組，比預設值 1,464 個位元組更符合 Opticonnect 「最大傳輸單位 (MTU)」大小，可以明顯增強叢集訊息的效能。這可以減少分割和重組大量訊息的額外執行時間。當然，其優點是根據叢集應用程式及這些應用程式產生的叢集傳訊使用情形而定。其他參數定義在 API 文件中，可用來調整叢集傳訊的效能，或變更叢集對於分割的敏感度。

相關概念

第 106 頁的『調整叢集效能』

因為您的通訊環境中可能存在著明顯的差異，所以您可以調整影響叢集通訊的變數來切合自己的環境。

叢集解除配置核對清單

如果您需要刪除叢集或 CRG，則必須有系統地移除各種叢集元件，以確保能完全解除配置。

表 18. 叢集的獨立磁碟儲存區解除配置核對清單

獨立磁碟儲存區基本要求	
—	如果您計劃移除獨立磁碟儲存區群組的子集，或移除切換式裝置中的最後一個獨立磁碟儲存區，則必須先結束 CRG。請使用「結束叢集資源群組 (ENDCRG)」指令。
—	如果您想要刪除加入叢集的獨立磁碟儲存區，強烈建議您先從切換式裝置 (亦稱為裝置叢集資源群組 (CRG)) 移除磁碟儲存區的配置物件。若要從切換式裝置刪除磁碟儲存區的配置物件，請遵循下列步驟： 若要從切換式裝置刪除磁碟儲存區，請遵循下列步驟： <ol style="list-style-type: none"> 1. 在「iSeries 領航員」中，展開管理中心 → 叢集。 2. 展開包含切換式裝置的叢集名稱 → 切換式裝置。 3. 按一下切換式裝置的名稱。 4. 在「iSeries 領航員」的右窗格中，以滑鼠右鍵按一下磁碟儲存區，然後選取移除。 您還可以使用「移除 CRG 裝置項目 (RMVCRGDEVE)」指令，從 CRG 移除獨立磁碟儲存區的配置物件。 從叢集切換式裝置中移除獨立磁碟儲存區的配置物件之後，即可刪除獨立磁碟儲存區。
—	完成下列作業，即可刪除獨立磁碟儲存區的裝置說明： <ol style="list-style-type: none"> 1. 在指令行介面上輸入 <code>WRKDEVD DEVD(*ASP)</code>，再按 Enter 鍵。 2. 向下捲動頁面，直到您看到要刪除之獨立磁碟儲存區的裝置說明為止。 3. 依裝置說明名稱選取「選項 4 (刪除)」，再按 Enter 鍵。

表 19. 叢集的叢集資源群組解除配置核對清單

叢集資源群組基本要求	
—	完成下列任一項作業，即可刪除叢集資源群組： <ol style="list-style-type: none"> 1. 如果節點上的叢集不在作用中，則在指令行介面上輸入 <code>DLTCRG CRG(CRGNAME)</code>。CRGNAME 是您要刪除之 CRG 的名稱。按 Enter 鍵。 2. 如果節點上的叢集處於作用中，則在指令行介面上輸入 <code>DLTCRGCLU CLUSTER(CLUSTERNAME) CRG(CRGNAME)</code>。CLUSTERNAME 是叢集的名稱。CRGNAME 是您要刪除之 CRG 的名稱。按 Enter 鍵。

叢集管理網域規劃

叢集管理網域需要進行規劃，以管理在叢集管理網域內的節點間共用的資源。

當您建立叢集管理網域時，會自動建立對等節點 CRG 來代表該網域。您可以使用 API、CL 指令及「iSeries 領航員」來管理叢集管理網域。

叢集管理者可以建立叢集管理網域，然後新增在節點間共用的受監視資源。「i5/OS 叢集」提供可在叢集管理網域內的節點間共用的系統資源清單，並以受監視資源項目 (MRE) 來表示。請參閱受監視資源，以取得可監視的系統資源之完整清單。

設計叢集管理網域時，您應該回答下列問題：

將共用哪些資源？

您將需要判定需要共用哪些系統資源。您可以選取每個資源的屬性，以自訂在節點間共用的項目。在多個節點上執行的應用程式可能需要特定的環境變數才能正確執行。此外，跨數個節點的資料也可能需要特定的使用者設定檔才能存取。在判定要共用的資源之前，應該先了解應用程式及資料的作業基本要求。

叢集管理網域中將包括哪些節點？

您應該判定叢集中的哪些節點要由叢集管理網域來管理。節點不能在多個叢集管理網域中。例如，如果叢集中有四個節點（「節點 A」、「節點 B」、「節點 C」及「節點 D」）。「節點 A」與「節點 B」在一個叢集管理網域中，而「節點 C」與「節點 D」在另一個叢集管理網域中。然而，您不能將「節點 B」與「節點 C」放在另一個叢集管理網域中。

叢集管理網域的命名慣例為何？

根據叢集環境的複雜性及大小，您可能想要為對等節點 CRG 及叢集管理網域建立標準命名慣例。因為會建立對等節點 CRG 來代表叢集管理網域，所以您會想要區分對等節點 CRG 與監視叢集中的資源之 CRG。例如，代表叢集管理網域的對等節點 CRG 可以命名為 *ADMDMN1*、*ADMDMN2*，依此類推，而其他對等節點 CRG 可以命名為 *PEER1*。您還可以使用「列出叢集資源群組資訊 (QcstListClusterResourceGroupIn) API」，判定對等節點 CRG 是否用作叢集管理網域。

叢集管理網域核對清單

涵蓋建立叢集管理網域之前必須完成的所有必備項目。

表 20. 叢集管理網域核對清單

叢集管理網域基本要求	
—	驗證是否已配置叢集。請參閱叢集配置核對清單。
—	如果您計劃監視叢集中使用密碼同步化的使用者設定檔，則必須將「保持伺服器安全性 (QRETSVRSEC)」系統值設為 1。
—	若要新增資源至叢集管理網域，則叢集管理網域中的所有節點都必須處於作用中，並參與群組且未經分割。

配置叢集

瞭解如何建立叢集。

IBM 及 IBM 叢集中介軟體「IBM 商業夥伴」已組成一個團隊，可提供最先進的叢集資源服務功能，以及叢集管理的圖形式使用者介面 (GUI)。i5/OS 叢集資源服務提供一組整合式服務，用來維護叢集拓樸、執行活動訊號監視，以及建立與管理叢集配置及叢集資源群組。叢集資源服務也提供可靠的傳訊功能，它會記錄叢集中每一

個節點，並確定所有節點且有關於叢集資源狀態的一致資訊。此外，叢集資源服務還提供一組控制語言 (CL) 指令，以及應用程式介面 (API) 和機能，可供 iSeries 應用程式提供者或客戶使用來增強其應用程式可用性。叢集資源服務功能也可以透過「iSeries 領航員」叢集管理與叢集中介軟體「IBM 商業夥伴」產品所提供的圖形式使用者介面解決方案來存取。

入門

請遵循下列步驟來配置叢集：

1. 選取軟體解決方案。

如需配置及管理叢集的完整選項說明，請參閱第 70 頁的『配置與管理叢集的解決方案』。

2. 滿足硬體、軟體及通訊基本要求。

複查叢集規劃中的叢集基本要求。

3. 設定叢集的網路與伺服器環境。

使用第 87 頁的『叢集配置核對清單』，確定您已準備好在您的環境中配置叢集。

4. 配置您的叢集。

相關概念

第 141 頁的『叢集支援的客服電話』

如果您需要聯絡 IBM 來解決叢集問題，請參閱此主題。

建立叢集

若要建立及配置叢集，您至少要將一個節點併入叢集中，且您必須具有將在叢集中的至少一個節點的存取權限。

- 1 若僅指定一個節點，則它必須是您目前存取的伺服器。如需建立叢集的完整基本要求清單，請參閱第 87 頁的『叢集配置核對清單』。

若您將在叢集中使用可切換的裝置，則有其他的需求，不同於不使用可切換的裝置之叢集的需求。若要設定包括可切換的裝置之叢集環境，則必須注意，這樣可避免越過叢集的衝突。如需如何配置叢集以使用切換式裝置的逐步指示，請參閱建立切換式獨立磁碟儲存區。

使用「iSeries 領航員」

這需要安裝及授權選項 41 (HA 可切換資源)。

「iSeries 領航員」叢集管理擁有精靈的特性，可逐步引導您建立及啟動由一或兩個叢集節點組成的簡式叢集。一旦您建立了一個節點或兩個節點的叢集，您可將節點新增到叢集中。在「iSeries 領航員」中建立及管理的叢集最多可包含四個節點。此精靈將逐步引導您指定要併入的伺服器，以及建立叢集資源群組。當您正建立簡式叢集時，您正建立叢集的伺服器必須是其中一個節點。

若要使用「iSeries 領航員」中的「新建叢集」精靈來建立簡式叢集，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 以滑鼠右鍵按一下**叢集**，然後選取**新建叢集**。
3. 遵循精靈的指示來建立叢集。

一旦您建立了叢集，請確定要：

1. 新增您要併入叢集的所有節點。最多有四個節點可新增至在「iSeries 領航員」中建立及管理的叢集。
2. 將需要的節點新增到裝置網域 (以供使用可切換的硬體群組及獨立磁碟儲存區)。

3. 建立及啓動可切換的資源 (切換式裝置、切換式應用程式及切換式資料)。

「iSeries 領航員」中的線上說明包含了完成這些作業的逐步程序。

使用 CL 指令及 API

您也可以使用 CL 指令或 API，以建立叢集：

1. **建立叢集。**
 - 建立叢集 (CRTCLU) 指令
 - 建立叢集 (QcstCreateCluster) API
2. **將節點從作用中叢集節點新增到您的叢集。**
 - 新增叢集節點項目 (ADDCLUNODE) 指令
 - 新增叢集節點項目 (QcstAddClusterNodeEntry) API
3. **啓動叢集節點。**
 - 啓動叢集節點 (STRCLUNOD) 指令
 - 啓動叢集節點 (QcstStartClusterNode) API
4. **定義裝置網域。** 若您計畫使用可切換的裝置，則必須將需要的節點併入裝置網域中。
 - 新增裝置網域項目 (ADDDEVDMNE) 指令
 - 新增裝置網域項目 (QcstAddDeviceDomainEntry) API
5. **建立叢集資源群組 (CRG)。**
 - 建立叢集資源群組 (CRTCRG) 指令
 - 建立叢集資源群組 (QcstCreateClusterResourceGroup) API
6. **啓動叢集資源群組 (CRG)。**
 - 啓動叢集資源群組 (STRCRG) 指令
 - 啓動叢集資源群組 (QcstStartClusterResourceGroup) API

管理叢集

本主題的資訊涵蓋一些涉及管理叢集的作業。

若您尚未考慮要用來管理叢集的介面類型，在繼續之前，請先參閱管理叢集的解決方案。

在您配置叢集之後，可對叢集做的一些變更包括：

叢集作業

- 將節點新增到叢集中
- 從叢集中移除節點
- 啓動叢集節點
- 結束叢集節點
- 將叢集的叢集版本調整為最新層次
- 刪除叢集
- 變更叢集節點

叢集資源群組作業

- 建立新的叢集資源群組
- 刪除現有的叢集資源群組
- 啓動叢集資源群組

- | • 將節點新增到叢集資源群組
- | • 從叢集資源群組中移除節點
 - 結束叢集資源群組
 - 變更叢集資源群組 (CRG) 的回復網域
 - 執行切換
 - 將節點新增到裝置網域
 - 從裝置網域中移除節點

本主題也將協助您儲存叢集配置。您可能想要了解叢集資源服務工作的結構及叢集 API 如何利用使用者佇列。您可讀到有關結束叢集工作的正確方式，以及如何監視叢集狀態。您也會學到可靠訊息功能和活動訊號監視如何讓您保持更新為最新的叢集狀態。

| 叢集管理網域作業

- | • 建立叢集管理網域
- | • 新增受監視資源
- | • 刪除叢集管理網域

相關概念

第 25 頁的『可靠的訊息功能』

叢集資源服務的可靠的訊息功能會記錄叢集中之每一個節點，並確定所有節點都有關於叢集資源狀態的一致資訊。

第 23 頁的『活動訊號監督』

活動訊號監視是一種叢集資源服務功能，它可將信號從叢集中的每個節點，傳送到叢集中的所有其他節點，通知它們仍處於作用中，以確定每個節點都在作用中。

將節點新增到叢集中

您可以使用「iSeries 領航員」或指令，新增節點至叢集。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

「iSeries 領航員」支援的簡式叢集最多可由四個節點組成。如果叢集中已存在四個節點，則會停用**新增節點**選項。如果您的叢集需要擴充到四個節點以上，您應該使用「叢集」指令及 API 或叢集中介軟體「IBM 商業夥伴」產品，以支援最多 128 個節點。

若要將節點新增到現有的叢集中，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開「管理中心」。
2. 展開**叢集**。
3. 展開您要新增節點的叢集。
4. 在**節點**上按一下滑鼠右鍵，然後選取**新增節點...**。

使用「叢集」指令及 API

您也可以使用下列各項，以將節點新增到叢集中：

- 新增叢集節點項目 (ADDCLUNODE) 指令
- 新增叢集節點項目 (QcstAddClusterNodeEntry) API

相關概念

第 72 頁的『叢集指令及 API』

i5/OS 叢集資源服務提供一組控制語言 (CL) 指令、應用程式介面 (API) 及機能，供 iSeries 應用程式提供者或客戶增強其應用程式可用性。

第 76 頁的『叢集中介軟體「IBM 商業夥伴」與可用的叢集產品』

您可以向 IBM 叢集中介軟體「IBM 商業夥伴」，購買提供形成叢集所需的邏輯抄寫功能及簡化叢集建立與管理的產品。

啓動叢集節點

啓動叢集節點會在叢集中的節點上啓動叢集資源服務。自叢集的第 3 版開始，節點可以啓動自己，且只要能夠在叢集中找到作用中節點，就能夠重新結合現行的作用中叢集。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

當指定的節點上順利啓動叢集資源服務時，節點的狀態將設定為已啓動。

若要在節點上啓動形成叢集，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 展開含有您要啓動形成叢集之節點的叢集。
4. 按一下**節點**。
5. 以滑鼠右鍵按一下您要啓動叢集的節點，然後選取**叢集** → **啓動**。

叢集

使用 CL 指令及 API

您也可以使用 CL 指令或 API 來啓動節點。當指定的節點上順利啓動叢集資源服務時，節點的狀態將設定為作用中。

- 啓動叢集節點 (STRCLUNOD) 指令
- 啓動叢集節點 (QcstStartClusterNode) API

相關工作

第 107 頁的『結束叢集工作』

請勿嘗試直接結束叢集工作。

第 133 頁的『從叢集工作失敗中回復』

叢集資源服務工作失敗通常表示有其它問題。

| 結束叢集節點

| 停止或結束節點會停止該節點上的叢集資源服務。

| 使用「iSeries 領航員」

| 這需要安裝及授權「選項 41 (HA 可切換資源)」。

| 當指定的節點上順利停止叢集資源服務時，節點的狀態將設定為已停止。

- | 若要在節點上結束形成叢集，請遵循下列步驟：
- | 1. 在「iSeries 領航員」中，展開**管理中心**。
- | 2. 展開**叢集**。
- | 3. 展開含有您要停止形成叢集之節點的叢集。
- | 4. 按一下**節點**。
- | 5. 以滑鼠右鍵按一下您要結束叢集的節點，然後選取**叢集** → **停止**。

| 使用 **CL** 指令及 **API**

| 您也可以使用 **CL** 指令或 **API** 來結束節點。當指定的節點上順利結束叢集資源服務時，節點的狀態將設定為 **非作用中**。

- | • 結束叢集節點 (ENDCLUNOD) 指令
- | • 結束叢集節點 (QcstEndClusterNode) API

| **相關工作**

- | 第 107 頁的『結束叢集工作』
- | 請勿嘗試直接結束叢集工作。
- | 第 133 頁的『從叢集工作失敗中回復』
- | 叢集資源服務工作失敗通常表示有其它問題。

調整叢集的叢集版本

叢集版本定義叢集中所有節點正在以此相互通訊的層次。

叢集版本化技術可讓叢集包含多重版次的系統，並可判斷所要使用的通訊協定層次，讓系統可以完全互用。

若要變更叢集版本，叢集中的所有節點必須是同一個潛在版本。然後，可以變更叢集版本來符合潛在版本。這可讓新的功能產生作用。版本只能以 1 遞增。如果沒有先刪除叢集再重建較低版本，則無法遞減版本。現行叢集版本一開始是由叢集中定義的第一個節點來設定。新增到叢集中的後續節點必須等於現行叢集版本或下一個層次版本，否則它們無法新增到叢集中。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

若要調整叢集的叢集版本，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 以滑鼠右鍵按一下叢集，選取**內容**。
4. 將「叢集」版本變更成您想要的設定。

使用「叢集」指令及 **API**

您也可以使用下列方法來調整叢集的叢集版本：

- 變更叢集版本 (CHGCLUVER) 指令
- 調整叢集版本 (QcstAdjustClusterVersion) API

相關概念

- 第 12 頁的『叢集版本』
- 叢集版本代表叢集上可用功能的層次。

第 127 頁的『一般叢集問題』

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

相關工作

『刪除叢集』

刪除叢集時，會在所有作用中叢集節點結束叢集資源服務，並將從叢集中移除。

刪除叢集

刪除叢集時，會在所有作用中叢集節點結束叢集資源服務，並將從叢集中移除。

- | **重要：** 若您在叢集中已有獨立磁碟儲存區，則在刪除叢集之前，應該先使用「移除裝置網域項目 (RMVDEVDMNE)」指令，從裝置網域中移除每個節點。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

若要刪除叢集，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 以滑鼠右鍵按一下您要刪除的叢集，然後選取**刪除**。

使用 CL 指令及 API

您也可以使用 CL 指令或 API，以刪除叢集。

- | • 刪除叢集 (DLTCLU) 指令
- | • 刪除叢集 (QcstDeleteCluster) API

相關工作

第 97 頁的『調整叢集的叢集版本』

叢集版本定義叢集中所有節點正在以此相互通訊的層次。

建立 CRG

- | 您可以建立數種類型的 CRG：應用程式、資料、裝置及對等節點 CRG。

| 若要在叢集中建立 CRG，請完成下列步驟：

- | 1. 在「iSeries 領航員」中，展開**管理中心** → **叢集**。
- | 2. 展開您要新增 CRG 的叢集。
 - | a. 如果您要建立「裝置 CRG」，請以滑鼠右鍵按一下**切換式硬體**，然後選取**新建群組**。附註：只有在啟動回復網域中的所有節點時才可以使用**新建群組**選項。如需詳細資訊，請參閱「啟動叢集節點」。
 - | b. 如果您要建立「應用程式 CRG」，請以滑鼠右鍵按一下**切換式軟體**，然後選取**新增產品**。
 - | c. 如果您要建立「資料 CRG」，請以滑鼠右鍵按一下**切換式資料**，然後選取**新建群組**。
 - | d. 如果您要建立「對等節點 CRG」，請以滑鼠右鍵按一下**對等節點資源**，然後選取**新建對等節點 CRG**。

使用 CL 指令及 API

| 您可以使用下列指令及 API 來建立 CRG：

- | • 建立叢集資源群組 (CRTCRG) 指令
- | • 建立叢集資源群組 (QcstCreateClusterResourceGroup) API

以作用中接管 IP 位址來建立應用程式 CRG

您可以指定在建立應用程式 CRG 時，允許使用作用中接管 IP 位址。只有在使用者配置接管 IP 位址時才允許這樣做。

以前，若使用者配置了作用中接管 IP 位址，您就能利用它來建立應用程式 CRG。但是，如果接管 IP 位址已在作用中，便無法啟動應用程式 CRG。現在，您可以指定在建立應用程式 CRG 時允許作用中接管 IP 位址。當您啟動允許作用中接管 IP 位址的應用程式 CRG 時，便允許啟動該 CRG。

若要在建立應用程式 CRG 時允許使用作用中接管 IP 位址，請完成下列步驟：

1. 在指令行介面中，輸入：

```
CRTCRCG CLUSTER(MYCLUSTER) CRG(MYCRG) CRGTYPE(*APP) EXITPGM(QDEVELOP/EXITPGM)
USRPRF(USER) RCYDMN((NODE1 *PRIMARY)(NODE2 *BACKUP)) TKVINTNETA('10.1.2.1') CFGINTNETA(*USR *YES)
```

TKVINTNETA 參數識別要使用的接管 IP 位址，而 **CFGINTNETA** 參數識別使用者將配置接管 IP 位址，而且此位址可在「啟動 CRG」時處於作用中。

在建立應用程式 CRG 以允許作用中接管 IP 位址之後，您可以啟動 CRG。

啟動 CRG

您可以啟動數種類型的 CRG：應用程式、資料、裝置及對等節點 CRG。

若要啟動 CRG，請完成下列作業：

- 在「iSeries 領航員」中，展開**管理中心** → **叢集**。
- 展開要啟動 CRG 的叢集。
 - 如果您要啟動「裝置 CRG」，請按一下**切換式硬體**，並以滑鼠右鍵按一下要啟動的切換式硬體群組，然後選取**啟動**。
 - 如果您要啟動「應用程式 CRG」，請按一下**切換式軟體**，並以滑鼠右鍵按一下要啟動的切換式軟體產品，然後選取**啟動**。
 - 如果您要啟動「資料 CRG」，請按一下**切換式資料**，並以滑鼠右鍵按一下要啟動的切換式資料群組，然後選取**啟動**。
 - 如果您要啟動「對等節點 CRG」，請按一下**對等節點資源**列出所有的對等節點 CRG，並以滑鼠右鍵按一下要啟動的對等節點 CRG，然後選取**啟動**。

使用 CL 指令及 API

您可以使用下列指令及 API 來啟動 CRG：

- 「啟動叢集資源群組 (STRCRG)」指令
- 「啟動叢集資源群組 (QcstStartClusterResourceGroup) API」

變更叢集資源群組 (CRG) 的回復網域

您可以變更節點在叢集資源群組之回復網域中的角色，以及在回復網域中新增或移除節點。針對裝置叢集資源群組，您也可以變更回復網域之節點的站台名稱與資料埠 IP 位址。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

若要在叢集資源群組 (可切換硬體、可切換軟體或可切換資料) 的回復網域中變更節點的角色，或要在回復網域中新增或移除節點，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 展開含有您要變更回復網域的切換式硬體、軟體或資料的叢集。
4. 展開可切換硬體、軟體或資料。
5. 以滑鼠右鍵按一下可切換硬體、軟體或資料，並選取**內容**。
6. 選取**回復網域**頁。

在「回復網域」頁上按一下「說明」，取得如何變更角色、新增或移除節點的指示。

使用 CL 指令及 API

若要在回復網域中變更節點的角色、新增或移除節點，請使用下列 CL 指令和 API：

- 新增叢集資源群組節點項目 (ADDCRGNODE) 指令
- 新增節點至回復網域 (QcstAddNodeToRcvyDomain) API
- | • 變更叢集資源群組 (CHGCRG) 指令
- | • 變更叢集資源群組 (QcstChangeClusterResourceGroup) API
- 移除叢集資源群組節點項目 (RMVCRGNODE) 指令
- 從回復網域中移除節點 (QcstRemoveNodeFromRcvyDomain) API

相關概念

第 10 頁的『回復網域』

- | 回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

執行切換

執行手動切換會造成現行主要節點切換到備份節點，亦即叢集資源群組之回復網域中定義的備份節點。

發生這種狀況時，在叢集資源群組的回復網域中，節點的現行角色會變更，例如：

- 現行主節點會被指定為最後作用中備份的角色。
 - 現行第一個備份會被指定為主要角色。
 - 後續的備份在備份次序中會向上移動一層。
- | 唯有具「作用中」狀態的主要備份模型 CRG 才容許切換。

註：如果您要在切換式裝置 (亦稱為裝置 CRG) 上執行切換，基於效能原因，您應該同步化使用者設定檔名稱、UID 及 GID。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

若要將回復網域中的資源 (切換式硬體群組、切換式應用程式或切換式資料群組) 從主要節點切換至備份節點，則資源的狀態必須是**已啟動**。

若要在資源上執行切換，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。

2. 展開**叢集**。
3. 展開含有想要的資源的叢集。
4. 按一下**可切換硬體、可切換軟體或可切換資料**。
5. 以滑鼠右鍵按一下想要的資源，然後選取**切換**。

使用叢集 API

您也可以使用下列方法來執行切換：

- 變更叢集資源群組主要節點 (CHGCRGPRI) 指令
- 起始切換 (QcstInitiateSwitchOver) API

相關概念

第 10 頁的『回復網域』

回復網域是叢集節點的子集，這些節點是因為共同的目的 (例如執行回復動作或同步化事件) 而一起組合在叢集資源群組 (CRG) 中。

相關工作

第 18 頁的『切換』

當您將資源存取權從某個伺服器手動切換至另一個伺服器時，會發生**切換**。

同步化使用者設定檔名稱、UID 及 GID

將節點新增到裝置網域

裝置網域是共用裝置資源的叢集中之節點的子集。

在將一個節點新增到裝置叢集資源群組 (CRG) 的回復網域之前，該節點必須先定義成裝置網域的一個成員。將位於裝置 CRG 的回復網域中的所有節點，必須位於相同的裝置網域。一個叢集節點只能屬於一個裝置網域。

若要建立及管理裝置網域，您必須安裝「選項 41 (HA 可切換資源)」，並且在將位於裝置網域中的所有叢集節點上都必須存在有效的授權碼。

使用「iSeries 領航員」

若要在「iSeries 領航員」中將節點新增到裝置網域，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 展開含有您要新增到裝置網域之節點的叢集。
4. 按一下**節點**。
5. 以滑鼠右鍵按一下您要新增到裝置網域的節點，然後選取**內容**。
6. 在**形成叢集**頁上，於**裝置網域**欄位中指定您要新增節點的裝置網域名稱。

使用 CL 指令及 API

您也可以使用下列方法來將節點新增到裝置網域：

- 新增裝置網域項目 (ADDDEVDMNE) 指令
- 新增裝置網域項目 (QcstAddDeviceDomainEntry) API

相關概念

第 15 頁的『裝置網域』

裝置網域是共用裝置資源的叢集中之節點的子集。更特別的是，裝置網域中的節點可參與回復裝置資源之部分集合的切換動作。

相關工作

『從裝置網域中移除節點』

裝置網域是共用裝置資源的叢集中之節點的子集。

從裝置網域中移除節點

裝置網域是共用裝置資源的叢集中之節點的子集。

重要:

從裝置網域中移除節點時，請小心。如果您從裝置網域中移除節點，而該節點是任何獨立磁碟儲存區的現行主要存取點，則移除此節點之後，路那些獨立的磁碟儲存區會保留下來。這表示那些獨立的磁碟儲存區將無法再被裝置網域中的剩餘節點所存取。

節點一旦從裝置網域中移除之後，如果有一或多個現有的叢集節點仍然屬於這個相同的裝置網域，則該節點無法再新增回到相同的裝置網域。若要將節點新增回到裝置網域，您必須：

1. 刪除要新增到裝置網域的節點目前所擁有的獨立磁碟儲存區。
2. 在節點上執行系統重新啟動 (IPL)。
3. 將節點新增到裝置網域。請參閱「將節點新增到裝置網域」。
4. 重建在「步驟 1」中刪除的獨立磁碟儲存區。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

若要在「iSeries 領航員」中移除裝置網域中的節點，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 展開含有您要從裝置網域中移除之節點的叢集。
4. 按一下**節點**。
5. 以滑鼠右鍵按一下您要從裝置網域中移除的節點，然後選取**內容**。
6. 在「形成叢集」頁上，移除**裝置網域**欄位中的項目。

使用 CL 指令及 API

您也可以使用下列方法來從裝置網域中移除節點：

- 移除裝置網域項目 (RMVDEVDMNE) 指令
- 移除裝置網域項目 (QcstRemoveDeviceDomainEntry) API

相關概念

第 15 頁的『裝置網域』

裝置網域是共用裝置資源的叢集中之節點的子集。更特別的是，裝置網域中的節點可參與回復裝置資源之部分集合的切換動作。

相關工作

第 101 頁的『將節點新增到裝置網域』
裝置網域是共用裝置資源的叢集中之節點的子集。
新增硬碟機或磁碟儲存區

系統事件對叢集的影響

結束系統功能的某些指令 (如關閉系統電源 (PWRDWNSYS)、結束系統 (ENDSYS) 及結束子系統 (ENDSBS) 指令) 會驟然結束叢集，導致發生叢集分割。

在 V5R4 中，已加強 PWRDWNSYS、ENDSYS 及 ENDSBS 指令的功能。執行這些指令時，如果節點上的形成叢集作業處於作用中，則會發出結束叢集節點 (QcstEndClusterNode) API。

如果您要讓這些指令完成，應該使用 OPTION(*CNTRLD)，並在 DELAY 參數中指定適當的延遲時間。否則，在將控制傳回終端系統功能之前，可能無法完成「結束叢集節點 API」。

註: 如果使用者指定 **OPTION(*IMMED)**，則在系統結束之前，大約有 30 秒的時間來完成「結束叢集節點 (QcstEndClusterNode) API」。這可能會導致錯誤後移轉而不是結束叢集節點。

建立叢集管理網域

可在「iSeries 領航員」中，或使用 CRTADMDMN (建立叢集管理網域) 指令建立叢集管理網域。

若要建立及管理叢集管理網域，使用者必須獲得授權，才能使用所建立的 CRG、CRG 指令及 QCLUSTER 使用者設定檔。

使用「iSeries 領航員」

若要建立叢集管理網域，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心** → **叢集**。
2. 展開您要新增叢集管理網域的叢集。
3. 以滑鼠右鍵按一下**對等節點資源**，並選取**新建管理網域**。

使用 CL 指令及 API

您可以使用下列指令及 API 來建立叢集管理網域：

- 建立叢集管理網域 (CRTADMDMN) 指令
- 沒有用於建立叢集管理網域的 API。

相關概念

第 72 頁的『叢集指令及 API』

i5/OS 叢集資源服務提供一組控制語言 (CL) 指令、應用程式介面 (API) 及機能，供 iSeries 應用程式提供者或客戶增強其應用程式可用性。

新增受監視的資源項目

您可以將受監視的資源項目新增至代表在節點間共用之資源的叢集管理網域。

若要新增受監視的資源項目，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心** → **叢集**。
2. 展開您要新增受監視之資源項目的叢集。
3. 展開**對等節點資源**，檢視叢集中所有對等節點資源的清單。

- | 4. 展開您要新增受監視之資源項目的叢集管理網域。
 - | 5. 以滑鼠右鍵按一下受監視的資源類型，然後選取**新增受監視的資源項目**。
 - | 6. 為受監視的資源項目選取要監視的屬性，然後按一下**確定**。
- | **註：**如果要新增使用密碼同步化作為受監視資源項目的使用者設定檔，則必須將「保持伺服器安全性 (QRETSVRSEC)」系統值設為 1。

| 使用 CL 指令及 API

- | 您可以使用下列指令及 API 來新增受監視的資源：
- | • 沒有相等於此功能的受支援 CL 指令。QUSRTOOL 檔案庫中提供了不受支援之指令的來源及呼叫處理程式 (CPP)。若要了解此指令來源及 CPP，請查看檔案 QATTINFO 中的成員 QFPADINFO。
 - | • 新增受監視資源項目 (QfpadAddMonitoredResourceEntry) API

| 監視叢集管理網域

| 建立叢集管理網域並新增適當的受監視資源項目之後，叢集管理者應該監視管理網域內的活動，以確保受監視的資源保持一致。

| 如果受監視資源的廣域狀態不一致，管理者應採取必要的步驟來判定資源不一致的原因、更正問題，然後重新同步化資源。

| 如果資源不一致的原因是一或多個節點的更新失敗，則會保留 MRE 資訊，協助您判定失敗的原因。在發生失敗的節點上，MRE 會記載一則訊息，說明更新失敗的原因。在其他節點上，會記載一則訊息，告訴您發生失敗，以及更新失敗的節點清單。

| 判定不一致的原因後，可在發生失敗的節點上執行更新作業，或先結束再重新啟動管理網域，藉以重新同步化資源。

| 如果刪除、重新命名或移動網域中任何節點上的資源，受監視資源的廣域狀態一律會設為不一致。如果是這種情況，則應該移除 MRE，因為叢集管理網域不會再將資源同步化。

| 使用「iSeries 領航員」

- | 若要監視叢集管理網域，請遵循下列步驟：
- | 1. 在「iSeries 領航員」中，展開**管理中心** → **叢集**。
 - | 2. 展開叢集管理網域關聯的叢集。
 - | 3. 展開**對等節點資源**，以滑鼠右鍵按一下**新建管理網域**，然後選取**內容**。作用中叢集管理網域上的資源狀態之可能值如下：
 - | **一致** 在叢集管理網域內的所有作用中節點上，系統監視的所有資源屬性值都相同。
 - | **不一致** 在叢集管理網域內的所有作用中節點上，系統監視的所有資源屬性值並不相同，或者叢集管理網域不處於作用中。
 - | **擱置中** 受監視的屬性值正在叢集管理網域中進行同步化。
 - | **已新增** 受監視資源項目已新增至叢集管理網域中的受監視資源目錄，但尚未同步化。

| 使用 CL 指令及 API

| 您可以使用下列指令及 API 來監視叢集管理網域：

- 沒有相等於此功能的受支援 CL 指令。QUSRTOOL 檔案庫中提供了不受支援之指令的來源及呼叫處理程式 (CPP)。若要了解此指令來源及 CPP，請查看檔案 QATTINFO 中的成員 QFPADINFO。
- 擷取受監視資源資訊 (QfpadRtvMonitoredResourceInfo) API

監督叢集狀態

在必要時採取適當的動作，叢集資源服務便會使用可靠的訊息功能及活動訊號監視，來執行叢集及其元件的基本監視作業。

您也可以手動監督叢集及其元件的狀態。

使用「iSeries 領航員」

這需要安裝及授權「選項 41 (HA 可切換資源)」。

若要在「iSeries 領航員」中監視叢集的狀態：

1. 在「iSeries 領航員」中，展開「管理中心」。
2. 展開叢集。
3. 在「iSeries 領航員」資料夾中尋找想要的叢集，使用「iSeries 領航員」清單中的「狀態」直欄來檢視叢集、其節點及資源的狀態。線上說明包含「狀態」直欄各種可能值的說明。您也可以用滑鼠右鍵按一下叢集的元件，再選取內容來檢視叢集的相關資訊。

使用 CL 指令及 API

您可以使用下列指令和 API 來監督叢集狀態：

叢集資訊

擷取關於叢集的資訊，例如叢集中的節點、每一個節點上使用的配接卡 IP 位址、以及叢集中每一個節點的狀態。

- 顯示叢集資訊 (DSPCLUINF) 指令
- 列出叢集資訊 (QcstListClusterInfo) API
- 列出裝置網域資訊 (QcstListDeviceDomainInfo) API
- 擷取叢集資源服務資訊 (QcstRetrieveCRSInfo) API
- 擷取叢集資訊 (QcstRetrieveClusterInfo) API

叢集資源群組資訊

產生叢集資源群組清單和關於叢集中該叢集資源群組的資訊，例如叢集中每一個 CRG 的主節點名稱。

- 顯示叢集資源群組資訊 (DSPCRGINF) 指令
- 列出叢集資源群組 (QcstListClusterResourceGroups) API
- 列出叢集資源群組資訊 (QcstListClusterResourceGroupInf) API

相關概念

第 25 頁的『可靠的訊息功能』

叢集資源服務的可靠的訊息功能會記錄叢集中之每一個節點，並確定所有節點都有關於叢集資源狀態的一致資訊。

第 23 頁的『活動訊號監督』

活動訊號監視是一種叢集資源服務功能，它可將信號從叢集中的每個節點，傳送到叢集中的所有其他節點，通知它們仍處於作用中，以確定每個節點都在作用中。

叢集效能

變更叢集會影響管理叢集所需的額外執行時間。

形成叢集唯一需要的資源是那些在執行活動訊號監視、管理叢集資源群組及叢集節點，以及處理叢集資源群組與叢集節點之間任何傳訊時必要的資源。一旦您的叢集環境可以運作時，只有在您變更叢集時才會增加額外執行時間。

在正常作業環境下，叢集活動對您的叢集系統應該只產生最小的影響。

相關概念

第 23 頁的『活動訊號監督』

活動訊號監視是一種叢集資源服務功能，它可將信號從叢集中的每個節點，傳送到叢集中的所有其他節點，通知它們仍處於作用中，以確定每個節點都在作用中。

第 127 頁的『一般叢集問題』

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

平衡叢集的網路負載

將您用來連接叢集節點的通訊線路之間的工作分割，可以平衡網路負載。

您愈能平衡工作來降低資源使用率，您的系統就執行愈順暢。

備份節點的 CPU 負載：

充分發揮備份系統的功能，但您要注意如果發生錯誤後移轉，會將額外的工作量傳送到備份節點。

知道什麼東西對企業重要，什麼不重要，是很重要的。如果您非常重要的應用程式失敗，您需要確定備份節點上的中央處理單元 (CPU) 負載不會太高，使重要的應用程式無法執行。

調整叢集效能

因為您的通訊環境中可能存在著明顯的差異，所以您可以調整影響叢集通訊的變數來切合自己的環境。

大部份一般環境應該可以接受預設值。如果您的特定環境不太適合這些預設值，您可以調整叢集通訊來切合您的環境。有兩種可用的調整層次。

基本層次調整

基本層次調整可讓您將調整參數設為一組預先定義的值，來識別高、低和正常逾時及傳訊間隔值。選取正常層次時，則使用叢集通訊效能和配置參數的預設值。選取低層次會造成形成叢集增加活動訊號間隔和各種訊息逾時值。活動訊號愈少和逾時值愈長，叢集對於通訊失敗的敏感度愈低。選取高層次會造成形成叢集減少活動訊號間隔和各種訊息逾時值。活動訊號愈頻繁和逾時值愈短，叢集對於通訊失敗的敏感度愈高。

進階層次調整

進階層次調整還可以在預先定義的值範圍上再調整個別參數。這可以進行更細部的調整來符合您通訊環境的任何特殊情況。若想要進行進階層次的調整，建議您取得 IBM 支援人員或同等技術人員的協助。個別參數的設定不正確，很容易導致效能降低。

相關概念

第 90 頁的『可調整的叢集通訊參數』

「變更叢集資源服務 (QcstChgClusterResourceServices) API」可以調整部分叢集拓樸服務及叢集通訊效能和配置參數，以充份適用於形成叢集的許多獨特應用程式及網路環境。這個 API 適用於執行叢集版本 2 或更高版本的任何叢集。

相關參考

變更叢集資源服務 (QcstChgClusterResourceServices) API

結束叢集工作

請勿嘗試直接結束叢集工作。

如果您需要停止叢集環境中任何正在執行的工作，您應該：

1. 結束叢集節點。
2. 解決問題。
3. 啓動叢集節點。

相關工作

第 96 頁的『結束叢集節點』

停止或結束節點會停止該節點上的叢集資源服務。

第 96 頁的『啓動叢集節點』

啓動叢集節點會在叢集中的節點上啓動叢集資源服務。自叢集的第 3 版開始，節點可以啓動自己，且只要能夠在叢集中找到作用中節點，就能夠重新結合現行的作用中叢集。

資源監視與控制 (RMC)

「資源監視與控制 (RMC)」是管理、監視及操作實體或邏輯系統實體等資源的一般性組織架構。

RMC 是一種向「硬體管理主控台 (HMC)」報告服務事件的通訊機制。如果 RMC 不在作用中，便不會向 HMC 報告服務事件。下列清單說明與 RMC 相關聯的服務：

CAS 常駐程式

用途：用作 RMC 的鑑別伺服器。

工作名稱：QCSTCTCASD

RMC 常駐程式

用途：資源監視程式 (藉由與「資源管理程式」通訊的方式)。

工作名稱：QCSTCTRMCD

SRC 常駐程式

用途：監視其他 RMC 工作的狀態；如果工作非預期地結束，它會重新啓動該特定工作。

工作名稱：QCSTSRCD

資源管理程式 (RM)

「資源管理程式 (RM)」工作會管理並提供 RMC 與實際實體或邏輯實體之間的介面。雖然 RMC 提供基本摘要 (例如，資源類別、資源及屬性) 來代表實體或邏輯實體，但它本身不代表任何實際實體。RM 會將實際實體對應到 RMC 的摘要。下列清單說明 RMC 支援的不同「資源管理程式」：

審核日誌 RM

用途：提供記錄系統作業相關資訊的機能。

工作名稱：QYUSALRMD

CSMAgent RM

用途：提供資源類別來代表「管理伺服器」(即 HMC)。

工作名稱：QYUSCMCRMD

主電腦 RM

用途：提供資源類別來代表個別機器。

工作名稱：QCSTCTHRMD

服務 RM

用途：管理問題資訊，並準備將它遞送至 HMC。

工作名稱：QSVRMSERMD

啓動或結束 RMC

所有 RMC 工作 (包括 RM 工作) 都在 QSYSWRK 子系統中，並且會在啓動子系統時自動啓動。TCP/IP 必須處於作用中，才能完成啓動。「RMC 常駐程式」需要 TCP/IP 處於作用中。如果 TCP/IP 成爲非作用中，「RMC 常駐程式」便會結束。一旦 TCP/IP 重新成爲作用中，「SRC 常駐程式」便會自動重新啓動「RMC 常駐程式」。處於正常狀況下的使用者無需執行任何步驟。如果需要手動啓動 RMC，請執行下列指令：

```
SBMJOB CMD(CALL PGM(QSYS/QCSTCTSRCD)) JOBD(QSYS/QCSTSRCD) PRTDEV(*JOB) OUTQ(*JOB)
USER(*JOB) PRTEXT(*JOB) RTGDTA(RUNPTY50)
```

如果需要手動結束 RMC，請使用 ENDJOB 指令來結束 QCSTSRCD 工作。此指令應該會結束所有 RMC 工作。如果未結束所有工作，請手動結束上面列出的每一個工作。

工作結構和使用者佇列

在管理叢集時，您需要了解工作結構及使用者佇列。

叢集資源服務工作結構

叢集資源服務由一組多緒的工作組成。當伺服器上正在形成叢集時，下列工作會以 QSYS 使用者設定檔在 QSYSWRK 子系統中執行。工作會使用 QDFTSVR 工作說明來執行，但會設定日誌記載層次，以產生工作日誌。

- 叢集控制由一個名爲 QCSTCTL 的工作組成。
- 叢集資源群組管理員由一個名爲 QCSTCRGM 的工作組成。
- 叢集資源群組由每一個叢集資源群組物件的一個工作組成。工作名稱與叢集資源群組名稱相同。這包括叢集管理網域。
- 當回復裝置 CRG 中的一或多個裝置清單項目已設定爲切換時或錯誤後移轉時成爲線上作業，則將提出額外的工作來執行轉接功能。

QCSTCTL 和 QCSTCRGM 工作是重要的叢集工作。亦即，必須執行這些工作才能使節點在叢集中作用。

大部份叢集資源群組 API 會導致提出個別工作，並使用在建立叢集資源群組時指定的使用者設定檔。提出的工作會呼叫叢集資源群組中定義的跳出程式。依預設值，工作會提出到 QBATCH 工作佇列。一般而言，這個工作佇列是用於生產批次作業，將延遲或防止跳出程式完成。爲了讓 API 有效執行，請建立個別的使用者設定檔、工作說明及工作佇列，供叢集資源群組使用。請爲您建立的所有叢集資源群組指定新的使用者設定檔。爲叢集資源群組所定義的回復網域內，所有節點上會處理相同的程式。

- 您可以使用變更叢集回復 (CHGCLURCY) 指令，重新啓動已結束的叢集資源群組工作，而不需要先結束再重新啓動節點上的叢集作業。

叢集 API 對使用者佇列的使用

- 具有結果資訊參數的 API 所執行的函數會以非同步方式操作，一旦 API 完成處理，就會將其結果傳送到使用者佇列。呼叫 API 之前，必須先建立使用者佇列。您可以使用建立使用者佇列 (QUSCRTUQ) API 來建立使

- | 用者佇列。必須將佇列建立成爲金鑰佇列。使用者佇列的金鑰是以使用者佇列項目的格式來描述。使用者佇登
- | 記稱會傳遞到 API。「叢集 API」文件包含的範例是關於如何讓使用者佇列與「叢集 API」搭配使用。

使用「分送資訊 (QcstDistributeInformation) API」時，在節點之間傳送的資訊會存入建立 CRG 時所指定的使用者佇列中。在使用分送資訊 API 之前，必須先由回復網域中所有作用中節點的使用者來建立這個佇列。如需分送資訊佇列何時必須存在的詳細資訊，請參閱「建立叢集資源群組 (QcstCreateClusterResourceGroup) API」。

錯誤後移轉訊息佇列會接收關於錯誤後移轉活動的訊息。

相關概念

第 86 頁的『維護所有節點上的使用者設定檔』

- | 您可以使用兩種機制來維護叢集內所有節點上的使用者設定檔。

第 116 頁的『判斷叢集問題是否存在』

從這裡開始診斷叢集問題。

相關工作

第 133 頁的『從叢集工作失敗中回復』

叢集資源服務工作失敗通常表示有其它問題。

錯誤後移轉訊息佇列

錯誤後移轉訊息佇列會接收關於錯誤後移轉活動的訊息。

使用錯誤後移轉訊息佇列可在錯誤後移轉發生之前，先通知管理者。若此時需要的行爲是防止錯誤後移轉，則這會提供管理者取消錯誤後移轉的能力。

當使用「建立叢集 (QcstCreateCluster) API」來建立叢集資源群組時，會定義錯誤後移轉訊息佇列。也可使用 CL 指令及 API 來修改它，以供變更叢集資源群組使用。錯誤後移轉訊息佇列不能與「iSeries 領航員」叢集管理介面搭配使用。

在「叢集資源群組 API」文件中，可以找到錯誤後移轉訊息佇列進一步的詳細資訊。請參閱下列說明，以取得如何使用錯誤後移轉訊息佇列的詳細資訊。

CL 指令

- CRTCRG (建立叢集資源群組) 指令
- CHGCRG (變更叢集資源群組) 指令

API

- 建立叢集資源群組 (QcstCreateClusterResourceGroup) API
- 變更叢集資源群組 (QcstCreateClusterResourceGroup) API

維護所有節點上的使用者設定檔

- | 您可以使用兩種機制來維護叢集內所有節點上的使用者設定檔。
- | 一種機制是建立叢集管理網域，以監視叢集中各節點上的共用資源。除了使用者設定檔之外，叢集管理網域還可以監視數種類型的資源，因此能輕鬆地管理在各節點上共用的資源。如需這些資源的詳細資訊，請參閱「受監視資源」。更新使用者設定檔時，如果叢集管理網域處於作用中，則會自動將變更傳送至其他節點。如果叢集管理網域不在作用中，則將在啓動叢集管理網域後立即傳送變更。
- | **註：**如果您計劃共用叢集中使用密碼同步化的使用者設定檔，則必須將「保持伺服器安全性 (QRETSVRSEC)」系統值設爲 1。

- | 利用第二種機制，管理者還可以使用「iSeries 領航員」中的「管理中心」，跨多個系統及系統群組執行功能。
- | 這項支援包含一些通用的使用者管理作業，是操作員在其叢集內的多重系統上需要執行的作業。「管理中心」
- | 可以對系統群組執行使用者設定檔功能。建立使用者設定檔時，管理者可以指定在目標系統上執行一個後置傳遞指令。

叢集的備份及回復服務程式

如果您在系統上使用叢集，則建立備份及回復策略來保護資料仍然很重要。

若您規劃使用叢集作為您的備份策略，以便當已備份叢集時，在其他系統當機的同時，您有一個現在可執行的系統，則建議叢集中至少有三個系統。若叢集中有三個系統，則當發生失敗時，您將永遠有一個系統可切換。

保存及復置叢集資源群組

不論叢集是作用中或非作用中，您都可以儲存叢集資源群組。下列限制適用於復置叢集資源群組：

- 若叢集已完成，且該叢集知道叢集資源群組，則您無法復置叢集資源群組。
- 若沒有為叢集配置節點，則您無法復置叢集資源群組。

若叢集作用中、該叢集不知道叢集資源群組、節點在該叢集資源群組的回復網域中、及叢集名稱與叢集資源群組中的名稱相符，則您可以復置叢集資源群組。若叢集已配置但沒有在該節點上作用，及若該節點在該叢集資源群組的回復網域中，則您可以復置叢集資源群組。

為災害預做準備

在災害的情況下，您將需要重新配置您的叢集。為因應這種情況，建議您儲存叢集配置資訊，並保存將該資訊列印出來的輸出品。

1. 進行叢集配置變更之後，使用儲存配置 (SAVCFG) 指令或儲存系統 (SAVSYS) 指令，以使復置的內部叢集資訊是現行的，且與叢集中的其他節點一致。如需執行 SAVCFG 或 SAVSYS 的詳細資訊，請參閱「儲存配置資訊」。
- | 2. 每次您變更叢集配置資訊時，就列印一份該資訊。您可以使用顯示叢集資訊 (DSPCLUINF) 指令來列印叢集配置。假如發生災害，您將需要重新配置您的整個叢集，所以請保存一份要使用的備份磁帶。

相關概念

第 135 頁的『從備份磁帶復置叢集』

在正常作業期間，您不需要從備份磁帶復置。

儲存配置資訊

第 134 頁的『在失去完整系統後回復叢集』

當您的伺服器意外斷電時，若要在失去完整系統後回復整個系統，請將此資訊與「備份及回復」手冊中適當的核對清單一起使用。

『儲存叢集配置』

您可以使用指令來儲存叢集資源群組物件。

第 135 頁的『在災害之後回復叢集』

在遺失您所有節點的災害情況下，您需要重新配置叢集。

相關工作

規劃備份及回復策略

列印系統資訊

儲存叢集配置

您可以使用指令來儲存叢集資源群組物件。

使用 SAVSYS (儲存系統) 指令來儲存整個系統，而不只是已配置的叢集。您可以使用 SAVCFG (儲存配置) 指令來儲存已配置的系統。

SAVOBJ(QUSRSYS/*ALL) OBJTYPE (*CRG)

註：只能儲存現行版次的叢集資源群組物件。

相關工作

第 110 頁的『叢集的備份及回復服務程式』

如果您在系統上使用叢集，則建立備份及回復策略來保護資料仍然很重要。

相關參考

SAVSYS (儲存系統) 指令

SAVCFG (儲存配置) 指令

範例：叢集配置

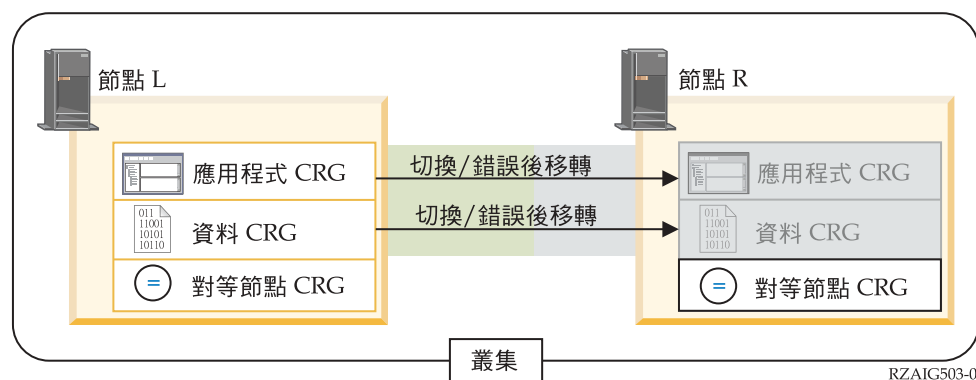
使用這些典型的叢集實作範例來了解何時、為何及如何使用叢集最有利。

範例：簡式雙節點叢集

此配置範例說明包含兩個節點的基本叢集。

此範例配置提供下列內容：

- 單向抄寫與錯誤後移轉
- 雙層環境
- 應用程式與資料同時移動
- 用於離線處理資料的備份
- 對等節點 CRG 上的持續作業



- 在此範例中，節點 L 目前作為兩個叢集資源群組 (應用程式 CRG 和 資料 CRG) 的主節點。它還針對這兩個節點各包含一個提供持續作業的對等節點 CRG。兩個跳出程式會在節點 L 上定期對應用程式 CRG 執行。兩個跳出程式可能同時執行的理由是，如果您呼叫啟動 CRG API，便會啟動一個跳出程式並在應用程式 CRG 作用中同時繼續執行。如果您對應用程式 CRG 呼叫結束 CRG API，則也會啟動另一個跳出程式。節點 R 是每個叢集資源群組回復網域中指定的第一個且唯一的備份節點。與資料 CRG 相關聯的資料，以及與應用程式 CRG 相關聯的適當應用程式資訊，會從節點 L 抄寫到節點 R。如果節點 L 失敗或基於管理原因而需要關閉，便會起始錯誤後移轉或切換，而且節點 R 會變成應用程式及資料 CRG 的主要節點。節點 R 會接管定義給應用程式 CRG 的網際網路通訊協定 (IP) 位址。

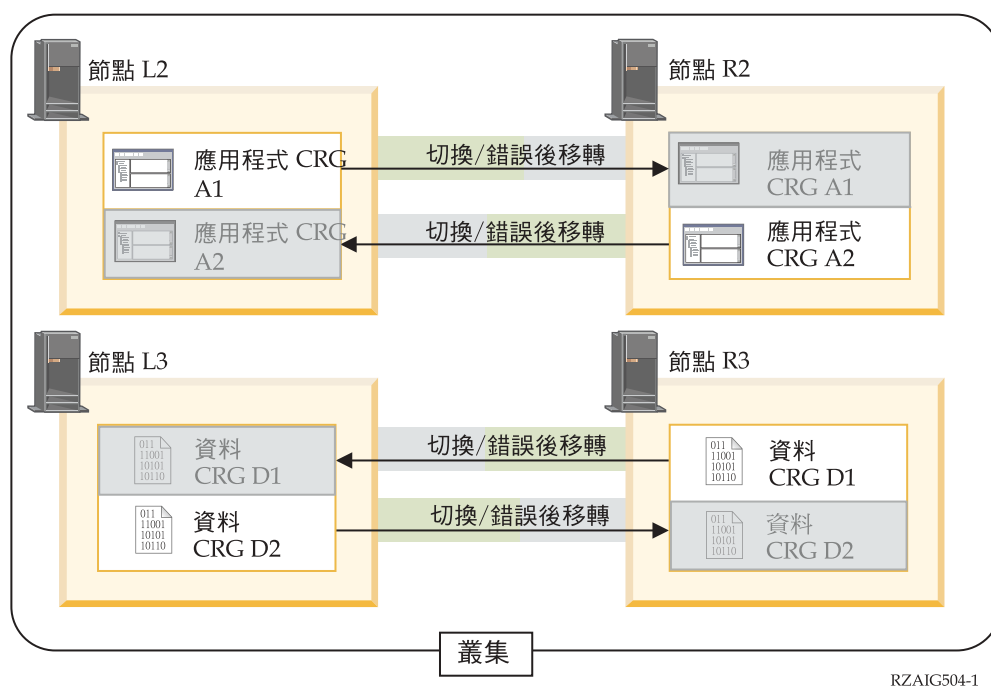
註：當節點 L 關閉時，萬一節點 R 也失敗，則系統可用性會因為沒有備份而受到影響，節點 R 也會因而失敗。當節點 L 回復並重新結合叢集時，會對兩個叢集資源群組建立備份。此時，會從節點 R 抄寫到節點 L。如果您要讓節點 L 重新擔任主要角色，則應執行管理切換。

範例：四節點叢集

使用此範例來建立包含四個節點而較複雜的叢集。

此範例配置提供下列內容：

- 雙向抄寫與錯誤後移轉
- 三層環境
- 應用程式與資料獨立移動
- 備份使用於不同工作量的正常生產



四節點範例顯示 iSeries 叢集可能具有的其他彈性。有兩個應用程式叢集資源群組 (A1 和 A2) 及兩個資料叢集資源群組 (D1 和 D2)。與 D1 相關的資料就是與 A1 相關的應用程式之重要資料。與 D2 相關的資料就是與 A2 相關的應用程式之重要資料。由於這是一個三層環境，應用程式存在於第二層 (節點 L2 和節點 R2)，資料分散在第三層 (節點 L3 和節點 R3)。

叢集資源群組 (CRG)	主要	備份
應用程式 CRG A1	L2	R2
應用程式 CRG A2	R2	L2
資料 CRG D1	R3	L3
資料 CRG D2	L3	R3

這會啟用應用程式與資料層次的相互接管功能。所有四個節點均用於正常生產作業。它們也用於備份叢集中的其它系統。兩個應用程式及其相關資料恆可在此叢集中使用。任何單一節點的故障不會中斷可用性。此外，應用程式層次節點與資料層次節點的同時故障也不會中斷可用性。

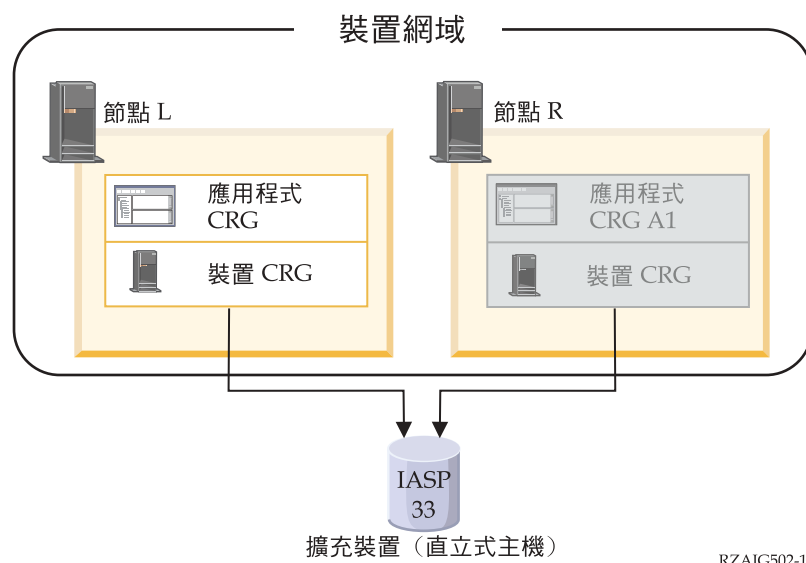
註：在任一案例中，叢集的執行會受到影響，亦即某些叢集資源將無法在節點關閉時被複製。解決此問題的方法是任何重要叢集資源要有不只一個備份。

範例：使用獨立磁碟儲存區的切換式磁碟叢集

使用切換式磁碟技術的叢集提供了抄寫資料的選擇方案。在切換式磁碟叢集中，資料實際上包含在獨立磁碟儲存區（也稱為獨立 ASP）中。

此範例配置提供下列內容：

- 具有閒置待用伺服器的一個可切換獨立磁碟儲存區。此獨立磁碟儲存區包含在可切換硬碟機集合內。
- 雙層環境
- 應用程式與資料同時移動
- 用於與此應用程式資料無關的不同工作量之備份
- 無資料抄寫；此叢集中僅有一個資料副本



RZAIG502-1

在此範例中，節點 L 和節點 R 屬於相同裝置網域。節點 L 目前作為兩個叢集資源群組（應用程式 CRG 和裝置 CRG）的主節點。節點 R 是兩個叢集資源群組的第一個（而且是唯一的）備份。與裝置 CRG 相關的資料包含在可切換資源中，如外部擴充裝置（直立式主機）。與應用程式 CRG 相關的適當應用程式資訊會儲存在相同的直立式主機中，或已從節點 L 抄寫到節點 R。如果節點 L 失敗或基於管理原因而需要關閉，節點 R 變成兩個叢集資源群組的主節點。節點 R 會接管定義給應用程式 CRG 的「網際網路通訊協定 (IP)」位址。節點 R 也會取得已定義給裝置 CRG 的可切換資源所有權。

註：當節點 L 關閉時，萬一節點 R 也失敗，則系統可用性會因為沒有備份而受到影響，節點 R 也會因而失敗。當節點 L 回復並重新結合叢集時，會對兩個叢集資源群組建立備份。如果您要讓節點 L 重新擔任主要角色，則應執行管理切換。

相關概念

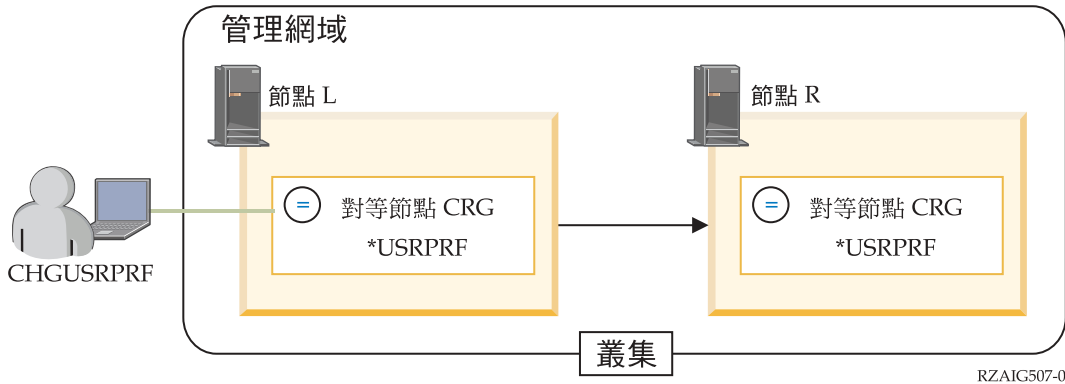
獨立磁碟儲存區配置

範例：管理對等節點資源的叢集管理網域

提供用來監視叢集內的資源之叢集管理網域的範例配置。

此範例配置提供下列內容：

- 雙節點叢集
- 網域節點清單中有兩個節點的叢集管理網域
- 要在網域內同步化之使用者設定檔的「受監視資源項目 (MRE)」。



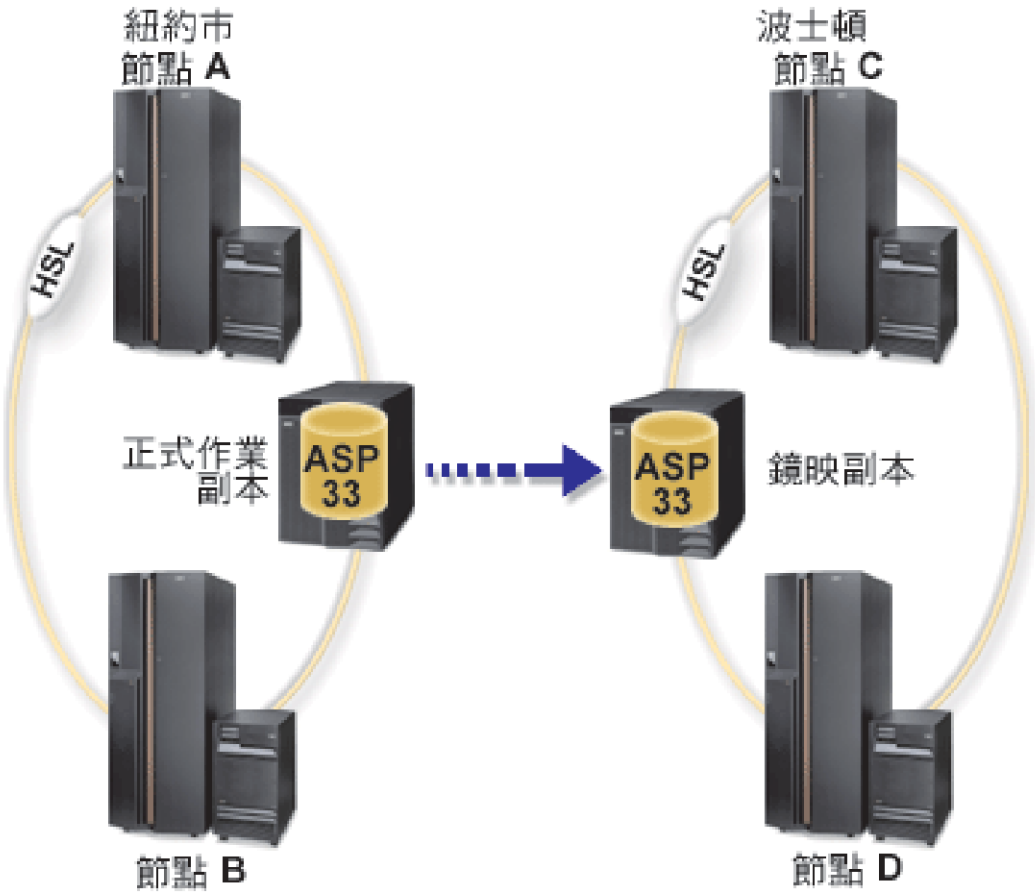
在此範例中，管理者想要確保使用者設定檔在整個叢集中保持一致，所以建立叢集管理網域來監視及同步化使用者設定檔的變更。包含節點 L 及節點 R 的對等節點 CRG 代表叢集管理網域。將受監視資源項目新增至使用者設定檔的叢集管理網域。在此範例中，會在新增 MRE 時指定使用者設定檔的所有屬性。因此，在節點 L 或節點 R 上變更使用者設定檔的任何屬性時，一旦啓動 CRG，變更就會自動傳送至網域中的作用中節點。

下列步驟說明管理者設定此範例要完成的動作：

1. 建立包含節點 L 及 R 的叢集。
2. 在節點 L 及 R 上建立叢集管理網域。
3. 新增代表使用者設定檔的 MRE。
4. 啓動代表叢集管理網域的「對等節點 CRG」。
5. 變更節點 L 或節點 R 上的使用者設定檔。叢集管理網域將自動變更其他節點上的使用者設定檔。如果順利完成變更，受監視資源的廣域狀態將保持一致。

範例：具有地理鏡映的獨立磁碟儲存區

下列範例顯示一種可以配置地理鏡映的方法。「節點 A」與「節點 B」位在紐約市。「節點 C」與「節點 D」位在波士頓。這四個節點都在相同的回復網域中配置。可在節點 A 與 B 之間切換正式作業副本。可在節點 C 與 D 之間切換鏡映副本。由於所有節點都在同一個回復網域中，所以紐約的來源系統也可以與波士頓的目標系統交換角色，讓波士頓主控正式作業副本。



此公司已針對回復網域中的節點定義下列角色：

節點	角色
節點 A	主要
節點 B	備份 1
節點 C	備份 2
節點 D	備份 3

當紐約發生天然災害時，波士頓的「節點 C」會將它的鏡映副本升級為正式作業副本，而成為主要節點。儘管由於紐約發生天然災害而沒有目標節點，因此暫停地理鏡映，但是「節點 C」會成為地理鏡映的來源系統。當紐約站回復時，「節點 A」會成為備份節點，而其先前的正式作業副本則成為鏡映副本。

叢集的疑難排解

尋找針對叢集問題的錯誤復原解決方案。

有時，可能會出現叢集運作不正常的情況。此主題涵蓋您可能遇到的叢集問題之相關資訊。

判斷叢集問題是否存在

從這裡開始診斷叢集問題。

有時，可能會出現叢集未正常運作的情況。當您認為有問題時，您可以使用下列來協助判斷問題是否存在及問題的種類。

- **判斷系統上的叢集是否處於作用中。**

若要判斷叢集資源服務是否作用中，請在 `QSYSWRK` 子系統中尋找兩個工作，即 `QCSTCTL` 和 `QCSTCRGM`。如果這些工作為作用中，表示叢集資源服務也是作用中。您可以使用「iSeries 領航員」中的「工作管理」功能來檢視子系統中的工作，或使用 `WRKACTJOB` (處理作用中的工作) 指令來執行此動作。您也可以使用 `DSPCLUINF` (顯示叢集資訊) 指令來檢視叢集的狀態資訊。

- 叢集資源服務的其他工作可能也是作用中。叢集資源服務工作結構提供如何格式化叢集資源服務工作的相關資訊。

- **尋找指出問題的訊息。**

- 在 `QSYSOPR` 中尋找等待回應的查詢訊息。

- 在 `QSYSOPR` 中尋找指示叢集問題的錯誤訊息。一般而言，這些訊息是在 `CPFBB00` 至 `CPFBBFF` 範圍內。

- 顯示指出叢集問題之訊息的歷程日誌 (`DSPLOG CL` 指令)。一般而言，這些訊息是在 `CPFBB00` 至 `CPFBBFF` 範圍內。

- **查看叢集工作的工作日誌，找出嚴重錯誤。**

這些工作一開始設定的記載層次是 (4 0 *SECLVL)，方便您查閱必要的錯誤訊息。您應該確定這些工作及跳出程式工作皆已適當設定記載層次。如果形成叢集非作用中，您仍然可以在排存檔中尋找叢集工作及跳出程式工作。

- **如果您懷疑發生某種擱置狀況，請查看叢集工作的呼叫堆疊。**

判斷是否有任何程式發生某種 `DEQW` (移出佇列等待)。如果有的話，請檢查每個執行緒的呼叫堆疊，查看任何執行緒在呼叫堆疊中是否有 `getSpecialMsg`。

- **檢查叢集垂直授權內碼 (VLIC) 日誌項目。**

這些日誌項目具有 4800 主碼。

- **使用 `NETSTAT` 指令來判斷通訊環境中是否有任何異常狀況。**

`NETSTAT` 可傳回有關您系統上 `TCP/IP` 網路遞送、介面、`TCP` 連線及 `UDP` 埠狀態的資訊。

- 使用 `Netstat Option 1` (處理 `TCP/IP` 介面狀態) 來確定選擇用於形成叢集的 `IP` 位址顯示「作用中」狀態。同時確定迴路位址 (127.0.0.1) 也是作用中。

- 使用 `Netstat Option 3` (處理 `TCP/IP` 連線狀態) 來顯示埠號 (F14)。本端埠 5550 應處於「接收」狀態。此埠必須透過「作用中工作」清單中現存 `QTOGINTD` (使用者 `QTCP`) 工作顯示的 `STRTCPSVR *INETD` 指令開啓。如果形成叢集在節點上啓動，本端埠 5551 必須開啓且處於 '*UDP' 狀態。如果形成叢集未啓動，不可開啓埠 5551，否則，它會真的造成主旨節點上無法順利開始形成叢集。

- 使用連通測試 (ping)。如果您嘗試啟動叢集節點但無法連通測試 (ping) 它，則會收到內部形成叢集錯誤 (CPFBB46)。

- 使用 **CLUSTERINFO** 巨集來顯示叢集中之節點的叢集資源服務檢視畫面、不同叢集資源群組中的節點，以及目前使用的叢集 IP 位址。

如果叢集未如預期般執行，在這裡找到的分歧處便能協助確定問題區域。如需使用及解譯 **CLUSTERINFO** 巨集結果的詳細資訊，請參閱第 121 頁的『使用 **CLUSTERINFO** 巨集來調查問題』。

相關概念

第 108 頁的『工作結構和使用者佇列』

在管理叢集時，您需要了解工作結構及使用者佇列。

相關工作

檢視子系統中的工作

相關參考

WRKACTJOB (處理作用中的工作)

DSPCLUINF (顯示叢集資訊) 指令

收集叢集的回復資訊

您可以使用叢集 (**WRKCLU**) 指令來收集關於叢集的完整資訊。此資訊可用來協助解析錯誤。

「使用叢集 (**WRKCLU**)」指令可用來顯示及使用叢集節點及物件。執行此指令時，會顯示「使用叢集」顯示畫面。除了顯示叢集中的節點及叢集資訊之外，您還可以使用此指令來檢視叢集資訊，以及收集叢集的相關資料。

若要收集錯誤回復資訊，請完成下列步驟：

1. 在文字介面上，輸入 **WRKCLU OPTION(OPTION)**。您可以指定下列選項，指出要使用的叢集狀態資訊。

***SELECT**

顯示「使用叢集」功能表。

***NODE**

顯示「叢集資訊」畫面，此為叢集中的節點清單。

***CFG** 顯示叢集的完整配置參數。它還能取得叢集資源群組的詳細資訊。

***CRG** 顯示叢集中的叢集資源群組清單。

***SERVICE**

收集叢集中的所有叢集資源服務工作之相關追蹤及除錯資訊。此資訊會對應每個叢集資源服務工作的成員來寫入檔案。只有在服務提供者的指示下才使用此選項。它將顯示傾出叢集追蹤 (**DMPCLUTRC**) 的提示畫面。

使用「傾出叢集追蹤 (**DMPCLUTRC**)」指令來調查問題

使用「傾出叢集追蹤 (**DMPCLUTRC**)」指令來協助判定及解決叢集的問題。

傾出叢集追蹤 (**DMPCLUTRC**) 指令可以協助您判定叢集工作是否已完成，或該工作目前正在處理什麼事。此指令會將叢集相關的追蹤及除錯資訊傾出至檔案中。資訊會在本端的一或多個叢集節點上傾出。此指令可用來傾出一個或所有叢集資源服務 (**CRS**) 工作。每一個傾出的 **CRS** 工作在檔案中都有檔案成員。檔案成員的名稱是 **CRS** 工作的名稱。形成叢集必須處於作用中，才能讓指令產生輸出。僅具有作用中 **CRS** 工作的節點才會有輸出。傾出的資訊來自使用者追蹤，而其他資訊則是從叢集物件獲得。傾出的資訊量視傾出層次而定。不同的傾出層次為基本資訊、錯誤資訊、參考資訊及詳細資訊。傾出層次會決定傳送至檔案的資訊量。在大多數情況

下，IBM 客戶服務代表會根據您的需要，通知您要指定哪個層次；不過，對大部分的疑難排解實務範例來說，只要使用 LEVEL(*ERROR) 就夠了。如果您對哪個層次適合您的狀況有疑問，請聯絡 IBM 客戶服務代表。

解譯追蹤結果

您可以分析追蹤結果，以了解形成叢集所做的事，例如哪個叢集工作導致通訊協定等待。來自使用者追蹤的輸出將包含連續數個等號 (=) 的分隔線。發出 DMPCLUTRC 的次數，將影響檔案中出現的分隔線數量。在相同檔案中可多次呼叫 DMPCLUTRC。最後一組堆疊傾出包含最新的資訊。在某些情況下，CRG 工作可以有兩個群組。每一個群組在檔案中都有單獨的傾出區段。

在下列「傾出叢集追蹤」結果的範例中，在名稱爲 MYCLUSTER 的叢集中有兩個節點 (SYSTEM1 及 SYSTEM2)。它有一個名稱爲 MYCRG 的 CRG。兩個節點都在 MYCRG 的回復網域中。使用者發出 STRCRG CL 指令後，程序要花費很長的時間來傳回結果。在另一個工作站，使用者在指令行介面上輸入 DMPCLUTRC CLUSTER(MYCLUSTER) CRG(*ALL) LEVEL(*ERROR) FILE(MYFILE)。

在此範例中，DMPCLUTRC 指令的輸出位於成員 MYCRG 中名稱爲 MYFILE 的檔案中。爲了協助解釋成員 MYCRG 的內容，已將它分成數個區段。在這些區段中，數字會以括弧來強調顯示，以識別要說明的資訊。這些詳細資訊可協助您疑難排解叢集問題。

註：垂直的省略符號表示已移除部分追蹤，因此未顯示在輸出中。

DMPCLUTRC 結果的第 1 個區段

```
User Trace Dump for job 073586/QSYS/MYCRG. Size: 300K, Wrapped 0 times.
|
| --- 08/22/2005 16:43:32 ---
| (1a) 00000006:658536 Main thread handle 2
| (1b) 00000008:748016 Work thread 1 handle 13
| (1b) 00000007:754576 Work thread 2 handle 11
| --- 08/22/2005 16:46:04 ---
| 00000008:269608 CSTDAMBR 1115: WaitForMsg 4 1005 CPFBB3C
| --- 08/22/2005 16:48:17 ---
| 00000006:925112
| (1c) DMPCLUTRC Node SYSTEM1 Group MYCRG
| =====
```

第一個區段包含叢集工作的緒號及控點。叢集工作可擁有兩個或更多個緒。在此範例中，有一個主緒 (1a) (此爲所有工作進入的位置)，以及兩個工作緒 (1b)。此區段還包含此追蹤來自哪個系統，以及它專屬於哪個叢集工作 (1c) 的相關資訊。

DMPCLUTRC 結果的第 2 個區段

```
00000006:925168 Stack Dump For Target Thread: Handle 2 (0x00000002)
| 00000006:925192 Stack:
| (2aa) Main Thread Stack MYCRG
| 00000006:925256 Stack: Library / Program Module Stmt Procedure
| 00000006:933432 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 0 : _CXX_PEP_Fv
| 00000006:933488 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 46 : main
| 00000006:933536 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 65 : completeStartup_FP8CstDAMbr
| 00000006:933584 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 26 : mainQueueProcessLoop_FP8Cs
| DAMbr
| 00000006:933616 Stack: QSYS / QCSTCMN CSTDAMBR 57 : processQueueMsgs__8CstDAMbrF
| Q2_8CstDAMbr13CstQueueIndex
| 00000006:933664 Stack: QSYS / QCSTCMN CSTDAMBR 53 : processMsg__8CstDAMbrFP6CstM
| sg
| 00000006:933712 Stack: QSYS / QCSTCMN CSTDAMBR 17 : callFnPtr__8CstDAMbrFPQ2_8Cs
| tDAMbr19MsgFunctionPtrEntryP6
| 00000006:933744 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 94 : crgDump_FP6CstMsgP8CstDAMbr
| 00000006:933792 Stack: QSYS / QCSTCMN CSTACK 95 : CstAckQueryMsg
| 00000006:933832 Stack: QSYS / QP0ZCPA QP0ZDBG 3 : Qp0zDumpTargetStack
```

```

00000006:933864 Stack: QSYS      / QP0ZSCPA  QP0ZSDBG  12   : Qp0zSUDumpTargetStack
| 00000006:934016 Stack: Exception      In Stack Dump Code
| 00000006:934040 Stack: thread is likely terminated or no longer running the same code as the
| captured stack
| 00000006:934080
| (2a) Work Thread Index 1 Group MYCRG      Last or current values
| (2e) 00000006:934112 Request handle 8E3E1002 EE3218A1 824F0004 AC000456
| (2c) 00000006:934136 SPI name QcstStartClusterResourceGroup
| 00000006:934160 (2g) POF 10, Completed ack round 1 (2i)
| 00000006:934176 (2o) In waitForJobEnd QDFTJOB  MYCLUSTER      073590
| 00000006:934216 Node      Ack Status      POF      (2bb) Nack Msg Id
| 00000006:934240 (2n) SYSTEM1 (2cc) Ready
| 00000006:934272 SYSTEM2 Ack      10 (2k)
| 00000006:934296 Messages
| 00000006:934320 Stack Dump For Target Thread: Handle 13 (0x0000000d)
| 00000006:934344 Stack: Work Thread 1 Stack MYCRG
| 00000006:934792 Stack: Library / Program Module Stmt Procedure
| 00000006:934840 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 9 : workThreadRoutine_FPv
| 00000006:934888 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 28 : workQueueProcessLoop_FP8Cst
| DAMbr
| 00000006:941688 Stack: QSYS / QCSTCMN CSTDAMBR 57 : processQueueMsgs__8CstDAMbrF
| Q2_8CstDAMbr13CstQueueIndex
| 00000006:941696 Stack: QSYS / QCSTCMN CSTDAMBR 33 : processMsg__8CstDAMbrFP6CstM
| sg
| 00000006:941712 Stack: QSYS / QCSTCMN CSTDAMBR 17 : callFnPtr__8CstDAMbrFPQ2_8Cs
| tDAMbr19MsgFunctionPtrEntryP6
| 00000006:941728 Stack: QSYS / QCSTCMN CSTACK 3 : CstStripOffHeaderMsgPart
| 00000006:941736 Stack: QSYS / QCSTCMN CSTDAMBR 53 : processMsg__8CstDAMbrFP6CstM
| sg
| 00000006:941752 Stack: QSYS / QCSTCMN CSTDAMBR 17 : callFnPtr__8CstDAMbrFPQ2_8Cs
| tDAMbr19MsgFunctionPtrEntryP6
| 00000006:970888 Stack: QSYS / QCSTCRGS2 CSTCRGSS 39 : startCrg
| 00000006:970912 Stack: QSYS / QCSTCRGS2 CSTCRGSS 344 : doMessageProcessing_FP6CstM
| sgP8CstDAMbr
| 00000006:970928 Stack: QSYS / QCSTCRGS2 CSTCRGSS 57 : doExitPgmPhase_FP6CstMsgP8C
| stDAMbr
| 00000006:981984 Stack: QSYS / QCSTCMN CSTDAMBR 52 : waitForJobEnd__8CstDAMbrFPA2
| 6_ci
| 00000006:982000 Stack: QSYS / QCSTCMN CSTDAMBR 73 : waitForSpecialMsg__8CstDAMbr
| FP17CstSpecialMsgListPA8_ciT3
| 00000006:982016 Stack: QSYS / QC2UTIL1 QC2MI3 1 : (2dd) deq
| 00000006:982136 Stack: Exception      In Stack Dump Code
| 00000006:982136 Stack: thread is likely terminated or no longer running the same code as the
| captured stack
| 00000006:982160
| (2b) Work Thread Index 2 Group MYCRG      Last or current values
| (2f) 00000006:982176 Request handle D9C3C8C3 E2E3F5F2 0003 0000
| (2cc) 00000006:982176 SPI name
| 00000006:982184 (2h) POF 0, (2d) Completed ack (2j) round 0
| 00000006:982184 In getNextWorkMsg
| 00000006:982208 Node      Ack Status      POF      Nack Msg Id
| (21) 00000006:982208 SYSTEM1 Ready
| (21) 00000006:982232 SYSTEM2 Ready
| 00000006:982248 Messages
| 00000006:982256 Stack Dump For Target Thread: Handle 11 (0x0000000b)
| 00000006:982256 Stack: Work Thread 2 Stack MYCRG
| 00000006:982344 Stack: Library / Program Module Stmt Procedure
| 00000006:982360 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 9 : workThreadRoutine_FPv
| 00000006:982376 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 28 : workQueueProcessLoop_FP8Cst
| DAMbr
| 00000006:982392 Stack: QSYS / QCSTCMN CSTDAMBR 51 : processQueueMsgs__8CstDAMbrF
| Q2_8CstDAMbr13CstQueueIndex
| (2m) 00000006:982400 Stack: QSYS / QCSTCMN CSTDAMBR 105 : getNextWorkMsg__8CstDAMbrFv
| 00000006:982416 Stack: QSYS / QC2UTIL1 QC2MI3 1 : deq
| 00000006:982480 Stack: Exception      In Stack Dump Code
| 00000006:982480 Stack: thread is likely terminated or no longer running the same code as the
| captured stack

```

第二個區段包含作為叢集工作一部分之每個緒的呼叫堆疊。大部分是主緒將顯示剛完成的 DMPCLUTRC (2aa)。工作緒 (2a 及 2b) 包含的追蹤資訊將協助判定叢集工作發生的狀況。此區段包含呼叫堆疊的詳細資訊，例如：SPI 名稱 (2c)、已完成的認可 (ACK) (2d)、相關 API 的要求控點 (2e) 或最後完成的要求控點 (2f)、目前的失敗點 (POF) (2g 及 2h)、目前一輪的認可 (ACK) (2i 及 2j) 及已認可 (ACK) 的節點 (2k 及 2l)。

目前的失敗點 (POF) 是代表程式碼中目前通訊協定位置的內部值，且不一定表示已發生失敗。Ack 表示節點已成功完成通訊協定的這個部分，且正在等待所有其他節點認可 (ACK) 或 Nack。Nack 表示節點無法成功完成通訊協定的這個部分，且正在等待所有其他節點回應。會在下一個直欄 (2bb) 中給定 Nack 的訊息 ID。這是傳送至發送端之 RIQ 的相同訊息。如果節點在通訊協定期間失敗，其狀態會顯示為「失敗」，且不一定會視為 Nack (視通訊協定及節點而定)。「非作用中」的 Ack 狀態表示節點未參與通訊協定。值 Ready 表示節點尚未回應。當緒處於 getNextWorkMsg (2m) 時，表示緒正在等待工作執行。

從呼叫堆疊底端開始由下往上讀取程序名稱。此範例檔案包含 deq (2dd)，具有 waitForSpecialMsg、waitForJobEnd 及 doExitPgmPhase。這表示通訊協定要等待跳出程式完成後，才能繼續處理。從 Ack Status (2k)，可以判定通訊協定正在等待哪個節點。在此範例中，正在等待節點 SYSTEM1 (2n)。完整的工作名稱 (2e) 表示系統正在等待的工作。一旦決定工作名稱，您就可以使用工作來疑難排解延遲的原因。某些可能的原因是工作仍在工作佇列中等待、工作正在執行但需要時間進行處理，或者工作正在等待已鎖定的物件。

在此範例中，通訊協定正在等待跳出程式完成。要判定通訊協定是否正在等待跳出程式或轉接工作完成還有更簡單的方式，那就是查看第一個區段是否在 waitForJobEnd (2o) 中。正在等待的工作名稱會在同一行上。如此就不需要查看堆疊。

DMPCLUTRC 結果的第 3 個區段

```
5722SS1 V5R4M0 060210 AS/400 DUMP 073586/QSYS/MYCRG 08/22/05 16:48:18 PAGE 1
DMPYSOBY PARAMETERS
(3a)OBJ- MYCRGAIX CONTEXT- QTEMP
TYPE- *ALL SUBTYPE-*ALL
OBJECT TYPE- INDEX *CRGM
NAME- MYCRGAIX TYPE- 0E SUBTYPE- A5
LIBRARY- QTEMP 006B8A19B00C9E807000 TYPE- 04 SUBTYPE- C1
CREATION- 08/22/05 16:43:32 SIZE- 0000007000
ATTRIBUTES- 0000 ADDRESS- C7FE286F04 000000
.
.
.POINTERS-
NONE
OIR DATA-
NONE
END OF DUMP
***** END OF LISTING *****
```

所顯示的第三個區段是包含叢集工作相關資訊的內部物件。在此範例中，它是名稱為 MYCRGAIX (3a) 的內部索引。此處的資訊與上面的第 2 個區段相比更易於閱讀。

DMPCLUTRC 結果的第 4 個區段

```
5722SS1 V5R4M0 060210 AS/400 DUMP 073586/QSYS/USER 08/22/05 16:48:18 PAGE 1
DMPYSOBY PARAMETERS
(4a)OBJ- MYCRGTQ CONTEXT- QTEMP
TYPE- 0A SUBTYPE-EF
OBJECT TYPE- QUEUE *QTQ
NAME- MYCRGTQ TYPE- 0A SUBTYPE- EF
LIBRARY- QTEMP 006B8A19B00C9E807000 TYPE- 04 SUBTYPE- C1
CREATION- 08/22/05 16:43:32 SIZE- 000002C000
ATTRIBUTES- 0000 ADDRESS- CC6765CAA2 000000
QUEUE ATTRIBUTES-
.
```

```

| .
| .
| .POINTERS-
| 000010 SPP 1A EF MYCRG QSYS 073586 00002160 0000
| 000020 SPP 1A EF MYCRG QSYS 073586 00001540 8000
| 000030 SPP 1A EF MYCRG QSYS 073586 000016E0 0000
| 000040 SPP 1A EF MYCRG QSYS 073586 00001690 0000
| 000050 SPP 1A EF MYCRG QSYS 073586 000016A0 0000
| 000070 SPP 1A EF MYCRG QSYS 073586 00002160 0000
| OIR DATA-
| NONE
| END OF DUMP
|
| * * * * * E N D O F L I S T I N G * * * * *

```

所顯示的第四個區段稱為追蹤佇列 (4a)。在此狀況下稱為 MYCRGTQ。這包含叢集讓此工作執行的內容，以及每一個工作如何回應要求的相關資訊。

註：每一個訊息都未在此完整說明。

DMPCLUTRC 結果的第 5 個區段

```

| 5722SS1 V5R4M0 060210 AS/400 DUMP 073586/QSYS/MYCRG 08/22/05 16:48:18 PAGE 1
| DMPYSOBJ PARAMETERS
| (5a) OBJ- MYCRG CONTEXT- QUSRSYS
| TYPE- 19 SUBTYPE-2C
| OBJECT TYPE- SPACE *CRG
| NAME- MYCRG TYPE- 19 SUBTYPE- 2C
| LIBRARY- QUSRSYS TYPE- 04 SUBTYPE- 01
| CREATION- 08/17/05 07:16:40 SIZE- 0000002000
| OWNER- MYCLUSTER TYPE- 08 SUBTYPE- 01
| ATTRIBUTES- 0800 ADDRESS- 1EC687A1F3 000000
| SPACE ATTRIBUTES-
| .
| .
| .
| END OF DUMP
|
| * * * * * E N D O F L I S T I N G * * * * *

```

第五個區段包含 CRG 物件的相關資訊 (5a)。

使用 CLUSTERINFO 巨集來調查問題

CLUSTERINFO 巨集顯示叢集資源服務所包含之有關節點、CRG 及作用中叢集 IP 位址的資訊。

CLUSTERINFO 巨集會建立現行叢集之相關資訊的瞬像。此指令會瀏覽叢集物件，並在本端節點上建立叢集的說明。CLUSTERINFO 巨集為不同的叢集物件提供飛行記錄器，並且可以協助判定叢集中問題的來源。若要存取 CLUSTERINFO 巨集，請完成下列步驟：

1. 在文字介面上，輸入 STRSST。
2. 以您的「服務工具」使用者設定檔登入。
3. 在「啟動服務工具」上，選取「選項 1 (啟動服務工具)」。
4. 選取「選項 4 (顯示/變更/傾出)」。
5. 選取「選項 2 (傾出至印表機)」。
6. 選取「選項 2 (授權內碼 (LIC) 資料)」。
7. 選取「選項 14 (進階分析)」。
8. 在 CLUSTERINFO 巨集選項前輸入 1。按 Enter 鍵。

一旦 CLUSTERINFO 巨集顯示，請使用 -H 選項來顯示此巨集可用的所有選項之說明。下列用法圖解說明 CLUSTERINFO 巨集可用的每個選項：

表 21. CLUSTERINFO 巨集的選項

選項	說明
-H	顯示選項的說明畫面
-A	顯示所有資訊
-FR	顯示飛行記錄器項目
-HB	顯示活動訊號資訊
-PERF	顯示效能計數器
-Q	顯示送出訊息佇列狀態
-G	取消抑制顯示所有 CC 廣播群組
-T	顯示調整參數
-M	顯示所有 DA 的矩陣
-DP	顯示 dataPort 資訊

解譯 CLUSTERINFO 巨集結果

在此範例中，指定 -A 選項，以便傾出所有欄位。飛行記錄器是用於除錯的主要工具。請注意，在結束或刪除叢集節點時，會刪除這些飛行記錄器。爲了分析問題，必須在結束或刪除叢集前執行 CLUSTERINFO 巨集。在某些情況下，可以在刪除或結束叢集時將飛行記錄器寫入 vlog。飛行記錄器會記錄影響叢集結構及效能的各種事件。此資訊涵蓋的範圍不包括對飛行記錄器的資料提供詳細的解釋。

- 註：垂直的省略符號表示已移除部分結果，因此未顯示在輸出中。

CLUSTERINFO 巨集結果的第 1 個區段

```

DISPLAY/ALTER/DUMP  CLUSTERINFO -NEW2  08/23/05  13:36:37  PAGE  1
Running macro: CLUSTERINFO  -A
Use -H for command information
Cluster Name : MYCLUSTER
Local Node Name: SYSTEM1
CC/CTS Version : 5
Macro Timestamp: 08/23/05 13:36:37.079

```

第 1 個區段包含有關叢集的一般資訊，例如，叢集名稱、叢集版本及產生報告時的時間戳記。在此範例中，叢集名稱爲 MYCLUSTER，本端節點名稱爲 SYSTEM1。

CLUSTERINFO 巨集結果的第 2 個區段

```

Cluster Object Addresses
CstcClusterServices Address: DBF08681C9161580
Cluster Address : FC5B04B0D4001000
Cluster Task Address : B00010000E932000
Cluster Task Q Address : DBF08681C9169A00
Clue Group Services Address: CDAB6D0339001000
CC Services Address : FC5B04B0D4008000

```

第 2 個區段提供主要叢集物件位置的指標。

CLUSTERINFO 巨集結果的第 3 個區段

```
Message Statistics
Number of non-fragmented messages: 250
Number of fragmented messages   : 1
Number of fragments              : 7
Number of acks                   : 148
```

第 3 個區段包含叢集的傳訊統計值，例如片段數及認可數 (ack)。

CLUSTERINFO 巨集結果的第 4 個區段

```
Node Map
Node ID : SYSTEM1
GenesisSubnetId : 9.5.251.0
CCNode *   : FC5B04B0D4007000
CCSrvNode *: FC5B04B0D404F000
Adapter 1  : 9.5.251.46 Primary
Status    : 0x01 Reachable
Line Type : 0x09 Ethernet
Node ID : SYSTEM2
GenesisSubnetId : 9.5.251.0
CCNode *   : FC5B04B0D4060000
CCSrvNode *: FC5B04B0D4061000
Adapter 1  : 9.5.251.47 Primary
Status    : 0x01 Reachable
Line Type : 0x09 Ethernet
```

第 4 個區段列出節點對映表中的所有現行作用中叢集節點。在此範例中有兩個作用中節點 SYSTEM1 及 SYSTEM2。

CLUSTERINFO 巨集結果的第 5 個區段

```
Subnet Map
Subnet ID: 127.0.0.0
CCSubnet *   : FC5B04B0D4006000
CCSrvSubnet* : FC5B04B0D400C000
Retries      : 0
Msg Timeouts : 0
Bad Msg Counter : 0
Failed Default Address: 0
Subnet ID: 226.5.5.5
CCSubnet *   : FC5B04B0D405B100
CCSrvSubnet* : FC5B04B0D405C000
Retries      : 0
Msg Timeouts : 0
Bad Msg Counter : 0
Failed Default Address: 0
```

第 5 個區段包含叢集中所有子網路物件的清單。

CLUSTERINFO 巨集結果的第 6 個區段

```
Group Map
Group ID: 0x0000000000000001
Name : CTS
CCGroup *   : FC5B04B0D405FF00
CCSrvGroup *: FC5B04B0D4064B00
Member Nodes
SYSTEM1
SYSTEM2
Group ID: 0x0000000000000002
Name : CTS
CCGroup *   : FC5B04B0D4055100
CCSrvGroup *: FC5B04B0D4055200
Member Nodes
```

```
SYSTEM1
SYSTEM2
```

```
.
```

第 6 個區段列出所有的現行叢集群組。每個群組都稱為分散式活動群組。這些群組可用於叢集中每個作用中節點上的群組之間的通訊。大部分的群組處理是針對授權內碼 (LIC)。可由群組名稱 CTS 及 BADA 來識別。您還將看到群組：CCTL (作業系統中的 QCSTCTL 工作)、CRGM (作業系統中的 QCSTCRGM 工作) 及每個叢集資源群組 (CRG) 工作。CRG 工作的群組沒有群組名稱。每個群組都具有成員節點。成員節點是此群組中一起進行通訊的節點。

CLUSTERINFO 巨集結果的第 7 個區段

```
Partition Map
Partition Map is empty
```

第 7 個區段包含 SLIC 分割區清單中所有節點的清單。

註：這與 XPF 已分割節點的概念不同。

CLUSTERINFO 巨集結果的第 8 個區段

```
CTS Client List
CTS Client List is empty
```

第 8 個區段包含所有項目的叢集用戶端 (例如，資料埠) 的清單。

CLUSTERINFO 巨集結果的第 9 個區段

```
Flight Recorder      : CSTCSVFR
Flight Recorder Address: DBF08681C9161620
```

第 9 個區段包含 IPL 之前一直在系統上的叢集服務飛行記錄器 (CSTCSVFR)。

CLUSTERINFO 巨集結果的第 10 個區段

```
Message Statistics
Number of non-fragmented messages: 250
Number of fragmented messages    : 1
Number of fragments              : 7
Number of acks                   : 148
Time Stamp: 08/18/05 14:00:15.329
Trace Point: 0x0010 CstcClusterServicesTracePtCreatedFlightRecorder
C3D9C5C1E3C5C6D9 <CREATEFR>
Time Stamp: 08/22/05 16:43:28.912
Trace Point: 0x0020 CstcClusterServicesTracePtCreatedClusterObject
D4D6D9C5E8404040 4040C5F8D3770500 <MYCLUSTER E8L...>
1000 <.. >
Time Stamp: 08/23/05 13:33:40.935
Trace Point: 0x0030 CstcClusterServicesTracePtDeletedClusterObject
D4D6D9C5E8404040 404040E2E3 <MYCLUSTER ST >
Time Stamp: 08/23/05 13:33:41.204
Trace Point: 0x0030 CstcClusterServicesTracePtDeletedClusterObject
C3D4D7E3 <CMPT >
Time Stamp: 08/23/05 13:33:55.122
Trace Point: 0x0020 CstcClusterServicesTracePtCreatedClusterObject
D4D6D9C5E8404040 4040FC5B04B0D400 <MYCLUSTER ....M.>
1000 <.. >
```

第 10 個區段包含 CSTCCCFR 的飛行記錄器。此叢集飛行記錄器會一直在系統上，直到此節點上的叢集結束為止。

CLUSTERINFO 巨集結果的第 11 個區段

```
Flight Recorder      : CSTCCLFR
Flight Recorder Address: FC5B04B0D4001E80
-----
Time Stamp: 08/23/05 13:33:54.944
Trace Point: 0x1010 CstcClusterTracePtCreatedSubnetObject
7F000000FC5B04B0 D4006000 <.....M.-. >
Time Stamp: 08/23/05 13:33:55.062
Trace Point: 0x1000 CstcClusterTracePtCreatedNodeObject
C3E2E3D9D9C3C8C3 E2E3F5F2FC5B04B0 <CSTRSYSTEM1....>
D4007000 <M... >
Time Stamp: 08/23/05 13:33:55.122
Trace Point: 0x1020 CstcClusterTracePtCreatedMCGroupObject
0000000000000001 00000000D9C3C8C3 <.....RCHC>
E2E3F5F2 <ST52 >
.
.
.
```

第 11 個區段包含叢集通訊飛行記錄器 (CSTCCLFR)。此叢集飛行記錄器會一直在系統上，直到此節點上的叢集結束為止。

CLUSTERINFO 巨集結果的第 12 個區段

```
Flight Recorder      : CSTCCCFR
Flight Recorder Address: FC5B04B0D4006380
-----
Time Stamp: 08/23/05 13:33:55.080
Trace Point: 0x3000 CstcCCScamTracePtScamOpen
FC5B04B0D400E480 0000000000000000 <....M.U.....>
Time Stamp: 08/23/05 13:33:55.097
Trace Point: 0x3010 CstcCCScamTracePtScamBind
FC5B04B0D400E480 0000000000000000 <....M.U.....>
Time Stamp: 08/23/05 13:33:55.100
Trace Point: 0x3000 CstcCCScamTracePtScamOpen
FC5B04B0D400E480 0000000000000000 <....M.U.....>
D6E4E3 <OUT >
Time Stamp: 08/23/05 13:33:55.100
Trace Point: 0x3010 CstcCCScamTracePtScamBind
FC5B04B0D400E480 0000000000000000 <....M.U.....>
.
.
.
```

第 12 個區段包含線索飛行記錄器 (CSTCCCFR)，它會一直在系統上，直到此節點上的叢集結束為止。

CLUSTERINFO 巨集結果的第 13 個區段

```
Time Stamp: 08/23/05 13:33:55.201
C3A2A385C7E27A7A C3A2A385C7E24082 <CsteGS::CsteGS b>
85878995A2 <egins >
Time Stamp: 08/23/05 13:33:55.201
C3A2A385C4C14083 9695A2A399A483A3 <CsteDA construct>
85847A40C2C1C4C1 404040404040 <ed: BADA >
Time Stamp: 08/23/05 13:33:55.201
C3A2A385C7E27A7A C3A2A385C7E24081 <CsteGS::CsteGS a>
8484408281848140 A39640C4C16D9389 <dd bada to DA_li>
A2A3 <st >
.
.
.
```

第 13 個區段顯示傳送佇列及作用中訊息佇列的內容。如果在擴充期間內此區段不是空的，表示叢集有問題。

CLUSTERINFO 巨集結果的第 14 個區段

Flight Recorder : CSTECLF2
Flight Recorder Address: CDAB6D0339001300

```
-----  
Time Stamp: 08/23/05 13:33:55.201  
C3A2A385C4C17A7A C3A2A385C4C16B40 <CsteDA::CsteDA, >  
83998581A385D4C3 C79996A49740C9C4 <createMCGroup ID>  
407E40F0A7F1F5 < = 0x15 >  
Time Stamp: 08/23/05 13:33:55.209  
C3A2A385E2C3D985 977A7A84859389A5 <CsteSCRep::deliv>  
85994094A287E3A8 97857EF0A7F140A2 <er msgType=0x1 s>  
A482E3A897857EF0 A7F240C4C17EC2C1 <ubType=0x2 DA=BA>  
C4C1404040404040 <DA >  
Time Stamp: 08/23/05 13:33:55.209  
C3A2A385C4C17A7A A58985A66B409985 <CsteDA::view, re>  
9496A585D4C3C799 96A497D485948285 <moveMCGroupMembe>  
99A240C9C4407E40 F0A7F1F540969384 <rs ID = 0x15 old>  
6D959684856D9389 A2A340A289A98540 <_node_list size >  
7E40F0A7F0 <= 0x0 >  
.  
.  
.
```

第 14 個區段包含飛行記錄器資訊。

CLUSTERINFO 巨集結果的第 15 個區段

Message Queues

Send Queues:

```
Send Queue: 00 Messages: 00 MessageQueue*: FC5B04B0D400BF80  
Send Queue: 01 Messages: 00 MessageQueue*: FC5B04B0D400DF80  
Send Queue: 02 Messages: 00 MessageQueue*: FC5B04B0D400E600  
Send Queue: 03 Messages: 00 MessageQueue*: FC5B04B0D400E680  
Send Queue: 04 Messages: 00 MessageQueue*: FC5B04B0D400E700  
Send Queue: 05 Messages: 00 MessageQueue*: FC5B04B0D400E780  
Send Queue: 06 Messages: 00 MessageQueue*: FC5B04B0D400E800  
Send Queue: 07 Messages: 00 MessageQueue*: FC5B04B0D400E880  
Send Queue: 08 Messages: 00 MessageQueue*: FC5B04B0D400E900  
Send Queue: 09 Messages: 00 MessageQueue*: FC5B04B0D400E980  
Send Queue: 10 Messages: 00 MessageQueue*: FC5B04B0D400EA00  
Send Queue: 11 Messages: 00 MessageQueue*: FC5B04B0D400EA80  
Send Queue: 12 Messages: 00 MessageQueue*: FC5B04B0D400EB00  
Send Queue: 13 Messages: 00 MessageQueue*: FC5B04B0D400EB80  
Send Queue: 14 Messages: 00 MessageQueue*: FC5B04B0D400EC00  
Send Queue: 15 Messages: 00 MessageQueue*: FC5B04B0D400EC80  
Send Queue: 16 Messages: 00 MessageQueue*: FC5B04B0D400ED00  
Send Queue: 17 Messages: 00 MessageQueue*: FC5B04B0D400ED80  
Send Queue: 18 Messages: 00 MessageQueue*: FC5B04B0D400EE00  
Send Queue: 19 Messages: 00 MessageQueue*: FC5B04B0D400EE80
```

Active Message Queues:

```
Active Message Queue: 00 Messages: 00 MessageQueue*: FC5B04B0D4008570  
Active Message Queue: 01 Messages: 00 MessageQueue*: FC5B04B0D4008640  
Active Message Queue: 02 Messages: 00 MessageQueue*: FC5B04B0D4008710  
Active Message Queue: 03 Messages: 00 MessageQueue*: FC5B04B0D40087E0  
Active Message Queue: 04 Messages: 00 MessageQueue*: FC5B04B0D40088B0  
Active Message Queue: 05 Messages: 00 MessageQueue*: FC5B04B0D4008980  
Active Message Queue: 06 Messages: 00 MessageQueue*: FC5B04B0D4008A50  
Active Message Queue: 07 Messages: 00 MessageQueue*: FC5B04B0D4008B20  
Active Message Queue: 08 Messages: 00 MessageQueue*: FC5B04B0D4008BF0  
Active Message Queue: 09 Messages: 00 MessageQueue*: FC5B04B0D4008CC0  
Active Message Queue: 10 Messages: 00 MessageQueue*: FC5B04B0D4008D90  
Active Message Queue: 11 Messages: 00 MessageQueue*: FC5B04B0D4008E60  
Active Message Queue: 12 Messages: 00 MessageQueue*: FC5B04B0D4008F30  
Active Message Queue: 13 Messages: 00 MessageQueue*: FC5B04B0D4009000  
Active Message Queue: 14 Messages: 00 MessageQueue*: FC5B04B0D40090D0  
Active Message Queue: 15 Messages: 00 MessageQueue*: FC5B04B0D40091A0
```

```
Active Message Queue: 16 Messages: 00 MessageQueue*: FC5B04B0D4009270
Active Message Queue: 17 Messages: 00 MessageQueue*: FC5B04B0D4009340
Active Message Queue: 18 Messages: 00 MessageQueue*: FC5B04B0D4009410
Active Message Queue: 19 Messages: 00 MessageQueue*: FC5B04B0D40094E0
```

Tuning Values

```
cstcRcvSendTimerRatio      : 2 Default: 2
cstcMcastRelayTimerRatio   : 8 Default: 8
cstcMcastRelayHBTTimerRatio : 4 Default: 4
cstcHeartbeatBaseTimer     : 12288000000 Default: 12288000000
cstcHeartbeatBasePrecision : 4096000000 Default: 4096000000
cstcRetryPrecision         : 4096000000 Default: 4096000000
cstcRetryTimerVal          : 4096000000 Default: 4096000000
cstcCDATBaseTimer          : 49152000000 Default: 49152000000
cstcCDATBasePrecision      : 4096000000 Default: 4096000000
cstcRecoveryBaseTimer      : 3686400000000 Default: 3686400000000
cstcRecoveryBasePrecision  : 491520000000 Default: 491520000000
cstcMaxRetryTime           : 32768000000 Default: 32768000000
cstcCCFragmentSize         : 1464 Default: 1464
cstcCCSendQOverflow        : 1024 Default: 1024
cstcBadMsgCtrThreshold     : 3 Default: 3
cstcUnreachableHBAckThreshold : 1 Default: 1
cstcReachableHBAckThreshold : 3 Default: 3
cstcUnreachableHBThreshold : 4 Default: 4
cstcReachableHBThreshold   : 4 Default: 4
cstcMaxHBThreshold         : 16 Default: 16
cstcDisableMsgTimer        : 0 Default: 0
cstcRepeatAckThreshold     : 10 Default: 10
DISPLAY/ALTER/DUMP        CLUSTERINFO -NEW2 08/23/05 13:36:37 PAGE 87
cstcDelayedAckTimer        : 409600000 Default: 409600000
cstcDelayedAckPrecision    : 40960000 Default: 40960000
cstcCCSendWindow           : 2 Default: 2
cstcCCEnableMcast          : 1 Default: 1
cstcCCPerfClass            : 2 Default: 2
```

***** END OF DUMP *****

第 15 個區段包含調整值。調整值等同於「擷取叢集資源服務資訊 (QcstRetrieveCRSInfo) API」中所說明的叢集效能及配置資訊。第 14 個區段包含這些欄位的現行值及預設值。

一般叢集問題

列出叢集中可能發生的一些常見問題，以及避免發生問題與從問題中回復的方法。

下列一般問題可輕易避免或解決。

您無法啟動或重新啟動叢集節點

此狀況通常是由於通訊環境發生一些問題。若要避免此狀況，請確定您網路屬性的設定正確，包括叢集通訊的 IP 位址、迴路位址、INETD 設定值及 ALWADDCLU 屬性。

- 如果嘗試啟動遠端節點，則必須在目標節點上適當設定 ALWADDCLU 網路屬性。根據您的環境，這應該設為 *ANY 或 *RQSAUT。
- 選擇用來在本端及目標節點上形成叢集的 IP 位址必須顯示作用中狀態。
- 在本端及目標節點的迴路位址 (127.0.0.1) 也必須是作用中。
- 本端及任何遠端節點必須能夠使用形成叢集的 IP 位址來連通測試 (ping)，以確定網路遞送在作用中。
- INETD 在目標節點上必須作用中。當 INETD 處於作用中時，目標節點上的埠 5550 應該處於接聽狀態。如需啟動 INETD 伺服器的詳細資訊，請參閱 INETD 伺服器。
- 嘗試啟動某節點之前，不可開啓該節點的埠 5551，事實上，它會造成主旨節點上形成叢集無法順利啟動。

您結束了數個不連續的單節點叢集

當啟動的節點無法與其它叢集節點通訊時，會發生此情況。請檢查通訊路徑。

跳出程式的回應太慢

此狀況的一般原因是跳出程式使用的工作說明設定不正確。例如，MAXACT 參數可能設得太低，造成一次只能有一個跳出程式在作用中。建議將此值設為 *NOMAX。

一般而言，效能似乎太低

此徵狀有許多常見原因。

- 最可能的原因是共用通訊線路上的通訊傳輸量太大。
- 另一個可能的原因是通訊環境與叢集訊息調整參數不一致。您可以使用擷取叢集資源服務資訊 (QcstRetrieveCRSInfo) API 來檢視調整參數的現行設定值，並使用變更叢集資源服務 (QcstChgClusterResourceServices) API 來變更設定值。如果使用舊的配接卡硬體，則叢集效能可能降到預設叢集調整參數設定值之下。併入舊定義中的配接卡硬體類型為 2617、2618、2619、2626 及 2665。在此情況下，請將效能類別調整參數設為一般。
- 此狀況的另一個一般原因是 IP 多點傳播群組有問題。如果數個節點的主要叢集位址 (建立叢集或新增節點時對指定節點輸入的第一個位址) 位於通用 LAN 上，叢集將利用 IP 多點傳播功能。使用 NETSTAT 指令，確保主要叢集位址顯示多點傳送主機群組 226.5.5.5。對主旨位址使用選項 14 顯示多點傳送群組，即可看到此位址。如果多點傳送群組不存在，請使用擷取叢集資源服務資訊 (QcstRetrieveCRSInfo) API，驗證為啟用多點傳送叢集調整參數設定的預設值是否為 TRUE。
- 如果叢集的所有節點都在本端 LAN 上，或具有遞送能力來處理整個網路遞送中大於 1,464 個位元組的「最大傳輸單位 (MTU)」封包大小，則可增加訊息片段大小的叢集調整參數值，使它更符合遞送 MTU，從而大幅加快大型叢集訊息轉送速度 (大於 1,536K 個位元組)。

您無法使用新版次的任何功能

如果您嘗試使用新版次功能但出現錯誤訊息 CPFBB70，則表示您目前的叢集版本仍設定在先前的版本層次。您必須將所有叢集節點升級至新版次，然後使用調整叢集版本介面將現行叢集版本設為新版次。如需詳細資訊，請參閱「調整叢集的叢集版本」。

您無法將節點新增至裝置網域或存取「iSeries 領航員」叢集管理介面

若要存取「iSeries 領航員」叢集管理介面或要使用切換式裝置，必須在系統上安裝 i5/OS「選項 41 (HA 可切換資源)」。您也必須具備此選項的有效授權碼。

您套用了叢集 PTF 但似乎沒有作用

1 您應該確定在套用 PTF 之後已完成下列作業：

1. 結束叢集
2. 登出，然後登入。

舊程式仍在啟動群組作用中，直到啟動群組損毀為止。所有叢集程式碼 (甚至叢集 API) 是在預設啟動群組中執行。

3. 啟動叢集

大部分的叢集 PTF 需要在節點上先結束再重新啟動叢集作業，才能啟動 PTF。

CEE0200 出現在跳出程式工作日誌中

在此錯誤訊息中，來源模組為 QLEPM，而來源程序為 Q_LE_leBdyPeilog。跳出程式呼叫的任何程式必須在 *CALLER 或指定的啟動群組中執行。您必須變更跳出程式或發生錯誤的程式，以更正此狀況。

叢集資源服務工作日誌中出現了 CPD000D，其後接著 CPF0001

當您收到此錯誤訊息時，請確定 QMLTTHDACN 系統值設為 1 或 2。

叢集暫停

請確定叢集資源群組跳出程式未完成。若要檢查跳出程式，請使用 WRKACTJOB (處理作用中的工作) 指令，然後查看「函數」直欄中是否有出現 PGM-QCSTCRGEXT。

相關概念

第 85 頁的『啟用要新增到叢集的節點』

在您可將節點新增到叢集之前，您必須為「允許新增至叢集 (ALWADDCLU)」網路屬性設定一個值。

第 106 頁的『叢集效能』

變更叢集會影響管理叢集所需的額外執行時間。

第 12 頁的『叢集版本』

叢集版本代表叢集上可用功能的層次。

第 71 頁的『「iSeries 領航員」叢集管理』

IBM 提供的叢集管理介面可透過「iSeries 領航員」使用，並透過「選項 41 (i5/OS - HA 可切換資源)」存取。

相關工作

第 97 頁的『調整叢集的叢集版本』

叢集版本定義叢集中所有節點正在以此相互通訊的層次。

分割區錯誤

某些叢集狀況很容易解決。如果發生叢集分割，您可以學習如何回復。此主題也告訴您如何避免叢集分割並舉例說明如何把分割區合併在一起。

每當叢集中的一或多個節點之間失去連繫，並且無法確認遺失的節點是否故障時，便會在叢集中發生叢集分割。不要將此分割區與邏輯分割區 (LPAR) 環境中的分割區混淆。

如果您收到歷程日誌 (QHST) 或 QCSTCTL 工作日誌中的錯誤訊息 CPFBB20，表示發生叢集分割，您必須知道如何回復。下列範例顯示包含一個由四個節點 (A、B、C 及 D) 組成的叢集之叢集分割區。此範例顯示叢集節點 B 和 C 之間失去通訊，這導致叢集分成兩個叢集分割區。在發生叢集分割之前，有四個叢集資源群組 (稱為 CRG A、CRG B、CRG C 及 CRG D)，它們可以是任何類型。此範例顯示每個叢集資源群組的回復網域。

表 22. 叢集分割期間的回復網域範例

節點 A	節點 B	x	節點 C	節點 D
CRG A (備份 1)	CRG A (主要)			
	CRG B (主要)		CRG B (備份 1)	
	CRG C (主要)		CRG C (備份 1)	CRG C (備份 2)
CRG D (備份 2)	CRG D (主要)		CRG D (備份 1)	
分割區 1			分割區 2	

如果通訊路徑中任何位置的最大傳輸單位 (MTU) 小於叢集通訊可調整參數 (訊息片段大小)，叢集會分割。可在主旨節點上，以使用 TCP/IP 網路狀態 (WRKTCPSTS) 指令來驗證叢集 IP 位址的 MTU。在每一個步驟上，也必須根據整個通訊路徑來驗證 MTU。如果 MTU 小於訊息片段大小，請提高路徑的 MTU 或降低訊息片段大小。您可以使用擷取叢集資源服務資訊 (QcstRetrieveCRSInfo) API 來檢視調整參數的現行設定值，以及使用變更叢集資源服務 (QcstChgClusterResourceServices) API 來變更設定值。

一旦叢集分割狀況的原因更正之後，叢集會偵測重新建立的通訊鏈接並發出歷程日誌 (QHST) 或 QCSTCTL 工作日誌中的 CPFBB21 訊息。這會通知操作員該叢集已從叢集分割區回復。注意：一旦解決叢集分割區狀況後，可能需要幾分鐘叢集才會合併在一起。

相關概念

第 26 頁的『叢集分割區』

叢集分割區是導因於通訊失敗之作用中叢集節點的子集。分割區中的成員會保持彼此之間的連通性。

第 80 頁的『避免叢集分割』

在叢集的所有節點之間配置備用通訊路徑，最能有效的避免一般網路相關的叢集分割。

第 21 頁的『合併』

合併作業與重新結合作業類似，不同處在於前者是在分割的節點重新開始通訊時發生。

第 16 頁的『範例：失效』

通常，節點失效會產生錯誤後移轉，但其它原因也可能產生錯誤後移轉。

判定主要及次要叢集分割區

- | 爲了判定您可以在叢集分割區內採取的叢集資源群組動作類型，您需要知道分割區是主要還是次要叢集分割區。
- | 偵測分割區時，將每個分割區指定爲在叢集中定義之每個叢集資源群組的主要或次要分割區。
- | 若爲主要備份模型，主要分割區包含目前節點角色爲主要的節點。所有其他分割區爲次要分割區。所有叢集資源群組的主要分割區可能不同。
- | 對等節點模型具有下列分割區規則：
 - | • 如果回復網域節點完全包含在一個分割區中，它將是主要分割區。
 - | • 如果回復網域節點跨越分割區，則沒有主要分割區。這兩個分割區均爲次要分割區。
 - | • 如果叢集資源群組處於作用中且指定分割區中沒有對等節點，則叢集資源群組將在該分割區中結束。
 - | • 只要符合作業變更的限制，就允許在次要分割區中進行作業變更。
 - | • 不允許在次要分割區中進行配置變更。

每個「叢集資源群組 API」的限制如下：

表 23. 叢集資源群組 API 分割區限制

叢集資源群組 API	容許在主要分割區中	容許在次要分割區中
將節點新增到回復網域中	X	
新增 CRG 裝置項目		
變更叢集資源群組	X	
變更 CRG 裝置項目	X	X
建立叢集資源群組		
刪除叢集資源群組	X	X
分送資訊	X	X
結束叢集資源群組 ¹	X	
起始切換	X	

表 23. 叢集資源群組 API 分割區限制 (繼續)

叢集資源群組 API	容許在主要分割區中	容許在次要分割區中
列出叢集資源群組	X	X
列出叢集資源群組資訊	X	X
從回復網域中移除節點	X	
移除 CRG 裝置項目	X	
啓動叢集資源群組 ¹	X	
註:		
1. 在對等節點叢集資源群組的所有分割區中都允許執行，但只會影響執行 API 的分割區。		

藉由套用這些限制，當叢集不再分割時，可同步化叢集資源群組。當節點從分割的狀態重新結合叢集時，主要分割區中叢集資源群組的版本會從次要分割區複製到節點。

- 1 當合併對等節點模型的兩個次要分割區時，將其叢集資源群組狀態為「作用中」的分割區宣告為勝出。如果兩個分割區的叢集資源群組狀態相同，則將包含叢集資源群組回復網域中所列出之第一個節點的分割區宣告為勝出。勝出之分割區中的叢集資源群組版本會複製到另一分割區中的節點。

偵測分割區時，不能在任何分割區中執行新增叢集節點項目、調整叢集版本及建立叢集 API。僅在裝置網域中沒有分割節點時，才能執行新增裝置網域項目 API。其它所有「叢集控制 API」可在任何分割區執行。不過，API 執行的動作僅在執行 API 的分割區中才會生效。

變更分割的節點失敗

有時，當節點故障真的發生時會報告分割的狀況。當叢集資源服務與一或多個節點失去通訊，但無法偵測節點是否仍然在運作時，就會發生這種狀況。當發生此狀況時，有一個簡式機制會指示節點已失效。

警告： 當您告知叢集資源服務節點失效時，它會讓從分割區狀態回復變得較簡單。不過，事實上，當節點仍在作用中且發生真實分割時變更節點狀態為失敗不應完成。這樣做會導致多個分割區中的某個節點要擔任叢集資源群組的主要角色。當兩個節點認為它們是主要節點時，如果多個節點分別變更它們的檔案副本，則檔案或資料庫資料可能會不連續或毀損。此外，當每個分割區中的節點都指定了主要角色時，兩個分割區無法合併在一起。

當節點狀態變更為「失敗」時，分割區中每個叢集資源群組回復網域中節點的角色可能重新排序。設為「失敗」的節點會被指定為上一個備份。如果多個節點失敗且其狀態需要變更，則變更的節點次序會影響回復網域備份節點的最終次序。如果失敗節點為 CRG 的主節點，第一個作用中備份將被重新分派為新的主節點。

相關概念

第 21 頁的『合併』

合併作業與重新結合作業類似，不同處在於前者是在分割的節點重新開始通訊時發生。

第 19 頁的『重新結合』

重新結合表示已成為未參加的成員之後，再成為叢集的作用中成員。

相關工作

第 132 頁的『叢集分割區秘訣』

使用這些叢集分割區的要訣。

相關參考

CHGCLUNODE 指令

變更叢集節點項目 (QcstChangeClusterNodeEntry) API

STRCLUNOD 指令

啓動叢集節點 (QcstStartClusterNode) API

使用「iSeries 領航員」：

這需要安裝及授權「選項 41 (i5/OS - HA 可切換資源)」。

當叢集資源服務與節點失去通訊，但無法偵測節點是否仍在運作時，叢集節點將在「iSeries 領航員」的「節點」儲存區中具有**不在通訊中**狀態。您可能需要將節點狀態從**不在通訊中**變更為**失敗**。然後，您才能夠重新啓動節點。

若要將節點狀態從**不在通訊中**變更為**失敗**，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 展開包含您要變更其狀態之節點的叢集。
4. 按一下**節點**。
5. 以滑鼠右鍵按一下您要變更其狀態的節點，然後選取**叢集** → **變更狀態**。

選取 ClusterChange 狀態

若要重新啓動節點，請遵循下列步驟：

1. 以滑鼠右鍵按一下節點，然後選取**叢集** → **啓動**。

使用 CL 指令及 API:

若要將節點狀態從**不在通訊中**變更為**失敗**，請遵循下列步驟：

1. 使用 CHGCLUNODE 指令或變更叢集節點項目 (QcstChangeClusterNodeEntry) API，將節點的狀態從分割變更為**失敗**。這應該對實際失敗的所有節點執行。
2. 使用 STRCLUNOD 指令或啓動叢集節點 (QcstStartClusterNode) API 來啓動叢集節點，並允許節點重新結合叢集。

叢集分割區秘訣

使用這些叢集分割區的要訣。

1. 限制分割區作業的規則是為了使分割區合併可行而設計的。如果沒有這些限制，則需要花費很大的功夫來重新建構叢集。
2. 如果主要分割區中的節點已毀損，則在次要分割區中需要特殊處理程序。導致此狀況的最常見實務範例是失去組成主要分割區的站台。使用從分割區錯誤中回復的範例，並假設分割區 1 已損毀。在這個情況下，叢集資源群組 B、C 及 D 的主節點必須位於 分割區 2 中。最簡單的回復方式是使用「變更叢集節點項目」將「節點 A」和「節點 B」設為**失敗**。如需如何執行此動作的詳細資訊，請參閱將分割的節點變更為**失敗**。也可以經由手動方式完成回復。若要這麼做，請執行以下作業：
 - a. 從分割區 2 中的叢集移除節點 A 和 B。分割區 2 目前是叢集。
 - b. 建立新叢集所需的任何邏輯抄寫環境。亦即，啓動「叢集資源群組 API/CL」指令等等。

因為節點已從分割區 2 的叢集定義中移除，所以嘗試合併分割區 1 和分割區 2 會**失敗**。為了更正叢集定義中的不符情況，請在分割區 1 的每一個節點上執行刪除叢集 (QcstDeleteCluster) API。然後將節點從分割區 1 新增至叢集，並重新建立所有的叢集資源群組定義、回復網域及邏輯抄寫。這需要很大的功夫，而且容易出錯。請務必僅在站台遺失情況下才執行此程序。

3. 處理啓動節點作業與要啓動的節點的狀態有關：

節點失敗或「結束節點」作業結束了節點：

- a. 叢集資源服務在新增的節點上啟動
- b. 叢集定義從叢集的作用中節點複製到啟動的節點。
- c. 具有在回復網域中啟動的節點之任何叢集資源群組是從叢集的作用中節點複製到啟動的節點。沒有叢集資源群組會從啟動的節點複製到叢集的作用中節點。

節點為分割的節點：

- a. 比較作用中節點的叢集定義與啟動的節點之叢集定義。如果定義相同，將繼續以合併作業來啟動。如果定義不符，合併會停止，且需要使用者介入。
- b. 如果合併繼續，啟動的節點會設為作用中狀態。
- c. 具有在回復網域中啟動的節點之任何叢集資源群組是從叢集資源群組的主要分割區複製到叢集資源群組的次要分割區。叢集資源群組可從啟動的節點複製到叢集已作用中的節點。

相關工作

第 131 頁的『變更分割的節點失敗』

有時，當節點故障真的發生時會報告分割的狀況。當叢集資源服務與一或多個節點失去通訊，但無法偵測節點是否仍然在運作時，就會發生這種狀況。當發生此狀況時，有一個簡式機制會指示節點已失效。

相關參考

刪除叢集 (QcstDeleteCluster) API

叢集回復

瞭解如何從可能發生的其它叢集失敗中回復。

從叢集工作失敗中回復

叢集資源服務工作失敗通常表示有其它問題。

您應該尋找與失敗工作相關的工作日誌，以及 尋找說明失敗原因的訊息。請更正任何錯誤狀況。

- 1 您可以使用變更叢集回復 (CHGCLURCY) 指令來重新啟動已結束的叢集資源群組工作，而不需要先結束再重新啟動節點上的形成叢集作業。
- 1 1. CHGCLURCY CLUSTER(EXAMPLE)CRG(CRG1)NODE(NODE1)ACTION(*STRCRGJOB) 此指令將會提交節點 NODE1 上的叢集資源群組工作 CRG1。若要啟動 NODE1 上的叢集資源群組工作，NODE1 上的形成叢集作業需要在作用中。
- 1 2. 重新啟動節點上的形成叢集。

如果您正在使用「IBM 商業夥伴」叢集管理產品，請參閱產品附帶提供的文件。

相關概念

第 108 頁的『工作結構和使用者佇列』

在管理叢集時，您需要了解工作結構及使用者佇列。

相關工作

第 96 頁的『結束叢集節點』

停止或結束節點會停止該節點上的叢集資源服務。

第 96 頁的『啟動叢集節點』

啟動叢集節點會在叢集中的節點上啟動叢集資源服務。自叢集的第 3 版開始，節點可以啟動自己，且只要能夠在叢集中找到作用中節點，就能夠重新結合現行的作用中叢集。

回復受損的叢集物件

雖然您不太可能遇到受損的物件，但叢集資源服務物件可能受損。

如果系統是作用中節點，它會嘗試從叢集中另一個作用中節點回復。系統會執行下列回復步驟：

針對已損壞的内部物件

1. 結束有損壞的節點。
2. 如果叢集內至少有一個作用中節點，受損的節點會自己自動重新啓動並重新結合叢集。重新結合的處理會解決受損的狀況。

針對已損壞的叢集資源群組

1. 具有受損的 CRG 之節點會使得目前與 CRG 相關的任何進行中作業失敗。然後，系統會嘗試從另一個作用中節點自動回復 CRG。
2. 如果回復網域中至少有一個作用中成員，則 CRG 回復有效。否則，會結束 CRG 工作。

若系統無法識別或連上其他作用中節點，您必須執行這些回復步驟。

針對已損壞的内部物件

您收到內部形成叢集錯誤 (CPFBB46、CPFBB47 或 CPFBB48)。

1. 結束有損壞的節點之形成叢集。
2. 重新啓動有損壞的節點之形成叢集。請從叢集中另一個作用中節點執行。
3. 如果步驟 1 和 2 沒有解決問題，請從叢集中移除受損的節點。
4. 使系統回到叢集中及適當叢集資源群組的回復網域中。

針對已損壞的叢集資源群組

您收到指出物件已損壞的錯誤 (CPF9804)。

1. 結束含有受損叢集資源群組之節點的形成叢集。
2. 刪除 CRG (使用 DLTCRG 指令)。
3. 如果叢集中沒有其它含有 CRG 物件的節點在作用中，請從媒體中回復。
4. 啓動含有受損叢集資源群組之節點的形成叢集。這可以從任何作用中節點執行。
5. 當啓動形成叢集時，系統會重新同步所有叢集資源群組。如果叢集中沒有其它節點含有 CRG，您可能需要重建 CRG。

在失去完整系統後回復叢集

當您的伺服器意外斷電時，若要在失去完整系統後回復整個系統，請將此資訊與「備份及回復」手冊中適當的核對清單一起使用。

實務範例 1：復置至相同系統

1. 爲防止「授權內碼」與 i5/OS 之間裝置網域資訊的不一致，建議您使用選項 3 (安裝授權內碼與回復配置) 來安裝「授權內碼」。

註：若要「安裝授權內碼與回復配置」作業成功，您必須具有相同硬碟機，除了失效的載入來源硬碟機之外。您也必須回復相同版次。

2. 安裝「授權內碼」之後，請遵循備份及回復手冊第 5 章中的如何回復您的磁碟配置程序。這些步驟可協助您避免重新配置 ASP。

3. 在回復您的系統資訊之後準備要在您剛才所回復節點上開始形成叢集時，您必須從作用中節點開始形成叢集。這會將大部份現行配置資訊傳送到回復的節點。

實務範例 2：復置至不同的系統

在回復您的系統資訊及檢查工作日誌確定所有物件皆已復置後，您必須執行下列步驟來取得正確的叢集裝置網域配置。

1. 從您剛才復置的節點中刪除叢集。
2. 從作用中節點執行下列步驟：
 - a. 從叢集中移除回復的節點。
 - b. 將回復的節點新增回叢集中。
 - c. 將回復的節點新增至裝置網域。
 - d. 建立叢集資源群組或將節點新增至回復網域。

相關工作

第 110 頁的『叢集的備份及回復服務程式』

如果您在系統上使用叢集，則建立備份及回復策略來保護資料仍然很重要。

相關資訊

備份及回復

在災害之後回復叢集

在遺失您所有節點的災害情況下，您需要重新配置叢集。

為因應這種情況，建議您儲存叢集配置資訊，並保存將該資訊列印出來的輸出品。

相關工作

第 110 頁的『叢集的備份及回復服務程式』

如果您在系統上使用叢集，則建立備份及回復策略來保護資料仍然很重要。

從備份磁帶復置叢集

在正常作業期間，您不需要從備份磁帶復置。

只在發生災害且叢集中的所有節點均遺失時，才需要這樣做。如果發生災害，請遵循您建立備份及回復策略之後所準備的正常回復程序來進行回復。

相關工作

第 110 頁的『叢集的備份及回復服務程式』

如果您在系統上使用叢集，則建立備份及回復策略來保護資料仍然很重要。

相關資訊

備份及回復

「iSeries 領航員」叢集管理常見問題集

有關用於建立及管理叢集之「iSeries 領航員」圖形式使用者介面的問題和回答。

建立及管理叢集的 IBM 圖形式使用者介面，可在「iSeries 領航員」中使用，並可透過「選項 41 (HA 可切換資源)」存取。如需此介面的詳細資訊，請參閱「iSeries 領航員」叢集管理。

以下是「iSeries 領航員」叢集管理問答清單。

一般

1. 是否有核對清單列舉出建立叢集的必備項目？

「iSeries 領航員」叢集管理

1. 「iSeries 領航員」介面的「叢集」功能在何處？
2. 如何建立叢集？
3. 「叢集」資料夾和「管理中心」系統群組之間的關係為何？
4. 我已經在網路中的一些 iSeries 系統上定義一個叢集。我要如何新增它，才能透過「iSeries 領航員」來檢視和管理它？
5. 我的叢集中沒有任何節點的狀態是「已啟動」。我應該先啟動哪一個節點？
6. 我為什麼要在乎先啟動哪一個節點？
7. 切換式裝置及切換式應用程式資料夾中的「現行主要節點」直欄代表什麼？
8. 如何在「iSeries 領航員」中尋找裝置叢集資源群組 (CRG)？
9. 如何在「iSeries 領航員」中尋找應用程式叢集資源群組 (CRG)？
10. 如何在「iSeries 領航員」中尋找資料叢集資源群組 (CRG)？
11. 我想要查看切換式裝置 (裝置 CRG) 狀態，但不要回到「切換式硬體」資料夾。我應該如何做呢？

通訊

1. 「iSeries 領航員」中的「叢集」功能使用什麼 IP 位址來與叢集中的節點通訊？不是使用節點名稱的 IP 位址嗎？

安全性

1. 在「iSeries 領航員」的「叢集」資料夾中，為什麼大部分的快速功能表都停用或不見了？
2. 「iSeries 領航員」中的「叢集」功能使用「應用程式管理」值嗎？
3. 「iSeries 領航員」中的「叢集」功能為什麼對我在叢集中的節點顯示登入視窗？

疑難排解

1. 「管理中心」為什麼沒有顯示「叢集」資料夾？
2. 我已經有一個叢集，但它沒有顯示在「叢集」資料夾中。為什麼？
3. 「叢集」資料夾中為什麼沒有顯示最新的狀態？
4. 我的切換式硬體群組或切換式應用程式為什麼沒有執行錯誤後移轉？
5. 我收到已損壞物件的訊息。我要如何因應？
6. 我使用精靈中的「瀏覽」按鈕來讓節點瀏覽 IP 位址。瀏覽視窗為什麼沒有顯示我預期的所有 TCP/IP 位址？
7. 在「iSeries 領航員」的「叢集」資料夾中，為什麼大部分的快速功能表都停用或不見了？
8. 我在使用「新建叢集」精靈時出現一個畫面，標題是：「新建叢集 - 找不到切換式軟體」。這是錯誤嗎？
9. 我有一個節點的狀態是「沒有通訊」。我要如何更正呢？

一般

是否有核對清單列舉出建立叢集的必備項目？

是。使用叢集配置核對清單，確定您已準備好在您的環境中配置叢集。

[返回問題](#)

「iSeries 領航員」叢集管理：「iSeries 領航員」介面的「叢集」功能在何處？

「iSeries 領航員」叢集管理介面是 IBM iSeries Access 套裝軟體的一部分。「叢集」功能位於「iSeries 領航員」的「管理中心」資料夾中。如需詳細資訊，請參閱「iSeries 領航員」叢集管理。

[返回問題](#)

如何建立叢集？

若要使用「iSeries 領航員」中的「新建叢集」精靈來建立簡式叢集，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 以滑鼠右鍵按一下**叢集**，然後選取**新建叢集**。
3. 遵循精靈的指示來建立叢集。

如需建立及配置叢集的完整詳細資訊，請參閱**配置叢集**。

[返回問題](#)

「叢集」資料夾和「管理中心」系統群組之間的關係為何？

使用「iSeries 領航員」建立叢集時，也會在「管理中心」伺服器上建立系統群組。系統群組的名稱與叢集名稱相同，且系統群組中的端點系統是叢集中的節點。系統群組也有自己的特殊類型，因此「iSeries 領航員」知道它是一個代表叢集的特殊系統群組。

重要：「管理中心」系統包含系統群組。如果您選擇變更「iSeries 領航員」的現行「管理中心」系統，新的管理中心系統將不會有特殊叢集系統群組，因此那些叢集不會出現在「叢集」資料夾中。

[返回問題](#)

我已經在網路中的一些 iSeries 系統上定義一個叢集。我要如何新增它，才能透過「iSeries 領航員」來檢視和管理它？

若要透過「iSeries 領航員」來新增現有的叢集，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 以滑鼠右鍵按一下**叢集**，然後選取**新增現有的叢集**。
3. 在**新增現有的叢集**視窗上，指定叢集中的其中一個伺服器。
4. 按一下「確定」。

[返回問題](#)

我的叢集中沒有任何節點的狀態是「已啓動」。我應該先啓動哪一個節點？

您應該先啓動最近狀態曾為「已啓動」的節點。例如，假設您的叢集中有兩個節點：A 和 B。節點 A 和節點 B 目前都未啓動。不過，節點 B 是上次執行時狀態為「已啓動」的節點。你應該先啓動節點 B，因為它含有最新的叢集資訊。

[返回問題](#)

我為什麼要在乎先啓動哪一個節點？

您應該在乎，因為最近狀態曾為「已啓動」的節點是含有最新叢集資訊的節點。這點很重要，因為如果您啓動已停用最久的其他節點，則它可能包含過期的叢集資訊。當叢集中的其他節點啓動時，過期資訊就會傳送到這

些節點，而帶來危險。例如，假設有一個節點叢集包含節點 A 和節點 B。如果節點 B 是最近曾作用且當時狀態是「已啟動」的節點，則它包含最新的叢集資訊。如果您選擇先啟動節點 A，它可能包含部份過期的資訊，但仍然會啟動。當您再啟動節點 B 時，它會結合叢集中目前作用中的節點 (與節點 A 結合)。節點 A 的過期叢集資訊將傳送到節點 B，造成兩個節點都包含過期的叢集資訊。這就是為何先啟動節點 B 比較重要的原因。過期叢集資訊會影響切換式裝置的配置。如果您發現，因為切換式硬體群組在顯示不同現行節點時出現備份節點上的硬碟機報告，而使得在啟動切換式裝置時有些問題，那麼您必須變更回復網域中的節點角色，使擁有硬碟機的節點成為主要節點。

[返回問題](#)

切換式硬體、切換式軟體及切換式資料等資料夾中的「現行主要節點」直欄代表什麼？

「現行主要節點」直欄指出節點目前是切換式裝置或切換式軟體產品的主要節點。或者，以叢集 API 術語而言，它是表示在主要回復網域中扮演現行角色的節點。

[返回問題](#)

如何在「iSeries 領航員」中尋找裝置叢集資源群組 (CRG)？

裝置 CRG (叢集資源群組) 稱為「可切換硬體群組」，位於「叢集」資料夾的**可切換硬體**資料夾中。

[返回問題](#)

如何在「iSeries 領航員」中尋找應用程式叢集資源群組 (CRG)？

應用程式 CRG (叢集資源群組) 稱為「可切換軟體產品」，位於「叢集」資料夾的**可切換軟體**資料夾中。

[返回問題](#)

如何在「iSeries 領航員」中尋找資料叢集資源群組 (CRG)？

資料 CRG (叢集資源群組) 稱為「可切換資料群組」，位於「叢集」資料夾的**可切換資料**資料夾中。

[返回問題](#)

我想要查看「切換式硬體群組」(裝置 CRG) 狀態，但不要回到「切換式硬體」資料夾。我應該如何做呢？

不必每次檢視狀態時都要導覽到「可切換硬體」資料夾，您也可以用滑鼠右鍵按一下**可切換硬體**資料夾並選取**開啓**，來開啓一個具有「可切換硬體」檢視畫面的新視窗。這個不同的視窗將顯示「可切換硬體群組」(裝置 CRG) 及其相關的狀態資訊。這也適用於**可切換軟體**和**可切換資料**。

[返回問題](#)

通訊：「iSeries 領航員」中的「叢集」功能使用什麼 IP 位址來與叢集中的節點通訊？不是使用節點名稱的 IP 位址嗎？

主要的「叢集」資料夾中有一個「伺服器」直欄顯示您配置的叢集的相關資訊。每一個叢集的內容畫面也會顯示伺服器名稱。「伺服器」直欄中列出的伺服器是「iSeries 領航員」介面用來與叢集通訊的叢集中之節點。它只適用於「iSeries 領航員」如何與伺服器上的叢集物件通訊，不適用於叢集中的節點如何互相通訊。「iSeries 領航員」叢集管理所用的伺服器與現行「管理中心」伺服器無關。

如果「iSeries 領航員」用來與叢集通訊的節點停用，則您可以將通訊機制變更為叢集中的不同節點來執行叢集動作。

若要變更「iSeries 領航員」介面用來與叢集通訊的伺服器，請遵循下列步驟：

1. 在「iSeries 領航員」中，展開**管理中心**。
2. 展開**叢集**。
3. 以滑鼠右鍵按一下叢集，然後選取**變更伺服器**。

返回問題

安全性：在「iSeries 領航員」的「叢集」資料夾中，為什麼大部分的快速功能表都停用或不見了？

部份作業會根據叢集的現行配置狀態如何來決定是否可用。例如，您無法停止一個已停止的節點您無法在已配置最大節點數量 (4) 的叢集中新增節點。特定作業的線上說明解釋為何有些項目停用或無法使用。

如果您不具備足夠的權限，則無法使用部份作業。若您使用「iSeries 領航員」且具有 *SECOFR 使用者類別權限，您將具有所有叢集作業及管理的存取權限。「iSeries 領航員」使用現行「管理中心」系統的「應用程式管理」權限，決定您是否具備各種「iSeries 領航員」叢集管理作業的「應用程式管理」權限。

如需使用「應用程式管理」的詳細資訊，請參閱應用程式管理。

返回問題

「iSeries 領航員」中的叢集功能使用「應用程式管理」值嗎？

是。「iSeries 領航員」叢集管理使用現行「管理中心」系統的「應用程式管理」權限值，決定您是否具有各種叢集作業的「應用程式管理」權限。

「iSeries 領航員」有兩種用於存取的權限設定值：**叢集作業**和**叢集管理**

具備**叢集作業**權限，您可以：

- 檢視叢集的狀態
- 啟動及停止節點
- 啟動及停止切換式硬體與切換式軟體
- 執行可切換硬體和可切換軟體的手動切換作業

具備**叢集管理**權限，您可以：

- 建立/刪除叢集
- 新增及移除節點
- 新增及刪除切換式硬體、切換式硬體與磁碟儲存區
- 變更可切換硬體和可切換軟體的內容

返回問題

「iSeries 領航員」中的「叢集」功能為什麼對我在叢集中的節點顯示登入視窗？

在某些時候，「iSeries 領航員」會嘗試與叢集中的所有節點通訊。這要視叢集的狀態而定。當「iSeries 領航員」需要與節點通訊時，會先搜尋「iSeries 領航員」中現有的登入快取，以嘗試找出現有開啓的連線。如果找不到現有的連線，則會要求使用者登入。若您取消登入視窗，「iSeries 領航員」會嘗試容許使用者執行叢集作業。若「iSeries 領航員」無法與節點通訊，可能無法使用某些作業。

返回問題

疑難排解：「管理中心」為什麼沒有顯示「叢集」資料夾？

您可能未在 PC 上執行 iSeries Access for Windows® 的完整安裝。您可能是執行基本安裝或選擇一些自訂選項。如需安裝的詳細資訊，請參閱 iSeries Access。

返回問題

我已經有一個叢集，但它沒有顯示在「叢集」資料夾中。為什麼？

簡單的回答：因為您的「管理中心」系統上沒有一個代表叢集的系統群組，所以沒有顯示。當建立叢集或使用「新增現有的叢集」動作在「叢集」資料夾中新增叢集時，「iSeries 領航員」叢集管理會建立代表叢集的系統群組。您可以在「管理中心」展開系統群組資料夾來查看系統群組。叢集系統群組會顯示為「協力廠商」系統群組，但不要假設所有「協力廠商」系統群組都是叢集。

返回問題

「叢集」資料夾中為什麼沒有顯示最新的狀態？

「iSeries 領航員」會到叢集節點取得最新的叢集資訊，然後顯示在「iSeries 領航員」視窗中，以瞬像方式來顯示已配置叢集的相關資訊。它不會自動執行資訊的定期更新。取得資訊最新瞬像的最好方法是手動重新整理。您可以使用「iSeries 領航員」的檢視功能表，然後選擇重新整理選項。另一個方法是設定「iSeries 領航員」執行自動重新整理。

返回問題

我的切換式裝置、切換式應用程式或切換式資料群組為什麼沒有執行錯誤後移轉？

最常見的情況是您在叢集中未啟動可切換資源 (叢集資源群組)。換言之，在發生自動錯誤後移轉之前，可切換資源的狀態不是「已啟動」。您的可切換資源必須啟動，才能發生錯誤後移轉。

返回問題

我收到已損壞物件的訊息。我要如何因應？

您可能會收到類似的訊息：QCLUMGT 中的 CPF811C 使用者佇列 QUGCLUSRQ 損壞

選項 1：一種作法是刪除物件再復置它。這只在您先前已儲存物件的前提下才有可能。

選項 2：刪除已損壞的物件。例如，如果檔案庫 QCLUMGT 中的 QUGCLUSRQ 已損壞，則刪除物件。然後，在「iSeries 領航員」中新增現有的叢集。新增叢集時，叢集 GUI 會檢查物件是否存在，如果不存在則重建。請參閱「我要如何新增現有的叢集，才能透過「iSeries 領航員」來檢視和管理它？」，以取得新增現有叢集的詳細資訊。

返回問題

我使用精靈中的「瀏覽」按鈕來讓節點瀏覽 IP 位址。瀏覽視窗為什麼沒有顯示我預期的所有 TCP/IP 位址？

這個清單只是可能的 IP 位址的候選清單。您不必受限於視窗中顯示的可能位址清單。您可以輸入任何想要的叢集介面位址。不過要注意，以後如果「iSeries 領航員」無法使用您指定作為主要 IP 位址的 IP 位址連接，您將收到錯誤。「iSeries 領航員」使用主要 IP 位址來連接叢集中的節點。

返回問題

我在使用「新建叢集」精靈時出現一個畫面，標題是：「新建叢集 - 找不到切換式軟體」。這是錯誤嗎？

不，這沒什麼問題，也不是錯誤。顧名思義，這意思就是「iSeries 領航員」介面找不到任何可使用精靈自動安裝的切換式軟體。「iSeries 領航員」需要自動安裝的切換式軟體符合具備叢集之應用程式的 i5/OS 架構。此外，「iSeries 領航員」僅支援此架構的子集，並非全部。

返回問題

我有一個節點的狀態是「沒有通訊」。我要如何更正呢？

如果您在叢集中的一或多個節點之間失去連繫，且無法確認遺失節點的故障情形，則會發生叢集分割。如需詳細資訊，請參閱分割區錯誤。

有時，當實際上有一個節點故障時，則會報告分割狀況。當叢集資源服務與一或多個節點失去通訊，但無法偵測節點是否仍在運作中時，就會發生這種狀況。當發生這種狀況時，有一個簡單的機制可讓您指出節點已失敗。如需詳細資訊，請參閱將分割的節點變更為失敗。

返回問題

叢集支援的客服電話

如果您需要聯絡 IBM 來解決叢集問題，請參閱此主題。

如果您在判定叢集是否對您的業務有益時需要協助，或您在實作叢集後遇到問題，可以聯絡下列服務中心：

- 如需其他技術性業務協助或要租用 IBM 諮詢服務，請傳送電子郵件至 rhc1st@us.ibm.com，以聯絡「iSeries 技術中心」內的 Continuous Availability Center。
- 如果有其他問題，請聯絡您向其購買叢集套裝軟體的「IBM 商業夥伴」，或電洽 1-800-IBM-4YOU (1-800-426-4968)。






相關工作

第 92 頁的『配置叢集』
瞭解如何建立叢集。

叢集的相關資訊

尋找叢集的相關資訊。

Redbooks

- Data Resilience Solutions for IBM i5/OS High Availability Clusters 
- Clustering and IASPs for Higher Availability 
- High Availability on the AS/400® System: A System Manager's Guide 
- IBM eServer iSeries Independent ASPs: A Guide to Moving Applications to IASPs 
- The System Administrator's Companion to AS/400 Availability and Recovery 

網站


- High Availability and Clusters  (www.ibm.com/servers/eserver/series/ha)
- IBM 網站，可取得「高可用性與叢集」的相關資訊

儲存 PDF 檔

若要在工作站上儲存 PDF 以供檢視或列印：

1. 以滑鼠右鍵按一下瀏覽器內的 PDF (以滑鼠右鍵按一下上述的鏈結)。
2. 如果是使用 Internet Explorer，請按一下**另存目標**。如果是使用 Netscape Communicator，請按一下**另存鏈結為**。
3. 瀏覽至要儲存 PDF 的目錄。
4. 按一下**儲存**。

下載 Adobe Acrobat Reader

您需要 Adobe Acrobat Reader 才可檢視或列印這些 PDF。您可以從 Adobe 網站 (www.adobe.com/products/acrobat/readstep.html)  下載副本。

程式碼授權及免責聲明資訊

IBM 授與您使用所有程式設計程式碼範例的非專屬授權，您可以利用這些範例來產生符合您需求的類似函數。

除法律規定不得排除的保證外，IBM、IBM 之程式開發人員及供應商不附具任何明示或默示之保證，包含且不限於任何相關技術支援之未侵害他人權利之保證、或可商用性及符合特定效用等之默示保證。

在任何情況下，IBM、IBM 之程式開發者或供應商對下列情事均不負賠償責任，即使被告知該情事有可能發生時，亦同：

1. 資料之滅失或毀損；
2. 直接、特殊、附帶或間接的傷害或其他衍生之經濟損害；或
3. 利潤、營業、收益、商譽或預期節餘等項之損失。

倘法律規定不得排除或限制賠償責任時，則該排除或限制無效。

附錄. 注意事項

本資訊是針對 IBM 在美國所提供之產品與服務開發出來的。

而在其他國家中，IBM 不見得有提供本書中所提的各項產品、服務、或功能。要知道您所在區域是否可用到這些產品與服務時，請向當地的 IBM 服務代表查詢。本書在提及 IBM 產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其他非 IBM 產品、程式或服務在運作上的評價與驗證，其責任屬於使用者。

在這本書或文件中可能包含著 IBM 所擁有之專利或專利申請案。本書使用者並不享有前述專利之任何授權。您可以用書面方式來查詢授權，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

若要查詢有關二位元組 (DBCS) 資訊的特許權限事宜，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

下列段落若與當地之法令抵觸，則不適用之：IBM 僅以「現狀」提供本出版品，而不為任何明示或默示之保證 (包括但不限於產品未涉侵權、可售性或符合特定效用的保證。) 倘若干地區在特定交易中並不許可相關明示或默示保證之棄權聲明，則於該等地區之特定交易，此項聲明不適用之。

本資訊中可能包含技術上或排版印刷上的錯誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 得隨時修改或變更本出版品中所提及的產品及程式。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該等網站並不提供保證。該等網站上的資料，並非 IBM 產品所用資料的一部分，如因使用該等網站而造成損害，其責任由 貴客戶自行負責。

IBM 得以其認定之各種適當方式使用或散布由 貴客戶提供的任何資訊，而無需對您負責。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方法 若有任何問題請聯絡：

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

- | IBM 基於雙方之「IBM 客戶合約」、「IBM 國際程式授權合約」、「IBM 機器碼授權合約」或任何同等合約
- | 之條款，提供本出版品中所述之授權程式與其所有適用的授權資料。

任何此處涵蓋的執行效能資料都是在一個受控制的環境下決定出來的。因此，於其他不同作業環境之下所得的結果，可能會有很大差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。再者，有些測定可能已透過推測方式評估過。但實際結果可能並非如此。本文件的使用者應根據其特有的環境，驗證出適用的資料。

本資訊所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其他公開管道取得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性、或任何對產品的其他主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

有關 IBM 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

所有顯示之 IBM 產品售價僅為 IBM 產品之一般市場價格，可能於未事先聲明之情況下有所變動。經銷商售價可能有所不同。

本資訊僅供規劃用途。所提及的產品發行之前，本書內含的資訊有變動的可能。

本資訊中含有日常商業活動所用的資料及報告範例。為了提供完整的說明，這些範例包括個人、公司、廠牌和產品名稱。這些名稱全屬虛構，若與任何公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊包含原始語言的範例應用程式，用以說明各種作業平台上的程式設計技術。您可以基於研發、使用、銷售或散佈符合作業平台（用於執行所撰寫的範例程式）之應用程式設計介面的應用程式等目的，以任何形式複製、修改及散佈這些範例程式，而無需付費給 IBM。這些範例尚未徹底經過所有情況的測試。因此，IBM 不保證或暗示這些程式的穩定性、服務能力或功能。

這些範例程式或是任何衍生著作的每一份拷貝或任何部份，都必須具有下列的著作權聲明：

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

若您是以電子檔檢視此資訊，則照片和彩色圖例可能不會出現。

程式設計介面資訊

本「叢集」出版品文件是使用允許客戶撰寫程式以取得 IBM i5/OS 服務的「程式設計介面」。

商標

下列術語是 IBM 公司在美國及 (或) 其它國家的商標。

- | 400
- | i5/OS
- | IBM
- | iSeries
- | OS/400
- | Redbooks

- | Intel、Intel Inside (標誌)、MMX 及 Pentium 是 Intel Corporation 在美國及 (或) 其他國家的商標。

Microsoft、Windows、Windows NT 以及 Windows 商標是 Microsoft Corporation 在美國及 (或) 其他國家的商標。

Java 以及所有與 Java 有關的商標是 Sun Microsystems, Inc. 在美國及 (或) 其他國家的商標。

| Linux 是 Linus Torvalds 在美國及 (或) 其他國家的商標。

UNIX 是 The Open Group 在美國及其他國家的註冊商標。

其他公司、產品及服務名稱，可能是其他公司的商標或服務標誌。

條款

根據下述條款，授予您對這些出版品的使用權限。

個人使用：您可複製該等出版品供個人及非商業性用途使用，惟應註記 IBM 著作權標示及其他所有權歸屬 IBM 之相關文字。未經 IBM 明示同意，您不得散佈、展示或改作該等出版品或其任何部份。

商業使用：您可以複製、散佈及展示該等出版品僅供企業內部使用，惟應註記 IBM 著作權標示及其他所有權歸屬 IBM 之相關文字。未經 IBM 明示同意，您不得改作該等出版品，也不得於企業外複製、散佈或展示該等出版品或其任何部份。

除本使用聲明中明確授予之許可外，使用者就出版品或任何包含於其中之資訊、資料、軟體或其他智慧財產權，並未取得其他任何明示或默許之許可、軟體授權或權利。

使用者對於出版品之使用如危害 IBM 的權益，或 IBM 認定其未遵照上述指示使用出版品時，IBM 得隨時撤銷此處所授予之許可。

除非您完全遵守所有適用之一切法規，包括所有美國出口法規，否則您不得下載、出口或再輸出此等資訊。

IBM 對於該等出版品之內容不為任何保證。出版品依其「現狀」提供，不附帶任何明示或默示之擔保，其中包括 (但不限於) 適售性、未涉侵權及適合特定用途之默示擔保責任。

讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您閱讀後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號 (✓)；我們會在下一版中，作適當修訂，謝謝您的合作!

評估項目	評估意見	備註
正確性	內容說明與實際程序是否符合	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	參考書目是否正確	<input type="checkbox"/> 是 <input type="checkbox"/> 否
一致性	文句用語及風格，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	實際產品介面訊息與本書中所提是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
完整性	是否遺漏您想知道的項目	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字句、章節是否有遺漏	<input type="checkbox"/> 是 <input type="checkbox"/> 否
術語使用	術語之使用是否恰當	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	術語之使用，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
可讀性	文句用語是否通順	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	有否不知所云之處	<input type="checkbox"/> 是 <input type="checkbox"/> 否
內容說明	內容說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	例題說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
排版方式	本書的形狀大小，版面安排是否方便閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字體大小，顏色編排，是否有助於閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
目錄索引	目錄內容之編排，是否便於查找	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	索引語錄之排定，是否便於查找	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	※評估意見為 "否" 者，請於備註欄提供建議。	

其他：(篇幅不夠時，請另外附紙說明。)

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。
註：您也可將寶貴的意見以電子郵件寄至 tscadmin@tw.ibm.com，謝謝。

IBM 系統 - iSeries

RZAI-G000-06

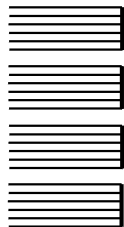
系統管理
叢集

版本 5 版次 4

折疊線

110 台北市信義區松仁路 7 號 3 樓

臺灣國際商業機器股份有限公司
大中華研發中心 軟體國際部 啟



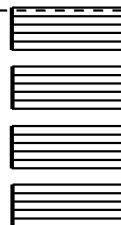
廣告回信
台灣北區郵政管理局
登記
北台字第 00176 號

(免貼郵票)

寄件人 姓名：
地址：

寄

折疊線



IBM