

9.1

*IBM MQ* 的監視及效能

**IBM**

## 附註

使用本資訊及其支援的產品之前，請先閱讀第 339 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® MQ 9.1.1 版及所有後續版本與修訂版。

當您將資訊傳送至 IBM 時，您授與 IBM 非專屬權利，以任何其認為適當的方式使用或散佈資訊，而無需對您負責。

© Copyright International Business Machines Corporation 2007, 2024.

# 目錄

<b>監視及效能</b> .....	<b>5</b>
監視 IBM MQ 網路.....	5
事件監視.....	5
訊息監視.....	50
帳戶及統計資料訊息.....	120
應用程式活動追蹤.....	180
監視及活動追蹤的系統主題.....	261
監視 IBM MQ Bridge to Salesforce.....	269
即時監視.....	272
監視叢集.....	282
監視應用程式平衡.....	284
監視效能及資源使用情形.....	288
調整 IBM MQ 網路.....	327
調整用戶端及伺服器連線通道.....	327
調整分散式發佈/訂閱網路.....	328
減少主題樹狀結構中不想要的主题數目.....	335
Aspera gateway 可以改善高延遲網路的效能.....	337
<b>注意事項</b> .....	<b>339</b>
程式設計介面資訊.....	340
商標.....	340



# IBM MQ 監視及效能

---

使用本節中的監視資訊及指引，以及特定的調整提示，來協助改善佇列管理程式網路的效能。

## 關於這項作業

視佇列管理程式網路的大小及複雜性而定，您可以從監視網路取得一系列資訊。您可以使用該資訊，以及特定調整提示中提供的資訊，來協助您調整網路效能。

## 監視 IBM MQ 網路

---

IBM MQ 中提供許多監視技術，可取得佇列管理程式網路執行方式的統計資料及其他特定相關資訊。請使用本節中的監視資訊及指引，以協助改善佇列管理程式網路的效能。

下列清單提供監視佇列管理程式網路的原因範例：

- 偵測佇列管理程式網路中的問題。
- 協助判斷佇列管理程式網路中問題的原因。
- 提高佇列管理程式網路的效率。
- 熟悉佇列管理程式網路的執行。
- 請確認佇列管理程式網路正確執行中。
- 發生特定事件時產生訊息。
- 記錄訊息活動。
- 決定訊息的前次已知位置。
- 即時檢查佇列管理程式網路的各種統計資料。
- 產生審核追蹤。
- 應用程式資源用量的帳戶。
- 產能規劃。

## 事件監視

事件監視是在佇列管理程式網路中偵測發生 檢測事件 的程序。檢測事件是由佇列管理程式或通道實例偵測到的事件邏輯組合。這類事件會導致佇列管理程式或通道實例將特殊訊息 (稱為 事件訊息) 放置在事件佇列上。

IBM MQ 檢測事件提供佇列管理程式中錯誤、警告及其他重要事件的相關資訊。使用這些事件來監視佇列管理程式網路中的佇列管理程式作業，以達到下列目標：

- 偵測佇列管理程式網路中的問題。
- 協助判斷佇列管理程式網路中問題的原因。
- 產生審核追蹤。
- 回應佇列管理程式狀態變更

### 相關參考

第 8 頁的『事件類型』

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

[事件訊息參照](#)

[事件訊息格式](#)

## 檢測事件

檢測事件是佇列管理程式或通道實例偵測並將特殊訊息 (稱為 事件訊息) 放置在事件佇列上之條件的邏輯組合。

IBM MQ 檢測事件提供佇列管理程式中錯誤、警告及其他重要事件的相關資訊。 您可以使用這些事件來監視佇列管理程式的作業 (使用其他方法, 例如 Tivoli NetView for z/OS)。

第 7 頁的圖 1 說明檢測事件的概念。

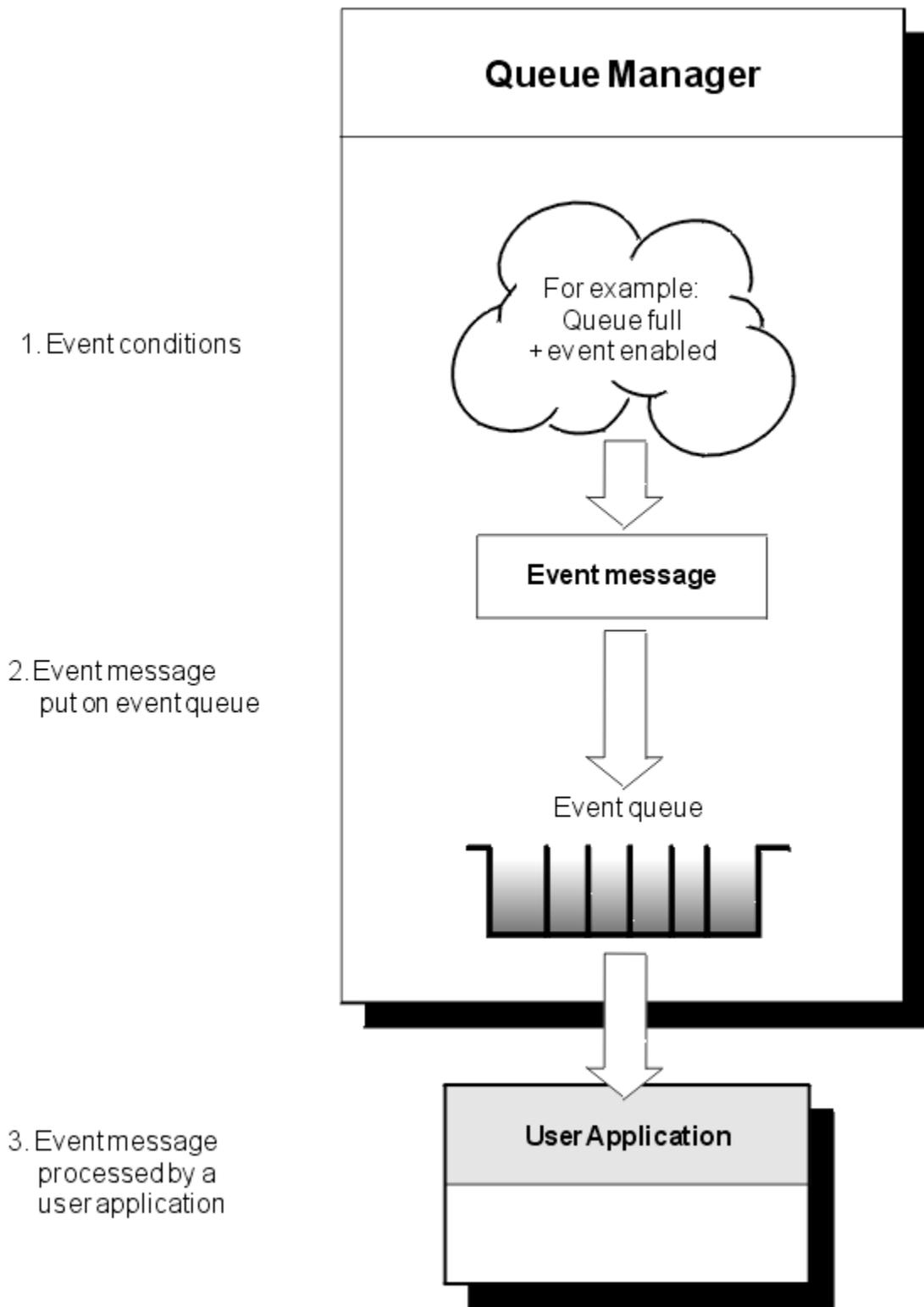


圖 1: 瞭解檢測事件

### 事件監視應用程式

使用事件來監視佇列管理程式的應用程式必須包含下列規定:

1. 設定網路中佇列管理程式之間的通道。

2. 實作必要的資料轉換。資料轉換的一般規則適用。例如，如果您從 z/OS 佇列管理程式監視 UNIX 系統佇列管理程式上的事件，請確保將 EBCDIC 轉換為 ASCII。

## 透過事件佇列的事件通知

當事件發生時，佇列管理程式會將事件訊息放入適當的事件佇列 (如果已定義的話)。事件訊息包含您可以透過撰寫執行下列步驟的適當 MQI 應用程式來擷取之事件的相關資訊：

- 從佇列取得訊息。
- 處理訊息以擷取事件資料。

相關資訊說明事件訊息的格式。

## 導致事件的條件

下列清單提供可能導致檢測事件的條件範例：

- 已達到佇列上訊息數的臨界值限制。
- 通道實例已啟動或已停止。
- 佇列管理程式會變成作用中，或被要求停止。
- 應用程式嘗試開啟佇列，指定在 IBM MQ for IBM i、Windows、UNIX and Linux® 系統上未獲授權的使用者 ID。
- 會建立、刪除、變更或重新整理物件。
- MQSC 或 PCF 指令順利執行。
- 佇列管理程式會開始寫入新的日誌範圍。
- 如果符合事件條件，則將訊息放置在無法傳送郵件的佇列上。

## 相關概念

第 18 頁的『效能事件』

效能事件與可能影響使用指定佇列之應用程式效能的狀況相關。效能事件的範圍是佇列。某個佇列上的 **MQPUT** 呼叫及 **MQGET** 呼叫不會影響在另一個佇列上產生效能事件。

第 47 頁的『在 Multiplatforms 上監視檢測事件的範例程式』

**amqsevt** 會格式化佇列管理程式可以建立且隨 IBM MQ for Multiplatforms 提供的檢測事件。程式會從事件佇列讀取訊息，並將它們格式化成可讀取的字串。

## 事件類型

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

IBM MQ 檢測事件具有下列類型：

- 佇列管理程式事件
- 通道及橋接器事件
- 效能事件
- 配置事件
- 指令事件
- 日誌程式事件
- 本端事件

對於每一個佇列管理程式，每一個事件種類都有自己的事件佇列。該種類中的所有事件都會導致將事件訊息放入相同的佇列。

### 此事件佇列：

SYSTEM.ADMIN.QMGR.EVENT  
SYSTEM.ADMIN.CHANNEL.EVENT

### 包含來自下列的訊息：

佇列管理程式事件  
通道事件

### 此事件佇列:

SYSTEM.ADMIN.PERFM.EVENT  
SYSTEM.ADMIN.CONFIG.EVENT  
SYSTEM.ADMIN.COMMAND.EVENT  
SYSTEM.ADMIN.LOGGER.EVENT  
SYSTEM.ADMIN.PUBSUB.EVENT

### 包含來自下列的訊息:

效能事件  
配置事件  
指令事件  
日誌程式事件  
取得與發佈/訂閱相關的事件。僅與多重播送一起使用。如需相關資訊，請參閱 [多重播送應用程式監視](#)。

透過將檢測事件併入您自己的系統管理應用程式，您可以監視多個佇列管理程式之間、多個不同節點之間的活動，以及多個 IBM MQ 應用程式的活動。特別是，您可以從單一節點 (針對支援 IBM MQ 事件的節點) 監視系統中的所有節點，如第 9 頁的圖 2 所示。

檢測事件可以透過使用者撰寫的報告機制向管理應用程式報告，該管理應用程式可以將事件呈現給操作員。

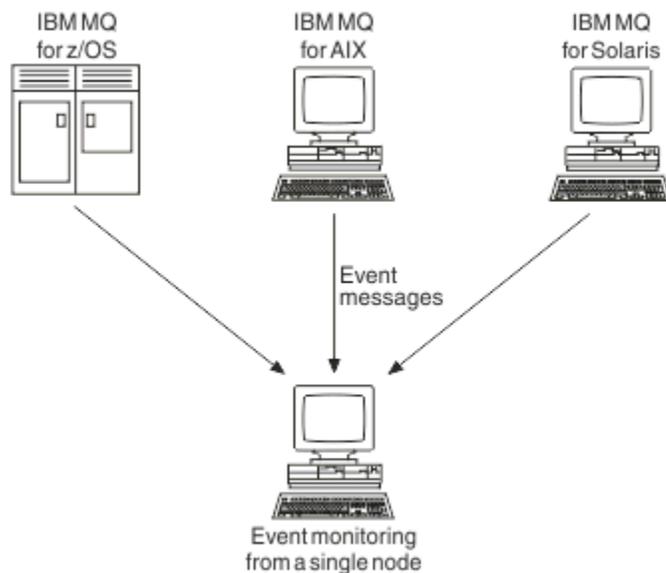


圖 2: 在單一節點上跨不同平台監視佇列管理程式

檢測事件也可讓應用程式作為其他管理網路 (例如 Tivoli NetView for z/OS) 的代理程式，以監視報告並建立適當的警示。

### 佇列管理程式事件

佇列管理程式事件與使用佇列管理程式內的資源相關。例如，如果應用程式嘗試將訊息放置在不存在的佇列上，則會產生佇列管理程式事件。

下列範例顯示可能導致佇列管理程式事件的條件:

- 應用程式發出失敗的 MQI 呼叫。呼叫中的原因碼與事件訊息中的原因碼相同。

在佇列管理程式的內部作業期間可能會發生類似狀況; 例如，產生報告訊息時。事件訊息中的原因碼可能符合 MQI 原因碼，即使它未與任何應用程式相關聯。請勿假設，因為事件訊息原因碼看起來像是 MQI 原因碼，所以事件必然是由來自應用程式的未順利完成 MQI 呼叫所造成。

- 向佇列管理程式發出指令，且處理此指令會導致事件。例如：
  - 佇列管理程式已停止或已啟動。
  - 發出指令時，相關聯的使用者 ID 未獲該指令的授權。

IBM MQ 會將佇列管理程式事件的訊息放置在 SYSTEM.ADMIN.QMGR.EVENT 佇列，並支援下列佇列管理程式事件類型:

**權限 (僅限 Windows 及 UNIX)**

權限事件會報告授權，例如應用程式嘗試開啟沒有必要權限的佇列，或從沒有必要權限的使用者 ID 發出指令。權限事件訊息可以包含下列事件資料：

- [未獲授權 \(類型 1\)](#)
- [未獲授權 \(類型 2\)](#)
- [未獲授權 \(類型 3\)](#)
- [未獲授權 \(類型 4\)](#)
- [未獲授權 \(類型 5\)](#)
- [未獲授權 \(類型 6\)](#)

所有權限事件僅適用於 Windows 及 UNIX。

**禁止**

禁止事件指出已針對禁止放置或取得佇列的佇列，或針對禁止發佈主題的主題，嘗試 MQPUT 或 MQGET 作業。禁止事件訊息可以包含下列事件資料：

- [禁止取得](#)
- [禁止放置](#)

**本端**

當應用程式或佇列管理程式無法存取本端佇列或其他本端物件 (例如，因為尚未定義物件) 時，佇列管理程式可以產生本端事件訊息。本端事件訊息可以包含下列事件資料：

- [別名基本佇列類型錯誤](#)
- [不明的別名基本佇列](#)
- [不明的物件名稱](#)

**遠端**

當應用程式或佇列管理程式無法存取另一個佇列管理程式上的遠端佇列 (例如，未正確定義傳輸佇列) 時，佇列管理程式可以產生遠端事件訊息。遠端事件訊息可以包含下列事件資料：

- [預設傳輸佇列類型錯誤](#)
- [預設傳輸佇列用法錯誤](#)
- [佇列類型錯誤](#)
- [遠端佇列名稱錯誤](#)
- [傳輸佇列類型錯誤](#)
- [傳輸佇列使用錯誤](#)
- [不明預設傳輸佇列](#)
- [不明的遠端佇列管理程式](#)
- [不明的傳輸佇列](#)

**啟動及停止**

啟動和停止事件指出佇列管理程式已啟動或已要求停止或靜止。

**z/OS** z/OS 僅支援開始事件。

除非 SYSTEM.ADMIN.QMGR.EVENT 佇列定義為持續性。啟動和停止事件訊息可以包含下列事件資料：

- [佇列管理程式作用中](#)
- [佇列管理程式非作用中](#)

對於此清單中的每一種事件類型，您可以設定佇列管理程式屬性來啟用或停用事件類型。

**通道及橋接器事件**

通道會報告這些事件，因為在其作業期間偵測到條件。例如，當通道實例停止時。

在下列情況下會產生通道事件：

- 當指令啟動或停止通道時。
- 通道實例啟動或停止時。
- 當通道在取得訊息時收到轉換錯誤警告。
- 嘗試自動建立通道時，不論嘗試成功或失敗都會產生事件。

註：用戶端連線不會導致「通道已啟動」或「通道已停止」事件。

當使用指令來啟動通道時，會產生事件。當通道實例啟動時，會產生另一個事件。不過，由接聽器、**runmqchl** 指令或佇列管理程式觸發訊息啟動通道不會產生事件。在這些情況下，只有在通道實例啟動時才會產生事件。

成功啟動或停止通道指令至少會產生兩個事件。這些事件是針對通道所連接的兩個佇列管理程式所產生 (前提是它們支援事件)。

如果將通道事件放置在事件佇列上，則錯誤狀況會導致佇列管理程式建立事件。

通道及橋接器事件的事件訊息會放置在 SYSTEM.ADMIN.CHANNEL.EVENT 佇列。

通道事件訊息可以包含下列事件資料：

- [通道已啟動](#)
- [通道自動定義錯誤](#)
- [通道自動定義正常](#)
- [通道轉換錯誤](#)
- [未啟動通道](#)
- [通道已啟動](#)
- [停止通道](#)
- [使用者已停止通道](#)
- [已封鎖通道](#)

## IMS 橋接器事件 (僅限 z/OS)



當 IMS 橋接器啟動或停止時，會報告這些事件。

IMS 橋接器事件訊息可以包含下列事件資料：

- [橋接已啟動](#)
- [停止橋接](#)

## SSL 事件

唯一的 TLS 事件是「通道 SSL 錯誤」事件。使用 TLS 的通道無法建立 TLS 連線時，會報告此事件。

SSL 事件訊息可以包含下列事件資料：

- [通道 SSL 錯誤](#)
- [通道 SSL 警告](#)

## 效能事件

效能事件是指資源已達到臨界條件的通知。例如，已達到佇列深度限制。

效能事件與可能影響使用指定佇列之應用程式效能的狀況相關。不會針對事件佇列本身產生它們。

事件類型會在訊息資料的指令 ID 欄位中傳回。

如果佇列管理程式嘗試將佇列管理程式事件或效能事件訊息放置在事件佇列上，且偵測到通常會建立事件的錯誤，則不會建立另一個事件，且不會採取任何動作。

不論工作單元是已確定還是已取消，工作單元內的 MQGET 和 MQPUT 呼叫都可以產生效能事件。

效能事件的事件訊息會放置在 SYSTEM.ADMIN.PERFM.EVENT 佇列。

效能事件有兩種類型：

#### 佇列深度事件

佇列深度事件與佇列上的訊息數相關；亦即佇列已滿或清空的程度。共用佇列支援這些事件。佇列深度事件訊息可以包含下列事件資料：

- [佇列深度偏高](#)
- [佇列深度低](#)
- [佇列已滿](#)

#### 佇列服務間隔事件

佇列服務間隔事件與是否在使用者指定的時間間隔內處理訊息相關。共用佇列不支援這些事件。

 IBM MQ for z/OS 支援 QSGDISP (SHARED) 佇列的佇列深度事件，但不支援服務間隔事件。佇列管理程式及通道事件不受共用佇列影響。佇列服務事件訊息可以包含下列事件資料：

- [佇列服務間隔高](#)
- [佇列服務間隔正常](#)

#### 配置事件

當明確要求配置事件時，會產生配置事件；當建立、修改或刪除物件時，會自動產生配置事件。

配置事件訊息包含物件屬性的相關資訊。例如，如果建立名單物件，則會產生配置事件訊息，並包含名單物件屬性的相關資訊。

配置事件的事件訊息會放置在 SYSTEM.ADMIN.CONFIG.EVENT 佇列。

配置事件以下列方式運作：

- 佇列管理程式配置的變更審核追蹤是透過事件訊息來保留，事件訊息會寫入 SYSTEM.ADMIN.CONFIG.EVENT 佇列。您可以透過 [ALTER QMGR](#) 指令上的 **CONFIGEV** 參數來寫入這些事件。
- 當 DEFINE、ALTER 或 DELETE 指令處理物件或使用 MQSET 呼叫時，會產生這些事件。
- 您可以使用 [REFRESH QMGR TYPE \(CONFIGEV\)](#) 指令來建立現行佇列管理程式配置的基準線圖片，這會針對佇列管理程式中的每個物件建立事件訊息。請注意，由於這可能是一項耗時的作業，如果您有許多物件，您可以在指令上使用 NAME 和 OBJECT 限定元，將作業細分成較小的物件集。
- 事件訊息會記錄下列四個可能原因之一：
  - MQRC\_CONFIG\_CHANGE\_OBJECT
  - MQRC\_CONFIG\_CREATE\_OBJECT
  - MQRC\_CONFIG\_DELETE\_OBJECT
  - MQRC\_CONFIG\_REFRESH\_OBJECT

MQRC\_CONFIG\_CHANGE\_OBJECT、MQRC\_CONFIG\_CREATE\_OBJECT 或 MQRC\_CONFIG\_DELETE\_OBJECT 會針對您可能對物件發出的個別 MQSC 或 PCF 指令發生。

在建立基準線圖片時，針對寫入的那些事件訊息會發生 MQRC\_CONFIG\_REFRESH\_OBJECT。

配置事件有四種類型：

#### 建立物件事件

建立物件事件會在建立物件時產生。事件訊息包含下列事件資料：[建立物件](#)。

#### 變更物件事件

變更物件事件會在變更物件時產生。事件訊息包含下列事件資料：[變更物件](#)。

#### 刪除物件事件

刪除物件事件會在刪除物件時產生。事件訊息包含下列事件資料：[刪除物件](#)。

#### 重新整理物件事件

重新整理物件事件是由明確的重新整理要求所產生。事件訊息包含下列事件資料：[Refresh object](#)。

## 指令事件

當 MQSC 或 PCF 指令順利執行時，會報告指令事件。

指令事件訊息包含指令來源、環境定義及內容的相關資訊。例如，如果 MQSC 指令 ALTER QLOCAL 順利執行，則會產生具有這類資訊的指令事件訊息。

指令事件的事件訊息會放置在 SYSTEM.ADMIN.COMMAND.EVENT 佇列。

指令事件包含下列事件資料: [指令](#)。

## Multi 日誌程式事件

當使用線性記載的佇列管理程式開始將日誌記錄寫入新的日誌範圍 **IBM i** 或在 IBM i 上寫入新的日誌接收器時，會報告日誌程式事件。 **z/OS** IBM MQ for z/OS 無法使用日誌程式事件。

日誌程式事件訊息包含指定佇列管理程式重新啟動佇列管理程式或媒體回復所需的日誌範圍的資訊。

日誌程式事件的事件訊息會放置在 SYSTEM.ADMIN.LOGGER.EVENT 佇列。

日誌程式事件訊息包含下列事件資料: [Logger](#)。

## 事件訊息資料摘要

使用此摘要來取得每一種事件訊息類型可包含之事件資料的相關資訊。

事件類型	請參閱下列主題
權限事件	<a href="#">未獲授權 (類型 1)</a>
	<a href="#">未獲授權 (類型 2)</a>
	<a href="#">未獲授權 (類型 3)</a>
	<a href="#">未獲授權 (類型 4)</a>
	<a href="#">未獲授權 (類型 5)</a>
	<a href="#">未獲授權 (類型 6)</a>
通道事件	<a href="#">通道已啟動</a>
	<a href="#">通道自動定義錯誤</a>
	<a href="#">通道自動定義正常</a>
	<a href="#">已封鎖通道</a>
	<a href="#">通道轉換錯誤</a>
	<a href="#">未啟動通道</a>
	<a href="#">通道已啟動</a>
	<a href="#">停止通道</a>
	<a href="#">使用者已停止通道</a>
	指令事件
配置事件	<a href="#">建立物件</a>
	<a href="#">變更物件</a>
	<a href="#">刪除物件</a>
	<a href="#">重新整理物件</a>

事件類型	請參閱下列主題
IMS 橋接器事件	<a href="#">橋接已啟動</a>
	<a href="#">停止橋接</a>
禁止事件	<a href="#">禁止取得</a>
	<a href="#">禁止放置</a>
本端事件	<a href="#">別名基本佇列類型錯誤</a>
	<a href="#">不明的別名基本佇列</a>
	<a href="#">不明的物件名稱</a>
日誌程式事件	<a href="#">日誌程式</a>
效能事件	<a href="#">佇列深度偏高</a>
	<a href="#">佇列深度低</a>
	<a href="#">佇列已滿</a>
	<a href="#">佇列服務間隔高</a>
	<a href="#">佇列服務間隔正常</a>
遠端事件	<a href="#">預設傳輸佇列類型錯誤</a>
	<a href="#">預設傳輸佇列用法錯誤</a>
	<a href="#">佇列類型錯誤</a>
	<a href="#">遠端佇列名稱錯誤</a>
	<a href="#">傳輸佇列類型錯誤</a>
	<a href="#">傳輸佇列使用錯誤</a>
	<a href="#">不明預設傳輸佇列</a>
	<a href="#">不明的遠端佇列管理程式</a>
	<a href="#">不明的傳輸佇列</a>
SSL 事件	<a href="#">通道 SSL 錯誤</a>
啟動和停止事件	<a href="#">佇列管理程式作用中</a>
	<a href="#">佇列管理程式非作用中</a>

## 控制事件

視事件類型而定，您可以透過指定佇列管理程式及/或佇列屬性的適當值來啟用及停用事件。

您必須啟用要產生的每一個檢測事件。例如，導致「佇列已滿」事件的條件如下：

- 針對指定的佇列啟用「佇列已滿」事件，以及
- 應用程式發出 MQPUT 要求以將訊息放置在該佇列上，但要求失敗，因為佇列已滿。

使用下列任何技術來啟用及停用事件：

- IBM MQ Script 指令 (MQSC)。
- 對應的 IBM MQ PCF 指令。
-  z/OS 上佇列管理程式的作業及控制面板。
- IBM MQ Explorer.

註: 您只能透過指令來設定與佇列及佇列管理程式的事件相關的屬性。 MQI 呼叫 MQSET 不支援與事件相關的屬性。

### 相關概念

[第 6 頁的『檢測事件』](#)

檢測事件是佇列管理程式或通道實例偵測並將特殊訊息 (稱為 事件訊息) 放置在事件佇列上之條件的邏輯組合。

[介紹作業及控制面板](#)

### 相關工作

[自動化管理作業](#)

[使用可程式指令格式](#)

### 相關參考

[第 8 頁的『事件類型』](#)

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

### MQSC 指令

#### 控制佇列管理程式事件

您可以使用佇列管理程式屬性來控制佇列管理程式事件。若要啟用佇列管理程式事件，請將適當的佇列管理程式屬性設為 ENABLED。若要停用佇列管理程式事件，請將適當的佇列管理程式屬性設為 DISABLED。

若要啟用或停用佇列管理程式事件，請使用 MQSC 指令 **ALTER QMGR**，並指定適當的佇列管理程式屬性。第 15 頁的表 1 彙總如何啟用佇列管理程式事件。若要停用佇列管理程式事件，請將適當的參數設為 DISABLED。

事件	ALTER QMGR 參數
權限管理中心 禁止 本端 遠端 啟動及停止	AUTHOREV (ENABLED) 抑制 (ENABLED) LOCALEV (ENABLED) REMOTEEV (ENABLED) STRSTPEV (ENABLED)

#### 控制通道及橋接器事件

您可以使用佇列管理程式屬性來控制通道事件。若要啟用通道事件，請將適當的佇列管理程式屬性設為 ENABLED。若要停用通道事件，請將適當的佇列管理程式屬性設為 DISABLED。

若要啟用或停用通道事件，請使用 MQSC 指令 **ALTER QMGR**，並指定適當的佇列管理程式屬性。第 15 頁的表 2 彙總如何啟用通道及橋接器事件。若要停用佇列管理程式事件，請將適當的參數設為 DISABLED。

**限制:**  在 IBM MQ for z/OS 上無法使用通道自動定義事件。

事件	ALTER QMGR 參數
通道 僅與通道錯誤相關 IMS 橋接 SSL 通道自動定義	CHLEV (已啟用) CHLEV (異常狀況) BRIDGEV (ENABLED) SSLEV (已啟用) CHADEV (ENABLED)

當 CHLEV 設為異常狀況時，會產生下列回覆碼及對應的原因限定元:

- 已啟動 MQRC\_CHANNEL\_ACTIVATED
- MQRC\_CHANNEL\_CONV\_ERROR

- 已啟動 MQRC\_CHANNEL\_NOT\_ACTIVATED
- MQRC\_CHANNEL\_STOPPED
  - 具有下列 ReasonQualifiers:
    - MQRC\_CHANNEL\_STOPPED\_ERROR
    - MQRC\_CHANNEL\_STOPPED\_RETRY
    - MQRC\_CHANNEL\_STOPPED\_DISABLED
- MQRC\_CHANNEL\_STOPPED\_BY\_USER
- MQRC\_CHANNEL\_BLOCKED
  - 具有下列 ReasonQualifiers:
    - MQRC\_CHANNEL\_BLOCKED\_NOACCESS
    - MQRC\_CHANNEL\_BLOCKED\_USERID
    - MQRC\_CHANNEL\_BLOCKED\_ADDRESS

#### 控制效能事件

您可以使用 PERFMEV 佇列管理程式屬性來控制效能事件。若要啟用效能事件，請將 PERFMEV 設為 ENABLED。若要停用效能事件，請將 PERFMEV 佇列管理程式屬性設為 DISABLED。

若要將 PERFMEV 佇列管理程式屬性設為 ENABLED，請使用下列 MQSC 指令：

```
ALTER QMGR PERFMEV (ENABLED)
```

若要啟用特定的效能事件，請設定適當的佇列屬性。此外，請指定導致事件的條件。

#### 佇列深度事件

依預設，會停用所有佇列深度事件。若要為任何佇列深度事件配置佇列，請執行下列動作：

1. 在佇列管理程式上啟用效能事件。
2. 在必要佇列上啟用事件。
3. 必要的話，將限制設為適當的層次，以佇列深度上限的百分比表示。

#### 佇列服務間隔事件

若要配置佇列服務間隔事件的佇列，您必須：

1. 在佇列管理程式上啟用效能事件。
2. 視需要在佇列上設定「佇列服務間隔高」或「正常」事件的控制屬性。
3. 將佇列的 QSVCINT 屬性設為適當的時間長度，以指定服務間隔時間。

**註：**啟用後，可以在任何適當時間產生佇列服務間隔事件，不必等到對佇列發出 MQI 呼叫為止。不過，如果在佇列上使用 MQI 呼叫來放置或移除訊息，則會在該時間產生任何適用的效能事件。當經歷時間變成等於服務間隔時間時，不會產生事件。

#### 控制配置、指令及日誌程式事件

您可以使用佇列管理程式屬性 CONFIGEV、CMDEV 及 LOGGEREV 來控制配置、指令及日誌程式事件。若要啟用這些事件，請將適當的佇列管理程式屬性設為 ENABLED。若要停用這些事件，請將適當的佇列管理程式屬性設為 DISABLED。

#### 配置事件

若要啟用配置事件，請將 CONFIGEV 設為 ENABLED。如果要停用配置事件，請將 CONFIGEV 設為 DISABLED。例如，您可以使用下列 MQSC 指令來啟用配置事件：

```
ALTER QMGR CONFIGEV (ENABLED)
```

## 指令事件

若要啟用指令事件，請將 CMDEV 設為 ENABLED。若要啟用指令 (DISPLAY MQSC 指令及 Inquire PCF 指令除外) 的指令事件，請將 CMDEV 設為 NODISPLAY。若要停用指令事件，請將 CMDEV 設為 DISABLED。例如，您可以使用下列 MQSC 指令來啟用指令事件：

```
ALTER QMGR CMDEV (ENABLED)
```

## 日誌程式事件

若要啟用日誌程式事件，請將 LOGGEREV 設為 ENABLED。若要停用日誌程式事件，請將 LOGGEREV 設為 DISABLED。例如，您可以使用下列 MQSC 指令來啟用日誌程式事件：

```
ALTER QMGR LOGGEREV(ENABLED)
```

## 事件佇列

當事件發生時，佇列管理程式會將事件訊息放置在已定義的事件佇列上。事件訊息包含事件的相關資訊。

您可以將事件佇列定義為：

- 本端佇列
- 別名佇列
- 遠端佇列的本端定義，或作為
- 遠端叢集佇列

如果您在一個佇列管理程式上將所有事件佇列定義為相同遠端佇列的本端定義，則可以集中監視活動。

您不得將事件佇列定義為傳輸佇列，因為事件訊息的格式與傳輸佇列所需的訊息格式不相容。

共用事件佇列是使用 QSGDISP (SHARED) 值定義的本端佇列。

如需在 z/OS 上定義共用佇列的相關資訊，請參閱 [使用共用佇列進行應用程式設計](#)。

## 當事件佇列無法使用時

如果事件佇列無法使用時發生事件，則會遺失事件訊息。例如，如果您未定義事件種類的事件佇列，則該種類的所有事件訊息都會遺失。例如，事件訊息不會儲存在無法傳送的郵件 (無法遞送的訊息) 佇列中。

不過，您可以將事件佇列定義為遠端佇列。然後，如果遠端系統將訊息放入已解決的佇列時發生問題，則事件訊息會到達遠端系統的無法傳送郵件的佇列。

事件佇列可能因為許多不同的原因而無法使用，包括：

- 尚未定義佇列。
- 已刪除佇列。
- 佇列已滿。
- 已禁止放置佇列。

缺少事件佇列並不會阻止事件發生。例如，在效能事件之後，佇列管理程式會變更佇列屬性並重設佇列統計資料。不論事件訊息是否放置在效能事件佇列上，都會發生此變更。在配置及指令事件的情況下也是如此。

## 使用觸發事件佇列

您可以設定具有觸發程式的事件佇列，以便在產生事件時，放置在事件佇列上的事件訊息會啟動使用者撰寫的監視應用程式。此應用程式可以處理事件訊息並採取適當的動作。例如，某些事件可能需要通知操作員，其他事件可能會啟動自動執行部分管理作業的應用程式。

事件佇列可以具有與其相關聯的觸發動作，並且可以建立觸發訊息。不過，如果這些觸發訊息依序導致通常會產生事件的狀況，則不會產生任何事件。未在此實例中產生事件，可確保不會發生迴圈。

## 相關概念

[第 14 頁的『控制事件』](#)

視事件類型而定，您可以透過指定佇列管理程式及/或佇列屬性的適當值來啟用及停用事件。

[第 18 頁的『事件訊息的格式』](#)

事件訊息包含事件及其原因的相關資訊。與其他 IBM MQ 訊息一樣，事件訊息有兩個部分：訊息描述子和訊息資料。

[使用共用佇列進行應用程式設計](#)

[觸發事件的條件](#)

## 相關參考

[QSGDisp \(MQLONG\)](#)

## 事件訊息的格式

事件訊息包含事件及其原因的相關資訊。與其他 IBM MQ 訊息一樣，事件訊息有兩個部分：訊息描述子和訊息資料。

- 訊息描述子基於 MQMD 結構。
- 訊息資料由事件標頭及事件資料組成。事件標頭包含識別事件類型的原因碼。放置事件訊息及任何後續動作，不會影響導致事件的 MQI 呼叫所傳回的原因碼。事件資料提供事件的進一步相關資訊。

一般而言，您可以使用自訂的系統管理應用程式來處理事件訊息，以符合其執行所在企業的需求。

當佇列共用群組中的佇列管理程式偵測到產生事件訊息的條件時，數個佇列管理程式可以產生共用佇列的事件訊息，產生數個事件訊息。為了確保系統可以使來自不同佇列管理程式的多個事件訊息產生關聯，這些事件訊息在訊息描述子 (MQMD) 中設定唯一的相關性 ID (*CorrelId*)。

## 相關參考

[第 85 頁的『活動報告 MQMD \(訊息描述子\)』](#)

請利用這個頁面來檢視活動報告的 MQMD 結構所包含的值

[第 89 頁的『活動報告 MQEPH \(內嵌 PCF 標頭\)』](#)

請利用這個頁面來檢視活動報告的 MQEPH 結構所包含的值

[第 90 頁的『活動報告 MQCFH \(PCF 標頭\)』](#)

請利用這個頁面來檢視活動報告的 MQCFH 結構所包含的 PCF 值

[事件訊息參照](#)

[事件訊息格式](#)

[事件訊息 MQMD \(訊息描述子\)](#)

[事件訊息 MQCFH \(PCF 標頭\)](#)

[事件訊息說明](#)

## 效能事件

效能事件與可能影響使用指定佇列之應用程式效能的狀況相關。效能事件的範圍是佇列。某個佇列上的 **MQPUT** 呼叫及 **MQGET** 呼叫不會影響在另一個佇列上產生效能事件。

效能事件訊息可以在任何適當的時間產生，不一定要等到對佇列發出 MQI 呼叫為止。不過，如果您在佇列上使用 MQI 呼叫來放置或移除訊息，則會在該時間產生任何適當的效能事件。

所產生的每一個效能事件訊息都會放置在佇列 SYSTEM.ADMIN.PERFM.EVENT。

事件資料包含一個原因碼，可識別事件的原因、一組效能事件統計資料，以及其他資料。下列清單說明可在效能事件訊息中傳回的事件資料類型：

- [佇列深度偏高](#)
- [佇列深度低](#)
- [佇列已滿](#)
- [佇列服務間隔高](#)
- [佇列服務間隔正常](#)

說明如何使用效能事件的範例假設您使用適當的 IBM MQ 指令 (MQSC) 來設定佇列屬性。在 z/OS 上，您也可以使用佇列管理程式的作業及控制項畫面來設定佇列屬性。

## 相關參考

[第 8 頁的『事件類型』](#)

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

## 效能事件統計資料

事件訊息中的效能事件資料包含事件的相關統計資料。使用統計資料來分析指定佇列的行為。

事件訊息中的事件資料包含系統管理程式的事件相關資訊。對於所有效能事件，事件資料包含佇列管理程式的名稱，以及與事件相關聯的佇列。事件資料也包含與事件相關的統計資料。[第 19 頁的表 3](#) 彙總您可以用來分析佇列行為的事件統計資料。所有統計資料都是指自前次重設統計資料以來所發生的情況。

參數	說明
TimeSinceReset	自前次重設統計資料以來所經歷的時間。
HighQDepth	自前次重設統計資料以來佇列上的訊息數上限。
MsgEnqCount	自前次重設統計資料以來放入佇列的訊息數 (對佇列的 MQPUT 呼叫數)。
MsgDeqCount	自前次重設統計資料以來已移出佇列的訊息數 (對佇列的 MQGET 呼叫數)。

當發生下列任何變更時，會重設效能事件統計資料：

- 發生效能事件 (重設所有作用中佇列管理程式的統計資料)。
- 佇列管理程式會停止並重新啟動。
- 從應用程式發出 PCF 指令「重設佇列統計資料」。
-  僅在 z/OS 上，會在主控台發出 RESET QSTATS 指令。

## 相關概念

[第 18 頁的『效能事件』](#)

效能事件與可能影響使用指定佇列之應用程式效能的狀況相關。效能事件的範圍是佇列。某個佇列上的 MQPUT 呼叫及 MQGET 呼叫不會影響在另一個佇列上產生效能事件。

[第 21 頁的『服務計時器』](#)

佇列服務間隔事件使用稱為服務計時器的內部計時器，由佇列管理程式控制。只有在啟用佇列服務間隔事件時，才會使用服務計時器。

[第 21 頁的『佇列服務間隔事件的規則』](#)

正式規則控制何時設定服務計時器及產生佇列服務間隔事件。

## 相關工作

[第 22 頁的『啟用佇列服務間隔事件』](#)

若要配置佇列服務間隔事件的佇列，您可以設定適當的佇列管理程式及佇列屬性。

## 相關參考

[佇列深度偏高](#)

[重設佇列統計資料](#)

[重設 QSTATS](#)

## 佇列服務間隔事件

佇列服務間隔事件指出是否在使用者定義的時間間隔 (稱為服務間隔) 內對佇列執行作業。視您的安裝而定，您可以使用佇列服務間隔事件來監視訊息是否快速離開佇列。

共用佇列不支援佇列服務間隔事件。

可能會發生下列類型的佇列服務間隔事件，其中術語 取得作業 是指從佇列中移除訊息的 **MQGET** 呼叫或活動，例如使用 **CLEAR QLOCAL** 指令：

### 佇列服務間隔正常

指出在下列其中一項作業之後：

- MQPUT 呼叫
- 保留非空佇列的 get 作業

已在使用者定義的時段 (稱為 服務間隔) 內執行取得作業。

只有 get 作業可以導致「佇列服務間隔 OK」事件訊息。「佇列服務間隔正常」事件有時稱為「正常」事件。

### 佇列服務間隔高

指出在下列其中一項作業之後：

- MQPUT 呼叫
- 保留非空佇列的 get 作業

在使用者定義的服務間隔內 未 執行取得作業。

取得作業或 MQPUT 呼叫可能導致「佇列服務間隔高」事件訊息。「佇列服務間隔高」事件有時稱為「高」事件。

若要同時啟用「佇列服務間隔正常」及「佇列服務間隔高」事件，請將 `QServiceIntervalEvent` 控制屬性設為「高」。當產生「佇列服務間隔高」事件時，會自動啟用「佇列服務間隔正常」事件。您不需要獨立啟用「佇列服務間隔正常」事件。

「正常」與「高」事件互斥，因此如果其中一個已啟用，則會停用另一個。不過，可以同步停用這兩個事件。

第 20 頁的圖 3 顯示佇列深度與時間的圖形。在 P1 時間，應用程式會發出 MQPUT，將訊息放置在佇列上。在時間 G1，另一個應用程式會發出 MQGET 來移除佇列中的訊息。

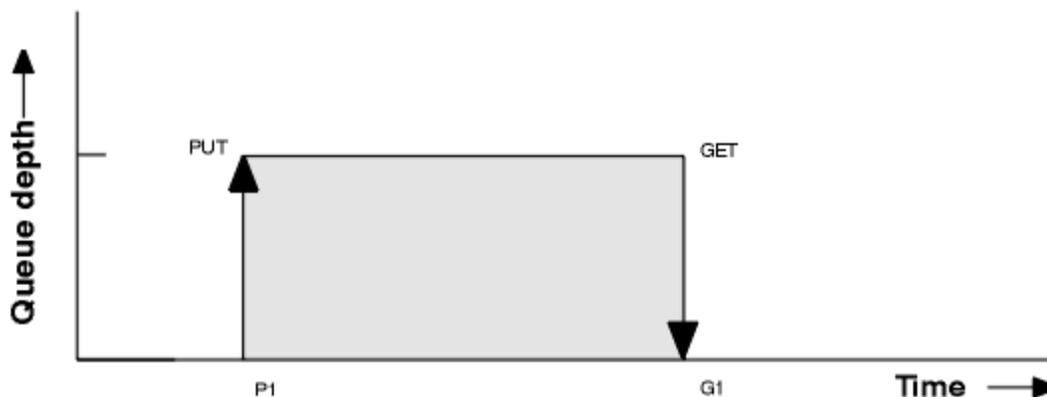


圖 3: 瞭解佇列服務間隔事件

佇列服務間隔事件的可能結果如下：

- 如果 put 與 get 之間的經歷時間小於或等於服務間隔：
  - 如果啟用佇列服務間隔事件，則會在時間 G1 產生 佇列服務間隔正常 事件
- 如果 put 與 get 之間的經歷時間大於服務間隔：
  - 如果啟用佇列服務間隔事件，則會在時間 G1 產生 佇列服務間隔高 事件。

第 21 頁的『佇列服務間隔事件的規則』中說明啟動服務計時器及產生事件的演算法。

### 相關參考

[佇列服務間隔正常](#)

[佇列服務間隔高](#)

[QServiceInterval 事件 \(MQLONG\)](#)

## QServiceInterval 事件 (10 位數帶正負號的整數)

### ServiceInterval 事件內容

#### 服務計時器

佇列服務間隔事件使用稱為 服務計時器的內部計時器，由佇列管理程式控制。只有在啟用佇列服務間隔事件時，才會使用服務計時器。

#### 服務計時器精確測量的是什麼？

服務計時器會測量對空佇列或取得作業的 MQPUT 呼叫與下一個放置或取得作業之間的經歷時間，前提是這兩個作業之間的佇列深度不是零。

#### 服務計時器何時在作用中？

如果佇列上有訊息 (深度為非零)，且已啟用佇列服務間隔事件，則服務計時器一律處於作用中 (執行中)。如果佇列變成空的 (佇列深度為零)，則會將計時器置於 OFF 狀態，並在下一次放置時重新啟動。

#### 何時重設服務計時器？

在取得作業之後，一律會重設服務計時器。它也會由對空佇列的 MQPUT 呼叫重設。不過，它不一定會在佇列服務間隔事件上重設。

#### 如何使用服務計時器？

在取得作業或 MQPUT 呼叫之後，佇列管理程式會比較服務計時器所測量的經歷時間與使用者定義的服務間隔。此比較的結果如下：

- 如果有 get 作業，且經歷時間小於或等於服務間隔，且已啟用此事件，則會產生 OK 事件。
- 如果經歷時間大於服務間隔，且已啟用此事件，則會產生高事件。

#### 應用程式可以讀取服務計時器嗎？

否，服務計時器是應用程式無法使用的內部計時器。

#### TimeSinceReset 參數如何？

TimeSinceReset 參數會作為事件資料中事件統計資料的一部分傳回。除非重設事件統計資料，否則它會指定連續佇列服務間隔事件之間的時間。

#### 佇列服務間隔事件的規則

正式規則控制何時設定服務計時器及產生佇列服務間隔事件。

## 服務計時器的規則

服務計時器會重設為零並重新啟動，如下所示：

- 對空佇列進行 MQPUT 呼叫之後。
- 在 MQGET 呼叫之後，如果佇列在 MQGET 呼叫之後不是空的。

計時器的重設並不取決於是否已產生事件。

在佇列管理程式啟動時，如果佇列深度大於零，則服務計時器會設為啟動時間。

如果在取得作業之後佇列是空的，則計時器會進入 OFF 狀態。

## 佇列服務間隔高事件

必須啟用「佇列服務間隔」事件 (設為 HIGH)。

當產生「佇列服務間隔正常」事件時，會自動啟用「佇列服務間隔高」事件。

如果服務時間大於服務間隔，則會在下一個 MQPUT 或 get 作業上或之前產生事件。

## 佇列服務間隔確定事件

當產生「佇列服務間隔高」事件時，會自動啟用「佇列服務間隔正常」事件。

如果服務時間 (經歷時間) 小於或等於服務間隔，則會在下一次取得作業上或之前產生事件。

## 相關工作

第 22 頁的『啟用佇列服務間隔事件』

若要配置佇列服務間隔事件的佇列，您可以設定適當的佇列管理程式及佇列屬性。

### 啟用佇列服務間隔事件

若要配置佇列服務間隔事件的佇列，您可以設定適當的佇列管理程式及佇列屬性。

## 關於這項作業

高事件與「確定」事件互斥；也就是說，當啟用一個事件時，另一個事件會自動停用：

- 在佇列上產生高事件時，佇列管理程式會自動停用高事件，並針對該佇列啟用「確定」事件。
- 在佇列上產生 OK 事件時，佇列管理程式會自動停用 OK 事件，並啟用該佇列的高事件。

佇列服務間隔事件	「佇列」屬性
佇列服務間隔高 佇列服務間隔正常 無佇列服務間隔事件	QSVCI EV (高) QSVCI EV (正常) QSVCI EV (NONE)
服務間隔 (service interval)	QSVCI NT ( <i>tt</i> )，其中 <i>tt</i> 是服務 間隔時間 (毫秒)。

請執行下列步驟來啟用佇列服務間隔事件：

## 程序

1. 將佇列管理程式屬性 **PERFMEV** 設為 ENABLED。  
在佇列管理程式上啟用效能事件。
2. 視需要為佇列上的「佇列服務間隔高」或「確定」事件設定控制屬性 **QSVCI EV**。
3. 設定佇列的 **QSVCI NT** 屬性，以指定適當的服務間隔時間。

## 範例

若要啟用服務間隔時間為 10 秒 (10 000 毫秒) 的「佇列服務間隔高事件」，請使用下列 MQSC 指令：

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QSVCI NT(10000) QSVCI EV(HIGH)
```

## 佇列服務間隔事件範例

請使用本節中的範例來瞭解您可以從佇列服務間隔事件取得的資訊。

這三個子主題範例提供漸進的更複雜圖解，說明佇列服務間隔事件的使用。

每個子主題中的範例隨附圖具有相同的結構：

- 圖 1 是佇列深度與時間的圖形，顯示個別 MQGET 呼叫和 MQPUT 呼叫。
- 「註解」區段顯示時間限制的比較。您必須考量三個時段：
  - 使用者定義服務間隔。
  - 服務計時器所測量的時間。
  - 自前次重設事件統計資料以來的時間 (事件資料中的 TimeSince 重設)。

- 「事件統計資料摘要」區段會顯示在任何即時啟用哪些事件，以及產生哪些事件。

這些範例說明佇列服務間隔事件的下列層面：

- 佇列深度隨時間變化的方式。
- 服務計時器所測量的經歷時間與服務間隔的比較。
- 啟用哪個事件。
- 產生哪些事件。

**記住：**範例 1 顯示一個簡式案例，其中訊息是間歇性，且在下一個訊息到達之前從佇列中移除每一個訊息。從事件資料中，您知道佇列上的訊息數上限是 1。因此，您可以找出每個訊息在佇列上的時間長度。

不過，在一般情況下，當佇列上有多個訊息，且 MQGET 呼叫和 MQPUT 呼叫的順序無法預測時，您無法使用佇列服務間隔事件來計算個別訊息保留在佇列上的時間長度。TimeSince 「重設」參數 (在事件資料中傳回) 可以包括佇列上沒有訊息時的時間比例。因此，您從這些統計資料衍生的任何結果都會隱含地平均，以包括這些時間。

### **相關概念**

第 19 頁的『佇列服務間隔事件』

佇列服務間隔事件指出是否在使用者定義的時間間隔 (稱為 服務間隔) 內對佇列執行作業。視您的安裝而定，您可以使用佇列服務間隔事件來監視訊息是否快速離開佇列。

第 21 頁的『服務計時器』

佇列服務間隔事件使用稱為 服務計時器的內部計時器，由佇列管理程式控制。只有在啟用佇列服務間隔事件時，才會使用服務計時器。

佇列服務間隔事件: 範例 1

MQGET 呼叫和 MQPUT 呼叫的基本序列，其中佇列深度一律為 1 或零。

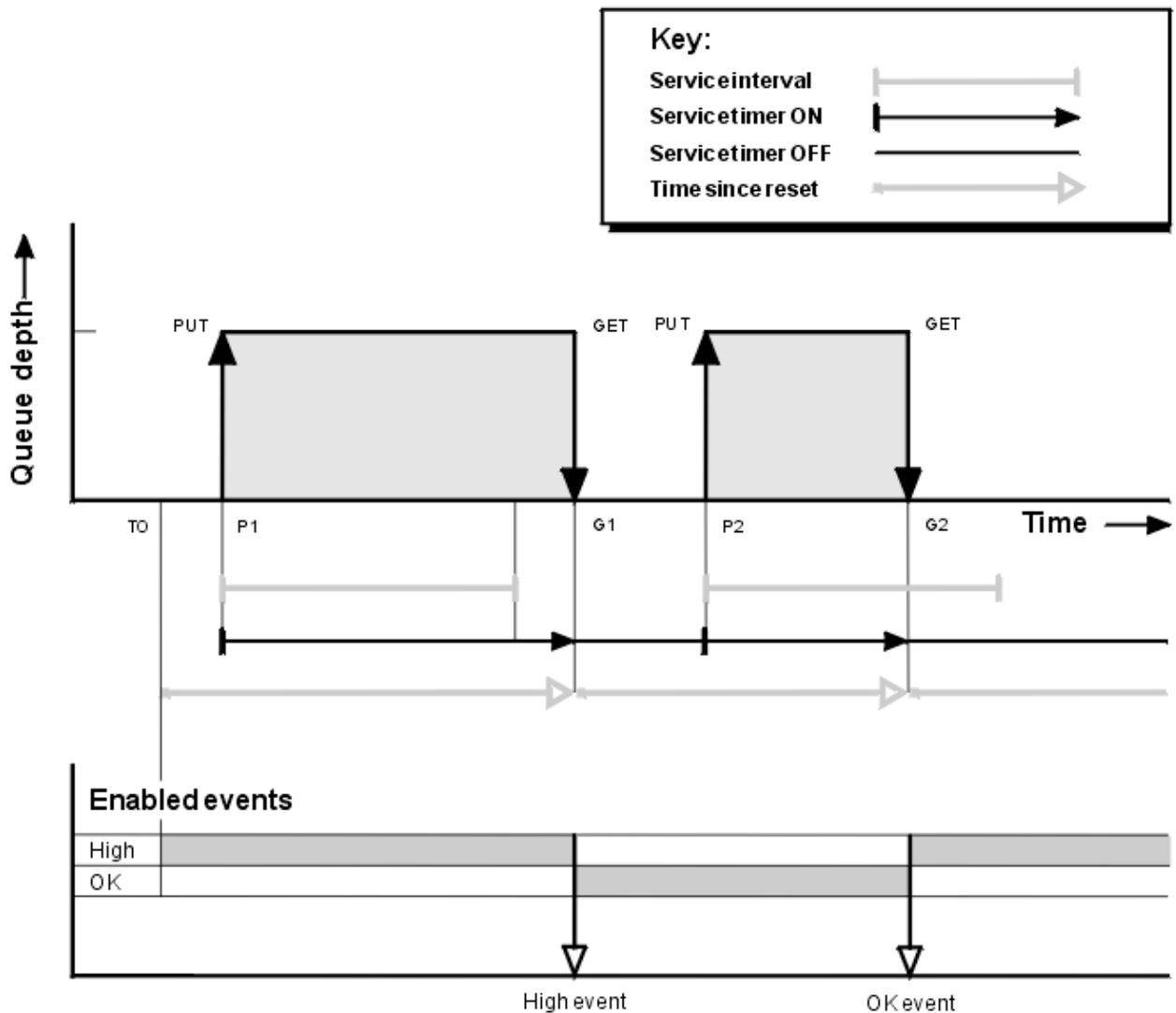


圖 4: 佇列服務間隔事件-範例 1

## 註解

- 在 P1, 應用程式會將訊息放入空佇列。這會啟動服務計時器。  
請注意, T0 可能是佇列管理程式啟動時間。
- 在 G1, 另一個應用程式會從佇列中取得訊息。因為 P1 與 G1 之間的經歷時間大於服務間隔, 所以在 G1 的 MQGET 呼叫上會產生「佇列服務間隔高」事件。產生高事件時, 佇列管理程式會重設事件控制屬性, 以便:
  - 會自動啟用「確定」事件。
  - 已停用高事件。
因為佇列現在是空的, 所以服務計時器會切換至 OFF 狀態。
- 在 P2, 第二則訊息會放入佇列中。這會重新啟動服務計時器。
- 在 G2, 訊息會從佇列中移除。不過, 因為 P2 與 G2 之間的經歷時間小於服務間隔, 所以在 G2 的 MQGET 呼叫上會產生「佇列服務間隔正常」事件。當產生 OK 事件時, 佇列管理程式會重設控制屬性, 以便:
  - 會自動啟用高事件。
  - 「確定」事件已停用。

因為佇列是空的，所以服務計時器再次切換至 OFF 狀態。

## 事件統計資料摘要

第 25 頁的表 5 彙總此範例的事件統計資料。

表 5: 範例 1 的事件統計資料摘要		
屬性	事件 1	事件 2
事件時間	T (G1)	T (G2)
事件的類型	高	確定
TimeSinceReset	T (G1)-T (0)	T (G2)-T (G1)
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

第 24 頁的圖 4 的中間部分顯示相較於該佇列的服務間隔，服務計時器所測量的經歷時間。若要查看是否可能發生佇列服務間隔事件，請比較代表服務計時器 (含箭頭) 的水平線與代表服務間隔的水平線的長度。如果服務計時器線路較長，且已啟用「佇列服務間隔高」事件，則下次取得時會發生「佇列服務間隔高」事件。如果計時器線路較短，且已啟用「佇列服務間隔正常」事件，則下次取得時會發生「佇列服務間隔正常」事件。

佇列服務間隔事件: 範例 2

一系列 MQPUT 呼叫和 MQGET 呼叫，其中佇列深度不一定是 1 或零。

此範例也顯示重設計時器而不產生事件的實例，例如，在時間 P2。

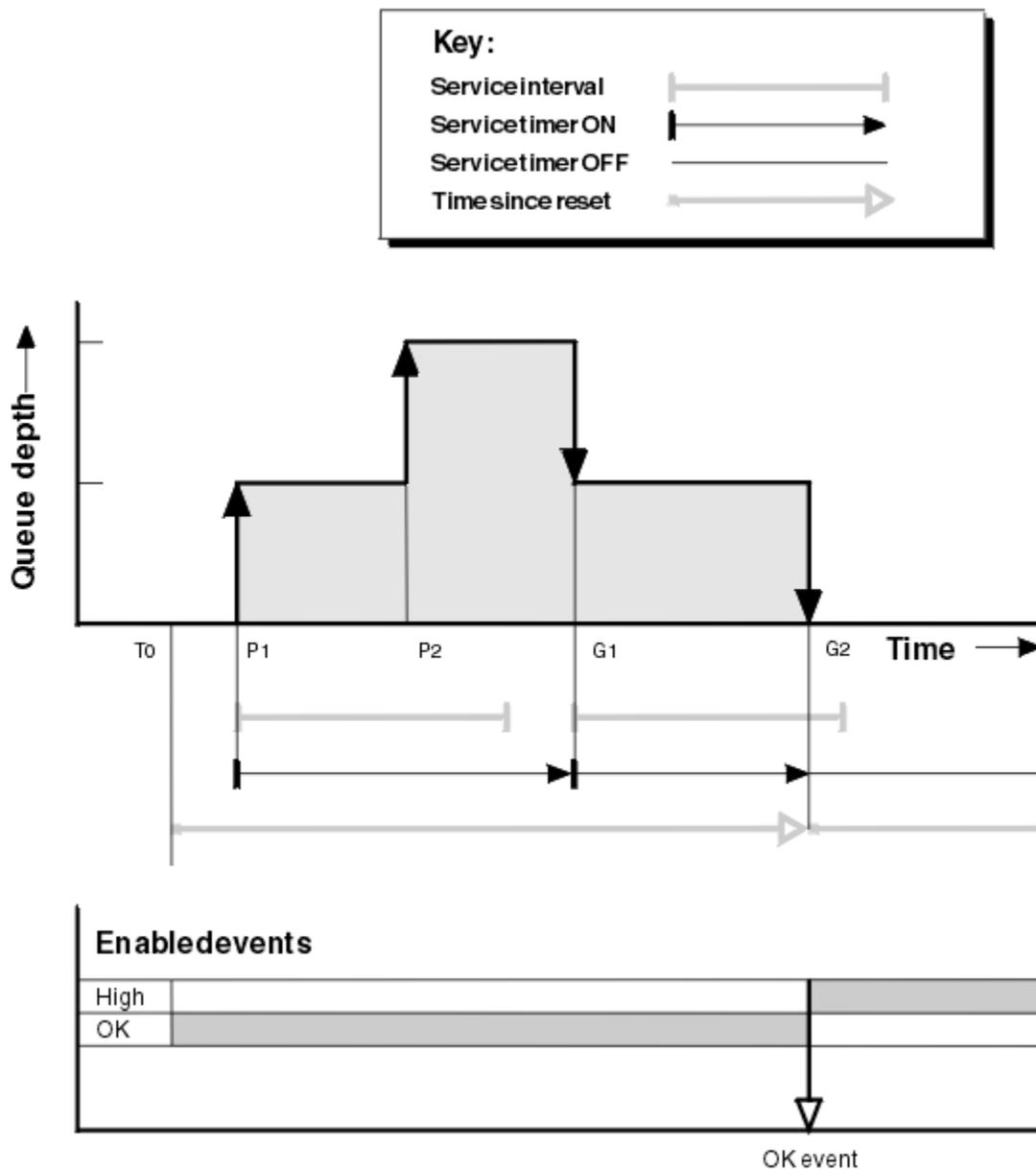


圖 5: 佇列服務間隔事件-範例 2

### 註解

在此範例中，一開始會啟用 OK 事件，並在 T0 時重設佇列統計資料。

1. 在 P1，第一個放置會啟動服務計時器。
2. 在 P2，第二個放置不會產生事件，因為放置無法導致 OK 事件。
3. 在 G1 時，現在已超出服務間隔，因此不會產生 OK 事件。不過，MQGET 呼叫會導致重設服務計時器。
4. 在 G2，在服務間隔內發生第二次取得，此時會產生 OK 事件。佇列管理程式會重設事件控制屬性，以便：
  - a. 會自動啟用高事件。
  - b. 「確定」事件已停用。

因為佇列現在是空的，所以服務計時器會切換至 OFF 狀態。

### 事件統計資料摘要

第 27 頁的表 6 彙總此範例的事件統計資料。

表 6: 範例 2 的事件統計資料摘要	
屬性	事件 2
事件時間	T (G2)
事件的類型	確定
TimeSinceReset	T (G2)-T (0)
HighQDepth	2
MsgEnqCount	2
MsgDeqCount	2

佇列服務間隔事件: 範例 3

MQGET 呼叫和 MQPUT 呼叫的順序, 比先前範例更零星。

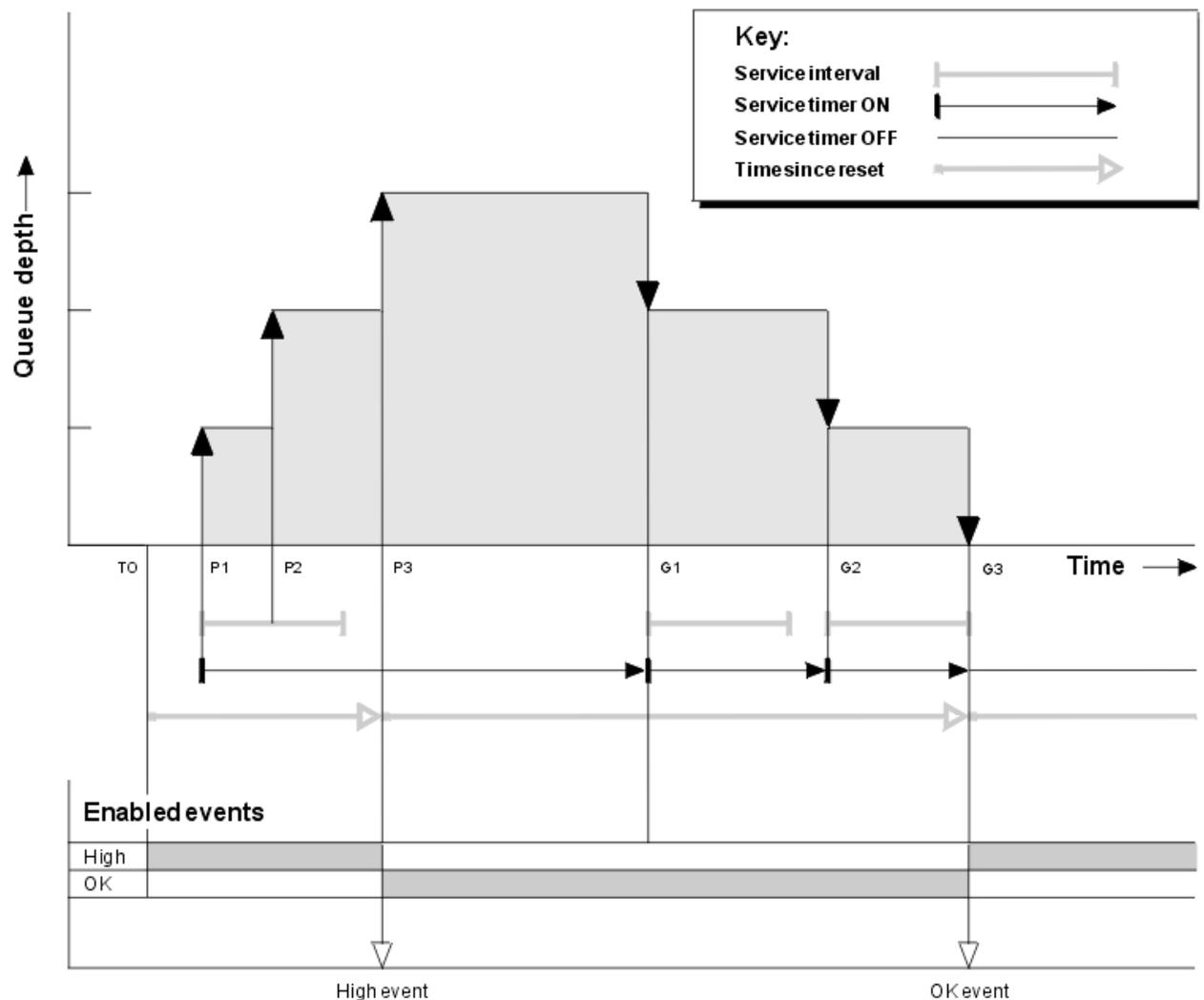


圖 6: 佇列服務間隔事件-範例 3

### 註解

1. 在時間 T (0), 會重設佇列統計資料, 並啟用「佇列服務間隔高」事件。
2. 在 P1, 第一個放置會啟動服務計時器。

3. 在 P2，第二個放置會將佇列深度增加至 2。這裡未產生高事件，因為未超出服務間隔時間。
4. 在 P3，第三個放置會導致產生高事件。(計時器已超出服務間隔。) 因為放置之前佇列深度不是零，所以未重設計時器。不過，會啟用「確定」事件。
5. 在 G1，MQGET 呼叫不會產生事件，因為已超出服務間隔且已啟用 OK 事件。不過，MQGET 呼叫會重設服務計時器。
6. 在 G2，MQGET 呼叫不會產生事件，因為已超出服務間隔且已啟用「確定」事件。同樣地，MQGET 呼叫會重設服務計時器。
7. 在 G3，第三個取得會清空佇列，且服務計時器等於服務間隔。因此會產生 OK 事件。重設服務計時器並啟用高事件。MQGET 呼叫會清空佇列，這會將計時器置於 OFF 狀態。

## 事件統計資料摘要

第 28 頁的表 7 彙總此範例的事件統計資料。

表 7: 範例 3 的事件統計資料摘要		
屬性	事件 1	事件 2
事件時間	T (P3)	T (G3)
事件的類型	高	確定
TimeSinceReset	T (P3)-T (0)	T (G3)-T (P3)
HighQDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

## 佇列深度事件

佇列深度事件與佇列深度相關，即佇列上的訊息數。

在 IBM MQ 應用程式中，佇列不得變滿。如果它們這麼做，則應用程式無法再將訊息放置在它們指定的佇列上。雖然如果發生此情況，訊息不會遺失，但佇列已滿可能會造成相當大的不便。如果將訊息放入佇列的速度比處理訊息的應用程式將訊息移除的速度更快，則可以在佇列上累積的訊息數。

此問題的解決方案視特定情況而定，但可能涉及：

- 將部分訊息轉移至另一個佇列。
- 正在啟動新的應用程式，以從佇列中移除更多訊息。
- 正在停止非必要的訊息資料流量。
- 增加佇列深度以克服暫時性上限。

提前警告問題可能即將發生，讓您更容易採取預防性動作。基於此目的，IBM MQ 提供下列佇列深度事件：

### 佇列深度高事件

指出佇列深度已增加至預先定義的臨限值，稱為「佇列深度高限制」。

### 佇列深度低事件

指出佇列深度已減少至預先定義的臨限值，稱為「佇列深度低限制」。

### 佇列已滿事件

指出佇列已達到其深度上限，即佇列已滿。

當應用程式嘗試將訊息放置在已達到其深度上限的佇列上時，會產生「佇列已滿事件」。「佇列深度高」事件會提供進階警告，指出佇列正在填滿。這表示在收到此事件之後，系統管理者需要採取一些預防性動作。您可以配置佇列管理程式，以便在預防性動作成功且佇列深度降至安全層次時，佇列管理程式會產生「佇列深度低」事件。

第一個佇列深度事件範例說明假設動作防止佇列變滿的效果。

## 相關概念

第 31 頁的『佇列深度事件範例』

使用這些範例來瞭解您可以從佇列深度事件取得的資訊

## 相關參考

[佇列已滿](#)

[佇列深度偏高](#)

[佇列深度低](#)

## 啟用佇列深度事件

若要為任何佇列深度事件配置佇列，請設定適當的佇列管理程式及佇列屬性。

## 關於這項作業

依預設，會停用所有佇列深度事件。啟用時，會產生佇列深度事件，如下所示：

- 將訊息放置在佇列上時，會產生「佇列深度高」事件，導致佇列深度大於或等於為 **QDepthHighLimit** 設定的值。
  - 「佇列深度高」事件由相同佇列上的「佇列深度低」事件自動啟用。
  - 「佇列深度高」事件會自動啟用相同佇列上的「佇列深度低」及「佇列已滿」事件。
- **Multi** GET 作業從佇列中移除訊息時，會產生「佇列深度低值」事件，導致佇列深度小於或等於 **QDepthLowLimit** 設定的值。
  - 「佇列深度低」事件由相同佇列上的「佇列深度高」事件或「佇列已滿」事件自動啟用。
  - 「佇列深度低」事件會自動啟用相同佇列上的「佇列深度高」及「佇列已滿」事件。
- **z/OS** 當訊息由 GET 作業從佇列中移除，或已移除但已過期，導致佇列深度小於或等於為 **QDepthLowLimit** 設定的值時，會產生「佇列深度低值」事件。
  - 「佇列深度低」事件由相同佇列上的「佇列深度高」事件或「佇列已滿」事件自動啟用。
  - 「佇列深度低」事件會自動啟用相同佇列上的「佇列深度高」及「佇列已滿」事件。
- 當應用程式因為佇列已滿而無法將訊息放入佇列時，會產生「佇列已滿」事件。
  - 「佇列已滿」事件由相同佇列上的「佇列深度高」或「佇列深度低」事件自動啟用。
  - 「佇列已滿」事件會自動啟用相同佇列上的「佇列深度低」事件。

請執行下列步驟，為任何佇列深度事件配置佇列：

## 程序

1. 使用佇列管理程式屬性 **PERFMEV**，在佇列管理程式上啟用效能事件。  
事件會移至 **SYSTEM.ADMIN.PERFM.EVENT** 佇列。
2. 設定下列其中一個屬性，以在必要佇列上啟用事件：
  - **QDepthHigh** 事件 (MQSC 中的 **QDPHIEV**)
  - **QDepthLow** 事件 (MQSC 中的 **QDPLOEV**)
  - **QDepthMax** 事件 (MQSC 中的 **QDPMAXEV**)
3. 選擇性的：若要設定限制，請以佇列深度上限的百分比來指派下列屬性：
  - **QDepthHigh** 限制 (MQSC 中的 **QDEPHTHI**)
  - **QDepthLow** 限制 (MQSC 中的 **QDEPHTLO**)

**限制：QDEPHTHI 不得小於 QDEPHTLO。**

如果 **QDEPHTHI** 等於 **QDEPHTLO**，則每次佇列深度以任一方向傳遞值時都會產生事件訊息，因為當佇列深度低於該值時會啟用高臨界值，而當深度高於該值時會啟用低臨界值。

## 結果

註：

**Multi** 當 GET 作業從佇列中移除過期訊息，導致佇列深度小於或等於為 **QDepthLowLimit** 設定的值時，不會產生「佇列深度低值」事件。只有在順利完成 GET 作業期間，IBM MQ 才會產生佇列深度低事件

訊息。因此，當從佇列中移除過期訊息時，不會產生佇列深度低事件訊息。此外，從佇列中移除這些過期訊息之後，不會重設 *QDepthHigh* 事件和 *QDepthLow* 事件。

**z/OS** IBM MQ 會在成功的破壞性 GET 作業期間產生佇列深度低事件訊息，或在符合訊息未過期時成功的破壞性 GET 作業期間產生低事件訊息。否則，在一般背景處理期間從佇列中移除過期訊息時，不會產生佇列深度低事件訊息。此外，在一般背景處理期間從佇列中移除過期訊息之後，不會重設 *QDepthHigh* 事件和 *QDepthLow* 事件。如需過期訊息處理的相關資訊，請參閱在 [IBM MQ for z/OS](#) 上調整佇列管理程式。

## 範例

若要在限制設為 80% 的佇列 MYQUEUE 上啟用「佇列深度高」事件，請使用下列 MQSC 指令：

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

若要啟用佇列 MYQUEUE 上限制設為 20% 的「佇列深度低值」事件，請使用下列 MQSC 指令：

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHLO(20) QDPLOEV(ENABLED)
```

若要在佇列 MYQUEUE 上啟用「佇列已滿」事件，請使用下列 MQSC 指令：

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDPMAXEV(ENABLED)
```

## z/OS z/OS 上的共用佇列及佇列深度事件

在 IBM MQ for z/OS 上，對於使用共用佇列的應用程式而言，如果佇列共用群組中的所有佇列管理程式都具有 PERFMEV 屬性的相同設定，則事件監視會更直接明確。

當共用佇列上發生佇列深度事件，且佇列管理程式屬性 **PERFMEV** 設為 **ENABLED** 時，佇列共用群組中的佇列管理程式會產生事件訊息。如果部分佇列管理程式上的 **PERFMEV** 設為 **DISABLED**，則那些佇列管理程式不會產生事件訊息，使得從應用程式進行事件監視更加困難。如需更直接明確的監視，請為每一個佇列管理程式提供 **PERFMEV** 屬性的相同設定。

每一個佇列管理程式所產生的這則事件訊息代表其共用佇列的個別用法。如果佇列管理程式在共用佇列上不執行任何活動，則事件訊息中的各種值都是空值或零。您可以使用空值事件訊息，如下所示：

- 請確定佇列共用群組中的每一個作用中佇列管理程式都會產生一則事件訊息
- 強調顯示產生事件訊息之佇列管理程式的共用佇列上無活動的案例

## 協調佇列管理程式

當佇列管理程式發出佇列深度事件時，它會更新共用佇列物件定義，以切換作用中效能事件屬性。例如，視佇列屬性的定義而定，「佇列深度高」事件會啟用「佇列深度低」及「佇列已滿」事件。順利更新共用佇列物件之後，偵測到效能事件的佇列管理程式最初會變成協調佇列管理程式。

如果針對效能事件啟用，協調佇列管理程式會執行下列動作：

1. 發出事件訊息，以擷取自前次建立事件訊息或自前次重設佇列統計資料以來所收集的所有共用佇列效能資料。此訊息的訊息描述子 (MQMD) 包含協調佇列管理程式所建立的唯一相關性 ID (*CorrelId*)。
2. 播送至相同佇列共用群組中的所有其他作用中佇列管理程式，以要求產生共用佇列的事件訊息。廣播包含由協調佇列管理程式針對事件訊息集所建立的相關性 ID。

收到來自協調佇列管理程式的要求之後，如果佇列共用群組中有啟用效能事件的作用中佇列管理程式，則該作用中佇列管理程式會針對共用佇列發出事件訊息。發出的事件訊息包含自前次建立事件訊息以來，或自前次重設統計資料以來，接收(作用中)佇列管理程式所執行之所有作業的相關資訊。此事件訊息的訊息描述子 (MQMD) 包含協調佇列管理程式指定的唯一相關性 ID (*CorrelId*)。

當共用佇列上發生效能事件時，會產生  $n$  個事件訊息，其中  $n$  是從 1 到佇列共用群組中作用中佇列管理程式的數目。每一個事件訊息都包含與產生事件訊息之佇列管理程式的共用佇列活動相關的資料。

## 共用與非共用佇列之間的差異

在共用佇列上啟用佇列深度事件不同於在非共用佇列上啟用它們。主要差異是即使在佇列管理程式上 **PERFMEV** 為 **DISABLED**，也會針對共用佇列啟用事件。這不是非共用佇列的情況。

請考量下列範例，其說明此差異：

- QM1 是一個佇列管理程式，其 *PerformanceEvent* (MQSC 中的 **PERFMEV**) 設為 **DISABLED**。
- SQ1 是 MQSC 中 **QSGDISP** 設為 (SHARED) QLOCAL 的共用佇列。
- LQ1 是在 MQSC 中 **QSGDISP** 設為 (QMGR) QLOCAL 的非共用佇列。

這兩個佇列在其定義上都設定了下列屬性：

- **QDPHIEV (ENABLED)**
- **QDPLOEV (DISABLED)**
- **QDPMAXEV (DISABLED)**

如果訊息同時放置在兩個佇列上，因此深度符合或超出 **QDEPTHHI** 臨界值，則 SQ1 上的 **QDPHIEV** 值會切換至 **DISABLED**。此外，**QDPLOEV** 和 **QDPMAXEV** 會切換至 **ENABLED**。符合事件準則時，會針對每一個效能事件自動切換 SQ1 的屬性。

相反地，LQ1 的屬性維持不變，直到佇列管理程式上的 **PERFMEV** 為 **ENABLED** 為止。例如，這表示如果佇列管理程式的 **PERFMEV** 屬性是 **ENABLED**，已停用然後再次設為 **ENABLED**，則共用佇列上的效能事件設定可能與非共用佇列的效能事件設定不一致，即使它們最初可能相同。

## 佇列深度事件範例

使用這些範例來瞭解您可以從佇列深度事件取得的資訊

第一個範例提供佇列深度事件的基本圖解。第二個範例更廣泛，但原則與第一個範例相同。這兩個範例都使用相同的佇列定義，如下所示：

佇列 MYQUEUE1 的深度上限為 1000 則訊息。高佇列深度限制為 80%，低佇列深度限制為 20%。一開始，會啟用「佇列深度高」事件，但會停用其他佇列深度事件。

用來配置此佇列的 IBM MQ 指令 (MQSC) 如下：

```
ALTER QMGR PERFMEV(ENABLED)

DEFINE QLOCAL('MYQUEUE1') MAXDEPTH(1000) QDPMAXEV(DISABLED) QDEPTHHI(80)
QDPHIEV(ENABLED) QDEPTHLO(20) QDPLOEV(DISABLED)
```

### 相關概念

[第 28 頁的『佇列深度事件』](#)

佇列深度事件與佇列深度相關，即佇列上的訊息數。

### 相關工作

[第 29 頁的『啟用佇列深度事件』](#)

若要為任何佇列深度事件配置佇列，請設定適當的佇列管理程式及佇列屬性。

### 相關參考

[MQSC 指令](#)

佇列深度事件: 範例 1

佇列深度事件的基本順序。

[第 32 頁的圖 7](#) 顯示隨時間變化的佇列深度。

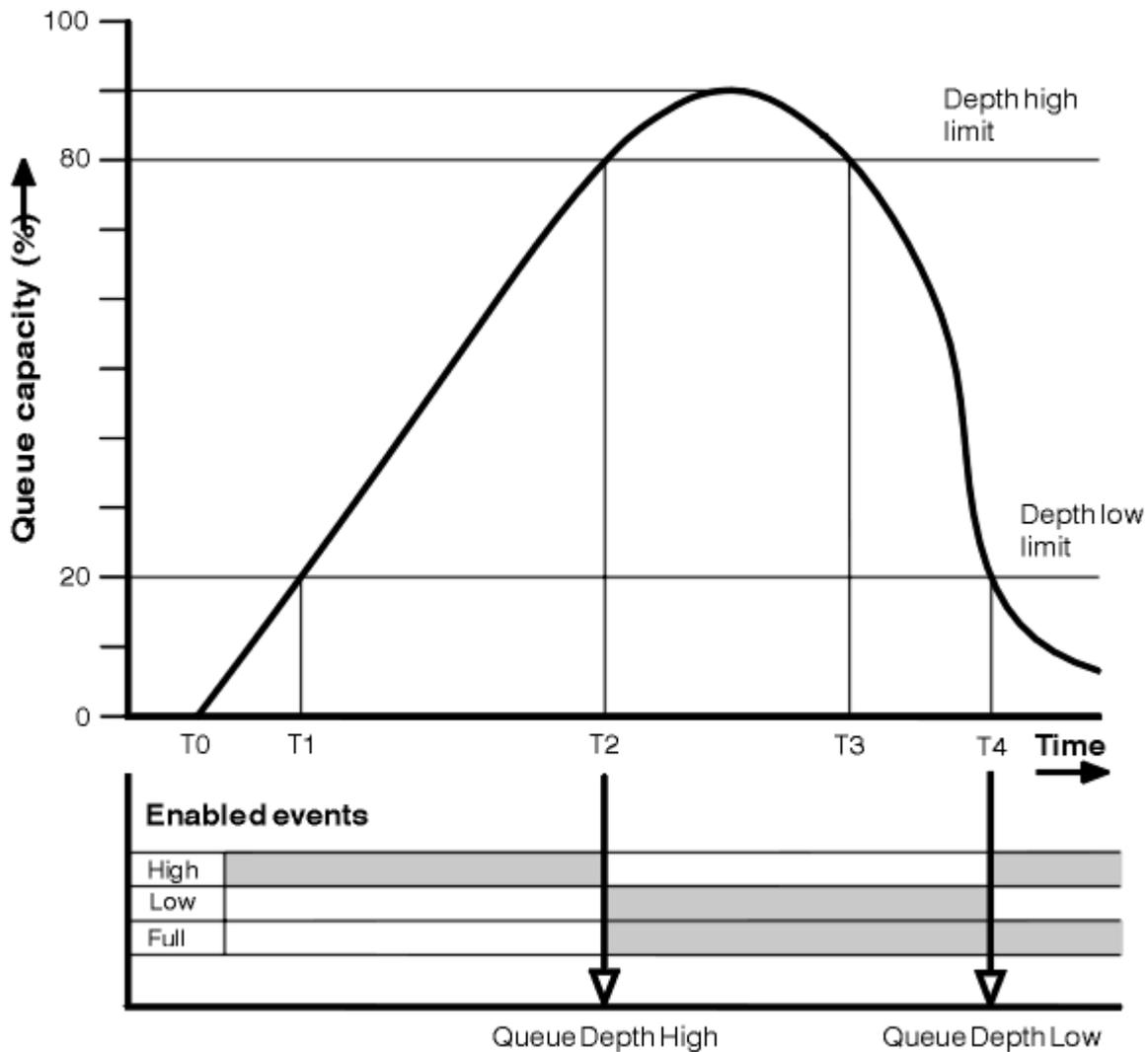


圖 7: 佇列深度事件 (1)

### 註解

1. 在 T (1)，佇列深度正在增加 (MQPUT 呼叫數比 MQGET 呼叫數多)，且超過「佇列深度低」限制。目前未產生任何事件。
2. 當達到深度高限制 (80%) 且產生「佇列深度高」事件時，佇列深度會繼續增加，直到 T (2) 為止。  
這會啟用「佇列已滿」及「佇列深度低值」事件。
3. 事件所引起的 (假設) 預防性動作會防止佇列變滿。依時間 T (3)，已再次達到「佇列深度高限制」，從上述時間開始。目前未產生任何事件。
4. 當佇列深度達到深度低限制 (20%) 且產生「佇列深度低」事件時，佇列深度會持續下降，直到 T (4) 為止。  
這會同時啟用「佇列已滿」及「佇列深度高」事件。

### 事件統計資料摘要

第 33 頁的表 8 彙總佇列事件統計資料，並 第 33 頁的表 9 彙總已啟用哪些事件。

表 8: 佇列深度事件的事件統計資料摘要 (範例 1)		
事件統計資料	事件 2	事件 4
事件時間	T (2)	T (4)
事件的類型	佇列深度偏高	佇列深度低
TimeSinceReset	T (2)-T (0)	T (4)-T (2)
HighQDepth (重設後的佇列深度上限)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

表 9: 顯示已啟用哪些事件的摘要			
時段	佇列深度高事件	佇列深度低事件	佇列已滿事件
在 T (1) 之前	ENABLED	-	-
T (1) 至 T (2)	ENABLED	-	-
T (2) 至 T (3)	-	ENABLED	ENABLED
T (3) 至 T (4)	-	ENABLED	ENABLED
T (4) 之後	ENABLED	-	ENABLED

佇列深度事件: 範例 2  
更廣泛的佇列深度事件序列。

第 34 頁的圖 8 顯示隨時間變化的佇列深度。

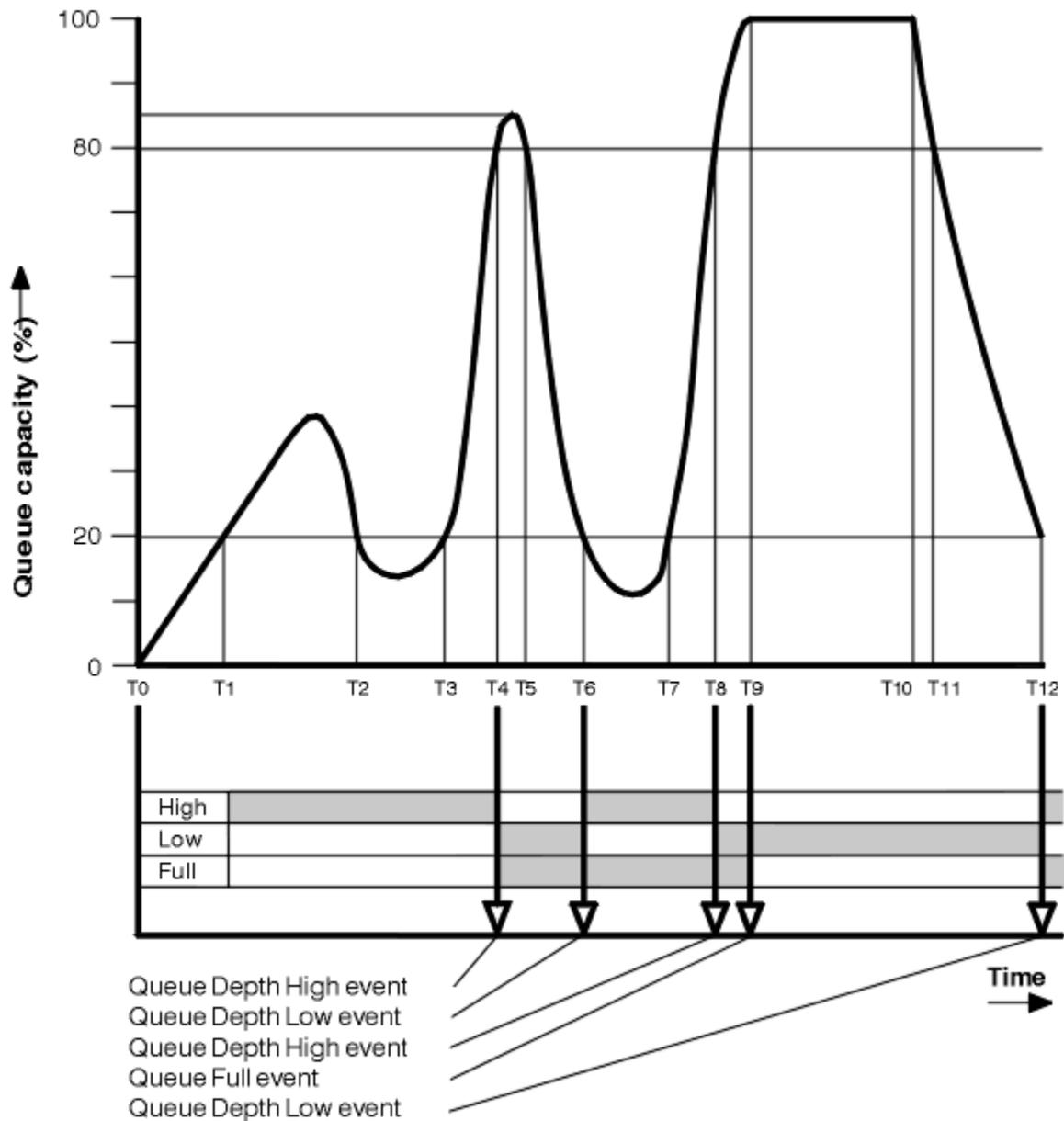


圖 8: 佇列深度事件 (2)

### 註解

1. 在下列時間不會產生「佇列深度低值」事件:
  - T (1) (佇列深度增加, 且未啟用)
  - T (2) (未啟用)
  - T (3) (佇列深度增加, 但未啟用)
2. 在 T (4) 時, 會發生「佇列深度高」事件。這會啟用「佇列已滿」及「佇列深度低值」事件。
3. 在 T (9), 因為佇列已滿而無法放置在佇列上的第一個訊息 **之後** 會發生「佇列已滿」事件。
4. 在 T (12) 時, 會發生「佇列深度低」事件。

### 事件統計資料摘要

第 35 頁的表 10 彙總佇列事件統計資料, 並 第 35 頁的表 11 彙總在此範例的不同時間啟用的事件。

事件統計資料	事件 4	事件 6	事件 8	事件 9	事件 12
事件時間	T (4)	T (6)	T (8)	T (9)	T (12)
事件的類型	佇列深度偏高	佇列深度低	佇列深度偏高	佇列已滿	佇列深度低
TimeSinceReset	T (4)-T (0)	T (6)-T (4)	T (8)-T (6)	T (9)-T (8)	T (12)-T (9)
HighQDepth	800	855	800	1000	1000
MsgEnqCount	1645	311	1377	324	221
MsgDeqCount	845	911	777	124	1021

時段	佇列深度高事件	佇列深度低事件	佇列已滿事件
T (0) 至 T (4)	ENABLED	-	-
T (4) 至 T (6)	-	ENABLED	ENABLED
T (6) 至 T (8)	ENABLED	-	ENABLED
T (8) 至 T (9)	-	ENABLED	ENABLED
T (9) 至 T (12)	-	ENABLED	-
T (12) 之後	ENABLED	-	ENABLED

註: 事件超出同步點。因此, 您可能有空的佇列, 然後填滿它會導致事件, 然後在同步點管理程式的控制下回復所有訊息。不過, 已自動設定事件啟用, 因此下次佇列填滿時, 不會產生任何事件。

## 配置事件

配置事件是在建立、變更或刪除物件時產生的通知, 也可以由明確要求產生。

如需如何開啟事件的相關資訊, 請參閱 [第 16 頁的『控制配置、指令及日誌程式事件』](#)。

配置事件會通知您物件屬性的變更。配置事件有四種類型:

- 建立物件事件
- 變更物件事件
- 刪除物件事件
- 重新整理物件事件

事件資料包含下列資訊:

### 原始資訊

包含進行變更的來源佇列管理程式、進行變更的使用者 ID, 以及進行變更的方式, 例如透過主控台指令。

### 環境定義資訊

來自指令訊息之訊息資料中的環境定義資訊抄本。

只有在輸入指令作為 SYSTEM.COMMAND.INPUT 佇列。

### 物件身分

包含物件的名稱、類型及處置。

### 物件屬性

包含物件中所有屬性的值。

事件是 PCF 格式的訊息。如需相關資訊, 請參閱:

- [變更物件](#)

- [建立物件](#)
- [刪除物件](#)
- [重新整理物件](#)

如果是變更物件事件，則會產生 before 訊息和 after 訊息。它們具有下列共同欄位，因此您可以識別相符訊息：

- 相同的相關性 ID
- 在 PCF 標頭中：
  - before 訊息標頭具有 `MsgSeqNumber': 1, 'Control': 'NOT_LAST'`
  - After 訊息標頭具有 `MsgSeqNumber': 2, 'Control': 'LAST'`

每一個產生的配置事件訊息都會放在佇列 SYSTEM.ADMIN.CONFIG.EVENT。

### 相關概念

[第 12 頁的『配置事件』](#)

當明確要求配置事件時，會產生配置事件；當建立、修改或刪除物件時，會自動產生配置事件。

### 相關參考

[建立物件](#)

[變更物件](#)

[刪除物件](#)

[重新整理物件](#)

[第 8 頁的『事件類型』](#)

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

### 配置事件產生

請利用這個頁面來檢視導致產生配置事件的指令，以及瞭解未產生配置事件的情況

當 CONFIGEV 佇列管理程式屬性為 ENABLED 且

- 會發出下列任何指令或其 PCF 對等項目：
  - DELETE AUTHINFO
  - DELETE CFSTRUCT
  - 刪除通道
  - 刪除名單
  - 刪除處理程序
  - DELETE QMODEL/QALIAS/QREMOTE
  - DELETE STGCLASS
  - 刪除主題
  - 重新整理佇列管理程式
- 即使物件沒有變更，也會發出下列任何指令或其 PCF 對等項目：
  - DEFINE/ALTER AUTHINFO
  - DEFINE/ALTER CFSTRUCT
  - 定義/變更通道
  - DEFINE/ALTER NAMELIST
  - DEFINE/ALTER PROCESS
  - DEFINE/ALTER QMODEL/QALIAS/QREMOTE
  - DEFINE/ALTER STGCLASS
  - DEFINE/ALTER TOPIC
  - DEFINE MAXSMGS

- SET CHLAUTH
- ALTER QMGR, 除非 CONFIGEV 屬性已停用且未變更為 ENABLED
- 對非暫時動態的本端佇列發出下列任何指令或其 PCF 對等項目, 即使佇列沒有變更也一樣。
  - DELETE QLOCAL
  - DEFINE/ALTER QLOCAL
- 即使物件沒有變更, 也會發出 MQSET 呼叫, 但暫時動態佇列除外。

## 未產生配置事件時

在下列情況下, 不會產生配置事件訊息:

- 當指令或 MQSET 呼叫失敗時
- 當佇列管理程式在嘗試將配置事件放入事件佇列時發生錯誤, 在此情況下, 指令或 MQSET 呼叫會完成, 但不會產生事件訊息
- 若為暫時動態佇列
- 對 TRIGGER 佇列屬性進行內部變更時
- 若為配置事件佇列 SYSTEM.ADMIN.CONFIG.EVENT, 但 REFRESH QMGR 指令除外
- 對於會導致叢集作業變更的 REFRESH/RESET CLUSTER 和 RESUME/SUSPEND QMGR 指令
- 建立或刪除佇列管理程式時

## 相關概念

[第 35 頁的『配置事件』](#)

配置事件是在建立、變更或刪除物件時產生的通知, 也可以由明確要求產生。

## 相關參考

[MQSC 指令](#)

[可程式指令格式簡介](#)

[MQSET-設定物件屬性](#)

[MQSET-設定物件屬性](#)

## 配置事件使用情形

請利用這個頁面來檢視如何使用配置事件來取得系統的相關資訊, 以及瞭解可能影響您使用配置事件的因素 (例如 CMDSCOPE)。

您可以將配置事件用於下列目的:

1. 產生及維護中央配置儲存庫, 您可以從中產生報告, 以及產生系統結構的相關資訊。
2. 產生審核追蹤。例如, 如果物件非預期地變更, 則可以儲存關於進行變更的人員及完成時間的資訊。

這在同時啟用指令事件時特別有用。如果 MQSC 或 PCF 指令導致產生配置事件及指令事件, 則這兩則事件訊息會在其訊息描述子中共用相同的相關性 ID。

對於 MQSET 呼叫或下列任何指令:

- DEFINE 物件
- ALTER 物件
- 刪除物件

如果佇列管理程式屬性 CONFIGEV 已啟用, 但配置事件訊息無法放置在配置事件佇列上 (例如, 尚未定義事件佇列), 則不論如何都會執行指令或 MQSET 呼叫。

## CMDSCOPE 的影響

對於使用 CMDSCOPE 的指令, 將在執行指令的佇列管理程式或佇列管理程式上產生配置事件訊息, 而不是在輸入指令的位置。不過, 事件資料中的所有原始及環境定義資訊將與輸入的原始指令相關, 即使使用 CMDSCOPE 的指令是來源佇列管理程式所產生的指令也是如此。

如果佇列共用群組包含非現行版本的佇列管理程式，則會針對透過 CMDSCOPE 在現行版本的佇列管理程式上執行的任何指令產生事件，但不會針對舊版的佇列管理程式產生事件。即使輸入指令的佇列管理程式是舊版，也會發生這種情況，雖然在這種情況下事件資料中不會包含任何環境定義資訊。

## 相關概念

可程式指令格式簡介

[第 35 頁的『配置事件』](#)

配置事件是在建立、變更或刪除物件時產生的通知，也可以由明確要求產生。

## 相關參考

[MQSET-設定物件屬性](#)

[MQSET-設定物件屬性](#)

## 重新整理物件配置事件

「重新整理物件」配置事件與其他配置事件不同，因為只有在明確要求時才會發生。

建立、變更及刪除事件是由 MQSET 呼叫或用來變更物件的指令所產生，但只有在 MQSC 指令、REFRESH QMGR 或其 PCF 對等項目明確要求時，才會發生重新整理物件事件。

REFRESH QMGR 指令不同於所有其他產生配置事件的指令。所有其他指令都適用於特定物件，並為該物件產生單一配置事件。REFRESH QMGR 指令可以產生許多配置事件訊息，可能代表佇列管理程式所儲存的每個物件定義。每一個選取的物件都會產生一則事件訊息。

REFRESH QMGR 指令使用三個選取準則的組合來過濾涉及的物件數：

- 物件名稱
- 物件類型
- 重新整理間隔

如果您在 REFRESH QMGR 指令上未指定任何選取準則，則會針對每一個選取準則使用預設值，並針對佇列管理程式所儲存的每一個物件定義產生重新整理配置事件訊息。這可能會導致無法接受的處理時間及事件訊息產生。請考慮指定一些選取準則。

在下列情況下，可以使用產生重新整理事件的 REFRESH QMGR 指令：

- 當需要系統中所有或部分物件的配置資料時，不論物件最近是否已操作 (例如，第一次啟用配置事件時)。請考慮使用數個指令，每一個指令都有不同的物件選項，但全部都會包含在內。
- 如果 SYSTEM.ADMIN.CONFIG.EVENT 佇列。在此情況下，不會針對「建立」、「變更」或「刪除」事件產生任何配置事件訊息。當已更正佇列上的錯誤時，可使用「重新整理佇列管理程式」指令來要求產生事件訊息，但在佇列中發生錯誤時遺失這些訊息。在此狀況下，請考量將重新整理間隔設為佇列無法使用的時間。

## 相關概念

[第 35 頁的『配置事件』](#)

配置事件是在建立、變更或刪除物件時產生的通知，也可以由明確要求產生。

## 相關參考

[重新整理佇列管理程式](#)

[重新整理佇列管理程式](#)

## 指令事件

指令事件是 MQSC 或 PCF 指令已順利執行的通知。

事件資料包含下列資訊：

### 原始資訊

包含從中發出指令的佇列管理程式、發出指令的使用者 ID，以及如何發出指令，例如由主控台指令。

### 環境定義資訊

來自指令訊息之訊息資料中的環境定義資訊抄本。如果未使用訊息輸入指令，則會省略環境定義資訊。

只有在輸入指令作為 SYSTEM.COMMAND.INPUT 佇列。

## 指令資訊

發出的指令類型。

## 指令資料

- 對於 PCF 指令，指指令資料的抄本
- 對於 MQSC 指令，指令文字

指令資料格式不一定符合原始指令的格式。例如，在 [多平台](#) 上，指令資料格式一律為 PCF 格式，即使原始要求是 MQSC 指令也一樣。

所產生的每一則指令事件訊息都會放在指令事件佇列 SYSTEM.ADMIN.COMMAND.EVENT。

## 相關參考

[指令](#)

[第 8 頁的『事件類型』](#)

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

## 指令事件產生

請利用這個頁面來檢視導致產生指令事件的狀況，以及瞭解未產生指令事件的情況

在下列狀況下會產生指令事件訊息：

- 當 CMDEV 佇列管理程式屬性指定為 ENABLED 且 MQSC 或 PCF 指令順利執行時。
- 當 CMDEV 佇列管理程式屬性指定為 NODISPLAY 時，任何指令都會順利執行，但 DISPLAY 指令 (MQSC) 及 Inquire 指令 (PCF) 除外。
- 當您執行 MQSC 指令、ALTER QMGR 或 PCF 指令、「變更佇列管理程式」及 CMDEV 佇列管理程式屬性時，符合下列任一條件：
  - 變更之後，CMDEV 未指定為 DISABLED
  - 在變更之前，CMDEV 未指定為 DISABLED

如果針對指令事件佇列 SYSTEM.ADMIN.COMMAND.EVENT: 如果佇列仍存在且未禁止放置，則會產生指令事件。

## 未產生指令事件時

在下列情況下，不會產生指令事件訊息：

- 當指令失敗時
- 當佇列管理程式在嘗試將指令事件放入事件佇列時發生錯誤，在這種情況下，指令會執行，但不會產生任何事件訊息
- 對於 MQSC 指令 REFRESH QMGR TYPE (早期)
- 對於 MQSC 指令 START QMGR MQSC
- 若為 MQSC 指令 SUSPEND QMGR，如果指定參數 LOG 的話
- 對於 MQSC 指令 RESUME QMGR，如果指定參數 LOG 的話

## 相關概念

[第 38 頁的『指令事件』](#)

指令事件是 MQSC 或 PCF 指令已順利執行的通知。

## 相關參考

[重新整理佇列管理程式](#)

[開始佇列管理程式](#)

[SUSPEND 佇列管理程式](#)

[回復佇列管理程式](#)

[SUSPEND QMGR、RESUME QMGR 及叢集](#)

## 指令事件用法

請利用這個頁面來檢視如何使用指令事件來產生已執行之指令的審核追蹤

例如，如果物件非預期地變更，則可以儲存關於進行變更的人員及完成時間的資訊。當也啟用配置事件時，這會特別有用。如果 MQSC 或 PCF 指令導致產生指令事件及配置事件，則這兩個事件訊息會在其訊息描述子中共用相同的相關性 ID。

如果已產生指令事件訊息，但無法放置在指令事件佇列上 (例如，如果尚未定義指令事件佇列)，則仍會執行針對其產生指令事件的指令。

## CMDSCOPE 的影響

對於使用 CMDSCOPE 的指令，將在執行指令的佇列管理程式或佇列管理程式上產生指令事件訊息，而不是在輸入指令的位置。不過，事件資料中的所有原始及環境定義資訊將與輸入的原始指令相關，即使使用 CMDSCOPE 的指令是來源佇列管理程式所產生的指令也是如此。

### 相關概念

第 38 頁的『指令事件』

指令事件是 MQSC 或 PCF 指令已順利執行的通知。

第 39 頁的『指令事件產生』

請利用這個頁面來檢視導致產生指令事件的狀況，以及瞭解未產生指令事件的情況

### 相關參考

[MQSC 指令](#)

[群組中的 PCF 指令及回應](#)

## Multi 日誌程式事件

日誌程式事件是指佇列管理程式已開始寫入新的日誌範圍 **IBM i** 或 IBM i 上的日誌接收器的通知。

**z/OS** IBM MQ for z/OS 無法使用日誌程式事件訊息。

事件資料包含下列資訊：

- 現行日誌範圍的名稱。
- 重新啟動回復所需的最早日誌範圍名稱。
- 媒體回復所需的最早日誌範圍名稱。
- 日誌範圍所在的目錄。
- **V9.1.0** 需要保存通知的最早日誌範圍名稱。

從 IBM MQ 9.1.0 開始，會針對保存日誌管理產生日誌程式事件，亦即，當 **ARCHLOG** 變更時，且 **ARCHLOG** 值會包含在日誌程式事件訊息中。

如需所有這些參數的相關資訊，請參閱 [DISPLAY QMSTATUS](#)。

您可以使用 **CURRLOG** 和 **ARCHLOG** 值來決定應該保存的項目。每當 **CURRLOG** 變更時，可以傳送數字小於 **CURRLOG** 的檔案來進行保存，一旦完成延伸範圍的保存，您應該呼叫 [SET LOG](#) 來通知佇列管理程式。

**ARCHLOG** 是需要保存的最舊延伸範圍。當您已保存該延伸範圍並呼叫 [SET LOG](#) 來通知佇列管理程式已保存它時，佇列管理程式會依序將 **ARCHLOG** 移至下一個延伸範圍。佇列管理程式會排定要刪除或重複使用早於 **ARCHLOG** 的延伸範圍。

在針對此延伸範圍呼叫 **SET LOG** 之後，即會發出新事件，因為 **ARCHLOG** 將會變更，而且您需要知道在新事件之後需要保存哪些項目。

如果您的保存程序因某些原因而失敗，且發生大量的通知建置，則您的管理者可以手動發出指令 **RESET QMGR TYPE (ARCHLOG)**。這會通知佇列管理程式，它可以重複使用或刪除所有早於 (包括) 您指定的延伸範圍的延伸範圍。

每一個產生的日誌程式事件訊息都會放置在日誌程式事件佇列 **SYSTEM.ADMIN.LOGGER.EVENT**。

## 相關參考

日誌程式

第 8 頁的『事件類型』

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

### Multi 日誌程式事件產生

請利用這個頁面來檢視導致產生日誌程式事件的狀況，以及瞭解未產生日誌程式事件的情況

在下列狀況中會產生日誌程式事件訊息：

- 當 LOGGEREV 佇列管理程式屬性指定為 ENABLED 且佇列管理程式開始寫入新的日誌範圍或在 IBM i 上寫入日誌接收器時。
- 當 LOGGEREV 佇列管理程式屬性指定為 ENABLED 且佇列管理程式啟動時。
- 當 LOGGEREV 佇列管理程式屬性從 DISABLED 變更為 ENABLED 時。
- **V9.1.0** 當通知 LOGGEREV 佇列管理程式屬性已保存其佇列管理程式正在等待保存通知 (ARCHLOG) 的最舊日誌範圍名稱時。

**提示:** 您可以使用 RESET QMGR MQSC 指令，要求佇列管理程式開始寫入新的日誌範圍。

### 未產生日誌程式事件時

在下列情況下，不會產生日誌程式事件訊息：

- 當佇列管理程式配置為使用循環式記載時。

在此情況下，LOGGEREV 佇列管理程式屬性會設為 DISABLED，且無法變更。

- 當佇列管理程式在嘗試將日誌程式事件放入事件佇列時發生錯誤，在此情況下會完成導致事件的動作，但不會產生事件訊息。

## 相關概念

第 40 頁的『日誌程式事件』

日誌程式事件是指佇列管理程式已開始寫入新的日誌範圍 **IBM i** 或 IBM i 上的日誌接收器的通知。

**z/OS** IBM MQ for z/OS 無法使用日誌程式事件訊息。

## 相關參考

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(10 位數帶正負號的整數\)](#)

[RESET QMGR](#)

### Multi 日誌程式事件使用情形

請利用這個頁面來檢視如何使用日誌程式事件來判斷佇列管理程式重新啟動或媒體回復不再需要的日誌範圍。

從作用中日誌目錄中移除日誌範圍之前，您可以將多餘的日誌範圍保存至媒體 (例如磁帶) 以進行災難回復。定期移除多餘的日誌延伸範圍會將磁碟空間用量保持在最小。

如果已啟用 LOGGEREV 佇列管理程式屬性，但日誌程式事件訊息無法放置在日誌程式事件佇列上 (例如，因為尚未定義事件佇列，導致事件繼續執行的動作)。

## 相關概念

第 40 頁的『日誌程式事件』

日誌程式事件是指佇列管理程式已開始寫入新的日誌範圍 **IBM i** 或 IBM i 上的日誌接收器的通知。

**z/OS** IBM MQ for z/OS 無法使用日誌程式事件訊息。

## 相關參考

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(10 位數帶正負號的整數\)](#)

請利用這個頁面來檢視導致產生日誌程式事件的狀況，以及瞭解未產生日誌程式事件的情況

### Multi 監視日誌程式事件佇列的範例 C 程式

請利用這個頁面來檢視範例 C 程式，它會監視日誌程式事件佇列中是否有新的事件訊息、讀取那些訊息，以及將訊息內容放到 stdout 中。

```

/*****
/*
/* Program name: AMQSL0G0.C
/*
/* Description: Sample C program to monitor the logger event queue and
/* display formatted message content to stdout when a logger
/* event occurs
/*
/* <copyright
/* notice="lm-source-program"
/* pids="5724-H72,"
/* years="2005, 2024"
/* crc="186943832" >
/* Licensed Materials - Property of IBM
/*
/* 5724-H72,
/*
/* (C) Copyright IBM Corp. 2005, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* </copyright>
/*****
/*
/* Function: AMQSL0G is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and displays the
/* formatted contents of the message to stdout.
/*****
/*
/* AMQSL0G has 1 parameter - the queue manager name (optional, if not
/* specified then the default queue manager is implied)
/*****

/*****
/* Includes
/*****
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include <cmqc.h> /* MQI constants*/
#include <cmqcfc.h> /* PCF constants*/

/*****
/* Constants
/*****

#define MAX_MESSAGE_LENGTH 8000

typedef struct _ParmTableEntry
{
    MQLONG ConstVal;
    PMQCHAR Desc;
} ParmTableEntry;

ParmTableEntry ParmTable[] =
{
    {0, ""},
    {MQCA_Q_MGR_NAME, "Queue Manager Name"},
    {MQCMD_LOGGER_EVENT, "Logger Event Command"},
    {MQRC_LOGGER_STATUS, "Logger Status"},
    {MQCACF_ARCHIVE_LOG_EXTENT_NAME, "Archive Log Extent"},
    {MQCACF_CURRENT_LOG_EXTENT_NAME, "Current Log Extent"},
    {MQCACF_RESTART_LOG_EXTENT_NAME, "Restart Log Extent"},
    {MQCACF_MEDIA_LOG_EXTENT_NAME, "Media Log Extent"},
    {MQCACF_LOG_PATH, "Log Path"}
};

```

```

#if defined(MQ_64_BIT)
#define Int32
#else
#define Int32 "l"
#endif

/*****
/* Function prototypes
*****/

static void ProcessPCF(MQHCONN hConn,
                      MQHOBJ hEventQueue,
                      PMQCHAR pBuffer);

static PMQCHAR ParmToString(MQLONG Parameter);

/*****
/* Function: main
*****/

int main(int argc, char * argv[])
{
    MQLONG CompCode;
    MQLONG Reason;
    MQHCONN hConn = MQHC_UNUSABLE_HCONN;
    MQOD ObjDesc = { MQOD_DEFAULT };
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1] = "";
    MQCHAR LogEvQ[MQ_Q_NAME_LENGTH] = "SYSTEM.ADMIN.LOGGER.EVENT";
    MQHOBJ hEventQueue = MQHO_UNUSABLE_HOBJ;
    PMQCHAR pBuffer = NULL;

    printf("\n/*****/\n");
    printf("/* Sample Logger Event Monitor start */\n");
    printf("/*****/\n");

    /*****
    /* Parse any command line options
    *****/
    if (argc > 1)
    {
        strncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    }

    pBuffer = (PMQCHAR)malloc(MAX_MESSAGE_LENGTH);
    if (pBuffer == NULL)
    {
        printf("Can't allocate %d bytes\n", MAX_MESSAGE_LENGTH);
        goto MOD_EXIT;
    }

    /*****
    /* Connect to the specified (or default) queue manager
    *****/
    MQCONN(QMName,
           &hConn,
           &CompCode,
           &Reason);

    if (Reason != MQRC_NONE)
    {
        printf("MQCONN ended with reason code %" Int32 "d\n", Reason);
        goto MOD_EXIT;
    }

    /*****
    /* Open the logger event queue for input
    *****/
    strncpy(ObjDesc.ObjectQMgrName, QMName, MQ_Q_MGR_NAME_LENGTH);
    strncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

    MQOPEN(hConn,
           &ObjDesc,
           MQOO_INPUT_EXCLUSIVE,
           &hEventQueue,
           &CompCode,
           &Reason);

    if (Reason != MQRC_NONE)
    {
        printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %" Int32 "d\n",
              ObjDesc.ObjectQMgrName,
              ObjDesc.ObjectName,

```

```

        Reason);
        goto MOD_EXIT;
    }
else
{
    /******
    /* Start processing event messages */
    /******
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:
if (pBuffer != NULL)
{
    free(pBuffer);
}

/******
/* Close the logger event queue */
/******
if (hEventQueue != MQHO_UNUSABLE_HOBJ)
{
    MQCLOSE(hConn, &hEventQueue, MQCO_NONE, &CompCode, &Reason);
}

/******
/* Disconnect */
/******
if (hConn != MQHC_UNUSABLE_HCONN)
{
    MQDISC(&hConn, &CompCode, &Reason);
}

return 0;
}

/******
/* Function: ProcessPCF */
/******
/*
/* Input Parameters:  Handle to queue manager connection */
/*                   Handle to the opened logger event queue object */
/*                   Pointer to a memory buffer to store the incoming PCF */
/*                   message */
/*
/* Output Parameters: None */
/*
/* Logic: Wait for messages to appear on the logger event queue and display */
/*         their formatted contents. */
/*
/******
static void ProcessPCF(MQHCONN    hConn,
                    MQHOBJ     hEventQueue,
                    PMQCHAR    pBuffer)
{
    MQCFH    * pCfh;
    MQCFST   * pCfst;
    MQGMO    Gmo      = { MQGMO_DEFAULT };
    MQMD     Mqmd     = { MQMD_DEFAULT };
    PMQCHAR  pPCFCmd;
    MQLONG   CompCode = MQCC_OK;
    MQLONG   Reason   = MQRC_NONE;
    MQLONG   MsgLen;
    PMQCHAR  Parm = NULL;

    Gmo.Options = MQGMO_WAIT +
                 MQGMO_CONVERT +
                 MQGMO_FAIL_IF QUIESCING;
    Gmo.WaitInterval = MQWI_UNLIMITED; /* Set timeout value */

    /******
    /* Process response Queue */
    /******
    while (Reason == MQRC_NONE)
    {
        memcpy(&Mqmd.MsgId, MQMI_NONE, sizeof(Mqmd.MsgId));
        memset(&Mqmd.CorrelId, 0, sizeof(Mqmd.CorrelId));

        MQGET( hConn,
              hEventQueue,

```

```

        &Mqmd,
        &Gmo,
        MAX_MESSAGE_LENGTH,
        pBuffer,
        &MsgLen,
        &CompCode,
        &Reason );

if (Reason != MQRC_NONE)
{
    switch(Reason)
    {
        case MQRC_NO_MSG_AVAILABLE:
            printf("Timed out");
            break;

        default:
            printf("MQGET ended with reason code %" Int32 "d\n", Reason);
            break;
    }
    goto MOD_EXIT;
}

/*****
/* Only expect PCF event messages on this queue */
/*****
if (memcmp(Mqmd.Format, MQFMT_EVENT, MQ_FORMAT_LENGTH))
{
    printf("Unexpected message format '%8.8s' received\n", Mqmd.Format);
    continue;
}

/*****
/* Build the output by parsing the received PCF message, first the */
/* header, then each of the parameters */
/*****
pCfh = (MQCFH *)pBuffer;

if (pCfh->Reason != MQRC_NONE)
{
    printf("-----\n");
    printf("Event Message Received\n");

    Parm = ParmToString(pCfh->Command);
    if (Parm != NULL)
    {
        printf("Command  :%s \n",Parm);
    }
    else
    {
        printf("Command  :%" Int32 "d \n",pCfh->Command);
    }

    printf("CompCode :%" Int32 "d\n" ,pCfh->CompCode);

    Parm = ParmToString(pCfh->Reason);
    if (Parm != NULL)
    {
        printf("Reason   :%s \n",Parm);
    }
    else
    {
        printf("Reason   :%" Int32 "d \n",pCfh->Reason);
    }
}

pPCFCmd = (PMQCHAR) (pCfh+1);
printf("-----\n");
while(pCfh->ParameterCount-->0)
{
    pCfst = (MQCFST *) pPCFCmd;
    switch(pCfst->Type)
    {
        case MQCFT_STRING:
            Parm = ParmToString(pCfst->Parameter);
            if (Parm != NULL)
            {
                printf("%-32s",Parm);
            }
            else
            {
                printf("%-32" Int32 "d",pCfst->Parameter);
            }

```

```

    }

    fwrite(pCfst->String, pCfst->StringLength, 1, stdout);
    pPCFCmd += pCfst->StrucLength;
    break;

    default:
        printf("Unrecognised datatype %" Int32 "d returned\n", pCfst->Type);
        goto MOD_EXIT;
    }
    putchar('\n');
}
}

```

## 範例輸出

此應用程式會產生下列輸出形式:

```

/*****
/* Sample Logger Event Monitor start */
*****/
-----
Event Message Received
Command :Logger Event Command
CompCode :0
Reason :Logger Status
-----
Queue Manager Name          CSIM

Current Log Extent          AMQA000001
Restart Log Extent         AMQA000001
Media Log Extent           AMQA000001
Log Path                    QMCSIM
-----

```

## 相關概念

[第 41 頁的『日誌程式事件使用情形』](#)

請利用這個頁面來檢視如何使用日誌程式事件來判斷佇列管理程式重新啟動或媒體回復不再需要的日誌範圍。

[第 40 頁的『指令事件用法』](#)

請利用這個頁面來檢視如何使用指令事件來產生已執行之指令的審核追蹤

## 相關參考

[第 41 頁的『日誌程式事件產生』](#)

請利用這個頁面來檢視導致產生日誌程式事件的狀況，以及瞭解未產生日誌程式事件的情況

## 權限配置事件

透過指令行、MQSC、PCF 或對應的 IBM i 指令，從任何安全控制作業進行變更時，即會輸出權限配置事件。

事件資料包含下列資訊:

### 原始資訊

包含進行變更的來源佇列管理程式、進行變更的使用者 ID，以及進行變更的方式，例如透過主控台指令。

### 環境定義資訊

來自指令訊息之訊息資料中的環境定義資訊抄本。

當輸入指令作為 SYSTEM.ADMIN.COMMAND.QUEUE 佇列。

### 權限記錄身分

包含設定檔名稱，以及權限記錄的物件類型。

### 物件屬性

包含權限記錄中所有屬性的值。

如果是變更權限記錄事件，則會產生兩則訊息，其中一則具有變更之前的資訊，另一則具有變更之後的資訊。

每個產生的事件訊息都會放置在 SYSTEM.ADMIN.CONFIG.EVENT 佇列。

## 相關參考

第 8 頁的『事件類型』

請利用這個頁面來檢視佇列管理程式或通道實例可以報告的檢測事件類型

## 產生權限配置事件

請利用這個頁面來檢視導致產生權限配置事件的狀況，以及瞭解未產生權限配置事件的情況。

權限配置事件會通知您權限記錄屬性的變更。權限配置事件有三種類型：

- [變更權限記錄](#)
- [刪除權限記錄](#)
- [重新整理權限記錄](#)

當 **CONFIGEV** 佇列管理程式屬性設為 *ENABLED*，且發出下列任何指令或其 MQSC 對等項目時，即使權限記錄沒有實際變更，也會將權限事件訊息放置在配置事件佇列中：

- [刪除權限記錄 PCF 指令](#)
- [設定權限記錄 PCF 指令](#)
- [setmqaut 控制指令](#)
- [RVKMQMAUT CL 指令](#)
- [GRTMQMAUT CL 指令](#)

## 未產生權限配置事件時

在下列情況下，不會產生權限配置事件訊息：

- 當指令失敗時
- 當佇列管理程式在嘗試將訊息放入事件佇列時發生錯誤，在此情況下指令會完成，但不會產生事件訊息
- 建立或刪除佇列管理程式時
- 刪除物件時，不論刪除指令上的 **AUTHREC** 選項為何。對應的指令事件會顯示該作業，該作業不適用於個別使用者的權限記錄。

## 相關概念

第 38 頁的『指令事件』

指令事件是 MQSC 或 PCF 指令已順利執行的通知。

## 相關參考

[重新整理佇列管理程式](#)

Multi

## 在 Multiplatforms 上監視檢測事件的範例程式

**amqsevt** 會格式化佇列管理程式可以建立且隨 IBM MQ for Multiplatforms 提供的檢測事件。程式會從事件佇列讀取訊息，並將它們格式化成可讀取的字串。

作為範例程式，同時提供來源和二進位。此範例在所有 Multiplatforms 上提供，包括 IBM i。

單一二進位檔 **amqsevt** (或 **amqsevt.exe**) 隨附於範例檔案集，並安裝在範例 bin (**tools\c\samples\bin** 或 **bin64**) 目錄中。

原始檔 **amqsevt.c** 也隨附於範例檔案集中，並安裝在 **samples** 目錄中，即 Windows 上的 **tools\c\samples**。

請注意，程式可以使用 [MQCB](#) 來擷取訊息，從多個事件佇列讀取並訂閱多個主題。

以用戶端身分執行時，範例可以連接至任何佇列管理程式，包括 z/OS。

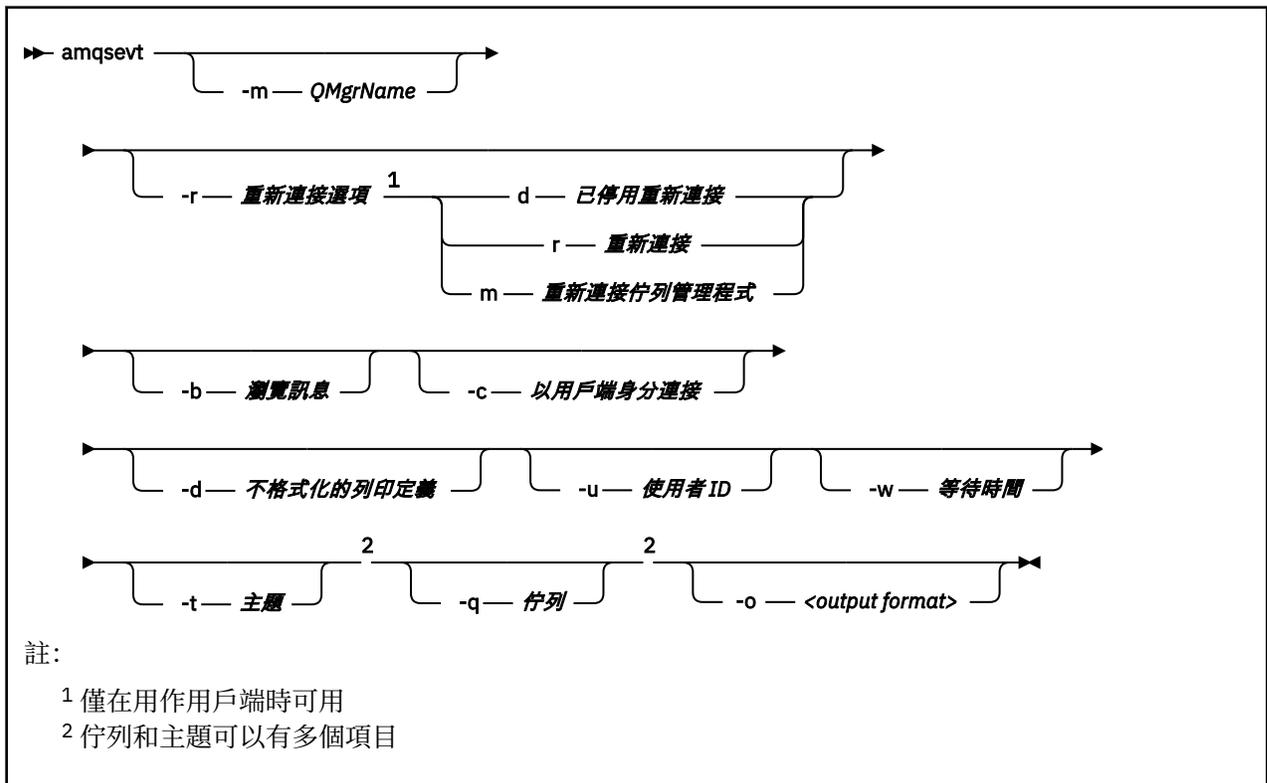


**小心：**您可以在不指定任何參數的情況下使用程式，在此情況下，程式會嘗試連接至預設佇列管理程式，並從標準事件佇列集 (SYSTEM.ADMIN.\*.EVENT)。

在此狀況下，程式會一直等待訊息，直到您按 Enter 鍵結束程式為止。

不過，您更有可能以說明的各種選項來使用程式。

## 語法



## 選用參數

### `-m QueueManagerName`

指定用於讀取事件的特定佇列管理程式。

### `-r Reconnection Options`

作為用戶端使用時自動重新連線選項。可能值包括：

**d**  
已停用重新連接用戶端

**r**  
重新連接用戶端

**m**  
重新連接佇列管理程式

**-b**  
僅瀏覽記錄，而不是破壞性地讀取訊息

**-c**  
選取連線作為用戶端。

**-d**  
選取第二個範例中使用的列印模式。MQI 常數的列印方式與它們出現在標頭檔中的方式完全相同。

### `-u User ID`

指定特定使用者並導致出現要求密碼的提示

### `-w Wait`

如果在指定的秒數內沒有任何事件訊息到達，則會導致程式結束。

請注意，如果您未指定時間，則只有在您按 Enter 鍵時，程式才會正常結束。

## **-t Topic** 及

## **-q Queue**

在指令行上可以多次指定 **-q** 及 **-t** 選項。

因此，可以從程式的單一執行中讀取部分標準佇列，也可以從主題 (如果事件傳送至它們) 讀取。

如果指令行上未指定任何佇列或主題，則會開啟預設事件佇列。

**註:** 程式會偵測它是否已作為用戶端連接至 z/OS 佇列管理程式，並適當地變更預設事件佇列集，因為 z/OS 沒有 SYSTEM.ADMIN.LOGGER.EVENT 佇列。

使用主題時，程式會使用具有受管理佇列的不可延續訂閱，以便在結束時清除所有項目。

### **V 9.1.0** **-o <output format>**

輸出的格式。可能的值為：

#### **text**

標準文字格式; 這是預設值。

#### **json**

標準 JSON 格式; 任何瞭解 JSON 的應用程式都可以採用此輸出並直接處理它。

## 範例輸出

下列三個範例顯示程式的輸出。

第一個範例使用預設格式化選項，其中程式會採用欄位的 MQI 定義並格式化輸出，讓輸出更容易讀取。

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : Queue Mgr Event
Reason               : Unknown Alias Base Queue
Event created        : 2015/06/17 13:47:07.02 GMT
  Queue Mgr Name     : V8003_A
  Queue Name         : EVT.NO.BASE.QUEUE
  Base Object Name   : EVT.NOT.DEFINED
  Appl Type          : Unix
  Appl Name          : amqsput
  Base Type          : Queue
```

第二個範例顯示不嘗試轉換 MQI 常數的替代格式 (使用 **-d** 選項)。對於一些尋找特定 MQI 值的 Scripting 工具而言，這可能更適合。

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : MQCMD_Q_MGR_EVENT
Reason               : MQRC_UNKNOWN_ALIAS_BASE_Q
Event created        : 2015/06/17 13:52:48.18 GMT
  MQCA_Q_MGR_NAME    : V8003_A
  MQCA_Q_NAME        : EVT.NO.BASE.QUEUE
  MQCA_BASE_OBJECT_NAME : EVT.NOT.DEFINED
  MQIA_APPL_TYPE     : MQAT_UNIX
  MQCACF_APPL_NAME   : amqsput
  MQIA_BASE_TYPE     : MQOT_Q
```

### **V 9.1.0** 第三個範例顯示指令的 JSON 輸出

```
amqsevt -m QM1 -q SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE -o json
```

```
{
  "eventSource" : { "objectName": "SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE",
                   "objectType" : "Queue" },
  "eventType" : {
    "name" : "Activity Trace",
    "value" : 209
  },
  "eventReason" : {
    "name" : "None",
    "value" : 0
  },
  "eventCreation" : {
    "timeStamp" : "2018-07-10T12:44:26Z",
    "epoch" : 1531226666
  },
  "eventData" : {
    "queueMgrName" : "QM1",
    "hostName" : "<yourhostname>",
    "startDate" : "2018-07-10",
    "startTime" : "13:44:25",
    "endDate" : "2018-07-10",
    "endTime" : "13:44:26",
    "commandLevel" : 910,
    ...
  }
}
```

## 範例用法

下列範例顯示如何使用多個佇列:

```
amqsevt -m QM1 -q SYSTEM.ADMIN.QMGR.EVENT -q SYSTEM.ADMIN.PERM.EVENT -w 1
```

### 相關概念

#### [第 5 頁的『事件監視』](#)

事件監視是在佇列管理程式網路中偵測發生 檢測事件的程序。檢測事件是由佇列管理程式或通道實例偵測到的事件邏輯組合。這類事件會導致佇列管理程式或通道實例將特殊訊息 (稱為 事件訊息) 放置在事件佇列上。

#### [第 6 頁的『檢測事件』](#)

檢測事件是佇列管理程式或通道實例偵測並將特殊訊息 (稱為 事件訊息) 放置在事件佇列上之條件的邏輯組合。

### 相關參考

#### [C 程式設計](#)

#### [第 42 頁的『監視日誌程式事件佇列的範例 C 程式』](#)

請利用這個頁面來檢視範例 C 程式，它會監視日誌程式事件佇列中是否有新的事件訊息、讀取那些訊息，以及將訊息內容放到 `stdout` 中。

## 訊息監視

訊息監視是識別訊息透過佇列管理程式網路所採取的路徑的處理程序。透過識別活動類型，以及代表訊息執行的活動順序，可以判定訊息路徑。

當訊息通過佇列管理程式網路時，各種程序會代表訊息執行活動。請使用下列其中一項技術來決定訊息路徑:

- IBM MQ 顯示路徑應用程式 (`dspmqrte`)
- 活動記錄中
- 追蹤路徑傳訊

這些技術都會產生特殊訊息，其中包含訊息在通過佇列管理程式網路時所執行之活動的相關資訊。請使用這些特殊訊息中傳回的資訊來達成下列目標:

- 記錄訊息活動。
- 決定訊息的前次已知位置。
- 偵測佇列管理程式網路中的遞送問題。

- 協助判斷佇列管理程式網路中遞送問題的原因。
- 請確認佇列管理程式網路正確執行中。
- 熟悉佇列管理程式網路的執行。
- 追蹤已發佈的訊息。

## 相關參考

### 訊息類型

## 活動和業務

活動是應用程式代表訊息執行的離散動作。活動由作業組成，這些作業是應用程式執行的單一工作片段。

下列動作是活動的範例：

- 訊息通道代理程式 (MCA) 將訊息從傳輸佇列傳送至通道
- MCA 會從通道接收訊息，並將它放置在其目標佇列中
- 應用程式從佇列取得訊息，並在回應中放置回覆訊息。
- IBM MQ 發佈/訂閱引擎會處理訊息。

活動由一或多個 作業組成。作業是應用程式執行的單一工作片段。例如，MCA 從通道下的傳輸佇列傳送訊息的活動包含下列作業：

1. 從傳輸佇列取得訊息 (取得 作業)。
2. 透過通道傳送訊息 (傳送 作業)。

在發佈/訂閱網路中，IBM MQ 發佈/訂閱引擎處理訊息的活動可能包含下列多個作業：

1. 將訊息放入主題字串 (放置 作業)。
2. 對於每一個被視為接收訊息的訂閱者 (發佈 作業、捨棄的發佈 作業或排除的發佈 作業)，沒有任何以上的作業。

## 來自活動的資訊

當訊息透過佇列管理程式網路遞送時，您可以記錄資訊，以識別在訊息上執行的活動順序。您可以根據在訊息上執行的活動順序，決定透過佇列管理程式網路的訊息路徑，並可取得下列資訊：

### 訊息的前次已知位置

如果訊息未到達其預期的目的地，您可以從完整或局部訊息路徑來判斷訊息的前次已知位置。

### 佇列管理程式網路的配置問題

在研究透過佇列管理程式網路的訊息路徑時，您可能看到訊息未達到預期。發生這種情況有許多原因，例如，如果通道非作用中，則訊息可能採用替代路徑。

對於發佈/訂閱應用程式，您也可以判定發佈至主題的訊息路徑，以及因發佈至訂閱者而在佇列管理程式網路中流動的任何訊息。

在這種情況下，系統管理者可以判斷佇列管理程式網路中是否有任何問題，如果適當的話，請更正這些問題。

## 訊息路徑

視您決定訊息路徑的原因而定，您可以使用下列一般方法：

### 使用針對追蹤路徑訊息所記錄的活動資訊

追蹤路徑訊息會記錄特定目的的活動資訊。您可以使用它們來判斷佇列管理程式網路的配置問題，或判斷訊息的前次已知位置。如果產生追蹤路徑訊息以判斷未到達其預期目的地之訊息的前次已知位置，則它可以模擬原始訊息。這可讓追蹤路徑訊息有最大機會遵循原始訊息所採取的路徑。

IBM MQ 顯示路徑應用程式可以產生追蹤路徑訊息。

## 使用針對原始訊息所記錄的活動資訊

您可以針對活動記錄啟用任何訊息，並代表其記錄活動資訊。如果訊息未到達其預期的目的地，您可以使用記錄的活動資訊來判斷訊息的前次已知位置。透過使用原始訊息中的活動資訊，可以判定最精確的訊息路徑，並指向最後一個已知位置。若要使用此方法，必須針對活動記錄啟用原始訊息。

**警告:** 避免啟用佇列管理程式網路中的所有訊息來進行活動記錄。啟用活動記錄的訊息可以代表其產生許多活動報告。如果佇列管理程式網路中的每一則訊息都已啟用活動記錄，則佇列管理程式網路資料流量可能會增加至不可接受的層次。

### 相關概念

第 50 頁的『[訊息監視](#)』

訊息監視是識別訊息透過佇列管理程式網路所採取的路徑的處理程序。透過識別活動類型，以及代表訊息執行的活動順序，可以判定訊息路徑。

第 52 頁的『[訊息遞送技術](#)』

活動記錄和追蹤路徑傳訊技術可讓您在訊息透過佇列管理程式網路遞送時，記錄訊息的活動資訊。

第 57 頁的『[追蹤路徑傳訊](#)』

追蹤路徑傳訊是一種使用 [追蹤路徑訊息](#) 來記錄訊息活動資訊的技術。追蹤路徑傳訊包括將追蹤路徑訊息傳送至佇列管理程式網路。

### 相關工作

[撰寫您自己的訊息通道代理程式](#)

## 訊息遞送技術

活動記錄和追蹤路徑傳訊技術可讓您在訊息透過佇列管理程式網路遞送時，記錄訊息的活動資訊。

### 活動記錄中

如果訊息已指定適當的報告選項，則會要求應用程式在透過佇列管理程式網路遞送時產生活動報告。當應用程式代表訊息執行活動時，可以產生活動報告，並遞送至適當的位置。活動報告包含在訊息上執行之活動的相關資訊。

使用活動報告收集的活動資訊必須依序排列，才能決定訊息路徑。

### 追蹤路徑傳訊

追蹤路徑傳訊是一種涉及將 [追蹤路徑訊息](#) 傳送至佇列管理程式網路的技術。當應用程式代表追蹤路徑訊息執行活動時，可以在追蹤路徑訊息的訊息資料中累計活動資訊，或產生活動報告。如果在追蹤路徑訊息的訊息資料中累計活動資訊，則當它到達其目標佇列時，可以產生包含追蹤路徑訊息中所有資訊的追蹤路徑回覆訊息，並將其遞送至適當的位置。

因為追蹤路徑訊息專用於記錄代表其執行的活動序列，所以與要求活動報告的一般訊息相比，有更多處理選項可用。

## 活動記錄和追蹤路徑傳訊的比較

活動記錄及追蹤路徑傳訊都可以提供活動資訊，以判定訊息透過佇列管理程式網路所採取的路徑。這兩種方法各有優勢。

優點	活動記錄中	追蹤路徑傳訊
可判定訊息的前次已知位置	是	是
可以判定佇列管理程式網路的配置問題	是	是
可由任何訊息要求 (不限制與追蹤路徑訊息一起使用)	是	否
訊息資料維持未修改	是	否
已正常處理訊息	是	否
活動資訊可以在訊息資料中累計	否	是

優點	活動記錄中	追蹤路徑傳訊
將選用訊息遞送至目標佇列	否	是
如果在無限迴圈中捕捉到訊息，則可以偵測並處理該訊息	否	是
活動資訊可以可靠地放置在順序中	否	是
提供以顯示活動資訊的應用程式	否	是

## 訊息路徑完整性

在某些情況下，無法識別代表訊息執行的完整活動序列，因此只能判定局部訊息路徑。訊息遞送的佇列管理程式網路會直接影響訊息遞送的完整性。訊息路徑的完整性取決於佇列管理程式網路中的佇列管理程式層次，如下所示：

### IBM WebSphere MQ 6.0 及後續版本的佇列管理程式

MCA 及連接至佇列管理程式 (IBM WebSphere MQ 6.0 或後續版次) 的使用者撰寫應用程式可以記錄與代表訊息執行之活動相關的資訊。活動資訊的記錄由佇列管理程式屬性 ACTIVREC 和 ROUTERC 控制。如果佇列管理程式網路僅包含 IBM WebSphere MQ 6.0 或後續版次的佇列管理程式，則可以判定完整訊息路徑。

### IBM WebSphere MQ 之前的佇列管理程式 IBM WebSphere MQ 6.0

在 IBM WebSphere MQ 6.0 之前連接至佇列管理程式的應用程式不會記錄它們代表訊息所執行的活動。如果佇列管理程式網路包含 IBM WebSphere MQ 6.0 之前的任何佇列管理程式，則只能判斷局部訊息路徑。

## 如何儲存活動資訊

IBM MQ 會將活動資訊儲存在活動報告、追蹤路徑訊息或追蹤路徑回覆訊息中。在每一種情況下，資訊都會儲存在稱為活動 PCF 群組的結構中。追蹤路徑訊息或追蹤路徑回覆訊息可以包含許多活動 PCF 群組，視對訊息執行的活動數目而定。活動報告包含一個「活動 PCF」群組，因為會針對每個記錄的活動產生個別活動報告。

使用追蹤路徑傳訊，可以記錄其他資訊。此其他資訊儲存在稱為 *TraceRoute* PCF 群組的結構中。*TraceRoute* PCF 群組包含許多 PCF 結構，用來儲存其他活動資訊，以及指定選項以決定在透過佇列管理程式網路遞送時如何處理追蹤路徑訊息。

### 相關概念

第 53 頁的『活動記錄中』

活動記錄是一種技術，用來決定訊息透過佇列管理程式網路所採用的路徑。為了判斷訊息所採取的路徑，會記錄代表訊息執行的活動。

第 57 頁的『追蹤路徑傳訊』

追蹤路徑傳訊是一種使用追蹤路徑訊息來記錄訊息活動資訊的技術。追蹤路徑傳訊包括將追蹤路徑訊息傳送至佇列管理程式網路。

### 相關參考

第 61 頁的『TraceRoute PCF 群組』

*TraceRoute* PCF 群組中的屬性會控制追蹤路徑訊息的行為。*TraceRoute* PCF 群組位於每個追蹤路徑訊息的訊息資料中。

第 91 頁的『活動報告訊息資料』

請利用這個頁面來檢視活動報告訊息中的活動 PCF 群組所包含的參數。只有在已執行特定作業時，才會傳回部分參數。

## 活動記錄中

活動記錄是一種技術，用來決定訊息透過佇列管理程式網路所採用的路徑。為了判斷訊息所採取的路徑，會記錄代表訊息執行的活動。

使用活動記錄時，每一個代表訊息執行的活動都可以記錄在活動報告中。活動報告是一種報告訊息類型。每一個活動報告都包含代表訊息執行活動之應用程式的相關資訊、活動發生時間，以及作為活動一部分執行

之作業的相關資訊。活動報告通常會遞送至回覆目的地佇列，並在其中一起收集。透過研究與訊息相關的活動報告，您可以判定訊息透過佇列管理程式網路所採取的路徑。

## 活動報告使用情形

透過佇列管理程式網路遞送訊息時，可以產生活動報告。您可以使用下列方式來使用活動報告資訊：

### 決定訊息的前次已知位置

如果啟用活動記錄的訊息未達到其預期的目的地，則可以研究透過佇列管理程式網路遞送訊息時為訊息產生的活動報告，以判定訊息的最後一個已知位置。

### 判斷佇列管理程式網路的配置問題

啟用活動記錄的訊息數可以傳送至佇列管理程式網路。透過研究與每一個訊息相關的活動報告，可能會發現它們未採取預期的路徑。發生這種情況有許多原因，例如，通道可能已停止，強制訊息採取替代路徑。在這些狀況下，系統管理者可以判斷佇列管理程式網路中是否有任何問題，如果有，請更正它們。

註：您可以使用 IBM MQ 顯示路徑應用程式，將活動記錄與追蹤路徑訊息一起使用。

## 活動報告格式

活動報告是應用程式代表訊息執行活動所產生的 PCF 訊息。活動報告是包含訊息描述子及訊息資料的標準 IBM MQ 報告訊息，如下所示：

### 訊息描述子

- MQMD 結構

### 訊息資料

- 內嵌 PCF 標頭 (MQEPH)
- 活動報告訊息資料

活動報告訊息資料包含活動 PCF 群組，如果針對追蹤路徑訊息產生，則為 *TraceRoute* PCF 群組。

### 相關參考

[MQMD-訊息描述子](#)

[MQEPH-內嵌 PCF 標頭](#)

## 控制活動記錄

在佇列管理程式層次啟用活動記錄。若要啟用整個佇列管理程式網路，請個別啟用網路中的每個佇列管理程式，以進行活動記錄。如果您啟用更多佇列管理程式，則會產生更多活動報告。

## 關於這項作業

當訊息透過佇列管理程式遞送時，如果要產生訊息的活動報告：請定義訊息來要求活動報告；啟用佇列管理程式以進行活動記錄；並確定在訊息上執行活動的應用程式能夠產生活動報告。

如果您不想要在訊息透過佇列管理程式遞送時針對訊息產生活動報告，請停用佇列管理程式以進行活動記錄。

## 程序

### 1. 要求訊息的活動報告

- a) 在訊息的訊息描述子中，於 **報告** 欄位中指定 `MQRO_ACTIVITY`。
- b) 在訊息的訊息描述子中，於 **ReplyToQ** 欄位中指定回覆目的地佇列的名稱。

**警告：**避免啟用佇列管理程式網路中的所有訊息來進行活動記錄。啟用活動記錄的訊息可以代表其產生許多活動報告。如果佇列管理程式網路中的每一則訊息都已啟用活動記錄，則佇列管理程式網路資料流量可能會增加至不可接受的層次。

### 2. 針對活動記錄啟用或停用佇列管理程式。

使用 `MQSC` 指令 `ALTER QMGR` 並指定參數 `ACTIVREC`，以變更佇列管理程式屬性的值。值可以為：

## MSG

佇列管理程式已啟用活動記錄。任何產生的活動報告都會遞送至訊息的訊息描述子中指定的回覆目的地佇列。這是預設值。

## 佇列

佇列管理程式已啟用活動記錄。任何產生的活動報告都會遞送至本端系統佇列 SYSTEM.ADMIN.ACTIVITY.QUEUE。系統佇列也可以用來將活動報告轉遞至共用佇列。

## 已停用

已停用佇列管理程式進行活動記錄。在此佇列管理程式的範圍內，不會產生任何活動報告。

例如，啟用佇列管理程式以進行活動記錄，並指定將產生的任何活動報告遞送至本端系統佇列 SYSTEM.ADMIN.ACTIVITY.QUEUE，請使用下列 MQSC 指令：

```
ALTER QMGR ACTIVREC(QUEUE)
```

**記住：**當您修改 *ACTIVREC* 佇列管理程式屬性時，除非重新啟動通道，否則執行中 MCA 不會偵測變更。

3. 請確定您的應用程式使用與 MCA 相同的演算法來判斷是否要產生訊息的活動報告：

- a) 請驗證訊息已要求產生活動報告
- b) 驗證訊息目前所在的佇列管理程式是否已啟用活動記錄
- c) 將活動報告放置在 *ACTIVREC* 佇列管理程式屬性所決定的佇列上

## 設定活動報告的共用佇列

當報告遞送至本端系統佇列時，如果要判斷與特定訊息相關的活動報告位置，在單一節點上使用共用佇列會更有效率

## 開始之前

設定 **ACTIVREC** 參數，以啟用佇列管理程式進行活動記錄，並指定將產生的任何活動報告遞送至本端系統佇列 SYSTEM.ADMIN.ACTIVITY.QUEUE。

## 關於這項作業

如果佇列管理程式網路中的佇列管理程式數目設定為將活動報告遞送至本端系統佇列，則判定與特定訊息相關的活動報告位置可能非常耗時。或者，使用單一節點，它是管理一般佇列的佇列管理程式。佇列管理程式網路中的所有佇列管理程式都可以將活動報告遞送至這個共用佇列。使用一般佇列的好處是佇列管理程式不需要將活動報告遞送至訊息中指定的回覆目的地佇列，而且在決定與訊息相關的活動報告位置時，您只會查詢一個佇列。

若要設定共用佇列，請執行下列步驟：

## 程序

1. 選取或定義佇列管理程式作為單一節點
2. 在單一節點上，選取或定義要作為共用佇列的佇列
3. 在所有要將活動報告遞送至共用佇列的佇列管理程式上，重新定義本端系統佇列 SYSTEM.ADMIN.ACTIVITY.QUEUE 作為遠端佇列定義：
  - a) 指定單一節點的名稱作為遠端佇列管理程式名稱
  - b) 指定共用佇列的名稱作為遠端佇列名稱

## 決定訊息遞送資訊

若要判定訊息路徑，請從收集的活動報告中取得資訊。判斷回覆目的地佇列上是否有足夠的活動報告，可讓您判斷必要的資訊，並依序排列活動報告。

## 關於這項作業

將活動報告放置在回覆目的地佇列上的順序，不一定會與執行活動的順序產生關聯。除非針對追蹤路徑訊息產生活動報告，否則您必須手動排序活動報告，在此情況下，您可以使用 IBM MQ 顯示路徑應用程式來排序活動報告。

判斷回覆目的地佇列上是否有足夠的活動報告可讓您取得必要的資訊：

## 程序

1. 比較活動報告和原始訊息的 ID，以識別回覆目的地佇列上所有相關的活動報告。請確定您設定原始訊息的報告選項，以便活動報告可以與原始訊息產生關聯。
2. 從回覆目的地佇列中排序所識別的活動報告。  
您可以從活動報告中使用下列參數：

### **OperationType**

執行的作業類型可讓您判定在現行活動報告之前或之後直接產生的活動報告。

例如，活動報告詳細說明 MCA 從通道下的傳輸佇列傳送訊息。活動報告中詳細說明的前次作業具有 *OperationType send*，以及使用通道 CH1 將訊息傳送至目的地佇列管理程式 QM1 的詳細資料。這表示對訊息執行的下一個活動將發生在佇列管理程式 QM1 上，且它將從通道 CH1 開始 *receive* 作業。透過使用此資訊，您可以識別下一個活動報告，前提是它存在且已獲得。

### **OperationDate 和 OperationTime**

您可以從每一個活動報告中作業的日期和時間，決定活動的一般順序。

**警告：**除非佇列管理程式網路中每個佇列管理程式的系統時鐘都已同步，否則依日期和時間排序並不保證活動報告的順序正確。您必須手動建立訂單。

活動報告的順序代表訊息透過佇列管理程式網路所採用的路徑或局部路徑。

3. 從依序活動報告中的活動資訊取得您需要的資訊。  
如果訊息的相關資訊不足，您可能可以獲得進一步的活動報告。

## 擷取進一步活動報告

若要判定訊息路徑，必須從收集的活動報告中提供足夠的資訊。如果您從訊息指定的回覆目的地佇列中擷取與訊息相關的活動報告，但您沒有必要的資訊，請尋找進一步的活動報告。

## 關於這項作業

若要判定任何進一步活動報告的位置，請執行下列步驟：

## 程序

1. 對於佇列管理程式網路中將活動報告遞送至共用佇列的任何佇列管理程式，請從共用佇列中擷取其 *CorrelId* 符合原始訊息的 *MsgId* 的活動報告。
2. 對於佇列管理程式網路中未將活動報告遞送至一般佇列的任何佇列管理程式，請擷取活動報告，如下所示：
  - a) 請檢查現有的活動報告，以識別透過其遞送訊息的佇列管理程式。
  - b) 針對這些佇列管理程式，識別已啟用活動記錄的佇列管理程式。
  - c) 針對這些佇列管理程式，識別未將活動報告傳回指定的回覆目的地佇列的任何佇列。
  - d) 對於您識別的每一個佇列管理程式，請檢查系統佇列 *SYSTEM.ADMIN.ACTIVITY.QUEUE* 並擷取其 *CorrelId* 符合原始訊息的 *MsgId* 的任何活動報告。
  - e) 如果您在系統佇列上找不到活動報告，請檢查佇列管理程式無法傳送郵件的佇列 (如果有的話)。只有在設定報告選項 *MQRO\_DEAD\_LETTER\_Q* 時，才能將活動報告遞送至無法傳送郵件的佇列。
3. 依序排列所有獲得的活動報告。  
然後，活動報告的順序代表訊息所採用的路徑或部分路徑。
4. 從依序活動報告中的活動資訊取得您需要的資訊。  
在某些情況下，記錄的活動資訊無法到達指定的回覆目的地佇列、一般佇列或系統佇列。

## 未取得活動資訊的情況

若要判斷代表訊息執行的完整活動順序，必須獲得與每個活動相關的資訊。如果尚未記錄或獲得與任何活動相關的資訊，您只能判斷部分活動序列。

在下列情況下，不會記錄活動資訊：

- 訊息由早於 IBM WebSphere MQ 6.0 的佇列管理程式處理。
- 訊息由未啟用活動記錄的佇列管理程式處理。
- 預期處理訊息的應用程式不在執行中。

在下列情況下，所記錄的活動資訊無法呼叫到指定的回覆目的地佇列：

- 未定義通道將活動報告遞送至回覆目的地佇列。
- 將活動報告遞送至回覆目的地佇列的通道不在執行中。
- 將活動報告遞送回回覆目的地佇列所在的佇列管理程式 (佇列管理程式別名) 的遠端佇列定義未定義。
- 產生原始訊息的使用者沒有開啟或放置佇列管理程式別名的權限。
- 產生原始訊息的使用者沒有開啟或放置回覆目的地佇列的權限。
- 禁止放置回覆目的地佇列。

在下列情況下，所記錄的活動資訊無法到達系統佇列或一般佇列：

- 如果要使用共用佇列，且沒有定義通道將活動報告遞送至共用佇列。
- 如果要使用共用佇列，且將活動報告遞送至共用佇列的通道不在執行中。
- 如果要使用共用佇列，且未正確定義系統佇列。
- 產生原始訊息的使用者對系統佇列沒有開啟或放置權限。
- 禁止放置系統佇列。
- 如果要使用共用佇列，且產生原始訊息的使用者沒有開啟或放置共用佇列的權限。
- 如果要使用共用佇列，且禁止放置共用佇列。

在這些情況下，如果活動報告沒有指定報告選項 `MQRO_DISCARD_MSG`，則如果在拒絕活動報告的佇列管理程式上定義了活動報告，則可以從無法傳送郵件的佇列中擷取活動報告。只有在從中產生活動報告的原始訊息已在訊息描述子的「報告」欄位中指定 `MQRO_PASS_DISCARD_AND_EXPIRY` 及 `MQRO_DISCARD_MSG` 時，活動報告才會指定此報告選項。

## 追蹤路徑傳訊

追蹤路徑傳訊是一種使用追蹤路徑訊息來記錄訊息活動資訊的技術。追蹤路徑傳訊包括將追蹤路徑訊息傳送至佇列管理程式網路。

當透過佇列管理程式網路遞送追蹤路徑訊息時，會記錄活動資訊。此活動資訊包括執行活動的應用程式、執行活動的時間，以及作為活動一部分執行的作業的相關資訊。基於下列目的，您可以使用使用追蹤路徑傳訊所記錄的資訊：

### 判斷訊息的最後一個已知位置

如果訊息未到達其預期的目的地，您可以使用針對追蹤路徑訊息所記錄的活動資訊來判斷訊息的最後一個已知位置。追蹤路徑訊息會傳送至與原始訊息具有相同目標目的地的佇列管理程式網路，並預期它遵循相同的路徑。活動資訊可以累積在追蹤路徑訊息的訊息資料中，或使用活動報告來記錄。若要增加追蹤路徑訊息遵循與原始訊息相同的路徑的機率，您可以修改追蹤路徑訊息以模擬原始訊息。

### 判斷佇列管理程式網路的配置問題

追蹤路徑訊息會傳送至佇列管理程式網路，並記錄活動資訊。透過研究針對追蹤路徑訊息所記錄的活動資訊，可以發現追蹤路徑訊息未遵循預期的路徑。發生這種情況的原因有很多，例如，通道可能處於非作用中狀態，強制訊息採取替代路徑。在這些狀況下，系統管理者可以判斷佇列管理程式網路中是否有任何問題，如果有，請更正它們。

您可以使用 IBM MQ 顯示路徑應用程式，將追蹤路徑訊息配置、產生及放置到佇列管理程式網路中。

**警告：**如果您將追蹤路徑訊息放置到配送清單，則結果未定義。

## 相關概念

第 108 頁的『[追蹤路徑訊息參照](#)』

請利用這個頁面來取得追蹤路徑訊息格式的概觀。追蹤路徑訊息資料包括的參數說明追蹤路徑訊息所造成的活動

## 如何記錄活動資訊

利用追蹤路徑傳訊，您可以將活動資訊記錄在追蹤路徑訊息的訊息資料中，或使用活動報告。或者，您可以同時使用這兩種技術。

## 累計追蹤路徑訊息的訊息資料中的活動資訊

當透過佇列管理程式網路遞送追蹤路徑訊息時，可以在追蹤路徑訊息的訊息資料中累計代表追蹤路徑訊息所執行之活動的相關資訊。活動資訊儲存在活動 PCF 群組中。對於代表追蹤路徑訊息執行的每個活動，會將活動 PCF 群組寫入追蹤路徑訊息的訊息資料中 PCF 區塊的結尾。

其他活動資訊記錄在稱為 *TraceRoute* PCF 群組的 PCF 群組中追蹤路徑傳訊中。其他活動資訊儲存在此 PCF 群組中，可用來協助判斷所記錄活動的順序。此技術由 *TraceRoute* PCF 群組中的累計參數控制。

## 使用活動報告來記錄活動資訊

當透過佇列管理程式網路遞送追蹤路徑訊息時，可以針對代表追蹤路徑訊息執行的每個活動產生活動報告。活動資訊儲存在活動 PCF 群組中。對於代表追蹤路徑訊息執行的每個活動，會產生包含活動 PCF 群組的活動報告。追蹤路徑訊息的活動記錄的運作方式與任何其他訊息的運作方式相同。

針對追蹤路徑訊息產生的活動報告包含與針對任何其他訊息產生的活動資訊相比較的其他活動資訊。其他資訊會以 *TraceRoute* PCF 群組傳回。*TraceRoute* PCF 群組中包含的資訊僅在產生活動報告時才準確。您可以使用其他資訊來協助判斷代表追蹤路徑訊息所執行的活動順序。

## 正在獲得已記錄的活動資訊

當追蹤路徑訊息已到達其預期的目的地或被捨棄時，您用來獲得活動資訊的方法取決於該資訊的記錄方式。

## 開始之前

如果您不熟悉活動資訊，請參閱 [第 58 頁的『如何記錄活動資訊』](#)。

## 關於這項作業

在 trace-route 訊息已到達其預期的目的地或已捨棄之後，請使用下列方法來獲得活動資訊：

### 程序

- 擷取追蹤路徑訊息。

*TraceRoute* PCF 群組中的遞送參數可控制是否在到達時將追蹤路徑訊息放置在目標佇列上，或是否捨棄該訊息。如果追蹤路徑訊息遞送至目標佇列，您可以從這個佇列擷取追蹤路徑訊息。然後，您可以使用 IBM MQ 顯示路徑應用程式來顯示活動資訊。

若要要求在追蹤路徑訊息的訊息資料中累計活動資訊，請將 *TraceRoute* PCF 群組中的累計參數設為 `MQROUTE_ACCUMULATE_IN_MSG`。

- 使用追蹤路徑回覆訊息。

當追蹤路徑訊息到達其預期的目的地，或追蹤路徑訊息無法在佇列管理程式網路中進一步遞送時，可以產生追蹤路徑回覆訊息。追蹤路徑回覆訊息包含追蹤路徑訊息中所有活動資訊的副本，且會遞送至指定的回覆目的地佇列或系統佇列 `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`。您可以使用 IBM MQ 顯示路徑應用程式來顯示活動資訊。

若要要求追蹤路徑回覆訊息，請將 *TraceRoute* PCF 群組中的累計參數設為 `MQROUTE_ACCUMULATE_AND_REPLY`。

- 使用活動報告。

如果針對追蹤路徑訊息產生活動報告，您必須先找到活動報告，才能獲得活動資訊。然後，若要判定活動的順序，您必須排序活動報告。

## 控制追蹤路徑傳訊

在佇列管理程式層次啟用追蹤路徑傳訊，以便該佇列管理程式範圍內的應用程式可以將活動資訊寫入追蹤路徑訊息。若要啟用整個佇列管理程式網路，請個別啟用網路中的每個佇列管理程式，以進行追蹤路徑傳訊。如果您啟用更多佇列管理程式，則會產生更多活動報告。

## 開始之前

如果您使用活動報告來記錄追蹤路徑訊息的活動資訊，請參閱 [第 54 頁的『控制活動記錄』](#)。

## 關於這項作業

若要在透過佇列管理程式遞送追蹤路徑訊息時記錄其活動資訊，請執行下列步驟：

## 程序

- 定義如何記錄追蹤路徑訊息的活動資訊。  
請參閱 [第 61 頁的『產生及配置追蹤路徑訊息』](#)
- 如果您想要在追蹤路徑訊息中累計活動資訊，請確定佇列管理程式已啟用追蹤路徑傳訊
- 如果您想要在追蹤路徑訊息中累計活動資訊，請確定在追蹤路徑訊息上執行活動的應用程式能夠將活動資訊寫入追蹤路徑訊息的訊息資料

## 相關概念

[第 61 頁的『產生及配置追蹤路徑訊息』](#)

一種追蹤路由訊息包括特定訊息描述子和訊息資料部分。若要產生追蹤路徑訊息，請手動建立訊息，或使用 IBM MQ 顯示路徑應用程式。

## 相關工作

[第 54 頁的『控制活動記錄』](#)

在佇列管理程式層次啟用活動記錄。若要啟用整個佇列管理程式網路，請個別啟用網路中的每個佇列管理程式，以進行活動記錄。如果您啟用更多佇列管理程式，則會產生更多活動報告。

## 啟用佇列管理程式進行追蹤路徑傳訊

若要控制是否啟用或停用佇列管理程式進行追蹤路徑傳訊，請使用佇列管理程式屬性 ROUTEREC。

請使用 MQSC 指令 ALTER QMGR，並指定 ROUTEREC 參數來變更佇列管理程式屬性的值。此值可以是下列任一值：

## MSG

佇列管理程式已啟用追蹤路徑傳訊。佇列管理程式範圍內的應用程式可以將活動資訊寫入追蹤路徑訊息。

如果 *TraceRoute* PCF 群組中的 *累計* 參數設為 MQROUTE\_ACCUMULATE\_AND\_REPLY，且要對追蹤路徑訊息執行的下一個活動：

- 是捨棄
- 是本端佇列的放置 (目標佇列或無法傳送郵件的佇列)
- 將導致在 *trace-route* 訊息上執行的活動總數超出 *TraceRoute* PCF 群組中 *MaxActivities* 參數的值。

會產生追蹤路徑回覆訊息，並遞送至追蹤路徑訊息的訊息描述子中指定的回覆目的地佇列。

## 佇列

佇列管理程式已啟用追蹤路徑傳訊。佇列管理程式範圍內的應用程式可以將活動資訊寫入追蹤路徑訊息。

如果 *TraceRoute* PCF 群組中的 *累計* 參數設為 MQROUTE\_ACCUMULATE\_AND\_REPLY，且要對追蹤路徑訊息執行的下一個活動：

- 是捨棄
- 是本端佇列的放置 (目標佇列或無法傳送郵件的佇列)
- 將導致在 *trace-route* 訊息上執行的活動總數超出 *TraceRoute* PCF 群組中 *MaxActivities* 參數的值。

會產生追蹤路徑回覆訊息，並遞送至本端系統佇列 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE。

## 已停用

佇列管理程式已停用追蹤路徑傳訊。在追蹤路徑訊息中不會累計活動資訊，不過在此佇列管理程式的範圍內，可以更新 *TraceRoute PCF* 群組。

例如，若要停用佇列管理程式進行追蹤路徑傳訊，請使用下列 MQSC 指令：

```
ALTER QMGR ROUTEREC(DISABLED)
```

**記住：**當您修改 *ROUTEREC* 佇列管理程式屬性時，執行中 MCA 不會偵測變更，直到通道重新啟動為止。

啟用應用程式以進行追蹤路徑傳訊

若要啟用使用者應用程式的追蹤路徑傳訊，請根據訊息通道代理程式 (MCA) 所使用的演算法來執行演算法

## 開始之前

如果您不熟悉追蹤路徑訊息的格式，請參閱 [第 108 頁的『追蹤路徑訊息參照』](#)。

## 關於這項作業

已啟用訊息通道代理程式 (MCA) 進行追蹤路徑傳訊。若要啟用使用者應用程式進行追蹤路徑傳訊，請從 MCA 使用的演算法中使用下列步驟：

## 程序

1. 判斷所處理的訊息是否為追蹤路徑訊息。  
如果訊息不符合追蹤路徑訊息的格式，則不會將訊息當作追蹤路徑訊息來處理。
2. 判斷是否要記錄活動資訊。  
如果所執行活動的詳細程度不小於 *Detail* 參數指定的詳細程度，則在特定情況下會記錄活動資訊。只有在追蹤路徑訊息要求累積，且佇列管理程式已啟用追蹤路徑傳訊，或追蹤路徑訊息要求活動報告且佇列管理程式已啟用活動記錄時，才會記錄此資訊。
  - 如果要記錄活動資訊，請增加 *RecordedActivities* 參數。
  - 如果不要記錄活動資訊，請增加 *UnrecordedActivities* 參數。
3. 判斷在追蹤路徑訊息上執行的活動總數是否超出 *MaxActivities* 參數的值。  
活動總數是 *RecordedActivities*、*UnrecordedActivities* 及 *DiscontinuityCount* 的總和。  
如果活動總數超出 *MaxActivities*，請拒絕具有意見回饋 MQFB\_MAX\_ACTIVITIES 的訊息。
4. 如果累計的值設為 MQROUTE\_ACCUMULATE\_IN\_MSG 或 MQROUTE\_ACCUMULATE\_AND\_REPLY，且佇列管理程式已啟用追蹤路徑傳訊，請將活動 PCF 群組寫入追蹤路徑訊息的訊息資料中 PCF 區塊的結尾。
5. 將追蹤路徑訊息遞送至本端佇列。
  - 如果參數 *Deliver* 指定為 MQROUTE\_DELIVER\_NO，則會拒絕具有回饋 MQFB\_NOT\_DELIVERED 的追蹤路徑訊息。
  - 如果參數遞送指定為 MQROUTE\_DELIVER\_YES，請將追蹤路徑訊息遞送至本端佇列。
6. 如果符合下列所有條件，則產生追蹤路徑回覆訊息：
  - 追蹤路徑訊息已遞送至本端佇列或被拒絕
  - 參數累計的值為 MQROUTE\_ACCUMULATE\_AND\_REPLY
  - 佇列管理程式已啟用追蹤路徑傳訊追蹤路徑回覆訊息放置在 ROUTEREC 佇列管理程式屬性所決定的佇列上。
7. 如果追蹤路徑訊息要求活動報告，且佇列管理程式已啟用活動記錄，請產生活動報告。  
活動報告會放置在由 ACTIVREC 佇列管理程式屬性所決定的佇列上。

## 產生及配置追蹤路徑訊息

一種追蹤路由訊息包括特定訊息描述子和訊息資料部分。若要產生追蹤路徑訊息，請手動建立訊息，或使用 IBM MQ 顯示路徑應用程式。

追蹤路徑訊息由下列部分組成：

### 訊息描述子

MQMD 結構，其格式欄位設為 MQFMT\_ADMIN 或 MQFMT\_embedded\_pcf。

### 訊息資料

下列其中一個組合：

- 如果格式設為 MQFMT\_ADMIN，則為 PCF 標頭 (MQCFH) 及追蹤路徑訊息資料
- 內嵌 PCF 標頭 (MQEPH)、追蹤路徑訊息資料及其他使用者指定的訊息資料 (如果格式設為 MQFMT\_EMBEDDED\_PCF)

追蹤路徑訊息資料由 *TraceRoute* PCF 群組和一或多個活動 PCF 群組組成。

## 手動產生

手動產生追蹤路徑訊息時，不需要活動 PCF 群組。當 MCA 或使用者撰寫的應用程式代表其執行活動時，會將活動 PCF 群組寫入追蹤路徑訊息的訊息資料。

## IBM MQ 顯示路徑應用程式

使用 IBM MQ 顯示路徑應用程式 `dspmqrte` 來配置、產生追蹤路徑訊息，並將其放入佇列管理程式網路中。將訊息描述子中的格式參數設為 MQFMT\_ADMIN。您無法將使用者資料新增至 IBM MQ 顯示路徑應用程式所產生的追蹤路徑訊息。

**限制：**無法在 IBM WebSphere MQ 6.0 之前的佇列管理程式或 IBM MQ for z/OS 佇列管理程式上發出 `dspmqrte`。如果您想要第一個佇列管理程式將追蹤路徑訊息遞送至此類型的佇列管理程式，請使用選用參數 `-c` 連接至佇列管理程式作為 IBM WebSphere MQ 6.0 或更新版本的用戶端。

### 模擬原始訊息

當使用追蹤路徑訊息來判斷另一個訊息已透過佇列管理程式網路取得的路徑時，追蹤路徑訊息越接近模擬原始訊息，追蹤路徑訊息遵循與原始訊息相同的路徑的機會就越大。

下列訊息性質可能會影響訊息在佇列管理程式網路內轉遞至的位置：

### 優先順序

可以在訊息的訊息描述子中指定優先順序。

### 持續性

您可以在訊息的訊息描述子中指定持續性。

### 期限

可以在訊息的訊息描述子中指定有效期限。

### 報告選項

報告選項可以在訊息的訊息描述子中指定。

### 訊息大小

若要模擬訊息的大小，可以將其他資料寫入訊息的訊息資料。基於此目的，其他訊息資料可能沒有意義。

**提示：**IBM MQ 顯示路徑應用程式無法指定訊息大小。

### 訊息資料

部分佇列管理程式網路會使用內容型遞送來決定轉遞訊息的位置。在這些情況下，需要寫入追蹤路徑訊息的訊息資料，以模擬原始訊息的訊息資料。

**提示：**IBM MQ 顯示路徑應用程式無法指定訊息資料。

### *TraceRoute* PCF 群組

*TraceRoute* PCF 群組中的屬性會控制追蹤路徑訊息的行為。*TraceRoute* PCF 群組位於每個追蹤路徑訊息的訊息資料中。

下表列出 MCA 辨識之 *TraceRoute* 群組中的參數。如果撰寫使用者撰寫的應用程式來辨識它們，則可以新增進一步參數，如第 66 頁的『其他活動資訊』中所述。

參數	類型
TraceRoute	MQCFGR
明細	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
累計	MQCFIN
向前轉	MQCFIN
交付	MQCFIN

*TraceRoute* PCF 群組中每一個參數的說明如下：

#### 明細

指定要記錄之活動資訊的詳細層次。此值可以是下列任一值：

##### **MQRROUTE\_DETAIL\_LOW**

只會記錄使用者應用程式所執行的活動。

##### **MQRROUTE\_DETAIL\_MEDIUM**

應該記錄 MQRROUTE\_DETAIL\_LOW 中指定的活動。此外，還會記錄 MCA 所執行的活動。

##### **MQRROUTE\_DETAIL\_HIGH**

應該記錄 MQRROUTE\_DETAIL\_LOW 和 MQRROUTE\_DETAIL\_MEDIUM 中指定的活動。MCA 不會在此詳細程度上記錄任何進一步的活動資訊。只有要記錄進一步活動資訊的使用者應用程式才能使用此選項。例如，如果使用者應用程式透過考量特定訊息性質來決定訊息所採用的路徑，則遞送邏輯的相關資訊可以包含在此詳細程度中。

#### **RecordedActivities**

指定代表追蹤路徑訊息執行的記錄活動數。如果活動的相關資訊已寫入追蹤路徑訊息，或已產生活動報告，則會將活動視為已記錄。對於每個記錄的活動，*RecordedActivities* 會增量 1。

#### **UnrecordedActivities**

指定代表追蹤路徑訊息執行的未記錄活動數。如果啟用追蹤路徑傳訊的應用程式既未累計，也未將相關活動資訊寫入活動報告，則會將活動視為未記錄。

在下列情況下，不會記錄代表追蹤路徑訊息執行的活動：

- 所執行活動的明細層次小於明細參數指定的明細層次。
- 追蹤路徑訊息會要求活動報告，但不會累積，且不會啟用佇列管理程式進行活動記錄。
- 追蹤路徑訊息要求累積，但未要求活動報告，且佇列管理程式未啟用追蹤路徑傳訊。
- 追蹤路徑訊息同時要求累積及活動報告，且佇列管理程式未啟用活動記錄及追蹤路徑傳訊。
- 追蹤路徑訊息既不要求累計，也不要求活動報告。

對於每個未記錄的活動，參數 *UnrecordedActivities* 會以 1 為增量。

#### **DiscontinuityCount**

指定已透過佇列管理程式遞送追蹤路徑訊息的次數，該佇列管理程式具有未啟用追蹤路徑傳訊的應用程式。佇列管理程式會增加此值。如果此值大於 0，則只能判定部分訊息路徑。

#### **MaxActivities**

指定可代表追蹤路徑訊息執行的活動數目上限。

活動總數是 *RecordedActivities*、*UnrecordedActivities* 及 *DiscontinuityCount* 的總和。活動總數不得超過 *MaxActivities* 的值。

*MaxActivities* 的值可以是：

## 正整數

活動數目上限。

如果超出活動數目上限，則會拒絕追蹤路徑訊息，並提供意見 MQFB\_MAX\_ACTIVITIES。如果捕捉到無限迴圈，這可以防止追蹤路徑訊息無限期轉遞。

## MQROUTE\_UNLIMITED\_ACTIVITIES

可以代表追蹤路徑訊息執行無限制數量的活動。

## 累計

指定用來累計活動資訊的方法。此值可以是下列任一值：

### MQROUTE\_ACCUMULATE\_IN\_MSG

如果啟用佇列管理程式進行追蹤路徑傳訊，則會在追蹤路徑訊息的訊息資料中累計活動資訊。

如果指定此值，則追蹤路徑訊息資料由下列項目組成：

- *TraceRoute* PCF 群組。
- 零個以上 活動 PCF 群組。

### MQROUTE\_ACCUMULATE\_AND\_REPLY

如果啟用佇列管理程式進行追蹤路徑傳訊，則會在追蹤路徑訊息的訊息資料中累計活動資訊，並在發生下列任何情況時產生追蹤路徑回覆訊息：

- IBM WebSphere MQ 6 (或更新版本) 佇列管理程式會捨棄追蹤路徑訊息。
- 「IBM WebSphere MQ 6」(或更新版本) 佇列管理程式會將追蹤路徑訊息放入本端佇列 (目標佇列或無法傳送郵件的佇列)。
- 在追蹤路徑訊息上執行的活動數超出 *MaxActivities* 的值。

如果指定此值，則追蹤路徑訊息資料由下列項目組成：

- *TraceRoute* PCF 群組。
- 零個以上 活動 PCF 群組。

### MQROUTE\_ACCUMULATE\_NONE

活動資訊不會累積在追蹤路徑訊息的訊息資料中。

如果指定此值，則追蹤路徑訊息資料由下列項目組成：

- *TraceRoute* PCF 群組。

## 向前轉

指定追蹤路徑訊息可以轉遞至何處。值可以為：

### MQROUTE\_FORWARD\_IF\_SUPPORTED

追蹤路徑訊息只會轉遞至將允許使用 *TraceRoute* 群組中 遞送 參數值的佇列管理程式。

### MQROUTE\_FORWARD\_ALL

不論是否允許使用 遞送 參數的值，都會將追蹤路徑訊息轉遞至任何佇列管理程式。

在決定是否將追蹤路徑訊息轉遞至遠端佇列管理程式時，佇列管理程式會使用下列演算法：

1. 判斷遠端佇列管理程式是否能夠支援追蹤路徑傳訊。
  - 如果遠端佇列管理程式能夠支援追蹤路徑傳訊，演算法會繼續執行步驟 [第 64 頁的『4』](#)。
  - 如果遠端佇列管理程式無法支援追蹤路徑傳訊，演算法會繼續執行步驟 [第 63 頁的『2』](#)。
2. 判定 *TraceRoute* 群組中的 遞送 參數是否包含 MQROUTE\_DELIVER\_REJ\_UNSUPP\_MASK 位元遮罩中任何無法辨識的遞送選項。
  - 如果找到任何無法辨識的遞送選項，則會使用意見 MQFB\_UNSUPPORTED\_DELIVERY 來拒絕追蹤路徑訊息。
  - 如果找不到無法辨識的遞送選項，演算法會繼續執行步驟 [第 63 頁的『3』](#)。
3. 從追蹤路徑訊息中的 *TraceRoute* PCF 群組決定 遞送 參數值。
  - 如果 遞送 指定為 MQROUTE\_DELIVER\_YES，則會將追蹤路徑訊息轉遞至遠端佇列管理程式。

- 如果遞送指定為 MQROUTE\_DELIVER\_NO，則演算法會繼續執行步驟 第 64 頁的『4』。
4. 判定 *TraceRoute* 群組中的 *Forward* 參數是否包含 MQROUTE\_FORWARDING\_REJ\_UNSUPP\_MASK 位元遮罩中任何無法辨識的轉遞選項。
    - 如果找到任何無法辨識的轉遞選項，則會使用意見 MQFB\_UNSUPPORTED\_FORWARDING 來拒絕 trace-route 訊息。
    - 如果找不到無法辨識的轉遞選項，演算法會繼續執行步驟 第 64 頁的『5』。
  5. 從追蹤路徑訊息中的 *TraceRoute* PCF 群組，決定轉遞參數值。
    - 如果將轉遞指定為 MQROUTE\_FORWARD\_IF\_SUPPORTED，則會拒絕追蹤路徑訊息，並回應 MQFB\_NOT\_FORWARDED。
    - 如果轉遞指定為 MQROUTE\_FORWARD\_ALL，則追蹤路徑訊息可以轉遞至遠端佇列管理程式。

## 交付

指定追蹤路徑訊息到達其預期目的地時要採取的動作。使用者撰寫的應用程式必須先檢查此屬性，然後再將追蹤路徑訊息放置在其目標佇列上。此值可以是下列任一值：

### MQROUTE\_DELIVER\_YES

到達時，追蹤路徑訊息會放置在目標佇列上。對目標佇列執行 *get* 作業的任何應用程式都可以擷取追蹤路徑訊息。

### MQROUTE\_DELIVER\_NO

到達時，追蹤路徑訊息不會遞送至目標佇列。訊息會根據其報告選項來處理。

## 設定追蹤路徑回覆訊息的共用佇列

當報告遞送至本端系統佇列時，如果要判斷與特定訊息相關的追蹤路徑回覆訊息的位置，在單一節點上使用共用佇列會更有效率

## 開始之前

設定 **ROUTEREC** 參數，以啟用佇列管理程式進行追蹤路徑傳訊，並指定將產生的任何追蹤路徑回覆訊息遞送至本端系統佇列 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE。

## 關於這項作業

如果佇列管理程式網路中的佇列管理程式數目設定為將追蹤路徑回覆訊息遞送至本端系統佇列，則判斷與特定訊息相關的追蹤路徑回覆訊息位置可能很費時。或者，使用單一節點，它是管理一般佇列的佇列管理程式。佇列管理程式網路中的所有佇列管理程式都可以將追蹤路徑回覆訊息遞送至此共用佇列。使用一般佇列的好處是佇列管理程式不需要將追蹤路徑回覆訊息遞送至訊息中指定的回覆目的地佇列，而且在判斷與訊息相關的追蹤路徑回覆訊息位置時，您只會查詢一個佇列。

若要設定共用佇列，請執行下列步驟：

## 程序

1. 選取或定義佇列管理程式作為單一節點
2. 在單一節點上，選取或定義要作為共用佇列的佇列
3. 在將追蹤路徑回覆訊息轉遞至共用佇列的所有佇列管理程式上，重新定義本端系統佇列 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE 作為遠端佇列定義
  - a) 指定單一節點的名稱作為遠端佇列管理程式名稱
  - b) 指定共用佇列的名稱作為遠端佇列名稱

## 取得及使用記錄的資訊

使用下列任何技術來獲得追蹤路徑訊息的已記錄活動資訊

請注意，未獲得活動資訊的情況也適用於追蹤路徑回覆訊息。

當同時停用活動記錄和追蹤路徑傳訊的佇列管理程式處理追蹤路徑訊息時，不會記錄活動資訊。

從追蹤路徑回覆訊息取得資訊

若要獲得活動資訊，請尋找追蹤路徑回覆訊息。然後擷取訊息並分析活動資訊。

## 關於這項作業

只有在您知道追蹤路徑回覆訊息的位置時，才能從追蹤路徑回覆訊息獲得活動資訊。尋找訊息並處理活動資訊，如下所示：

### 程序

1. 請檢查在追蹤路徑訊息的訊息描述子中指定的回覆目的地佇列。如果追蹤路徑回覆訊息不在回覆目的地佇列上，請檢查下列位置：
  - 本端系統佇列 `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`，在追蹤路徑訊息的目標佇列管理程式上
  - 共用佇列 (如果您已設定追蹤路徑回覆訊息的共用佇列)
  - 本端系統佇列 `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`，在佇列管理程式網路中的任何其他佇列管理程式上，如果追蹤路徑訊息已放入無法傳送郵件的佇列，或已超出活動數目上限，則可能會發生此情況
2. 擷取追蹤路徑回覆訊息
3. 使用 IBM MQ 顯示路徑應用程式來顯示記錄的活動資訊
4. 研究活動資訊並取得您需要的資訊

#### 從追蹤路徑訊息獲得資訊

若要獲得活動資訊，您可以尋找追蹤路徑訊息，它必須在 *TraceRoute* PCF 群組中具有適當的參數。然後擷取訊息並分析活動資訊。

## 關於這項作業

只有在您知道追蹤路徑訊息的位置，且它在指定為 `MQRROUTE_ACCUMULATE_IN_MSG` 或 `MQRROUTE_ACCUMULATE_AND_REPLY` 的 *TraceRoute* PCF 群組中具有參數 `累計` 時，才能從追蹤路徑訊息獲得活動資訊。

若要將追蹤路徑訊息遞送至目標佇列，*TraceRoute* PCF 群組中的 `遞送` 參數必須指定為 `MQRROUTE_DELIVER_YES`。

### 程序

1. 請檢查目標佇列。如果追蹤路徑訊息不在目標佇列上，您可以嘗試使用啟用活動記錄的追蹤路徑訊息來尋找追蹤路徑訊息。使用產生的活動報告，嘗試判斷追蹤路徑訊息的最後一個已知位置。
2. 擷取追蹤路徑訊息
3. 使用 IBM MQ 顯示路徑應用程式來顯示記錄的活動資訊
4. 研究活動資訊並取得您需要的資訊

#### 從活動報告獲得資訊

若要獲得活動資訊，您可以尋找活動報告，其必須在訊息描述子中指定報告選項。然後擷取活動報告並分析活動資訊。

## 關於這項作業

只有在您知道活動報告的位置，且追蹤路徑訊息的訊息描述子中指定了報告選項 `MQRO_ACTIVITY` 時，才能從活動報告獲得活動資訊。

### 程序

1. 尋找並排序針對追蹤路徑訊息所產生的活動報告。  
當您找到活動報告時，您可以手動排序它們，或使用 IBM MQ 顯示路徑應用程式來自動排序及顯示活動資訊。
2. 研究活動資訊並取得您需要的資訊

## 其他活動資訊

當透過佇列管理程式網路遞送追蹤路徑訊息時，使用者應用程式可以透過在將活動群組寫入追蹤路徑訊息或活動報告的訊息資料時併入一個以上其他 PCF 參數，來記錄其他資訊。

其他活動資訊可協助系統管理者識別追蹤路徑訊息所採用的路徑，或為何採用該路徑。

如果您使用 IBM MQ 顯示路徑應用程式來顯示追蹤路徑訊息的記錄資訊，除非 IBM MQ 顯示路徑應用程式可辨識每一個參數的參數 ID，否則任何其他 PCF 參數只能以數值 ID 顯示。若要辨識參數 ID，必須使用下列 PCF 參數來記錄其他資訊。在活動 PCF 群組中適當的位置包含這些 PCF 參數。

### GroupName

表 13: 群組名稱	
說明	指定其他資訊的分組參數。
ID	MQGACF_VALUE_NAMING。
資料類型	MQCFGR
群組中的參數	<i>ParameterName</i> <i>ParameterValue</i>

### ParameterName

表 14: 參數名稱	
說明	包含要由 IBM MQ 顯示路徑應用程式顯示的名稱，其會將 <i>ParameterValue</i> 的值放入環境定義中。
ID	MQCA_VALUE_NAME。
資料類型	MQCFST
包含在 PCF 群組中:	<i>GroupName</i> 。
值:	要顯示的名稱。

### ParameterValue

表 15: 參數值	
說明	包含要由 IBM MQ 顯示路徑應用程式顯示的值。
ID:	其他資訊的 PCF 結構 ID。
資料類型:	其他資訊的 PCF 結構資料類型。
包含在 PCF 群組中:	<i>GroupName</i> 。
值:	要顯示的值。

## 記錄其他活動資訊的範例

下列範例說明當代表追蹤路徑訊息執行活動時，使用者應用程式如何記錄其他資訊。在這兩個範例中，都會使用 IBM MQ 顯示路徑應用程式來產生追蹤路徑訊息，並顯示傳回給它的活動資訊。

### 記錄其他活動資訊: 範例 1

使用者應用程式會以 IBM MQ 顯示路徑應用程式無法辨識參數 ID 的格式來記錄其他活動資訊。

1. IBM MQ 顯示路徑應用程式可用來產生追蹤路徑訊息，並將其放入佇列管理程式網路中。必要的選項設定為要求下列各項：
  - 活動資訊會累計在追蹤路徑訊息的訊息資料中。

- 到達目標佇列時，會捨棄追蹤路徑訊息，並產生追蹤路徑回覆訊息，並遞送至指定的回覆目的地佇列。
- 收到追蹤路徑回覆訊息時，IBM MQ 顯示路徑應用程式會顯示累計的活動資訊。

追蹤路徑訊息會放入佇列管理程式網路中。

2. 當透過佇列管理程式網路遞送追蹤路徑訊息時，啟用追蹤路徑傳訊的使用者應用程式會代表訊息執行低詳細資料活動。除了將標準活動資訊寫入追蹤路徑訊息之外，使用者應用程式還會將下列 PCF 參數寫入「活動」群組結尾：

#### ColorValue

##### ID

65536

##### 資料類型

MQCFST

##### 值

'紅色'

這個額外的 PCF 參數提供已執行之活動的進一步相關資訊，不過它是以 IBM MQ 顯示路徑應用程式無法辨識參數 ID 的格式撰寫。

3. 追蹤路徑訊息會到達目標佇列，並將追蹤路徑回覆訊息傳回 IBM MQ 顯示路徑應用程式。其他活動資訊顯示如下：

```
65536: 'Red'
```

IBM MQ 顯示路徑應用程式無法辨識 PCF 參數的參數 ID，並將其顯示為數值。其他資訊的環境定義不明確。

如需 IBM MQ 顯示路徑應用程式辨識 PCF 參數的參數 ID 時的範例，請參閱第 67 頁的『記錄其他活動資訊: 範例 2』。

#### 記錄其他活動資訊: 範例 2

使用者應用程式會以 IBM MQ 顯示路徑應用程式可辨識參數 ID 的格式來記錄其他活動資訊。

1. IBM MQ 顯示畫面遞送應用程式可用來產生追蹤遞送訊息，並以第 66 頁的『記錄其他活動資訊: 範例 1』中的相同方式將其放置到佇列管理程式網路中。
2. 當透過佇列管理程式網路遞送追蹤路徑訊息時，啟用追蹤路徑傳訊的使用者應用程式會代表訊息執行低詳細資料活動。除了將標準活動資訊寫入追蹤路徑訊息之外，使用者應用程式還會將下列 PCF 參數寫入「活動」群組結尾：

#### ColorInfo

表 16: 顏色資訊	
說明	指定顏色相關資訊的分組參數。
ID:	MQGACF_VALUE_NAMING。
資料類型:	MQCFGR。
群組中的參數:	<i>ColorName</i> <i>ColorValue</i>

#### ColorName

表 17: 顏色名稱	
說明	包含要由 IBM MQ 顯示路徑應用程式顯示的名稱，該應用程式會將 <i>ColorValue</i> 的值放入環境定義中。
ID:	MQCA_VALUE_NAME。
資料類型:	MQCFST。

表 17: 顏色名稱 (繼續)	
說明	包含要由 <b>IBM MQ 顯示路徑應用程式</b> 顯示的名稱，該應用程式會將 <b>ColorValue</b> 的值放入環境定義中。
包含在 PCF 群組中:	<i>ColorInfo</i> .
值:	'顏色'

### ColorValue

表 18: 色彩值	
說明	包含要由 <b>IBM MQ 顯示路徑應用程式</b> 顯示的值。
ID:	65536.
資料類型:	MQCFST.
包含在 PCF 群組中:	<i>ColorInfo</i> .
值:	'紅色'

這些其他 PCF 參數提供所執行活動的進一步相關資訊。這些 PCF 參數以 IBM MQ 顯示路徑應用程式可辨識參數 ID 的格式撰寫。

- 追蹤路徑訊息會到達目標佇列，並將追蹤路徑回覆訊息傳回 IBM MQ 顯示路徑應用程式。其他活動資訊顯示如下：

```
Color: 'Red'
```

IBM MQ 顯示路徑應用程式可辨識包含其他活動資訊值之 PCF 結構的參數 ID 具有對應的名稱。會顯示對應的名稱，而不是數值。

## IBM MQ 顯示路徑應用程式

使用 IBM MQ 顯示路徑應用程式 (**dspmqrte**) 使用指令行介面來使用追蹤路徑訊息及與追蹤路徑訊息相關的活動資訊。  IBM MQ 顯示路徑應用程式未隨附於 IBM MQ for z/OS，但您可以在發出 **dspmqrte** 指令時指定 **-c** 參數，從分散式安裝架構執行它，並以用戶端身分將它連接至 IBM MQ for z/OS 佇列管理程式。

您可以基於下列目的使用 IBM MQ **dspmqrte** 顯示路徑應用程式：

- 將追蹤路徑訊息配置、產生及放入佇列管理程式網路中。  
透過將追蹤路徑訊息放入佇列管理程式網路，可以收集活動資訊，並使用活動資訊來判定追蹤路徑訊息所採用的路徑。您可以指定追蹤路徑訊息的性質，如下所示：
  - 追蹤路徑訊息的目的地。
  - 追蹤路徑訊息如何模擬另一個訊息。
  - 在透過佇列管理程式網路遞送追蹤路徑訊息時，應如何處理追蹤路徑訊息。
  - 是否使用活動記錄或追蹤路徑傳訊來記錄活動資訊。
- 訂購並顯示與追蹤路徑訊息相關的活動資訊。

如果 IBM MQ 顯示路徑應用程式已將追蹤路徑訊息放入佇列管理程式網路，則在傳回相關活動資訊之後，可以立即訂購並顯示該資訊。或者，IBM MQ 顯示遞送應用程式可用來排序及顯示與先前產生的追蹤遞送訊息相關的活動資訊。

### 相關參考

[dspmqrte](#)

## 追蹤路徑訊息的參數

請利用這個頁面來取得 IBM MQ 顯示路徑應用程式 **dspmqrte** 所提供的參數概觀，以判斷追蹤路徑訊息的性質，包括如何將它當作透過佇列管理程式網路來遞送。

### 相關參考

[dspmqrte](#)

### 佇列管理程式連線

請利用這個頁面來指定 IBM MQ 顯示路徑應用程式所連接的佇列管理程式

#### -c

指定 IBM MQ 顯示路徑應用程式連接為用戶端應用程式。

如果您未指定此參數，則 IBM MQ 顯示路徑應用程式不會作為用戶端應用程式連接。

#### -m *QMgrName*

IBM MQ 顯示畫面遞送應用程式所連接的佇列管理程式名稱。名稱最多可以包含 48 個字元。

如果您未指定此參數，則會使用預設佇列管理程式。

### 目標目的地

請利用這個頁面來指定追蹤路徑訊息的目標目的地

#### -q *TargetQName*

如果使用 IBM MQ 顯示路徑應用程式將追蹤路徑訊息傳送至佇列管理程式網路，則 *TargetQName* 會指定目標佇列的名稱。

#### -ts *TargetTopic* 字串

指定主題字串。

#### -qm *TargetQMGr*

限定目標目的地；然後將套用一般佇列管理程式名稱解析。目標目的地以 *-q TargetQName* 或 *-ts TargetTopicString* 指定。

如果未指定此參數，則會使用 IBM MQ 顯示路徑應用程式所連接的佇列管理程式作為目標佇列管理程式。

#### -o

指定目標目的地未連結至特定目的地。一般而言，當要在叢集中放置追蹤路徑訊息時，會使用這個參數。使用選項 `MQOO_BIND_NOT_FIXED` 開啟目標目的地。

如果未指定此參數，則目標目的地會連結至特定目的地。

### 發佈主題

對於發佈/訂閱應用程式，請利用這個頁面來指定要發佈之 IBM MQ 顯示路徑應用程式的追蹤路徑訊息主題字串

#### -ts *TopicName*

指定 IBM MQ 顯示路徑應用程式要發佈追蹤路徑訊息的主題字串，並將此應用程式置於主題模式。在此模式中，應用程式會追蹤發佈要求所產生的所有訊息。

您也可以使用 IBM MQ 顯示路徑應用程式，來顯示針對發佈訊息所產生之活動報告的結果。

### 訊息模擬

請利用這個頁面來配置追蹤路徑訊息以模擬訊息，例如當原始訊息未到達其預期的目的地時

追蹤路徑傳訊的其中一個用途是協助判斷未到達其預期目的地之訊息的最後一個已知位置。IBM MQ 顯示路徑應用程式提供參數，可協助配置追蹤路徑訊息以模擬原始訊息。當模擬訊息時，您可以使用下列參數：

#### -l 持續性

指定所產生追蹤路徑訊息的持續性。持續性的可能值如下：

##### yes

產生的追蹤路徑訊息是持續的。(MQPER\_PERSISTENT)。

##### no

產生的追蹤路徑訊息不持續。(MQPER\_NOT\_PERSISTENT)。

## q

產生的追蹤路徑訊息會從 `-q TargetQName` 或 `-ts TargetTopicString` 指定的目的地繼承其持續性值。(MQPER\_PERSISTENCE\_AS\_Q\_DEF)。

所傳回的追蹤路徑回覆訊息或任何報告訊息將與原始追蹤路徑訊息共用相同的持續性值。

如果 持續性 指定為 **yes**，您必須指定 `-rq ReplyToQ` 參數。回覆目的地佇列不得解析為暫時動態佇列。

如果您未指定此參數，則產生的追蹤路徑訊息 **不會** 持續。

## -p 優先順序

指定追蹤路徑訊息的優先順序。Priority 的值大於或等於 0，或 MQPRI\_PRIORITY\_AS\_Q\_DEF。MQPRI\_PRIORITY\_AS\_Q\_DEF 會指定從 `-q TargetQName` 或 `-ts TargetTopicString` 指定的目的地取得優先順序值。

如果未指定此參數，則會從 `-q TargetQName` 或 `-ts TargetTopicString` 指定的目的地取得優先順序值。

## -xs 期限

指定追蹤路徑訊息的到期時間 (以秒為單位)。

如果未指定此參數，則到期時間會指定為 60 秒。

## -ro none |ReportOption

### none

指定不設定報告選項。

### ReportOption

指定追蹤路徑訊息的報告選項。可以使用逗點作為分隔字元來指定多個報告選項。ReportOption 的可能值如下：

#### 活動

已設定報告選項 MQRO\_ACTIVITY。

#### 抵達確認

已設定報告選項 MQRO\_COA\_WITH\_FULL\_DATA。

#### COD

已設定報告選項 MQRO\_COD\_WITH\_FULL\_DATA。

#### 異常狀況

已設定報告選項 MQRO\_EXCEPTION\_WITH\_FULL\_DATA。

#### expiration

已設定報告選項 MQRO\_EXPIRATION\_WITH\_FULL\_DATA。

#### 捨棄

已設定報告選項 MQRO\_DISCARD\_MSG。

如果既未指定 `-ro ReportOption` 也未指定 `-ro none`，則會指定 MQRO\_ACTIVITY 及 MQRO\_DISCARD\_MSG 報告選項。

IBM MQ 顯示遞送應用程式不容許您將使用者資料新增至追蹤遞送訊息。如果您需要將使用者資料新增至追蹤路徑訊息，則必須手動產生追蹤路徑訊息。

## 已記錄活動資訊

請利用這個頁面來指定用來傳回記錄的活動資訊的方法，然後您可以使用這個方法來判斷追蹤路徑訊息所採用的路徑

記錄的活動資訊可以如下所示傳回：

- 在活動報告中
- 在追蹤路徑回覆訊息中
- 在追蹤路徑訊息本身 (已放在目標佇列上)

使用 **dspmqrte** 時，會使用下列參數來決定用來傳回記錄的活動資訊的方法：

## -ro activity

指定使用活動報告傳回活動資訊。依預設，會啟用活動記錄。

**-ac-ar**

指定在追蹤路徑訊息中累計活動資訊，並產生追蹤路徑回覆訊息。

**-ac**

指定在追蹤路徑訊息內累計活動資訊。

如果您未指定此參數，則不會在追蹤路徑訊息內累計活動資訊。

**-阿爾**

要求在下列情況下產生包含所有累計活動資訊的追蹤路徑回覆訊息：

- IBM MQ 佇列管理程式會捨棄追蹤路徑訊息。
- 追蹤路徑訊息由 IBM MQ 佇列管理程式放入本端佇列 (目標佇列或無法傳送郵件的佇列)。
- 在追蹤路徑訊息上執行的活動數目超出 *-s Activities* 中指定的值。

**-ac -d yes**

指定在追蹤路徑訊息中累計活動資訊，並在到達時將追蹤路徑訊息放置在目標佇列上。

**-ac**

指定在追蹤路徑訊息內累計活動資訊。

如果您未指定此參數，則不會在追蹤路徑訊息內累計活動資訊。

**-d yes**

到達時，即使佇列管理程式不支援追蹤路徑傳訊，也會將追蹤路徑訊息放入目標佇列。

如果您未指定此參數，則不會將追蹤路徑訊息放置到目標佇列。

然後可以從目標佇列擷取追蹤路徑訊息，並取得記錄的活動資訊。

您可以視需要結合這些方法。

此外，可以使用下列參數來指定所記錄活動資訊的詳細層次：

**-t 明細**

指定記錄的活動。詳細資料的可能值如下：

**低**

僅記錄使用者定義應用程式所執行的活動。

**中型**

記錄以低值指定的活動。此外，還會記錄 MCA 所執行的發佈活動和活動。

**high**

會記錄以低及中格式指定的活動。MCA 不會在此詳細程度上公開任何進一步的活動資訊。此選項僅適用於要公開進一步活動資訊的使用者定義應用程式。例如，如果使用者定義的應用程式透過考量特定訊息性質來決定訊息所採用的路徑，則遞送邏輯可以併入此詳細程度。

如果您未指定此參數，則會記錄中等層次活動。

依預設，IBM MQ 顯示路徑應用程式會使用暫時動態佇列來儲存傳回的訊息。當 IBM MQ 顯示路徑應用程式結束時，會關閉暫時動態佇列，並清除任何訊息。如果在現行執行 IBM MQ 顯示路徑應用程式結束之後需要傳回的訊息，則必須使用下列參數來指定永久佇列：

**-rq ReplyToQ**

指定追蹤路徑訊息的所有回應將傳送至其中的回覆目的地佇列名稱。如果追蹤路徑訊息持續存在，或指定 *-n* 參數，則必須指定不是暫時動態佇列的回覆目的地佇列。

如果您未指定此參數，則會使用系統預設模型佇列 SYSTEM.DEFAULT.MODEL.QUEUE。

**-rqm ReplyTo 佇列管理程式**

指定回覆目的地佇列所在的佇列管理程式名稱。名稱最多可以包含 48 個字元。

如果您未指定此參數，則會使用 IBM MQ 顯示畫面遞送應用程式所連接的佇列管理程式作為回覆目的地佇列管理程式。

**如何處理追蹤路徑訊息**

請利用這個頁面來控制如何處理透過佇列管理程式網路遞送的追蹤路徑訊息。

下列參數可以限制在佇列管理程式網路中可以遞送追蹤路徑訊息的位置:

#### **-d 遞送**

指定是否在到達時將追蹤路徑訊息遞送至目標佇列。遞送的可能值如下:

##### **yes**

到達時, 即使佇列管理程式不支援追蹤路徑傳訊, 也會將追蹤路徑訊息放入目標佇列。

##### **no**

到達時, 追蹤路徑訊息不會放入目標佇列。

如果您未指定此參數, 則不會將追蹤路徑訊息放置到目標佇列。

#### **-f 轉遞**

指定追蹤路徑訊息可轉遞至其中的佇列管理程式類型。如需佇列管理程式用來決定是否將訊息轉遞至遠端佇列管理程式的演算法詳細資料, 請參閱第 61 頁的『TraceRoute PCF 群組』。向前的可能值如下:

##### **all**

追蹤路徑訊息會轉遞至任何佇列管理程式。

**警告:** 如果轉遞至早於 IBM WebSphere MQ 6.0 的佇列管理程式, 則追蹤路徑訊息將無法辨識, 而且即使 *-d Deliver* 參數值也可以遞送至本端佇列。

##### **supported**

追蹤路徑訊息只會轉遞至將允許使用 *TraceRoute PCF* 群組中遞送參數的佇列管理程式。

如果您未指定此參數, 則追蹤路徑訊息只會轉遞至允許使用遞送參數的佇列管理程式。

下列參數可防止追蹤路徑訊息無限期留在佇列管理程式網路中:

#### **-s 活動**

指定在捨棄追蹤路徑訊息之前, 可以代表追蹤路徑訊息執行的已記錄活動數目上限。如果捕捉到無限迴圈, 這會防止追蹤路徑訊息無限期轉遞。活動的值大於或等於 1, 或 `MQROUTE_UNLIMITED_ACTIVITIES`。 `MQROUTE_UNLIMITED_ACTIVITIES` 指定可以代表 *trace-route* 訊息執行無限制數量的活動。

如果您未指定此參數, 則可以代表追蹤遞送訊息執行無限制數量的活動。

#### **-xs 期限**

指定追蹤路徑訊息的到期時間 (以秒為單位)。

如果未指定此參數, 則到期時間會指定為 60 秒。

#### **-xp PassExpiry**

指定是否將追蹤路徑訊息中的到期時間傳遞至追蹤路徑回覆訊息。 *PassExpiry* 的可能值如下:

##### **yes**

報告選項 `MQRO_PASS_DISCARD_AND_EXPIRY` 指定在追蹤路徑訊息的訊息描述子中。

如果針對追蹤路徑訊息產生追蹤路徑回覆訊息或活動報告, 則會產生 `MQRO_DISCARD` 報告選項 (如果已指定), 並傳遞剩餘到期時間。

這是預設值。

##### **no**

未指定報告選項 `MQRO_PASS_DISCARD_AND_EXPIRY`。

如果針對追蹤路徑訊息產生追蹤路徑回覆訊息, 則 **不會** 傳遞追蹤路徑訊息中的捨棄選項及到期時間。

如果您未指定此參數, 則不會指定 `MQRO_PASS_DISCARD_AND_EXPIRY`。

#### **-ro discard**

指定 `MQRO_DISCARD_MSG` 報告選項。這可以防止追蹤路徑訊息無限期保留在佇列管理程式網路中。

### **活動資訊的顯示**

IBM MQ 顯示路徑應用程式可以顯示剛放入佇列管理程式網路之追蹤路徑訊息的活動資訊, 也可以顯示先前產生之追蹤路徑訊息的活動資訊。它也可以顯示使用者撰寫的應用程式所記錄的其他資訊。

若要指定是否顯示針對追蹤路徑訊息傳回的活動資訊, 請指定下列參數:

**-n**

指定不顯示追蹤路徑訊息所傳回的活動資訊。

如果此參數隨附追蹤路徑回覆訊息 (*-ar*) 的要求, 或從 (*-ro ReportOption*) 產生選項的任何報告, 則必須使用 *-rq ReplyToQ* 來指定特定的 (非模型) 回覆目的地佇列。依預設, 只會要求活動報告訊息。

將追蹤路徑訊息放置到指定的目標佇列之後, 會顯示 48 個字元的十六進位字串, 其中包含追蹤路徑訊息的訊息 ID。IBM MQ 顯示路徑應用程式可以使用訊息 ID, 稍後使用 *-i CorrelId* 參數來顯示追蹤路徑訊息的活動資訊。

如果您未指定此參數, 則針對追蹤路徑訊息傳回的活動資訊會以 *-v* 參數指定的格式顯示。

當顯示剛放入佇列管理程式網路之追蹤路徑訊息的活動資訊時, 可以指定下列參數:

**-w WaitTime**

指定 IBM MQ 顯示遞送應用程式等待活動報告或追蹤遞送回覆訊息回到指定的回覆目的地佇列的時間 (以秒為單位)。

如果未指定此參數, 則會將等待時間指定為追蹤路徑訊息的到期時間加上 60 秒。

當顯示先前累計的活動資訊時, 必須設定下列參數:

**-q TargetQName**

如果正在使用 IBM MQ 顯示路徑應用程式來檢視先前收集的活動資訊, 則 *TargetQName* 會指定儲存活動資訊的佇列名稱。

**-i CorrelId**

當使用 IBM MQ 顯示路徑應用程式來僅顯示先前累計的活動資訊時, 會使用此參數。在 *-q TargetQName* 指定的佇列上, 可能會有許多活動報告及追蹤路徑回覆訊息。*CorrelId* 用來識別與追蹤路徑訊息相關的活動報告或追蹤路徑回覆訊息。在 *CorrelId* 中指定原始追蹤路徑訊息的訊息 ID。

*CorrelId* 的格式為 48 個字元的十六進位字串。

當顯示先前累計的活動資訊時, 或當顯示追蹤路徑訊息的現行活動資訊時, 可以使用下列參數:

**-b**

指定 IBM MQ 顯示遞送應用程式只會瀏覽活動報告或與訊息相關的追蹤遞送回覆訊息。這可讓稍後再次顯示活動資訊。

如果您未指定此參數, 則 IBM MQ 顯示路徑應用程式會破壞性地取得活動報告或與訊息相關的追蹤路徑回覆訊息。

**-v summary | all | none | outline DisplayOption**

**摘要**

顯示透過其遞送追蹤路徑訊息的佇列。

**all**

即會顯示所有可用的資訊。

**無**

不顯示任何資訊。

**outline DisplayOption**

指定追蹤路徑訊息的顯示選項。可以使用逗點作為分隔字元來指定多個顯示選項。

如果未提供任何值, 則會顯示下列:

- 應用程式名稱
- 每一個作業的類型
- 任何作業特定的參數

*DisplayOption* 的可能值為:

**活動**

即會顯示活動 PCF 群組中的所有非 PCF 群組參數。

**ID**

顯示參數 ID 為 MQBACF\_MSG\_ID 或 MQBACF\_CORREL\_ID 的值。這會置換 *msgdelta*。

## 訊息

會顯示 訊息 PCF 群組中的所有非 PCF 群組參數。指定此值時，您無法指定 *msgdelta*。

## msgdelta

即會顯示 訊息 PCF 群組中自前次作業以來已變更的所有非 PCF 群組參數。指定此值時，您無法指定 *message*。

## operation

畫面上會顯示 作業 PCF 群組中的所有非 PCF 群組參數。

## 追蹤路徑

畫面上會顯示 *TraceRoute* PCF 群組中的所有非 PCF 群組參數。

如果您未指定此參數，則會顯示訊息路徑的摘要。

## 顯示其他資訊

當透過佇列管理程式網路遞送追蹤路徑訊息時，使用者撰寫的應用程式可以透過將一個以上其他 PCF 參數寫入追蹤路徑訊息的訊息資料或活動報告的訊息資料，來記錄其他資訊。若要讓 IBM MQ 顯示路徑應用程式以可讀取的格式顯示其他資訊，必須以特定格式記錄該資訊，如第 66 頁的『其他活動資訊』中所述。

## IBM MQ 顯示路由應用程式範例

下列範例顯示如何使用 IBM MQ 顯示路徑應用程式。在每一個範例中，兩個佇列管理程式 (QM1 及 QM2) 透過兩個通道 (QM2.TO.QM1 和 QM1.TO.QM2)。

### 範例 1-要求活動報告

從遞送至目標佇列的追蹤路徑訊息顯示活動資訊

在此範例中，IBM MQ 顯示畫面路徑應用程式會連接至佇列管理程式 QM1，並用來產生追蹤路徑訊息並遞送至目標佇列 TARGET.Q，在遠端佇列管理程式 QM2 上。指定必要的報告選項，以便在遞送追蹤路徑回覆訊息時要求活動報告。到達目標佇列時，會捨棄追蹤路徑訊息。使用活動報告傳回 IBM MQ 顯示路徑應用程式的活動資訊會依序排列並顯示。

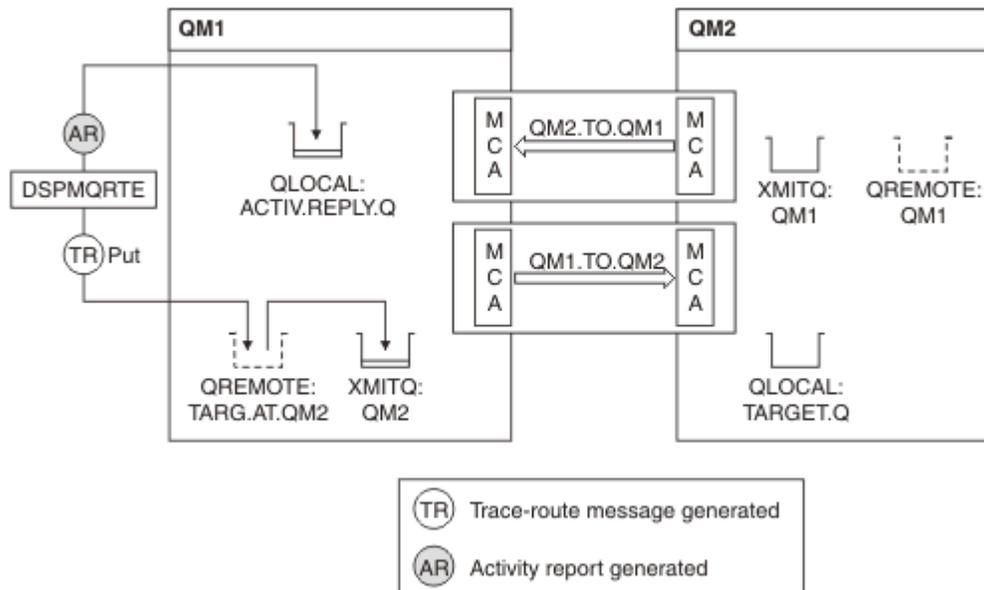


圖 9: 要求活動報告，圖 1

- 每一個佇列管理程式 (QM1 及 QM2) 的 ACTIVREC 屬性設為 MSG。
- 發出下列指令:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1 是 IBM MQ 顯示路徑應用程式所連接的佇列管理程式名稱 TARG.AT.QM2 是目標佇列的名稱，而 ACTIV.REPLY.Q 是佇列名稱，要求將追蹤路徑訊息的所有回應傳送至該佇列。

所有未指定的選項都會採用預設值，但請特別注意 -f 選項 (追蹤路徑訊息只會轉遞至允許使用 TraceRoute PCF 群組之 Deliver 參數的佇列管理程式)、-d 選項 (到達時，追蹤路徑訊息不會放置在目標佇列上)、-ro 選項 (指定 MQRO\_ACTIVITY 及 MQRO\_DISCARD\_MSG 報告選項) 及 -t 選項 (記錄中等詳細層次活動)。

- DSPMQRTE 會產生追蹤路徑訊息，並將它放置在遠端佇列 TARG.AT.QM2。
- 然後，DSPMQRTE 會查看佇列管理程式 QM1 的 ACTIVREC 屬性值。值為 MSG，因此 DSPMQRTE 會產生生活動報告，並將它放置在回覆佇列 ACTIV.REPLY.Q。

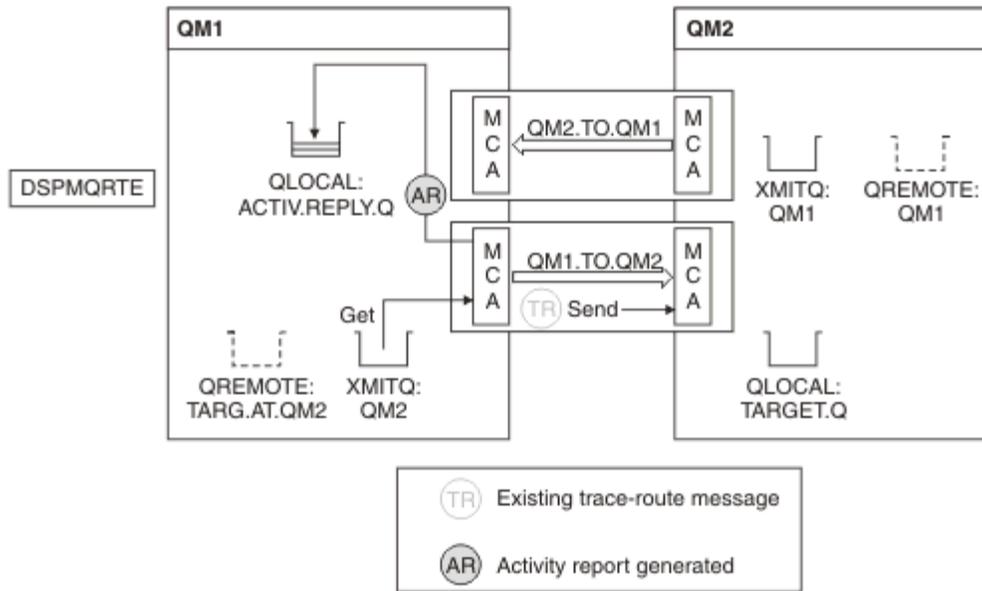


圖 10: 要求活動報告，圖表 2

- 傳送訊息通道代理程式 (MCA) 會從傳輸佇列取得追蹤路徑訊息。訊息是追蹤路徑訊息，因此 MCA 會開始記錄活動資訊。
- 佇列管理程式 (QM1) 的 ACTIVREC 屬性是 MSG，且在訊息描述子的「報告」欄位中指定 MQRO\_ACTIVITY 選項，因此 MCA 稍後會產生活動報告。TraceRoute PCF 群組中的 RecordedActivities 參數值會增加 1。
- MCA 會檢查是否未超出 TraceRoute PCF 群組中的 MaxActivities 值。
- 在將訊息轉遞至 QM2 之前，MCA 會遵循轉遞中說明的演算法 (步驟第 63 頁的『1』、第 64 頁的『4』及第 64 頁的『5』) MCA 選擇傳送訊息。
- 然後 MCA 會產生活動報告，並將它放在回覆佇列 (ACTIV.REPLY.Q)。

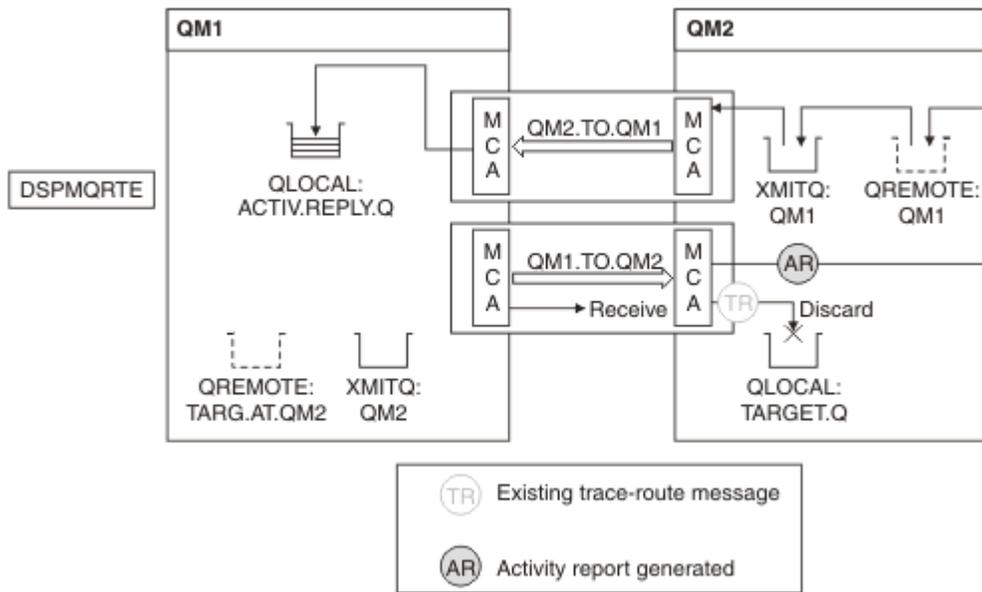


圖 11: 要求活動報告，圖 3

- 接收端 MCA 會從通道接收追蹤路徑訊息。訊息是追蹤路徑訊息，因此 MCA 會開始記錄活動的相關資訊。
- 如果追蹤路徑訊息來自的佇列管理程式是 IBM WebSphere MQ 5.3.1 或更早版本，則 MCA 會將 TraceRoute PCF 的 DiscontinuityCount 參數遞增 1。這不是這樣的。
- 佇列管理程式 (QM2) 的 ACTIVREC 屬性是 MSG，並指定 MQRO\_ACTIVITY 選項，因此 MCA 會產生活動報告。RecordedActivities 參數值會增加 1。
- 目標佇列是本端佇列，因此會根據 TraceRoute PCF 群組中的「遞送」參數值，使用意見回饋 MQFB\_NOT\_DELIVERED 來捨棄訊息。
- 然後，MCA 會產生最終活動報告，並將它放置在回覆佇列上。這會解析為與佇列管理程式 QM1 相關聯的傳輸佇列，並將活動報告傳回佇列管理程式 QM1 (ACTIV.REPLY.Q)。

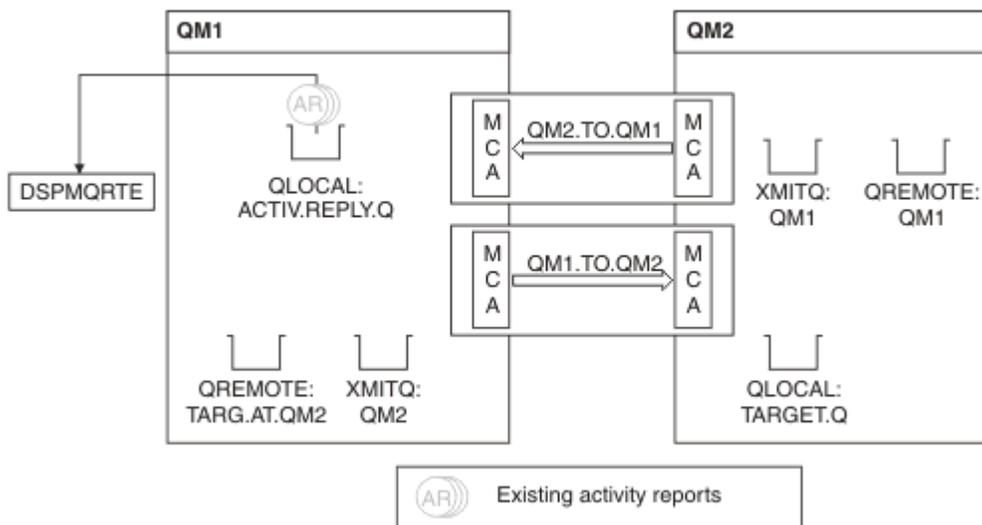


圖 12: 要求活動報告，圖 4

- 同時，DSPMQRTE 已持續在回覆佇列 (ACTIV.REPLY.Q)，等待活動報告。因為啟動 DSPMQRTE 時未指定 -w，所以它最多會等待 120 秒 (比追蹤路徑訊息的到期時間長 60 秒)。
- DSPMQRTE 從回覆佇列中取得 3 個活動報告。

- 針對每一個活動，使用 TraceRoute PCF 群組中的 RecordedActivities、UnrecordedActivities 及 DiscontinuityCount 參數來排序活動報告。在此範例中，唯一非零的值是 RecordedActivities，因此這是實際使用的唯一參數。
- 一旦顯示捨棄作業，程式即會結束。即使最終作業是捨棄，也會被視為因為回饋是 MQFB\_NOT\_DELIVERED 而發生放置。

顯示的輸出如下：

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

### 範例 2-要求追蹤路徑回覆訊息

產生追蹤路徑訊息並遞送至目標佇列

在此範例中，IBM MQ 顯示畫面路徑應用程式會連接至佇列管理程式 QM1，並用來產生追蹤路徑訊息並遞送至目標佇列 TARGET.Q，在遠端佇列管理程式 QM2 上。指定必要的選項，以便在追蹤路徑訊息中累計活動資訊。到達目標佇列時，會要求追蹤路徑回覆訊息，並捨棄追蹤路徑訊息。

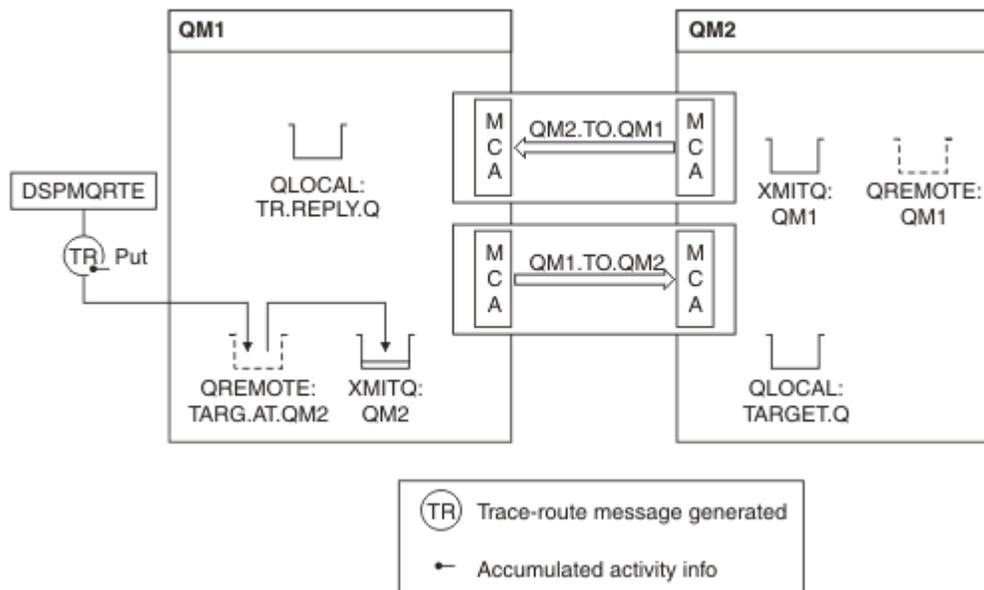


圖 13: 要求追蹤路徑回覆訊息，圖表 1

- 每一個佇列管理程式 (QM1 及 QM2) 的 ROUTEREC 屬性設為 MSG。
- 發出下列指令：

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq TR.REPLY.Q -ac -ar -ro discard
```

QM1 是 IBM MQ 顯示路徑應用程式所連接的佇列管理程式名稱 TARG.AT.QM2 是目標佇列的名稱，而 ACTIV.REPLY.Q 是佇列名稱，要求將追蹤路徑訊息的所有回應傳送至該佇列。-ac 選項指定在追蹤路徑訊息中累計活動資訊，-ar 選項指定將所有累計活動傳送至 -rq 選項 (即 TR.REPLY.Q)。-ro 選項指定設定報告選項 MQRO\_DISCARD\_MSG，這表示在此範例中不會產生活動報告。

- DSPMQRTE 會先累計追蹤路徑訊息中的活動資訊，然後再將訊息放置在目標路徑上。佇列管理程式屬性 ROUTEREC 不可為 DISABLED，才能發生此情況。

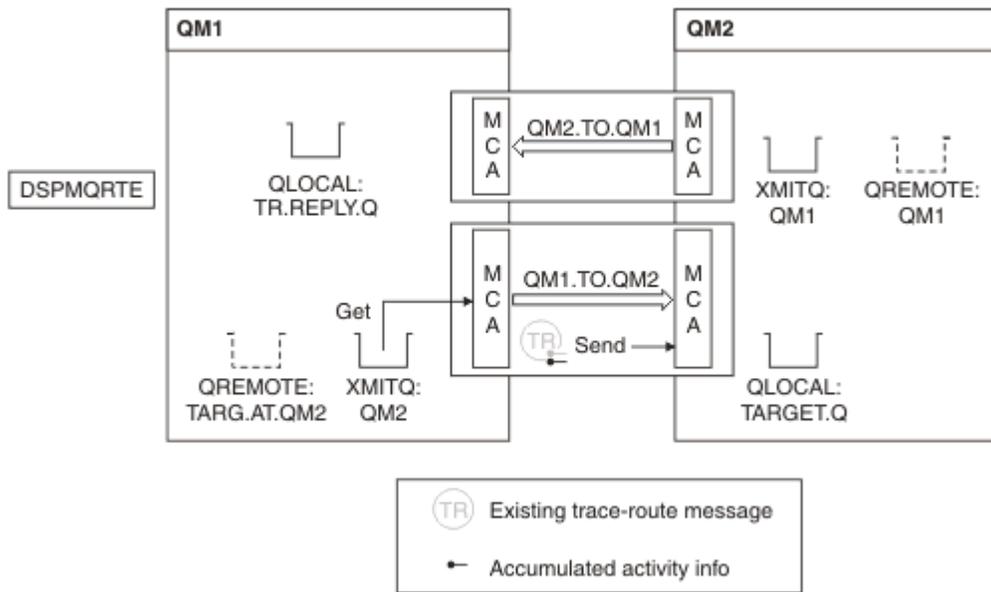


圖 14: 要求追蹤路徑回覆訊息，圖 2

- 訊息是追蹤路徑訊息，因此傳送端 MCA 會開始記錄活動的相關資訊。
- QM1 上的佇列管理程式屬性 ROUTEREC 不是 DISABLED，因此在將訊息轉遞至佇列管理程式 QM2 之前，MCA 會累計訊息內的活動資訊。

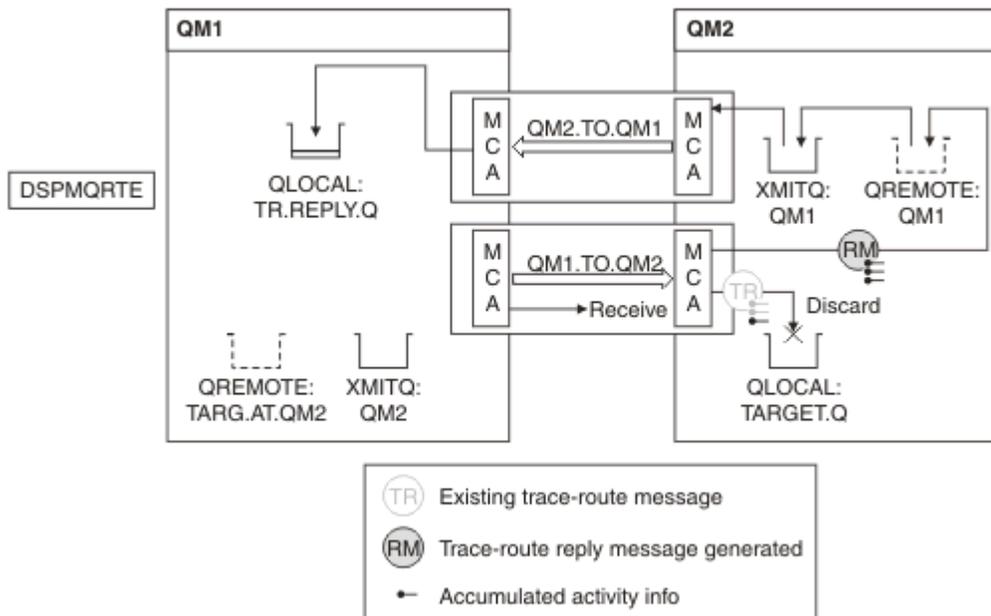


圖 15: 要求追蹤路徑回覆訊息，圖 3

- 訊息是追蹤路徑訊息，因此接收 MCA 會開始記錄活動的相關資訊。
- QM2 上的佇列管理程式屬性 ROUTEREC 不是 DISABLED，因此 MCA 會累計訊息內的資訊。
- 目標佇列是本端佇列，因此會根據 TraceRoute PCF 群組中的「遞送」參數值，使用意見回饋 MQFB\_NOT\_DELIVERED 來捨棄訊息。
- 這是在訊息上執行的最後一個活動，而且因為 QM1 上的佇列管理程式屬性 ROUTEREC 不是 DISABLED，所以 MCA 會根據「累計」值產生追蹤路徑回覆訊息。ROUTEREC 的值是 MSG，因此回覆訊息會放置在回覆佇列上。回覆訊息包含追蹤路徑訊息中所有累計的活動資訊。

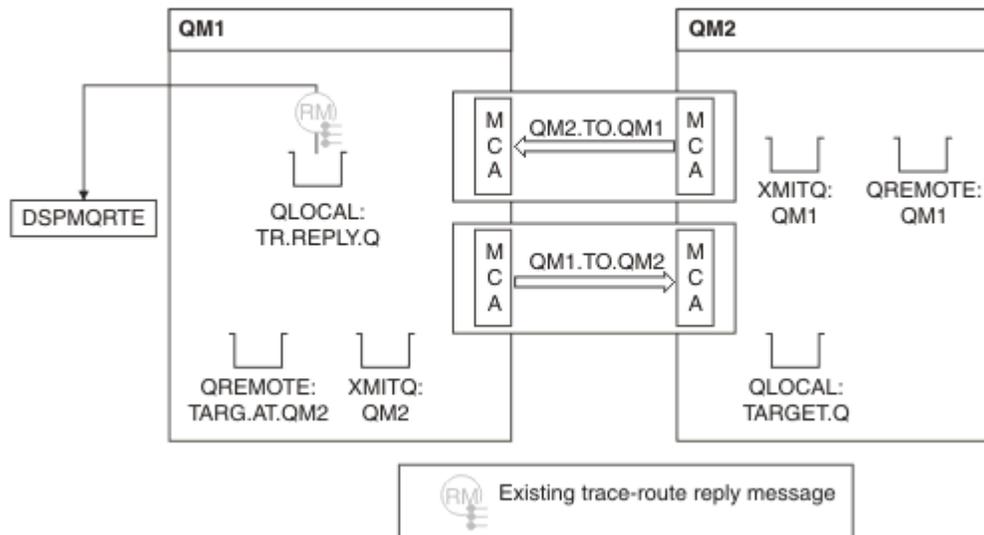


圖 16: 要求追蹤路徑回覆訊息，圖 4

- 同時 DSPMQRTE 正在等待追蹤路徑回覆訊息回到回覆佇列。當它傳回時，DSPMQRTE 會剖析它包含的每一個活動，並將它印出。最終作業是捨棄作業。DSPMQRTE 在列印之後結束。

顯示的輸出如下：

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
TR.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

### 範例 3-將活動報告遞送至系統佇列

偵測活動報告何時遞送至回覆目的地佇列以外的佇列，並使用 IBM MQ 顯示路徑應用程式從其他佇列讀取活動報告。

此範例與第 74 頁的『範例 1-要求活動報告』相同，只不過 QM2 現在將 ACTIVREC 佇列管理屬性的值設為 QUEUE。通道 QM1.TO.QM2，此動作才會生效。

此範例示範如何偵測活動報告何時遞送至回覆目的地佇列以外的佇列。一旦偵測到，即會使用 IBM MQ 顯示路徑應用程式從另一個佇列讀取活動報告。

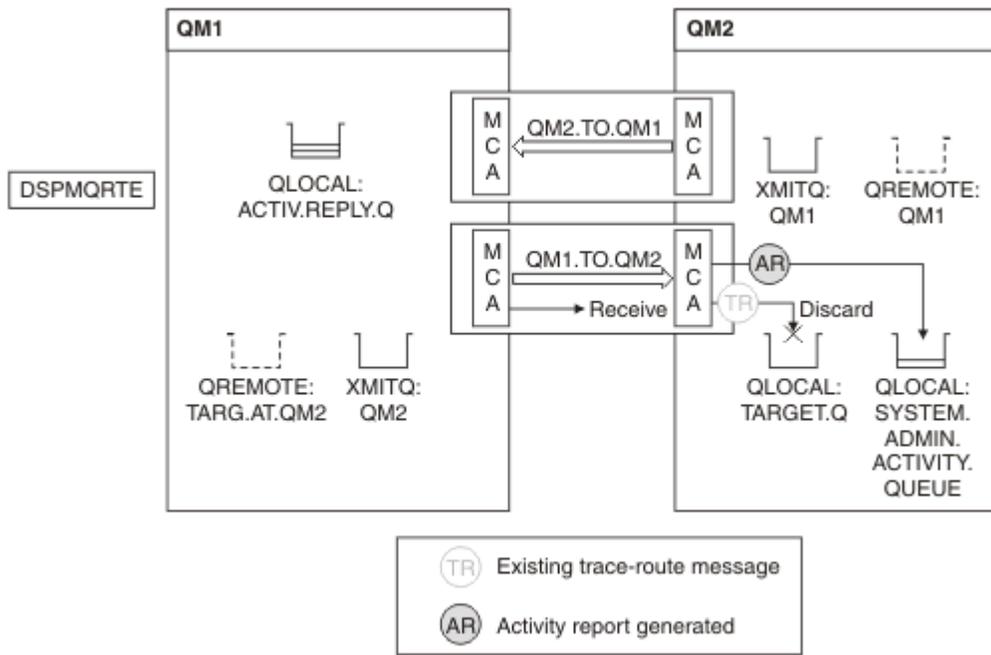


圖 17: 將活動報告遞送至系統佇列，圖 1

- 訊息是追蹤路徑訊息，因此接收 MCA 會開始記錄活動的相關資訊。
- QM2 上的 ACTIVREC 佇列管理程式屬性值現在是 QUEUE，因此 MCA 會產生活動報告，但會將它放在系統佇列 (SYSTEM.ADMIN.ACTIVITY.QUEUE)，而不是在回覆佇列上 (ACTIV.REPLY.Q)。

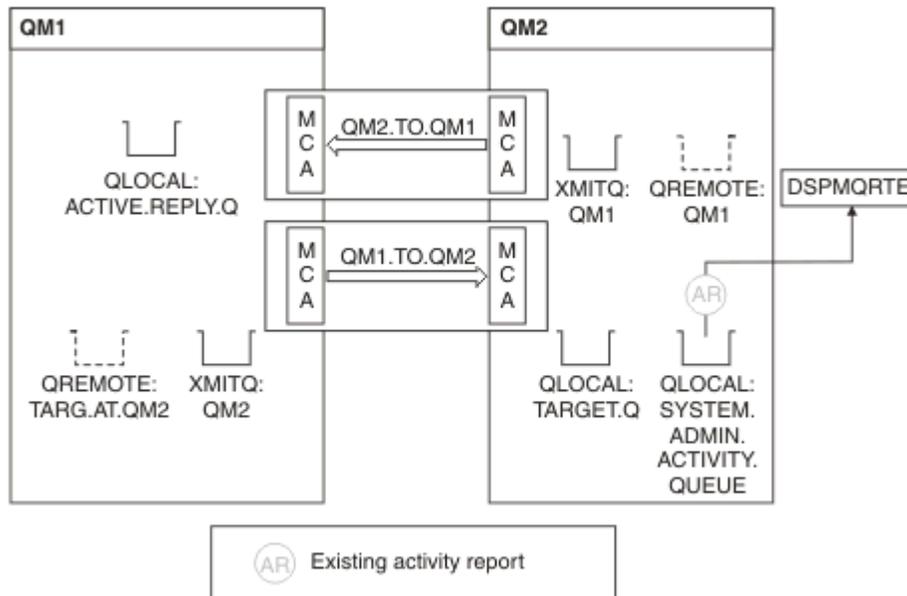


圖 18: 將活動報告遞送至系統佇列，圖 2

- 同時 DSPMQRTE 一直在等待活動報告到達 ACTIV.REPLY.Q。只有兩個人來了 DSPMQRTE 會繼續等待 120 秒，因為路徑似乎尚未完成。

顯示的輸出如下：

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
ACTIV.REPLY.Q -v outline identifiers'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

```
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'
```

```
Operation:
  OperationType: Put
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C4152474551202020202020A3C9154220001502'
  CorrelId: X'414D51204C4152474551202020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
```

```
-----
Activity:
  ApplName: 'cann\output\bin\runmqchl.EXE'
```

```
Operation:
  OperationType: Get
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001505'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001502'
```

```
EmbeddedMQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'
```

```
Operation:
  OperationType: Send
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  RemoteQMgrName: 'QM2'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Sender
  XmitQName: 'QM2'
```

```
-----
AMQ8652: DSPMQRTE command has finished.
```

- DSPMQRTE 觀察到的前次作業是「傳送」，因此通道正在執行中。現在，我們必須找出為何不再從佇列管理程式 QM2 (如 RemoteQMgrName 中所識別) 接收任何活動報告。
- 若要檢查系統佇列上是否有任何活動資訊，請在 QM2 上啟動 DSPMQRTE，以嘗試並收集更多活動報告。使用下列指令來啟動 DSPMQRTE:

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

其中 414D51204C41524745512020202020A3C9154220001502 是所放置追蹤路徑訊息的 MsgId。

- 然後，DSPMQRTE 會重新執行一系列 MQGET，等待與具有指定 ID 之追蹤路徑訊息相關的系統活動佇列上的回應。
- DSPMQRTE 再取得一個活動報告，它會顯示此報告。DSPMQRTE 判定遺漏之前的活動報告，並顯示一則訊息指出此情況。不過，我們已經知道這部分的路線。

顯示的輸出如下:

```

AMQ8653: DSPMQRTE command started with options '-m QM2
-q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C915420001502 -v outline'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  Activity information unavailable.
-----
Activity:
  ApplName: 'cann\output\bin\AMQRMPPA.EXE'

  Operation:
  OperationType: Receive
  QMgrName: 'QM2'
  RemoteQMGrName: 'QM1'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Receiver

  Operation:
  OperationType: Discard
  QMgrName: 'QM2'
  QName: 'TARGET.Q'
  Feedback: NotDelivered
-----
AMQ8652: DSPMQRTE command has finished.

```

- 此活動報告指出路徑資訊現在已完成。未發生任何問題。
- 僅因為路徑資訊無法使用，或因為 DSPMQRTE 無法顯示所有路徑，這並不表示未遞送訊息。例如，不同佇列管理程式的佇列管理程式屬性可能不同，或可能未定義回覆佇列來取回回應。

#### 範例 4-診斷通道問題

診斷追蹤路徑訊息未到達目標佇列的問題

在此範例中，IBM MQ 顯示路徑應用程式連接至佇列管理程式 QM1 會產生追蹤路徑訊息，然後嘗試將它遞送至目標佇列 TARGET.Q，在遠端佇列管理程式 QM2 上。在此範例中，trace-route 訊息未到達目標佇列。可用的活動報告是用來診斷問題。

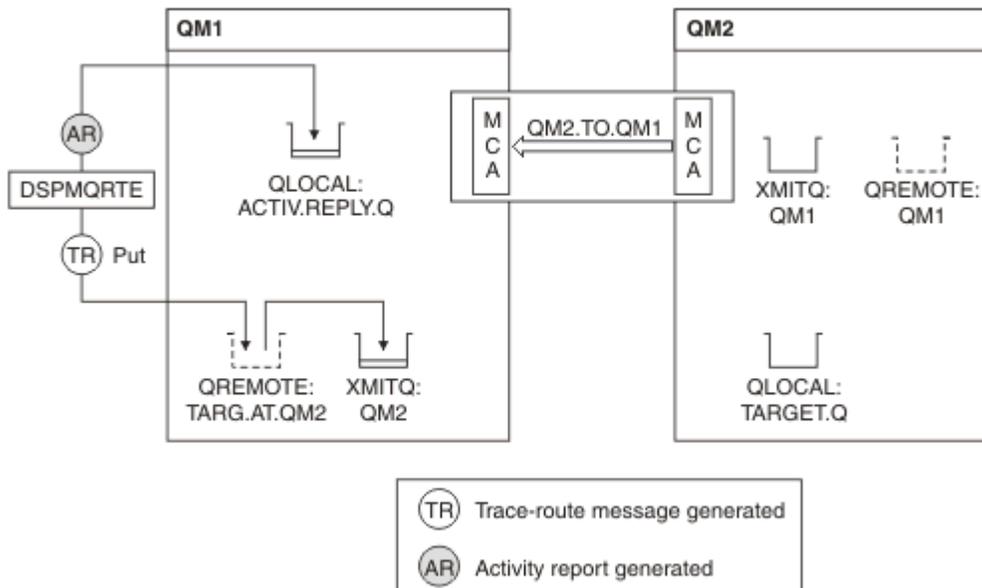


圖 19: 診斷通道問題

- 在此範例中，通道 QM1.TO.QM2 不在執行中。
- DSPMQRTE 將追蹤路徑訊息 (如範例 1) 放置到目標佇列並產生活動報告。

- 沒有 MCA 可從傳輸佇列 (QM2) 取得訊息，因此這是 DSPMQRTE 從回覆佇列取回的唯一活動報告。此時，路徑未完成的事實指出有問題。管理者可以使用在 ResolvedQName 中找到的傳輸佇列來調查未處理傳輸佇列的原因。

顯示的輸出如下：

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'

  Operation:
    OperationType: Put
    QMgrName: 'QM1'
    QName: 'TARG.AT.QM2'
    ResolvedQName: 'QM2'
    RemoteQName: 'TARGET.Q'
    RemoteQMgrName: 'QM2'
-----
AMQ8652: DSPMQRTE command has finished.
```

## 活動報告參照

請利用這個頁面來取得活動報告訊息格式的概觀。活動報告訊息資料包含說明活動的參數。

## 活動報告格式

活動報告是包含訊息描述子及訊息資料的標準 IBM MQ 報告訊息。活動報告是由應用程式所產生的 PCF 訊息，這些應用程式代表已透過佇列管理程式網路遞送活動的訊息來執行活動。

活動報告包含下列資訊：

### 訊息描述子

MQMD 結構

### 訊息資料

由下列項目組成：

- 內嵌 PCF 標頭 (MQEPH)。
- 活動報告訊息資料。

活動報告訊息資料由活動 PCF 群組以及 *TraceRoute* PCF 群組 (如果針對追蹤路徑訊息產生的話) 組成。

第 84 頁的表 19 顯示這些報告的結構，包括僅在特定條件下傳回的參數。

表 19: 活動報告格式

MQMD 結構	內嵌 PCF 標頭 MQEPH 結構	活動報告訊息資料
結構 ID 結構版本 報告選項 訊息類型 有效期限 意見 編碼 編碼字集 ID 訊息格式 優先順序 持續性 訊息 ID 相關性 ID 取消次數 回覆目的地佇列 回覆目的地佇列管理程式 使用者 ID 帳戶記號 應用程式身分資料 應用程式類型 應用程式名稱 放置日期 放置時間 應用程式原始資料 群組 ID 訊息序號 偏移 訊息旗標 原始長度	結構 ID 結構版本 結構長度 編碼 編碼字集 ID 訊息格式 旗標 PCF 標頭 (MQCFH) 結構類型 結構長度 結構版本 指令 ID 訊息序號 控制選項 完成碼 原因碼 參數計數	活動 活動應用程式名稱 活動應用程式類型 活動說明 作業 作業類型 作業日期 作業時間 訊息 訊息長度 MQMD <sup>8</sup> EmbeddedMQMD 佇列管理程式名稱 佇列共用群組名稱 佇列名稱 <sup>1</sup> 2 <sup>3</sup> 7 已解析佇列名稱 <sup>1</sup> 版 7 遠端佇列名稱 <sup>3</sup> 7 遠端佇列管理程式名稱 <sup>2</sup> 3 <sup>4</sup> 5 版 7 訂閱層次 <sup>9</sup> 訂閱 ID 9 意見 <sup>2</sup> 10 通道名稱 <sup>4</sup> 5 通道類型 <sup>4</sup> 5 傳輸佇列名稱 <sup>5</sup> TraceRoute <sup>6</sup> 明細 記錄的活動 未記錄的活動 中斷計數 活動數上限 累計 交付

**附註:**

1. 針對「取得」及「瀏覽」作業傳回。
2. 針對捨棄作業傳回。
3. 針對「放置」、「放置回覆」及「放置報告」作業傳回。
4. 針對「接收」作業傳回。

5. 針對「傳送」作業傳回。
6. 傳回追蹤路徑訊息。
7. 對於「發佈」活動內包含的主題的「放置」作業未傳回。
8. 未針對「排除的發佈」作業傳回。對於「發佈及捨棄的發佈」作業，傳回包含參數子集。
9. 針對「發佈」、「捨棄的發佈」及「排除的發佈」作業傳回。
10. 針對「已捨棄發佈」及「已排除發佈」作業傳回。

## 活動報告 MQMD (訊息描述子)

請利用這個頁面來檢視活動報告的 MQMD 結構所包含的值

### StrucId

結構 ID:

資料類型

MQCHAR4

值

MQMD\_STRUC\_ID。

### Version

結構版本號碼

資料類型

MQLONG

值

從原始訊息描述子複製。可能的值為：

#### MQMD\_VERSION\_1

Version-1 訊息描述子結構，在所有環境中都受支援。

#### MQMD\_VERSION\_2

Version-2 訊息描述子結構，在下列環境中受支援：

-  AIX
-  IBM i
-  Linux
-  Solaris
-  Windows
-  z/OS

及所有連接至這些系統的 IBM MQ MQI clients 。

### Report

進一步報告訊息的選項

資料類型

MQLONG

值

如果在原始訊息描述子的 報告 欄位中指定 MQRO\_PASS\_DISCARD\_AND\_EXPIRY 或 MQRO\_DISCARD\_MSG:

#### 捨棄 MQRO\_DISCARD

如果報告無法遞送至目的地佇列，則會捨棄該報告。

否則：

#### MQRO\_NONE

不需要報告。

### **MsgType**

指出訊息類型

資料類型

MQLONG

值

MQMT\_REPORT

### **Expiry**

報告訊息生命期限

資料類型

MQLONG

值

如果原始訊息描述子中的 報告 欄位指定為 MQRO\_PASS\_DISCARD\_AND\_EXPIRY，則會使用原始訊息中的剩餘到期時間。

否則：

**MQEI\_UNLIMITED**

報告沒有到期時間。

### **Feedback**

說明：意見或原因碼。

資料類型：MQLONG。

值：**MQFB\_ACTIVITY**  
活動報告。

### **Encoding**

說明：報告訊息資料的數值編碼。

資料類型：MQLONG。

值：MQENC\_NATIVE。

### **CodedCharSetId**

說明：報告訊息資料的字集 ID。

資料類型：MQLONG。

值：適當設定。

### **Format**

說明：報告訊息資料的格式名稱

資料類型：MQCHAR8。

值：**MQFMT\_EMBEDDED\_PCF**  
內嵌 PCF 訊息。

### **Priority**

說明：報告訊息優先順序。

資料類型：MQLONG。

值：從原始訊息描述子複製。

### **Persistence**

說明： 報告訊息持續性。  
資料類型： MQLONG。  
值： 從原始訊息描述子複製。

### **MsgId**

說明： 訊息 ID。  
資料類型： MQBYTE24。  
值： 如果原始訊息描述子中的 報告 欄位指定為 MQRO\_PASS\_MSG\_ID，則會使用原始訊息中的訊息 ID。  
否則，佇列管理程式會產生唯一值。

### **CorrelId**

說明： 相關性 ID。  
資料類型： MQBYTE24。  
值： 如果原始訊息描述子中的 報告 欄位指定為 MQRO\_PASS\_CORREL\_ID，則會使用原始訊息中的相關性 ID。  
否則，會從原始訊息複製訊息 ID。

### **BackoutCount**

說明： 取消計數器。  
資料類型： MQLONG。  
值： 0。

### **ReplyToQ**

說明： 回覆佇列的名稱。  
資料類型： MQCHAR48。  
值： 空白。

### **ReplyToQMgr**

說明： 回覆佇列管理程式的名稱。  
資料類型： MQCHAR48。  
值： 產生報告訊息的佇列管理程式名稱。

### **UserIdentifier**

說明： 產生報告訊息之應用程式的使用者 ID。  
資料類型： MQCHAR12。  
值： 從原始訊息描述子複製。

### **AccountingToken**

說明： 帳戶記號，可讓應用程式對因訊息而完成的工作收取費用。  
資料類型： MQBYTE32。

值： 從原始訊息描述子複製。

### ***ApplIdentityData***

說明： 與身分相關的應用程式資料。

資料類型： MQCHAR32.

值： 從原始訊息描述子複製。

### ***PutApplType***

說明： 放置報告訊息的應用程式類型。

資料類型： MQLONG。

值： **MQAT\_QMGR**  
佇列管理程式已產生訊息。

### ***PutApplName***

說明： 放置報告訊息的應用程式名稱。

資料類型： MQCHAR28.

值： 佇列管理程式名稱的前 28 個位元組，或產生報告訊息的 MCA 名稱。

### ***PutDate***

說明： 放置訊息的日期。

資料類型： MQCHAR8.

值： 由佇列管理程式產生。

### ***PutTime***

說明： 放置訊息的時間。

資料類型： MQCHAR8.

值： 由佇列管理程式產生。

### ***ApplOriginData***

說明： 與來源相關的應用程式資料。

資料類型： MQCHAR4.

值： 空白。

如果版本是 MQMD\_VERSION\_2，則會呈現下列其他欄位：

### ***GroupId***

說明： 識別實體訊息所屬的訊息群組或邏輯訊息。

資料類型： MQBYTE24.

值： 從原始訊息描述子複製。

### ***MsgSeqNumber***

說明： 群組內邏輯訊息的序號。

資料類型： MQLONG。

值: 從原始訊息描述子複製。

### **Offset**

說明: 實體訊息中資料從邏輯訊息開始的偏移。

資料類型: MQLONG。

值: 從原始訊息描述子複製。

### **MsgFlags**

說明: 指定訊息屬性或控制其處理的訊息旗標。

資料類型: MQLONG。

值: 從原始訊息描述子複製。

### **OriginalLength**

說明: 原始訊息的長度。

資料類型: MQLONG。

值: 從原始訊息描述子複製。

## **活動報告 MQEPH (內嵌 PCF 標頭)**

請利用這個頁面來檢視活動報告的 MQEPH 結構所包含的值

MQEPH 結構包含活動報告訊息資料隨附的 PCF 資訊，以及它之後的應用程式訊息資料的說明。

對於活動報告，MQEPH 結構包含下列值:

### **StrucId**

說明: 結構 ID。

資料類型: MQCHAR4。

值: MQEPH\_STRUC\_ID。

### **Version**

說明: 結構版本號碼。

資料類型: MQLONG。

值: MQEPH\_VERSION\_1。

### **StrucLength**

說明: 結構長度。

資料類型: MQLONG。

值: 結構的總長度，包括其後的 PCF 參數結構。

### **Encoding**

說明: 遵循最後一個 PCF 參數結構之訊息資料的數值編碼。

資料類型: MQLONG。

值: 如果報告訊息中包含來自原始應用程式訊息資料的任何資料，則會從原始訊息描述子的編碼欄位複製該值。

否則為 0。

### **CodedCharSetId**

- 說明： 遵循最後一個 PCF 參數結構之訊息資料的字集 ID。
- 資料類型： MQLONG。
- 值： 如果報告訊息中包含來自原始應用程式訊息資料的任何資料，則會從原始訊息描述子的 *CodedCharSetId* 欄位複製值。  
否則， MQCCSI\_UNDEFINED。

### **Format**

- 說明： 遵循最後一個 PCF 參數結構之訊息資料的格式名稱。
- 資料類型： MQCHAR8。
- 值： 如果報告訊息中包含來自原始應用程式訊息資料的任何資料，則會從原始訊息描述子的 格式 欄位複製值。  
否則為 MQFMT\_NONE。

### **Flags**

- 說明： 指定結構屬性或控制其處理的旗標。
- 資料類型： MQLONG。
- 值： **MQEPH\_CCSID\_EMBEDDED**  
指定在每一個結構中的 *CodedCharSetId* 欄位內個別指定包含字元資料之參數的字集。

### **PCFHeader**

- 說明： 可程式指令格式標頭
- 資料類型： MQCFH。
- 值： 請參閱 [第 90 頁的『活動報告 MQCFH \(PCF 標頭\)』](#)。

### **活動報告 MQCFH (PCF 標頭)**

請利用這個頁面來檢視活動報告的 MQCFH 結構所包含的 PCF 值

對於活動報告， MQCFH 結構包含下列值：

#### **Type**

- 說明： 識別報告訊息內容的結構類型。
- 資料類型： MQLONG。
- 值： **MQCFT\_REPORT**  
訊息是報告。

#### **StrucLength**

- 說明： 結構長度。
- 資料類型： MQLONG。
- 值： **MQCFH\_STRUC\_LENGTH**  
MQCFH 結構的長度 (以位元組為單位)。

### **Version**

說明：結構版本號碼。  
資料類型：MQLONG。  
值：MQCFH\_VERSION\_3

### **Command**

說明：指令 ID。這會識別訊息的種類。  
資料類型：MQLONG。  
值：**MQCMD\_ACTIVITY\_MSG**  
訊息活動。

### **MsgSeqNumber**

說明：訊息序號。這是相關訊息群組內的訊息序號。  
資料類型：MQLONG。  
值：1.

### **Control**

說明：控制選項。  
資料類型：MQLONG。  
值：MQCFC\_LAST。

### **CompCode**

說明：完成碼。  
資料類型：MQLONG。  
值：MQCC\_OK。

### **Reason**

說明：原因碼定義完成碼。  
資料類型：MQLONG。  
值：MQRC\_NONE。

### **ParameterCount**

說明：參數結構計數。這是遵循 MQCFH 結構的參數結構數目。群組結構 (MQCFGR) 及其包含的參數結構只會計為一個結構。  
資料類型：MQLONG。  
值：1 或以上。

## **活動報告訊息資料**

請利用這個頁面來檢視活動報告訊息中的活動 PCF 群組所包含的參數。只有在已執行特定作業時，才會傳回部分參數。

活動報告訊息資料由活動 PCF 群組以及 *TraceRoute* PCF 群組 (如果針對追蹤路徑訊息產生的話) 組成。本主題詳述活動 PCF 群組。

部分參數 (說明為 [作業特定活動報告訊息資料](#)) 僅在已執行特定作業時才會傳回。

對於活動報告，活動報告訊息資料包含下列參數：

### **Activity**

說明：	說明活動的分組參數。
ID：	MQGACF_ACTIVITY。
資料類型：	MQCFGR。
包含在 PCF 群組中：	無。
PCF 群組中的參數：	<i>ActivityApplName</i> <i>ActivityApplType</i> <i>ActivityDescription</i> <i>Operation</i> <i>TraceRoute</i>
已傳回：	始終。

### **ActivityApplName**

說明：	執行活動的應用程式名稱。
ID：	MQCACF_APPL_NAME。
資料類型：	MQCFST。
包含在 PCF 群組中：	活動。
長度上限：	MQ_APPL_NAME_LENGTH。
已傳回：	始終。

### **ActivityApplType**

說明：	執行活動的應用程式類型。
ID：	MQIA_APPL_TYPE。
資料類型：	MQCFIN。
包含在 PCF 群組中：	活動。
已傳回：	始終。

### **ActivityDescription**

說明：	應用程式所執行活動的說明。
ID：	MQCACF_ACTIVITY_DESCRIPTION。
資料類型：	MQCFST。
包含在 PCF 群組中：	活動。
長度上限：	64
已傳回：	始終。

### **Operation**

說明：	說明活動作業的分組參數。
-----	--------------

ID: MQGACF\_OPERATION。

資料類型: MQCFGR。

包含在 PCF 群組中: 活動。

PCF 群組中的參數:

- OperationType*
- OperationDate*
- OperationTime*
- Message*
- QMgrName*
- QSGName*

註: 視作業類型而定, 會在此群組中傳回其他參數。這些其他參數說明為 [作業特定活動報告訊息資料](#)。

已傳回: 活動中每個作業一個 作業 PCF 群組。

### ***OperationType***

說明: 執行的作業類型。

ID: MQIACF\_OPERATION\_TYPE。

資料類型: MQCFIN。

包含在 PCF 群組中: 作業。

值: MQOPER\_ \*。

已傳回: 始終。

### ***OperationDate***

說明: 執行作業的日期。

ID: MQCACF\_OPERATION\_DATE。

資料類型: MQCFST。

包含在 PCF 群組中: 作業。

長度上限: MQ\_DATE\_LENGTH。

已傳回: 始終。

### ***OperationTime***

說明: 執行作業的時間。

ID: MQCACF\_OPERATION\_TIME。

資料類型: MQCFST。

包含在 PCF 群組中: 作業。

長度上限: MQ\_TIME\_LENGTH。

已傳回: 始終。

### ***Message***

說明: 分組參數, 說明導致活動的訊息。

ID: MQGACF\_MESSAGE。  
資料類型: MQCFGR。  
包含在 PCF 群組中: 作業。  
群組中的參數: *MsgLength*  
*MQMD*  
*EmbeddedMQMD*  
已傳回: 一律，但「排除的發佈」作業除外。

### **MsgLength**

說明: 在發生活動之前，造成活動的訊息長度。  
ID: MQIACF\_MSG\_LENGTH。  
資料類型: MQCFIN。  
包含在 PCF 群組中: 訊息。  
已傳回: 始終。

### **MQMD**

說明: 與導致活動之訊息的訊息描述子相關的分組參數。  
ID: MQGACF\_MQMD。  
資料類型: MQCFGR。  
包含在 PCF 群組中: 訊息。

群組中的參數:

*StrucId*  
*Version*  
*Report*  
*MsgType*  
*Expiry*  
*Feedback*  
*Encoding*  
*CodedCharSetId*  
*Format*  
*Priority*  
*Persistence*  
*MsgId*  
*CorrelId*  
*BackoutCount*  
*ReplyToQ*  
*ReplyToQMgr*  
*UserIdentifier*  
*AccountingToken*  
*ApplIdentityData*  
*PutApplType*  
*PutApplName*  
*PutDate*  
*PutTime*  
*ApplOriginData*  
*GroupId*  
*MsgSeqNumber*  
*Offset*  
*MsgFlags*  
*OriginalLength*

已傳回: 一律，但「排除的發佈」作業除外。

### ***EmbeddedMQMD***

說明: 分組參數，說明內嵌在傳輸佇列訊息中的訊息描述子。

ID: MQGACF\_EMBEDDED\_MQMD。

資料類型: MQCFGR。

包含在 PCF 群組中: 訊息。

群組中的參數:

*StrucId*  
*Version*  
*Report*  
*MsgType*  
*Expiry*  
*Feedback*  
*Encoding*  
*CodedCharSetId*  
*Format*  
*Priority*  
*Persistence*  
*MsgId*  
*CorrelId*  
*BackoutCount*  
*ReplyToQ*  
*ReplyToQMgr*  
*UserIdentifier*  
*AccountingToken*  
*ApplIdentityData*  
*PutApplType*  
*PutApplName*  
*PutDate*  
*PutTime*  
*ApplOriginData*  
*GroupId*  
*MsgSeqNumber*  
*Offset*  
*MsgFlags*  
*OriginalLength*

已傳回: 對於佇列解析為傳輸佇列的 Get 作業。

### **StrucId**

說明: 結構 ID

ID: MQCACF\_STRUC\_ID。

資料類型: MQCFST。

包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。

長度上限: 4。

已傳回: 一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **Version**

說明: 結構版本號碼。

ID: MQIACF\_VERSION。

資料類型: MQCFIN。

包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。

已傳回：一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **Report**

說明：報告訊息的選項。

ID：MQIACF\_REPORT。

資料類型：MQCFIN。

包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。

已傳回：一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **MsgType**

說明：指出訊息的類型。

ID：MQIACF\_MSG\_TYPE。

資料類型：MQCFIN。

包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。

已傳回：一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **Expiry**

說明：訊息生命期限。

ID：MQIACF\_EXPIRY。

資料類型：MQCFIN。

包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。

已傳回：一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **Feedback**

說明：意見或原因碼。

ID：MQIACF\_FEEDBACK。

資料類型：MQCFIN。

包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。

已傳回：一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **Encoding**

說明：訊息資料的數值編碼。

ID：MQIACF\_ENCODING。

資料類型：MQCFIN。

包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。  
已傳回: 一律, 但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **CodedCharSetId**

說明: 訊息資料的字集 ID。  
ID: MQIA\_CODED\_CHAR\_SET\_ID。  
資料類型: MQCFIN。  
包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。  
已傳回: 一律, 但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **Format**

說明: 訊息資料的格式名稱  
ID: MQCACH\_FORMAT\_NAME。  
資料類型: MQCFST。  
包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。  
長度上限: MQ\_FORMAT\_LENGTH。  
已傳回: 一律, 但「排除的發佈」作業除外。

### **Priority**

說明: 訊息優先順序。  
ID: MQIACF\_PRIORITY。  
資料類型: MQCFIN。  
包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。  
已傳回: 一律, 但「排除的發佈」作業除外。

### **Persistence**

說明: 訊息持續性。  
ID: MQIACF\_PERSISTENCE。  
資料類型: MQCFIN。  
包含在 PCF 群組中: MQMD 或 *EmbeddedMQMD*。  
已傳回: 一律, 但「排除的發佈」作業除外。

### **MsgId**

說明: 訊息 ID。  
ID: MQBACF\_MSG\_ID。  
資料類型: MQCFBS。

包含在 PCF 群組中：  
MQMD 或 *EmbeddedMQMD*。  
長度上限：  
MQ\_MSG\_ID\_LENGTH。  
已傳回：  
一律，但「排除的發佈」作業除外。

### **CorrelId**

說明：  
相關性 ID。  
ID：  
MQBACF\_CORREL\_ID。  
資料類型：  
MQCFBS。  
包含在 PCF 群組中：  
MQMD 或 *EmbeddedMQMD*。  
長度上限：  
MQ\_CORREL\_ID\_LENGTH。  
已傳回：  
一律，但「排除的發佈」作業除外。

### **BackoutCount**

說明：  
取消計數器。  
ID：  
MQIACF\_BACKOUT\_COUNT。  
資料類型：  
MQCFIN。  
包含在 PCF 群組中：  
MQMD 或 *EmbeddedMQMD*。  
已傳回：  
一律，但「排除的發佈」作業及 MQMD 中的「發佈」及「捨棄的發佈」作業除外。

### **ReplyToQ**

說明：  
回覆佇列的名稱。  
ID：  
MQCACF\_REPLY\_TO\_Q。  
資料類型：  
MQCFST。  
包含在 PCF 群組中：  
MQMD 或 *EmbeddedMQMD*。  
長度上限：  
MQ\_Q\_NAME\_LENGTH。  
已傳回：  
一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」作業除外。

### **ReplyToQMgr**

說明：  
回覆佇列管理程式的名稱。  
ID：  
MQCACF\_REPLY\_TO\_Q\_MGR。  
資料類型：  
MQCFST。  
包含在 PCF 群組中：  
MQMD 或 *EmbeddedMQMD*。  
長度上限：  
MQ\_Q\_MGR\_NAME\_LENGTH。  
已傳回：  
一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」除外。

### **UserIdentifier**

說明:	產生訊息之應用程式的使用者 ID。
ID:	MQCACF_USER_IDENTIFIER。
資料類型:	MQCFST。
包含在 PCF 群組中:	MQMD 或 <i>EmbeddedMQMD</i> 。
長度上限:	MQ_USER_ID_length。
已傳回:	一律，但排除的發佈作業除外。

### **AccountingToken**

說明:	帳戶記號，可讓應用程式對因訊息而完成的工作收取費用。
ID:	MQBACF_ACCOUNTING_TOKEN。
資料類型:	MQCFBS。
包含在 PCF 群組中:	MQMD 或 <i>EmbeddedMQMD</i> 。
長度上限:	MQ_ACCOUNTING_TOKEN_LENGTH。
已傳回:	一律，但排除的發佈作業除外。

### **ApplIdentityData**

說明:	與身分相關的應用程式資料。
ID:	MQCACF_APPL_IDENTITY_DATA。
資料類型:	MQCFST。
包含在 PCF 群組中:	MQMD 或 <i>EmbeddedMQMD</i> 。
長度上限:	MQ_APPL_IDENTITY_DATA_LENGTH。
已傳回:	一律，但排除的發佈作業除外。

### **PutApplType**

說明:	放置訊息的應用程式類型。
ID:	MQIA_APPL_TYPE。
資料類型:	MQCFIN。
包含在 PCF 群組中:	MQMD 或 <i>EmbeddedMQMD</i> 。
已傳回:	一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」除外。

### **PutApplName**

說明:	放置訊息的應用程式名稱。
ID:	MQCACF_APPL_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	MQMD 或 <i>EmbeddedMQMD</i> 。
長度上限:	MQ_APPL_NAME_LENGTH。

已傳回：一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」除外。

### **PutDate**

說明：放置訊息的日期。  
ID：MQCACF\_PUT\_DATE。  
資料類型：MQCFST。  
包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。  
長度上限：MQ\_PUT\_DATE\_LENGTH。  
已傳回：一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」除外。

### **PutTime**

說明：放置訊息的時間。  
ID：MQCACF\_PUT\_TIME。  
資料類型：MQCFST。  
包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。  
長度上限：MQ\_PUT\_TIME\_LENGTH。  
已傳回：一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」除外。

### **ApplOriginData**

說明：與來源相關的應用程式資料。  
ID：MQCACF\_APPL\_ORIGIN\_DATA。  
資料類型：MQCFST。  
包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。  
長度上限：MQ\_APPL\_ORIGIN\_DATA\_LENGTH。  
已傳回：一律，但「排除的發佈作業」及 MQMD 中的「發佈及捨棄的發佈作業」除外。

### **GroupId**

說明：識別實體訊息所屬的訊息群組或邏輯訊息。  
ID：MQBACF\_GROUP\_ID。  
資料類型：MQCFBS。  
包含在 PCF 群組中：MQMD 或 *EmbeddedMQMD*。  
長度上限：MQ\_GROUP\_ID\_LENGTH。  
已傳回：如果版本指定為 MQMD\_VERSION\_2。未在「排除的發佈作業」及 MQMD 中針對「發佈及捨棄的發佈作業」傳回。

### **MsgSeqNumber**

說明：群組內邏輯訊息的序號。  
ID：MQIARCH\_MSG\_SEQUENCE\_NUMBER。

資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>MQMD</i> 或 <i>EmbeddedMQMD</i> 。
已傳回:	如果版本指定為 <i>MQMD_VERSION_2</i> 。未在「排除的發佈作業」及 <i>MQMD</i> 中針對「發佈及捨棄的發佈作業」傳回。

### **Offset**

說明:	實體訊息中資料從邏輯訊息開始的偏移。
ID:	MQIACF_OFFSET。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>MQMD</i> 或 <i>EmbeddedMQMD</i> 。
已傳回:	如果版本指定為 <i>MQMD_VERSION_2</i> 。未在「排除的發佈作業」及 <i>MQMD</i> 中針對「發佈及捨棄的發佈作業」傳回。

### **MsgFlags**

說明:	指定訊息屬性或控制其處理的訊息旗標。
ID:	MQIACF_MSG_FLAGS。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>MQMD</i> 或 <i>EmbeddedMQMD</i> 。
已傳回:	如果版本指定為 <i>MQMD_VERSION_2</i> 。未在「排除的發佈作業」及 <i>MQMD</i> 中針對「發佈及捨棄的發佈作業」傳回。

### **OriginalLength**

說明:	原始訊息的長度。
ID:	MQIACF_ORIGINAL_LENGTH。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>MQMD</i> 或 <i>EmbeddedMQMD</i> 。
已傳回:	如果版本指定為 <i>MQMD_VERSION_2</i> 。未在「排除的發佈作業」及 <i>MQMD</i> 中針對「發佈及捨棄的發佈作業」傳回。

### **QMgrName**

說明:	執行活動所在的佇列管理程式名稱。
ID:	MQCA_Q_MGR_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_MGR_NAME_LENGTH
已傳回:	始終。

### **QSGName**

說明:	執行活動的佇列管理程式所屬的佇列共用群組名稱。
-----	-------------------------

ID: MQCA\_QSG\_NAME。  
 資料類型: MQCFST。  
 包含在 PCF 群組中: 作業。  
 長度上限: MQ\_QSG\_NAME\_LENGTH  
 已傳回: 如果活動是在 IBM MQ for z/OS 佇列管理程式上執行。

### **TraceRoute**

說明: 指定追蹤路徑訊息屬性的分組參數。  
 ID: MQGACF\_TRACE\_ROUTE。  
 資料類型: MQCFGR。  
 包含在 PCF 群組中: 活動。  
 群組中的參數: *Detail*  
                   *RecordedActivities*  
                   *UnrecordedActivities*  
                   *DiscontinuityCount*  
                   *MaxActivities*  
                   *Accumulate*  
                   *Forward*  
                   *Deliver*  
 已傳回: 如果已代表追蹤路徑訊息執行活動。

*TraceRoute* PCF 群組中的參數值是產生活動報告時來自追蹤路徑訊息的參數值。

### **作業特定的活動報告訊息資料**

請利用這個頁面來檢視活動報告中 PCF 群組 作業 可能傳回的其他 PCF 參數，視 *OperationType* 參數的值而定。

其他參數視下列作業類型而有所不同:

取得/瀏覽 (MQOPER\_GET/MQOPER\_BROWSE)

針對 Get/Browse (MQOPER\_GET/MQOPER\_BROWSE) 作業類型 (已取得或瀏覽佇列上的訊息)，在 PCF 群組 作業 中傳回的其他活動報告訊息資料參數。

### **QName**

說明: 已開啟的佇列名稱。  
 ID: MQCA\_Q\_NAME。  
 資料類型: MQCFST。  
 包含在 PCF 群組中: 作業。  
 長度上限: MQ\_Q\_NAME\_LENGTH  
 已傳回: 始終。

### **ResolvedQName**

說明: 開啟的佇列解析成的名稱。  
 ID: MQCACF\_RESOLVED\_Q\_NAME。

資料類型: MQCFST。  
包含在 PCF 群組 作業。  
中:  
長度上限: MQ\_Q\_NAME\_LENGTH  
已傳回: 始終。

#### 捨棄 (MQOPER\_DISCARD)

針對捨棄 (MQOPER\_DISCARD) 作業類型 (已捨棄訊息), 在 PCF 群組 作業 中傳回的其他活動報告訊息資料參數。

#### **Feedback**

說明: 捨棄訊息的原因。  
ID: MQIACF\_FEEDBACK。  
資料類型: MQCFIN。  
包含在 PCF 群組 作業。  
中:  
已傳回: 始終。

#### **QName**

說明: 已開啟的佇列名稱。  
ID: MQCA\_Q\_NAME。  
資料類型: MQCFST。  
長度上限: MQ\_Q\_NAME\_LENGTH  
包含在 PCF 群組 作業。  
中:  
已傳回: 如果因為未順利將訊息放入佇列而捨棄該訊息。

#### **RemoteQMgrName**

說明: 訊息目的地的佇列管理程式名稱。  
ID: MQCA\_REMOTE\_Q\_MGR\_NAME。  
資料類型: MQCFST。  
長度上限: MQ\_Q\_MGR\_NAME\_LENGTH  
包含在 PCF 群組 作業。  
中:  
已傳回: 如果 *Feedback* 的值為 MQFB\_NOT\_FORWARDED。

#### 發佈/捨棄的發佈/排除的發佈 (MQOPER\_PUBLISH/MQOPER\_DISCARDE\_PUBLISH/ MQOPER\_EXCLUDED\_PUBLISH)

在 PCF 群組 作業 中針對發佈/捨棄發佈/排除發佈 (MQOPER\_PUBLISH/MQOPER\_DISCARDED\_PUBLISH/  
MQOPER\_EXCLUDED\_PUBLISH) 作業類型傳回的其他活動報告訊息資料參數 (已遞送、捨棄或排除發佈/訂  
閱訊息)。

#### **SubId**

說明: 訂閱 ID。  
ID: MQBACF\_SUB\_ID。  
資料類型: MQCFBS。

包含在 PCF 群組 作業。  
中：  
已傳回： 始終。

### **SubLevel**

說明： 訂閱層次。  
ID： MQIACF\_SUB\_LEVEL。  
資料類型： MQCFIN。  
包含在 PCF 群組 作業。  
中：  
已傳回： 始終。

### **Feedback**

說明： 捨棄訊息的原因。  
ID： MQIACF\_FEEDBACK。  
資料類型： MQCFIN。  
包含在 PCF 群組 作業。  
中：  
已傳回： 如果因為訊息未遞送至訂閱者而捨棄該訊息，或因為訂閱者已被排除而未遞送該訊息。

「發佈」作業 MQOPER\_PUBLISH 提供遞送至特定訂閱者之訊息的相關資訊。此作業說明向前訊息的元素，這些元素可能已從相關聯的「放置」作業中說明的訊息變更。與「放置」作業類似，它包含訊息群組 MQGACF\_MESSAGE 及 MQMD 群組 MQGACF\_MQMD。不過，此 MQMD 群組只包含下列欄位，訂閱者可以置換這些欄位：格式、優先順序、持續性、*MsgId*、*CorrelId*、*UserIdentifier*、*AccountingToken*、*ApplIdentity* 資料。

作業資訊中包括訂閱者的 *SubId* 和 *SubLevel*。您可以搭配使用 *SubID* 與 MQCMD\_INQUIRE\_SUBSCRIBER PCF 指令，以擷取訂閱者的所有其他屬性。

「捨棄的發佈」作業 MQOPER\_DISCARDED\_PUBLISH 類似於在點對點傳訊中未遞送訊息時所使用的「捨棄」作業。如果明確要求訊息不遞送至本端目的地，且此訂閱者指定本端目的地，則訊息不會遞送至訂閱者。如果將訊息取得至目的地佇列時發生問題（例如，因為佇列已滿），則也會將訊息視為未遞送。

「捨棄的發佈」作業中的資訊與「發佈」作業中的資訊相同，新增意見欄位會提供未遞送訊息的原因。此回饋欄位包含與 MQOPER\_DISCARD 作業共用的 MQFB\_\* 或 MQRC\_\* 值。捨棄發佈而不是排除發佈的原因與捨棄放置的原因相同。

「排除的發佈」作業 MQOPER\_EXCLUDED\_PUBLISH 提供已考量遞送訊息之訂閱者的相關資訊，因為訂閱者所訂閱的主題符合相關聯「放置」作業的主題，但訊息未遞送至訂閱者，因為其他選取準則不符合正在放置至主題的訊息。與「捨棄的發佈」作業一樣，意見欄位提供排除此訂閱之原因的相關資訊。不過，與「捨棄的發佈」作業不同，未提供任何訊息相關資訊，因為未針對此訂閱者產生任何訊息。

放置/放置回覆/放置報告 (MQOPER\_PUT/MQOPER\_PUT\_REPLY/MQOPER\_PUT\_REPORT)  
在 PCF 群組 作業 中針對「放置/放置回覆/放置報告」(MQOPER\_PUT/MQOPER\_PUT\_REPLY/  
MQOPER\_PUT\_REPORT) 作業類型 (訊息、回覆訊息或報告訊息已放入佇列) 傳回的其他活動報告訊息資料  
參數。

### **QName**

說明： 已開啟的佇列名稱。

ID:	MQCA_Q_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_NAME_LENGTH
已傳回:	一律，除了一個異常狀況之外: 如果「放置」作業是針對發佈活動內所包含的主題，則不會傳回。

### ***ResolvedQName***

說明:	開啟的佇列解析成的名稱。
ID:	MQCACF_RESOLVED_Q_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_NAME_LENGTH
已傳回:	何時可以解析開啟的佇列。 如果「放置」作業是針對發佈活動內包含的主題，則不會傳回。

### ***RemoteQName***

說明:	在遠端佇列管理程式上已知的已開啟佇列名稱。
ID:	MQCA_REMOTE_Q_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_NAME_LENGTH
已傳回:	如果開啟的佇列是遠端佇列。 如果「放置」作業是針對發佈活動內包含的主題，則不會傳回。

### ***RemoteQMgrName***

說明:	在其中定義遠端佇列的遠端佇列管理程式名稱。
ID:	MQCA_REMOTE_Q_MGR_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_MGR_NAME_LENGTH
已傳回:	如果開啟的佇列是遠端佇列。 如果「放置」作業是針對發佈活動內包含的主題，則不會傳回。

### ***TopicString***

說明:	要放置訊息的完整主題字串。
ID:	MQCA_TOPIC_STRING。
資料類型:	MQCFST。

包含在 PCF 群組 作業。  
中：  
已傳回： 如果「放置」作業是針對發佈活動內包含的主題。

### **Feedback**

說明： 將訊息放置在無法傳送郵件的佇列上的原因。  
ID： MQIACF\_FEEDBACK。  
資料類型： MQCFIN。  
包含在 PCF 群組 作業。  
中：  
已傳回： 如果訊息已放置在無法傳送郵件的佇列上。

### 接收 (MQOPER\_RECEIVE)

針對「接收 (MQOPER\_RECEIVE)」作業類型 (在通道上收到訊息)，在 PCF 群組 作業 中傳回的其他活動報告訊息資料參數。

#### **ChannelName**

說明： 接收訊息的通道名稱。  
ID： MQCACH\_CHANNEL\_NAME。  
資料類型： MQCFST。  
包含在 PCF 群組 作業。  
中：  
長度上限： MQ\_CHANNEL\_NAME\_LENGTH  
已傳回： 始終。

#### **ChannelType**

說明： 接收訊息的通道類型。  
ID： MQIACH\_CHANNEL\_TYPE。  
資料類型： MQCFIN。  
包含在 PCF 群組 作業。  
中：  
已傳回： 始終。

#### **RemoteQMgrName**

說明： 從中接收訊息的佇列管理程式名稱。  
ID： MQCA\_REMOTE\_Q\_MGR\_NAME。  
資料類型： MQCFST。  
包含在 PCF 群組 作業。  
中：  
長度上限： MQ\_Q\_MGR\_NAME\_LENGTH  
已傳回： 始終。

### 傳送 (MQOPER\_SEND)

針對「傳送 (MQOPER\_SEND)」作業類型 (已在通道上傳送訊息)，在 PCF 群組 作業 中傳回的其他活動報告訊息資料參數。

### **ChannelName**

說明:	傳送訊息的通道名稱。
ID:	MQCACH_CHANNEL_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_CHANNEL_NAME_LENGTH。
已傳回:	始終。

### **ChannelType**

說明:	傳送訊息的通道類型。
ID:	MQIACH_CHANNEL_TYPE。
資料類型:	MQCFIN。
包含在 PCF 群組中:	作業。
已傳回:	始終。

### **XmitQName**

說明:	從中擷取訊息的傳輸佇列。
ID:	MQCACH_XMIT_Q_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_NAME_LENGTH。
已傳回:	始終。

### **RemoteQMgrName**

說明:	訊息傳送至其中的遠端佇列管理程式名稱。
ID:	MQCA_REMOTE_Q_MGR_NAME。
資料類型:	MQCFST。
包含在 PCF 群組中:	作業。
長度上限:	MQ_Q_MGR_NAME_LENGTH
已傳回:	始終。

## **追蹤路徑訊息參照**

請利用這個頁面來取得追蹤路徑訊息格式的概觀。追蹤路徑訊息資料包括的參數說明追蹤路徑訊息所造成的活動

### **追蹤路徑訊息格式**

追蹤路徑訊息是包含訊息描述子及訊息資料的標準 IBM MQ 訊息。訊息資料包含在追蹤路徑訊息上執行的活動相關資訊，因為它已透過佇列管理程式網路遞送。

追蹤路徑訊息包含下列資訊:

### 訊息描述子

MQMD 結構，其格式欄位設為 MQFMT\_ADMIN 或 MQFMT\_embedded\_pcf。

### 訊息資料

由下列任一項組成：

- 如果格式設為 MQFMT\_ADMIN，則為 PCF 標頭 (MQCFH) 及追蹤路徑訊息資料，或
- 內嵌 PCF 標頭 (MQEPH)、追蹤路徑訊息資料及其他使用者指定訊息資料 (如果格式設為 MQFMT\_EMBEDDED\_PCF)。

使用 IBM MQ 顯示路徑應用程式來產生追蹤路徑訊息時，格式會設為 MQFMT\_ADMIN。

追蹤路徑訊息資料的內容由 TraceRoute PCF 群組中的累計參數決定，如下所示：

- 如果累計設為 MQROUTE\_ACCUMULATE\_NONE，則追蹤路徑訊息資料會包含 TraceRoute PCF 群組。
- 如果累計設為 MQROUTE\_ACCUMULATE\_IN\_MSG 或 MQROUTE\_ACCUMULATE\_AND\_REPLY，則追蹤路徑訊息資料會包含 TraceRoute PCF 群組及零個以上活動 PCF 群組。

第 109 頁的表 20 顯示追蹤路徑訊息的結構。

MQMD 結構	內嵌 PCF 標頭 MQEPH 結構	追蹤-遞送訊息資料
結構 ID 結構版本 報告選項 訊息類型 有效期限 意見 編碼 編碼字集 ID 訊息格式 優先順序 持續性 訊息 ID 相關性 ID 取消次數 回覆目的地佇列 回覆目的地佇列管理程式 使用者 ID 帳戶記號 應用程式身分資料 應用程式類型 應用程式名稱 放置日期 放置時間 應用程式原始資料 群組 ID 訊息序號 偏移 訊息旗標 原始長度	結構 ID 結構版本 結構長度 編碼 編碼字集 ID 訊息格式 旗標 PCF 標頭 (MQCFH) 結構類型 結構長度 結構版本 指令 ID 訊息序號 控制選項 完成碼 原因碼 參數計數	TraceRoute 明細 記錄的活動 未記錄的活動 中斷計數 活動數上限 累計 交付

### 追蹤路徑訊息 MQMD (訊息描述子)

請利用這個頁面來檢視追蹤路徑訊息的 MQMD 結構所包含的值

### **StrucId**

說明：結構 ID。  
資料類型：MQCHAR4。  
值：MQMD\_STRUC\_ID。

### **Version**

說明：結構版本號碼。  
資料類型：MQLONG。  
值：**MQMD\_VERSION\_1**。

### **Report**

說明：報告訊息的選項。  
資料類型：MQLONG。  
值：根據需求設定。一般報告選項如下：

#### **MQRO\_DISCARD\_MSG**

到達本端佇列時會捨棄訊息。

#### **MQRO\_PASS\_DISCARD\_AND\_EXPIRY**

每個回應 (活動報告或追蹤路徑回覆訊息) 都會設定報告選項 MQRO\_DISCARD\_MSG，且會傳遞剩餘期限。這可確保回應不會無限期保留在佇列管理程式網路中。

### **MsgType**

說明：訊息類型。  
資料類型：MQLONG。  
值：如果 TraceRoute 群組中的 累計 參數指定為 MQROUTE\_ACCUMULATE\_AND\_REPLY，則訊息類型為 MQMT\_REQUEST  
否則：  
**MQMT\_DATAGRAM**。

### **Expiry**

說明：訊息生命期限。  
資料類型：MQLONG。  
值：根據需求設定。此參數可用來確保追蹤路徑訊息不會無限期留在佇列管理程式網路中。

### **Feedback**

說明：意見或原因碼。  
資料類型：MQLONG。  
值：**MQFB\_NONE**。

### **Encoding**

說明：訊息資料的數值編碼。  
資料類型：MQLONG。

值： 適當設定。

### **CodedCharSetId**

說明： 訊息資料的字集 ID。

資料類型： MQLONG。

值： 適當設定。

### **Format**

說明： 訊息資料的格式名稱

資料類型： MQCHAR8。

值： **MQFMT\_ADMIN**  
管理訊息。沒有使用者資料遵循 *TraceRoute* PCF 群組。  
**MQFMT\_EMBEDDED\_PCF**  
內嵌 PCF 訊息。使用者資料遵循 *TraceRoute* PCF 群組。

### **Priority**

說明： 訊息優先順序。

資料類型： MQLONG。

值： 根據需求設定。

### **Persistence**

說明： 訊息持續性。

資料類型： MQLONG。

值： 根據需求設定。

### **MsgId**

說明： 訊息 ID。

資料類型： MQBYTE24。

值： 根據需求設定。

### **CorrelId**

說明： 相關性 ID。

資料類型： MQBYTE24。

值： 根據需求設定。

### **BackoutCount**

說明： 取消計數器。

資料類型： MQLONG。

值： 0。

### **ReplyToQ**

說明： 回覆佇列的名稱。

資料類型： MQCHAR48。

值： 根據需求設定。  
如果 *MsgType* 設為 MQMT\_REQUEST，或者 報告 已設定任何報告產生選項，則此參數必須為非空白。

### ***ReplyToQMgr***

說明： 回覆佇列管理程式的名稱。  
資料類型： MQCHAR48.  
值： 根據需求設定。

### ***UserIdentifier***

說明： 產生訊息之應用程式的使用者 ID。  
資料類型： MQCHAR12.  
值： 設定為正常。

### ***AccountingToken***

說明： 帳戶記號，可讓應用程式對因訊息而完成的工作收取費用。  
資料類型： MQBYTE32.  
值： 設定為正常。

### ***ApplIdentityData***

說明： 與身分相關的應用程式資料。  
資料類型： MQCHAR32.  
值： 設定為正常。

### ***PutApplType***

說明： 放置訊息的應用程式類型。  
資料類型： MQLONG。  
值： 設定為正常。

### ***PutApplName***

說明： 放置訊息的應用程式名稱。  
資料類型： MQCHAR28.  
值： 設定為正常。

### ***PutDate***

說明： 放置訊息的日期。  
資料類型： MQCHAR8.  
值： 設定為正常。

### ***PutTime***

說明： 放置訊息的時間。  
資料類型： MQCHAR8.

值: 設定為正常。

### **ApplOriginData**

說明: 與來源相關的應用程式資料。

資料類型: MQCHAR4.

值: 設為正常 ..

### **追蹤路徑訊息 MQEPH (內嵌 PCF 標頭)**

請利用這個頁面來檢視追蹤路徑訊息的 MQEPH 結構所包含的值

MQEPH 結構包含追蹤路徑訊息的訊息資料所隨附的 PCF 資訊，以及它之後的應用程式訊息資料的說明。只有在其他使用者訊息資料遵循 TraceRoute PCF 群組時，才會使用 MQEPH 結構。

對於追蹤路徑訊息，MQEPH 結構包含下列值:

#### **StrucId**

說明: 結構 ID。

資料類型: MQCHAR4.

值: MQEPH\_STRUC\_ID。

#### **Version**

說明: 結構版本號碼。

資料類型: MQLONG。

值: MQEPH\_VERSION\_1.

#### **StrucLength**

說明: 結構長度。

資料類型: MQLONG。

值: 結構的總長度，包括其後的 PCF 參數結構。

#### **Encoding**

說明: 遵循最後一個 PCF 參數結構之訊息資料的數值編碼。

資料類型: MQLONG。

值: 訊息資料的編碼。

#### **CodedCharSetId**

說明: 遵循最後一個 PCF 參數結構之訊息資料的字集 ID。

資料類型: MQLONG。

值: 訊息資料的字集。

#### **Format**

說明: 遵循最後一個 PCF 參數結構之訊息資料的格式名稱。

資料類型: MQCHAR8.

值: 訊息資料的格式名稱。

## Flags

說明:	指定結構屬性或控制其處理的旗標。
資料類型:	MQLONG。
值:	<b>MQEPH_NONE</b> 未指定任何旗標。 <b>MQEPH_CCSID_EMBEDDED</b> 指定在每一個結構中的 <i>CodedCharSetId</i> 欄位內個別指定包含字元資料之參數的字集。

## PCFHeader

說明:	可程式指令格式標頭
資料類型:	MQCFH。
值:	請參閱 <a href="#">第 114 頁的『追蹤路徑訊息 MQCFH (PCF 標頭)』</a> 。

## 追蹤路徑訊息 MQCFH (PCF 標頭)

請利用這個頁面來檢視 MQCFH 結構所包含的追蹤路徑訊息 PCF 值

若為追蹤路徑訊息，MQCFH 結構包含下列值:

### Type

說明:	識別訊息內容的結構類型。
資料類型:	MQLONG。
值:	<b>MQCFT_TRACE_ROUTE</b> 訊息是追蹤路徑訊息。

### StrucLength

說明:	結構長度。
資料類型:	MQLONG。
值:	<b>MQCFH_STRUC_LENGTH</b> MQCFH 結構的長度 (以位元組為單位)。

### Version

說明:	結構版本號碼。
資料類型:	MQLONG。
值:	MQCFH_VERSION_3

### Command

說明:	指令 ID。這會識別訊息的種類。
資料類型:	MQLONG。
值:	<b>MQCMD_TRACE_ROUTE</b> 追蹤路徑訊息。

### MsgSeqNumber

說明:	訊息序號。這是相關訊息群組內的訊息序號。
資料類型:	MQLONG。

值: 1.

### **Control**

說明: 控制選項。  
資料類型: MQLONG。  
值: MQCFC\_LAST。

### **CompCode**

說明: 完成碼。  
資料類型: MQLONG。  
值: MQCC\_OK。

### **Reason**

說明: 原因碼定義完成碼。  
資料類型: MQLONG。  
值: MQRC\_NONE。

### **ParameterCount**

說明: 參數結構計數。這是遵循 MQCFH 結構的參數結構數目。群組結構 (MQCFGR) 及其包含的參數結構只會計為一個結構。  
資料類型: MQLONG。  
值: 1 或以上。

## **追蹤-遞送訊息資料**

請利用這個頁面來檢視組成追蹤路徑訊息資料的 *TraceRoute* PCF 群組部分的參數

追蹤路徑訊息資料的內容取決於 *TraceRoute* PCF 群組中的 累計 參數。追蹤路徑訊息資料包含 *TraceRoute* PCF 群組，以及零個以上 活動 PCF 群組。本主題詳述 *TraceRoute* PCF 群組。如需 活動 PCF 群組的詳細資料，請參閱相關資訊。

Trace-route 訊息資料包含下列參數:

### **TraceRoute**

說明: 指定追蹤路徑訊息屬性的分組參數。對於追蹤路徑訊息，可以變更其中部分參數，以控制其處理方式。

ID: MQGACF\_TRACE\_ROUTE。

資料類型: MQCFGR。

包含在 PCF 群組中: 無。

群組中的參數:

- Detail*
- RecordedActivities*
- UnrecordedActivities*
- DiscontinuityCount*
- MaxActivities*
- Accumulate*
- Forward*
- Deliver*

## **Detail**

說明:	將為活動記錄的詳細層次。
ID:	MQIACF_ROUTE_DETAIL。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>TraceRoute</i> 。
值:	<b>MQROUTE_DETAIL_LOW</b> 記錄使用者撰寫的應用程式所執行的活動。 <b>MQROUTE_DETAIL_MEDIUM</b> 會記錄 MQROUTE_DETAIL_LOW 中指定的活動。此外，還會記錄 MCA 所執行的活動。 <b>MQROUTE_DETAIL_HIGH</b> 會記錄 MQROUTE_DETAIL_LOW 及 MQROUTE_DETAIL_MEDIUM 中指定的活動。MCA 不會在此詳細程度上記錄任何進一步的活動資訊。此選項僅適用於使用者撰寫的應用程式，這些應用程式會記錄進一步的活動資訊。

## **RecordedActivities**

說明:	追蹤路徑訊息所造成的活動數目，其中記錄了資訊。
ID:	MQIACF_RECORD_ACTIVITIES。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>TraceRoute</i> 。

## **UnrecordedActivities**

說明:	追蹤路徑訊息所造成的活動數目，其中未記錄資訊。
ID:	MQIACF_UNRECORD_ACTIVITIES。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>TraceRoute</i> 。

## **DiscontinuityCount**

說明:	從不支援追蹤路徑傳訊的佇列管理程式收到追蹤路徑訊息的次數。
ID:	MQIACF_DISCONTINUITY_COUNT。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>TraceRoute</i> 。

## **MaxActivities**

說明:	在停止處理追蹤路徑訊息之前，追蹤路徑訊息可涉及的活動數目上限。
ID:	MQIACF_MAX_ACTIVITIES。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>TraceRoute</i> 。

值: 正整數  
活動數目上限。  
**MQROUTE\_UNLIMITED\_ACTIVITIES**  
活動數目無限制。

### **Accumulate**

說明: 指定是否在追蹤路徑訊息內累計活動資訊, 以及是否在捨棄追蹤路徑訊息或將其置於非傳輸佇列之前產生包含累計活動資訊的回覆訊息。

ID: MQIACF\_ROUTE\_ACCUMULATION。

資料類型: MQCFIN。

包含在 PCF 群組中: *TraceRoute*。

值: **MQROUTE\_ACCUMULATE\_NONE**  
活動資訊不會累積在追蹤路徑訊息的訊息資料中。

**MQROUTE\_ACCUMULATE\_IN\_MSG**  
活動資訊會累積在追蹤路徑訊息的訊息資料中。

**MQROUTE\_ACCUMULATE\_AND\_REPLY**  
活動資訊會累積在追蹤路徑訊息的訊息資料中, 且會產生追蹤路徑回覆訊息。

### **Forward**

說明: 指定追蹤路徑訊息可以轉遞至其中的佇列管理程式。在決定是否將訊息轉遞至遠端佇列管理程式時, 佇列管理程式會使用 轉遞 中說明的演算法。

ID: MQIACF\_ROUTE\_FORWARDING。

資料類型: MQCFIN。

包含在 PCF 群組中: *TraceRoute*。

值: **MQROUTE\_FORWARD\_IF\_SUPPORTED**  
追蹤路徑訊息只會轉遞至將允許使用 *TraceRoute* 群組中 遞送 參數值的佇列管理程式。

**MQROUTE\_FORWARD\_ALL**  
不論是否允許使用 遞送 參數的值, 都會將追蹤路徑訊息轉遞至任何佇列管理程式。

### **Deliver**

說明: 指定追蹤路徑訊息順利到達目的地佇列時要採取的動作。

ID: MQIACF\_ROUTE\_DELIVERY。

資料類型: MQCFIN。

包含在 PCF 群組中: *TraceRoute*。

值: **MQROUTE\_DELIVER\_YES**  
到達時, 追蹤路徑訊息會放置在目標佇列上。在目標佇列上執行破壞性取得的任何應用程式都可以接收追蹤路徑訊息。

**MQROUTE\_DELIVER\_NO**  
到達時, 會捨棄追蹤路徑訊息。

## 追蹤路徑回覆訊息參照

請利用這個頁面來取得追蹤路徑回覆訊息格式的概觀。 追蹤路徑回覆訊息資料與為其產生追蹤路徑訊息的追蹤路徑訊息資料重複

### 追蹤路徑回覆訊息格式

追蹤路徑回覆訊息是包含訊息描述子及訊息資料的標準 IBM MQ 訊息。 訊息資料包含在追蹤路徑訊息上執行的活動相關資訊，因為它已透過佇列管理程式網路遞送。

追蹤路徑回覆訊息包含下列資訊：

#### 訊息描述子

MQMD 結構

#### 訊息資料

PCF 標頭 (MQCFH) 及追蹤路徑回覆訊息資料

追蹤路徑回覆訊息資料由一或多個 活動 PCF 群組組成。

當追蹤路徑訊息到達其目標佇列時，可以產生追蹤路徑回覆訊息，其中包含追蹤路徑訊息中的活動資訊副本。 追蹤路徑回覆訊息將遞送至回覆目的地佇列或系統佇列。

第 118 頁的表 21 顯示追蹤路徑回覆訊息的結構，包括僅在特定條件下傳回的參數。

MQMD 結構	PCF 標頭 MQCFH 結構	追蹤-遞送回覆訊息資料
結構 ID 結構版本 報告選項 訊息類型 有效期限 意見 編碼 編碼字集 ID 訊息格式 優先順序 持續性 訊息 ID 相關性 ID 取消次數 回覆目的地佇列 回覆目的地佇列管理程式 使用者 ID 帳戶記號 應用程式身分資料 應用程式類型 應用程式名稱 放置日期 放置時間 應用程式原始資料 群組 ID 訊息序號 偏移 訊息旗標 原始長度	PCF 標頭 (MQCFH) 結構類型 結構長度 結構版本 指令 ID 訊息序號 控制選項 完成碼 原因碼 參數計數	活動 活動應用程式名稱 活動應用程式類型 活動說明 作業 作業類型 作業日期 作業時間 訊息 訊息長度 MQMD EmbeddedMQMD 佇列管理程式名稱 佇列共用群組名稱 佇列名稱 <sup>1 2 3</sup> 已解析佇列名稱 <sup>1 3</sup> 遠端佇列名稱 <sup>3</sup> 遠端佇列管理程式- 名稱 <sup>2 3 4 5</sup> 意見 <sup>2</sup> 通道名稱 <sup>4 5</sup> 通道類型 <sup>4 5</sup> 傳輸佇列名稱 <sup>5</sup> TraceRoute 明細 記錄的活動 未記錄的活動 中斷計數 活動數上限 累計 交付

表 21: 追蹤路徑回覆訊息格式 (繼續)

MQMD 結構	PCF 標頭 MQCFH 結構	追蹤-遞送回覆訊息資料
<p>註:</p> <ol style="list-style-type: none"> <li>1. 針對「取得」及「瀏覽」作業傳回。</li> <li>2. 針對捨棄作業傳回。</li> <li>3. 針對「放置」、「放置回覆」及「放置報告」作業傳回。</li> <li>4. 針對「接收」作業傳回。</li> <li>5. 針對「傳送」作業傳回。</li> </ol>		

### 追蹤路徑回覆訊息 MQMD (訊息描述子)

請利用這個頁面來檢視追蹤路徑回覆訊息的 MQMD 結構所包含的值

對於追蹤路徑回覆訊息，MQMD 結構包含 [活動報告訊息描述子](#) 中說明的參數。追蹤路徑回覆訊息描述子中的部分參數值與活動報告訊息描述子中的部分參數值不同，如下所示：

#### MsgType

說明： 訊息類型。  
 資料類型： MQLONG。  
 值： **MQMT\_REPLY**

#### Feedback

說明： 意見或原因碼。  
 資料類型： MQLONG。  
 值： **MQFB\_NONE**

#### Encoding

說明： 訊息資料的數值編碼。  
 資料類型： MQLONG。  
 值： 從 trace-route 訊息描述子複製。

#### CodedCharSetId

說明： 訊息資料的字集 ID。  
 資料類型： MQLONG。  
 值： 從 trace-route 訊息描述子複製。

#### Format

說明： 訊息資料的格式名稱  
 資料類型： MQCHAR8。  
 值： **MQFMT\_ADMIN**  
 管理訊息。

### 追蹤路徑回覆訊息 MQCFH (PCF 標頭)

請利用這個頁面來檢視 MQCFH 結構所包含追蹤路徑回覆訊息的 PCF 值

追蹤路徑回覆訊息的 PCF 標頭 (MQCFH) 與追蹤路徑訊息相同。

## 追蹤-遞送回覆訊息資料

追蹤路徑回覆訊息資料與為其產生追蹤路徑訊息的追蹤路徑訊息資料重複

追蹤路徑回覆訊息資料包含一或多個活動群組。這些參數在 [第 91 頁的『活動報告訊息資料』](#) 中說明。

## 帳戶及統計資料訊息

佇列管理程式會產生結算及統計資料訊息，以記錄 IBM MQ 應用程式所執行 MQI 作業的相關資訊，或記錄 IBM MQ 系統中所發生活動的相關資訊。

### 帳戶訊息

結算訊息用來記錄 IBM MQ 應用程式所執行 MQI 作業的相關資訊，請參閱 [第 120 頁的『帳戶訊息』](#)。

### 統計資料訊息

統計資料訊息用來記錄 IBM MQ 系統中發生的活動相關資訊，請參閱 [第 123 頁的『統計資料訊息』](#)。  
統計資料訊息中所記錄的部分活動與內部佇列管理程式作業相關。

 這裡說明的帳戶訊息及統計資料訊息無法在 IBM MQ for z/OS 上使用，但可透過「系統管理機能 (SMF)」使用同等功能。

結算及統計資料訊息會遞送至兩個系統佇列之一。使用者應用程式可以從這些系統佇列擷取訊息，並將記錄的資訊用於各種用途：

- 應用程式資源使用的帳戶。
- 記錄應用程式活動。
- 產能規劃。
- 偵測佇列管理程式網路中的問題。
- 協助判斷佇列管理程式網路中問題的原因。
- 提高佇列管理程式網路的效率。
- 熟悉佇列管理程式網路的執行。
- 請確認佇列管理程式網路正確執行中。

### 相關概念

[第 292 頁的『使用系統管理機能』](#)

您可以使用 SMF 來收集統計資料及帳戶資訊。如果要使用 SMF，必須在 z/OS 和 IBM MQ 中設定某些參數。

## 帳戶訊息

結算訊息會記錄 IBM MQ 應用程式所執行 MQI 作業的相關資訊。帳戶訊息是包含許多 PCF 結構的 PCF 訊息。

當應用程式切斷與佇列管理程式的連線時，會產生結算訊息並遞送至系統結算佇列 (SYSTEM.ADMIN.ACCOUNTING.QUEUE)。對於長時間執行的 IBM MQ 應用程式，會產生中間結算訊息，如下所示：

- 自建立連線以來的時間超出配置的間隔。
- 自前次中間結算訊息以來的時間超出配置的間隔時。

帳戶訊息分為下列種類：

### MQI 結算訊息

MQI 結算訊息包含使用佇列管理程式連線所進行的 MQI 呼叫數相關資訊。

### 佇列帳戶訊息

佇列結算訊息包含使用佇列管理程式的連線 (依佇列分組) 所進行的 MQI 呼叫數相關資訊。

每一個佇列結算訊息最多可以包含 100 筆記錄，每一筆記錄都與應用程式所執行的活動相關 (與特定佇列相關)。

只會記錄本端佇列的帳戶訊息。如果應用程式對別名佇列發出 MQI 呼叫，則會針對基本佇列記錄結算資料，而針對遠端佇列，則會針對傳輸佇列記錄結算資料。

## 相關參考

第 137 頁的『MQI 結算訊息資料』  
請利用這個頁面來檢視 MQI 結算訊息的結構

第 147 頁的『佇列帳戶訊息資料』  
請利用這個頁面來檢視佇列帳戶訊息的結構

## 帳戶訊息格式

帳戶訊息包含一組由訊息描述子和訊息資料組成的 PCF 欄位。

### 訊息描述子

- 結算訊息 MQMD (訊息描述子)

### 帳戶訊息資料

- 帳戶訊息 MQCFH (PCF 標頭)
- 一律傳回的帳戶訊息資料
- 傳回的結算訊息資料 (如果有的話)

結算訊息 MQCFH (PCF 標頭) 包含應用程式的相關資訊，以及記錄結算資料的間隔。

統計訊息資料包含儲存統計資訊的 PCF 參數。結算訊息的內容視訊息種類而定，如下所示：

### MQI 結算訊息

MQI 結算訊息資料由許多 PCF 參數組成，但沒有 PCF 群組。

### 佇列帳戶訊息

佇列結算訊息資料由許多 PCF 參數組成，範圍為 1 到 100 *QAccountingData* PCF 群組。

每個已收集帳戶資料的佇列都有一個 *QAccountingData* PCF 群組。如果應用程式存取超過 100 個佇列，則會產生多則結算訊息。每一則訊息都會相應地更新 MQCFH (PCF 標頭) 中的 *SeqNumber*，而序列中的最後一則訊息會將 MQCFH 中的 *Control* 參數指定為 MQCFC\_LAST。

## 帳戶資訊集合

使用佇列及佇列管理程式屬性來控制帳戶資訊的收集。您也可以使用 MQCONNX 選項來控制連線層次的集合。

### 控制 MQI 結算資訊的收集

使用佇列管理程式屬性 ACCTMQI 來控制 MQI 結算資訊的收集。

若要變更此屬性的值，請使用 MQSC 指令 ALTER QMGR，並指定參數 ACCTMQI。只有在啟用帳戶之後開始的連線才會產生帳戶訊息。ACCTMQI 參數可以具有下列值：

#### 開啟

會針對佇列管理程式的每個連線收集 MQI 結算資訊。

#### 關閉

未收集 MQI 帳戶資訊。這是預設值。

例如，若要啟用 MQI 結算資訊收集，請使用下列 MQSC 指令：

```
ALTER QMGR ACCTMQI(ON)
```

### 佇列帳戶資訊

使用佇列屬性 ACCTQ 及佇列管理程式屬性 ACCTQ 來控制佇列結算資訊的收集。

若要變更佇列屬性的值，請使用 MQSC 指令 ALTER QLOCAL，並指定參數 ACCTQ。只有在啟用帳戶之後開始的連線才會產生帳戶訊息。請注意，此值的變更僅對在變更屬性之後發生的佇列管理程式連線有效。

佇列屬性 ACCTQ 可以具有下列值：

#### 開啟

每個開啟佇列之佇列管理程式的連線都會收集此佇列的佇列結算資訊。

## 關閉

不收集此佇列的佇列帳戶資訊。

## QMGR

根據佇列管理程式屬性 ACCTQ 的值來控制此佇列的佇列帳戶資訊收集。這是預設值。

若要變更佇列管理程式屬性的值，請使用 MQSC 指令 ALTER QMGR 並指定參數 ACCTQ。佇列管理程式屬性 ACCTQ 可以具有下列值：

## 開啟

針對佇列屬性 ACCTQ 設為 QMGR 的佇列收集佇列帳戶資訊。

## 關閉

對於將佇列屬性 ACCTQ 設為 QMGR 的佇列，不會收集佇列帳戶資訊。這是預設值。

## 無

不論佇列屬性 ACCTQ 為何，都會停用所有佇列的佇列統計資訊收集。

如果佇列管理程式屬性 ACCTQ 設為 NONE，則不論佇列屬性 ACCTQ 為何，都會停用所有佇列的佇列結算資訊收集。

例如，若要啟用佇列 Q1 的帳戶資訊收集，請使用下列 MQSC 指令：

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

若要針對將佇列屬性 ACCTQ 指定為 QMGR 的所有佇列啟用帳戶資訊收集，請使用下列 MQSC 指令：

```
ALTER QMGR ACCTQ(ON)
```

用於控制帳戶資訊收集的 *MQCONN* 選項

在 *MQCONN* 呼叫中使用 **ConnectOpts** 參數可透過置換佇列管理程式屬性 ACCTMQI 及 ACCTQ 的有效值，來修改連線層次中 MQI 及佇列結算資訊的收集。

**ConnectOpts** 參數可以具有下列值：

### MQCNO\_ACCOUNTING\_MQI\_ENABLED

如果佇列管理程式屬性 ACCTMQI 的值指定為 OFF，則會針對此連線啟用 MQI 結算。這相當於指定為 ON 的佇列管理程式屬性 ACCTMQI。

如果佇列管理程式屬性 ACCTMQI 的值未指定為 OFF，則此屬性沒有作用。

### MQCNO\_ACCOUNTING\_MQI\_DISABLED

如果佇列管理程式屬性 ACCTMQI 的值指定為 ON，則會停用此連線的 MQI 結算。這相當於指定為 OFF 的佇列管理程式屬性 ACCTMQI。

如果佇列管理程式屬性 ACCTMQI 的值未指定為 ON，則此屬性沒有作用。

### MQCNO\_ACCOUNTING\_Q\_ENABLED

如果佇列管理程式屬性 ACCTQ 的值指定為 OFF，則會針對此連線啟用佇列結算。所有 ACCTQ 指定為 QMGR 的佇列都會啟用佇列統計。這相當於指定為 ON 的佇列管理程式屬性 ACCTQ。

如果佇列管理程式屬性 ACCTQ 的值未指定為 OFF，則此屬性沒有作用。

### MQCNO\_ACCOUNTING\_Q\_DISABLED

如果佇列管理程式屬性 ACCTQ 的值指定為 ON，則會停用此連線的佇列帳戶。這相當於指定為 OFF 的佇列管理程式屬性 ACCTQ。

如果佇列管理程式屬性 ACCTQ 的值未指定為 ON，則此屬性沒有作用。

依預設會停用這些置換。若要啟用它們，請將佇列管理程式屬性 ACCTCONO 設為 ENABLED。若要啟用個別連線的帳戶置換，請使用下列 MQSC 指令：

```
ALTER QMGR ACCTCONO(ENABLED)
```

應用程式斷線時產生帳戶訊息

當應用程式與佇列管理程式中斷連線時，會產生結算訊息。對於長時間執行的 IBM MQ 應用程式，也會寫入中間結算訊息。

當應用程式中斷連線時，會以下列其中一種方式產生帳戶訊息：

- 應用程式發出 MQDISC 呼叫
- 佇列管理程式可確認應用程式已終止

當自建立連線以來的間隔，或自寫入的最後一則中間結算訊息超出配置の間隔時，會針對長時間執行的 IBM MQ 應用程式寫入中間結算訊息。佇列管理程式屬性 ACCTINT 會指定時間 (以秒為單位)，在此之後可以自動寫入中間結算訊息。只有在應用程式與佇列管理程式互動時，才會產生結算訊息，因此長時間保持連接至佇列管理程式而不執行 MQI 要求的應用程式，不會產生結算訊息，直到在結算間隔完成之後執行第一個 MQI 要求為止。

預設統計間隔為 1800 秒 (30 分鐘)。例如，若要將帳戶間隔變更為 900 秒 (15 分鐘)，請使用下列 MQSC 指令：

```
ALTER QMGR ACCTINT(900)
```

## 統計資料訊息

統計資料訊息會記錄 IBM MQ 系統中發生之活動的相關資訊。統計資料訊息是包含許多 PCF 結構的 PCF 訊息。

統計資料訊息會遞送至系統佇列 (SYSTEM.ADMIN.STATISTICS.QUEUE)。

統計資料訊息分為下列種類：

### MQI 統計資料訊息

MQI 統計資料訊息包含在已配置間隔期間所進行的 MQI 呼叫數相關資訊。例如，資訊可以包括佇列管理程式發出的 MQI 呼叫數。

### 佇列統計資料訊息

佇列統計資料訊息包含已配置間隔期間佇列活動的相關資訊。此資訊包括放置在佇列上及從佇列擷取的訊息數，以及佇列所處理的位元組總數。

每一個佇列統計資料訊息最多可以包含 100 筆記錄，每一筆記錄與針對其收集統計資料的每個佇列的活動相關。

只會記錄本端佇列的統計資料訊息。如果應用程式對別名佇列發出 MQI 呼叫，則會針對基本佇列記錄統計資料，而針對遠端佇列，則會針對傳輸佇列記錄統計資料。

### 通道統計資料訊息

通道統計資料訊息包含已配置間隔期間通道活動的相關資訊。例如，資訊可能是通道傳送的訊息數，或通道傳送的位元組數。

每一個通道統計資料訊息最多包含 100 筆記錄，每一筆記錄與針對其收集統計資料的每個通道的活動相關。

## 相關參考

第 124 頁的『控制 MQI 統計資料收集』

使用佇列管理程式屬性 STATMQI 統計資料資訊的收集。

第 124 頁的『控制佇列統計資料收集』

使用佇列屬性 STATQ 及佇列管理程式屬性 STATQ 來控制佇列統計資料資訊的收集

第 125 頁的『控制通道統計資料收集』

使用通道屬性 STATCHL 來控制通道統計資料資訊的收集。您也可以設定佇列管理程式屬性來控制資訊收集。

## 統計資料訊息格式

統計資料訊息包含一組由訊息描述子及訊息資料組成的 PCF 欄位。

## 訊息描述子

- 統計資料訊息 MQMD (訊息描述子)

## 帳戶訊息資料

- 統計資料訊息 MQCFH (PCF 標頭)
- 一律傳回的統計資料訊息資料
- 傳回的統計資料訊息資料 (如果有的話)

統計資料訊息 MQCFH (PCF 標頭) 包含記錄統計資料之間隔的相關資訊。

統計資料訊息資料包含儲存統計資料資訊的 PCF 參數。統計資料訊息的內容取決於訊息種類，如下所示：

### MQI 統計資料訊息

MQI 統計資料訊息資料由許多 PCF 參數組成，但沒有 PCF 群組。

### 佇列統計資料訊息

佇列統計資料訊息資料由許多 PCF 參數組成，且在 1 到 100 *QStatisticsData* PCF 群組的範圍內。

在間隔期間，每個作用中佇列都有一個 *QStatisticsData* PCF 群組。如果間隔中作用中的佇列超過 100 個，則會產生多個統計資料訊息。每一則訊息都會相應地更新 MQCFH (PCF 標頭) 中的 *SeqNumber*，而序列中的最後一則訊息會將 MQCFH 中的 *Control* 參數指定為 MQCFC\_LAST。

### 通道統計資料訊息

通道統計資料訊息資料包含許多 PCF 參數，且在範圍 1 到 100 *ChlStatistics* 資料 PCF 群組中。

在間隔期間，每個作用中通道都有一個 *ChlStatistics* 資料 PCF 群組。如果間隔中作用中的通道超過 100 個，則會產生多個統計資料訊息。每一則訊息都會相應地更新 MQCFH (PCF 標頭) 中的 *SeqNumber*，而序列中的最後一則訊息會將 MQCFH 中的 *Control* 參數指定為 MQCFC\_LAST。

## 統計資料資訊收集

使用佇列、佇列管理程式及通道屬性來控制統計資料資訊的收集

### 控制 MQI 統計資料收集

使用佇列管理程式屬性 STATMQI 統計資料資訊的收集。

若要變更此屬性的值，請使用 MQSC 指令 ALTER QMGR 並指定參數 **STATMQI**。只有在啟用統計資料收集之後開啟的佇列才會產生統計資料訊息。**STATMQI** 參數可以具有下列值：

#### 開啟

佇列管理程式的每個連線都會收集 MQI 統計資料資訊。

#### 關閉

未收集 MQI 統計資料資訊。這是預設值。

例如，若要啟用 MQI 統計資料資訊收集，請使用下列 MQSC 指令：

```
ALTER QMGR STATMQI(ON)
```

### 控制佇列統計資料收集

使用佇列屬性 STATQ 及佇列管理程式屬性 STATQ 來控制佇列統計資料資訊的收集

您可以啟用或停用個別佇列或多個佇列的佇列統計資料資訊收集。若要控制個別佇列，請設定佇列屬性 STATQ。您可以使用佇列管理程式屬性 STATQ，在佇列管理程式層次啟用或停用佇列統計資料資訊收集。對於佇列屬性 STATQ 已指定值 QMGR 的所有佇列，佇列統計資料資訊收集是在佇列管理程式層次進行控制。

只有使用在啟用統計資料收集之後開啟的 IBM MQ MQI 物件控點的作業，才會增加佇列統計資料。

只有在前一個時段已收集其統計資料資料的佇列，才會產生「佇列統計資料」訊息。

相同的佇列可以有數個放置作業，並透過數個「物件控點」來取得作業。部分「物件控點」可能在啟用統計資料收集之前已開啟，但其他則在之後開啟。因此，佇列統計資料可以記錄部分放置作業及取得作業的活動，而非全部。

若要確保「佇列統計資料」記錄所有應用程式的活動，您必須關閉並重新開啟所監視佇列上的新「物件控制點」。達到此目的的最佳方式是在啟用統計資料收集之後結束並重新啟動所有應用程式。

若要變更佇列屬性 STATQ 的值，請使用 MQSC 指令 ALTER QLOCAL 並指定參數 STATQ。佇列屬性 STATQ 可以具有下列值：

#### 開啟

會針對開啟佇列之佇列管理程式的每個連線收集佇列統計資料資訊。

#### OFF

不會收集此佇列的佇列統計資料資訊。

#### QMGR

此佇列的佇列統計資料資訊收集是根據佇列管理程式屬性 STATQ 的值來控制。這是預設值。

若要變更佇列管理程式屬性 STATQ 的值，請使用 MQSC 指令 ALTER QMGR 並指定參數 STATQ。佇列管理程式屬性 STATQ 可以具有下列值：

#### 開啟

針對佇列屬性 STATQ 設為 QMGR 的佇列收集佇列統計資料資訊

#### OFF

對於將佇列屬性 STATQ 設為 QMGR 的佇列，不會收集佇列統計資料資訊。這是預設值。

#### NONE

不論佇列屬性 STATQ 為何，都會停用所有佇列的佇列統計資料資訊收集。

如果佇列管理程式屬性 STATQ 設為 NONE，則不論佇列屬性 STATQ 為何，都會停用所有佇列的佇列統計資料資訊收集。

例如，若要啟用佇列 Q1 的統計資料資訊收集，請使用下列 MQSC 指令：

```
ALTER QLOCAL(Q1) STATQ(ON)
```

若要針對將佇列屬性 STATQ 指定為 QMGR 的所有佇列啟用統計資料資訊收集，請使用下列 MQSC 指令：

```
ALTER QMGR STATQ(ON)
```

#### 控制通道統計資料收集

使用通道屬性 STATCHL 來控制通道統計資料資訊的收集。您也可以設定佇列管理程式屬性來控制資訊收集。

您可以啟用或停用個別通道或多個通道的通道統計資料資訊收集。若要控制個別通道，您必須設定通道屬性 STATCHL，以啟用或停用通道統計資料資訊收集。若要一起控制許多通道，您可以使用佇列管理程式屬性 STATCHL 來啟用或停用佇列管理程式層次的通道統計資料資訊收集。對於通道屬性 STATCHL 已指定值 QMGR 的所有通道，通道統計資料資訊收集是在佇列管理程式層次進行控制。

自動定義的叢集傳送端通道不是 IBM MQ 物件，因此沒有與通道物件相同的屬性。若要控制自動定義的叢集傳送端通道，請使用佇列管理程式屬性 STATACLS。此屬性決定針對通道統計資料資訊收集，是啟用還是停用佇列管理程式內自動定義的叢集傳送端通道。

您可以將通道統計資料資訊收集設為下列三個監視層次之一：低、中或高。您可以在物件層次或佇列管理程式層次設定監視層次。選擇要使用的層次取決於您的系統。收集統計資料資訊資料可能需要一些相對昂貴的指令，因此為了減少通道統計資料資訊收集的影響，中、低監視選項會定期測量資料樣本，而不是一直收集資料。[第 125 頁的表 22 彙總通道統計資料資訊收集可用的層次：](#)

層次	說明	使用情形
低	定期測量資料的小型樣本。	適用於處理大量訊息的物件。
中	定期測量資料樣本。	適用於大部分物件。

表 22: 通道統計資料資訊收集的詳細層次 (繼續)		
層次	說明	使用情形
高	定期測量所有資料。	對於每秒只處理幾則訊息的物件，最新資訊很重要。

若要變更通道屬性 `STATCHL` 的值，請使用 `MQSC` 指令 `ALTER CHANNEL` 並指定參數 `STATCHL`。

若要變更佇列管理程式屬性 `STATCHL` 的值，請使用 `MQSC` 指令 `ALTER QMGR` 並指定參數 `STATCHL`。

若要變更佇列管理程式屬性 `STATACLS` 的值，請使用 `MQSC` 指令 `ALTER QMGR` 並指定參數 `STATACLS`。

通道屬性 `STATCHL` 可以具有下列值：

#### 低

通道統計資料資訊收集的詳細程度較低。

#### MEDIUM

通道統計資料資訊是以中等詳細程度來收集。

#### HIGH

通道統計資料資訊是以高階詳細資料來收集。

#### OFF

未收集此通道的通道統計資料資訊。

#### QMGR

通道屬性設為 `QMGR`。此通道的統計資料資訊收集是由佇列管理程式屬性 `STATCHL` 的值所控制。

這是預設值。

 在 `z/OS` 系統上，不論您選取的值為何，啟用此參數只會開啟統計資料收集。指定 `LOW`、`MEDIUM` 或 `HIGH` 對您的結果不會造成任何差別。必須啟用此參數才能收集通道統計記錄。

佇列管理程式屬性 `STATCHL` 可以具有下列值：

#### 低

針對通道屬性 `STATCHL` 設為 `QMGR` 的所有通道，收集通道統計資料資訊的詳細程度較低。

#### MEDIUM

針對通道屬性 `STATCHL` 設為 `QMGR` 的所有通道，會收集通道統計資料資訊，並提供中等詳細程度。

#### HIGH

針對通道屬性 `STATCHL` 設為 `QMGR` 的所有通道，會收集通道統計資料資訊並提供高階詳細資料。

#### OFF

未針對通道屬性 `STATCHL` 設為 `QMGR` 的所有通道收集通道統計資料資訊。

這是預設值。

#### NONE

不論通道屬性 `STATCHL` 為何，都會停用所有通道的通道統計資料資訊收集。

 在 `z/OS` 系統上，不論您選取的值為何，啟用此參數只會開啟統計資料收集。指定 `LOW`、`MEDIUM` 或 `HIGH` 對您的結果不會造成任何差別。必須啟用此參數才能收集通道統計記錄。

佇列管理程式屬性 `STATACLS` 可以具有下列值：

#### 低

對於自動定義的叢集傳送端通道，會以低詳細程度來收集統計資料資訊。

#### MEDIUM

對於自動定義的叢集傳送端通道，會以中等詳細程度來收集統計資料資訊。

#### HIGH

收集統計資料資訊時，會針對自動定義的叢集傳送端通道提供高階詳細資料。

#### OFF

不會收集自動定義叢集傳送端通道的統計資料資訊。

## QMGR

自動定義叢集傳送端通道的統計資料資訊收集由佇列管理程式屬性 `STATCHL` 的值控制。

這是預設值。

**z/OS** 在 z/OS 系統上，不論您選取的值為何，啟用此參數只會開啟統計資料收集。指定 `LOW`、`MEDIUM` 或 `HIGH` 對您的結果不會造成任何差別。必須啟用此參數才能收集通道統計記錄。

例如，若要啟用傳送端通道 `QM1.TO.QM2` 的統計資料資訊收集 (具有中等詳細程度)，請使用下列 `MQSC` 指令：

```
ALTER CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

若要針對將通道屬性 `STATCHL` 指定為 `QMGR` 的所有通道啟用中等詳細層次的統計資料資訊收集，請使用下列 `MQSC` 指令：

```
ALTER QMGR STATCHL(MEDIUM)
```

若要針對所有自動定義的叢集傳送端通道啟用中等詳細程度的統計資料資訊收集，請使用下列 `MQSC` 指令：

```
ALTER QMGR STATACLS(MEDIUM)
```

### 統計資料訊息產生

統計資料訊息會以已配置的時間隔產生，當佇列管理程式以受控制的方式關閉時也會產生。

配置的時間隔由 `STATINT` 佇列管理程式屬性控制，該屬性指定產生統計資料訊息之間的時間隔 (以秒為單位)。預設統計資料間隔為 1800 秒 (30 分鐘)。若要變更統計資料間隔，請使用 `MQSC` 指令 `ALTER QMGR` 並指定 `STATINT` 參數。例如，若要將統計資料間隔變更為 900 秒 (15 分鐘)，請使用下列 `MQSC` 指令：

```
ALTER QMGR STATINT(900)
```

若要在統計資料收集間隔到期之前，將目前收集的統計資料寫入統計資料佇列，請使用 `MQSC` 指令 `RESET QMGR TYPE(STATISTICS)`。發出此指令會將收集的統計資料寫入統計資料佇列，並開始新的統計資料資料收集間隔。

## 顯示帳戶及統計資料資訊

若要使用結算及統計資料訊息中記錄的資訊，請執行應用程式 (例如 `amqsmon` 範例程式)，以將記錄的資訊轉換為適當的格式

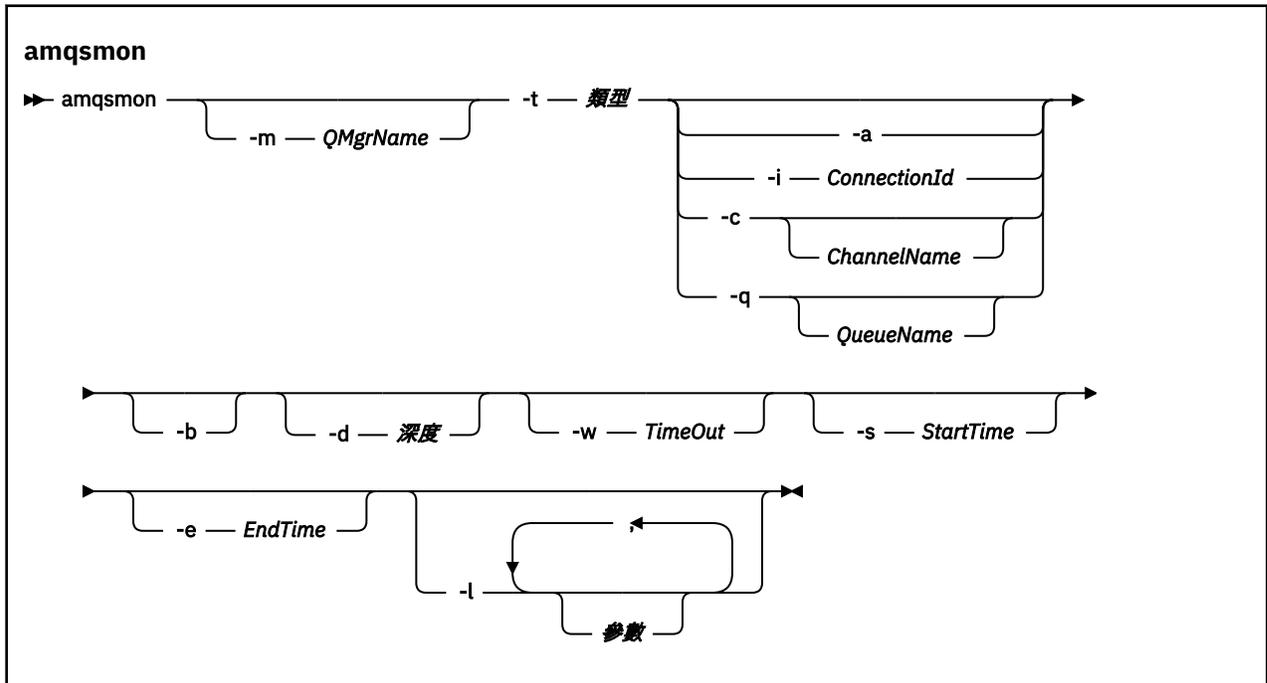
帳戶和統計資料訊息會寫入系統帳戶和統計資料佇列。`amqsmon` 是 IBM MQ 隨附的範例程式，可處理來自結算及統計資料佇列的訊息，並以可讀格式將資訊顯示至畫面。

因為 `amqsmon` 是範例程式，所以您可以使用所提供的原始碼作為範本，撰寫您自己的應用程式來處理帳戶或統計資料訊息，或修改 `amqsmon` 原始碼以符合您自己的特定需求。

### **amqsmon** (顯示格式化監視資訊)

使用 `amqsmon` 範例程式，以可讀取的格式顯示帳戶及統計資料訊息中包含的資訊。`amqsmon` 程式會從帳戶佇列 `SYSTEM.ADMIN.ACCOUNTING.QUEUE`。並從統計資料佇列 `SYSTEM.ADMIN.STATISTICS.QUEUE`。

## 語法



## 必要參數

### -t type

要處理的訊息類型。將 類型 指定為下列其中一項：

#### 統計作業

已處理帳戶記錄。從系統佇列 SYSTEM.ADMIN.ACCOUNTING.QUEUE。

#### statistics

已處理統計資料記錄。從系統佇列 SYSTEM.ADMIN.STATISTICS.QUEUE。

## 選用性參數

### -m QMgrName

要從中處理結算或統計資料訊息的佇列管理程式名稱。

如果您未指定此參數，則會使用預設佇列管理程式。

### -a

僅處理包含 MQI 記錄的訊息。

僅顯示 MQI 記錄。不包含 MQI 記錄的訊息一律會留在從中讀取它們的佇列中。

### -q QueueName

QueueName 是選用參數。

如果未提供 QueueName :

只顯示佇列結算及佇列統計資料記錄。

如果提供 QueueName :

只顯示 QueueName 所指定之佇列的佇列結算及佇列統計資料記錄。

如果未指定 `-b`，則會捨棄從中產生記錄的帳戶及統計資料訊息。由於結算及統計資料訊息也可以包含來自其他佇列的記錄，因此如果未指定 `-b`，則可以捨棄未看到的記錄。

### -c ChannelName

ChannelName 是選用參數。

如果未提供 *ChannelName* : 僅顯示通道統計資料記錄。  
如果提供 *ChannelName* : 只顯示 *ChannelName* 指定之通道的通道統計資料記錄。  
如果未指定 *-b* , 則會捨棄從中產生記錄的統計資料訊息。由於統計資料訊息也可以包含來自其他通道的記錄, 因此如果未指定 *-b* , 則可以捨棄未看到的記錄。

此參數僅在顯示統計資料訊息時可用 (*-t statistics*)。

#### **-i ConnectionId**

只顯示與 *ConnectionId* 指定的連線 ID 相關的記錄。

此參數僅在顯示帳戶訊息時可用 (*-t accounting*)。

如果未指定 *-b* , 則會捨棄從中產生記錄的統計資料訊息。由於統計資料訊息也可以包含來自其他通道的記錄, 因此如果未指定 *-b* , 則可以捨棄未看到的記錄。

#### **-b**

瀏覽訊息。

以非破壞性方式擷取訊息。

#### **-d 深度**

可處理的訊息數目上限。

如果您未指定此參數, 則可以處理不限數目的訊息。

#### **-w timeout**

等待訊息變成可用的時間秒數上限。

如果未指定此參數, 則一旦沒有其他要處理的訊息, *amqsmon* 即會結束。

#### **-s StartTime**

僅處理在指定的 *StartTime* 之後放置的訊息。

*StartTime* 是以 *yyyy-mm-dd hh.mm.ss* 格式指定。如果指定的日期沒有時間, 則時間會預設為指定日期的 00.00.00。時間以 GMT 為單位。

如需不指定此參數的效果, 請參閱 [附註 1](#)。

#### **-e EndTime**

僅處理放在指定 *EndTime* 之前的訊息。

*EndTime* 以 *yyyy-mm-dd hh.mm.ss* 格式指定。如果指定的日期沒有時間, 則時間會預設為指定日期的 23.59.59。時間以 GMT 為單位。

如需不指定此參數的效果, 請參閱 [附註 1](#)。

#### **-l 參數**

僅顯示從處理的記錄中選取的欄位。參數 是以逗點區隔的整數值清單, 每一個整數值對映至欄位的數值常數, 請參閱 [amqsmon 範例 5](#)。

如果您未指定此參數, 則會顯示所有可用的欄位。

#### **註:**

1. 如果您未指定 *-s StartTime* 或 *-e EndTime*, 則可處理的訊息不受放置時間限制。

### **amqsmon (顯示格式化監視資訊) 範例**

請利用這個頁面來檢視執行 *amqsmon* (顯示格式化監視資訊) 範例程式的範例

1. 如需屬性的說明, 請參閱 [第 168 頁的『佇列統計資料訊息資料』](#)。

下列指令會顯示來自佇列管理程式 *saturn.queue.manager* 的所有 MQI 統計資料訊息:

```
amqsmon -m saturn.queue.manager -t statistics -a
```

此指令的輸出如下:

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0
```

2. 下列指令會顯示佇列管理程式 LOCALQ 上 saturn.queue.manager 佇列的所有佇列統計資料訊息:

```
amqsmon -m saturn.queue.manager -t statistics -q LOCALQ
```

此指令的輸出如下:

```
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics:
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  QueueDefinitionType: Local
  QMinDepth: 0
  QMaxDepth: 18
  AverageQueueTime: [29827281, 0]
  PutCount: [26, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [88, 0]
  GetCount: [18, 0]
  GetBytes: [52, 0]
  GetFailCount: 0
  BrowseCount: [0, 0]
  BrowseBytes: [0, 0]
  BrowseFailCount: 1
  NonQueuedMsgCount: 0
  ExpiredMsgCount: 0
  PurgedMsgCount: 0
```

3. 下列指令會顯示自 2005 年 4 月 30 日 15:30 以來從佇列管理程式 saturn.queue.manager 記錄的所有統計資料訊息。

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

此指令的輸出如下:

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0]
...
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics: 0
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...
QueueStatistics: 1
  QueueName: 'SAMPLEQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...
```

4. 如需屬性的說明, 請參閱 第 147 頁的『佇列帳戶訊息資料』。

下列指令會顯示 2005 年 4 月 30 日從佇列管理程式 saturn.queue.manager 記錄的所有結算訊息:

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

此指令的輸出如下:

```
RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
```

```

GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
SeqNumber: 0
ApplicationName: 'runmqsc'
ApplicationPid: 8615
ApplicationTid: 1
...

```

5. 下列指令會瀏覽結算佇列，並顯示有 MQI 結算資訊可用的每個應用程式的應用程式名稱及連線 ID:

```
amqsmon -m saturn.queue.manager -t accounting -b -l 7006,3024
```

此指令的輸出如下:

```

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01e80322'
ApplicationName: 'WebSphere MQ\bin\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01ea0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01e90322'
ApplicationName: 's\IBM\MQ_4\bin64\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01ef0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01e60322'
ApplicationName: 's\IBM\MQ_4\bin64\runmqsc.exe'
QueueAccounting: 0

```

如需如何尋找本主題中使用的變數的詳細資料，請參閱 [第 133 頁的『尋找 ApplicationName \(3024\) 和 ConnectionId \(7006\) 的對映』](#)。

如需在 **amqsmon** 指令中為 **Type** 參數選取 統計量 時如何尋找變數的詳細資料，請參閱 [第 133 頁的『尋找 AvgTimeOnQ \(703\) 和 QmaxDepth \(739\) 的對映』](#)。

尋找 **ApplicationName (3024)** 和 **ConnectionId (7006)** 的對映  
請利用這個主題來說明如何找到範例中用來監視統計資料的變數。

摘要：

- 3024 表示 MQCACF\_APPL\_NAME 是 **ApplicationName**
- 7006 表示 MQBACF\_CONNECTION\_ID，其為 **ConnectionId**

若要找出對映，您需要執行兩個程序：

1. 請造訪 [第 147 頁的『佇列帳戶訊息資料』](#)，以取得您在 **amqsmon** 指令中針對 **Type** 參數選取 *accounting* 時，屬性 [第 149 頁的『ApplicationName』](#) 及 [第 149 頁的『ConnectionId』](#) 的說明。

在每一種情況下，請尋找屬性 **Identifier**。

**ApplicationName** 的 **Identifier** 是 MQCACF\_APPL\_NAME，而 **ConnectionId** 的是 MQBACF\_CONNECTION\_ID

2. 搜尋您在步驟 [第 133 頁的『1』](#) 中找到的 ID。

移至 **常數** 區段並向下捲動，直到您找到 MQCACF\_\* (指令格式「字元參數類型」) 清單為止。找到 MQCACF\_APPL\_NAME，您會看到值 3024。

同樣地，尋找 MQBACF\_\* (指令格式位元組參數類型) 清單。找到 MQBACF\_CONNECTION\_ID，您會看到值 7006。

尋找 **AvgTimeOnQ (703)** 和 **QmaxDepth (739)** 的對映  
請利用這個主題來說明如何找到範例中用來監視統計資料的變數。

摘要：

- 703 表示 MQIAMO64\_AVG\_Q\_TIME 是 **AvgTimeOnQ**
- 739 表示 MQIAMO\_Q\_MAX\_DEPTH，其為 **QMaxDepth**

若要找出對映，您需要執行兩個程序：

1. 請造訪 [第 168 頁的『佇列統計資料訊息資料』](#)，以取得您在 **amqsmon** 指令中選取 統計 作為 **Type** 參數的屬性 [第 171 頁的『AvgTimeOnQ』](#) 及 [第 171 頁的『QMaxDepth』](#) 的說明。

在每一種情況下，請尋找屬性 **Identifier**。

**AvgTimeOnQ** 的 **Identifier** 為 MQIAMO64\_AVG\_Q\_TIME 且適用於。 **QMaxDepth** 是 MQIAMO\_Q\_MAX\_DEPTH。

2. 搜尋您在步驟 [第 133 頁的『1』](#) 中找到的 ID。

移至 **常數** 區段並向下捲動，直到您找到 MQIAMO\_\* (指令格式「整數監視參數類型」) 清單為止。找出 MQIAMO\_Q\_MAX\_DEPTH，您會看到值 739。

同樣地，尋找 MQIAMO64\_\* (指令格式 64 位元整數監視參數類型) 清單。找到 MQIAMO64\_AVG\_Q\_TIME，您會看到值 703。

## 帳戶及統計資料訊息參照

請利用這個頁面來取得帳戶和統計資料訊息的格式概觀，以及這些訊息中所傳回的資訊

結算及統計資料訊息是包含訊息描述子及訊息資料的標準 IBM MQ 訊息。 訊息資料包含 IBM MQ 應用程式所執行 MQI 作業的相關資訊，或 IBM MQ 系統中所發生活動的相關資訊。

## 訊息描述子

- MQMD 結構

## 訊息資料

- PCF 標頭 (MQCFH)
- 一律傳回的帳戶或統計資料訊息資料
- 傳回的帳戶或統計資料訊息資料 (如果有的話)

## 帳戶和統計資料訊息格式

請利用這個頁面作為 MQI 結算訊息的結構範例

表 23: MQI 結算訊息結構

MQMD 結構	結算訊息標頭 MQCFH 結構	MQI 結算訊息資料 <sup>1</sup>
結構 ID 結構版本 報告選項 訊息類型 有效期限 回饋碼 編碼 編碼字集 ID 訊息格式 訊息優先順序 持續性 訊息 ID 相關性 ID 取消次數 回覆目的地佇列 回覆目的地佇列管理程式 使用者 ID 帳戶記號 應用程式身分資料 應用程式類型 應用程式名稱 放置日期 放置時間 應用程式原始資料 群組 ID 訊息序號 偏移 訊息旗標 原始長度	結構類型 結構長度 結構版本 指令 ID 訊息序號 控制選項 完成碼 原因碼 參數計數	佇列管理程式 間隔開始日期 間隔開始時間 間隔結束日期 間隔結束時間 指令層次 連線 ID 序號 應用程式名稱 應用程式程序 ID 應用程式執行緒 ID 使用者 ID 連線日期 連線時間 連線名稱 通道名稱 中斷連線日期 斷線時間 斷線類型 開啟計數 開啟失敗計數 關閉計數 關閉失敗計數 放置計數 放置失敗計數 Put1 計數 PUT1 失敗計數 放置位元組 取得計數 取得失敗計數 取得位元組 瀏覽計數 瀏覽失敗計數 瀏覽位元組 確定計數 確定失敗計數 取消次數 查詢計數 查詢失敗計數 設定計數 設定失敗計數
<p><b>註:</b></p> <p>1. 顯示的參數是針對 MQI 結算訊息所傳回的參數。實際結算或統計資料訊息資料取決於訊息種類。</p>		

### 結算及統計資料訊息 MQMD (訊息描述子)

請利用這個頁面來瞭解結算和統計資料訊息的訊息描述子與事件訊息的訊息描述子之間的差異

結算及統計資料訊息的訊息描述子中的參數及值與事件訊息的訊息描述子中的參數及值相同，但有下列異常狀況:

## Format

說明:	訊息資料的格式名稱。
資料類型:	MQCHAR8.
值:	<b>MQFMT_ADMIN</b> 管理訊息。

結算及統計資料訊息的訊息描述子中包含的部分參數包含產生訊息的佇列管理程式所提供的固定資料。

MQMD 也會指定放置訊息的佇列管理程式名稱 (截斷為 28 個字元)，以及將訊息放置在計數或統計資料佇列上的日期和時間。

## 結算及統計資料訊息中的訊息資料

統計及統計值訊息中的訊息資料基於可程式化指令格式 (PCF)，在 PCF 指令查詢及回應中使用。結算及統計資料訊息中的訊息資料由 PCF 標頭 (MQCFH) 及結算或統計資料報告組成。

## 結算及統計資料訊息 MQCFH (PCF 標頭)

結算及統計資料訊息的訊息標頭是 MQCFH 結構。結算及統計資料訊息的訊息標頭中的參數及值與事件訊息的訊息標頭中的參數及值相同，但有下列例外：

### Command

說明:	指令 ID。這會識別帳戶或統計資料訊息種類。
資料類型:	MQLONG。
值:	<b>MQCMD_ACCOUNTING_MQI</b> MQI 結算訊息。 <b>MQCMD_ACCOUNTING_Q</b> 佇列帳戶訊息。 <b>MQCMD_STATISTICS_MQI</b> MQI 統計資料訊息。 <b>MQCMD_STATISTICS_Q</b> 佇列統計資料訊息。 <b>MQCMD_STATISTICS_CHANNEL</b> 通道統計資料訊息。

### Version

說明:	結構版本號碼。
資料類型:	MQLONG。
值:	<b>MQCFH_VERSION_3</b> Version-3，代表帳戶和統計資料訊息。

## 帳戶和統計資料訊息資料

結算及統計資料訊息資料的內容取決於結算或統計資料訊息的種類，如下所示：

### MQI 結算訊息

MQI 結算訊息資料由許多 PCF 參數組成，但沒有 PCF 群組。

### 佇列帳戶訊息

佇列結算訊息資料由許多 PCF 參數組成，範圍為 1 到 100 *QAccountingData* PCF 群組。

### MQI 統計資料訊息

MQI 統計資料訊息資料由許多 PCF 參數組成，但沒有 PCF 群組。

### 佇列統計資料訊息

佇列統計資料訊息資料由許多 PCF 參數組成，且在 1 到 100 *QStatisticsData* PCF 群組的範圍內。

### 通道統計資料訊息

通道統計資料訊息資料包含許多 PCF 參數，且在範圍 1 到 100 *ChlStatistics* 資料 PCF 群組中。

### MQI 結算訊息資料

請利用這個頁面來檢視 MQI 結算訊息的結構

訊息名稱:	MQI 結算訊息。
平台:	全部，但 IBM MQ for z/OS 除外。
系統佇列:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

#### **QueueManager**

說明:	佇列管理程式的名稱
ID:	MQCA_Q_MGR_NAME
資料類型:	MQCFST
長度上限:	MQ_Q_MGR_NAME_LENGTH
已傳回:	一律

#### **IntervalStartDate**

說明:	監視期間開始的日期
ID:	MQCAMO_START_DATE
資料類型:	MQCFST
長度上限:	MQ 資料長度
已傳回:	一律

#### **IntervalStartTime**

說明:	監視期間開始的時間
ID:	MQCAMO_START_TIME
資料類型:	MQCFST
長度上限:	MQ 時間長度
已傳回:	一律

#### **IntervalEndDate**

說明:	監視期間結束的日期
ID:	MQCAMO_END_DATE
資料類型:	MQCFST
長度上限:	MQ 資料長度
已傳回:	一律

#### **IntervalEndTime**

說明:	監視期間結束的時間
ID:	MQCAOM_END_TIME

資料類型: MQCFST  
長度上限: MQ 時間長度  
已傳回: 一律

#### **CommandLevel**

說明: 佇列管理程式指令層次  
ID: MQIA\_COMMAND\_LEVEL  
資料類型: MQCFIN  
已傳回: 一律

#### **ConnectionId**

說明: IBM MQ 連線的連線 ID  
ID: MQBACF\_CONNECTION\_ID  
資料類型: MQCFBS  
長度上限: MQ\_CONNECTION\_ID\_LENGTH  
已傳回: 一律

#### **SeqNumber**

說明: 序號。對於長時間執行連線的每一筆後續記錄，此值會增加。  
ID: MQIACF\_SEQUENCE\_NUMBER  
資料類型: MQCFIN  
已傳回: 一律

#### **ApplicationName**

說明: 應用程式的名稱。此欄位的內容相當於訊息描述子中 *PutApplName* 欄位的內容。  
ID: MQCACF\_APPL\_NAME  
資料類型: MQCFST  
長度上限: MQ\_APPL\_NAME\_LENGTH  
已傳回: 一律

#### **ApplicationPid**

說明: 應用程式的作業系統處理程序 ID  
ID: MQIACF\_PROCESS\_ID  
資料類型: MQCFIN  
已傳回: 一律

#### **ApplicationTid**

說明: 應用程式中連線的 IBM MQ 執行緒 ID  
ID: MQIACF\_THREAD\_ID  
資料類型: MQCFIN  
已傳回: 一律

### **UserId**

說明:	應用程式的使用者 ID 環境定義
ID:	MQCACF_USER_IDENTIFIER
資料類型:	MQCFST
長度上限:	MQ_USER_ID_LENGTH
已傳回:	一律

### **ConnDate**

說明:	MQCONN 作業的日期
ID:	MQCAMO_CONN_DATE
資料類型:	MQCFST
長度上限:	MQ 時間長度
已傳回:	可用時

### **ConnTime**

說明:	MQCONN 作業的時間
ID:	MQCAMO_CONN_TIME
資料類型:	MQCFST
長度上限:	MQ 時間長度
已傳回:	可用時

### **ConnName**

說明:	用戶端連線的連線名稱
ID:	MQ 快取 _ 連線名稱
資料類型:	MQCFST
長度上限:	MQ_CONN_NAME_LENGTH
已傳回:	可用時

### **ChannelName**

說明:	用戶端連線的通道名稱
ID:	MQ 快取 _ 通道名稱
資料類型:	MQCFST
長度上限:	MQ_CHANNEL_NAME_LENGTH
已傳回:	可用時

### **V9.1.0 RemoteProduct**

說明:	用戶端連線的遠端產品 ID，如 <a href="#">DISPLAY CHSATUS</a> 的 <b>RPRODUCT</b> 欄位中所示
ID:	MQ 快取 _ 遠端產品
資料類型:	MQCFST
長度上限:	MQ 遠端產品長度

已傳回: 可用時

### **V 9.1.0 RemoteVersion**

說明: 用戶端連線的遠端產品版本, 如 DISPLAY CHSTATUS 的 **RVERSION** 欄位中所顯示  
ID: MQ 快取\_遠端版本  
資料類型: MQCFST  
長度上限: MQ 遠端版本長度  
已傳回: 可用時

### **DiscDate**

說明: MQDISC 作業的日期  
ID: MQCAMO\_DISC\_DATE  
資料類型: MQCFST  
長度上限: MQ 資料長度  
已傳回: 可用時

### **DiscTime**

說明: MQDISC 作業的時間  
ID: MQCAMO\_DISC\_TIME  
資料類型: MQCFST  
長度上限: MQ 時間長度  
已傳回: 可用時

### **DiscType**

說明: 斷線的類型  
ID: MQIAMO\_DISC\_TYPE  
資料類型: MQCFIN  
值: 可能值包括:  
**MQDISCONNECT\_NORMAL**  
由應用程式要求  
**MQDISCONNECT\_IMPLICIT**  
應用程式異常終止  
**MQDISCONNECT\_Q\_MGR**  
佇列管理程式中斷連線  
已傳回: 可用時

### **OpenCount**

說明: 透過直接對 MQOPEN 發出呼叫或使用 MQPUT1 動詞來順利開啟的物件數。此參數是依物件類型編製索引的整數清單, 請參閱 [參照附註 1](#)。  
ID: MQIamo\_opens  
資料類型: MQCFIL  
已傳回: 可用時

### **OpenFailCount**

說明:	嘗試開啟物件但失敗的次數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_OPENS_FAILED
資料類型:	MQCFIL
已傳回:	可用時

### **CloseCount**

說明:	已關閉的物件數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_CLOSES
資料類型:	MQCFIL
已傳回:	可用時

### **CloseFailCount**

說明:	嘗試關閉物件但失敗的次數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_CLOSES_FAILED
資料類型:	MQCFIL
已傳回:	可用時

### **PutCount**

說明:	順利放入佇列的持續及非持續訊息數，但使用 MQPUT1 呼叫放置的訊息除外。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT
資料類型:	MQCFIL
已傳回:	可用時

### **PutFailCount**

說明:	不成功的放置訊息嘗試次數
ID:	MQIAMO_PUTS_FAILED
資料類型:	MQCFIN
已傳回:	可用時

### **Put1Count**

說明:	使用 MQPUT1 呼叫順利置入佇列的持續及非持續訊息數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT1S
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **Put1FailCount**

說明:	使用 MQPUT1 呼叫放置訊息但失敗的嘗試次數
ID:	MQIAMO_PUT1S_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	QAccountingData
已傳回:	可用時

### **PutBytes**

說明:	使用持續及非持續訊息的放置呼叫所寫入的位元組數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO64_PUT_BYTES
資料類型:	MQCFIL64
已傳回:	可用時

### **GetCount**

說明:	持續及非持續訊息的成功破壞性 MQGET 呼叫數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_GETS
資料類型:	MQCFIL
已傳回:	可用時

### **GetFailCount**

說明:	失敗的破壞性 MQGET 呼叫數
ID:	MQIAMO_GETS_FAILED
資料類型:	MQCFIN
已傳回:	可用時

### **GetBytes**

說明:	針對持續及非持續訊息所擷取的位元組總數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO64_GET_BYTES
資料類型:	MQCFIL64
已傳回:	可用時

### **BrowseCount**

說明:	持續及非持續訊息的成功非破壞性 MQGET 呼叫數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_BROWSES
資料類型:	MQCFIL
已傳回:	可用時

### **BrowseFailCount**

說明： 不成功的非破壞性 MQGET 呼叫數  
ID： MQIAMO\_BROWSES\_FAILED  
資料類型： MQCFIN  
已傳回： 可用時

### **BrowseBytes**

說明： 持續及非持續訊息的已瀏覽位元組總數。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。  
ID： MQIAMO64\_BROWSE\_BYTES  
資料類型： MQCFIL64  
已傳回： 可用時

### **CommitCount**

說明： 成功的交易數。此數目包括已連接應用程式隱含地確定的那些交易。此計數中包括沒有未完成工作的確定要求。  
ID： MQIAMO\_COMMITS  
資料類型： MQCFIN  
已傳回： 可用時

### **CommitFailCount**

說明： 嘗試完成交易但失敗的次數  
ID： MQIAMO\_COMMITS\_FAILED  
資料類型： MQCFIN  
已傳回： 可用時

### **BackCount**

說明： 已處理的取消數，包括因異常斷線而產生的隱含取消  
ID： MQIAMO\_BACKOUTS  
資料類型： MQCFIN  
已傳回： 可用時

### **InqCount**

說明： 所查詢的成功物件數。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。  
ID： MQIAMO\_INQS  
資料類型： MQCFIL  
已傳回： 可用時

### **InqFailCount**

說明： 不成功的物件查詢嘗試次數。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。  
ID： MQIAMO\_INQS\_FAILED

資料類型: MQCFIL  
已傳回: 可用時

### **SetCount**

說明: 成功的 MQSET 呼叫數目。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。

ID: MQIamo\_SETS  
資料類型: MQCFIL  
已傳回: 可用時

### **SetFailCount**

說明: 失敗的 MQSET 呼叫數目。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。

ID: MQIAMO\_SETS\_FAILED  
資料類型: MQCFIL  
已傳回: 可用時

### **SubCountDur**

說明: 已建立、變更或回復可延續訂閱的成功訂閱要求數。這是依作業類型編製索引的值陣列

0 = 已建立的訂閱數  
1 = 已變更的訂閱數  
2 = 已回復的訂閱數

ID: MQIamo\_SUBS\_DUR  
資料類型: MQCFIL  
已傳回: 可用時。

### **SubCountNDur**

說明: 已建立、變更或回復不可延續訂閱的成功訂閱要求數。這是依作業類型編製索引的值陣列

0 = 已建立的訂閱數  
1 = 已變更的訂閱數  
2 = 已回復的訂閱數

ID: MQIAMO\_SUBS\_NDUR  
資料類型: MQCFIL  
已傳回: 可用時。

### **SubFailCount**

說明: 失敗的訂閱要求數目。

ID: MQIAMO\_SUBS\_FAILED  
資料類型: MQCFIN  
已傳回: 可用時。

### ***UnsubCountDur***

說明：可延續訂閱的成功取消訂閱要求數。這是依作業類型編製索引的值陣列  
0-訂閱已關閉但未移除  
1-已關閉並移除訂閱

ID：MQIAMO\_UNSUBS\_DUR

資料類型：MQCFIL

已傳回：可用時。

### ***UnsubCountNDur***

說明：可延續訂閱的成功取消訂閱要求數。這是依作業類型編製索引的值陣列  
0-訂閱已關閉但未移除  
1-已關閉並移除訂閱

ID：MQIAMO\_UNSUBS\_NDUR

資料類型：MQCFIL

已傳回：可用時。

### ***UnsubFailCount***

說明：不成功的取消訂閱要求數。

ID：MQIAMO\_UNSUBS\_FAILED

資料類型：MQCFIN

已傳回：可用時。

### ***SubRqCount***

說明：成功的 MQSUBRQ 要求數目。

ID：MQIAMO\_SUBRQS

資料類型：MQCFIN

已傳回：可用時。

### ***SubRqFailCount***

說明：失敗的 MQSUB 要求數。

ID：MQIAMO\_SUBRQS\_FAILED

資料類型：MQCFIN

已傳回：可用時。

### ***CBCount***

說明：成功 MQCB 要求數目。這是依作業類型編製索引的值陣列  
0-已建立或變更回呼  
1-已移除回呼  
2-已回復回呼  
3-已暫停回呼

ID: MQIAMO\_CBS  
資料類型: MQCFIN  
已傳回: 可用時。

#### ***CBFailCount***

說明: 失敗的 MQCB 要求數目。  
ID: MQIAMO\_CBS\_FAILED  
資料類型: MQCFIN  
已傳回: 可用時。

#### ***CtlCount***

說明: 順利完成的 MQCTL 要求數目。這是依作業類型編製索引的值陣列  
0-已啟動連線  
1-連線已停止  
2-已回復連線  
3-已暫停連線

ID: MQIAMO\_CTL5  
資料類型: MQCFIL  
已傳回: 可用時。

#### ***CtlFailCount***

說明: 失敗的 MQCTL 要求數目。  
ID: MQIAMO\_CTL5\_FAILED  
資料類型: MQCFIN  
已傳回: 可用時。

#### ***StatCount***

說明: 成功的 MQSTAT 要求數目。  
ID: MQIAMO\_STATS。  
資料類型: MQCFIN  
已傳回: 可用時。

#### ***StatFailCount***

說明: 失敗的 MQSTAT 要求數目。  
ID: MQIAMO\_STATS\_FAILED  
資料類型: MQCFIN  
已傳回: 可用時。

### **PutTopicCount**

說明：順利放置至主題的持續及非持續訊息數，但使用 MQPUT1 呼叫放置的訊息除外。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

附註：此值包括使用佇列別名 (解析為主題) 所放置的訊息。

ID: MQIAMO\_TOPIC\_PUT

資料類型: MQCFIL

已傳回: 可用時。

### **PutTopicFailCount**

說明：將訊息置入主題的失敗嘗試次數。

ID: MQIAMO\_TOPIC\_PUTS\_FAILED

資料類型: MQCFIN

已傳回: 可用時。

### **Put1TopicCount**

說明：使用 MQPUT1 呼叫順利放置到主題的持續及非持續訊息數。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

附註：此值包括使用佇列別名 (解析為主題) 所放置的訊息。

ID: MQIAMO\_TOPIC\_PUT1S

資料類型: MQCFIL

已傳回: 可用時。

### **Put1TopicFailCount**

說明：使用 MQPUT1 呼叫將訊息放置到主題的失敗嘗試次數。

ID: MQIAMO\_TOPIC\_PUT1S\_FAILED

資料類型: MQCFIN

已傳回: 可用時。

### **PutTopicBytes**

說明：針對解析為發佈作業的持續及非持續訊息使用放置呼叫所寫入的位元組數。這是應用程式所放置的位元組數，而不是遞送給訂閱者的結果位元組數。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_TOPIC\_PUT\_BYTES

資料類型: MQCFIL64

已傳回: 可用時。

### **佇列帳戶訊息資料**

請利用這個頁面來檢視佇列帳戶訊息的結構

訊息名稱:	佇列帳戶訊息。
平台:	全部，但 IBM MQ for z/OS 除外。
系統佇列:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

### **QueueManager**

說明:	佇列管理程式的名稱
ID:	MQCA_Q_MGR_NAME
資料類型:	MQCFST
長度上限:	MQ_Q_MGR_NAME_LENGTH
已傳回:	一律

### **IntervalStartDate**

說明:	監視期間開始的日期
ID:	MQCAMO_START_DATE
資料類型:	MQCFST
長度上限:	MQ 資料長度
已傳回:	一律

### **IntervalStartTime**

說明:	監視期間開始的時間
ID:	MQCAMO_START_TIME
資料類型:	MQCFST
長度上限:	MQ 時間長度
已傳回:	一律

### **IntervalEndDate**

說明:	監視期間結束的日期
ID:	MQCAMO_END_DATE
資料類型:	MQCFST
長度上限:	MQ 資料長度
已傳回:	一律

### **IntervalEndTime**

說明:	監視期間結束的時間
ID:	MQCAOM_END_TIME
資料類型:	MQCFST
長度上限:	MQ 時間長度
已傳回:	一律

### **CommandLevel**

說明:	佇列管理程式指令層次
ID:	MQIA_COMMAND_LEVEL
資料類型:	MQCFIN
已傳回:	一律

**ConnectionId**

說明:	IBM MQ 連線的連線 ID
ID:	MQBACF_CONNECTION_ID
資料類型:	MQCFBS
長度上限:	MQ_CONNECTION_ID_LENGTH
已傳回:	一律

**SeqNumber**

說明:	序號。對於長時間執行連線的每一筆後續記錄，此值會增加。
ID:	MQIACF_SEQUENCE_NUMBER
資料類型:	MQCFIN
已傳回:	一律

**ApplicationName**

說明:	應用程式的名稱。此欄位的內容相當於訊息描述子中 PutAppl 名稱欄位的內容。
ID:	MQCACF_APPL_NAME
資料類型:	MQCFST
長度上限:	MQ_APPL_NAME_LENGTH
已傳回:	一律

**ApplicationPid**

說明:	應用程式的作業系統處理程序 ID
ID:	MQIACF_PROCESS_ID
資料類型:	MQCFIN
已傳回:	一律

**ApplicationTid**

說明:	應用程式中連線的 IBM MQ 執行緒 ID
ID:	MQIACF_THREAD_ID
資料類型:	MQCFIN
已傳回:	一律

**UserId**

說明:	應用程式的使用者 ID 環境定義
ID:	MQCACF_USER_IDENTIFIER
資料類型:	MQCFST
長度上限:	MQ_USER_ID_LENGTH
已傳回:	一律

**ObjectCount**

說明:	在已記錄結算資料的間隔內存取的佇列數。此值設為訊息中包含的 <i>QAccountingData</i> PCF 群組數。
-----	---

ID: MQIAMO\_OBJECT\_COUNT  
資料類型: MQCFIN  
已傳回: 一律

### **QAccountingData**

說明: 指定佇列的帳戶詳細資料的分組參數

ID: MQGACF\_Q\_ACCOUNTING\_DATA

資料類型: MQCFGR

群組中的參數:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- OpenCount*
- OpenDate*
- OpenTime*
- CloseDate*
- CloseTime*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- PutMinBytes*
- PutMaxBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- GetMinBytes*
- GetMaxBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- BrowseMinBytes*
- BrowseMaxBytes*
- TimeOnQMin*
- TimeOnQAvg*
- TimeOnQMax*

已傳回: 一律

### **QName**

說明: 佇列的名稱

ID: MQCA\_Q\_NAME

資料類型: MQCFST

包含在 PCF 群組中: *QAccountingData*

長度上限: MQ\_Q\_NAME\_LENGTH  
已傳回: 可用時

### **CreateDate**

說明: 建立佇列的日期  
ID: MQCA\_CREATION\_DATE  
資料類型: MQCFST  
包含在 PCF 群組中: *QAccountingData*  
長度上限: MQ 資料長度  
已傳回: 可用時

### **CreateTime**

說明: 建立佇列的時間  
ID: MQCA\_CREATION\_TIME  
資料類型: MQCFST  
包含在 PCF 群組中: *QAccountingData*  
長度上限: MQ 時間長度  
已傳回: 可用時

### **QType**

說明: 佇列的類型  
ID: MQIA\_Q\_TYPE  
資料類型: MQCFIN  
包含在 PCF 群組中: *QAccountingData*  
值: 本端 MQQT\_LOCAL  
已傳回: 可用時

### **QDefinitionType**

說明: 佇列定義類型  
ID: MQIA\_DEFINITION\_TYPE  
資料類型: MQCFIN  
包含在 PCF 群組中: *QAccountingData*  
值: 可能的值為:  
**MQQDT\_PREDEFINED**  
**MQQDT\_PERMANENT\_DYNAMIC**  
**MQQDT\_TEMPORARY\_DYNAMIC**  
已傳回: 可用時

### **OpenCount**

說明:	應用程式在此間隔中透過直接對 MQOPEN 發出呼叫或使用 MQPUT1 動詞來開啟此佇列的次數。
ID:	MQIamo_opens
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **OpenDate**

說明:	在此記錄間隔中第一次開啟佇列的日期。如果在此間隔開始時已開啟佇列，則此值會反映最初開啟佇列的日期。
ID:	MQCAMO_OPEN_DATE
資料類型:	MQCFST
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **OpenTime**

說明:	在此記錄間隔中第一次開啟佇列的時間。如果在此間隔開始時已開啟佇列，則此值會反映最初開啟佇列的時間。
ID:	MQCAMO_OPEN_TIME
資料類型:	MQCFST
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **CloseDate**

說明:	在此記錄間隔中，佇列最終關閉的日期。如果佇列仍開啟，則不會傳回值。
ID:	MQCAMO_CLOSE_DATE
資料類型:	MQCFST
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **CloseTime**

說明:	在此記錄間隔中，佇列最終關閉的時間。如果佇列仍開啟，則不會傳回值。
ID:	MQCAMO_CLOSE_TIME
資料類型:	MQCFST
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **PutCount**

說明:	已順利放入佇列中的持續及非持續訊息數，但 MQPUT1 呼叫除外。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **PutFailCount**

說明:	不成功的放置訊息嘗試次數，但 MQPUT1 呼叫除外
ID:	MQIAMO_PUTS_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **Put1Count**

說明:	使用 MQPUT1 呼叫順利置入佇列的持續及非持續訊息數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT1S
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **Put1FailCount**

說明:	使用 MQPUT1 呼叫放置訊息但失敗的嘗試次數
ID:	MQIAMO_PUT1S_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **PutBytes**

說明:	持續及非持續訊息的放置位元組總數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO64_PUT_BYTES
資料類型:	MQCFIL64
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **PutMinBytes**

說明:	放置在佇列上的最小持續及非持續訊息大小。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT_MIN_BYTES
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **PutMaxBytes**

說明:	放置在佇列上的最大持續及非持續訊息大小。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT_MAX_BYTES
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **GeneratedMsgCount**

說明:	已產生的訊息數。產生的訊息為 <ul style="list-style-type: none"><li>• 佇列深度高事件</li><li>• 佇列深度低事件</li></ul>
ID:	MQIAMO_GENERATED_MSGS
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **GetCount**

說明:	持續及非持續訊息的成功破壞性 MQGET 呼叫數。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_GETS
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QAccountingData</i>
已傳回:	可用時

### **GetFailCount**

說明:	失敗的破壞性 MQGET 呼叫數
ID:	MQIAMO_GETS_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QAccountingData</i>

已傳回: 可用時

### **GetBytes**

說明: 持續及非持續訊息之破壞性 MQGET 呼叫中所讀取的位元組數。此參數是依持續性值編製索引的整數清單, 請參閱 [參照附註 2](#)。

ID: MQIAMO64\_GET\_BYTES

資料類型: MQCFIL64

包含在 PCF 群組中: QAccountingData

已傳回: 可用時

### **GetMinBytes**

說明: 從佇列擷取的最小持續及非持續訊息的大小。此參數是依持續性值編製索引的整數清單, 請參閱 [參照附註 2](#)。

ID: MQIAMO\_GET\_MIN\_BYTES

資料類型: MQCFIL

包含在 PCF 群組中: QAccountingData

已傳回: 可用時

### **GetMaxBytes**

說明: 從佇列擷取的最大持續及非持續訊息的大小。此參數是依持續性值編製索引的整數清單, 請參閱 [參照附註 2](#)。

ID: MQIAMO\_GET\_MAX\_BYTES

資料類型: MQCFIL

包含在 PCF 群組中: QAccountingData

已傳回: 可用時

### **BrowseCount**

說明: 持續及非持續訊息的成功非破壞性 MQGET 呼叫數。此參數是依持續性值編製索引的整數清單, 請參閱 [參照附註 2](#)。

ID: MQIAMO\_BROWSES

資料類型: MQCFIL

包含在 PCF 群組中: QAccountingData

已傳回: 可用時

### **BrowseFailCount**

說明: 不成功的非破壞性 MQGET 呼叫數

ID: MQIAMO\_BROWSES\_FAILED

資料類型: MQCFIN

包含在 PCF 群組中: QAccountingData

已傳回: 可用時

### **BrowseBytes**

說明: 在傳回持續訊息的非破壞性 MQGET 呼叫中讀取的位元組數

ID: MQIAMO64\_BROWSE\_BYTES

資料類型: MQCFIL64

包含在 PCF 群組中: *QAccountingData*

已傳回: 可用時

### **BrowseMinBytes**

說明: 從佇列瀏覽的最小持續及非持續訊息的大小。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO\_BROWSE\_MIN\_BYTES

資料類型: MQCFIL

包含在 PCF 群組中: *QAccountingData*

已傳回: 可用時

### **BrowseMaxBytes**

說明: 從佇列瀏覽的最大持續及非持續訊息的大小。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO\_BROWSE\_MAX\_BYTES

資料類型: MQCFIL

包含在 PCF 群組中: *QAccountingData*

已傳回: 可用時

### **TimeOnQMin**

說明: 在破壞性擷取之前，持續及非持續訊息留在佇列上的最短時間 (以微秒為單位)。對於在同步點下擷取的訊息，此值不包括在確定取得作業之前的時間。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_Q\_TIME\_MIN

資料類型: MQCFIL64

包含在 PCF 群組中: *QAccountingData*

已傳回: 可用時

### **TimeOnQAvg**

說明: 持續及非持續訊息在破壞性擷取之前留在佇列上的平均時間 (微秒)。對於在同步點下擷取的訊息，此值不包括在確定取得作業之前的時間。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_Q\_TIME\_AVG

資料類型: MQCFIL64

包含在 PCF 群組中: *QAccountingData*

已傳回: 可用時

### **TimeOnQMax**

說明: 在破壞性擷取之前，持續及非持續訊息留在佇列上的最長時間 (以微秒為單位)。對於在同步點下擷取的訊息，此值不包括在確定取得作業之前的時間。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_Q\_TIME\_MAX

資料類型: MQCFIL64

包含在 PCF 群組中: *QAccountingData*

已傳回: 可用時

### **CBCount**

說明: 成功 MQCB 要求數目。這是依作業類型編製索引的值陣列

0-已建立或變更回呼

1-已移除回呼

2-已回復回呼

3-已暫停回呼

ID: MQIAMO\_CBS

資料類型: MQCFIN

已傳回: 可用時。

### **CBFailCount**

說明: 失敗的 MQCB 要求數目。

ID: MQIAMO\_CBS\_FAILED

資料類型: MQCFIN

已傳回: 可用時。

## **MQI 統計資料訊息資料**

請利用這個頁面來檢視 MQI 統計資料訊息的結構

訊息名稱: MQI 統計資料訊息。

平台: 全部，但 IBM MQ for z/OS 除外。

系統佇列: SYSTEM.ADMIN.STATISTICS.QUEUE.

### **QueueManager**

說明: 佇列管理程式的名稱。

ID: MQCA\_Q\_MGR\_NAME。

資料類型: MQCFST。

長度上限: MQ\_Q\_MGR\_NAME\_LENGTH。

已傳回: 始終。

### ***IntervalStartDate***

說明：	監視期間開始的日期。
ID：	MQCAMO_START_DATE。
資料類型：	MQCFST。
長度上限：	MQ 資料長度
已傳回：	始終。

### ***IntervalStartTime***

說明：	監視期間開始時的時間。
ID：	MQCAMO_START_TIME。
資料類型：	MQCFST。
長度上限：	MQ 時間長度
已傳回：	始終。

### ***IntervalEndDate***

說明：	監視期間結束的日期。
ID：	MQCAMO_END_DATE。
資料類型：	MQCFST。
長度上限：	MQ 資料長度
已傳回：	始終。

### ***IntervalEndTime***

說明：	監視期間結束時的時間。
ID：	MQCAMO_END_TIME。
資料類型：	MQCFST。
長度上限：	MQ 時間長度
已傳回：	始終。

### ***CommandLevel***

說明：	佇列管理程式指令層次。
ID：	MQIA_COMMAND_LEVEL。
資料類型：	MQCFIN。
已傳回：	始終。

### ***ConnCount***

說明：	與佇列管理程式的成功連線數目。
ID：	MQIAMO_CONNS。
資料類型：	MQCFIN。
已傳回：	可用時。

### **ConnFailCount**

說明： 失敗的連線嘗試次數。  
ID： MQIAMO\_CONNS\_FAILED。  
資料類型： MQCFIN。  
已傳回： 可用時。

### **ConnsMax**

說明： 記錄間隔中的並行連線數目上限。  
ID： MQIAMO\_CONNS\_MAX。  
資料類型： MQCFIN。  
已傳回： 可用時。

### **DiscCount**

說明： 與佇列管理程式中斷連線的數目。這是一個整數陣列，由下列常數編製索引：  

- MQDISCONNECT\_NORMAL
- MQDISCONNECT\_IMPLICIT
- MQDISCONNECT\_Q\_MGR

ID： MQIAMO\_光碟。  
資料類型： MQCFIL。  
已傳回： 可用時。

### **OpenCount**

說明： 透過直接對 MQOPEN 發出呼叫或使用 MQPUT1 動詞來順利開啟的物件數。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。  
ID： MQIAMO\_OPENS。  
資料類型： MQCFIL。  
已傳回： 可用時。

### **OpenFailCount**

說明： 不成功的開啟物件嘗試次數。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。  
ID： MQIAMO\_OPENS\_FAILED。  
資料類型： MQCFIL。  
已傳回： 可用時。

### **CloseCount**

說明： 順利關閉的物件數。此參數是依物件類型編製索引的整數清單，請參閱 [參照附註 1](#)。  
ID： MQIAMO\_CLOSES。  
資料類型： MQCFIL。  
已傳回： 可用時。

### **CloseFailCount**

說明:	失敗的關閉物件嘗試次數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_CLOSES_FAILED。
資料類型:	MQCFIL。
已傳回:	可用時。

### **InqCount**

說明:	順利查詢的物件數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_INQS。
資料類型:	MQCFIL。
已傳回:	可用時。

### **InqFailCount**

說明:	不成功的物件查詢嘗試次數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_INQS_FAILED。
資料類型:	MQCFIL。
已傳回:	可用時。

### **SetCount**

說明:	已順利更新 (SET) 的物件數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_SETS。
資料類型:	MQCFIL。
已傳回:	可用時。

### **SetFailCount**

說明:	不成功的 SET 嘗試次數。此參數是依物件類型編製索引的整數清單，請參閱 <a href="#">參照附註 1</a> 。
ID:	MQIAMO_SETS_FAILED。
資料類型:	MQCFIL。
已傳回:	可用時。

### **PutCount**

說明:	已順利放入佇列中的持續及非持續訊息數，但 MQPUT1 要求除外。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PPUTS。
資料類型:	MQCFIL。
已傳回:	可用時。

### **PutFailCount**

說明：失敗的放置訊息嘗試次數。  
ID：MQIAMO\_PUTS\_FAILED。  
資料類型：MQCFIN。  
已傳回：可用時。

### **Put1Count**

說明：使用 MQPUT1 要求順利置入佇列的持續及非持續訊息數。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)  
ID：MQIAMO\_PUT1S。  
資料類型：MQCFIL。  
已傳回：可用時。

### **Put1FailCount**

說明：使用 MQPUT1 要求將持續及非持續訊息置入佇列的失敗嘗試次數。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)  
ID：MQIAMO\_PUT1S\_FAILED。  
資料類型：MQCFIL。  
已傳回：可用時。

### **PutBytes**

說明：使用放置要求寫入之持續及非持續訊息的位元組數。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)  
ID：MQIAMO64\_PUT\_BYTES。  
資料類型：MQCFIL64。  
已傳回：可用時。

### **GetCount**

說明：持續及非持續訊息的成功破壞性取得要求數目。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)  
ID：MQIAMO\_GETS。  
資料類型：MQCFIL。  
已傳回：可用時。

### **GetFailCount**

說明：失敗的破壞性取得要求數目。  
ID：MQIAMO\_GETS\_FAILED。  
資料類型：MQCFIN。  
已傳回：可用時。

### **GetBytes**

說明：持續及非持續訊息的破壞性取得要求中所讀取的位元組數。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)

ID: MQIAMO64\_GET\_BYTES.  
資料類型: MQCFIL64.  
已傳回: 可用時。

### **BrowseCount**

說明: 持續及非持續訊息的成功非破壞性取得要求數。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)

ID: MQIAMO\_BROWSES。  
資料類型: MQCFIL。  
已傳回: 可用時。

### **BrowseFailCount**

說明: 失敗的非破壞性取得要求數目。

ID: MQIAMO\_BROWSES\_FAILED。  
資料類型: MQCFIN。  
已傳回: 可用時。

### **BrowseBytes**

說明: 持續及非持續訊息之非破壞性取得要求中所讀取的位元組數。此參數是依持續性值檢索的整數清單，請參閱 [參照附註 2](#)

ID: MQIAMO64\_BROWSE\_BYTES。  
資料類型: MQCFIL64。  
已傳回: 可用時。

### **CommitCount**

說明: 順利完成的交易數。此數目包括應用程式斷線隱含地確定的交易，以及沒有未完成工作的確定要求。

ID: MQIAMO\_COMMITS。  
資料類型: MQCFIN。  
已傳回: 可用時。

### **CommitFailCount**

說明: 嘗試完成交易但失敗的次數。

ID: MQIAMO\_COMMITS\_FAILED。  
資料類型: MQCFIN。  
已傳回: 可用時。

### **BackCount**

說明: 已處理的取消數目（包括異常中斷連線時的隱含取消）。

ID: MQIAMO\_BACKOUTS。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***ExpiredMsgCount***

說明： 在可以擷取之前，因到期而捨棄的持續及非持續訊息數目。  
ID： MQIAMO\_MSGS\_EXPIRED。  
資料類型： MQCFIN。  
已傳回： 可用時。

### ***PurgeCount***

說明： 已清除佇列的次數。  
ID： MQIAMO\_MSGS\_PLEARDED。  
資料類型： MQCFIN。  
已傳回： 可用時。

### ***SubCountDur***

說明： 建立、變更或回復可延續訂閱的成功「訂閱」要求數。這是依作業類型編製索引的值陣列  
0 = 已建立的訂閱數  
1 = 已變更的訂閱數  
2 = 已回復的訂閱數  
ID： MQIAMO\_SUBS\_DUR。  
資料類型： MQCFIL  
已傳回： 可用時。

### ***SubCountNDur***

說明： 建立、變更或回復不可延續訂閱的成功「訂閱」要求數。這是依作業類型編製索引的值陣列  
0 = 已建立的訂閱數  
1 = 已變更的訂閱數  
2 = 已回復的訂閱數  
ID： MQIAMO\_SUBS\_NDUR。  
資料類型： MQCFIL。  
已傳回： 可用時。

### ***SubFailCount***

說明： 失敗的訂閱要求數目。  
ID： MQIAMO\_SUBS\_FAILED。  
資料類型： MQCFIN。  
已傳回： 可用時。

### ***UnsubCountDur***

說明：可延續訂閱的成功取消訂閱要求數。這是依作業類型編製索引的值陣列  
0-訂閱已關閉但未移除  
1-已關閉並移除訂閱

ID：MQIAMO\_UNSUBS\_DUR。

資料類型：MQCFIL。

已傳回：可用時。

### ***UnsubCountNDur***

說明：不可延續訂閱的成功取消訂閱要求數。這是依作業類型編製索引的值陣列  
0-訂閱已關閉但未移除  
1-已關閉並移除訂閱

ID：MQIAMO\_UNSUBS\_NDUR。

資料類型：MQCFIL。

已傳回：可用時。

### ***UnsubFailCount***

說明：失敗的取消訂閱要求數目。

ID：MQIAMO\_UNSUBS\_FAILED。

資料類型：MQCFIN。

已傳回：可用時。

### ***SubRqCount***

說明：成功的 MQSUBRQ 要求數目。

ID：MQIAMO\_SUBRQS

資料類型：MQCFIN

已傳回：可用時。

### ***SubRqFailCount***

說明：失敗的 MQSUBRQ 要求數目。

ID：MQIAMO\_SUBRQS\_FAILED。

資料類型：MQCFIN。

已傳回：可用時。

### ***CBCount***

說明：成功 MQCB 要求數目。這是依作業類型編製索引的值陣列  
0-已建立或變更回呼  
1-已移除回呼  
2-已回復回呼  
3-已暫停回呼

ID: MQIAMO\_CBS。  
資料類型: MQCFIL。  
已傳回: 可用時。

#### ***CBFailCount***

說明: 失敗的 MQCB 要求數目。  
ID: MQIamo\_CBS\_FAILED。  
資料類型: MQCFIN。  
已傳回: 可用時。

#### ***CtlCount***

說明: 順利完成的 MQCTL 要求數目。這是依作業類型編製索引的值陣列:  
0-已啟動連線  
1-連線已停止  
2-已回復連線  
3-已暫停連線

ID: MQIAMO\_CTLs。  
資料類型: MQCFIL。  
已傳回: 可用時。

#### ***CtlFailCount***

說明: 失敗的 MQCTL 要求數目。  
ID: MQIAMO\_CTLs\_FAILED。  
資料類型: MQCFIN。  
已傳回: 可用時。

#### ***StatCount***

說明: 成功的 MQSTAT 要求數目。  
ID: MQIAMO\_STATS。  
資料類型: MQCFIN。  
已傳回: 可用時。

#### ***StatFailCount***

說明: 失敗的 MQSTAT 要求數目。  
ID: MQIAMO\_STATS\_FAILED。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***SubCountDurHighWater***

說明： 在時間間隔期間可延續訂閱數的高臨界值。這是由 SUBTYPE 編製索引的值陣列

- 0-系統中所有可延續訂閱的高臨界值
- 1-可延續應用程式訂閱的高臨界值 (MQSUBTYPE\_API)
- 2-可延續管理訂閱 (MQSUBTYPE\_ADMIN) 的高臨界值
- 3-可延續 Proxy 訂閱的高臨界值 (MQSUBTYPE\_PROXY)

ID: MQIAMO\_SUB\_DUR\_HIGHW 水

資料類型: MQCFIL。

已傳回: 可用時。

### ***SubCountDurLowWater***

說明： 在時間間隔期間，可延續訂閱數的低臨界值。這是 SUBTYPE 所編製索引的值陣列。

- 0-系統中所有可延續訂閱的低臨界值
- 1-可延續應用程式訂閱的低臨界值 (MQSUBTYPE\_API)
- 2-可延續管理訂閱的低臨界值 (MQSUBTYPE\_ADMIN)
- 3-可延續 Proxy 訂閱的低臨界值 (MQSUBTYPE\_PROXY)

ID: MQIAMO\_SUB\_DUR\_LOWW 水

資料類型: MQCFIL。

已傳回: 可用時。

### ***SubCountNDurHighWater***

說明： 在時間間隔期間不可延續訂閱數的高臨界值。這是由 SUBTYPE 編製索引的值陣列

- 0-系統中所有不可延續訂閱的高臨界值
- 1-不可延續應用程式訂閱的高臨界值 (MQSUBTYPE\_API)
- 2-不可延續管理訂閱 (MQSUBTYPE\_ADMIN) 的高臨界值
- 3-不可延續 Proxy 訂閱的高臨界值 (MQSUBTYPE\_PROXY)

ID: MQIAMO\_SUB\_NDUR\_HIGHW 水

資料類型: MQCFIL。

已傳回: 可用時。

### ***SubCountNDurLowWater***

說明： 時間間隔期間不可延續訂閱數的低臨界值。這是 SUBTYPE 所編製索引的值陣列。

- 0-系統中所有不可延續訂閱的低臨界值
- 1-不可延續應用程式訂閱的低臨界值 (MQSUBTYPE\_API)
- 2-不可延續管理訂閱的低臨界值 (MQSUBTYPE\_ADMIN)
- 3-不可延續 Proxy 訂閱的低臨界值 (MQSUBTYPE\_PROXY)

ID: MQIAMO\_SUB\_NDUR\_LOWW 水

資料類型: MQCFIL。

已傳回: 可用時。

### ***PutTopicCount***

說明: 順利放置至主題的持續及非持續訊息數，但使用 MQPUT1 呼叫放置的訊息除外。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

附註: 此值包括使用佇列別名 (解析為主題) 所放置的訊息。

ID: MQIAMO\_TOPIC\_PUT。

資料類型: MQCFIL。

已傳回: 可用時。

### ***PutTopicFailCount***

說明: 將訊息置入主題的失敗嘗試次數。

ID: MQIAMO\_TOPIC\_PUTS\_FAILED。

資料類型: MQCFIN。

已傳回: 可用時。

### ***Put1TopicCount***

說明: 使用 MQPUT1 呼叫順利放置到主題的持續及非持續訊息數。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

附註: 此值包括使用佇列別名 (解析為主題) 所放置的訊息。

ID: MQIAMO\_TOPIC\_PUT1S。

資料類型: MQCFIL。

已傳回: 可用時。

### ***Put1TopicFailCount***

說明: 使用 MQPUT1 呼叫將訊息放置到主題的失敗嘗試次數。

ID: MQIAMO\_TOPIC\_PUT1S\_FAILED。

資料類型: MQCFIN。

已傳回: 可用時。

### ***PutTopicBytes***

說明: 針對解析為發佈作業的持續及非持續訊息使用放置呼叫所寫入的位元組數。這是應用程式所放置的位元組數，而不是遞送至訂閱者的結果位元組數，請參閱此值的 PublishMsg 位元組。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_TOPIC\_PUT\_BYTES。

資料類型: MQCFIL64。

已傳回: 可用時。

### ***PublishMsgCount***

說明: 在時間間隔內遞送至訂閱的訊息數。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_PUBLISH\_MSG\_COUNT  
資料類型: MQCFIL。  
已傳回: 可用時。

### ***PublishMsgBytes***

說明: 在時間間隔內遞送至訂閱的位元組數。此參數是依持續性值編製索引的整數清單，請參閱 [參照附註 2](#)。

ID: MQIAMO64\_PUBLISH\_MSG\_BYTES  
資料類型: MQCFIL64。  
已傳回: 可用時。

## **佇列統計資料訊息資料**

請利用這個頁面來檢視佇列統計資料訊息的結構

訊息名稱:	佇列統計資料訊息。
平台:	全部，但 IBM MQ for z/OS 除外。
系統佇列:	SYSTEM.ADMIN.STATISTICS.QUEUE.

### ***QueueManager***

說明: 佇列管理程式的名稱  
ID: MQCA\_Q\_MGR\_NAME  
資料類型: MQCFST  
長度上限: MQ\_Q\_MGR\_NAME\_LENGTH  
已傳回: 一律

### ***IntervalStartDate***

說明: 監視期間開始的日期  
ID: MQCAMO\_START\_DATE  
資料類型: MQCFST  
長度上限: MQ 資料長度  
已傳回: 一律

### ***IntervalStartTime***

說明: 監視期間開始時的時間  
ID: MQCAMO\_START\_TIME  
資料類型: MQCFST  
長度上限: MQ 時間長度  
已傳回: 一律

### ***IntervalEndDate***

說明: 監視期間結束時的日期  
ID: MQCAMO\_END\_DATE  
資料類型: MQCFST

長度上限: MQ 資料長度  
已傳回: 一律

### ***IntervalEndTime***

說明: 監視期間結束時的時間  
ID: MQCAOM\_END\_TIME  
資料類型: MQCFST  
長度上限: MQ 時間長度  
已傳回: 一律

### ***CommandLevel***

說明: 佇列管理程式指令層次  
ID: MQIA\_COMMAND\_LEVEL  
資料類型: MQCFIN  
已傳回: 一律

### ***ObjectCount***

說明: 在已記錄統計資料的間隔內存取的佇列物件數。此值設為訊息中包含的 QStatisticsData PCF 群組數目。  
ID: MQIAMO\_OBJECT\_COUNT  
資料類型: MQCFIN  
已傳回: 一律

### ***QStatisticsData***

說明: 指定佇列統計資料明細的分組參數  
ID: MQGACF\_Q\_STATISTICS\_DATA  
資料類型: MQCFGR

群組中的參數:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- QMinDepth*
- QMaxDepth*
- AvgTimeOnQ*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- NonQueuedMsgCount*
- ExpiredMsgCount*
- PurgeCount*

已傳回: 一律

### ***QName***

說明: 佇列的名稱

ID: MQCA\_Q\_NAME

資料類型: MQCFST

長度上限: MQ\_Q\_NAME\_LENGTH

已傳回: 一律

### ***CreateDate***

說明: 建立佇列的日期

ID: MQCA\_CREATION\_DATE

資料類型: MQCFST

長度上限: MQ 資料長度

已傳回: 一律

### ***CreateTime***

說明: 建立佇列的時間

ID: MQCA\_CREATION\_TIME

資料類型: MQCFST

長度上限: MQ 時間長度

已傳回: 一律

### **QType**

說明:	佇列的類型
ID:	MQIA_Q_TYPE
資料類型:	MQCFIN
值:	本端 MQOT_LOCAL
已傳回:	一律

### **QDefinitionType**

說明:	佇列定義類型
ID:	MQIA_DEFINITION_TYPE
資料類型:	MQCFIN
值:	可能的值為 <ul style="list-style-type: none"><li>• MQQDT_PREDEFINED</li><li>• MQQDT_PERMANENT_DYNAMIC</li><li>• MQQDT_TEMPORARY_DYNAMIC</li></ul>
已傳回:	可用時

### **QMinDepth**

說明:	監視期間的佇列深度下限
ID:	MQIAMO_Q_MIN_DEPTH
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **QMaxDepth**

說明:	監視期間的佇列深度上限
ID:	MQIAMO_Q_MAX_DEPTH
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **AvgTimeOnQ**

說明:	在監視期間，破壞性地從佇列擷取訊息的平均延遲時間 (微秒)。此參數是依持續性值編製索引的整數清單，請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO64_AVG_Q_TIME
資料類型:	MQCFIL64
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **PutCount**

說明:	已順利放入佇列中的持續及非持續訊息數，但 MQPUT1 要求除外。此參數是依持續性值檢索的整數清單。請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **PutFailCount**

說明:	嘗試將訊息放入佇列但失敗的次數
ID:	MQIAMO_PUTS_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **Put1Count**

說明:	使用 MQPUT1 呼叫順利置入佇列的持續及非持續訊息數。此參數是依持續性值檢索的整數清單。請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_PUT1S
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **Put1FailCount**

說明:	使用 MQPUT1 呼叫放置訊息但失敗的嘗試次數
ID:	MQIAMO_PUT1S_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **PutBytes**

說明:	在佇列中寫入放置要求的位元組數
ID:	MQIAMO64_PUT_BYTES
資料類型:	MQCFIL64
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **GetCount**

說明:	持續及非持續訊息的成功破壞性取得要求數目。此參數是依持續性值檢索的整數清單。請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_GETS
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **GetFailCount**

說明:	不成功的破壞性取得要求數
ID:	MQIAMO_GETS_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **GetBytes**

說明:	持續及非持續訊息之破壞性放置要求中所讀取的位元組數。此參數是依持續性值檢索的整數清單。請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO64_GET_BYTES
資料類型:	MQCFIL64
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **BrowseCount**

說明:	持續及非持續訊息的成功非破壞性取得要求數。此參數是依持續性值檢索的整數清單。請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO_BROWSES
資料類型:	MQCFIL
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **BrowseFailCount**

說明:	不成功的非破壞性取得要求數
ID:	MQIAMO_BROWSES_FAILED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **BrowseBytes**

說明:	持續及非持續訊息之非破壞性取得要求中所讀取的位元組數。此參數是依持續性值檢索的整數清單。請參閱 <a href="#">參照附註 2</a> 。
ID:	MQIAMO64_BROWSE_BYTES
資料類型:	MQCFIL64
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **NonQueuedMsgCount**

說明:	略過佇列並直接傳送至等待中應用程式的訊息數。 略過佇列只能在特定情況下發生。此數字代表 IBM MQ 能夠略過佇列的次數，而不是應用程式等待的次數。
ID:	MQIAMO_MSGS_NOT_QUEUED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **ExpiredMsgCount**

說明:	因為在可以擷取之前已過期而捨棄的持續及非持續訊息數。
ID:	MQIAMO_MSGS_EXPIRED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **PurgeCount**

說明:	已清除的訊息數。
ID:	MQIAMO_MSGS_PLEADED
資料類型:	MQCFIN
包含在 PCF 群組中:	<i>QStatisticsData</i>
已傳回:	可用時

### **通道統計資料訊息資料**

請利用這個頁面來檢視通道統計資料訊息的結構

訊息名稱:	通道統計資料訊息。
平台:	全部，但 IBM MQ for z/OS 除外。
系統佇列:	SYSTEM.ADMIN.STATISTICS.QUEUE.

### **QueueManager**

說明:	佇列管理程式的名稱。
-----	------------

ID: MQCA\_Q\_MGR\_NAME。  
資料類型: MQCFST。  
長度上限: MQ\_Q\_MGR\_NAME\_LENGTH。  
已傳回: 始終。

### ***IntervalStartDate***

說明: 監視期間開始的日期。  
ID: MQCAMO\_START\_DATE。  
資料類型: MQCFST。  
長度上限: MQ\_DATE\_LENGTH。  
已傳回: 始終。

### ***IntervalStartTime***

說明: 監視期間開始時的時間。  
ID: MQCAMO\_START\_TIME。  
資料類型: MQCFST。  
長度上限: MQ\_TIME\_LENGTH。  
已傳回: 始終。

### ***IntervalEndDate***

說明: 監視期間結束時的日期。  
ID: MQCAMO\_END\_DATE。  
資料類型: MQCFST。  
長度上限: MQ\_DATE\_LENGTH。  
已傳回: 始終。

### ***IntervalEndTime***

說明: 監視期間結束時的時間。  
ID: MQCAMO\_END\_TIME。  
資料類型: MQCFST。  
長度上限: MQ 時間長度。  
已傳回: 始終。

### ***CommandLevel***

說明: 佇列管理程式指令層次。  
ID: MQIA\_COMMAND\_LEVEL。  
資料類型: MQCFIN。  
已傳回: 始終。

### ***ObjectCount***

說明: 在已記錄統計資料的間隔內存取的通道物件數。此值設為訊息中包含的 ChIStatistics 資料 PCF 群組數目。

ID: MQIAMO\_OBJECT\_COUNT  
資料類型: MQCFIN。  
已傳回: 始終。

### ***ChlStatisticsData***

說明: 指定通道的統計資料明細的分組參數。

ID: MQGACF\_CHL\_STATISTICS\_DATA。

資料類型: MQCFGR。

群組中的參數:

- ChannelName*
- ChannelType*
- RemoteQmgr*
- ConnectionName*
- MsgCount*
- TotalBytes*
- NetTimeMin*
- NetTimeAvg*
- NetTimeMax*
- ExitTimeMin*
- ExitTimeAvg*
- ExitTimeMax*
- FullBatchCount*
- IncplBatchCount*
- AverageBatchSize*
- PutRetryCount*

已傳回: 始終。

### ***ChannelName***

說明: 頻道的名稱。

ID: MQCACH\_CHANNEL\_NAME。

資料類型: MQCFST。

長度上限: MQ\_CHANNEL\_NAME\_LENGTH。

已傳回: 始終。

### ***ChannelType***

說明: 通道類型。

ID: MQIACH\_CHANNEL\_TYPE。

資料類型: MQCFIN。

值: 可能的值為:

**MQCHT\_SENDER**  
傳送端通道。

**MQCHT\_SERVER**  
伺服器通道。

**MQCHT\_RECEIVER**  
接收端通道。

**MQCHT\_REQUESTER**  
要求端通道。

**MQCHT\_CLUSRCVR**  
叢集接收端通道。

**MQCHT\_CLUSSDR**  
叢集傳送端通道。

已傳回: 始終。

### **RemoteQmgr**

說明: 遠端佇列管理程式的名稱。

ID: MQCA\_REMOTE\_Q\_MGR\_NAME。

資料類型: MQCFST。

長度上限: MQ\_Q\_MGR\_NAME\_LENGTH

已傳回: 可用時。

### **ConnectionName**

說明: 遠端佇列管理程式的連線名稱。

ID: MQCACH\_CONNECTION\_NAME。

資料類型: MQCFST

長度上限: MQ\_CONN\_NAME\_LENGTH

已傳回: 可用時。

### **MsgCount**

說明: 傳送或接收的持續及非持續訊息數。

ID: MQIAMO\_MSGS。

資料類型: MQCFIN

已傳回: 可用時。

### **TotalBytes**

說明: 針對持續及非持續訊息所傳送或接收的位元組數。

ID: MQIAMO64\_BYTES。

資料類型: MQCFIN64.

已傳回: 可用時。

### **NetTimeMin**

說明: 在記錄間隔中測量的最短記錄通道來回時間 (以微秒為單位)。

ID: MQIAMO\_NET\_TIME\_MIN。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***NetTimeAvg***

說明: 在記錄間隔內測量的平均記錄通道來回轉換 (以微秒為單位)。  
ID: MQIAMO\_NET\_TIME\_AVG。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***NetTimeMax***

說明: 在記錄間隔中測量的最長記錄通道來回時間 (以微秒為單位)。  
ID: MQIAMO\_NET\_TIME\_MAX。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***ExitTimeMin***

說明: 在記錄間隔內執行使用者結束程式所耗費的最短記錄時間 (以微秒為單位)。  
ID: MQIAMO\_EXIT\_TIME\_MIN。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***ExitTimeAvg***

說明: 在記錄間隔內執行使用者結束程式所耗費的平均記錄時間 (微秒)。以微秒為單位測量。  
ID: MQIAMO\_EXIT\_TIME\_AVG。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***ExitTimeMax***

說明: 在記錄間隔內執行使用者結束程式所耗費的最長記錄時間 (以微秒為單位)。以微秒為單位測量。  
ID: MQIAMO\_EXIT\_TIME\_MAX。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***FullBatchCount***

說明: 由於達到通道屬性 BATCHSZ 或 BATCHLIM 的值而由通道處理且已傳送的批次數。  
ID: MQIAMO\_FULL\_BATCHES。  
資料類型: MQCFIN。  
已傳回: 可用時。

### ***IncomplBatchCount***

說明：通道所處理且未達到通道屬性 BATCHSZ 值的已傳送批次數。  
ID：MQIAMO\_INCOMPLETE\_BATCHES。  
資料類型：MQCFIN。  
已傳回：可用時。

### ***AverageBatchSize***

說明：通道所處理之批次的平均批次大小。  
ID：MQIAMO\_AVG\_BATCH\_SIZE。  
資料類型：MQCFIN。  
已傳回：可用時。

### ***PutRetryCount***

說明：在時間間隔內無法放置訊息並進入重試迴圈的次數。  
ID：MQIAMO\_PUT\_RETRIES。  
資料類型：MQCFIN。  
已傳回：可用時。

### **參照附註**

請利用這個頁面來檢視帳戶和統計資料訊息的結構說明所參照的附註

下列訊息資料說明參照這些附註：

- [第 137 頁的『MQI 結算訊息資料』](#)
- [第 147 頁的『佇列帳戶訊息資料』](#)
- [第 157 頁的『MQI 統計資料訊息資料』](#)
- [第 168 頁的『佇列統計資料訊息資料』](#)
- [第 174 頁的『通道統計資料訊息資料』](#)

1. 此參數與 IBM MQ 物件相關。此參數是由下列常數檢索的值陣列 (MQCFIL 或 MQCFIL64)：

物件類型	值環境定義
MQOT_Q (1)	包含與佇列物件相關的值。
MQOT_NAMELIST (2)	包含與名單物件相關的值。
MQOT_PROCESS (3)	包含與處理程序物件相關的值。
MQOT_Q_MGR (5)	包含與佇列管理程式物件相關的值。
MQOT_CHANNEL (6)	包含與通道物件相關的值。
MQOT_AUTH_INFO (7)	包含與鑑別資訊物件相關的值。
MQOT_TOPIC (8)	包含與主題物件相關的值。

註：會傳回 13 個 MQCFIL 或 MQCFIL64 值的陣列，但只有列出的那些值才有意義。

2. 此參數與 IBM MQ 訊息相關。此參數是由下列常數檢索的值陣列 (MQCFIL 或 MQCFIL64)：

表 25: 依持續性值編製索引的陣列	
常數	值
1	包含非持續訊息的值。
2	包含持續訊息的值。

註: 每一個陣列的索引都從零開始, 因此索引 1 是指陣列的第二列。這些表格中未列出的這些陣列元素未包含任何帳戶或統計資料資訊。

## 應用程式活動追蹤

應用程式活動追蹤會產生連接至佇列管理程式之應用程式行為的詳細資訊。它會追蹤應用程式的行為, 並在應用程式與 IBM MQ 資源互動時提供應用程式所使用參數的詳細視圖。它也會顯示應用程式所發出的 MQI 呼叫順序。

當您需要的資訊超過事件監視、訊息監視、結算及統計資料訊息及即時監視所提供的資訊時, 請使用應用程式活動追蹤。

註: 會在每一個應用程式的 IBM MQ 連線產生活動追蹤; 因此, 如果啟用結算訊息, 則活動追蹤作業會計入每一個應用程式的 MQI 結算資訊。

IBM MQ 支援兩種收集應用程式活動追蹤資料的方法。

- 集中收集應用程式活動追蹤資訊, 其中透過將活動追蹤 PCF 訊息寫入系統佇列 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE。
- 訂閱活動追蹤資料, 寫入特殊 IBM MQ 系統主題。

請注意, IBM MQ for z/OS 不支援活動追蹤。

 由於您現在可以在 IBM MQ 支援的大部分程式設計語言上指定應用程式名稱, 如需相關資訊, 請參閱 [以支援的程式設計語言指定應用程式名稱](#)。

## 配置應用程式活動追蹤資訊的中央集合

應用程式活動追蹤訊息是 PCF 訊息。您可以使用配置檔來配置活動追蹤。若要配置集中收集應用程式活動追蹤資訊, 請設定 ACTVTRC 佇列管理程式屬性。您可以使用 MQCONNX 選項在連線層次置換此設定, 或使用活動追蹤配置檔在應用程式段落層次置換此設定。

## 關於這項作業

活動追蹤訊息由 MQMD 結構組成: PCF (MQCFH) 標頭結構, 後面接著一些 PCF 參數。一系列 ApplicationTrace 資料 PCF 群組遵循 PCF 參數。這些 PCF 群組會收集應用程式在連接至佇列管理程式時所執行 MQI 作業的相關資訊。您可以使用稱為 mqat.ini 的配置檔來配置活動追蹤。

若要控制是否收集應用程式活動追蹤資訊, 請配置下列一或多個設定:

1. ACTVTRC 佇列管理程式屬性。
2. ACTVCONO 設定 (在 MQCONNX 中傳遞的 MQCNO 結構中)。
3. 活動追蹤配置檔 mqat.ini 中應用程式的相符合段落。

前一個順序是重要的。ACTVTRC 屬性會被 ACTVCONO 設定置換, 而這些設定會被 mqat.ini 檔案中的設定置換。

除非另有說明, 否則會在每一個作業完成之後寫入追蹤項目。這些項目會先寫入系統佇列 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE, 然後在應用程式與佇列管理程式中斷連線時寫入應用程式活動追蹤訊息。對於長時間執行的應用程式, 如果發生下列任何事件, 則會寫入中間訊息:

- 連線的生命期限達到定義的逾時值。
- 作業數達到指定的數目。
- 在記憶體中收集的資料量達到佇列容許的訊息長度上限。

您可以使用 **ActivityInterval** 參數來設定逾時值。您可以使用 **ActivityCount** 參數來設定作業數。這兩個參數都指定在活動追蹤配置檔 `mqat.ini` 中。

啟用應用程式活動追蹤可能會影響效能。可以透過調整 **ActivityCount** 和 **ActivityInterval** 設定來減少額外負擔。請參閱第 186 頁的『調整應用程式活動追蹤的效能影響』。

檢視應用程式活動追蹤訊息內容最簡單的方式是使用第 187 頁的『`amqsact` 範例程式』。

## 程序

1. 第 181 頁的『設定 **ACTVTRC** 以控制活動追蹤資訊的收集』。
2. 第 181 頁的『設定 **MQCONN** 選項以控制活動追蹤資訊的收集』。
3. 第 182 頁的『使用 `mqat.ini` 來配置活動追蹤行為』。
4. 第 186 頁的『調整應用程式活動追蹤的效能影響』。

## 設定 **ACTVTRC** 以控制活動追蹤資訊的收集

使用佇列管理程式屬性 **ACTVTRC** 來控制 MQI 應用程式活動追蹤資訊的收集

## 關於這項作業

只有在啟用應用程式活動追蹤之後開始的連線才會產生應用程式活動追蹤訊息。**ACTVTRC** 參數可以具有下列值：

### 開啟

已啟用 API 活動追蹤收集

### 關閉

已停用 API 活動追蹤收集

註：佇列管理程式 **ACTVCONO** 參數可以置換 **ACTVTRC** 設定。如果您將 **ACTVCONO** 參數設為 **ENABLED**，則可以使用 **MQCNO** 結構中的 **Options** 欄位來置換給定連線的 **ACTVTRC** 設定。請參閱第 181 頁的『設定 **MQCONN** 選項以控制活動追蹤資訊的收集』。

## 範例

若要變更 **ACTVTRC** 參數的值，請使用 MQSC 指令 `ALTER QMGR`。例如，若要啟用 MQI 應用程式活動追蹤資訊收集，請使用下列 MQSC 指令：

```
ALTER QMGR ACTVTRC(ON)
```

## 下一步

檢視應用程式活動追蹤訊息內容最簡單的方式是使用第 187 頁的『`amqsact` 範例程式』。

啟用應用程式活動追蹤可能會影響效能。可以透過調整 **ActivityCount** 和 **ActivityInterval** 設定來減少額外負擔。請參閱第 186 頁的『調整應用程式活動追蹤的效能影響』。

## 設定 **MQCONN** 選項以控制活動追蹤資訊的收集

如果佇列管理程式屬性 **ACTVCONO** 設為 **ENABLED**，您可以在 **MQCONN** 呼叫上使用 **ConnectOpts** 參數，以根據每個連線來啟用或停用應用程式活動報告。這些選項會置換佇列管理程式屬性 **ACTVTRC** 所定義的活動追蹤行為，並可由活動追蹤配置檔 `mqat.ini` 中的設定置換。

## 程序

1. 將佇列管理程式屬性 **ACTVCONO** 設為 **ENABLED**。

註：如果應用程式嘗試使用 **ConnectOpts** 參數來修改應用程式的帳戶行為，且 **QMGR** 屬性 **ACTVCONO** 設為 **DISABLED**，則不會傳回任何錯誤給應用程式，且活動追蹤收集是由佇列管理程式屬性或活動追蹤配置檔 `mqat.ini` 所定義。

2. 將 **MQCONN** 呼叫中的 **ConnectOpts** 參數設為 `MQCNO_ACTIVITY_TRACE_ENABLED`。

MQCONNX 呼叫的 **ConnectOpts** 參數可以具有下列值:

#### 已停用 **MQCNO\_ACTIVITY\_TRACE\_DISABLED**

已停用連線的活動追蹤。

#### **MQCNO\_ACTIVITY\_TRACE\_ENABLED**

已啟用連線的活動追蹤。

註: 如果應用程式同時為 MQCONNX 選取 **MQCNO\_ACTIVITY\_TRACE\_ENABLED** 及 **MQCNO\_ACTIVITY\_TRACE\_DISABLED**, 則呼叫會失敗, 原因碼為 **MQRC\_OPTIONS\_ERROR**。

3. 請檢查活動追蹤配置檔 `mqat.ini` 中的設定是否未置換這些活動追蹤設定。

請參閱 [第 182 頁的『使用 `mqat.ini` 來配置活動追蹤行為』](#)。

## 下一步

檢視應用程式活動追蹤訊息內容最簡單的方式是使用 [第 187 頁的『`amqsact` 範例程式』](#)。

啟用應用程式活動追蹤可能會影響效能。可以透過調整 **ActivityCount** 和 **ActivityInterval** 設定來減少額外負擔。請參閱 [第 186 頁的『調整應用程式活動追蹤的效能影響』](#)。

## 使用 `mqat.ini` 來配置活動追蹤行為

活動追蹤行為是使用稱為 `mqat.ini` 的配置檔來配置。此檔案用來定義報告活動追蹤資料的層次及頻率。此檔案也提供一種方法來定義規則, 以根據應用程式名稱來啟用及停用活動追蹤。

## 關於這項作業

**Linux** **UNIX** 在 UNIX and Linux 系統上, `mqat.ini` 位於佇列管理程式資料目錄中, 該目錄與 `qm.ini` 檔案位於相同位置。

**Windows** 在 Windows 系統上, `mqat.ini` 位於佇列管理程式資料目錄 `C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name` 中。執行要追蹤之應用程式的使用者需要讀取此檔案的許可權。

註: 從 IBM WebSphere MQ 7.1 或更早版本移轉的佇列管理程式將會遺漏 `mqat.ini` 檔案。在這種情況下, 需要手動建立 `mqat.ini` 檔案, 且需要對檔案設定 660 許可權。

當修改 `mqat.ini` 檔時, 新建立的 IBM MQ 連線會根據已修改的版本來處理。除非變更佇列管理程式參數 (例如遵循 [ALTER QMGR](#) 指令), 否則現有連線將繼續使用舊版。

此檔案遵循與 `mqs.ini` 及 `qm.ini` 檔案相同的段落索引鍵及參數值配對格式。

該檔案由單一段落 **AllActivity 追蹤** 組成, 用於配置依預設針對所有活動追蹤報告活動追蹤資料的層次和頻率。

該檔案也可以包含多個 **ApplicationTrace** 段落。其中每一個, 根據比對連線的應用程式名稱與規則, 定義一個以上連線的追蹤行為規則。

### AllActivity 追蹤段落

除非置換, 否則單一 **AllActivity 追蹤** 段落會定義套用至所有 IBM MQ 連線之活動追蹤的設定。

**AllActivity 追蹤** 段落中的個別值可以由 **ApplicationTrace 段落** 中的更具體資訊置換。

如果指定多個 **AllActivity 追蹤** 段落, 則會使用最後一個段落中的值。所選 **AllActivity 追蹤** 中遺漏的參數會採用預設值。會忽略先前 **AllActivity 追蹤** 段落中的參數和值。

可以在 **AllActivity 追蹤** 段落下指定下列參數:

名稱	值 (以粗體類型表示的預設值)	說明
ActivityInterval	0-99999999 ( <b>1</b> )	追蹤訊息之間的大約時間間隔 (以秒為單位)。連線在該間隔內執行的所有活動都會寫入單一訊息中。如果此值為 0，則會在連線中斷時 (或達到活動計數時) 寫入追蹤訊息。
ActivityCount	0-99999999 ( <b>100</b> )	追蹤訊息之間的 MQI 或 XA 作業數。如果此值為 0，則會在連線中斷時 (或活動間隔已過時) 寫入追蹤訊息。
TraceLevel	LOW / 中 / HIGH	針對每一個作業所追蹤的參數詳細資料數量。個別作業的說明詳細說明針對每一個追蹤層次包括哪些參數。
TraceMessage 資料	<b>0</b> -104 857 600 (上限為 100 MB)	MQGET、MQPUT、MQPUT1 及回呼作業所追蹤的訊息資料量 (以位元組為單位)
StopOnGetTrace 訊息	<b>ON</b> /OFF	不建議使用活動追蹤來追蹤同時處理活動追蹤訊息的應用程式，因為可能發生迴圈。
SubscriptionDelivery	<b>批次處理</b> /IMMEDIATE	決定當有一或多個活動追蹤訂閱存在時，是否要使用 ActivityInterval 和 ActivityCount 參數。將此參數設為 IMMEDIATE 會導致在追蹤資料具有相符訂閱時，將 ActivityInterval 及 ActivityCount 值置換為有效值 1。每一個活動追蹤記錄不會與來自相同連線的其他記錄一起批次處理，而是立即遞送至訂閱，沒有延遲。

### ApplicationTrace 段落

ApplicationTrace 段落包含一個規則，定義將根據應用程式名稱來追蹤或不追蹤哪些 IBM MQ 連線。選擇性地，在 Allsettings 之下定義的預設行為，會置換廣域追蹤層次和頻率設定。

此段落可以包括 ApplName、ApplFunction 及 ApplClass 參數，根據「連線比對規則」中定義的比對規則來使用這些參數，以判定段落是否適用於特定連線。

段落必須包含 Trace 參數，以判定此規則是否針對相符連線開啟或關閉活動追蹤。

關閉規則可用來明確停用追蹤更具體的應用程式名稱，以及置換佇列管理程式或活動追蹤連線選項的 ACTVTRC 設定。

可以在 **ApplicationTrace** 段落下指定下列參數：

名稱	值 (以粗體類型表示的預設值)	說明
追蹤	ON/OFF (必要參數-無預設值)	活動追蹤開關。此切換參數可以在應用程式特定段落中使用，以判定現行應用程式段落範圍的活動追蹤是否為作用中。請注意，此值會置換佇列管理程式的 ACTVTRC 及 ACTVCONO 設定。

表 27: 可在應用程式追蹤配置檔中使用的參數/值配對 (繼續)

名稱	值 (以粗體類型表示的預設值)	說明
ApplName	字串 (必要參數-無預設值)	<p>此值用來決定 ApplicationTrace 段落適用於哪些應用程式。它與 API 結束程式環境定義結構 (相當於 MQMD.PutApplName) 中的 ApplName 值相符。ApplName 值的內容會根據應用程式環境而有所不同。</p> <p>對於 z/OS 以外的平台, 只有 MQAXC.ApplName 符合段落中的值。進行比較時, 會忽略最右邊路徑分隔字元左側的字元。</p> <p><b>z/OS</b> 若為 z/OS 應用程式, 則是整個 MQAXC.ApplName 符合段落中的值。</p> <p>在 ApplName 值的結尾可以使用單一萬用字元 (*), 以符合該點之後的任何字元數。如果 ApplName 值設為單一萬用字元 (*), 則 ApplName 值會符合所有應用程式。</p>
  ApplFunction	字串 (預設值 *)	<p>此值用來限定 ApplicationTrace 段落及 ApplName 值適用的應用程式。</p> <p>該段落是選用的, 但僅適用於 IBM i 佇列管理程式。ApplName 值結尾可以使用單一萬用字元 (*) 來符合任意數目的字元。</p> <p>例如, 指定 ApplName = * 及 ApplFunction = AMQSPUTO 的 ApplicationTrace 段落適用於任何工作中 AMQSPUTO 程式的所有呼叫。</p>
ApplClass	USER/MCA/ <b>ALL</b>	應用程式的類別。如需 AppType 值如何對應至 IBM MQ 連線的說明, 請參閱下表。

下表顯示 AppClass 值如何對應於連線 API 結束程式環境定義結構中的 APICallerType 和 APIEnvironment 欄位。

表 28: Appclass 值及其對應於 APICallerType 和 APIEnvironment 欄位的方式

APPLCLASS	API 呼叫程式類型:	API 環境:	說明
使用者	MQXACT_EXTERNAL	MQXE_OTHER	只追蹤使用者應用程式
MCA	(任何值)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	用戶端及通道 (amqrmppa)
ALL	(任何值)	(任何值)	追蹤所有連線



**小心:** 您必須對用戶端使用者應用程式使用 **APPLCLASS MCA**, 因為 **USER** 類別不符合這些類別。

例如, 若要追蹤 **amqspu**tc 範例應用程式, 您可以使用下列程式碼:

```

ApplicationTrace:
ApplClass=MCA                                # Application type
                                              # Values: (USER | MCA | INTERNAL | ALL)
                                              # Default: USER
ApplName=amqspu      # Application name (may be wildcard)
                                              # (matched to app name without path)
                                              # Default: *
Trace=ON                                       # Activity trace switch for application
                                              # Values: ( ON | OFF )
                                              # Default: OFF
ActivityInterval=30                          # Time interval between trace messages
    
```

```

ActivityCount=1          # Values: 0-99999999 (0=off)
                        # Default: 0
                        # Number of operations between trace msgs
TraceLevel=MEDIUM      # Values: 0-99999999 (0=off)
                        # Default: 0
                        # Amount of data traced for each operation
TraceMessageData=1000  # Values: LOW | MEDIUM | HIGH
                        # Default: MEDIUM
                        # Amount of message data traced
                        # Values: 0-1000000000
                        # Default: 0

```

建立佇列管理程式時所產生的預設 `mqat.ini` 包含單一規則，可明確停用所提供活動追蹤範例 **amqsact** 的活動追蹤。

## 連線比對規則

佇列管理程式會套用下列規則，以決定要用於連線的段落設定。

1. 連線會使用 **AllActivityTrace** 段落中指定的值，除非該值也出現在 **ApplicationTrace** 段落中，且該段落符合點 2、3 及 4 中所說明的連線比對準則。
2. **App1Class** 符合 IBM MQ 連線的類型。如果 **App1Class** 不符合連線類型，則會忽略此連線的段落。
3. 段落中的 *App1Name* 值符合連線之 API 結束程式環境定義結構 (MQAXC) 中 **App1Name** 欄位的檔名部分。

檔名部分衍生自最終路徑分隔字元 (/或 \) 右側的字元。如果段落 **App1Name** 包含萬用字元 (\*), 則只會將萬用字元左側的字元與連線 **App1Name** 中的相等字元數進行比較。

例如，如果指定段落值 "FRE\*", 則比較中只會使用前三個字元，因此 "path/FREEDOM" 與 "path\FREDDY" 相符，但 "path/FRIEND" 不相符。如果段落的 *App1Name* 值不符合連線 **App1Name**，則此連線會忽略該段落。

4. 如果多個段落符合連線的 **App1Name** 及 **App1Class**，則會使用具有最特定 **App1Name** 的段落。

最明確的 **App1Name** 定義為使用最多字元來符合連線 **App1Name** 的項目。

例如，如果 ini 檔案包含具有 **App1Name** = "FRE\*" 的段落以及另一個具有 **App1Name** = "FREE\*" 的段落，則會選擇具有 **App1Name** = "FREE\*" 的段落作為與 **App1Name** = "path/FREEDOM" 連線的最佳相符項，因為它符合四個字元 (而 **App1Name** = "FRE\*" 只符合三個字元)。

5. 在點 2、3 及 4 中套用規則之後，如果有多個段落符合連線 **App1Name** 及 **App1Class**，則會使用最後一個相符項中的值，並忽略所有其他段落。

## 置換每一個規則的預設值

您可以選擇性地針對符合 **ApplicationTrace** 段落的那些連線，置換 **AllActivity** 追蹤段落下的廣域追蹤層次和頻率設定。

下列參數可以在 **ApplicationTrace** 段落下設定。如果未設定，則會從 **AllActivity** 追蹤段落設定繼承值：

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**
- **StopOnTraceMsg**

## mqat.ini 語法

mqat.ini 檔案格式的語法規則如下：

- 以雜湊或分號開頭的文字會被視為延伸到行尾的註解。
- 第一行重要 (非註解) 必須是段落索引鍵。
- 段落索引鍵由後面接著冒號的段落名稱組成。
- 參數-值配對是由參數名稱後面接著等號，然後再接著值所組成。
- 只有單一參數值配對可以出現在一行上。(參數值不得折返到另一行)。

- 系統不處理前導和尾端空格。段落名稱、參數名稱和值或參數/值配對之間的空格數量沒有限制。換行很重要，且不會被忽略
- 任何一行的長度上限為 2048 個字元
- 段落索引鍵、參數名稱及常數參數值不區分大小寫，但變數參數值 (*AppName* 及 *DebugPath*) 區分大小寫。

## 應用程式活動追蹤檔範例

下列範例顯示如何在「活動追蹤 ini」檔案中指定配置資料。

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
AppName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
AppName=amqsact*
Trace=OFF
```

上述 **AllActivity** 追蹤段落定義啟用時，依預設如何執行活動追蹤 (透過 `ApplicationTrace` 規則或透過佇列管理程式 `ACTVTRC` 屬性，或由應用程式以程式化方式啟用)。

第一個 **ApplicationTrace** 段落定義的規則會導致追蹤名稱以 "amqs" 開頭的應用程式進行任何 MQI 活動。針對這些應用程式所產生的追蹤將會非常詳細，且最多包含 1000 個位元組的訊息資料。將繼承活動間隔和計數參數

第二個 **ApplicationTrace** 段落定義一個規則，可針對名稱開頭為 "amqsact" (活動追蹤範例) 的應用程式關閉追蹤。此規則會置換 amqsact 應用程式先前的 'on' 規則，導致該應用程式沒有追蹤。

範例也會在 C samples 目錄 (與 amqsact.c 檔案相同的目錄) 中提供稱為 mqat.ini 的範例。對於已從舊版 IBM MQ 移轉的佇列管理程式，此檔案可以複製到佇列管理程式資料目錄。

## 下一步

啟用應用程式活動追蹤可能會影響效能。可以透過調整 **ActivityCount** 和 **ActivityInterval** 設定來減少額外負擔。請參閱第 186 頁的『調整應用程式活動追蹤的效能影響』。

## 調整應用程式活動追蹤的效能影響

啟用應用程式活動追蹤可能會導致效能損失。只要追蹤您需要的應用程式，增加耗盡佇列的應用程式數目，以及在 mqat.ini 中調整 **ActivityInterval**、**ActivityCount** 和 **TraceLevel**，即可減少此情況。

## 關於這項作業

針對應用程式或所有佇列管理程式應用程式選擇性地啟用應用程式活動追蹤，可能會導致其他傳訊活動，以及在佇列管理程式中需要其他儲存體空間。在傳訊效能很重要的環境中，例如在高工作量應用程式中，或在服務水準合約 (SLA) 需要傳訊提供者的最短回應時間時，可能不適合收集應用程式活動追蹤，或可能需要調整所產生追蹤活動訊息的詳細資料或頻率。mqat.ini 檔案中 **ActivityInterval**、**ActivityCount** 及 **TraceLevel** 的預設值提供詳細資料與效能的預設平衡。不過，您可以調整這些值，以符合系統的精確功能及效能需求。

## 程序

- 只追蹤您需要的應用程式。

作法是在 `mqat.ini` 中建立 `ApplicationTrace` 應用程式特定段落，或變更應用程式以在 `MQCONN` 呼叫的 `MQCNO` 結構上的選項欄位中指定 `MQCNO_ACTIVITY_TRACE_ENABLED`。請參閱第 182 頁的『使用 `mqat.ini` 來配置活動追蹤行為』和第 181 頁的『設定 `MQCONN` 選項以控制活動追蹤資訊的收集』。

- 在啟動追蹤之前，請檢查至少有一個應用程式在執行中，且已準備好從 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE` 擷取活動追蹤訊息資料。
- 增加耗盡佇列的應用程式數目，使佇列深度儘可能低。
- 在 `mqat.ini` 檔案中設定 **TraceLevel** 值，以收集所需的資料量下限。

`TraceLevel=LOW` 對傳訊效能的影響最低。請參閱第 182 頁的『使用 `mqat.ini` 來配置活動追蹤行為』。

- 調整 `mqat.ini` 中的 **ActivityCount** 和 **ActivityInterval** 值，以調整產生活動追蹤訊息的頻率。如果您要追蹤多個應用程式，則產生活動追蹤訊息的速度可能比從 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE` 中移除活動追蹤訊息的速度更快。不過，當您減少產生活動追蹤訊息的頻率時，也會增加佇列管理程式所需的儲存體空間，以及將訊息寫入佇列時訊息的大小。

## 下一步

檢視應用程式活動追蹤訊息內容最簡單的方式是使用第 187 頁的『`amqsact` 範例程式』。

## `amqsact` 範例程式

`amqsact` 會為您格式化「應用程式活動追蹤」訊息，並隨 IBM MQ 提供。

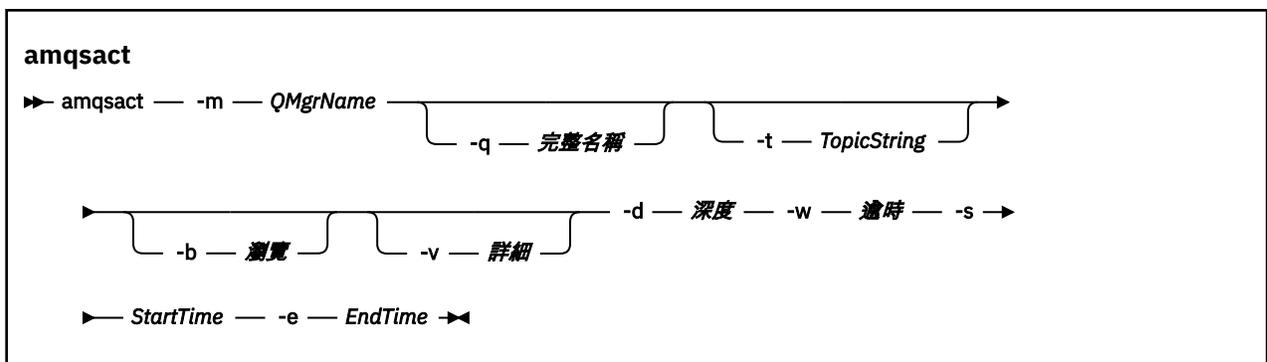
已編譯的程式位於 `samples` 目錄中：

-   在 Linux 和 UNIX `MQ_INSTALLATION_PATH/samp/bin` 上
-  在 Windows `MQ_INSTALLATION_PATH\tools\c\Samples\Bin` 上

## 顯示模式

依預設，顯示模式中的 `amqsact` 會處理 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`。您可以指定佇列名稱或主題字串來置換此行為。

您也可以控制顯示的追蹤期間，並指定在顯示之後移除或保留活動追蹤訊息。



## 顯示模式的必要參數

### **-m QMgrName**

佇列管理程式的名稱。

### **-d 深度**

要顯示的記錄數。

### **-w timeout**

等待時間 (以秒為單位)。如果在指定的期間內未出現任何追蹤訊息，則會結束 `amqsact`。

### **-s StartTime**

要處理之記錄的開始時間。

### **-e EndTime**

要處理的記錄結束時間。

## 顯示模式的選用參數

### **-q 完整名稱**

指定特定佇列以置換預設佇列名稱

### **-t TopicString**

訂閱事件主題

### **-b**

僅瀏覽記錄

### **-v**

詳細輸出

## 顯示模式的範例輸出

在 MQCONN API 呼叫中使用佇列管理程式 *TESTQM* 上的 **amqsact**，以及詳細輸出：

```
amqsact -m TESTQM -v
```

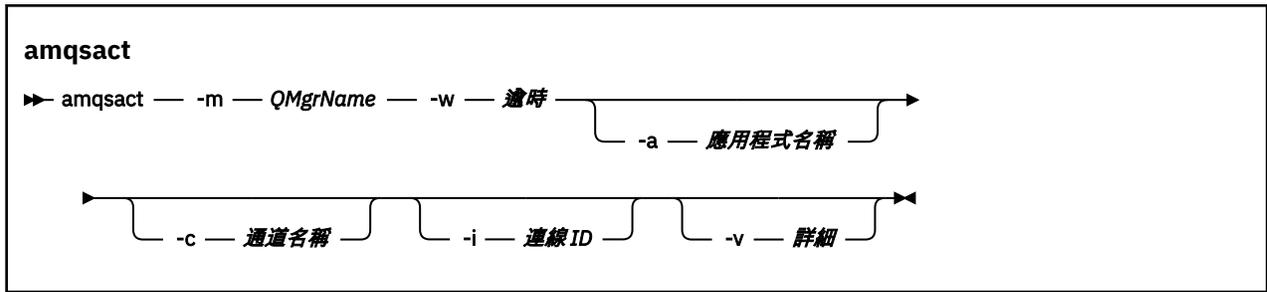
前述指令提供下列範例輸出：

```
MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: B5F6 4251 2000 E601
QueueManager: 'TESTQM'
Host Name: 'ADMINIB-1VTJ6N1'
IntervalStartDate: '2014-03-15'
IntervalStartTime: '12:08:10'
IntervalEndDate: '2014-03-15'
IntervalEndTime: '12:08:10'
CommandLevel: 750
SeqNumber: 0
ApplicationName: 'IBM MQ_1\bin\amqsput.exe'
Application Type: MQAT_WINDOWS_7
ApplicationPid: 14076
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: ''
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationTid: 1
OperationDate: '2014-03-15'
OperationTime: '12:08:10'
ConnectionId:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: FFFFFFFB5FFFFFFF6 4251 2000 FFFFFFFE601 '
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0
```

## 動態模式

您可以指定應用程式名稱、通道名稱或連線 ID 作為 **amqsact** 的引數來啟用動態模式。請注意，您可以在名稱中使用萬用字元。

在動態模式中，透過使用系統主題的不可延續訂閱，在取樣開始時啟用活動追蹤資料。當 **amqsact** 停止時，停止收集活動追蹤資料。您必須在動態模式中指定 **amqsact** 的逾時值。您可以同時執行多個 **amqsact** 副本，每一個實例都會接收任何活動追蹤資料的副本。



## 動態模式的必要參數

### **-m QMgrName**

佇列管理程式的名稱。

### **-w timeout**

等待時間 (以秒為單位)。如果在指定的期間內未出現任何追蹤訊息，則會結束 `amqsact`。

## 動態模式的選用參數

### **-a 應用程式名稱**

指定要收集其訊息的應用程式名稱

### **-c 通道名稱**

指定要收集其訊息的通道

### **-i 連線 ID**

指定要收集其訊息的連線。

### **-v**

詳細輸出

## 動態模式的輸出範例

下列指令會針對以文字 "amqs"開頭的應用程式所建立的任何連線，產生並顯示活動追蹤訊息。閒置 30 秒之後，`amqsact` 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

下列指令會針對 QMGR1.TO.QMGR2 通道。閒置 10 秒之後，`amqsact` 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

下列指令會針對具有 CONN "6B576B5420000701" 及 EXTCNN "414D5143514D47523120202020202020" 的現有 IBM MQ 連線上的任何活動，產生並顯示詳細活動追蹤訊息。在閒置一分鐘之後，`amqsact` 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

## 訂閱應用程式活動追蹤資訊

從 IBM MQ 9.0，您可以動態訂閱應用程式活動追蹤資訊，作為透過佇列管理程式層次配置來收集資訊的替代方案。

## 關於這項作業

應用程式活動追蹤會追蹤應用程式的行為，並提供應用程式在與 IBM MQ 資源互動時所使用參數的詳細視圖。它也會顯示應用程式所發出的 MQI 呼叫順序。

在 IBM MQ 9.0 之前，將活動追蹤 PCF 訊息寫入系統佇列 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE。如需相關資訊，請參閱第 180 頁的『配置應用程式活動追蹤資訊的中央集合』。

除了將追蹤資料寫入系統佇列之外，從 IBM MQ 9.0，IBM MQ 還可讓您動態訂閱寫入特殊 IBM MQ 系統主題的活動追蹤資料，而不是透過佇列管理程式層次配置來收集資訊。

請注意，IBM MQ 9.0 不會針對此目的使用結束程式。如果您先前已使用結束程式來追蹤應用程式活動，則必須切換至使用新方法來收集應用程式活動追蹤。

建立訂閱會啟用活動追蹤。對於集中收集追蹤資料，您不需要設定佇列管理程式或應用程式屬性。不過，透過在佇列管理程式或應用程式層次停用追蹤來明確封鎖活動追蹤，也會阻止將活動追蹤遞送至任何相符訂閱。

## 程序

- [第 190 頁的『應用程式活動追蹤的訂閱』](#)
- [第 190 頁的『建立應用程式活動追蹤的訂閱』](#)
- [第 191 頁的『使用 amqsact 來檢視追蹤訊息』](#)
- [第 193 頁的『使用 mqat.ini 來配置追蹤層次』](#)

## 應用程式活動追蹤的訂閱

您可以訂閱 IBM MQ 系統主題，以收集應用程式活動追蹤資訊。

您可以訂閱代表要追蹤之活動的特殊 IBM MQ 系統主題字串。訂閱會自動產生活動追蹤資料訊息，並將它們發佈至訂閱目的地佇列。如果您刪除訂閱，則會停止產生該訂閱的活動追蹤資料。

訂閱可以追蹤下列其中一項資源的活動：

- 指定的應用程式
- 指定的 IBM MQ 通道
- 現有的 IBM MQ 連線

您可以建立多個具有不同或相同主題字串的訂閱。當您使用相同的系統活動追蹤主題字串來建立多個訂閱時，每一個訂閱都會收到活動追蹤資料的副本，這可能會對效能產生負面影響。

啟用任何層次的活動追蹤可能會對效能造成負面影響。訂閱越多，或訂閱的資源越多，潛在的效能額外負擔就越大。為了將收集活動追蹤的額外負擔減至最少，資料會寫入訊息，並從應用程式活動本身非同步遞送至訂閱。通常會將多個作業寫入單一活動追蹤資料訊息。非同步作業可以在應用程式作業與接收記錄作業的追蹤資料之間產生延遲。

## 建立應用程式活動追蹤的訂閱

您可以建立特定主題的訂閱，以收集 IBM MQ 9.0 上的應用程式活動追蹤資料。

針對特定系統主題字串建立訂閱時，會自動將適當的活動追蹤 PCF 資料訊息發佈至該訂閱。如需訂閱主題的詳細資訊，請參閱 [發佈/訂閱傳訊](#)。

主題字串的格式如下：

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifier
```

其中：

- *qmgr\_name* 指定所追蹤應用程式連接的佇列管理程式。*qmgr\_name* is the name of the queue manager with all trailing blank characters removed and any forward slash (/) characters replaced by an ampersand (&) character.
- *resource\_type* 指定正在收集其資源資料的類型，它是下列其中一個字串：
  - *AppName* 以指定應用程式。要求會訂閱應用程式名稱符合 *resource\_identifier* 指定的所有 IBM MQ 連線。
  - *ChannelName*，以指定 IBM MQ 通道。
  - *ConnectionId* 以指定 IBM MQ 連線。

- *resource\_identifier* 會識別實際資源。格式視資源類型而定：
  - 對於資源類型 *ApplName*，*resource\_identifier* 是佇列管理程式所看到應用程式名稱的尾端部分 (在最後一個/或\之後的值)，並移除任何尾端空白字元。此值符合 API 結束程式環境定義結構 (MQAXC) 中的 *ApplName* 值。當您使用 MQSC 指令 **DISPLAY CONN** 時，連線的 *ApplName* 會以 APPLTAG 值傳回。
  - 對於資源類型 *ChannelName*，*resource\_identifier* 是要追蹤的通道名稱。如果通道名稱識別 SVRCONN 通道，則會追蹤已連接用戶端的所有應用程式活動。如果通道名稱識別佇列管理程式至佇列管理程式通道，則會追蹤送入及送出訊息。The *resource\_identifier* is the channel name with all trailing blank characters removed and any '/' characters replaced by a '&' character.
  - 若資源類型為 *ConnectionId*，*resource\_identifier* 是指派給每一個連線的唯一連線 ID。主題字串中的連線 ID 是以十六進位字串寫入的完整 24 位元組值。此值是 EXTCONN 後接 MQSC 指令 **DISPLAY CONN** 所傳回的 CONN 值的連結。

您可以在 *resource\_identifier* 中使用萬用字元，以符合單一訂閱中的多個資源身分。萬用字元可以是預設主題樣式 ('#' 或 '+') 或字元樣式 ('\*' 或 '?')。當您使用主題樣式萬用字元時，它無法與部分資源名稱結合，只能用來符合所有可能的應用程式、通道或連線。使用任何萬用字元會增加所產生追蹤資料的層次，這可能會影響效能。

如果要訂閱這些主題字串，您必須具備 "訂閱" 授權。系統主題不會從佇列管理程式主題樹狀結構的根目錄繼承授權。使用者必須獲得主題樹狀結構中 \$SYS/MQ 點或該點以上的受管理主題物件的存取權。如果您具有 SYSTEM.ADMIN.TOPIC，雖然這會授與對所有 \$SYS/MQ 主題字串的存取權，而不只是活動追蹤。如果要更具體地控制存取權，可以針對樹狀結構中的更深層點定義新的受管理主題物件，例如針對所有活動追蹤或特定應用程式名稱或通道名稱。

## 範例

下列範例顯示在 Windows 系統上執行且名為 amqsput 之應用程式的主題字串：

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ApplName/amqsputc.exe
```

下列範例顯示通道的主題字串：

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

下列範例顯示連線的主題字串：

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ConnectionId/414D5143514D4752312020202020206B576B5420000701
```

下列範例顯示建立訂閱的主題字串，以追蹤佇列管理程式 QMGR1: 上所有通道的資料：

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

下列範例顯示建立訂閱以追蹤名稱以 "amqs" 開頭之應用程式資料的主題字串 (請注意，若要使用 "\*" 萬用字元，必須使用字元萬用字元模型來建立訂閱)：

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ApplName/amqs*
```

## 相關概念

第 261 頁的『監視及活動追蹤的系統主題』

佇列管理程式主題樹狀結構中的系統主題用於資源監視 (其中部分類似於統計資料訊息的內容)，並作為耗用應用程式活動追蹤的方式。

## 使用 *amqsact* 來檢視追蹤訊息

您可以將 *amqsact* 程式與 IBM MQ 9.0 搭配使用，以產生及檢視追蹤訊息。

**amqsact** 程式是一個 IBM MQ 範例。若要使用此範例，您必須使用用戶端連接的執行檔 **amqsactc**。執行檔位於 `samples` 目錄中：

- 在 Linux 和 UNIX 平台上：`MQ_INSTALLATION_PATH/samp/bin64`
- 在 Windows 平台上：`MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`

您可以透過兩種方式來使用 **amqsact**：

### 顯示模式

格式化及顯示遞送至 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`。

### 動態模式

建立一組資源的訂閱，並執行 **amqsact** 來顯示產生的活動追蹤。

## 顯示模式

依預設，顯示模式中的 **amqsact** 會處理 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`。您可以指定佇列名稱或主題字串來置換此行為。必須使用 `收集應用程式活動追蹤資訊` 中說明的其中一個方法來啟用活動追蹤。您可以控制顯示的追蹤期間，並指定在顯示之後是否移除或保留活動追蹤訊息。在顯示模式中，**amqsact** 採用下列引數：

#### **-m queue\_manager\_name**

必要項目。指定要收集其追蹤訊息的佇列管理程式。

#### **-q 佇列名稱**

只顯示與指名佇列相關的追蹤訊息。

#### **-t topic\_string**

只顯示與具名主題相關的追蹤訊息。

#### **-b**

指定在顯示之後保留追蹤訊息。

#### **-v**

以詳細模式顯示追蹤訊息。

#### **-d 深度**

要顯示的訊息數目。

#### **-w timeout**

指定逾時。如果在該期間未出現任何追蹤訊息，則會結束 **amqsact**。

#### **-s start\_time**

將此引數與 `-e` 引數一起使用，以指定時段。會顯示指定時段的追蹤訊息。

#### **-e end\_time**

搭配使用此引數與 `-s` 引數，以指定時段。會顯示指定時段的追蹤訊息。

例如，下列指令會顯示 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`，並在顯示之後刪除訊息：

```
amqsact -m QMGR1
```

下列指令會顯示指定佇列 `SUB.QUEUE`，並在顯示之後刪除訊息。訊息會繼續顯示，直到 30 秒期間沒有新的訊息。例如，此指令可以與活動追蹤系統主題字串的訂閱搭配使用。

```
amqsact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

下列指令會以詳細格式顯示目前在 `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`。顯示之後，訊息仍會留在佇列中。

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

## 動態模式

您可以透過指定應用程式名稱、通道名稱或連線 ID 作為 **amqsact** 的引數來啟用動態模式。您可以在名稱中使用萬用字元。在動態模式中，透過使用系統主題的不可延續訂閱，在取樣開始時啟用活動追蹤資料。當 **amqsact** 停止時，停止收集活動追蹤資料。您必須在動態模式中指定 **amqsact** 的逾時值。您可以同時執行多個 **amqsact** 副本，且每一個實例都會收到任何活動追蹤資料的副本。在動態模式中，**amqsact** 會採用下列引數：

**-m queue\_manager\_name**

必要項目。指定要收集其追蹤訊息的佇列管理程式。

**-w timeout**

必要項目。指定逾時。如果在該期間未出現任何追蹤訊息，則會結束 **amqsact**。

**-a application\_name**

指定要收集訊息的應用程式。

**-c channel\_name**

指定要收集其訊息的通道。

**-i connection\_id**

指定要收集其訊息的連線。

**-v**

以詳細模式顯示追蹤訊息。

例如，下列指令會針對名為 "amqsget.exe" 的應用程式所建立的任何連線產生並顯示活動追蹤訊息。閒置 30 秒之後，**amqsact** 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

下列指令會針對以文字 "amqs" 開頭的應用程式所建立的任何連線，產生並顯示活動追蹤訊息。閒置 30 秒之後，**amqsact** 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

下列指令會針對 QMGR1.TO.QMGR2 通道。閒置 10 秒之後，**amqsact** 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

下列指令會產生並顯示任何通道上任何活動的活動追蹤訊息。閒置 10 秒之後，**amqsact** 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 10 -c #
```

下列指令會針對具有 CONN "6B576B5420000701" 及 EXTCONN "414D5143514D47523120202020202020" 的現有 IBM MQ 連線上的任何活動，產生並顯示詳細活動追蹤訊息。在閒置一分鐘之後，**amqsact** 程式會結束，且不會產生新的活動追蹤資料。

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

## 使用 *mqat.ini* 來配置追蹤層次

您可以透過設定 *mqat.ini* 配置檔的 AllActivity 追蹤段落值，來配置佇列管理程式的追蹤層次。

您可以設定 AllActivity 追蹤段落的下列值：

### ActivityInterval

追蹤訊息之間的時間間隔 (以秒為單位)。活動追蹤不使用計時器執行緒，因此追蹤訊息不會在經歷時間的確切時刻寫入，而是在經歷時間間隔之後執行第一個 MQI 作業時寫入。如果此值為 0，則會在連線中斷時 (或達到活動計數時) 寫入追蹤訊息。預設值為 1。

### ActivityCount

追蹤訊息之間的 MQI 作業數。如果此值為 0，則會在連線中斷 (或活動間隔過了) 時寫入追蹤訊息。預設值為 100。

### TraceLevel

針對每一個作業追蹤的參數詳細資料數量。個別作業的說明詳細說明針對每一個追蹤層次包括哪些參數。設為 LOW、MEDIUM 或 HIGH。預設為 MEDIUM。

### TraceMessage 資料

MQGET、MQPUT、MQPUT1 及回呼作業所追蹤的訊息資料量 (以位元組為單位)。預設為 0。

### StopOnGetTrace 訊息

可以設為 ON 或 OFF。預設為 ON。

### SubscriptionDelivery

可以設為 BATCHED 或 IMMEDIATE。決定當有一或多個活動追蹤訂閱存在時，是否要使用 ActivityInterval 和 ActivityCount 參數。將此參數設為 IMMEDIATE 會導致在追蹤資料具有相符訂閱時，將 ActivityInterval 及 ActivityCount 值置換為有效值 1。每一個活動追蹤記錄不會與來自相同連線的其他記錄一起批次處理，而是立即遞送至訂閱而沒有延遲。IMMEDIATE 設定會增加收集活動追蹤資料的效能額外負擔。預設值是 BATCHED。

## 應用程式活動追蹤訊息參照

請利用這個頁面來取得應用程式活動追蹤訊息的格式概觀，以及這些訊息中所傳回的資訊

應用程式活動追蹤訊息是包含訊息描述子及訊息資料的標準 IBM MQ 訊息。訊息資料包含 IBM MQ 應用程式所執行 MQI 作業的相關資訊，或 IBM MQ 系統中所發生活動的相關資訊。

### 訊息描述子

- MQMD 結構

### 訊息資料

- PCF 標頭 (MQCFH)
- 一律傳回的應用程式活動追蹤訊息資料
- 作業特定的應用程式活動追蹤訊息資料

### 應用程式活動追蹤訊息 MQMD (訊息描述子)

請利用這個頁面來瞭解應用程式活動追蹤訊息的訊息描述子與事件訊息的訊息描述子之間的差異

應用程式活動追蹤訊息的訊息描述子中的參數和值與事件訊息的訊息描述子中的參數和值相同，但有下列異常狀況：

#### Format

說明： 訊息資料的格式名稱。

值： **MQFMT\_ADMIN**  
管理訊息。

#### CorrelId

說明： 相關性 ID。

值： 以應用程式的 ConnectionId 起始設定

### MQCFH (PCF 標頭)

請利用這個頁面來檢視活動追蹤訊息的 MQCFH 結構所包含的 PCF 值

對於活動追蹤訊息，MQCFH 結構包含下列值：

#### Type

說明： 識別訊息內容的結構類型。

資料類型: MQLONG。  
值: MQCFT\_APP\_ACTIVITY

### **StrucLength**

說明: MQCFH 結構的長度 (以位元組為單位)。  
資料類型: MQLONG。  
值: MQCFH\_STRUC\_LENGTH

### **Version**

說明: 結構版本號碼。  
資料類型: MQLONG。  
值: MQCFH\_VERSION\_3

### **Command**

說明: 指令 ID。此欄位識別訊息的種類。  
資料類型: MQLONG。  
值: MQCMD\_ACTIVITY\_TRACE

### **MsgSeqNumber**

說明: 訊息序號。此欄位是相關訊息群組內的訊息序號。  
資料類型: MQLONG。  
值: 1

### **Control**

說明: 控制選項。  
資料類型: MQLONG。  
值: MQCFC\_LAST。

### **CompCode**

說明: 完成碼。  
資料類型: MQLONG。  
值: MQCC\_OK。

### **Reason**

說明: 原因碼定義完成碼。  
資料類型: MQLONG。  
值: MQRC\_NONE。

### **ParameterCount**

說明: 參數結構計數。此欄位是遵循 MQCFH 結構的參數結構數目。群組結構 (MQCFGR) 及其包含的參數結構只會計為一個結構。  
資料類型: MQLONG。  
值: 1 或以上

## 應用程式活動追蹤訊息資料

緊接在 PCF 標頭之後是一組參數，說明活動追蹤的時間間隔。這些參數也指出在寫入訊息時的訊息順序。不保證標頭後面的欄位順序及數目，容許在未來新增其他資訊。

訊息名稱：                  活動追蹤訊息。

系統佇列：                  SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

---

### **QueueManager**

說明：                      佇列管理程式的名稱。  
ID：                          MQCA\_Q\_MGR\_NAME  
資料類型：                  MQCFST  
長度上限：                  MQ\_Q\_MGR\_NAME\_LENGTH

### **QSGName**



說明：                      佇列管理程式所屬的佇列共用群組名稱 (僅限 z/OS)。  
ID：                          MQCA\_QSG\_NAME  
資料類型：                  MQCFST  
長度上限：                  MQ\_Q\_MGR\_NAME\_LENGTH

### **HostName**

說明：                      佇列管理程式執行所在之機器的主機名稱。  
ID：                          MQCACF\_HOST\_NAME  
資料類型：                  MQCFST

### **IntervalStartDate**

說明：                      監視期間開始的日期。  
ID：                          MQCAMO\_START\_DATE  
資料類型：                  MQCFST  
長度上限：                  MQ 資料長度

### **IntervalStartTime**

說明：                      監視期間開始的時間。  
ID：                          MQCAMO\_START\_TIME  
資料類型：                  MQCFST  
長度上限：                  MQ 時間長度

### **IntervalEndDate**

說明：                      監視期間結束的日期。  
ID：                          MQCAMO\_END\_DATE  
資料類型：                  MQCFST  
長度上限：                  MQ 資料長度

### ***IntervalEndTime***

說明： 監視期間結束的時間。  
ID: MQCAOM\_END\_TIME  
資料類型: MQCFST  
長度上限: MQ 時間長度

### ***CommandLevel***

說明： IBM MQ 指令層次。  
ID: MQIA\_COMMAND\_LEVEL  
資料類型: MQCFIN

### ***SeqNumber***

說明： 序號通常為零。對於長時間執行連線的每一筆後續記錄，此值會增加。  
ID: MQIACF\_SEQUENCE\_NUMBER  
資料類型: MQCFIN

### ***ApplicationName***

說明： 應用程式的名稱 (程式名稱)。  
ID: MQCACF\_APPL\_NAME  
資料類型: MQCFST  
長度上限: MQ\_APPL\_NAME\_LENGTH

### ***ApplClass***

說明： 執行活動的應用程式類型。可能的值 :MQAT\_\*  
ID: MQIA\_APPL\_TYPE  
資料類型: MQCFIN

### ***ApplicationPid***

說明： 應用程式的作業系統處理程序 ID。  
ID: MQIACF\_PROCESS\_ID  
資料類型: MQCFIN

### ***UserId***

說明： 應用程式的使用者 ID 環境定義。  
ID: MQCACF\_USER\_IDENTIFIER  
資料類型: MQCFST  
長度上限: MQ\_USER\_ID\_LENGTH

### ***APICallerType***

說明： 應用程式的類型。可能的值 :MQXACT\_EXTERNAL 或 MQXACT\_INTERNAL。  
ID: MQIACF\_API\_CALLER\_TYPE  
資料類型: MQCFIN

### **Environment**

說明： 應用程式的執行時期環境。可能的值 :MQXE\_\*  
ID: MQIACF\_API\_環境  
資料類型: MQCFIN

### **ChannelName**

說明： 與連線相關聯的通道名稱。只有在「環境」參數的值為 MQXE\_MCA 或 MQXE\_MCA\_SVRCONN 時，才會傳回此參數。  
ID: MQ 快取\_通道名稱  
資料類型: MQCFST  
長度上限: MQ\_CHANNEL\_NAME\_LENGTH

### **ConnectionName**

說明： 與連線相關聯的網路連線名稱。只有在「環境」參數的值為 MQXE\_MCA 或 MQXE\_MCA\_SVRCONN 時，才會傳回此參數。  
ID: MQ 快取\_連線名稱  
資料類型: MQCFST  
長度上限: MQ\_CONN\_NAME\_LENGTH

### **ChannelType**

說明： 與連線相關聯的通道類型。只有在「環境」參數的值為 MQXE\_MCA 或 MQXE\_MCA\_SVRCONN 時，才會傳回此參數。可能的值 :MQCHT\_\*  
ID: MQIACH\_CHANNEL\_TYPE  
資料類型: MQCFIN

### **V 9.1.0 RemoteProduct**

說明： 與連線相關聯的遠端產品 ID。只有在「環境」參數的值為 MQXE\_MCA 或 MQXE\_MCA\_SVRCONN 時，才會傳回此參數。  
ID: MQ 快取\_遠端產品  
資料類型: MQCFST  
長度上限: MQ 遠端產品長度

### **V 9.1.0 RemoteVersion**

說明： 與連線相關聯的遠端產品版本。只有在「環境」參數的值為 MQXE\_MCA 或 MQXE\_MCA\_SVRCONN 時，才會傳回此參數。  
ID: MQ 快取\_遠端版本  
資料類型: MQCFST  
長度上限: MQ 遠端版本長度

### **FunctionName**

說明： 起始執行緒所起始的最後一個高階函數名稱。  
ID: MQCACF\_APPL\_FUNCTION

資料類型: MQCFST

### **FunctionType**

說明: 起始執行緒所起始的最後一個高階函數的類型。可能的值 :MQFUN\_\*  
ID: MQIACF\_APPL\_FUNCTION\_TYPE  
資料類型: MQCFIN

### **Detail**

說明: 針對連線所記錄的詳細層次。可能的值: 1=LOW 2=MEDIUM 3=HIGH  
ID: MQIACF\_TRACE\_DETAIL  
資料類型: MQCFIN

### **TraceDataLength**

說明: 針對此連線追蹤的訊息資料長度 (以位元組為單位)。  
ID: MQIACF\_TRACE\_DATA\_LENGTH  
資料類型: MQCFIN

### **PointerSize**

說明: 應用程式正在執行之平台上的指標長度 (以位元組為單位) (以協助解譯二進位結構)。  
ID: MQIACF\_POINTER\_SIZE  
資料類型: MQCFIN

### **Platform**

說明: 佇列管理程式執行所在的平台。可能的值 :MQPL\_\*  
ID: MQIA\_PLATFORM  
資料類型: MQCFIN

## **應用程式活動 MQI 作業的變數參數**

應用程式活動資料 MQCFGR 結構後面接著一組 PCF 參數，這些參數對應於正在執行的作業。每一個作業的參數定義在下一節中。

追蹤層次指出要併入追蹤中的參數所需的追蹤精度層次。可能的追蹤層次值為:

#### **1. 低**

當針對應用程式配置 "low"、"medium" 或 "high" activity tracing 時，會包含這個參數。此設定表示參數一律包含在作業的 AppCompatActivityData 群組中。這組參數足以追蹤應用程式所建立的 MQI 呼叫，並查看它們是否成功。

#### **2. 中**

只有在為應用程式配置 "中" 或 "高" 活動追蹤時，此參數才會包含在作業的 AppCompatActivityData 群組中。這組參數會新增資源的相關資訊，例如應用程式所使用的佇列及主題名稱。

#### **3. 高**

只有在針對應用程式配置 "高" 活動追蹤時，才會將此參數包含在作業的 AppCompatActivityData 群組中。這組參數包括傳遞至 MQI 及 XA 函數之結構的記憶體傾出。基於此原因，它包含 MQI 及 XA 呼叫中所使用參數的相關資訊。結構記憶體傾出是結構的淺層副本。為了避免錯誤嘗試取消參照指標，結構中的指標值會設為 NULL。

**註:** 傾出的結構版本不一定與應用程式使用的版本相同。可以透過 API 交叉結束程式、活動追蹤程式碼或佇列管理程式來修改結構。佇列管理程式可以將結構修改為更新版本，但佇列管理程式絕不會將它變更為舊版結構。如果這樣做，將有遺失資料的風險。

### *MQBACK*

應用程式已啟動 MQBACK MQI 函數

#### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型	MQCFIN

#### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型	MQCFIN

### **QMGrOpDuration**

說明: 佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。

**註:** 此計時器的精確度會根據您企業使用的平台而有所不同。

PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### *MQBEGIN*

應用程式已啟動 MQBEGIN MQI 函數

#### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型	MQCFIN

#### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型	MQCFIN

## **MQBO**

說明:	MQBEGIN 選項結構。如果在 MQBEGIN 呼叫上使用 NULL 指標，則不包括此參數。
PCF 參數:	MQBACF_MQBO_STRUCT
追蹤層次:	3
類型	MQCFBS
長度:	MQBO 結構的長度 (以位元組為單位)。

## **QMgrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

## **MQcallback**

應用程式已啟動 MQCALLBACK 函數

## **ObjectHandle**

說明:	物件控點
PCF 參數:	MQIACF_HOBJ
追蹤層次:	1
類型	MQCFIN

## **CallType**

說明:	呼叫函數的原因。其中一個 MQCBCT_* 值
PCF 參數:	MQIACF_CALL_TYPE
追蹤層次:	1
類型	MQCFIN

## **MsgBuffer**

說明:	訊息資料。
PCF 參數:	MQBACF_MESSAGE_DATA
追蹤層次:	1
類型	MQCFBS
長度:	長度由 APPTTRACE 配置中設定的 TRACEDATA () 參數所控管。如果 TRACEDATA=NONE，則會省略此參數。

## **MsgLength**

說明:	訊息的長度。(取自 MQCBC 結構中的 DataLength 欄位)。
PCF 參數:	MQIACF_MSG_LENGTH

追蹤層次: 1  
類型 MQCFIN

### **HighResTime**

說明: 自 1970 年 1st (UTC) 午夜以來的作業時間 (微秒)  
註: 此計時器的精確度根據高解析度計時器的平台支援而有所不同

PCF 參數: MQIAMO64\_HIGHRES\_TIME  
追蹤層次: 2  
類型 MQCFIN64

### **ReportOptions**

說明: 報告訊息的選項  
PCF 參數: MQIACF\_REPORT  
追蹤層次: 2  
類型 MQCFIN

### **MsgType**

說明: 訊息類型  
PCF 參數: MQIACF\_MSG\_TYPE  
追蹤層次: 2  
類型 MQCFIN

### **Expiry**

說明: 訊息期限  
PCF 參數: MQIACF\_EXPIRY  
追蹤層次: 2  
類型 MQCFIN

### **Format**

說明: 訊息資料的格式名稱  
PCF 參數: MQCACH\_FORMAT\_NAME  
追蹤層次: 2  
類型 MQCFST  
長度: MQ 格式長度

### **Priority**

說明: 訊息優先順序  
PCF 參數: MQIACF\_PRIORITY  
追蹤層次: 2  
類型 MQCFIN

### **Persistence**

說明:	訊息持續性
PCF 參數:	MQIACF_持續性
追蹤層次:	2
類型	MQCFIN

### **MsgId**

說明:	訊息 ID
PCF 參數:	MQBACF_MSG_ID
追蹤層次:	2
類型	MQCFBS
長度:	MQ_MSG_ID_LENGTH

### **CorrelId**

說明:	相關性 ID
PCF 參數:	MQBACF_CORREL_ID
追蹤層次:	2
類型	MQCFBS
長度:	MQ_CORREL_ID_LENGTH

### **ObjectName**

說明:	已開啟物件的名稱。
PCF 參數:	MQCACF_OBJECT_NAME
追蹤層次:	2
類型	MQCFST
長度:	MQ_Q_NAME_LENGTH

### **ResolvedQName**

說明:	從中擷取訊息之佇列的本端名稱。
PCF 參數:	MQCACF_RESOLVED_Q_NAME
追蹤層次:	2
類型	MQCFST
長度:	MQ_Q_NAME_LENGTH

### **ReplyToQueue**

說明:	MQ_Q_NAME_LENGTH
PCF 參數:	MQCACF_REPLY_TO_Q
追蹤層次:	2
類型	MQCFST

### **ReplyToQMgr**

說明:	MQ_Q_MGR_NAME_LENGTH
PCF 參數:	MQCACF_REPLY_TO_Q_MGR
追蹤層次:	2
類型	MQCFST

### **CodedCharSetId**

說明:	訊息資料的字集 ID
PCF 參數:	MQIA_CODED_CHAR_SET_ID
追蹤層次:	2
類型	MQCFIN

### **Encoding**

說明:	訊息資料的數值編碼。
PCF 參數:	MQIACF_ENCODING
追蹤層次:	2
類型	MQCFIN

### **PutDate**

說明:	MQ_PUT_DATE_LENGTH
PCF 參數:	MQCACF_PUT_DATE
追蹤層次:	2
類型	MQCFST

### **PutTime**

說明:	MQ_PUT_TIME_LENGTH
PCF 參數:	MQCACF_PUT_TIME
追蹤層次:	2
類型	MQCFST

### **ResolvedQName**

說明:	當 ResolvedType 為 MQOT_Q 時, ObjectHandle 所參照的佇列名稱。
PCF 參數:	MQCACF_RESOLVED_LOCAL_q_name
追蹤層次:	2
類型	MQCFST
長度:	MQ_Q_NAME_LENGTH。

### **ResObjectString**

說明:	當 ResolvedType 是 MQOT_TOPIC 時, ObjectHandle 所參照的物件名稱。
PCF 參數:	MQCACF_RESOLVED_OBJECT_STRING
追蹤層次:	2

類型 MQCFST  
長度： 長度會改變。

### ***ResolvedType***

說明： ObjectHandle 所參照的物件類型。可能的值為 MQOT\_Q、MQOT\_TOPIC 或 MQOT\_NONE。  
PCF 參數： MQIACF\_RESOLVED\_TYPE  
追蹤層次： 2  
類型 MQCFIN

### ***PolicyName***

說明： 套用至此訊息的原則名稱。  
註： 僅限 AMS 受保護訊息  
PCF 參數： MQCA\_POLICY\_NAME  
追蹤層次： 2  
類型 MQCFST  
長度： MQ\_OBJECT\_NAME\_LENGTH

### ***XmitqMsgId***

說明： 傳輸佇列標頭中訊息的訊息 ID。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時  
PCF 參數： MQBACF\_XQH\_MSG\_ID  
追蹤層次： 2  
類型 MQCFBS  
長度： MQ\_MSG\_ID\_LENGTH

### ***XmitqCorrelId***

說明： 傳輸佇列標頭中訊息的相關性 ID。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時  
PCF 參數： MQBACF\_XQH\_CORREL\_ID  
追蹤層次： 2  
類型 MQCFBS  
長度： MQ\_CORREL\_ID\_LENGTH

### ***XmitqPutTime***

說明： 訊息在傳輸佇列標頭中的放置時間。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時  
PCF 參數： MQCACF\_XQH\_PUT\_TIME  
追蹤層次： 2  
類型 MQCFST

長度: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

說明: 訊息在傳輸佇列標頭中的放置日期。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_PUT\_DATE

追蹤層次: 2

類型: MQCFST

長度: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

說明: 傳輸佇列標頭中訊息的遠端佇列目的地。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_REMOTE\_Q\_Name

追蹤層次: 2

類型: MQCFST

長度: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMgr***

說明: 傳輸佇列標頭中訊息的訊息 ID。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_REMOTE\_Q\_MGR

追蹤層次: 2

類型: MQCFST

長度: MQ\_MSG\_ID\_LENGTH

### ***MsgDescStructure***

說明: MQMD 結構。如果使用第 4 版 MQGMO 來要求傳回「訊息控點」而非 MQMD，則會省略此參數

PCF 參數: MQBACF\_MQMD\_STRUCT

追蹤層次: 3

類型: MQCFBS

長度: MQMD 結構的長度 (以位元組為單位) (實際大小取決於結構版本)

### ***GetMsgOptsStructure***

說明: MQGMO 結構。

PCF 參數: MQBACF\_MQGMO\_STRUCT

追蹤層次: 3

類型: MQCFBS

長度: MQGMO 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **MQCBCContextStructure**

說明:	MQCBC 結構。
PCF 參數:	MQBACF_MQCBC_STRUCT
追蹤層次:	3
類型	MQCFBS
長度:	MQCBC 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### **MQCB**

應用程式已啟動管理回呼 MQI 函數

### **CallbackOperation**

說明:	管理回呼函數作業。設為其中一個 MQOP_* 值
PCF 參數:	MQIACF_MQCB 作業
追蹤層次:	1
類型	MQCFIN

### **CallbackType**

說明:	回呼函數的類型 (MQCBD 結構中的 CallbackType 欄位)。設為其中一個 MQCBT_* 值
PCF 參數:	MQIACF_MQCB_TYPE
追蹤層次:	1
類型	MQCFIN

### **CallbackOptions**

說明:	回呼選項。設為其中一個 MQCBDO_* 值
PCF 參數:	MQIACF_MQCB_OPTIONS
追蹤層次:	1
類型	MQCFIN

### **CallbackFunction**

說明:	回呼函數 (如果啟動為函數呼叫) 的指標。
PCF 參數:	MQBACF_MQCB_FUNCTION
追蹤層次:	1
類型	MQCFBS

長度: MQPTR 大小

### **CallbackName**

說明: 作為動態鏈結程式啟動時的回呼函數名稱。

PCF 參數: MQCACF\_MQCB\_NAME

追蹤層次: 1

類型: MQCFST

長度: MQCHAR128 的大小

### **ObjectHandle**

說明: 物件控點

PCF 參數: MQIACF\_HOBJ

追蹤層次: 1

類型: MQCFIN

### **MaxMsgLength**

說明: 訊息長度上限。設為整數，或特殊值 MQCBD\_FULL\_MSG\_LENGTH

PCF 參數: MQIACH\_MAX\_MSG\_LENGTH

追蹤層次: 2

類型: MQCFIN

### **CompCode**

說明: 指出作業結果的完成碼

PCF 參數: MQIACF\_COMP\_CODE

追蹤層次: 1

類型: MQCFIN

### **Reason**

說明: 作業的原因碼結果

PCF 參數: MQIACF\_REASON\_CODE

追蹤層次: 1

類型: MQCFIN

### **ResolvedQName**

說明: 當 ResolvedType 為 MQOT\_Q 時， ObjectHandle 所參照的佇列名稱。

PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME

追蹤層次: 2

類型: MQCFST

長度: MQ\_Q\_NAME\_LENGTH。

### **ResObjectString**

說明: 當 ResolvedType 是 MQOT\_TOPIC 時， ObjectHandle 所參照的物件名稱。

PCF 參數: MQCACF\_RESOLVED\_OBJECT\_STRING  
追蹤層次: 2  
類型: MQCFST  
長度: 長度會改變。

### **ResolvedType**

說明: ObjectHandle 所參照的物件類型。可能的值為 MQOT\_Q、MQOT\_TOPIC 或 MQOT\_NONE。  
PCF 參數: MQIACF\_RESOLVED\_TYPE  
追蹤層次: 2  
類型: MQCFIN

### **Callback DescriptorStructure**

說明: MQCBD 結構。如果將 NULL MQCBC 值傳遞至 MQCB 呼叫，則會省略此參數。  
PCF 參數: MQBACF\_MQCBD\_STRUCT  
追蹤層次: 3  
類型: MQCFBS  
長度: MQCBC 結構的長度 (以位元組為單位)

### **MsgDescStructure**

說明: MQMD 結構。如果將 NULL MQMD 值傳遞至 MQCB 呼叫，則會省略 MsgDescStructure 參數。  
PCF 參數: MQBACF\_MQMD\_STRUCT  
追蹤層次: 3  
類型: MQCFBS  
長度: MQMD 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **GetMsgOptsStructure**

說明: MQGMO 結構。如果將 NULL MQGMO 值傳遞至 MQCB 呼叫，則會省略此參數。  
PCF 參數: MQBACF\_MQGMO\_STRUCT  
追蹤層次: 3  
類型: MQCFBS  
長度: MQGMO 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **QMgrOpDuration**

說明: 佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。  
**註:** 此計時器的精確度會根據您企業使用的平台而有所不同。  
PCF 參數: MQIAMO64\_QMGR\_OP\_DURATION  
追蹤層次: 2  
類型: MQCFIN64

## *MQCLOSE*

應用程式已啟動 MQCLOSE MQI 函數

### **ObjectHandle**

說明:	物件控點
PCF 參數:	MQIACF_HOBJ
追蹤層次:	1
類型	MQCFIN

### **CloseOptions**

說明:	關閉選項
PCF 參數:	MQIACF_CLOSE_OPTIONS
追蹤層次:	1
類型	MQCFIN

### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型	MQCFIN

### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型	MQCFIN

### **ResolvedQName**

說明:	當 ResolvedType 為 MQOT_Q 時, ObjectHandle 所參照的佇列名稱。
PCF 參數:	MQCACF_RESOLVED_LOCAL_Q_NAME
追蹤層次:	2
類型	MQCFST
長度:	MQ_Q_NAME_LENGTH。

### **ResObjectString**

說明:	當 ResolvedType 是 MQOT_TOPIC 時, ObjectHandle 所參照的物件名稱。
PCF 參數:	MQCACF_RESOLVED_OBJECT_STRING
追蹤層次:	2
類型	MQCFST
長度:	長度會改變。

### **ResolvedType**

說明:	ObjectHandle 所參照的物件類型。可能的值為 MQOT_Q、MQOT_TOPIC 或 MQOT_NONE。
PCF 參數:	MQIACF_RESOLVED_TYPE
追蹤層次:	2
類型	MQCFIN

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### **MQCMIT**

應用程式已啟動 MQCMIT MQI 函數

### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型	MQCFIN

### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型	MQCFIN

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### **MQCONN 和 MQCONNX**

應用程式已啟動 MQCONN 或 MQCONNX MQI 函數

### **ConnectionId**

說明:	連線 ID (如果可用) 或 MQCONNID_NONE (如果無法使用)
PCF 參數:	MQBACF_CONNECTION_ID
追蹤層次:	1
類型:	MQCFBS
長度上限:	MQ_CONNECTION_ID_LENGTH

### **QueueManagerName**

說明:	MQCONN (X) 呼叫中所使用佇列管理程式的 (未解析) 名稱
PCF 參數:	MQCA_Q_MGR_NAME
追蹤層次:	1
類型:	MQCFST
長度上限:	MQ_Q_MGR_NAME_LENGTH

### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型:	MQCFIN

### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型:	MQCFIN

### **ConnectOptions**

說明:	從 MQCNO_* 值衍生的連線選項 註: 僅限 MQCONN
PCF 參數:	MQIACF_CONNECT_OPTIONS
追蹤層次:	2
類型:	MQCFIN

### **ConnectionOptionsStructure**

說明:	MQCNO 結構。 註: 僅限 MQCONN
PCF 參數:	MQBACF_MQCNO_STRUCT
追蹤層次:	3
類型:	MQCFBS
長度上限:	MQCNO 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **ChannelDefinitionStructure**

說明:	MQCD 結構。 註: 僅用戶端連線
PCF 參數:	MQBACF_MQCD_STRUCT
追蹤層次:	3
類型:	MQCFBS
長度上限:	MQCD 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如, 作為 IBM MQ 用戶端所花費的時間。 註: 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型:	MQCFIN64

### **MQCTL**

應用程式已啟動 MQCTL MQI 函數

### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型:	MQCFIN

### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型:	MQCFIN

### **CtlOperation**

說明:	其中一個 MQOP_* 值
PCF 參數:	MQIACF_CTL_OPERATION
追蹤層次:	1
類型:	MQCFIN

### **QMGrOpDuration**

說明：佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。

註：此計時器的精確度會根據您企業使用的平台而有所不同。

PCF 參數: MQIAMO64\_QMGR\_OP\_DURATION  
追蹤層次: 2  
類型: MQCFIN64

### **MQDISC**

應用程式已啟動 MQDISC MQI 函數

#### **CompCode**

說明：指出作業結果的完成碼  
PCF 參數: MQIACF\_COMP\_CODE  
追蹤層次: 1  
類型: MQCFIN

#### **Reason**

說明：作業的原因碼結果  
PCF 參數: MQIACF\_REASON\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **MQGET**

應用程式已啟動 MQGET MQI 函數

#### **ObjectHandle**

說明：物件控點  
PCF 參數: MQIACF\_HOBJ  
追蹤層次: 1  
類型: MQCFIN

#### **GetOptions**

說明：來自 MQGMO.Options  
PCF 參數: MQIACF\_GET\_OPTIONS  
追蹤層次: 1  
類型: MQCFIN

#### **CompCode**

說明：指出作業結果的完成碼  
PCF 參數: MQIACF\_COMP\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **Reason**

說明： 作業的原因碼結果  
PCF 參數： MQIACF\_REASON\_CODE  
追蹤層次： 1  
類型： MQCFIN

### **MsgBuffer**

說明： 訊息資料。如果 TRACEDATA=NONE，則會省略此參數  
PCF 參數： MQBACF\_MESSAGE\_DATA  
追蹤層次： 1  
類型： MQCFBS  
長度上限： 長度由 APPTRACE 配置中設定的 TRACEDATA () 參數所控管。(包括在追蹤訊息中作為 MQIACF\_TRACE\_DATA\_LENGTH)。

### **MsgLength**

說明： 訊息的長度。  
PCF 參數： MQIACF\_MSG\_LENGTH  
追蹤層次： 1  
類型： MQCFIN

### **HighResTime**

說明： 自 1970 年 1 月 1 日午夜 (UTC) 以來的作業時間 (微秒)  
註： 此計時器的精確度根據高解析度計時器的平台支援而有所不同  
PCF 參數： MQIAM064\_HIGHRES\_TIME  
追蹤層次： 2  
類型： MQCFIN64

### **BufferLength**

說明： 應用程式提供的緩衝區長度  
PCF 參數： MQIACF\_BUFFER\_LENGTH  
追蹤層次： 2  
類型： MQCFIN

### **ObjectName**

說明： 已開啟物件的名稱  
PCF 參數： MQCACF\_OBJECT\_NAME  
追蹤層次： 2  
類型： MQCFST  
長度： MQ\_Q\_NAME\_LENGTH

### **ResolvedQName**

說明:	從中擷取訊息之佇列的本端名稱。
PCF 參數:	MQCACF_RESOLVED_Q_NAME
追蹤層次:	2
類型:	MQCFST
長度上限:	MQ_Q_NAME_LENGTH

### **ReportOptions**

說明:	訊息報告選項
PCF 參數:	MQIACF_REPORT
追蹤層次:	2
類型:	MQCFIN

### **MsgType**

說明:	訊息類型
PCF 參數:	MQIACF_MSG_TYPE
追蹤層次:	2
類型:	MQCFIN

### **Expiry**

說明:	訊息期限
PCF 參數:	MQIACF_EXPIRY
追蹤層次:	2
類型:	MQCFIN

### **Format**

說明:	訊息資料的格式名稱
PCF 參數:	MQCACH_FORMAT_NAME
追蹤層次:	2
類型:	MQCFST
長度上限:	MQ 格式長度

### **Priority**

說明:	訊息優先順序
PCF 參數:	MQIACF_PRIORITY
追蹤層次:	2
類型:	MQCFIN

### **Persistence**

說明:	訊息持續性
PCF 參數:	MQIACF_持續性

追蹤層次: 2  
類型: MQCFIN

### ***MsgId***

說明: 訊息 ID  
PCF 參數: MQBACF\_MSG\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度上限: MQ\_MSG\_ID\_LENGTH

### ***CorrelId***

說明: 相關性 ID  
PCF 參數: MQBACF\_CORREL\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度上限: MQ\_CORREL\_ID\_LENGTH

### ***ReplyToQueue***

說明:  
PCF 參數: MQCACF\_REPLY\_TO\_Q  
追蹤層次: 2  
類型: MQCFST  
長度上限: MQ\_Q\_NAME\_LENGTH

### ***ReplyToQMgr***

說明:  
PCF 參數: MQCACF\_REPLY\_TO\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度上限: MQ\_Q\_MGR\_NAME\_LENGTH

### ***CodedCharSetId***

說明: 訊息資料的字集 ID  
PCF 參數: MQIA\_CODED\_CHAR\_SET\_ID  
追蹤層次: 2  
類型: MQCFIN

### ***Encoding***

說明: 訊息資料的數值編碼。  
PCF 參數: MQIACF\_ENCODING  
追蹤層次: 2

類型: MQCFIN

### **PutDate**

說明:

PCF 參數: MQCACF\_PUT\_DATE

追蹤層次: 2

類型: MQCFST

長度上限: MQ\_PUT\_DATE\_LENGTH

### **PutTime**

說明:

PCF 參數: MQCACF\_PUT\_TIME

追蹤層次: 2

類型: MQCFST

長度上限: MQ\_PUT\_TIME\_LENGTH

### **ResolvedQName**

說明: 當 ResolvedType 為 MQOT\_Q 時, ObjectHandle 所參照的佇列名稱。

PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME

追蹤層次: 2

類型: MQCFST

長度: MQ\_Q\_NAME\_LENGTH。

### **ResObjectString**

說明: 當 ResolvedType 是 MQOT\_TOPIC 時, ObjectHandle 所參照的物件名稱。

PCF 參數: MQCACF\_RESOLVED\_OBJECT\_STRING

追蹤層次: 2

類型: MQCFST

長度: 長度會改變。

### **ResolvedType**

說明: ObjectHandle 所參照的物件類型。可能的值為 MQOT\_Q、MQOT\_TOPIC 或 MQOT\_NONE。

PCF 參數: MQIACF\_RESOLVED\_TYPE

追蹤層次: 2

類型: MQCFIN

### **PolicyName**

說明: 套用至此訊息的原則名稱。

註: 僅限 AMS 受保護訊息

PCF 參數: MQCA\_POLICY\_NAME

追蹤層次: 2  
類型: MQCFST  
長度: MQ\_OBJECT\_NAME\_LENGTH

### ***XmitqMsgId***

說明: 傳輸佇列標頭中訊息的訊息 ID。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQBACF\_XQH\_MSG\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_MSG\_ID\_LENGTH

### ***XmitqCorrelId***

說明: 傳輸佇列標頭中訊息的相關性 ID。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQBACF\_XQH\_CORREL\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_CORREL\_ID\_LENGTH

### ***XmitqPutTime***

說明: 訊息在傳輸佇列標頭中的放置時間。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_PUT\_TIME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

說明: 訊息在傳輸佇列標頭中的放置日期。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_PUT\_DATE  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

說明: 傳輸佇列標頭中訊息的遠端佇列目的地。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_REMOTE\_Q\_NAME

追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMGr***

說明: 傳輸佇列標頭中訊息的遠端佇列管理程式目的地。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_REMOTE\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### ***MsgDescStructure***

說明: MQMD 結構。  
PCF 參數: MQBACF\_MQMD\_STRUCT  
追蹤層次: 3  
類型: MQCFBS  
長度上限: MQMD 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### ***GetMsgOptsStructure***

說明: MQGMO 結構。  
PCF 參數: MQBACF\_MQGMO\_STRUCT  
追蹤層次: 3  
類型: MQCFBS  
長度上限: MQGMO 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### ***QMGrOpDuration***

說明: 佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。  
持續時間不包括在佇列管理程式之外所花費的時間。例如, 作為 IBM MQ 用戶端所花費的時間。  
註: 此計時器的精確度會根據您企業使用的平台而有所不同。

PCF 參數: MQIAMO64\_QMGR\_OP\_DURATION  
追蹤層次: 2  
類型: MQCFIN64

### ***MQINQ***

應用程式已啟動 MQINQ MQI 函數

### ***ObjectHandle***

說明: 物件控點  
PCF 參數: MQIACF\_HOBJ  
追蹤層次: 1

類型: MQCFIN

### **CompCode**

說明: 指出作業結果的完成碼

PCF 參數: MQIACF\_COMP\_CODE

追蹤層次: 1

類型: MQCFIN

### **Reason**

說明: 作業的原因碼結果

PCF 參數: MQIACF\_REASON\_CODE

追蹤層次: 1

類型: MQCFIN

### **SelectorCount**

說明: 在「選取元」陣列中提供的選取元計數。

PCF 參數: MQIACF\_SELECTOR\_COUNT

追蹤層次: 2

類型: MQCFIN

### **Selectors**

說明: MQINQ 必須傳回其值的屬性 (整數或字元) 清單。

PCF 參數: MQIACF\_SELECTORS

追蹤層次: 2

類型: MQCFIL

### **ResolvedQName**

說明: 當 ResolvedType 為 MQOT\_Q 時, ObjectHandle 所參照的佇列名稱。

PCF 參數: MQCACF\_RESOLVED\_Q\_NAME

追蹤層次: 2

類型: MQCFST

長度上限: MQ\_Q\_NAME\_LENGTH

### **ResObjectString**

說明: 當 ResolvedType 是 MQOT\_TOPIC 時, ObjectHandle 所參照的物件名稱。

PCF 參數: MQCACF\_RESOLVED\_OBJECT\_STRING

追蹤層次: 2

類型: MQCFST

長度上限: 長度不同

### **ResolvedType**

說明:	ObjectHandle 所參照的物件類型。可能的值為 MQOT_Q、MQOT_TOPIC 或 MQOT_NONE。
PCF 參數:	MQIACF_RESOLVED_TYPE
追蹤層次:	2
類型:	MQCFIN

### **IntAttrCount**

說明:	查詢作業傳回的整數屬性數目
PCF 參數:	MQIACF_INTATTR_COUNT
追蹤層次:	3
類型:	MQCFIN

### **IntAttr**

說明:	查詢作業傳回的整數屬性值。只有在 MQINQ 傳回時 IntAttr 計數 > 0 時，才會出現此參數。
PCF 參數:	MQIACF_INT_ATTRS
追蹤層次:	3
類型:	MQCFIL

### **CharAttr**

說明:	查詢作業傳回的字元屬性。值會連結在一起。只有在 MQINQ 傳回時，CharAttr 長度 > 0 時才會包含此參數。
PCF 參數:	MQCACF_CHAR_ATTRS
追蹤層次:	3
類型:	MQCFST

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### **MQOPEN**

應用程式已啟動 MQOPEN MQI 函數

### **ObjectType**

說明:	傳入 MQOT.ObjectType
PCF 參數:	MQIACF_OBJECT_TYPE
追蹤層次:	1

類型: MQCFIN

### **ObjectName**

說明: 在嘗試解析任何佇列名稱之前, 傳遞至 MQI 呼叫的物件名稱。

PCF 參數: MQCACF\_OBJECT\_NAME

追蹤層次: 1

類型: MQCFST

長度上限: MQ\_Q\_NAME\_LENGTH

### **ObjectQMgrName**

說明: 在嘗試任何佇列名稱解析之前, 傳遞至 MQI 呼叫的物件佇列管理程式名稱。

PCF 參數: MQCACF\_OBJECT\_Q\_MGR\_NAME

追蹤層次: 1

類型: MQCFST

長度上限: MQ\_Q\_MGR\_NAME\_LENGTH

### **ObjectHandle**

說明: 物件控點

PCF 參數: MQIACF\_HOBJ

追蹤層次: 1

類型: MQCFIN

### **CompCode**

說明: 指出作業結果的完成碼

PCF 參數: MQIACF\_COMP\_CODE

追蹤層次: 1

類型: MQCFIN

### **Reason**

說明: 作業的原因碼結果

PCF 參數: MQIACF\_REASON\_CODE

追蹤層次: 1

類型: MQCFIN

### **OpenOptions**

說明: 用來開啟物件的選項

PCF 參數: MQIACF\_OPEN\_OPTIONS

追蹤層次: 1

類型: MQCFIN

### **AlternateUserId**

說明: 只有在指定 MQOO\_ALTERNATE\_USER\_AUTHORITY 時才會併入

PCF 參數: MQCACF\_ALTERNATE\_USERID  
追蹤層次: 2  
類型: MQCFST  
長度上限: MQ\_USER\_ID\_LENGTH

### **RecsPresent**

說明: 呈現的物件名稱記錄數。只有在 MQOD 版本 > = MQOD\_VERSION\_2 時才會併入  
PCF 參數: MQIACF\_RECS\_PRESENT  
追蹤層次: 1  
類型: MQCFIN

### **KnownDestCount**

說明: 僅當 MQOD 版本 > = MQOD\_VERSION\_2 時才順利開啟的本端佇列數  
PCF 參數: MQIACF\_KNOWN\_DEST\_COUNT  
追蹤層次: 1  
類型: MQCFIN

### **UnknownDestCount**

說明: 僅當 MQOD 版本 > = MQOD\_VERSION\_2 時才順利開啟的遠端佇列數  
PCF 參數: MQIACF\_UNKNOWN\_DEST\_COUNT  
追蹤層次: 1  
類型: MQCFIN

### **InvalidDestCount**

說明: 只有在 MQOD 版本 > = MQOD\_VERSION\_2 時，才會包含無法開啟的佇列數  
PCF 參數: MQIACF\_INVALID\_DEST\_COUNT  
追蹤層次: 1  
類型: MQCFIN

### **DynamicQName**

說明: 作為輸入傳遞至 MQOPEN 呼叫的動態佇列名稱。  
PCF 參數: MQCACF\_DYNAMIC\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度上限: MQ\_Q\_NAME\_LENGTH

### **ResolvedLocalQName** <sup>1 2</sup>

說明: 包含已執行名稱解析之後的本端佇列名稱。(例如，對於遠端佇列，這將是傳輸佇列的名稱)  
PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST

範圍: 若為 MQOD.Version 版小於 MQOD\_VERSION\_3, 這包含 MQOD.ObjectName 欄位。若為 MQOD.Version 版等於或大於 MQOD\_VERSION\_3, 這包含 MQOD 中的值。ResolvedQName 欄位。

長度上限: MQ\_Q\_NAME\_LENGTH

### **ResolvedLocalQMgrName** <sup>1 2</sup>

說明: 已執行名稱解析之後的本端佇列管理程式名稱。

PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_MGR

追蹤層次: 2

類型: MQCFST

範圍: 僅當 MQOD.Version > = MQOD\_VERSION\_3

長度上限: MQ\_Q\_MGR\_NAME\_LENGTH

### **ResolvedQName** <sup>1 2</sup>

說明: 已執行名稱解析之後的佇列名稱。

PCF 參數: MQCACF\_RESOLVED\_Q\_NAME

追蹤層次: 2

類型: MQCFST

範圍: 若為 MQOD.Version 版小於 MQOD\_VERSION\_3, 這包含 MQOD.ObjectName 欄位。若為 MQOD.Version 版等於或大於 MQOD\_VERSION\_3, 這包含 MQOD 中的值。ResolvedQName 欄位。

長度上限: MQ\_Q\_NAME\_LENGTH

### **ResolvedQMgrName** <sup>1 2</sup>

說明: 包含在執行名稱解析之後的佇列管理程式名稱。若為 MQOD.Version 版小於 MQOD\_VERSION\_3, 這包含 MQOD 的值。ObjectQMgrMQOPEN 呼叫完成之後的「名稱」欄位。若為 MQOD.Version 版等於或大於 MQOD\_VERSION\_3, 這包含 MQOD 中的值。ResolvedQMgr 名稱欄位。

PCF 參數: MQCACF\_RESOLVED\_Q\_MGR

追蹤層次: 2

類型: MQCFST

長度上限: MQ\_Q\_MGR\_NAME\_LENGTH

### **AlternateSecurityId**

說明: 替代安全 ID。僅當 MQOD.Version 版等於或大於 MQOD\_VERSION\_3、指定 MQOO\_ALTERNATE\_USER\_AUTHORITY 及 MQOD.AlternateSecurityId 不等於 MQSID\_NONE。

PCF 參數: MQBACF\_ALTERNATE\_SECURITYID

追蹤層次: 2

類型: MQCFBS

長度上限: MQ\_SECURITY\_ID\_LENGTH

### **ObjectString**

說明:	長物件名稱。僅在 MQOD.Version 版等於或大於 MQOD_VERSION_4 及 MQOD.ObjectString 是 MQVS_NULL_TERMINATED 或大於零。
PCF 參數:	MQCACF_OBJECT_STRING
追蹤層次:	2
類型:	MQCFST
長度上限:	長度會改變。

### **SelectionString**

說明:	Selection 字串。僅在 MQOD.Version 版等於或大於 MQOD_VERSION_4 及 MQOD 的 VSLength 欄位。SelectionString 是 MQVS_NULL_TERMINATED 或大於零。
PCF 參數:	MQCACF_SELECTION_STRING
追蹤層次:	2
類型:	MQCFST
長度上限:	長度會改變。

### **ResObjectString**

說明:	佇列管理程式解析 ObjectName 欄位中所提供的名稱之後的長物件名稱。只有在 MQOD.Version 版等於或大於 MQOD_VERSION_4，且 VSLength 是 MQVS_NULL_TERMINATED 或大於零。
PCF 參數:	MQCACF_RESOLVED_OBJECT_STRING
追蹤層次:	2
類型:	MQCFST
長度上限:	長度會改變。

### **ResolvedType**

說明:	正在開啟的已解析 (基本) 物件類型。僅在 MQOD.Version 版等於或大於 MQOD_VERSION_4。可能的值為 MQOT_Q、MQOT_TOPIC 或 MQOT_NONE。
PCF 參數:	MQIACF_RESOLVED_TYPE
追蹤層次:	2
類型:	MQCFIN

### **QMgrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型:	MQCFIN64

### 應用程式活動配送清單 PCF 群組標頭結構

如果 MQOPEN 函數開啟配送清單，則針對配送清單中的每一個佇列，MQOPEN 參數會包括一個 AppActivityDistList PCF 群組，最多可達 RecsPresent 中編號的結構數目。AppActivityDistList PCF 群組會結合來自 MQOR 及 MQRR 結構的資訊來識別佇列名稱，並指出佇列上開啟作業的結果。AppActivityDistList 群組一律以下列 MQCFGR 結構開始：

MQCFGR 欄位	值	說明
類型	MQCFT_GROUP	
StrucLength	MQCFGR 結構的長度 (以位元組為單位)	
參數	MQGACF_APP_DIST_LIST	配送清單群組參數
ParameterCount	4	此群組內包含的 MQCFGR 結構後面的參數結構數目。

### ObjectName

說明： 配送清單 MQ\_Q\_NAME\_LENGTH 中的佇列名稱。僅在提供 MQOR 結構時才包含。

PCF 參數： MQCACF\_OBJECT\_NAME

追蹤層次： 2

類型： MQCFST

長度： MQ\_Q\_NAME\_LENGTH。僅在提供 MQOR 結構時才包含。

### ObjectQMgrName

說明： 在其中定義 ObjectName 中所指定佇列的佇列管理程式名稱。

PCF 參數： MQCACF\_OBJECT\_Q\_MGR\_NAME

追蹤層次： 2

類型： MQCFST

長度： MQ\_Q\_MGR\_NAME\_LENGTH。僅在提供 MQOR 結構時才包含。

### CompCode

說明： 指出此物件開啟結果的完成碼。只有在提供 MQRR 結構且 MQOPEN 的原因碼為 MQRC\_MULTIPLE\_REASONS 時才會併入

PCF 參數： MQIACF\_COMP\_CODE

追蹤層次： 2

類型： MQCFIN

### Reason

說明： 指出開啟此物件的結果的原因碼。只有在提供 MQRR 結構且 MQOPEN 的原因碼為 MQRC\_MULTIPLE\_REASONS 時才會併入

PCF 參數： MQIACF\_REASON\_CODE

- 1 只有在所開啟的物件解析為佇列，且針對 MQOO\_INPUT\_\*、MQOO\_OUTPUT 或 MQOO\_BROWSE 開啟佇列時，才會包含此參數
- 2 只有在 ResolvedLocal 完整名稱參數不同於 ResolvedQName 參數時，才會包含此參數。

追蹤層次: 2  
類型: MQCFIN

### *MQPUT*

應用程式已啟動 MQPUT MQI 函數。

### **ObjectHandle**

說明: 物件控點  
PCF 參數: MQIACF\_HOBJ  
追蹤層次: 1  
類型: MQCFIN

### **PutOptions**

說明: 來自 MQPMO.Options  
PCF 參數: MQIACF\_PUT\_OPTIONS  
追蹤層次: 1  
類型: MQCFIN

### **CompCode**

說明: 指出作業結果的完成碼  
PCF 參數: MQIACF\_COMP\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **Reason**

說明: 作業的原因碼結果  
PCF 參數: MQIACF\_REASON\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **MsgBuffer**

說明: 訊息資料。  
PCF 參數: MQBACF\_MESSAGE\_DATA  
追蹤層次: 1  
類型: MQCFBS  
長度: 長度由 APPTTRACE 配置中設定的 TRACEDATA () 參數所控管。如果 TRACEDATA=NONE , 則會省略此參數。

### **MsgLength**

說明: 訊息的長度。  
PCF 參數: MQIACF\_MSG\_LENGTH  
追蹤層次: 1  
類型: MQCFIN

### **RecsPresent**

說明： 呈現的放置訊息記錄或回應記錄數目。 僅在 MQPMO 版本 > = MQPMO\_VERSION\_2 時才包含

PCF 參數： MQIACF\_RECS\_PRESENT

追蹤層次： 1

類型： MQCFIN

### **KnownDestCount**

說明： 順利傳送至本端佇列的訊息數

PCF 參數： MQIACF\_KNOWN\_DEST\_COUNT

追蹤層次： 1

類型： MQCFIN

### **UnknownDestCount**

說明： 順利傳送至遠端佇列的訊息數

PCF 參數： MQIACF\_UNKNOWN\_DEST\_COUNT

追蹤層次： 1

類型： MQCFIN

### **InvalidDestCount**

說明： 無法傳送的訊息數

PCF 參數： MQIACF\_INVALID\_DEST\_COUNT

追蹤層次： 1

類型： MQCFIN

### **HighResTime**

說明： 自 1970 年 1st (UTC) 午夜以來的作業時間 (微秒)  
註： 根據高解析度計時器的平台支援， 此計時器的精確度會有所不同。

PCF 參數： MQIAMO64\_HIGHRES\_TIME

追蹤層次： 2

類型： MQCFIN64

### **ObjectName**

說明： 已開啟物件的名稱。

PCF 參數： MQCACF\_OBJECT\_NAME

追蹤層次： 2

類型： MQCFST

長度： MQ\_Q\_NAME\_LENGTH

### **ResolvedQName**

說明： 已執行佇列名稱解析之後的佇列名稱。

PCF 參數: MQCACF\_RESOLVED\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### ***ResolvedQMgrName***

說明: 已執行名稱解析之後的佇列管理程式名稱。  
PCF 參數: MQCACF\_RESOLVED\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_MGR\_NAME\_LENGTH

### ***ResolvedLocalQName***<sup>3</sup>

說明: 包含已執行名稱解析之後的本端佇列名稱。  
PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST

### ***ResolvedLocalQMgrName***<sup>3</sup>

說明: 包含已執行名稱解析之後的本端佇列管理程式名稱。  
PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_MGR\_NAME\_LENGTH

### ***ReportOptions***

說明: 訊息報告選項  
PCF 參數: MQIACF\_REPORT  
追蹤層次: 2  
類型: MQCFIN

### ***MsgType***

說明: 訊息類型  
PCF 參數: MQIACF\_MSG\_TYPE  
追蹤層次: 2  
類型: MQCFIN

### ***Expiry***

說明: 訊息期限  
PCF 參數: MQIACF\_EXPIRY  
追蹤層次: 2

類型: MQCFIN

### **Format**

說明: 訊息資料的格式名稱  
PCF 參數: MQCACH\_FORMAT\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ 格式長度

### **Priority**

說明: 訊息優先順序  
PCF 參數: MQIACF\_PRIORITY  
追蹤層次: 2  
類型: MQCFIN

### **Persistence**

說明: 訊息持續性  
PCF 參數: MQIACF\_持續性  
追蹤層次: 2  
類型: MQCFIN

### **MsgId**

說明: 訊息 ID  
PCF 參數: MQBACF\_MSG\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_MSG\_ID\_LENGTH

### **CorrelId**

說明: 相關性 ID  
PCF 參數: MQBACF\_CORREL\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_CORREL\_ID\_LENGTH

### **ReplyToQueue**

說明:  
PCF 參數: MQCACF\_REPLY\_TO\_Q  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### ***ReplyToQMgr***

說明：  
PCF 參數: MQCACF\_REPLY\_TO\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_MGR\_NAME\_LENGTH

### ***CodedCharSetId***

說明: 訊息資料的字集 ID  
PCF 參數: MQIA\_CODED\_CHAR\_SET\_ID  
追蹤層次: 2  
類型: MQCFIN

### ***Encoding***

說明: 訊息資料的數值編碼。  
PCF 參數: MQIACF\_ENCODING  
追蹤層次: 2  
類型: MQCFIN

### ***PutDate***

說明：  
PCF 參數: MQCACF\_PUT\_DATE  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_PUT\_DATE\_LENGTH

### ***PutTime***

說明：  
PCF 參數: MQCACF\_PUT\_TIME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_PUT\_TIME\_LENGTH

### ***ResolvedQName***

說明: 當 ResolvedType 為 MQOT\_Q 時， ObjectHandle 所參照的佇列名稱。  
PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH。

### ***ResObjectString***

說明:	當 ResolvedType 是 MQOT_TOPIC 時, ObjectHandle 所參照的物件名稱。
PCF 參數:	MQCACF_RESOLVED_OBJECT_STRING
追蹤層次:	2
類型:	MQCFST
長度:	長度會改變。

### ***ResolvedType***

說明:	ObjectHandle 所參照的物件類型。可能的值為 MQOT_Q、MQOT_TOPIC 或 MQOT_NONE。
PCF 參數:	MQIACF_RESOLVED_TYPE
追蹤層次:	2
類型:	MQCFIN

### ***PolicyName***

說明:	套用至此訊息的原則名稱。 註: 僅限 AMS 受保護訊息
PCF 參數:	MQCA_POLICY_NAME
追蹤層次:	2
類型:	MQCFST
長度:	MQ_OBJECT_NAME_LENGTH

### ***XmitqMsgId***

說明:	傳輸佇列標頭中訊息的訊息 ID。 註: 僅當格式為 MQFMT_XMIT_Q_HEADER 時
PCF 參數:	MQBACF_XQH_MSG_ID
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_MSG_ID_LENGTH

### ***XmitqCorrelId***

說明:	傳輸佇列標頭中訊息的相關性 ID。 註: 僅當格式為 MQFMT_XMIT_Q_HEADER 時
PCF 參數:	MQBACF_XQH_CORREL_ID
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_CORREL_ID_LENGTH

### ***XmitqPutTime***

說明： 訊息在傳輸佇列標頭中的放置時間。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數： MQCACF\_XQH\_PUT\_TIME  
追蹤層次： 2  
類型： MQCFST  
長度： MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

說明： 訊息在傳輸佇列標頭中的放置日期。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數： MQCACF\_XQH\_PUT\_DATE  
追蹤層次： 2  
類型： MQCFST  
長度： MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

說明： 傳輸佇列標頭中訊息的遠端佇列目的地。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數： MQCACF\_XQH\_REMOTE\_Q\_NAME  
追蹤層次： 2  
類型： MQCFST  
長度： MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMgr***

說明： 傳輸佇列標頭中訊息的遠端佇列管理程式目的地。  
註： 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數： MQCACF\_XQH\_REMOTE\_Q\_MGR  
追蹤層次： 2  
類型： MQCFST  
長度： MQ\_Q\_NAME\_LENGTH

### ***PutMsgOptsStructure***

說明： MQPMO 結構。

PCF 參數： MQBACF\_MQPMO\_STRUCT  
追蹤層次： 3  
類型： MQCFBS  
長度： MQPMO 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

## **QMGrOpDuration**

說明：佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。

註：此計時器的精確度會根據您企業使用的平台而有所不同。

PCF 參數: MQIAMO64\_QMGR\_OP\_DURATION  
追蹤層次: 2  
類型: MQCFIN64

### **MQPUT 應用程式活動配送清單 PCF 群組標頭結構**

如果 MQPUT 函數放入配送清單，則 MQPUT 參數會包括一個 AppCompatActivityDistList PCF 群組。對於配送清單中的每一個佇列，請參閱第 227 頁的『應用程式活動配送清單 PCF 群組標頭結構』。AppCompatActivityDistList PCF 群組會結合來自 MQPMR 及 MQRR 結構的資訊，以識別 PUT 參數，並指出每一個佇列上 PUT 作業的結果。對於 MQPUT 作業，AppCompatActivityDistList 群組包含下列部分或所有參數 (如果原因碼為 MQRC\_MULTIPLE\_REASONS，且其他參數由 MQPMO.PutMsgRecFields 欄位決定，則會呈現 CompCode 及 Reason):

#### **CompCode**

說明：指出作業結果的完成碼。只有在提供 MQRR 結構且 MQPUT 的原因碼為 MQRC\_MULTIPLE\_REASONS 時才會併入

PCF 參數: MQIACF\_COMP\_CODE  
追蹤層次: 2  
類型: MQCFIN

#### **Reason**

說明：指出此物件放置結果的原因碼。只有在提供 MQRR 結構且 MQPUT 的原因碼為 MQRC\_MULTIPLE\_REASONS 時才會併入

PCF 參數: MQIACF\_REASON\_CODE  
追蹤層次: 2  
類型: MQCFIN

#### **MsgId**

說明：訊息 ID。僅當 MQPMR 結構為 provided.and PutMsgRecFields 包括 MQPMRF\_MSG\_ID 時才包括

PCF 參數: MQBACF\_MSG\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_MSG\_ID\_LENGTH

#### **CorrelId**

說明：相關性 ID。只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF\_CORREL\_ID 時才會併入

<sup>3</sup> 只有在 ResolvedLocal 完整名稱參數不同於 ResolvedQName 參數時，才會包含此參數。

PCF 參數: MQBACF\_CORREL\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_CORREL\_ID\_LENGTH

### **GroupId**

說明: 群組 ID。只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF\_GROUP\_ID 時才會併入。  
PCF 參數: MQBACF\_GROUP\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_GROUP\_ID\_LENGTH

### **Feedback**

說明: 意見。只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF\_FEEDBACK 時才會併入。  
PCF 參數: MQIACF\_ 回饋  
追蹤層次: 2  
類型: MQCFIN

### **AccountingToken**

說明: AccountingToken. 只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF\_ACCOUNTING\_TOKEN 時才會併入。  
PCF 參數: MQBACF\_ACCOUNTING\_TOKEN  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_ACCOUNTING\_TOKEN\_LENGTH。

### **MQPUT1**

應用程式已啟動 MQPUT1 MQI 函數

### **ObjectType**

說明: 傳入 MQOT.ObjectType  
PCF 參數: MQIACF\_OBJECT\_TYPE  
追蹤層次: 1  
類型: MQCFIN

### **ObjectName**

說明: 在嘗試解析任何佇列名稱之前, 傳遞至 MQI 呼叫的物件名稱。  
PCF 參數: MQCACF\_OBJECT\_NAME  
追蹤層次: 1  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### **ObjectQMgrName**

說明:	在嘗試任何佇列名稱解析之前，傳遞至 MQI 呼叫的物件佇列管理程式名稱。
PCF 參數:	MQCACF_OBJECT_Q_MGR_NAME
追蹤層次:	1
類型:	MQCFST
長度:	MQ_Q_MGR_NAME_LENGTH

### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1
類型:	MQCFIN

### **Reason**

說明:	作業的原因碼結果
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	1
類型:	MQCFIN

### **PutOptions**

說明:	來自 MQPMO.Options
PCF 參數:	MQIACF_PUT_OPTIONS
追蹤層次:	1
類型:	MQCFIN

### **AlternateUserId**

說明:	只有在指定 MQPMO_ALTERNATE_USER_AUTHORITY 時才會併入。
PCF 參數:	MQCACF_ALTERNATE_USERID
追蹤層次:	2
類型:	MQCFST
長度:	MQ_USER_ID_LENGTH

### **RecsPresent**

說明:	呈現的物件名稱記錄數
PCF 參數:	MQIACF_RECS_PRESENT
追蹤層次:	1
類型:	MQCFIN

### **KnownDestCount**

說明:	順利開啟的本端佇列數
PCF 參數:	MQIACF_KNOWN_DEST_COUNT

追蹤層次: 1  
類型: MQCFIN

### **UnknownDestCount**

說明: 順利開啟的遠端佇列數  
PCF 參數: MQIACF\_UNKNOWN\_DEST\_COUNT  
追蹤層次: 1  
類型: MQCFIN

### **InvalidDestCount**

說明: 無法開啟的佇列數  
PCF 參數: MQIACF\_INVALID\_DEST\_COUNT  
追蹤層次: 1  
類型: MQCFIN

### **MsgBuffer**

說明: 訊息資料。  
PCF 參數: MQBACF\_MESSAGE\_DATA  
追蹤層次: 1  
類型: MQCFBS  
長度: 長度由 APPTTRACE 配置中設定的 TRACEDATA () 參數所控管。如果 TRACEDATA=NONE , 則會省略此參數。

### **MsgLength**

說明: 訊息的長度。  
PCF 參數: MQIACF\_MSG\_LENGTH  
追蹤層次: 1  
類型: MQCFIN

### **HighResTime**

說明: 自 1970 年 1st (UTC) 午夜以來的作業時間 (微秒)  
註: 此計時器的精確度會根據高解析度計時器的平台支援而有所不同。  
PCF 參數: MQIAMO64\_HIGHRES\_TIME  
追蹤層次: 2  
類型: MQCFIN64

### **ResolvedQName**

說明: 已執行佇列名稱解析之後的佇列名稱。  
PCF 參數: MQCACF\_RESOLVED\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST

長度: MQ\_Q\_NAME\_LENGTH

### **ResolvedQMgrName**

說明: 已執行名稱解析之後的佇列管理程式名稱。

PCF 參數: MQCACF\_RESOLVED\_Q\_MGR

追蹤層次: 2

類型: MQCFST

長度: MQ\_Q\_MGR\_NAME\_LENGTH

### **ResolvedLocalQName** <sup>4</sup>

說明: 包含已執行名稱解析之後的本端佇列名稱

PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME

追蹤層次: 2

類型: MQCFST

### **ResolvedLocalQMgrName** <sup>4</sup>

說明: 包含已執行名稱解析之後的本端佇列管理程式名稱。

PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_MGR

追蹤層次: 2

類型: MQCFST

長度: MQ\_Q\_MGR\_NAME\_LENGTH

### **AlternateSecurityId**

說明: 替代安全 ID。僅當 MQOD.Version 版等於或大於 MQOD\_VERSION\_3 及 MQOD.AlternateSecurityId 不等於 MQSID\_NONE。

PCF 參數: MQBACF\_ALTERNATE\_SECURITYID

追蹤層次: 2

類型: MQCFBS

長度: MQ\_SECURITY\_ID\_LENGTH

### **ObjectString**

說明: 長物件名稱。僅在 MQOD.Version 版等於或大於 MQOD\_VERSION\_4 及 MQOD.ObjectString 是 MQVS\_NULL\_TERMINATED 或大於零。

PCF 參數: MQCACF\_OBJECT\_STRING

追蹤層次: 2

類型: MQCFST

長度: 長度會改變。

### **ResObjectString**

說明: 佇列管理程式解析 ObjectName 欄位中所提供的名稱之後的長物件名稱。只有在 MQOD.Version 版等於或大於 MQOD\_VERSION\_4, 且 VSLength 是 MQVS\_NULL\_TERMINATED 或大於零。

PCF 參數: MQCACF\_RESOLVED\_OBJECT\_STRING

追蹤層次: 2  
類型: MQCFST  
長度: 長度會改變。

### ***ResolvedType***

說明: 正在開啟的已解析 (基本) 物件類型。僅在 MQOD.Version 版等於或大於 MQOD\_VERSION\_4。可能的值為 MQOT\_Q、MQOT\_TOPIC 或 MQOT\_NONE。  
PCF 參數: MQIACF\_RESOLVED\_TYPE  
追蹤層次: 2  
類型: MQCFIN

### ***ReportOptions***

說明: 訊息報告選項  
PCF 參數: MQIACF\_REPORT  
追蹤層次: 2  
類型: MQCFIN

### ***MsgType***

說明: 訊息類型  
PCF 參數: MQIACF\_MSG\_TYPE  
追蹤層次: 2  
類型: MQCFIN

### ***Expiry***

說明: 訊息期限  
PCF 參數: MQIACF\_EXPIRY  
追蹤層次: 2  
類型: MQCFIN

### ***Format***

說明: 訊息資料的格式名稱  
PCF 參數: MQCACH\_FORMAT\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ 格式長度

### ***Priority***

說明: 訊息優先順序  
PCF 參數: MQIACF\_PRIORITY  
追蹤層次: 2  
類型: MQCFIN

### **Persistence**

說明: 訊息持續性  
PCF 參數: MQIACF\_持續性  
追蹤層次: 2  
類型: MQCFIN

### **MsgId**

說明: 訊息 ID  
PCF 參數: MQBACF\_MSG\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_MSG\_ID\_LENGTH

### **CorrelId**

PCF 參數: 相關性 ID  
說明: MQBACF\_CORREL\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_CORREL\_ID\_LENGTH

### **ReplyToQueue**

說明:  
PCF 參數: MQCACF\_REPLY\_TO\_Q  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### **ReplyToQMgr**

說明:  
PCF 參數: MQCACF\_REPLY\_TO\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度: MQCFST

### **CodedCharSetId**

說明: 訊息資料的字集 ID  
PCF 參數: MQIA\_CODED\_CHAR\_SET\_ID  
追蹤層次: 2  
類型: MQCFIN

### **Encoding**

說明:	訊息資料的數值編碼。
PCF 參數:	MQIACF_ENCODING
追蹤層次:	2
類型:	MQCFIN

### **PutDate**

說明:	
PCF 參數:	MQCACF_PUT_DATE
追蹤層次:	2
類型:	MQCFST
長度:	MQ_PUT_DATE_LENGTH

### **PutTime**

說明:	
PCF 參數:	MQCACF_PUT_TIME
追蹤層次:	2
類型:	MQCFST
長度:	MQ_PUT_TIME_LENGTH

### **PolicyName**

說明:	套用至此訊息的原則名稱。 註: 僅限 AMS 受保護訊息
PCF 參數:	MQCA_POLICY_NAME
追蹤層次:	2
類型:	MQCFST
長度:	MQ_OBJECT_NAME_LENGTH

### **XmitqMsgId**

說明:	傳輸佇列標頭中訊息的訊息 ID。 註: 僅當格式為 MQFMT_XMIT_Q_HEADER 時
PCF 參數:	MQBACF_XQH_MSG_ID
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_MSG_ID_LENGTH

### **XmitqCorrelId**

說明:	傳輸佇列標頭中訊息的相關性 ID。 註: 僅當格式為 MQFMT_XMIT_Q_HEADER 時
PCF 參數:	MQBACF_XQH_CORREL_ID

追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_CORREL\_ID\_LENGTH

### ***XmitqPutTime***

說明: 訊息在傳輸佇列標頭中的放置時間。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_PUT\_TIME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

說明: 訊息在傳輸佇列標頭中的放置日期。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_PUT\_DATE  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

說明: 傳輸佇列標頭中訊息的遠端佇列目的地。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_REMOTE\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMGr***

說明: 傳輸佇列標頭中訊息的遠端佇列管理程式目的地。  
註: 僅當格式為 MQFMT\_XMIT\_Q\_HEADER 時

PCF 參數: MQCACF\_XQH\_REMOTE\_Q\_MGR  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### ***PutMsgOptsStructure***

說明: MQPMO 結構。  
PCF 參數: MQBACF\_MQPMO\_STRUCT  
追蹤層次: 3

類型:	MQCFBS
長度:	MQPMO 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### **MQPUT1 AppActivityDistList PCF 群組標頭結構**

如果 MQPUT1 函數放入配送清單，則變數參數包括一個 AppActivityDistList PCF 群組。對於配送清單中的每一個佇列，請參閱第 227 頁的『應用程式活動配送清單 PCF 群組標頭結構』。AppActivityDistList PCF 群組會結合來自 MQOR、MQPMR 及 MQRR 結構的資訊，以識別物件及 PUT 參數，並指出每一個佇列上 PUT 作業的結果。對於 MQPUT1 作業，AppActivityDistList 群組包含下列部分或所有參數 (如果原因碼為 MQRC\_MULTIPLE\_REASONS 且其他參數由 MQPMO.PutMsgRecFields 欄位決定，則會呈現 CompCode、Reason、ObjectName 及 ObjectQMGr 名稱):

#### **CompCode**

說明:	指出此物件放置結果的完成碼。只有在提供 MQRR 結構且 MQPUT1 的原因碼為 MQRC_MULTIPLE_REASONS 時才會併入
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	2
類型:	MQCFIN

#### **Reason**

說明:	指出此物件放置結果的原因碼。只有在提供 MQRR 結構且 MQPUT1 的原因碼為 MQRC_MULTIPLE_REASONS 時才會併入
PCF 參數:	MQIACF_REASON_CODE
追蹤層次:	2
類型:	MQCFIN

#### **ObjectName**

說明:	配送清單中佇列的名稱。僅在提供 MQOR 結構時才包含。
PCF 參數:	MQCACF_OBJECT_NAME
追蹤層次:	2
類型:	MQCFST
長度:	MQ_Q_NAME_LENGTH

<sup>4</sup> 只有在 ResolvedLocal 完整名稱參數不同於 ResolvedQName 參數時，才會包含此參數。

### **MsgId**

說明:	訊息 ID。僅當 MQPMR 結構為 provided.and PutMsgRecFields 包括 MQPMRF_MSG_ID 時才包括
PCF 參數:	MQBACF_MSG_ID
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_MSG_ID_LENGTH

### **CorrelId**

說明:	相關性 ID。只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF_CORREL_ID 時才會併入
PCF 參數:	MQBACF_CORREL_ID
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_CORREL_ID_LENGTH

### **GroupId**

說明:	群組 ID。只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF_GROUP_ID 時才會併入。
PCF 參數:	MQBACF_GROUP_ID
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_GROUP_ID_LENGTH

### **Feedback**

說明:	意見。只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF_FEEDBACK 時才會併入。
PCF 參數:	MQIACF_ 回饋
追蹤層次:	2
類型:	MQCFIN

### **AccountingToken**

說明:	AccountingToken. 只有在 MQPMR 結構是 provided.and PutMsgRecFields 包括 MQPMRF_ACCOUNTING_TOKEN 時才會併入。
PCF 參數:	MQBACF_ACCOUNTING_TOKEN
追蹤層次:	2
類型:	MQCFBS
長度:	MQ_ACCOUNTING_TOKEN_LENGTH。

### **MQSET**

應用程式已啟動 MQSET MQI 函數

### **ObjectHandle**

說明:	物件控點
-----	------

PCF 參數: MQIACF\_HOBJ  
追蹤層次: 1  
類型: MQCFIN

### **CompCode**

說明: 指出作業結果的完成碼  
PCF 參數: MQIACF\_COMP\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **Reason**

說明: 作業的原因碼結果  
PCF 參數: MQIACF\_REASON\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **SelectorCount**

說明: 在「選取元」陣列中提供的選取元計數。  
PCF 參數: MQIACF\_SELECTOR\_COUNT  
追蹤層次: 2  
類型: MQCFIN

### **Selectors**

說明: MQSET 正在更新其值的屬性 (整數或字元) 清單。  
PCF 參數: MQIACF\_SELECTORS  
追蹤層次: 2  
類型: MQCFIL

### **ResolvedQName**

說明: 當 ResolvedType 為 MQOT\_Q 時, ObjectHandle 所參照的佇列名稱。  
PCF 參數: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
追蹤層次: 2  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH。

### **ResObjectString**

說明: 當 ResolvedType 是 MQOT\_TOPIC 時, ObjectHandle 所參照的物件名稱。  
PCF 參數: MQCACF\_RESOLVED\_OBJECT\_STRING  
追蹤層次: 2  
類型: MQCFST  
長度: 長度會改變。

### **ResolvedType**

說明:	ObjectHandle 所參照的物件類型。可能的值為 MQOT_Q、MQOT_TOPIC 或 MQOT_NONE。
PCF 參數:	MQIACF_RESOLVED_TYPE
追蹤層次:	2
類型	MQCFIN

### **IntAttrCount**

說明:	要由設定作業更新的整數屬性數目。
PCF 參數:	MQIACF_INTATTR_COUNT
追蹤層次:	3
類型:	MQCFIN

### **IntAttrS**

說明:	整數屬性值
PCF 參數:	MQIACF_INT_ATTRS
追蹤層次:	3
類型:	MQCFIL
範圍:	僅當 IntAttr 計數 > 0 時才會呈現此參數

### **CharAttrS**

說明:	要由設定作業更新的字元屬性。值會連結在一起。
PCF 參數:	MQCACF_CHAR_ATTRS
追蹤層次:	3
類型:	MQCFST
範圍:	僅當 CharAttr 長度 > 0 時才包括此參數

### **QMGrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
PCF 參數:	MQIAMO64_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

### **MQSUB**

應用程式已啟動 MQSUB MQI 函數

### **CompCode**

說明:	指出作業結果的完成碼
PCF 參數:	MQIACF_COMP_CODE
追蹤層次:	1

類型: MQCFIN

### **Reason**

說明: 作業的原因碼結果  
PCF 參數: MQIACF\_REASON\_CODE  
追蹤層次: 1  
類型: MQCFIN

### **SubHandle**

說明: 訂閱控點  
PCF 參數: MQIACF\_HSUB  
追蹤層次: 1  
類型: MQCFIN

### **ObjectHandle**

說明: 物件控點  
PCF 參數: MQIACF\_HOBJ  
追蹤層次: 1  
類型: MQCFIN

### **Options**

說明: 訂閱選項  
PCF 參數: MQIACF\_SUB\_OPTIONS  
追蹤層次: 1  
類型: MQCFIN

### **ObjectName**

說明: 物件的名稱。  
PCF 參數: MQCACF\_OBJECT\_NAME  
追蹤層次: 1  
類型: MQCFST  
長度: MQ\_Q\_NAME\_LENGTH

### **ObjectString**

說明: 長物件名稱。  
PCF 參數: MQCACF\_OBJECT\_STRING  
追蹤層次: 1  
類型: MQCFST  
範圍: 只有在 MQSD.ObjectString 大於零或 MQVS\_NULL\_TERMINATED。  
長度: 長度會改變。

### **AlternateUserId**

說明：  
PCF 參數: MQCACF\_ALTERNATE\_USERID  
追蹤層次: 2  
類型: MQCFST  
範圍: 只有在指定 MQSO\_ALTERNATE\_USER\_AUTHORITY 時才會併入。  
長度: MQ\_USER\_ID\_LENGTH

### **AlternateSecurityId**

說明: 替代安全 ID。  
PCF 參數: MQBACF\_ALTERNATE\_SECURITYID  
追蹤層次: 2  
類型: MQCFBS  
範圍: 只有在指定 MQSO\_ALTERNATE\_USER\_AUTHORITY 及 MQSD.AlternateSecurityId 不等於 MQSID\_NONE。  
長度: MQ\_SECURITY\_ID\_LENGTH

### **SubName**

說明: 訂閱名稱  
PCF 參數: MQCACF\_SUB\_NAME  
追蹤層次: 2  
類型: MQCFST  
範圍: 只有在 MQSD.SubName 的 VSLength 欄位大於零或 MQVS\_NULL\_TERMINATED 時才會併入。  
長度: 長度會改變。

### **SubUserData**

說明: 訂閱使用者資料  
PCF 參數: MQCACF\_SUB\_USER\_DATA  
追蹤層次: 2  
類型: MQCFST  
範圍: 只有在 MQSD.SubName 的 VSLength 欄位大於零或 MQVS\_NULL\_TERMINATED 時才會併入。  
長度: 長度會改變。

### **SubCorrelId**

說明: 訂閱相關性 ID  
PCF 參數: MQBACF\_SUB\_CORREL\_ID  
追蹤層次: 2  
類型: MQCFBS  
長度: MQ\_CORREL\_ID\_LENGTH

### **SelectionString**

說明:	Selection 字串。
PCF 參數:	MQCACF_SELECTION_STRING
追蹤層次:	2
類型:	MQCFST
範圍:	只有在 MQSD 的 VSLength 欄位時才包含。SelectionString 是 MQVS_NULL_TERMINATED 或大於零。
長度:	長度會改變。

### **ResolvedQName**

說明:	當 ResolvedType 為 MQOT_Q 時, ObjectHandle 所參照的佇列名稱。
PCF 參數:	MQCACF_RESOLVED_LOCAL_Q_NAME
追蹤層次:	2
類型	MQCFST
長度:	MQ_Q_NAME_LENGTH。

### **ResObjectString**

說明:	當 ResolvedType 是 MQOT_TOPIC 時, ObjectHandle 所參照的物件名稱。
PCF 參數:	MQCACF_RESOLVED_OBJECT_STRING
追蹤層次:	2
類型	MQCFST
長度:	長度會改變。

### **ResolvedType**

說明:	ObjectHandle 所參照的物件類型。可能的值為 MQOT_Q、MQOT_TOPIC 或 MQOT_NONE。
PCF 參數:	MQIACF_RESOLVED_TYPE
追蹤層次:	2
類型	MQCFIN

### **SubDescriptorStructure**

說明:	MQSD 結構。
PCF 參數:	MQBACF_MQSD_STRUCT
追蹤層次:	3
類型:	MQCFBS
長度:	MQSD 結構的長度 (以位元組為單位)。

### **QMgrOpDuration**

說明:	佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。 持續時間不包括在佇列管理程式之外所花費的時間。例如, 作為 IBM MQ 用戶端所花費的時間。 <b>註:</b> 此計時器的精確度會根據您企業使用的平台而有所不同。
-----	---

PCF 參數: MQIAMO64\_QMGR\_OP\_DURATION  
追蹤層次: 2  
類型 MQCFIN64

#### *MQSUBRQ*

應用程式已啟動 MQSUBRQ MQI 函數

#### **CompCode**

說明: 指出作業結果的完成碼  
PCF 參數: MQIACF\_COMP\_CODE  
追蹤層次: 1  
類型: MQCFIN

#### **Reason**

說明: 作業的原因碼結果  
PCF 參數: MQIACF\_REASON\_CODE  
追蹤層次: 1  
類型: MQCFIN

#### **SubHandle**

說明: 訂閱控點  
PCF 參數: MQIACF\_HSUB  
追蹤層次: 1  
類型: MQCFIN

#### **SubOptions**

說明: MQSB.Options  
PCF 參數: MQIACF\_SUBRQ\_OPTIONS  
追蹤層次: 2  
類型: MQCFIN

#### **Action**

說明: 訂閱要求動作 (MQSR\_\*)  
PCF 參數: MQIACF\_SUBRQ\_ACTION  
追蹤層次: 2  
類型: MQCFIN

#### **NumPubs**

說明: 由於此呼叫而傳送的發佈數 (來自 MQSB.NumPubs)  
PCF 參數: MQIACF\_NUM\_PUBS  
追蹤層次: 2  
類型: MQCFIN

### **QMGrOpDuration**

說明：佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。  
持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。

註：此計時器的精確度會根據您企業使用的平台而有所不同。

PCF 參數：MQIAMO64\_QMGR\_OP\_DURATION

追蹤層次：2

類型：MQCFIN64

### **MQSTAT**

應用程式已啟動 MQSTAT MQI 函數

### **CompCode**

說明：指出作業結果的完成碼

PCF 參數：MQIACF\_COMP\_CODE

追蹤層次：1

類型：MQCFIN

### **Reason**

說明：作業的原因碼結果

PCF 參數：MQIACF\_REASON\_CODE

追蹤層次：1

類型：MQCFIN

### **Type**

說明：所要求的狀態資訊類型

PCF 參數：MQIACF\_STATUS\_TYPE

追蹤層次：2

類型：MQCFIN

### **StatusStructure**

說明：MQSTS 結構。

PCF 參數：MQBACF\_MQSTS\_STRUCT

追蹤層次：3

類型：MQCFBS

長度：MQSTS 結構的長度 (以位元組為單位) (實際大小視結構版本而定)

### **QMGrOpDuration**

說明：佇列管理程式內大約的 API 呼叫持續時間 (以微秒為單位)。

持續時間不包括在佇列管理程式之外所花費的時間。例如，作為 IBM MQ 用戶端所花費的時間。

註：此計時器的精確度會根據您企業使用的平台而有所不同。

PCF 參數:	MQIAM064_QMGR_OP_DURATION
追蹤層次:	2
類型	MQCFIN64

## 應用程式活動 XA 作業的變數參數

XA 作業是應用程式可以進行的 API 呼叫，可讓 MQ 參與交易。每一個作業的參數定義在下一節中。

追蹤層次指出要併入追蹤中的參數所需的追蹤精度層次。可能的追蹤層次值為：

### 1. 低

當針對應用程式配置 "low"、"medium" 或 "high" activity tracing 時，會包含這個參數。此設定表示參數一律包含在作業的 AppActivityData 群組中。這組參數足以追蹤應用程式所建立的 MQI 呼叫，並查看它們是否成功。

### 2. 中

只有在為應用程式配置 "中" 或 "高" 活動追蹤時，此參數才會包含在作業的 AppActivityData 群組中。這組參數會新增資源的相關資訊，例如應用程式所使用的佇列及主題名稱。

### 3. 高

只有在針對應用程式配置 "高" 活動追蹤時，才會將此參數包含在作業的 AppActivityData 群組中。這組參數包括傳遞至 MQI 及 XA 函數之結構的記憶體傾出。基於此原因，它包含 MQI 及 XA 呼叫中所使用參數的相關資訊。結構記憶體傾出是結構的淺層副本。為了避免錯誤嘗試取消參照指標，結構中的指標值會設為 NULL。

**註：**傾出的結構版本不一定與應用程式使用的版本相同。可以透過 API 交叉結束程式、活動追蹤程式碼或佇列管理程式來修改結構。佇列管理程式可以將結構修改為更新版本，但佇列管理程式絕不會將它變更為舊版結構。如果這樣做，將有遺失資料的風險。

## AXREG

應用程式已啟動 AXREG ax 函數

### XID

說明:	XID 結構
PCF 參數:	MQBACF_XA_XID
追蹤層次:	1
類型:	MQCFBS
長度:	Sizeof (XID)

### Rmid

說明:	資源管理程式 ID
PCF 參數:	MQIACF_XA_RMID
追蹤層次:	1
類型:	MQCFIN

### Flags

說明:	旗標
PCF 參數:	MQ IACF_XA_FLAGS
追蹤層次:	1
類型:	MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### **AXUNREG**

應用程式已啟動 AXUNREG AX 函數

### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### **Flags**

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### **XACLOSE**

應用程式已啟動 XACLOSE AX 函數

### **Xa\_info**

說明: 用來起始設定資源管理程式的資訊。  
PCF 參數: MQ CACF\_XA\_INFO  
追蹤層次: 1  
類型: MQCFST

### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### **Flags**

說明: 旗標

PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### ***XARetCode***

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### ***XACOMMIT***

應用程式已啟動 XACOMMIT AX 函數

### ***XID***

說明: XID 結構  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

### ***Rmid***

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### ***Flags***

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### ***XARetCode***

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### ***XACOMplete***

應用程式已啟動 XACOMplete AX 函數

### ***Handle***

說明: 非同步作業的控點  
PCF 參數: MQIACF\_XA\_HANDLE

追蹤層次: 1  
類型: MQCFIN

### **Retval**

說明: 非同步函數的回覆值  
PCF 參數: MQIACF\_XA\_RETVAL  
追蹤層次: 1  
類型: MQCFINMQCFBS

### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### **Flags**

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### **XAEND**

應用程式已啟動 XAEND AX 函數

### **XID**

說明: XID 結構  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### **Flags**

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### **XAFORGET**

應用程式已啟動 AXREG ax 函數

### **XID**

說明: XID 結構  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### **Flags**

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### **XAOPEN**

應用程式已啟動 XAOPEN AX 函數

### ***Xa\_info***

說明： 用來起始設定資源管理程式的資訊。  
PCF 參數: MQ CACF\_XA\_INFO  
追蹤層次: 1  
類型: MQCFST

### ***Rmid***

說明： 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### ***Flags***

說明： 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### ***XARetCode***

說明： 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### ***XAPREPARE***

應用程式已啟動 XAPREPARE AX 函數

### ***XID***

說明： XID 結構  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

### ***Rmid***

說明： 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### ***Flags***

說明： 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS

追蹤層次: 1  
類型: MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

### **XARECOVER**

應用程式已啟動 XARECOVER AX 函數

### **Count**

說明: XID 計數  
PCF 參數: MQIACF\_XA\_COUNT  
追蹤層次: 1  
類型: MQCFIN

### **XIDs**

說明: XID 結構  
註: 此 PCF 參數有多個實例-每個 XID 結構有一個實例, 直到「計數 XID」為止  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

### **Flags**

說明: 旗標  
PCF 參數: MQIACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1

類型: MQCFIN

#### *XAROLLBACK*

應用程式已啟動 XAROLLBACK AX 函數

##### **XID**

說明: XID 結構  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

##### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1  
類型: MQCFIN

##### **Flags**

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

##### **XARetCode**

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

#### *XASTART*

應用程式已啟動 XASTART AX 函數

##### **XID**

說明: XID 結構  
PCF 參數: MQBACF\_XA\_XID  
追蹤層次: 1  
類型: MQCFBS  
長度: Sizeof (XID)

##### **Rmid**

說明: 資源管理程式 ID  
PCF 參數: MQIACF\_XA\_RMID  
追蹤層次: 1

類型: MQCFIN

### Flags

說明: 旗標  
PCF 參數: MQ IACF\_XA\_FLAGS  
追蹤層次: 1  
類型: MQCFIN

### XARetCode

說明: 回覆碼  
PCF 參數: MQIACF\_XA\_RETCODE  
追蹤層次: 1  
類型: MQCFIN

## Multi 監視及活動追蹤的系統主題

佇列管理程式主題樹狀結構中的系統主題用於資源監視 (其中部分類似於統計資料訊息的內容), 並作為耗用應用程式活動追蹤的方式。

### \$SYS/MQ 佇列管理程式主題樹狀結構的分支

每一個佇列管理程式的主題樹狀結構都包含 \$SYS/MQ 分支。佇列管理程式會發佈至此分支中的主題字串。授權使用者可以訂閱這些主題字串, 以接收佇列管理程式及其活動的相關資訊。這些系統主題用於應用程式活動追蹤及監視。如需相關資訊, 請參閱 [主題樹狀結構](#)。

\$SYS/MQ 分支的根目錄由 SYSTEM.ADMIN.TOPIC 主題物件。主題樹狀結構的 \$SYS/MQ 分支會以下列方式與主題樹狀結構的其餘部分隔離:

- 在樹狀結構中高於 \$SYS/MQ 的點使用萬用字元所建立的訂閱, 不符合 \$SYS/MQ 分支內的任何主題字串。SYSTEM.ADMIN.TOPIC 設為 "區塊", 且無法修改。當您在 `runmqsc` 指令 `DISPLAY TPSTATUS` 來顯示主題樹狀結構中的節點時, 也適用此限制。如果要檢視 \$SYS/MQ 分支內的主題節點, 請以 \$SYS/MQ 作為主題字串的開頭。例如, 使用 `$SYS/MQ/#` 來查看所有節點。
- 您必須在 \$SYS/MQ 獲得授權, 或獲得比 \$SYS/MQ 主題樹狀結構更深入的權限, 才能使用 \$SYS/MQ 主題樹狀結構。訂閱主題字串的授權是根據在主題樹狀結構中授與受管理主題物件的授權, 或高於主題字串的授權。以 root 使用者身分 (SYSTEM.BASE.TOPIC) 會授與使用者對所有主題字串的權限。不過, 如果是 \$SYS/MQ 分支, 則授與高於 \$SYS/MQ 的存取權不適用於 \$SYS/MQ 主題字串。
- 主題樹狀結構的 \$SYS/MQ 分支與樹狀結構中較高的主題屬性集隔離。SYSTEM.ADMIN.TOPIC 不會從主題樹狀結構中定義較高的主題物件繼承任何屬性。例如, 變更 SYSTEM.BASE.TOPIC 不會影響 \$SYS/MQ 分支的行為。

以 \$SYS/MQ 開頭的所有主題字串都保留給 IBM MQ 使用。這些主題字串具有下列限制:

- 您無法從主題樹狀結構的 \$SYS/MQ 分支啟用多重播送。
- \$SYS/MQ 分支不支援叢集作業。
- Proxy 訂閱機制無法設為 "force"。
- 應用程式無法發佈至 \$SYS/MQ 主題字串。
- 發佈及訂閱範圍僅預設為本端佇列管理程式。
- 下列點不能使用萬用字元:
  - \$SYS/MQ/
  - \$SYS/MQ/INFO
  - \$SYS/MQ/INFO/QMGR

- \$SYS/MQ/INFO/QMGR/queue\_manager\_name
- \$SYS/MQ/INFO/QMGR/queue\_manager\_name/ActivityTrace

嘗試在這些點上使用萬用字元會導致訂閱失敗，原因為 MQRC\_ADMIN\_TOPIC\_STRING\_ERROR。

## 透過發佈/訂閱提供的效能監視 API

從 IBM MQ 9.0 開始，您可以使用簡式發佈/訂閱機制及 amqsrua 範例應用程式或您自己的應用程式，來監視佇列管理程式統計資料。統計資料會發佈至 \$SYS/MQ/INFO/QMGR 下的系統主題，以協助使用者監視資源。您可以執行 amqsrua 範例應用程式，或以類似 **amqsrua** 的方式撰寫訂閱資源監視系統主題的應用程式，來檢視這些統計資料。

### 相關概念

第 180 頁的『應用程式活動追蹤』

應用程式活動追蹤會產生連接至佇列管理程式之應用程式行為的詳細資訊。它會追蹤應用程式的行為，並在應用程式與 IBM MQ 資源互動時提供應用程式所使用參數的詳細視圖。它也會顯示應用程式所發出的 MQI 呼叫順序。

Windows

Linux

## 使用 amqsrua 指令來監視系統資源使用情形

您可以使用 **amqsrua** 指令來查詢與佇列管理程式的系統資源使用相關的效能資料。

### 關於這項作業

amqsrua 範例應用程式展示一種使用 IBM MQ 監視發佈的方式，並顯示佇列管理程式所發佈的效能資料。此資料可以包括 CPU、記憶體及磁碟用量的相關資訊。您也可以查看與 STATMQI PCF 統計資料相等的資料。資料每 10 秒發佈一次，並在指令執行時報告。

您可以只使用佇列管理程式名稱來執行指令，並以互動方式逐步執行，以選擇 **CLASS**，然後在每一個步驟中選擇佇列管理程式可用的 **TYPE** 及 **object** 參數。如果您知道要查看其資訊的 **CLASS**、**TYPE** 及 **object** 名稱，則可以在執行 **amqsrua** 指令時指定它們。

V 9.1.0

依預設，amqsrua 應用程式會在主題樹狀結構 \$SYS/MQ/INFO/QMGR 下尋找佇列管理程式所發佈的統計資料。其他元件或應用程式可以使用類似機制，在不同的主題起點下發佈。例如，x86-64 Linux 平台上可用的 IBM MQ Bridge to Salesforce，會在 \$SYS/Application/runmqsfb 下發佈統計資料。從 IBM MQ 9.1.0 開始，您可以使用 **-p** 參數來指定 amqsrua 在 Linux 及 Windows 上尋找這些其他元件的統計資料。

**-m**

佇列管理程式名稱。佇列管理程式必須在執行中。如果您未指定佇列管理程式名稱，則會顯示預設佇列管理程式的統計資料。

**-c**

類別名稱。IBM MQ 資源使用發佈與類別相關聯。類別代表 meta 資料樹狀結構中說明可用資源使用資訊的最上層。

**CPU**

傳回 CPU 使用率的相關資訊。

**磁碟**

傳回磁碟用量的相關資訊。

**STATMQI**

傳回 MQI 使用情形的相關資訊。

**STATQ**

傳回每個佇列 MQI 使用的相關資訊。

V 9.1.5

**STATAPP**

傳回所指定應用程式的使用情形統計資料相關資訊。

如需如何指定 meta 資料主題樹狀結構的相關資訊，請參閱第 264 頁的『開發您自己的資源監視應用程式』，以及在支援的程式設計語言中使用應用程式名稱，以取得您可以在應用程式名稱中使用的字元。

註: 您可以使用 **-o** 選項來指定 STATAPP 正在監視的應用程式名稱, 以及 STATQ 的佇列名稱。

**V 9.1.0** 如需僅適用於 IBM MQ Bridge to Salesforce 的類別, 請參閱第 269 頁的『監視 IBM MQ Bridge to Salesforce』。

**-t**

TYPE 名稱。IBM MQ 資源使用發佈與類別內的類型相關聯。每一個出版品都包含類別和類型, 可讓您找到 class/type/element 定義, 並處理產生的出版品。在佇列管理程式啟動時, 會將 class/type/element 說明發佈為 meta 資料。

**-o**

物件名稱。資源使用發佈是由一連串 PCF 元素所組成的 PCF 訊息。針對每一個類別/類型配對發佈的 PCF 元素會在 meta 資料中公告。葉節點儲存在樹狀結構中, 用於說明每一個元素, 從而容許處理元素。

**V 9.1.0 -p**

meta 資料字首。指定主題樹狀結構起始點, 讓 **amqsrua** 可以在其中尋找佇列管理程式所發佈的統計資料。預設主題樹狀結構是 \$SYS/MQ/INFO/QMGR, 但其他元件或應用程式可能會在不同的主題樹狀結構起始點下發佈統計資料, 例如, IBM MQ Bridge to Salesforce 會在 \$SYS/Application/runmqsfb 下發佈統計資料。

**-n**

發佈計數。您可以指定在指令結束之前傳回多少報告。資料大約每 10 秒發佈一次, 因此如果您輸入值 50, 則指令會在 500 秒內傳回 50 個報告。如果您未指定此參數, 則會執行指令, 直到發生錯誤或佇列管理程式關閉為止。

**V 9.1.0.10 -s**

模型佇列。選擇性地指定要使用的模型佇列。(依預設, **amqsrua** 會使用 SYSTEM.DEFAULT.MODEL.QUEUE 佇列。)

**-h**

使用情形

## 程序

1. 從 samples 目錄發出下列指令, 以顯示佇列管理程式的可用資料:

- Linux 在 Linux 上, `MQ_INSTALLATION_PATH/samp/bin:`

```
./amqsrua -m QMgrName
```

- Windows 在 Windows 上, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64:`

```
amqsrua -m QMgrName
```

其中 *QMgrName* 指定您要查詢的佇列管理程式名稱。佇列管理程式必須在執行中。如果您未指定佇列管理程式名稱, 則會使用預設佇列管理程式。

可用選項如下:

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. 從 CLASS 選項清單中, 輸入 STATMQI。

```
==> STATMQI
CONNDISC : MQCONN and MQDISC
OPENCLOSE : MQOPEN and MQCLOSE
INQSET : MQINQ and MQSET
PUT : MQPUT
GET : MQGET
SYNCPPOINT : Commit and rollback
```

```
SUBSCRIBE : Subscribe
PUBLISH : Publish
Enter Type selection
==>
```

3. 從 TYPE 選項清單中，輸入 PUT。

```
==>PUT
Publication received PutDate:20170329 PutTime:17045485 Interval:4 minutes,13.978 seconds
Interval total MQPUT/MQPUT1 count 22
Interval total MQPUT/MQPUT1 byte count 25284 100/sec
Non-persistent message MQPUT count 22
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 25284 100/sec
Put persistent messages - byte count 0
MQSTAT count 0

Publication received PutDate:20170329 PutTime:17050485 Interval:10.001 seconds
Interval total MQPUT/MQPUT1 count 1
Interval total MQPUT/MQPUT1 byte count 524 52/sec
Non-persistent message MQPUT count 1
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 524 52/sec
Put persistent messages - byte count 0
MQSTAT count 0
```

## 結果

您以互動方式使用 **amqsrua** 範例應用程式，來查看佇列管理程式在 meta 資料字首 \$SYS/MQ/INFO/QMGR 下的系統主題上發佈的統計資料。

註：佇列管理程式可用的資源發佈項目類別及類型可能不同，視其配置、版本及平台而定。以互動方式使用 **amqsrua** 來尋找可用於特定佇列管理程式的類別、類型及元素。

## 下一步

若要開發您自己的監視應用程式，請參閱 [第 264 頁的『開發您自己的資源監視應用程式』](#)。

### 相關工作

第 269 頁的『[監視 IBM MQ Bridge to Salesforce](#)』

您可以使用 **amqsrua** 範例 C 程式來訂閱主題，並顯示 IBM MQ Bridge to Salesforce 所使用佇列管理程式的發佈資訊。

在 Windows 上準備及執行範例程式

在 UNIX 及 Linux 上準備及執行範例程式

## 開發您自己的資源監視應用程式

您可以開發自己的應用程式來監視系統資源。

每一個佇列管理程式都會將資源使用情形資料發佈至主題。這些主題的訂閱者會耗用此資料。當佇列管理程式啟動時，佇列管理程式會在 meta 主題上發佈一組訊息。這些訊息說明佇列管理程式支援哪些資源使用主題，以及發佈至那些主題的訊息內容。管理工具可以訂閱 meta 資料，以探索可用的資源使用資訊，以及哪些主題，然後訂閱所通告的主題。

meta 資料的主題樹狀結構具有下列結構：

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

以釐清您正在指定受監視的資源 (應用程式名稱上限為 28 個字元)，並避免與內部 meta 資料類型屬性混淆。

如需可能的類別清單，請參閱第 262 頁的『使用 amqsrua 指令來監視系統資源使用情形』。

amqsrua 應用程式的原始碼作為 IBM MQ 範例提供。您可以使用此範例應用程式作為指引，以建立您自己的監視應用程式。您可以從 IBM MQ 用戶端安裝架構擷取範例的來源。原始檔名稱為 amqsruaa.c，位於 samples 目錄中：

- **Linux** **UNIX** 在 UNIX 和 Linux 平台上: `MQ_INSTALLATION_PATH/samp/`
- **Windows** 在 Windows 平台上: `MQ_INSTALLATION_PATH\tools\c\Samples\`

amqsrua 應用程式會訂閱 IBM MQ 資源使用主題，並將產生的已發佈 PCF 資料格式化。應用程式來源提供一個基本範例，說明如何訂閱及耗用這種類型的管理資料。amqsrua 應用程式完成下列作業：

- 對輸入參數所識別的主題建立不可延續訂閱。
- 反覆地呼叫 MQGET 以從主題取得訊息，並寫入 stdout。
- 針對每一個 MQI 原因 (非 MQRC\_NONE) 寫入訊息。
- 如果 MQI 完成碼為 MQCC\_FAILED，或已耗用所要求的資源使用發佈數，則會停止。

## Multi 在系統主題上發佈的度量值

度量會分類為類別，並再分類為類型。在每一個度量類別和類型下發佈了各種度量。

### index

- [第 265 頁的『CPU \(平台中央處理裝置\)』](#)
- [第 265 頁的『DISK \(平台持續資料儲存庫\)』](#)
- [第 266 頁的『STATMQI \(API 使用情形統計資料\)』](#)
- [第 268 頁的『STATQ \(每個佇列的 API 使用情形統計資料\)』](#)

## CPU (平台中央處理裝置)

### SystemSummary (CPU 效能-平台層面)

使用者 CPU 時間百分比 X%  
 系統 CPU 時間百分比 X%  
 CPU 負載-一分鐘平均值 X  
 CPU 負載-五分鐘平均值 X  
 CPU 負載-15 分鐘平均值 X  
 RAM 可用百分比 X%  
 RAM 位元組總數 XMB

### QMGrSummary (CPU 效能-執行中佇列管理程式)

使用者 CPU 時間-佇列管理程式 X% 的百分比預估  
 系統 CPU 時間-佇列管理程式 X% 的百分比估計值  
 RAM 位元組總數-佇列管理程式 XMB 的估計值

## DISK (平台持續資料儲存庫)

### SystemSummary (磁碟用量-整個平台)

MQ 錯誤檔案系統-使用中位元組數 XMB

MQ 錯誤檔案系統-可用空間 X%  
MQ FDC 檔案計數 X  
MQ 追蹤檔案系統-使用中的位元組數 XMB  
MQ 追蹤檔案系統-可用空間 X%

#### **QMGrSummary (磁碟用量-執行中佇列管理程式)**

佇列管理程式檔案系統-使用中的位元組數 XMB  
佇列管理程式檔案系統-可用空間 X%

#### **日誌 (磁碟使用情形-佇列管理程式回復日誌)**

日誌-使用中的位元組 X  
日誌-位元組數上限 X  
日誌檔系統-使用中的位元組 X  
日誌檔系統-位元組數上限 X  
日誌-寫入的實體位元組 X  
日誌-寫入的邏輯位元組數 X  
日誌寫入延遲 X uSec

**V 9.1.0** 日誌寫入大小 X

**V 9.1.0** 日誌-等待保存的延伸範圍所佔用的位元組數 X

**V 9.1.0** 日誌-媒體回復所需的位元組數 X

**V 9.1.0** 日誌-可重複使用的延伸範圍 X 所佔用的位元組數

**V 9.1.0** 日誌-現行主要空間使用中 X%

**V 9.1.0** 日誌工作量主要空間使用率 X%

#### **STATMQI (API 使用情形統計資料)**

##### **CONDISC (MQCONN 及 MQDISC)**

MQCONN/MQCONNX 計數 X  
失敗 MQCONN/MQCONNX 計數 X  
並行連線-高臨界值 X  
MQDISC 計數 X

##### **OPENCLOSE (MQOPEN 和 MQCLOSE)**

MQOPEN 計數 X Y/秒  
失敗 MQOPEN 計數 X  
MQCLOSE 計數 X Y/秒  
失敗 MQCLOSE 計數 X

##### **INQSET (MQINQ 和 MQSET)**

MQINQ 計數 X  
失敗 MQINQ 計數 X  
MQSET 計數 X  
失敗 MQSET 計數 X

##### **PUT (MQPUT)**

間隔總計 MQPUT/MQPUT1 計數 X  
間隔總計 MQPUT/MQPUT1 位元組計數 X Y/秒  
非持續訊息 MQPUT 計數 X  
持續訊息 MQPUT 計數 X

失敗 MQPUT 計數 X  
非持續訊息 MQPUT1 計數 X  
持續訊息 MQPUT1 計數 X  
失敗 MQPUT1 計數 X  
放置非持續訊息-位元組計數 X Y/秒  
放置持續訊息-位元組計數 X  
MQSTAT 計數 X

#### **GET (MQGET)**

間隔破壞性取得總計-計數 X  
間隔總破壞性取得-位元組計數 X Y/秒  
非持續訊息破壞性取得-計數 X  
持續訊息破壞性取得-計數 X  
失敗 MQGET-計數 X  
取得非持續訊息-位元組計數 X Y/秒  
取得持續訊息-位元組計數 X  
非持續訊息瀏覽-計數 X  
持續訊息瀏覽-計數 X  
失敗瀏覽計數 X  
非持續訊息瀏覽-位元組計數 X Y/秒  
持續訊息瀏覽-位元組計數 X  
過期訊息計數 X  
已清除佇列計數 X  
MQCB 計數 X  
失敗 MQCB 計數 X  
MQCTL 計數 X

#### **SYNCPOINT (確定及回復)**

確定計數 X  
回復計數 X

#### **SUBSCRIBE (訂閱)**

建立可延續訂閱計數 X  
變更可延續訂閱計數 X  
回復可延續訂閱計數 X  
建立不可延續訂閱計數 X  
建立/變更/回復訂閱計數 X 失敗  
刪除可延續訂閱計數 X  
刪除不可延續訂閱計數 X  
訂閱刪除失敗計數 X  
MQSUBRQ 計數 X  
失敗 MQSUBRQ 計數 X  
可延續訂閱者-高臨界值 X  
可延續訂閱者-低臨界值 X  
不可延續訂閱者-高臨界值 X  
不可延續訂閱者-低臨界值 X

**PUBLISH (publish)**

主題 MQPUT/MQPUT1 間隔總計 X  
間隔主題放置位元組總數 X Y/秒  
已發佈至訂閱者-訊息計數 X  
已發佈至訂閱者-位元組計數 X  
非持續性-主題 MQPUT/MQPUT1 計數 X  
持續性主題 MQPUT/MQPUT1 計數 X  
失敗主題 MQPUT/MQPUT1 計數 X

**STATQ (每個佇列的 API 使用情形統計資料)****OPENCLOSE (MQOPEN 和 MQCLOSE)**

MQOPEN 計數 X  
MQCLOSE 計數 X

**INQSET (MQINQ 和 MQSET)**

MQINQ 計數 X  
MQSET 計數 X

**PUT (MQPUT 和 MQPUT1)**

MQPUT/MQPUT1 計數 X  
MQPUT 位元組計數 X  
MQPUT 非持續訊息計數 X  
MQPUT 持續訊息計數 X  
MQPUT1 非持續訊息計數 X  
MQPUT1 持續訊息計數 X  
非持續性位元組計數 X  
持續位元組計數 X  
鎖定競用 X%  
佇列已避免放置 X%  
佇列已避免位元組 X%

**GET (MQGET)**

MQGET 計數 X  
MQGET 位元組計數 X  
破壞性 MQGET 非持續訊息計數 X  
破壞性 MQGET 持續訊息計數 X  
破壞性 MQGET 非持續性位元組計數 X  
破壞性 MQGET 持續性位元組計數 X  
MQGET 瀏覽非持續訊息計數 X  
MQGET 瀏覽持續訊息計數 X  
MQGET 瀏覽非持續性位元組計數 X  
MQGET 瀏覽持續位元組計數 X  
訊息過期 X  
佇列已清除計數 X  
平均佇列時間 X uSec  
佇列深度 X

您可以使用 **amqsrua** 範例 C 程式來訂閱主題，並顯示 IBM MQ Bridge to Salesforce 所使用佇列管理程式的發佈資訊。

## 關於這項作業

在佇列管理程式啟動時，佇列管理程式會將一組關於本身的訊息發佈至 meta 資料字首 **\$SYS/MQ/INFO/QMGR** 下的系統主題。這些訊息指出哪些效能資料可供佇列管理程式使用。管理工具可以訂閱主題來探索可用的資源使用情形資訊，以便它們可以耗用資源使用情形資料。如需相關資訊，請參閱 [第 262 頁的『使用 amqsrua 指令來監視系統資源使用情形』](#)。

依預設，**amqsrua** 程式會在主題樹狀結構 **\$SYS/MQ/INFO/QMGR** 下尋找佇列管理程式所發佈的統計資料。其他元件或應用程式可以使用類似機制，在不同的主題起點下發佈。例如，x86-64 Linux 平台上可用的 IBM MQ Bridge to Salesforce，會在 **\$SYS/Application/runmqsfb** 下發佈統計資料。

您可以使用 **-p** 參數來指定 **amqsrua** 在 Linux 及 Windows 上尋找這些其他元件的統計資料的位置。

您可以只使用佇列管理程式名稱來執行指令，並以互動方式逐步執行，以選擇 **CLASS**，然後在每一個步驟中選擇佇列管理程式可用的 **TYPE** 及 **object** 參數。如果您知道要查看其資訊的 **CLASS**、**TYPE** 及 **object** 名稱，則可以在執行 **amqsrua** 指令時指定它們。

### -m

佇列管理程式名稱。佇列管理程式必須在執行中。如果您未指定佇列管理程式名稱，則會顯示預設佇列管理程式的統計資料。

### -c

類別名稱。IBM MQ 資源使用發佈與類別相關聯。類別代表 meta 資料樹狀結構中說明可用資源使用資訊的最上層。

您只能使用 **CHANNEL** 和 **STATUS** 類別來查看與 IBM MQ Bridge to Salesforce 搭配使用的佇列管理程式所收到的發佈資訊。當您執行 **amqsrua** 指令時，必須以 meta 資料字首 **"\ \$SYS/Application/runmqsfb"** 指定 **-p** 參數，以查看 **CHANNEL** 和 **STATUS** 類別的資訊。

### STATUS

傳回 IBM MQ Bridge to Salesforce 的狀態相關資訊。

### CHANNEL

傳回 Salesforce 事件的相關資訊。

### -t

TYPE 名稱。IBM MQ 資源使用發佈與類別內的類型相關聯。每一個出版品都包含類別和類型，可讓您找到 class/type/element 定義，並處理產生的出版品。在佇列管理程式啟動時，會將 class/type/element 說明發佈為 meta 資料。

當您針對與 IBM MQ Bridge to Salesforce 搭配使用的佇列管理程式執行 **amqsrua** 程式時，可以使用 **STATUS** 及 **CHANNEL** 類別。

適用於 **STATUS** 類別。

### Pushtopic

Salesforce 推送主題。

### 事件

Salesforce 平台事件。

### MQPE

使用此選項，您可以查看 IBM MQ 所建立平台事件的相關資訊。當您選取此選項時，會顯示下列資料：

- 在一個間隔內 (在 STATUS/MQPE 樹狀結構下) 處理的 IBM MQ 已建立平台事件總數。
- 在此間隔中看到唯一數目的 IBM MQ 已建立平台事件。
- 在此間隔中看到的 IBM MQ 已建立平台事件發佈數失敗。

適用於 **CHANNEL** 類別。

## 事件

特定 Salesforce 推送主題或平台事件的發佈總數。您需要指定類別 (CHANNEL) 類型 (EVENTS) 的物件名稱，例如 /topic/push\_topic\_name 或 /event/platform\_event\_name。

-o

物件名稱。資源使用發佈是由一連串 PCF 元素所組成的 PCF 訊息。針對每一個類別/類型配對發佈的 PCF 元素會在 meta 資料中公告。葉節點儲存在樹狀結構中，用於說明每一個元素，從而容許處理元素。

-p

meta 資料字首。指定主題樹狀結構起始點，讓 **amqsrua** 可以在其中尋找佇列管理程式所發佈的統計資料。預設主題樹狀結構是 \$SYS/MQ/INFO/QMGR，但其他元件或應用程式可能會在不同的主題樹狀結構起始點下發佈統計資料，例如，IBM MQ Bridge to Salesforce 會在 \$SYS/Application/runmqsfb 下發佈統計資料。

-n

發佈計數。您可以指定在指令結束之前傳回多少報告。資料大約每 10 秒發佈一次，因此如果您輸入值 50，則指令會在 500 秒內傳回 50 個報告。如果您未指定此參數，則會執行指令，直到發生錯誤或佇列管理程式關閉為止。

-h

使用情形

## 程序

1. 切換至 samples 目錄 `MQ_INSTALLATION_PATH/samp/bin`。
2. 發出帶有 **-p** 參數的 **amqsrua** 指令，以指向佇列管理程式在其中發佈 IBM MQ Bridge to Salesforce 推送主題及平台事件監視資訊的 "`\$SYS/Application/runmqsfb`" 主題：

```
./amqsrua -m QM1 -p "\$SYS/Application/runmqsfb"
```

```
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==>
```

3. 輸入類別選擇 CHANNEL

```
==> CHANNEL
```

```
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> CHANNEL
EVENTS : Events
Enter Type selection
==>
```

4. 輸入類型選項 EVENTS

```
==> EVENTS
```

```
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> CHANNEL
EVENTS : Events
Enter Type selection
==> EVENTS
An object name is required for Class(CHANNEL) Type(EVENTS)
Enter object name
==>
```

5. 輸入物件名稱，例如

```
==> /topic/push_topic_name
```

```
STATUS : Bridge Status  
CHANNEL : Salesforce Event Name  
Enter Class selection  
==> CHANNEL  
EVENTS : Events  
Enter Type selection  
==> EVENTS  
An object name is required for Class(CHANNEL) Type(EVENTS)  
Enter object name  
==> /topic/push_topic_name  
Publication received PutDate:20170215 PutTime:14513762 Interval:15.047 seconds  
Total Events 1
```

## 結果

您已透過輸入類別、類型及物件選項，以互動方式使用 `amqsrua` 程式，以查看所指定推送主題的事件總數。

在此範例中，您可以使用 `amqsrua` 來檢查 Salesforce 推送主題的數目：

```
./amqsrua -m QM1 -p "$SYS/Application/runmqsfb"  
STATUS : Bridge Status  
CHANNEL : Salesforce Event Name  
Enter Class selection  
==> STATUS  
PUSHTOPIC : Push Topics  
EVENTS : Platform Events  
Enter Type selection  
==> PUSHTOPIC  
Publication received PutDate:20170315 PutTime:10123086 Interval:15.016 seconds  
Total PushTopics 6  
Unique PushTopics 3  
  
Publication received PutDate:20170315 PutTime:10124588 Interval:15.012 seconds  
Total PushTopics 14 1/sec  
Unique PushTopics 7
```

在此範例中，您可以使用 `amqsrua` 來檢查 IBM MQ 建立的平台事件數目：

```
./amqsrua -m QM1 -p "$SYS/Application/runmqsfb"  
STATUS : Bridge Status  
CHANNEL : Salesforce Event Name  
Enter Class selection  
==> STATUS  
PUSHTOPIC : Pushtopics  
EVENTS : Platform Events  
MQPE : MQ-created Platform Events  
Enter Type selection  
==> MQPE  
Publication received PutDate:20170913 PutTime:12201271 Interval:15.018 seconds  
Total MQ-created Platform Events 1  
Unique MQ-created Platform Events 1  
Failed publications of MQ-created Platform Events 0
```

## 相關工作

[配置 IBM MQ 以與 Salesforce 推送主題及平台事件搭配使用](#)

[第 262 頁的『使用 amqsrua 指令來監視系統資源使用情形』](#)

您可以使用 `amqsrua` 指令來查詢與佇列管理程式的系統資源使用相關的效能資料。

[在 Windows 上準備及執行範例程式](#)

[在 UNIX 及 Linux 上準備及執行範例程式](#)

## 相關參考

[runmqsfb \(執行 IBM MQ Bridge to Salesforce\)](#)

## 即時監視

即時監視是一種技術，可讓您判定佇列管理程式內佇列及通道的現行狀態。傳回的資訊在發出指令時是正確的。

許多指令可用，當發出時，會傳回佇列及通道的相關即時資訊。可以傳回一或多個佇列或通道的資訊，且數量可能不同。即時監視可用於下列作業：

- 協助系統管理者瞭解其 IBM MQ 系統的穩定狀態。如果系統中發生問題，這有助於診斷問題。
- 隨時判斷佇列管理程式的狀況，即使未偵測到特定事件或問題。
- 協助判斷系統中問題的原因。

透過即時監視，可以傳回佇列或通道的資訊。傳回的即時資訊量是由佇列管理程式、佇列及通道屬性所控制。

- 您可以發出指令來監視佇列，以確保適當地處理佇列。在您可以使用部分佇列屬性之前，必須先啟用它們以進行即時監視。
- 您可以透過發出指令來監視通道，以確保通道適當地執行。您必須先啟用通道屬性以進行即時監視，才能使用部分通道屬性。

對佇列及通道的即時監視是除了效能及通道事件監視之外的其他項目，並且與這些效能及通道事件監視分開。

### 控制即時監視的屬性

如果啟用即時監視，則部分佇列及通道狀態屬性會保留監視資訊。如果未啟用即時監視，則這些監視屬性中不會保留任何監視資訊。範例示範如何使用這些佇列及通道狀態屬性。

您可以啟用或停用個別佇列或通道的即時監視，或多個佇列或通道的即時監視。若要控制個別佇列或通道，請設定佇列屬性 MONQ 或通道屬性 MONCHL，以啟用或停用即時監視。若要一起控制許多佇列或通道，請使用佇列管理程式屬性 MONQ 及 MONCHL 來啟用或停用佇列管理程式層次的即時監視。對於具有以預設值 QMGR 指定之監視屬性的所有佇列及通道物件，會在佇列管理程式層次控制即時監視。

自動定義的叢集傳送端通道不是 IBM MQ 物件，因此沒有與通道物件相同的屬性。若要控制自動定義的叢集傳送端通道，請使用佇列管理程式屬性 MONACLS。此屬性決定是否啟用或停用佇列管理程式內自動定義的叢集傳送端通道，以進行通道監視。

若要即時監視通道，您可以將 MONCHL 屬性設為下列三個監視層次之一：低、中或高。您可以在物件層次或佇列管理程式層次設定監視層次。層次的選擇取決於您的系統。收集監視資料可能需要一些相對昂貴的計算指示，例如取得系統時間。為了減少即時監視的影響，中及低監視選項會定期測量資料樣本，而不是一直收集資料。第 272 頁的表 30 彙總可用於即時監視通道的監視層次：

層次	說明	使用情形
低	定期測量資料的小型樣本。	適用於處理大量訊息的物件。
中	定期測量資料樣本。	適用於大部分物件。
高	定期測量所有資料。	對於每秒只處理幾則訊息的物件，最新資訊很重要。

若要即時監視佇列，您可以將 MONQ 屬性設為三個監視層次的其中一個：低、中或高。不過，這些值之間沒有區別。這些值都會啟用資料收集，但不會影響樣本大小。

### 範例

下列範例示範如何設定必要的佇列、通道及佇列管理程式屬性，以控制監視層次。在所有範例中，當啟用監視時，佇列及通道物件會有中等層次的監視。

1. 若要對佇列管理程式層次的佇列及通道啟用佇列及通道監視，請使用下列指令：

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

- 若要啟用監視所有佇列及通道，但本端佇列、Q1 及傳送端通道 QM1.TO.QM2 除外，請使用下列指令：

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

- 若要停用所有佇列及通道的佇列及通道監視，但本端佇列、Q1 及傳送端通道 QM1.TO.QM2 除外，請使用下列指令：

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

- 若要停用所有佇列及通道的佇列及通道監視，不論個別物件屬性為何，請使用下列指令：

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

- 若要控制自動定義叢集傳送端通道的監視功能，請使用下列指令：

```
ALTER QMGR MONACLS(MEDIUM)
```

- 若要指定自動定義的叢集傳送端通道將使用佇列管理程式設定進行通道監視，請使用下列指令：

```
ALTER QMGR MONACLS(QMGR)
```

## 相關概念

[第 272 頁的『即時監視』](#)

即時監視是一種技術，可讓您判定佇列管理程式內佇列及通道的現行狀態。傳回的資訊在發出指令時是正確的。

[第 291 頁的『使用 IBM MQ 線上監視』](#)

您可以設定 MONQ、MONCHL 及 MONACLS 屬性，來收集佇列及通道 (包括自動定義的叢集伺服器通道) 的監視資料。

## 相關工作

[第 273 頁的『顯示佇列及通道監視資料』](#)

若要顯示佇列或通道的即時監視資訊，請使用 IBM MQ Explorer 或適當的 MQSC 指令。部分監視欄位會顯示一對以逗點區隔的指示器值，以協助您監視佇列管理程式的作業。範例示範如何顯示監視資料。

[顯示及變更佇列管理程式屬性](#)

[監視 \(MONCHL\)](#)

## 顯示佇列及通道監視資料

若要顯示佇列或通道的即時監視資訊，請使用 IBM MQ Explorer 或適當的 MQSC 指令。部分監視欄位會顯示一對以逗點區隔的指示器值，以協助您監視佇列管理程式的作業。範例示範如何顯示監視資料。

## 關於這項作業

顯示一對以逗點區隔的值的監視欄位會提供自針對物件啟用監視以來所測量時間的短期及長期指示器，或從啟動佇列管理程式時開始：

- 短期指標是配對中的第一個值，其計算方式可讓最近的測量獲得較高的加權，並對這個值有較大的影響。這可指出所採取測量的最近趨勢。
- 配對中第二個值中的長期指標，並以某種方式計算，使得較近期的測量不會獲得如此高的加權。這會指出資源效能的長期活動。

這些指示器值最有助於偵測佇列管理程式作業中的變更。這需要瞭解這些指示器在正常使用時所顯示的時間，以便偵測這些時間的增加。透過定期收集並檢查這些值，您可以偵測佇列管理程式作業中的變動。這可以指出效能的變更。

取得即時監視資訊，如下所示：

## 程序

1. 若要顯示佇列的即時監視資訊，請使用 IBM MQ Explorer 或 MQSC 指令 DISPLAY QSTATUS，並指定選用參數 MONITOR。
2. 若要顯示通道的即時監視資訊，請使用 IBM MQ Explorer 或 MQSC 指令 DISPLAY CHSTATUS，並指定選用參數 MONITOR。

## 範例

佇列 Q1 將屬性 MONQ 設為預設值 QMGR，且擁有佇列的佇列管理程式將屬性 MONQ 設為 MEDIAL。若要顯示針對此佇列收集的監視欄位，請使用下列指令：

```
DISPLAY QSTATUS(Q1) MONITOR
```

佇列 Q1 的監視欄位及監視層次顯示如下：

```
QSTATUS(Q1)
TYPE(Queue)
MONQ(MEDIUM)
QTIME(11892157,24052785)
MSGAGE(37)
LPUTDATE(2005-03-02)
LPUTTIME(09.52.13)
LGETDATE(2005-03-02)
LGETTIME(09.51.02)
```

傳送端通道 QM1.TO.QM2 將屬性 MONCHL 設為預設值 QMGR，而擁有佇列的佇列管理程式將屬性 MONCHL 設為 MEDIAL。若要顯示針對此傳送端通道收集的監視欄位，請使用下列指令：

```
DISPLAY CHSTATUS(QM1.TO.QM2) MONITOR
```

傳送端通道 QM1.TO.QM2 的監視欄位及監視層次顯示如下：

```
CHSTATUS(QM1.TO.QM2)
XMITQ(Q1)
CONNNAME(127.0.0.1)
CURRENT
CHLTYPE(SDR)
STATUS(RUNNING)
SUBSTATE(MQGET)
MONCHL(MEDIUM)
XQTIME(755394737,755199260)
NETTIME(13372,13372)
EXITTIME(0,0)
XBATCHSZ(50,50)
COMPTIME(0,0)
STOPREQ(NO)
RQMNAME(QM2)
```

## 相關概念

第 272 頁的『即時監視』

即時監視是一種技術，可讓您判定佇列管理程式內佇列及通道的現行狀態。傳回的資訊在發出指令時是正確的。

## 相關參考

[DISPLAY QSTATUS](#)

## 監視佇列

請利用這個頁面來檢視作業，這些作業可協助您解決佇列及服務該佇列之應用程式的問題。有各種監視選項可用來判斷問題

通常，正在處理的佇列問題的第一個徵兆是佇列上的訊息數 (CURDEPTH) 增加。如果您預期在一天中的特定時間或在特定工作量下增加，則越來越多的訊息可能不會指出問題。不過，如果您沒有訊息數目增加的說明，則可能想要調查原因。

您可能會有應用程式發生問題的應用程式佇列，或通道發生問題的傳輸佇列。當服務佇列的應用程式是通道時，可以使用其他監視選項。

下列範例調查特定佇列 (稱為 Q1) 的問題，並說明您在各種指令輸出中查看的欄位：

## 判斷應用程式是否已開啟佇列

如果佇列有問題，請檢查您的應用程式是否已開啟佇列

## 關於這項作業

請執行下列步驟來判斷您的應用程式是否已開啟佇列：

### 程序

1. 請確定針對佇列執行的應用程式是您預期的應用程式。針對有問題的佇列發出下列指令：

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

在輸出中，查看 APPLTAG 欄位，並檢查是否顯示應用程式的名稱。如果未顯示您應用程式的名稱，或根本沒有輸出，請啟動您的應用程式。

2. 如果佇列是傳輸佇列，請查看 CHANNEL 欄位的輸出。  
如果通道名稱未顯示在 CHANNEL 欄位中，請判斷通道是否在執行中。
3. 請確定針對佇列執行的應用程式已開啟佇列以供輸入。發出下列指令：

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

在輸出中，查看 IPPROCS 欄位，以查看是否有任何應用程式開啟佇列以供輸入。如果值為 0 且這是使用者應用程式佇列，請確定應用程式開啟佇列以供輸入，以從佇列中取得訊息。

## 檢查佇列上的訊息是否可用

如果佇列上有大量訊息，且您的應用程式未處理其中任何訊息，請檢查佇列上的訊息是否可供您的應用程式使用

## 關於這項作業

請執行下列步驟，以調查應用程式未處理來自佇列的訊息的原因：

### 程序

1. 當您的應用程式應該處理佇列上的所有訊息時，請確定它沒有要求特定的訊息 ID 或相關性 ID。
2. 雖然佇列的現行深度可能顯示佇列上的訊息數目增加，但佇列上的部分訊息可能無法由應用程式取得，因為它們未確定；現行深度包括佇列中未確定的訊息 MQPUT 數。發出下列指令：

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

在輸出中，查看 UNCOM 欄位，以查看佇列上是否有任何未確定的訊息。

3. 如果您的應用程式嘗試從佇列取得任何訊息，請檢查放置應用程式是否正確地確定訊息。發出下列指令，以找出將訊息放入此佇列的應用程式名稱：

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) OPENTYPE(OUTPUT)
```

4. 然後發出下列指令，在 *appltag* 中插入前一個指令的輸出中的 APPLTAG 值：

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

這會顯示工作單元何時啟動，並協助您探索應用程式是否正在建立長時間執行的工作單元。如果放置應用程式是通道，您可能想要調查批次完成花費較長時間的原因。

## 檢查應用程式是否正在從佇列中取得訊息

如果佇列及服務該佇列的應用程式有問題，請檢查您的應用程式是否正在從佇列中取得訊息

### 關於這項作業

若要檢查應用程式是否正在從佇列中取得訊息，請執行下列檢查：

### 程序

1. 請確定針對佇列執行的應用程式實際上正在處理來自佇列的訊息。發出下列指令：

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

在輸出中，查看 LGETDATE 及 LGETTIME 欄位，這些欄位會顯示前次從佇列執行取得的時間。

2. 如果前次從這個佇列取得的時間比預期長，請確定應用程式正在正確處理訊息。  
如果應用程式是通道，請檢查訊息是否正在透過該通道移動

## 判斷應用程式是否可以足夠快地處理訊息

如果在佇列上建置訊息，但您的其他檢查未發現任何處理問題，請檢查應用程式是否可以足夠快地處理訊息。如果應用程式是通道，請檢查通道處理訊息的速度是否足夠快。

### 關於這項作業

如果要判斷應用程式處理訊息的速度是否足夠快，請執行下列測試：

### 程序

1. 定期發出下列指令，以收集佇列的相關效能資料：

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

如果 QTIME 指示器中的值偏高或在期間內不斷增加，且您已透過檢查佇列上的訊息是否可用，來排除長時間執行「工作單元」的可能性，則取得的應用程式可能無法跟上放置的應用程式。

2. 如果您的取得應用程式無法跟上放置應用程式的進度，請考量新增另一個取得應用程式來處理佇列。

您是否可以新增另一個取得應用程式，取決於應用程式的設計，以及多個應用程式是否可以共用佇列。訊息分組或依相關性 ID 取得之類的特性可能有助於確保兩個應用程式可以同時處理佇列。

## 當現行深度未增加時檢查佇列

即使佇列的現行深度沒有增加，監視佇列以檢查應用程式是否正確處理訊息仍可能很有用。

### 關於這項作業

若要收集佇列的相關效能資料：請定期發出下列指令：

### 程序

定期發出下列指令：

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

在輸出中，如果 MSGAGE 中的值在一段時間內增加，且您的應用程式設計為處理所有訊息，這可能表示完全未處理部分訊息。

## 監視頻道

請利用這個頁面來檢視作業，這些作業可協助您解決傳輸佇列及服務該佇列之通道的問題。有各種通道監視選項可用來判斷問題。

通常，正在處理的佇列問題的第一個徵兆是佇列上的訊息數 (CURDEPTH) 增加。如果您預期在一天中的特定時間或在特定工作量下增加，則越來越多的訊息可能不會指出問題。不過，如果您沒有訊息數目增加的說明，則可能想要調查原因。

您可能會有處理傳輸佇列的通道問題。有各種通道監視選項可協助您判斷問題。

下列範例調查稱為 QM2 的傳輸佇列及稱為 QM1.TO.QM2。此通道用來將訊息從佇列管理程式 QM1 傳送至佇列管理程式 QM2。佇列管理程式 QM1 的通道定義是傳送端或伺服器通道，而佇列管理程式 QM2 的通道定義是接收端或要求端通道。

## 判斷通道是否在執行中

如果傳輸佇列有問題，請檢查通道是否在執行中。

## 關於這項作業

執行下列步驟，以檢查處理傳輸佇列之通道的狀態：

## 程序

1. 發出下列指令，以找出您預期處理傳輸佇列 QM2: 的通道：

```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```

在此範例中，此指令的輸出顯示處理傳輸佇列的通道是 QM1.TO.QM2

2. 發出下列指令，以判定通道 QM1.TO.QM2:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

3. 檢查 **CHSTATUS** 指令輸出的 STATUS 欄位：
  - 如果 STATUS 欄位的值為 RUNNING，請檢查通道是否正在移動訊息
  - 如果指令的輸出未顯示狀態，或 STATUS 欄位的值為 STOPPED、RETRY、BINDING 或 REQUESTING，請執行適當的步驟，如下所示：
4. 選擇性的: 如果 STATUS 欄位的值未顯示狀態，則通道為非作用中，因此請執行下列步驟：
  - a) 如果通道應該已由觸發程式自動啟動，請檢查傳輸佇列上的訊息是否可用。  
如果傳輸佇列上有可用的訊息，請檢查傳輸佇列上的觸發設定是否正確。
  - b) 發出下列指令，以手動重新啟動通道：

```
START CHANNEL(QM1.TO.QM2)
```

5. 選擇性的: 如果 STATUS 欄位的值是 STOPPED，請執行下列步驟：
  - a) 請檢查錯誤日誌，以判斷通道停止的原因。如果通道因錯誤而停止，請更正問題。  
另請確定通道具有指定給重試屬性的值: *SHORTRTY* 及 *LONGRTY*。如果發生暫時性失敗 (例如網路錯誤)，則通道會嘗試自動重新啟動。
  - b) 發出下列指令，以手動重新啟動通道：

```
START CHANNEL(QM1.TO.QM2)
```

 在 IBM MQ for z/OS 上，您可以使用指令事件訊息來偵測使用者何時停止通道。

6. 選擇性的: 如果 STATUS 欄位的值是 RETRY，請執行下列步驟:
  - a) 請檢查錯誤日誌以識別錯誤，然後更正問題。
  - b) 發出下列指令，以手動重新啟動通道:

```
START CHANNEL(QM1.TO.QM2)
```

或 會等待通道在下次重試時順利連接。

7. 選擇性的: 如果 STATUS 欄位的值是 BINDING 或 REQUESTING，表示通道尚未順利連接至夥伴。執行下列步驟:
  - a) 在通道兩端發出下列指令，以判定通道的子狀態:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

註:

- i) 在某些情況下，可能只有通道一端有子狀態。
  - ii) 許多子狀態都是暫時性的，因此請多次發出指令來偵測通道是否停留在特定子狀態。
- b) 勾選 第 278 頁的表 31 以判定要採取的動作:

起始 MCA 子狀態 <sup>1</sup>	回應 MCA 子狀態 <sup>2</sup>	附註
NameServer		起始 MCA 正在等待名稱伺服器要求完成。請確定已在通道屬性 CONNAME 中指定正確的主機名稱，且已正確設定您的名稱伺服器。
SCYEXIT	SCYEXIT	MCA 目前透過安全結束程式在交談中。如需相關資訊，請參閱 第 280 頁的『判斷通道是否可以足夠快地處理訊息』。
	CHADEXIT	通道自動定義結束程式目前正在執行中。如需相關資訊，請參閱第 280 頁的『判斷通道是否可以足夠快地處理訊息』。
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	在 MQXR_INIT 通道啟動時呼叫結束程式。如果這需要很長時間，請檢閱結束程式此部分中的處理程序。如需相關資訊，請參閱第 280 頁的『判斷通道是否可以足夠快地處理訊息』。
序列化	序列化	此子狀態僅適用於處置方式為 SHARED 的通道。
NETCONNECT		如果由於網路配置不正確而導致連接延遲，則會顯示此子狀態。
SSL 信號交換	SSL 信號交換	TLS 信號交換由許多傳送和接收組成。如果網路時間緩慢，或查閱 CRL 的連線緩慢，這會影響執行信號交換所花費的時間。  在 IBM MQ for z/OS 上，此子狀態也可能表示沒有足夠的 SSLTASKS。

附註:

- i) 起始 MCA 是啟動交談之通道的結尾。這可以是傳送端、叢集傳送端、完整伺服器及要求端。在伺服器-要求端配對中，這是您從中啟動通道的結束。

- ii) 回應 MCA 是回應啟動交談之要求的通道結尾。這可以是接收端、叢集接收端、要求端 (當伺服器或傳送端啟動時)、伺服器 (當要求端啟動時) 及傳送端 (在通道的要求端-傳送端回呼配對中)。

## 檢查通道是否正在移動訊息

如果傳輸佇列有問題，請檢查通道是否在移動訊息

### 開始之前

發出指令 DIS CHSTATUS(QM1.TO.QM2) ALL。如果 STATUS 欄位的值為 RUNNING，則表示通道已順利連接至夥伴系統。

檢查傳輸佇列上沒有未確定的訊息，如第 275 頁的『[檢查佇列上的訊息是否可用](#)』中所述。

### 關於這項作業

如果有訊息可供通道取得及傳送，請執行下列檢查：

### 程序

1. 在顯示通道狀態指令 DIS CHSTATUS(QM1.TO.QM2) ALL 的輸出中，查看下列欄位：

#### MSGS

此階段作業期間 (自啟動通道以來) 已傳送或接收到的訊息數 (或者針對伺服器連線通道所處理的 MQI 呼叫數)。

#### BUFSENT

傳送的傳輸緩衝區數量。這只包括傳送控制資訊的傳輸。

#### BYTSENT

此階段作業期間 (自啟動通道以來) 傳送的位元組數。這包括訊息通道代理程式傳送的控制資訊。

#### LSTMSGDA

傳送最後一則訊息或處理 MQI 呼叫的日期，請參閱 LSTMSGTI。

#### LSTMSGTI

傳送最後一則訊息或處理 MQI 呼叫的時間。如果是傳送端或伺服器，這是傳送最後一個訊息 (如果它被分割的話則為最後一個部分) 的時間。如果是要求端或接收端，其為最後一個訊息置入其目標佇列的時間。如果是伺服器連線通道，其為最後 MQI 呼叫完成時的時間。

#### CURMSGS

如果是傳送端通道，則這是在現行批次中傳送的訊息數。如果是接收端通道，則這是在現行批次中接收的訊息數。當確定批次時，會將傳送端通道及接收端通道的值重設為零。

#### STATUS

通道的狀態，可以是 Starting、Binding、Initializing、Running、Stopping、Retrying、Paused、Stopped 或 Requesting。

#### SUBSTATE

通道目前執行的動作。

#### INDOUBT

目前通道是否無法確定。當傳送之「訊息通道代理程式」在等待一批已傳送的批次訊息已順利接收的確認通知時，這個參數只為 YES。其他所有時間為 NO，其中包括傳送訊息的週期期間，但要在要求認可之前。如果是接收端通道，這個值一律為 NO。

2. 判斷通道自啟動後是否已傳送任何訊息。如果已傳送任何訊息，請判斷傳送最後一則訊息的時間。
3. 通道可能已啟動尚未完成的批次，如 CURMSGS 中的非零值所指示。如果 INDOUBT 是 YES，通道正在等待接收通道另一端接收批次的確認通知。查看輸出中的 SUBSTATE 欄位，請參閱第 279 頁的表 32：

傳送者 SUBSTATE	接收端 SUBSTATE	附註
MQGET	接收	靜止通道的正常狀態。
傳送	接收	SEND 通常是暫時狀態。如果看到 SEND，則表示通訊協定緩衝區已填滿。這可能指出網路問題。

表 32: 傳送端和接收端 MCA 子狀態 (繼續)		
傳送者 SUBSTATE	接收端 SUBSTATE	附註
接收		如果看到傳送端處於 RECEIVE 子狀態達任何時間長度，表示它正在等待批次完成或活動訊號的回應。您可能想要檢查批次完成花費較長時間的原因。

註: 您可能想要判斷通道是否可以足夠快地處理訊息，尤其是通道具有與結束處理相關聯的子狀態時。

## 檢查批次完成花費較長時間的原因

批次可能需要較長時間才能完成的原因包括慢速網路或通道正在使用訊息重試處理。

## 關於這項作業

當傳送端通道已傳送批次訊息時，它會等待接收端對該批次的確認，除非通道已進行管線化。此作業中說明的因素可能會影響傳送端通道等待的時間。

## 程序

- 請檢查網路是否慢速。  
NETTIME 值是將批次結束要求傳送至通道遠端，並接收回應減去處理批次結束要求時間所花費的時間量 (以微秒為單位顯示)。由於下列任一原因，此值可能較大：
  - 網路速度緩慢。慢速網路可能會影響完成批次所需的時間。產生 NETTIME 欄位指示器的測量值會在批次結束時進行測量。不過，受網路減緩影響的第一個批次不會以 NETTIME 值的變更指出，因為它是在批次結束時測量。
  - 要求會在遠端系統排入佇列，例如通道可能重試放置，或放置要求可能因為頁集 I/O 而變慢。一旦完成任何排入佇列的要求，即會測量批次要求結束的持續時間。因此，如果您取得較大的 NETTIME 值，請檢查遠端系統的異常處理程序。
- 請檢查通道是否正在使用訊息重試。  
如果接收端通道無法將訊息放置到目標佇列，它可能會使用訊息重試處理，而不是立即將訊息放置到無法傳送郵件的佇列。重試處理可能會導致批次變慢。在 MQPUT 嘗試之間，通道將具有 STATUS (PAUSED)，指出它正在等待訊息重試間隔通過。

## 判斷通道是否可以足夠快地處理訊息

如果傳輸佇列上正在建置訊息，但您未發現任何處理問題，請判斷通道是否可以足夠快地處理訊息。

## 開始之前

在一段時間內反覆地發出下列指令，以收集通道的相關效能資料：

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

## 關於這項作業

確認傳輸佇列上沒有未確定的訊息，如第 275 頁的『檢查佇列上的訊息是否可用』中所述，然後檢查顯示通道狀態指令輸出中的 XQTIME 欄位。當 XQTIME 指示器的值持續高，或在測量期間增加時，表示通道不符合放置應用程式的速度。

執行下列測試：

## 程序

- 請檢查結束程式是否正在處理中。  
如果在遞送這些訊息的通道上使用結束程式，則它們可能會增加處理訊息所花費的時間。若要識別是否發生這種情況，請執行下列檢查：
  - 在指令 DIS CHSTATUS(QM1.TO.QM2) ALL 的輸出中，檢查 EXITTIME 欄位。

如果結束程式所花費的時間高於預期，請檢閱結束程式中的處理程序，以進行任何不必要的迴圈或額外處理，尤其是在訊息、傳送及接收結束程式中。這類處理會影響在通道中移動的所有訊息。

- b) 在指令 DIS CHSTATUS(QM1.TO.QM2) ALL 的輸出中，檢查 SUBSTATE 欄位。

如果通道在相當長時間內具有下列其中一個子狀態，請檢閱結束程式中的處理程序：

- SCYEXIT
- RCVEXIT
- SENDEXIT
- MSGEXIT
- MREXIT

如需通道子狀態的相關資訊，請參閱表格 [第 279 頁的表 32](#)。

2. 請檢查網路是否慢速。

如果訊息在通道中移動的速度不夠快，可能是因為網路速度緩慢。若要識別是否發生這種情況，請執行下列檢查：

- a) 在指令 DIS CHSTATUS(QM1.TO.QM2) ALL 的輸出中，檢查 NETTIME 欄位。

當傳送端通道要求其夥伴提供回應時，會測量這些指示器。這會在每一個批次結束時，以及在活動訊號期間通道閒置時發生。

- b) 如果此指示器顯示來回時間比預期長，請使用其他網路監視工具來調查網路效能。

3. 請檢查通道是否正在使用壓縮。

如果通道使用壓縮，這會增加處理訊息所花費的時間。如果通道只使用一個壓縮演算法，請執行下列檢查：

- a) 在指令 DIS CHSTATUS(QM1.TO.QM2) ALL 的輸出中，檢查 COMPTIME 欄位。

這些指示器顯示壓縮或解壓縮期間所花費的時間。

- b) 如果選擇的壓縮未依預期的數量減少要傳送的資料量，請變更壓縮演算法。

4. 如果通道使用多個壓縮演算法，請執行下列檢查：

- a) 在指令 DIS CHSTATUS(QM1.TO.QM2) ALL 的輸出中，檢查 COMPTIME、COMPHDR 及 COMPMMSG 欄位。

- b) 請變更通道定義上指定的壓縮演算法，或考量寫入訊息結束程式，以置換通道對特定訊息的壓縮演算法選項 (如果壓縮率或演算法選項未提供必要的壓縮或效能)。

## 解決叢集通道的問題

如果您在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 佇列，診斷問題的首要步驟是探索哪個或哪些通道在遞送訊息時發生問題。

## 關於這項作業

使用 SYSTEM.CLUSTER.TRANSMIT.QUEUE 在遞送訊息時發生問題。執行下列檢查：

## 程序

1. 發出下列指令：

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

**註：**如果您的忙碌叢集有許多訊息在移動中，請考慮以較高的數目發出這個指令，以排除只有少數訊息可遞送的通道。

2. 查看 XQMSGSA 欄位中具有較大值的一或多個通道的輸出。請判斷通道未移動訊息或移動訊息的速度不夠快的原因。請使用 [第 277 頁的『監視頻道』](#) 中所概述的作業，來診斷發現導致建置的通道問題。

在 IBM WebSphere MQ 7.0 及更早版本中，可以使用 Windows 效能監視器來監視 Windows 系統上本端佇列的效能。從 IBM WebSphere MQ 7.1，此效能監視方法不再可用。

您可以使用 [第 272 頁的『即時監視』](#) 中說明的方法來監視所有受支援平台上的佇列。

## 監視叢集

在叢集內，您可以監視應用程式訊息、控制訊息及日誌。當叢集在佇列的兩個以上實例之間進行負載平衡時，有一些特殊監視考量。

### 監視叢集中的應用程式訊息

通常，離開佇列管理程式的所有叢集訊息都會通過 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`，而不論使用哪個叢集傳送端通道來傳輸訊息。每一個通道都會與所有其他叢集傳送端通道平行排除該通道的目標訊息。在此佇列上不斷增加的訊息建置可能指出一或多個通道有問題，必須加以調查：

- 必須針對叢集設計適當地監視佇列深度。
- 下列指令會傳回在傳輸佇列上等待多個訊息的所有通道：

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

對於單一佇列上的所有叢集訊息，當通道開始填滿時，並不總是容易看到哪個通道有問題。使用此指令是查看哪個通道負責的簡單方法。

您可以將叢集佇列管理程式配置成具有多個傳輸佇列。如果您將佇列管理程式屬性 `DEFCLXQ` 變更為 `CHANNEL`，則每個叢集傳送端通道都會與不同的叢集傳輸佇列相關聯。或者，您可以手動配置個別傳輸佇列。若要顯示與叢集傳送端通道相關聯的所有叢集傳輸佇列，請執行下列指令：

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

定義叢集傳輸佇列，以便它們遵循左側佇列名稱的固定詞幹型樣。然後，您可以使用通用佇列名稱來查詢 **DISPLAY CLUSMGR** 指令所傳回所有叢集傳輸佇列的深度：

```
DISPLAY QUEUE (qname *) CURDEPTH
```

### 監視叢集中的控制訊息

`SYSTEM.CLUSTER.COMMAND.QUEUE` 佇列用於處理佇列管理程式的所有叢集控制訊息，這些訊息由本端佇列管理程式產生，或從叢集中的其他佇列管理程式傳送至這個佇列管理程式。當佇列管理程式正確維護其叢集狀態時，此佇列會趨向零。不過，在某些狀況下，此佇列上的訊息深度可能會暫時增加：

- 佇列上有大量訊息表示叢集狀態流失。
- 進行重大變更時，容許佇列在這些變更之間安定下來。例如，移動儲存庫時，在移動第二個儲存庫之前，容許佇列達到零。

當此佇列上存在訊息待辦事項時，不會處理叢集狀態或叢集相關指令的更新。如果長時間未從此佇列中移除訊息，則需要進行進一步調查，起始方式是檢驗佇列管理程式錯誤日誌 (或 `CHINIT z/OS` 上的日誌)，這可能會說明造成此狀況的處理程序。

`SYSTEM.CLUSTER.REPOSITORY.QUEUE` 會將叢集儲存庫快取資訊保留為一些訊息。訊息通常一律存在於此佇列中，而更多訊息則適用於較大的叢集。因此，此佇列上的訊息深度不是值得關注的問題。

### 監視日誌

在最初發生問題之後，由於快取資訊及叢集作業的分散式本質，叢集中發生的問題可能在許多天 (甚至數個月) 內不會對應用程式顯示外部症狀。不過，通常會在 IBM MQ 錯誤日誌 (以及 `z/OS` 上的 `CHINIT` 日誌) 中

報告原始問題。基於此原因，請務必主動監視這些日誌中所撰寫的任何與叢集作業相關的訊息。必須閱讀並瞭解這些訊息，並在必要時採取任何動作。

例如：與叢集中佇列管理程式的通訊中斷可能會導致瞭解由於叢集透過重新發佈資訊而定期重新驗證叢集資源的方式而被刪除的特定叢集資源。AMQ9465 或 CSQX465I (在 z/OS 系統上) 訊息會報告可能發生此類事件的警告。此訊息指出需要調查問題。

## 負載平衡的特殊考量

當叢集在佇列的兩個以上實例之間進行負載平衡時，消費端應用程式必須正在處理每一個實例上的訊息。如果一或多個消費端應用程式終止或停止處理訊息，叢集作業可能會繼續將訊息傳送至那些佇列實例。在此狀況下，除非應用程式再次正確運作，否則不會處理這些訊息。因此，監視應用程式是解決方案的重要部分，在該狀況下必須採取動作來重新遞送訊息。在此範例中可以找到自動化此類監視的機制範例：[The Cluster Queue Monitoring sample program \(AMQSCLM\)](#)。

### 相關概念

第 328 頁的『調整分散式發佈/訂閱網路』

請使用本節中的調整提示來協助改善 IBM MQ 分散式發佈/訂閱叢集和階層的效能。

第 332 頁的『在發佈/訂閱網路中平衡生產者和消費者』

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡，否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

## 監視傳輸佇列切換

請務必監視叢集傳送端通道切換傳輸佇列的處理程序，以便將對企業的影響降到最低。例如，當工作量偏高或同時切換多個通道時，您不應嘗試此處理程序。

## 切換通道的程序

用來切換通道的處理程序為：

1. 通道會開啟新的傳輸佇列來進行輸入，並開始從其中取得訊息 (使用依相關性 ID 取得)
2. 佇列管理程式會起始一個背景處理程序，將通道已排入佇列的任何訊息從其舊傳輸佇列移至其新傳輸佇列。當訊息移動時，通道的任何新訊息都會排入舊傳輸佇列，以保留排序。如果通道在其舊傳輸佇列上有大量訊息，或新訊息快速送達，則此處理程序可能需要一些時間才能完成。
3. 如果沒有任何已確定或未確定的訊息仍在其舊傳輸佇列上的通道佇列中，則交換器會完成。新的訊息現在會直接放置到新的傳輸佇列中。

為了避免多種通道同步切換的可能性，IBM MQ 可讓您切換一或多個未執行的通道的傳輸佇列。在：

- IBM MQ for Multiplatforms 指令稱為 **runswch1**
- IBM MQ for z/OS 可以改用 CSQUTIL 公用程式來處理 SWITCH CHANNEL 指令

## 監視交換器作業的狀態

若要瞭解交換器作業的狀態，管理者可以執行下列動作：

- 監視佇列管理程式錯誤日誌 (AMQERR01.LOG) 其中會輸出訊息，以指出作業期間的下列階段：
  - 已啟動切換作業
  - 已開始移動訊息
  - 定期更新要移動的訊息數 (如果切換作業未快速完成)
  - 已完成移動訊息
  - 交換器作業已完成

在 z/OS 上，這些訊息會輸出至佇列管理程式工作日誌，而不是通道起始程式工作日誌；不過，如果通道在啟動時起始切換，則通道會將單一訊息輸出至通道起始程式工作日誌。

- 請使用 DISPLAY CLUSQMGR 指令來查詢每一個叢集傳送端通道目前使用的傳輸佇列。

- 以查詢模式執行 **runswchl** 指令 (或 z/OS 上的 CSQUTIL)，以確定一個以上通道的切換狀態。此指令的輸出會識別每一個通道的下列項目：
  - 通道是否有擱置中的切換作業
  - 通道從哪一個傳輸佇列切換至哪一個傳輸佇列
  - 舊傳輸佇列上剩餘的訊息數

每一個指令都非常有用，因為在一次呼叫中，您可以判斷每一個通道的狀態、配置變更所造成的影響，以及所有交換器作業是否都已完成。

## 可能發生的潛在問題

請參閱 [切換傳輸佇列時的潛在問題](#)，以取得切換傳輸佇列時可能遇到的一些問題、其原因，以及最可能的解決方案。

### 相關概念

第 328 頁的『調整分散式發佈/訂閱網路』

請使用本節中的調整提示來協助改善 IBM MQ 分散式發佈/訂閱叢集和階層的效能。

第 332 頁的『在發佈/訂閱網路中平衡生產者和消費者』

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡，否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

## V 9.1.3 Multi 監視應用程式平衡

如何使用 **DISPLAY APSTATUS** 指令來監視統一叢集中應用程式平衡的狀態，以及調查應用程式未平衡的原因 (如果這是非預期的)。

### 在叢集中跨佇列管理程式監視應用程式的現行狀態

從統一叢集中的任何佇列管理程式，您可以執行 [DIS APSTATUS](#) 指令來取得叢集中所有佇列管理程式的應用程式現行狀態概觀。

例如，在剛啟動佇列管理程式之後，您可能會看到如下所示的輸出：

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(UNKNOWN)
```

V 9.1.4 在 IBM MQ 9.1.4 中，**TYPE** 欄位也會顯示在輸出中：

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)
```

這顯示統一叢集中有一個稱為 MYAPP 的應用程式，目前有八個實例，這八個實例都被視為可在統一叢集周圍移動。UNKNOWN 的平衡值是暫時值，指出佇列管理程式尚未嘗試在必要時重新平衡應用程式。

在短時間之後，您更有可能看到下列輸出：

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(NO)
```

#### V 9.1.4

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                       MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

此輸出顯示應用程式有八個實例，但它們未在統一叢集之間平衡。此時，值得查看叢集中應用程式的分佈。

若要執行此動作，請重新執行 **DIS APSTATUS** 指令。請注意，您可以針對統一叢集中的任何佇列管理程式執行此指令：

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                       MOVCCOUNT(6)
BALSTATE(HIGH)                  LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)             QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)             QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)             QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
```

#### V 9.1.4

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                       MOVCCOUNT(6)
BALSTATE(HIGH)                  LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)             QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)             QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)             QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
```

從此您可以在此時看到佇列管理程式 UNID001 有六個實例，但佇列管理程式 UNID0002 和 UNID0003 只有一個實例。BALSTATE 輸出會指出佇列管理程式前次報告它的平衡狀態。不過，請注意實例計數可能比 BALSTATE 欄位更新。

此輸出也指出統一叢集正在配送此應用程式的相關資訊。統一叢集中的所有佇列管理程式實例都知道應用程式，且最後一則訊息的日期和時間非常晚。

此外，「作用中」欄位指出叢集中的所有佇列管理程式都視為彼此通訊；如果在任何佇列管理程式上「作用中」設為「否」，則會指出與它的一般通訊已中斷。

如果留下重新平衡本身，輸出最終將會出現如下的結果：

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                       MOVCCOUNT(3)
```

```

BALSTATE(OK)                                LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)                          QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                             ACTIVE(YES)
COUNT(3)                                    MOVCCOUNT(3)
BALSTATE(OK)                                LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)                          QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                             ACTIVE(YES)
COUNT(2)                                    MOVCCOUNT(2)
BALSTATE(OK)                                LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)                          QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)

```

## V 9.1.4

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                             ACTIVE(YES)
COUNT(3)                                    MOVCCOUNT(3)
BALSTATE(OK)                                LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)                          QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                             ACTIVE(YES)
COUNT(3)                                    MOVCCOUNT(3)
BALSTATE(OK)                                LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)                          QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                             ACTIVE(YES)
COUNT(2)                                    MOVCCOUNT(2)
BALSTATE(OK)                                LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)                          QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

佇列管理程式的 BALSTATE 現在是「正常」，表示已達到穩定狀態。

## 監視個別應用程式實例

您可以查看個別應用程式實例，但會針對每一個佇列管理程式執行此動作。查看 UNID001:

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                                     IMMREASN(NONE)
IMMCOUNT(0)                                  IMMDATE( )
IMMTIME( )                                   MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                                     IMMREASN(NONE)
IMMCOUNT(0)                                  IMMDATE( )
IMMTIME( )                                   MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                                     IMMREASN(NONE)
IMMCOUNT(0)                                  IMMDATE( )
IMMTIME( )                                   MOVABLE(YES)
One MQSC command read.

```

## V 9.1.4

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                                     IMMREASN(NONE)
IMMCOUNT(0)                                  IMMDATE( )
IMMTIME( )                                   MOVABLE(YES)
AMQ8932I: Display application status details.

```

```

APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1) IMMREASN(NONE)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(YES)

```

AMQ8932I: Display application status details.

```

APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1) IMMREASN(NONE)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(YES)
TYPE (LOCAL)

```

One MQSC command read.

存在可移動 (YES) 表示必要的話，可以將此實例移至叢集中的另一個佇列管理程式。在下列範例中，應用程式實例無法移動，因為它未以用戶端身分連接：

```

3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)

```

AMQ8932I: Display application status details.

```

APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)

```

#### V 9.1.4

```

3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)

```

AMQ8932I: Display application status details.

```

APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

```

您可以使用 **CONNTAG** 欄位來查看來自該實例的個別佇列管理程式連線，這在應用程式實例是 IMMREASN (NOTRECONN) 時非常有用。在下列範例中，用戶端應用程式不可移動，且調查連接選項顯示它在 **CONNOPTS** 欄位中沒有 MQCNO\_RECONNECT。

```

1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)

```

AMQ8932I: Display application status details.

```

APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)

```

```

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL

```

AMQ8276I: Display Connection details.

```

CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE(CONN)
PID(14656) TID(20)
APPLDESC(IBM MQ Channel) APPLTAG(ClientApp)
APPLTYPE(USER) ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN) CLIENTID( )
CONNAME(127.0.0.1)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserid) UOWLOG( )
UOWSTDA( ) UOWSTTI( )
UOWLOGDA( ) UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0) UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)

```

#### V 9.1.4

```

1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)

```

AMQ8932I: Display application status details.

```

APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)

```

```

IMMCOUNT(0)                IMMDATE( )
IMMTIME( )                 MOVABLE(NO)
TYPE (LOCAL)

      2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE (CONN)
PID(14656)                  TID(20)
APPLDESC(IBM MQ Channel)   APPLTAG(ClientApp)
APPLTYPE(USER)             ASTATE(NONE)
CHANNEL (SYSTEM.DEF.SVRCONN) CLIENTID( )
CONNNAME (127.0.0.1)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserid)           UOWLOG( )
UOWSTDA( )                 UOWSTTI( )
UOWLOGDA( )                UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0)                UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
TYPE (CONN)

```

## 相關概念

[自動應用程式平衡](#)

## z/OS 監視效能及資源使用情形

請使用本主題來瞭解可用來監視 IBM MQ for z/OS 子系統的效能及資源使用情形的機能。

### 相關工作

[正在配置 z/OS](#)

[管理 IBM MQ for z/OS](#)

## z/OS 監視簡介

請使用本主題作為 IBM MQ for z/OS 可用的監視機能概觀。例如，使用 IBM MQ 追蹤、線上監視及事件來取得 Snapshot。

本主題說明如何監視 IBM MQ 的效能及資源使用情形。

- 它概述您可以擷取的部分資訊，並簡要說明調查效能問題的一般方法。
- 它概述您可以擷取的部分資訊，並簡要說明調查效能問題的一般方法。如需相關資訊，請參閱 [第 294 頁的『調查效能問題』](#)。
- 它說明如何使用 SMF 記錄來收集 IBM MQ 效能的相關統計資料。
- 它說明如何收集帳戶資料，讓您可以向客戶收取使用 IBM MQ 系統的費用。
- 它說明如何使用 IBM MQ 事件 (警示) 來監視系統。

以下是您可以用來監視 IBM MQ 的部分工具; 下列各節會說明這些工具:

- IBM MQ 提供的工具:
  - [使用 DISPLAY 指令](#)
  - [第 289 頁的『使用 CICS 配接器統計資料』](#)
  - [第 291 頁的『使用 IBM MQ 事件』](#)
- z/OS 服務輔助工具:
  - [第 292 頁的『使用系統管理機能』](#)
- 其他 IBM 授權程式:
  - [使用 Resource Measurement Facility](#)
  - [使用 Tivoli Decision Support for z/OS](#)

- 使用 CICS 監視機能

如需解譯效能統計資料追蹤所收集資料的相關資訊，請參閱第 294 頁的『[解譯 IBM MQ 效能統計資料](#)』。  
如需解譯帳戶追蹤所收集資料的相關資訊，請參閱第 316 頁的『[解譯 IBM MQ 帳戶資料](#)』。

**z/OS 使用 DISPLAY 指令取得 IBM MQ 的 Snapshot**

IBM MQ 提供 MQSC 機能，可提供使用 DISPLAY 指令的效能及資源使用的 Snapshot。

您可以使用 DISPLAY 指令及 CICS 配接器畫面 (若為 CICS 配接器) 來瞭解 IBM MQ 的現行狀態。

**使用 DISPLAY 指令**

您可以使用 IBM MQ MQSC DISPLAY 或 PCF Inquire 指令來取得 IBM MQ 現行狀態的相關資訊。它們提供指令伺服器、程序定義、佇列、佇列管理程式及其相關聯元件的狀態相關資訊。這些指令如下：

MQSC 指令	PCF 指令
顯示保存檔	查詢保存
DISPLAY AUTHINFO	查詢鑑別資訊物件
DISPLAY CFSTATUS	查詢 CF 結構狀態
DISPLAY CFSTRUCT	查詢 CF 結構
顯示通道	查詢通道
顯示 CHINIT	查詢通道起始程式
DISPLAY CHSTATUS	查詢通道狀態
DISPLAY CMDSERV	
DISPLAY CLUSQMGR	查詢叢集佇列管理程式
DISPLAY CONN	查詢連線
顯示群組	查詢群組
DISPLAY LOG	查詢日誌
DISPLAY PROCESS	查詢處理程序
DISPLAY QMGR	查詢佇列管理程式
DISPLAY QSTATUS	查詢佇列狀態
顯示佇列	查詢佇列
DISPLAY SECURITY	查詢安全
DISPLAY STGCLASS	查詢儲存類別
DISPLAY SYSTEM	查詢系統
顯示追蹤	
顯示使用情形	查詢使用情形

如需每一個指令的詳細語法，請參閱 [MQSC 指令](#) 或 [PCF 指令](#)。這些指令的所有功能 (DISPLAY CMDSERV 及 DISPLAY TRACE 除外) 也可以透過作業及控制面板來使用。

這些指令僅在處理指令時提供系統的 Snapshot。如果您想要檢查系統中的趨勢，則必須在一段時間內啟動 IBM MQ 追蹤並分析結果。

**z/OS 使用 CICS 配接器統計資料**

如果您是授權 CICS 使用者，則可以使用 CICS 配接卡控制台來動態顯示 CICS 配接卡統計資料。

這些統計資料提供 CICS 執行緒使用情形及所有執行緒忙碌時狀況相關資訊的 Snapshot。您可以按 Enter 鍵來重新整理顯示畫面連線畫面。如需相關資訊，請參閱 CICS Transaction Server for z/OS 4.1 產品說明文件中的 "CICS-IBM MQ 配接器" 一節，網址為 [CICS Transaction Server for z/OS 4.1](#)，即 [CICS-IBM MQ 配接器](#)。

## 使用 IBM MQ 追蹤

您可以使用 IBM MQ 追蹤機能來記錄 IBM MQ 的效能統計資料及結算資料。請利用這個主題來瞭解如何控制 IBM MQ 追蹤。

IBM MQ 產生的資料會傳送至：

- 系統管理機能 (SMF)，特別是作為效能統計資料追蹤的 SMF 記錄類型 115、子類型 1 和 2
- SMF，特別是作為 SMF 記錄類型 116，帳戶追蹤的子類型為零、1 和 2。

如果您喜歡，也可以將 IBM MQ 帳戶追蹤所產生的資料傳送至一般性追蹤機能 (GTF)。

## 啟動 IBM MQ 追蹤

您可以隨時發出 IBM MQ [START TRACE](#) 指令來啟動 IBM MQ 追蹤機能。

如果在應用程式執行時啟動或停止帳戶追蹤，則可能會遺失帳戶資料。若要順利收集統計資料，必須套用下列條件：

- 當應用程式啟動時，帳戶追蹤必須在作用中，當應用程式完成時，它仍必須在作用中。
- 如果帳戶追蹤已停止，則任何作用中的帳戶資料收集都會停止。

如果您在 CSQ6SYSP 巨集的 SMFSTAT (SMF STATISTICS) 及 SMFACCT (SMF ACCOUNTING) 參數上指定 YES，也可以自動開始收集部分追蹤資訊。 這些參數在 [使用 CSQ6SYSP](#) 中說明。

在啟動 IBM MQ 追蹤之前，請先閱讀 [第 292 頁的『使用系統管理機能』](#)。

## 控制 IBM MQ 追蹤

若要在啟動時控制 IBM MQ 追蹤資料收集，請在自訂 IBM MQ 時指定 CSQ6SYSP 巨集中參數的值。

 如需詳細資料，請參閱 [使用 CSQ6SYSP](#)。

當佇列管理程式使用下列指令執行時，您可以控制 IBM MQ 追蹤：

- 啟動追蹤
- ALTER TRACE
- 停止追蹤

您可以選擇追蹤資料傳送至的目的地。可能的目的地如下：

### SMF

系統管理機能

### GTF

一般性追蹤機能 (僅限帳戶追蹤)

### srv

供 IBM 服務人員使用的診斷服務程式常式

對於每日監視，資訊會傳送至 SMF (預設目的地)。SMF 資料集通常包含來自其他系統的資訊；在傾出 SMF 資料集之前，此資訊無法用於產生報告。

您也可以將帳戶追蹤資訊傳送至 GTF。此資訊具有事件 ID 5EE。 [MQI 呼叫及使用者參數](#)，以及 [z/OS 一般性追蹤機能 \(GTF\)](#) 說明如何處理傳送至 GTF 的 IBM MQ 追蹤資訊。

如需 IBM MQ 指令的相關資訊，請參閱 [MQSC 指令](#)。

## 追蹤對 IBM MQ 效能的影響

使用 IBM MQ 追蹤機能可能會對 IBM MQ 及交易效能產生重大影響。例如，如果您針對類別 1 或所有類別啟動廣域追蹤，則可能會將處理器用量及交易回應時間增加大約 50%。不過，如果您單獨對類別 2-4 啟動廣域追蹤，則處理器用量和交易回應時間的增加可能會低於 IBM MQ 呼叫成本的 1% 額外處理器成本。這也適用於統計資料或帳戶追蹤。

### 使用 IBM MQ 線上監視

您可以設定 MONQ、MONCHL 及 MONACLS 屬性，來收集佇列及通道 (包括自動定義的叢集伺服器通道) 的監視資料。

第 291 頁的表 33 彙總用於在不同層次設定這些屬性及顯示監視資訊的指令。

屬性	在此層次適用	使用指令設定	使用指令顯示監視資訊
MONQ	佇列	DEFINE QLOCAL DEFINE QMODEL ALTER QLOCAL ALTER QMODEL	DISPLAY QSTATUS
	佇列管理程式	ALTER QMGR	
MONCHL	通道	定義通道 ALTER CHANNEL	DISPLAY CHSTATUS
	佇列管理程式	ALTER QMGR	
MONACLS	佇列管理程式	ALTER QMGR	

如需這些指令的完整資料，請參閱 [MQSC 指令](#)。如需線上監視的相關資訊，請參閱 [第 5 頁的『監視 IBM MQ 網路』](#)。

### 使用 IBM MQ 事件

IBM MQ 檢測事件提供佇列管理程式中錯誤、警告及其他重要事件的相關資訊。您可以將這些事件納入您自己的系統管理應用程式中，以監視所有佇列管理程式的作業。

IBM MQ 檢測事件分為下列種類：

#### 佇列管理程式事件

這些事件與佇列管理程式內資源的定義相關。例如，應用程式嘗試將訊息放入不存在的佇列。

#### 效能事件

這些事件是資源已達到臨界條件的通知。例如，已達到佇列深度限制，或未在預先定義的時間限制內處理佇列。

#### 通道事件

通道會報告這些事件，因為在其作業期間偵測到條件。例如，通道實例已停止。

#### 配置事件

這些事件是已建立、變更或刪除物件的通知。

當事件發生時，佇列管理程式會將事件訊息放置在適當的事件佇列上 (若有定義的話)。事件訊息包含可由適當 IBM MQ 應用程式擷取之事件的相關資訊。

可以使用 IBM MQ 指令或作業及控制台來啟用 IBM MQ 事件。

如需產生訊息之 IBM MQ 事件的相關資訊，以及這些訊息格式的相關資訊，請參閱 [第 8 頁的『事件類型』](#)。如需啟用事件的相關資訊，請參閱 [事件訊息參照](#)。

## z/OS 使用系統管理機能

您可以使用 SMF 來收集統計資料及帳戶資訊。如果要使用 SMF，必須在 z/OS 和 IBM MQ 中設定某些參數。

系統管理機能 (SMF) 是 z/OS 服務輔助工具，用來從各種 z/OS 子系統收集資訊。此資訊會定期傾出及報告，例如每小時。您可以搭配使用 SMF 與 IBM MQ 追蹤機能，以從 IBM MQ 收集資料。透過此方式，您可以監視趨勢(例如，系統使用率及效能)，並使用 IBM MQ 來收集每一個使用者 ID 的相關帳戶資訊。

若要記錄效能統計資料 (記錄類型 115) 至 SMF，請在 SYS1.PARMLIB 或使用 SETSMF z/OS 操作員指令。

```
SYS(TYPE(115))
```

若要記錄 SMF 的帳戶資訊 (記錄類型 116)，請在 SYS1.PARMLIB 或使用 SETSMF z/OS 操作員指令。

```
SYS(TYPE(116))
```

若要使用 z/OS 指令 SETSMF，必須在 SMFPRM xx 成員中指定 PROMPT (ALL) 或 PROMPT (LIST)。如需相關資訊，請參閱 [z/OS MVS Initialization and Tuning Reference](#) 和 [z/OS MVS System Commands](#)。

如果您在 CSQ6SYSP 巨集的 SMFSTAT (SMF STATISTICS) 及 SMFACCT (SMF ACCOUNTING) 參數上指定 YES，則可以自動開始收集部分追蹤資訊；這在 [使用 CSQ6SYSP](#) 中有說明。

在 SMFSTAT 和 SMFACCT 參數上指定 YES 可讓您在佇列管理程式啟動時收集追蹤資訊。

您也可以在使用 **START TRACE** 指令執行佇列管理程式時，指定 START TRACE (A) 或 START TRACE (S)，開始收集資料。

您可以使用 **DEFINE QLOCAL**、**DEFINE QMODEL**、**ALTER QLOCAL**、**ALTER QMODEL** 或 **ALTER QMGR** 指令的 ACCTQ 參數，來開啟或關閉佇列或佇列管理程式層次的帳戶資訊記錄。如需這些指令的詳細資料，請參閱 [MQSC 指令](#)。

您可以使用 **DEFINE CHANNEL**、**ALTER CHANNEL** 或 **ALTER QMGR** 指令的 **STATCHL** 參數，來控制通道或佇列管理程式層次的通道結算資料收集。

您可以使用下列兩種方式之一，來指定 IBM MQ 收集統計資料及結算資料的間隔：

- 您可以在系統參數中指定 STATIME 的值 (如 [使用 CSQ6SYSP](#) 中所述)，以相同的間隔收集統計資料及結算資料。
- 您可以針對 STATIME 指定零，以 [使用發生的結算事件](#) 來收集統計資料及結算資料。

SMF 必須為執行中，您才能將資料傳送給它。如需 SMF 的相關資訊，請參閱 [z/OS MVS System Management Facilities \(SMF\)](#) 手冊。

若要重設統計資料及結算資料，在結算間隔期間必須至少發出一個 MQI 呼叫。

## 配置其他 SMF 緩衝區

當您啟動追蹤時，必須確保配置足夠的 SMF 緩衝區。在存取方法服務 **DEFINE CLUSTER** 陳述式的 VSAM BUFSP 參數上指定 SMF 緩衝。在每一個 SMF VSAM 資料集的 **DEFINE CLUSTER** 陳述式上指定 CISZ (4096) 和 BUFSP (81920)。

如果發生 SMF 緩衝區短缺，SMF 會拒絕任何傳送給它的追蹤記錄。發生此情況時，IBM MQ 會將 CSQW133I 訊息傳送至 z/OS 主控台。即使 SMF 資料可能會遺失，IBM MQ 仍會將錯誤視為暫時且保持作用中。當短缺已緩解且追蹤記錄已回復時，IBM MQ 會將 CSQW123I 訊息傳送至 z/OS 主控台。

## SMF 中的報告資料

您可以使用 SMF 程式 IFASMFDP (或 IFASMF DL (如果正在使用日誌串流))，將 SMF 記錄傾出至循序資料集，以便可以處理它們。

有數種方法可針對此資料進行報告，例如：

- 撰寫應用程式以讀取及報告 SMF 資料集的資訊。然後，您可以修改報告，以符合您的確切需求。
- 使用「效能報告程式」來處理記錄。如需相關資訊，請參閱 [第 293 頁的『搭配使用其他產品與 IBM MQ』](#)。

## 在長時間執行處理程序期間未產生間隔 CLASS (03) SMF 統計記錄

您正在收集 IBM MQ 的 CLASS (3) SMF116 帳戶記錄，但在長時間執行處理程序執行時未產生任何記錄。

CLASS (3) SMF116 統計記錄通常只有在處理程序結束時才會產生。對於長時間執行的處理程序 (例如 CICS)，這可能不會產生足夠的記錄數，因為處理程序可以執行一個月或更長時間。不過，您可能想要在處理程序執行時，以設定時間間隔收集 SMF116 記錄。

若要收集 CLASS (3) SMF116 帳戶記錄，您必須設定下列項目：

### SMFACCT

= 是

### SMFSTAT

= YES 或 NO，其中

#### YES

如果收到集合播送，則會產生記錄。

#### 否

導致您取得僅在處理程序結束時產生的 CLASS (3) SMF116 記錄

並發出下列指令：

```
START TRACE(ACCTG) DEST(SMF) CLASS(03)
```

如果您已設定 SMFSTAT = YES 且發生資料收集播送，則會針對目前正在執行且同時在前一個資料收集播送時執行的任何處理程序產生間隔 CLASS (3) SMF116 帳戶記錄。

您可以在 [CSQ6SYSP](#) 中設定 STATIME，將收集播送設為定期進行，如下所示：

- 如果 STATIME 已設為大於 0 的值，即您的播送間隔 (分鐘)。
- 如果 STATIME = 0，則會使用系統的 SMF 播送 (SMF INTVAL)
- 如果您的 STATIME = 0 且未設定 SMF INTVAL，則不會進行播送，且不會產生任何間隔記錄

## 搭配使用其他產品與 IBM MQ

您可以使用其他產品來協助您改進的呈現，或擴增與效能及會計相關的統計資料。例如，Resource Measurement Facility、Tivoli Decision Support 及 CICS 監視。

## 使用 Resource Measurement Facility

資源測量機能 (RMF) 是 IBM 授權程式 (程式號碼 5685-029)，提供系統層面的處理器使用率、I/O 活動、儲存體及分頁相關資訊。您可以使用 RMF 來動態監視整個系統中實體資源的使用率。如需相關資訊，請參閱 [z/OS Resource Measurement Facility User 's Guide](#)。

## 使用 Tivoli Decision Support for z/OS

您可以使用 Tivoli Decision Support for z/OS 來解譯 RMF 和 SMF 記錄。

Tivoli Decision Support for z/OS 是 IBM 授權程式 (程式號碼 5698-B06)，可讓您透過在 Db2 資料庫中收集效能資料並以各種格式呈現資料以用於系統管理，來管理系統效能。Tivoli Decision Support for 可以使用它儲存在其 Db2 資料庫中的系統管理資料來產生圖形和表狀報告。它包括管理對話框、報告對話框及日誌收集器，所有這些都與標準 Db2 資料庫互動。

這在 [IBM Tivoli Decision Support for z/OS: Administration Guide and Reference](#) 中有說明。

## 使用 CICS 監視機能

CICS 監視機能提供每一個執行中 CICS 交易的相關效能資訊。它可以用來調查已使用的資源，以及處理交易所花費的時間。如需背景資訊，請參閱 [CICS Performance Guide](#) 及 [Developing CICS System Programs](#)，以及之前稱為 [CICS Customization Guide](#) 的隨附參考手冊。

### 調查效能問題

效能問題可能來自各種因素。例如，不正確的資源配置、不良的應用程式設計及 I/O 限制。請利用這個主題來調查效能問題的部分可能原因。

效能可能受到下列不利影響：

- 大小不正確的緩衝池
- 缺少實際儲存體
- 頁集或日誌的 I/O 競用
- 未正確設定的日誌緩衝區臨限值
- 日誌緩衝區數目的設定不正確
- 大型訊息
- 持續很長時間的回復單元，合併每一個同步點的許多訊息
- 長時間保留在佇列上的訊息
- RACF 審核
- 不必要的安全檢查
- 無效率的程式設計

當您分析效能資料時，請一律先查看整體系統，然後再決定您有特定的 IBM MQ 問題。請記住，當發生競爭時，幾乎所有效能降低的症狀都會放大。例如，如果有 DASD 競用，交易回應時間可能會增加。此外，系統中的交易越多，處理器用量越大，對虛擬和實際儲存體的需求也越大。

在這類狀況下，系統會顯示大量使用其所有資源。不過，系統實際上正在經歷正常的系統壓力，而這種壓力可能隱藏效能降低的原因。若要尋找此類效能流失的原因，您必須考量可能影響作用中作業的所有項目。

### 調查整體系統

在 IBM MQ 內，效能問題是增加回應時間，或大量使用非預期且無法解釋的資源。第一個檢查因素，例如處理器用量總計、DASD 活動及分頁。用於檢查處理器用量總計的 IBM 工具是資源管理機能 (RMF)。一般而言，您必須詳細查看系統，以瞭解為何作業緩慢進行，或為何大量使用特定資源。

從查看一般作業活動開始，然後聚焦於特定活動，例如特定作業或特定時間間隔。

另一個可能性是系統實際儲存體有限；因此，由於分頁岔斷，作業的進度比預期來得慢。

### 調查個別作業

您可以使用帳戶追蹤來收集 IBM MQ 作業的相關資訊。這些追蹤記錄告訴您大量關於作業已執行的活動，以及作業等待門鎖所花費的暫停時間。追蹤記錄也包括作業所執行的 Db2 及連結機能活動數量的相關資訊。

第 316 頁的『[解譯 IBM MQ 帳戶資料](#)』中說明解譯 IBM MQ 帳戶資料。

工作日誌中出現訊息 CSQR026I，即可識別長時間執行的工作單元。此訊息指出作業已存在超過三個佇列管理程式檢查點，且其日誌記錄已延遲。 如需日誌記錄分流的說明，請參閱 [日誌檔](#)。

### 解譯 IBM MQ 效能統計資料

請利用這個主題作為 IBM MQ for z/OS 所建立之不同 SMF 記錄的索引。

IBM MQ 效能統計資料會寫入為 SMF 類型 115 記錄。統計資料記錄會以 CSQ6SYSP 系統參數模組的 STATIME 參數所指定的時間間隔定期產生，如果您對 STATIME 指定零，則會以 SMF 廣域統計間隔定期產生。SMF 記錄中提供的資訊來自 IBM MQ 的下列元件：

<b>緩衝區管理程式</b>	管理虛擬儲存體中的緩衝池，以及在緩衝池已滿時將頁面寫入頁集。也會管理從頁面集讀取頁面。
<b>連結機能管理程式</b>	管理具有連結機能的介面。
<b>Data Manager</b>	管理訊息與佇列之間的鏈結。它會呼叫緩衝區管理程式來處理含有訊息的頁面。
<b>Db2 管理員</b>	管理與用作共用儲存庫的 Db2 資料庫的介面。
<b>鎖定管理程式</b>	管理 IBM MQ for z/OS 的鎖定。
<b>日誌管理程式</b>	管理日誌記錄的寫入，如果有回復要求，則這是維護系統完整性的必要項目，如果有系統或媒體故障，則是回復的必要項目。
<b>訊息管理程式</b>	處理所有 IBM MQ API 要求。
<b>儲存體管理程式</b>	管理 IBM MQ for z/OS 的儲存體，例如儲存區配置、擴充及取消配置。
<b>主題管理程式</b>	管理 IBM MQ for z/OS 的主題及訂閱資訊。
<b>連結機能 SMDS 管理程式</b>	針對儲存在連結機能中的大型訊息，管理共用訊息資料集 (SMDS)。

IBM MQ 統計資料會以 SMF 類型 115 記錄形式寫入 SMF。可以呈現下列子類型：

- 1**  
例如，與日誌及儲存體相關的系統資訊。
- 2**  
訊息數及分頁資訊的相關資訊。與連結機能及 Db2 相關的佇列共用群組資訊。
- 5 和 6**  
佇列管理程式位址空間中內部儲存體使用情形的詳細資訊。雖然您可以檢視此資訊，但部分資訊僅供 IBM 使用。
- 7**  
儲存體管理程式摘要資訊。雖然您可以檢視此資訊，但部分資訊僅供 IBM 使用。
- 201**  
頁集輸入/輸出資訊
- 215**  
緩衝池資訊
- 231**  
通道起始程式位址空間的系統資訊。

請注意：

- 子類型 1、2、201 及 215 記錄以統計資料追蹤類別 1 建立。
- 子類型 7 記錄以統計資料追蹤類別 2 建立。
- 子類型 5 和 6 記錄以統計資料追蹤類別 3 建立。
- 子類型 231 記錄以統計資料追蹤類別 4 建立。

子類型指定在 SM115STF 欄位中 (如 第 296 頁的表 34 所示)。

## **SMF 類型 115 記錄的佈置**

您可以使用此區段作為 SMF 類型 115 記錄格式的參照。

SMF 記錄的標準佈置包含三個部分：

### **SMF 標頭**

提供記錄本身的格式、識別及時間和日期資訊。

## 自行定義區段

定義 SMF 記錄內個別資料記錄的位置和大小。

## 資料記錄

您要分析來自 IBM MQ 的實際資料。

如需 SMF 記錄格式的相關資訊，請參閱 [z/OS MVS 系統管理機能 \(SMF\)](#)。

## 相關參考

第 296 頁的『SMF 標頭』

請利用這個主題作為 SMF 標頭格式的參照。

第 296 頁的『自行定義區段』

請利用這個主題來作為 SMF 記錄自我定義區段的格式參照。

第 297 頁的『SMF 統計資料記錄的範例』

請利用這個主題來瞭解部分 SMF 記錄範例。

## z/OS SMF 標頭

請利用這個主題作為 SMF 標頭格式的參照。

第 296 頁的表 34 顯示 SMF 記錄標頭的格式 (SM115)。

偏移:Dec	偏移:十六進位	類型	Len	姓名	說明	範例
0	0	結構	28	SM115	SMF 記錄標頭。	
0	0	整數	2	SM115LEN	SMF 記錄長度。	14A0
2	2		2		保留。	
4	4	整數	1	SM115FLG	系統指示器。	5E
5	5	整數	1	SM115RTY	記錄類型。SMF 記錄類型，對於 IBM MQ 統計資料記錄，這一律是 115 (X'73 ')。	73
6	6	整數	4	SM115TME	SMF 移動記錄的時間。	00355575
10	A	整數	4	SM115DTE	SMF 移動記錄的日期。	0100223F
14	E	字元	4	SM115SID	z/OS 子系統 ID。定義收集記錄所在的 z/OS 子系統。	D4E5F4F1 (MV41)
18	12	字元	4	SM115SSI	IBM MQ 子系統 ID。	D4D8F0F7 (MQ07)
22	16	整數	2	SM115STF	記錄子類型。	0002
24	18	字元	3	SM115REL	IBM MQ 版本。	F6F0F0 (600)
27	1B		1		已保留	
28	1C	字元	0	SM115END	SMF 標頭的結尾和自我定義區段的開頭。	

## z/OS 自行定義區段

請利用這個主題來作為 SMF 記錄自我定義區段的格式參照。

類型 115 SMF 記錄的自行定義區段會告訴您尋找統計資料記錄的位置、記錄的時間長度，以及該記錄類型的重複次數 (具有不同的值)。自我定義區段會在標頭後面，從 SMF 記錄開始算起的固定偏移。每一個統計資料記錄都可以由醒目目標示字串來識別。

對於類型 115 記錄，使用者可以使用下列類型的自行定義區段。每一個自行定義區段都指向與其中一個 IBM MQ 元件相關的統計資料。第 297 頁的表 35 會從 SMF 記錄標頭開始，彙總統計資料的來源、醒目字串及自我定義區段的偏移。

表 35: 自我定義區段的偏移				
統計資料來源	記錄子類型 (SM115STF)	自行定義區段的偏移		資料捕眼器
		十二月	十六進位碼	
儲存體管理程式	1	100	X'64 '	QSST
日誌管理程式	1	116	X'74 '	QJST
訊息管理程式	2	36	X'24 '	QMST
Data Manager	2	44	X'2C'	QIST
<b>V9.1.0</b> <b>V9.1.0</b> 不再使用。自我定義區段將是二進位零。	2	52	X'34 '	
鎖定管理程式	2	60	X'3C'	QLST
Db2 manager	2	68	X'44 '	Q5ST
連結機能管理程式	2	76	X'4C'	QEST
主題管理程式	2	84	X'54 '	QTST
SMDS 使用情形	2	92	X'5C'	QESD
緩衝區管理程式-每一個緩衝池各一個	215	36	X'24 '	QPST
通道起始程式	231			QWSX
資料管理程式頁面集-每一個頁面集一個	201	36	X'24 '	QIS1
儲存體管理程式	5	36	X'24 '	QSPH
儲存體管理程式	6	36	X'24 '	QPGM
儲存體管理程式	7	36	X'24 '	QSRS

註: 子類型 5、6 及 7 記錄中的部分儲存體管理程式資訊僅供 IBM 使用。其他未列出的自行定義區段僅包含供 IBM 使用的資料。

每一個自我定義區段都有兩個完整單字，且具有下列格式:

```
sssssssllllnnn
```

其中:

- ssssssss 是一個全字組，包含 SMF 記錄開頭的偏移。
- llll 是提供此資料記錄長度的半字組。
- nnnn 是提供此 SMF 記錄中資料記錄數的半字。

如需相關資訊，請參閱第 297 頁的『SMF 統計資料記錄的範例』。

註: 一律在自行定義區段中使用偏移來尋找統計資料記錄。

## SMF 統計資料記錄的範例

請利用這個主題來瞭解部分 SMF 記錄範例。

第 298 頁的圖 20 顯示子類型 1 的 SMF 記錄組件範例。子類型 1 包括儲存體管理程式及日誌管理程式統計資料記錄。SMF 記錄標頭會顯示畫底線。

位於偏移 X'64' 的自行定義區段是指儲存體管理程式統計資料，而位於偏移 X'74' 的自行定義區段是指日誌管理程式統計資料，兩者都以 **粗體** 顯示。

儲存體管理程式統計資料記錄位於標頭開頭的偏移 X'0000011C' 處，且長度為 X'48' 個位元組。有一組儲存體管理程式統計資料，由醒目字串 QSST 識別。範例中也會顯示此統計資料記錄的開頭。

日誌管理程式統計資料記錄位於從標頭開始算起的偏移 X'00000164' 處，且長度為 X'78' 位元組。有一組日誌管理程式統計資料，由醒目標示字串 QJST 識別。

```

000000  02000000 5E730035 55750100 223FD4E5  *.....;.....MV*
000010  F4F1D4D8 F0F70001 F6F0F000 000001DC  *41MQ07..600....*
000020  00240001 00000000 00000000 00000000  *.....*
000030  00000000 00000000 00000000 0000007C  *.....@*
000040  00400001 000000BC 00600001 00000000  *.....*
000050  00000000 00000000 00000000 00000000  *.....*
000060  00000000 0000011C 00480001 00000000  *.....*
000070  00000000 00000164 00780001 00000000  *.....*
000080  00000000 00000000 00000000 00000000  *.....*
.
000110  00000000 00000000 00000000 003C0048  *.....*
000120  D8E2E2E3 0000004F 00000003 00000002  *QSST...|.....*

```

圖 20: SMF 記錄 115，子類型 1

第 299 頁的圖 21 顯示子類型 2 的 SMF 記錄組件範例。子類型 2 包括訊息、資料、鎖定、連結機能、主題及 Db2 管理程式的統計資料記錄。SMF 記錄標頭會畫底線顯示；自行定義區段會交替顯示 **粗體** 和 **斜體**。

- 偏移 X'24' 處的自行定義區段參照訊息管理程式統計資料。訊息管理程式統計資料記錄位於標頭開頭的偏移 X'00000064' 處，且長度為 X'48' 位元組。有一組這些統計資料，由醒目字串 QMST 識別。
- 位於偏移 X'2C' 的自行定義區段是指資料管理程式統計資料。資料管理程式統計資料記錄位於標頭開頭的偏移 X'000000AC' 處，且長度為 X'50' 位元組。這些統計資料有一組，由醒目字串 QIST 識別。
- **V9.1.0** 在舊版中，偏移 X'34' 處的自行定義區段參照緩衝區管理程式統計資料。由於此 SMF 記錄是從 IBM MQ 9.1.0 佇列管理程式取得，因此緩衝區管理程式自我定義區段會設為零，以指出沒有緩衝區管理程式統計資料。相反地，這些統計資料是在 SMF 115 子類型 215 記錄中。
- 位於偏移 X'3C' 的自行定義區段是指鎖定管理程式統計資料。鎖定管理程式統計資料記錄位於標頭開頭的偏移 X'000000FC' 處，且長度為 X'20' 位元組。有一組這些統計資料，由醒目標示字串 QLST 識別。
- 偏移 X'44' 處的自行定義區段是指 Db2 管理程式統計資料。Db2 管理程式統計資料記錄位於標頭開頭的偏移 X'0000011C' 處，且長度為 X'2A0' 個位元組。這些統計資料有一組，由醒目標示字串 Q5ST 識別。
- 位於偏移 X'4C' 的自行定義區段是指連結機能管理程式統計資料。連結機能管理程式統計資料記錄位於標頭開頭的偏移 X'000003BC' 處，且長度為 X'1008' 位元組。有一組這些統計資料，由醒目字串 QEST 識別。
- 位於偏移 X'54' 的自行定義區段會參照主題管理程式統計資料。主題管理程式統計資料記錄位於標頭開頭的偏移 X'000013C4' 處，且長度為 X'64' 個位元組。有一組這些統計資料，由醒目標示字串 QTST 所識別。
- 位於偏移 X'5C' 的自行定義區段適用於 SMDS 統計資料。此自行定義區段設為零，表示未使用 SMDS。

```

V 9.1.0
000000 09F40000 5E730033 4DBE0113 142FD4E5 *.4..;...(....MV*
000010 F4F1D4D8 F2F10002 F9F1F000 00001428 *41MQ21..910....*
000020 00240001 00000064 00480001 000000AC *......*
000030 00500001 00000000 00000000 000000FC *......*
000040 00200001 0000011C 02A00001 000003BC *......*
000050 10080001 000013C4 00640001 00000000 *......D.....*
000060 00000000 D40F0048 D8D4E2E3 00000000 *...M...QMST...*
000080 00000000 00000000 00000000 00000000 *......*
000090 00000000 00000000 00000000 00000000 *......*
0000A0 00000000 00000000 00000000 C90F0050 *......I.&*
0000B0 D8C9E2E3 00000000 00000000 00000000 *QIST.....*
0000C0 00000000 00000000 00000000 00000000 *......*
0000D0 00000000 00000000 00000000 00000000 *......*
0000E0 00000000 00000000 00000000 00000000 *......*
0000F0 00000000 00000000 00000000 D30F0020 *......L...*
000100 D8D3E2E3 00000000 00000000 00000000 *QLST.....*
000110 00000000 00000000 00000000 F50F02A0 *......5...*
000120 D8F5E2E3 00000008 00000000 00000000 *Q5ST.....*
.
.

```

圖 21: SMF 記錄 115，子類型 2

### z/OS 處理類型 115 SMF 記錄

請使用本主題作為處理類型 115 SMF 記錄的參照。

您必須處理從 SMF 收集的任何資料，以擷取有用資訊。當您處理資料時，請驗證記錄來自 IBM MQ，且它們是您預期的記錄。

驗證下列欄位的值：

- SM115RTY(SMF 記錄號碼) 必須是 X'73 ' (115)
- SM115STF(記錄子類型) 必須是 0001、0002、0005、0006、0007、0201、0215 或 0231

不支援從作用中 SMF 資料集(或 SMF 日誌串流) 讀取。您必須使用 SMF 程式 IFASMFDP (或 IFASMF DL，如果正在使用日誌串流)，將 SMF 記錄傾出至循序資料集，以便可以處理它們。如需相關資訊，請參閱第 292 頁的『使用系統管理機能』。

如需結構及欄位的詳細資料，請參閱 IBM MQ SupportPac MP1B。

有一個稱為 CSQ4SMFD 的 C 範例程式，它會從循序資料集列印 SMF 類型 115 和 116 記錄的內容。程式在 thlqual.SCSQC37S 中以原始檔形式提供，在 thlqual.SCSQLOAD 中以執行檔格式提供。thlqual.SCSQPROC(CSQ4SMFJ) 中提供範例 JCL。

### z/OS 儲存體管理程式資料記錄

請使用本主題作為儲存體管理程式資料記錄的參照。

儲存體管理程式統計資料記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDQSST) 中說明。

資料包含在統計資料間隔期間，佇列管理程式已配置、擴充、收合及刪除的固定及可變儲存區數目，以及 z/OS 的 GETMAIN、FREEMAIN 及 STORAGE 要求數的相關資訊，包括那些不成功的要求計數。其他資訊包括偵測到儲存體不足狀況的次數計數，以及因該狀況而發生異常終止的次數計數。

類別 2 及類別 3 統計資料追蹤會產生佇列管理程式中儲存體使用情形的其他資料。雖然您可以檢視此資訊，但部分資訊僅供 IBM 使用。

- 在組譯器巨集 thlqual.SCSQMACS(CSQDQSPH) 中說明呈現在子類型 5 記錄中的儲存體管理程式儲存區標頭統計資料記錄格式。
- 在組譯器巨集 thlqual.SCSQMACS(CSQDQPGM) 中說明儲存體管理程式 getmain 統計資料記錄的格式(在子類型 6 記錄中存在)。
- 儲存體管理程式區域摘要記錄的格式(在子類型 7 記錄中)在組譯器巨集 thlqual.SCSQMACS(CSQDQSRS) 中說明。

請使用本主題作為日誌管理程式資料記錄格式的參照。

日誌管理程式統計資料記錄的格式在組譯器巨集 `thlqual.SCSQMACS(CSQDQJST)` 中說明。

在統計資料中，這些計數很重要：

1. 日誌寫入要求總數：

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. 日誌讀取要求總數：

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

下表說明可使用日誌管理程式統計資料檢查的問題症狀。

### 症狀 1

QJSTWTB 不是零。

#### 原因

將儲存體內緩衝區寫入作用中日誌時，暫停作業。

寫入作用中日誌時可能發生問題。

CSQ6LOGP 內的 OUTBUFF 參數太小。

#### 動作

請調查寫入作用中日誌的問題。

增加 CSQ6LOGP 內 OUTBUFF 參數的值。

### 症狀 2

比例:  $\text{QJSTWTL} / N_{\text{logread}}$  大於 1%。

#### 原因

已起始必須從保存日誌讀取日誌讀取，但 IBM MQ 無法配置資料集，因為已配置 MAXRTU 資料集。

#### 動作

增加 MAXRTU。

### 症狀 3

比例:  $\text{QJSTRARH} / N_{\text{logread}}$  大於正常。

#### 原因

大部分日誌讀取要求應該來自輸出緩衝區或作用中日誌。為了滿足回復要求，會從儲存體內緩衝區、作用中日誌及保存日誌中讀取回復單元記錄。

長時間執行的回復單元 (在許多分鐘的期間內延伸) 可能會將日誌記錄分散在許多不同的日誌中。這會降低效能，因為必須執行額外工作才能回復日誌記錄。

#### 動作

變更應用程式以減少回復單元的長度。此外，請考慮增加作用中日誌的大小，以減少單一回復單元分散在多個日誌上的可能性。

#### 其他指標

比例  $N_{\text{logread}} / N_{\text{logwrite}}$  指出必須取消的工作量。

#### 症狀 4

QJSTLLCP 每小時超過 10 個。

#### 原因

在忙碌的系統上，您預期每小時通常會看到 10 個檢查點。如果 QJSTLLCP 值大於此值，則表示佇列管理程式的設定有問題。

最可能的原因是 CSQ6SYSP 中的 LOGLOAD 參數太小。導致檢查點的另一個事件是當作用中日誌填滿並切換至下一個作用中日誌資料集時。如果日誌太小，這可能會導致頻繁的檢查點。

QJSTLLCP 是檢查點總數的計數。

#### 動作

增加 LOGLOAD 參數，或視需要增加日誌資料集的大小。

#### 症狀 5

QJSTCmpFail > 0 或 QJSTCmpComp 不小於 QJSTCmpUncmp

#### 原因

佇列管理程式無法大幅壓縮日誌記錄。

QJSTCmpFail 是佇列管理程式無法達到任何記錄長度縮減的次數。您應該將此數目與 QJSTCmpReq (壓縮要求數目) 進行比較，以查看失敗數目是否顯著。

QJSTCmpComp 是寫入日誌的壓縮位元組總數，而 QJSTCmpUncmp 是壓縮之前的位元組總數。這兩個總計都不包含不適合壓縮之日誌記錄寫入的位元組數。如果數字相似，則壓縮沒有什麼好處。

#### 動作

關閉日誌壓縮。發出 SET LOG COMPLOG (NONE) 指令。如需詳細資料，請參閱 [SET LOG](#) 指令。

註：在系統啟動之後產生的第一組統計資料中，由於解析進行中的回復單元，可能有重要的日誌活動。

### 訊息管理程式資料記錄

請使用本主題作為訊息管理程式資料記錄的參照。

訊息管理程式統計資料記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDQMST) 中說明。

資料提供不同 IBM MQ API 要求的計數。

### 資料管理程式資料記錄

請使用本主題作為 Data Manager 資料記錄格式的參照。

資料管理程式統計資料記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDQIST) 中說明。

資料提供不同物件要求的計數。

### 資料管理程式頁面集資料記錄

使用此區段作為資料管理程式頁面集資料記錄格式的參照

資料管理程式頁集統計資料記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDQIS1) 中說明。

頁集使用情形資訊透過記錄諸如頁集輸入/輸出速率及最高使用情形之類的資訊，有助於更好地管理佇列管理程式內的本端佇列。

此資料提供與 MQSC **DISPLAY USAGE TYPE (PAGESET)** 指令或 PCF [Inquire Usage \(MQCMD\\_INQUIRE\\_USAGE\)](#) 指令輸出相同的基本頁集資訊。

例如：

- 總頁數
- 現行已使用的頁面
- 未用的持續及非持續頁面
- 擴充方法

- 延伸數
- 分段數目

資料也會提供一些效能指標，以及其他效能資訊。例如：

- 在 SMF 間隔期間，發生延遲寫入、立即寫入及讀取頁面 I/O 要求的次數，
- 移動的頁數、經歷時間，以及讀取及寫入作業數。
- 在檢查點中寫入的頁面數。
- 在 SMF 間隔期間是否發生擴充？
- 頁面集填滿的次數。
- 指出在頁集內配置新空間的位置。

從顯示的資訊中，您應該能夠瞭解每一個頁集的一般狀態，並考量是否需要重新調整系統。

### 相關參考

第 296 頁的『SMF 標頭』

請利用這個主題作為 SMF 標頭格式的參照。

第 296 頁的『自行定義區段』

請利用這個主題來作為 SMF 記錄自我定義區段的格式參照。

第 297 頁的『SMF 統計資料記錄的範例』

請利用這個主題來瞭解部分 SMF 記錄範例。

### 緩衝區管理程式資料記錄

請使用本主題作為緩衝區管理程式資料記錄格式的參照。

緩衝區管理程式統計資料記錄的格式在組譯器巨集 `thlqual.SCSQMACS(CSQDQPST)` 中說明。

**註：**只會為定義的緩衝池建立緩衝區管理程式統計資料記錄。如果已定義緩衝池但未使用，則不會設定任何值，且其緩衝區管理程式統計資料記錄將不會包含任何資料。

如需有效管理緩衝池的相關資訊，請參閱 [第 303 頁的『管理緩衝池』](#)。

在解譯統計資料時，建議您考量下列因素，因為這些欄位的值可用來增進系統效能：

1. 如果 QPSTSOS、QPSTDMC 或 QPSTIMW 大於零，您應該增加緩衝池的大小，或將頁集重新配置到不同的緩衝池。
  - QPSTSOS 是頁面取得要求沒有可用緩衝區的次數。如果 QPSTSOS 變成非零，則表示 IBM MQ 受到嚴重壓力。應該增加緩衝池大小。如果增加緩衝池大小不會使 QPSTSOS 值變成零，則 DASD 頁集上可能會有 I/O 競用。
  - QPSTDMC 是同步執行的更新數，因為緩衝池中等待寫入 I/O 的頁面超過 95%，或緩衝池中可用於讀取要求的緩衝池小於 5%。如果此數字不是零，則緩衝池可能太小且應該放大。如果增加緩衝池大小不會將 QPSTDMC 減少至零，則 DASD 頁集上可能會有 I/O 競用。
  - QPSTIMW 是同步寫出頁面的次數計數。如果 QPSTDMC 為零，則 QPSTIMW 是在佇列上找到等待至少已存在兩個檢查點之寫入 I/O 的頁面的次數。
2. 對於緩衝池零及包含短期訊息的緩衝池：
  - QPSTDWT 應該為零，且百分比 QPSTCBSL/QPSTNBUF 應該大於 15%。  
QPSTDWT 是因為緩衝池中超過 85% 的頁面等待寫入 I/O，或小於 15% 的緩衝池可用於讀取要求，而啟動非同步寫入處理器的次數。增加緩衝池大小應該會減少此值。如果沒有，則存取型樣是放置與取得之間的長時間延遲。
  - 由於檢查點檢查作業活動，QPSTTPW 可能大於零。
  - 除非在重新啟動佇列管理程式之後從頁集讀取訊息，否則 QPSTRIO 應該為零。

QPSTRIO 與 QPSTGETP 的比例顯示緩衝池內頁面擷取的效率。增加緩衝池大小應該會降低此比例，因此提高頁面擷取效率。如果未發生此情況，則表示未經常重新存取頁面。這意味著交易型樣在放置訊息之後擷取的訊息之間會有很長的延遲。

QPSTGETN 與 QPSTGETP 的比例指出已要求空白頁面 (相對於非空白頁面) 的次數。此比例是交易型樣的指示, 而不是可用來調整系統的值。

- 如果 QPSTSTL 的值大於零, 則表示現在正在使用之前未使用過的頁面。這可能是因為訊息速率增加、訊息的處理速度不如先前快 (導致訊息累積), 或使用較大的訊息。

QPSTSTL 是頁面存取要求找不到緩衝池中已存在頁面的次數計數。同樣地, QPSTSTL 與 (QPSTGETP + QPSTGETN) 的比例越低, 則頁面擷取效率越高。增加緩衝池大小應該會降低此比例, 但如果沒有, 則表示放置與取得之間有較長的延遲。

- 建議您有足夠的緩衝區來處理尖峰訊息速率。

### 3. 對於具有長期存在訊息的緩衝池, 其中的訊息數超過緩衝池可容納的訊息數:

- $(QPSTRIO + QPSTWIO)$  / 統計資料間隔是頁集的 I/O 速率。如果此值很高, 您應該考慮在不同磁區上使用多個頁集, 以容許平行執行 I/O。
- 在處理訊息的時段內 (例如, 如果訊息在白天寫入佇列並在夜間處理), 讀取 I/O (QPSTRIO) 的數目應該大約等於寫入的總頁數 (QPSTTPW)。這會顯示每寫入一頁都會讀取一頁。

如果 QPSTRIO 比 QPSTTPW 大很多, 則表示正在多次讀取頁面。這可能是應用程式在未檢索佇列時使用 *MsgId* 或 *CorrelId* 的 MQGET, 或使用 *get next* 來瀏覽佇列上的訊息的結果。

下列動作可能會緩解此問題:

- a. 除了任何已變更的頁面之外, 請增加緩衝池的大小, 以便有足夠的頁面來保留佇列。
- b. 使用 *INDXTYPE* 佇列屬性, 可讓 *MsgId* 或 *CorrelId* 檢索佇列, 且不需要循序掃描佇列。
- c. 變更應用程式的設計, 以避免使用 MQGET 搭配 *MsgId* 或 *CorrelId*, 或 *get next with browse* 選項。

註: 使用長期訊息的應用程式通常會處理第一個可用的訊息, 且不會將 MQGET 與 *MsgId* 或 *CorrelId* 搭配使用, 而且它們可能只會瀏覽第一個可用的訊息。

- d. 將頁面集移至不同的緩衝池, 以減少來自不同應用程式的訊息之間的競爭。

## z/OS 管理緩衝池

若要有效地管理緩衝池, 您必須考量影響緩衝池 I/O 作業的因素, 以及與緩衝池相關聯的統計資料。

下列因素會影響緩衝池 I/O 作業。

- 如果在緩衝池中找不到包含所需資料的頁面, 則會從其 DASD 頁集同步讀入可用的緩衝區。
- 每當更新頁面時, 便會將它放在要 (可能) 寫出至 DASD 的內部頁面佇列中。這表示在緩衝區寫入 DASD 之前, 任何其他頁面都無法使用該頁面所使用的緩衝區。
- 如果佇列要寫入 DASD 的頁數超過儲存區中緩衝區總數的 85%, 則會啟動非同步寫入處理器, 將緩衝區放入 DASD。

同樣地, 如果可用於頁面取得要求的緩衝區數目變成小於儲存區中緩衝區總數的 15%, 則會啟動非同步寫入處理器來執行寫入 I/O 作業。

當置入佇列要寫入 DASD 的頁面數已降至儲存區中緩衝區總數的 75% 時, 寫入處理器即會停止。

- 如果置入佇列以寫入 DASD 的頁數超過儲存區中緩衝區總數的 95%, 則所有更新都會導致同步將頁面寫入 DASD。

同樣地, 如果可用於頁面取得要求的緩衝區數目變成小於儲存區中緩衝區總數的 5%, 則所有更新都會導致同步將頁面寫入 DASD。

- 如果頁面取得要求可用的緩衝區數目達到零, 則會暫停遇到此狀況的交易, 直到非同步寫入處理器完成為止。
- 如果經常更新頁面, 頁面會將大部分時間花在等待寫入 DASD 的頁面佇列上。因為此佇列處於近期最少使用的順序, 所以放置在此近期最少使用佇列上的經常更新頁面可能永遠不會寫出至 DASD。因此, 在更新時, 如果發現頁面已在等待 DASD 佇列的寫入作業至少兩個檢查點, 則會同步寫入 DASD。會在檢查點時間進行更新, 並暫停直到非同步寫入處理器完成為止。

此演算法的目的是在系統負載讓緩衝池用量承受壓力時, 將緩衝池記憶體中所花費的時間最大化, 同時讓系統正常運作。

## z/OS 鎖定管理程式資料記錄

請使用本主題作為鎖定管理程式資料記錄格式的參照。

鎖定管理程式統計資料記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDQLST) 中說明。

記錄包含下列資訊的相關資料：

- 鎖定取得要求及鎖定釋放要求的數目。
- 鎖定取得要求判定所要求的鎖定已保留的次數。

## z/OS Db2 管理程式資料記錄

請使用本主題作為 Db2 管理程式資料記錄格式的參照。

下表及組譯器巨集 thlqual.SCSQMACS(CSQDQ5ST) 及 C 標頭檔 thlqual.SCSQC370(CSQDSMFC) 中說明 Db2 管理程式統計資料記錄的格式。C 中的欄位名稱都是小寫，例如 q5st、q5stid。

如果佇列管理程式未啟動為佇列共用群組的成員，則不會在此記錄中記錄任何資料。

偏移:Dec	偏移: 十六進位	類型	Len	姓名	說明
0	0	結構	668	Q5ST	Db2 管理程式統計資料
0	0	Bitstring	2	Q5STID	控制區塊 ID
2	2	整數	2	Q5STLL	控制區塊長度
4	4	字元	4	Q5STEYEC	控制區塊捕眼器
8	8	字元	660	Q5STZERO	偶爾清除 QMST 組件
8	8	整數	4	NUMTASKS	伺服器作業數
12	C	整數	4	ACTTASK	作用中伺服器作業數
16	10	整數	4	CONNCNT	連接要求數
20	14	整數	4	DISCCNT	斷線要求數
24	18	整數	4	DHIGMAX	要求佇列深度上限
28	1C	整數	4	ABNDCNT	Db2SRV 作業異常終止數
32	20	整數	4	REQUCNT	重新佇列的要求數
36	24	整數	4	DEADCNT	死鎖逾時數
40	28	整數	4	DELECNT	刪除要求數
44	2C	整數	4	LISTCNT	清單要求數
48	30	整數	4	READCNT	讀取要求數
52	34	整數	4	UPDCNT	更新要求數
56	38	整數	4	WRITCNT	寫入要求數
60	3C	整數	4	SCSSEL	SCST (共用通道狀態) 選取
64	40	整數	4	SCSINS	SCST 插入
68	44	整數	4	SCSUPD	SCST 更新
72	48	整數	4	SCSDEL	SCST 刪除

表 36: Db2 統計資料記錄 (Q5ST) (繼續)

偏移:Dec	偏移: 十六進位	類型	Len	姓名	說明
76	4C	整數	4	SSKSEL	SSKT (共用同步金鑰) 選取
80	50	整數	4	SKINS	SSKT 插入
84	54	整數	4	SSKDEL	SSKT 刪除數
88	58	整數	4	SCSBFTS	SCST 次數緩衝區太小
92	5C	整數	4	SCSMAXR	查詢上的 SCST 列數上限
96	60	整數	4	* (2)	已保留
104	68	字元	8	DELETCUW	累加 STCK 差異-執行緒刪除
112	70	字元	8	DELETMXW	最大 STCK 差異-執行緒刪除
120	78	字元	8	DELESCUW	累加 STCK 差異-SQL 刪除
128	80	字元	8	DELESMXW	最大 STCK 差異-SQL 刪除
136	88	字元	8	LISTTCUW	累積 STCK 差異-執行緒清單
144	90	字元	8	LISTTMXW	最大 STCK 差異-執行緒清單
152	98	字元	8	LISTSCUW	累加 STCK 差異-SQL 清單
160	A0	字元	8	LISTSMXW	最大 STCK 差異-SQL 清單
168	A8	字元	8	READTCUW	累加 STCK 差異-執行緒讀取
176	B0	字元	8	READTMXW	最大 STCK 差異-執行緒讀取
184	B8	字元	8	READSCUW	累加 STCK 差異-SQL 讀取
192	C0	字元	8	READSMXW	最大 STCK 差異-SQL 讀取
200	C8	字元	8	UPDTTCUW	累加 STCK 差異-執行緒更新
208	D0	字元	8	UPDTTMXW	最大 STCK 差異-執行緒更新
216	D8	字元	8	UPDTSCUW	累加 STCK 差異-SQL 更新
224	E0	字元	8	UPDTSMXW	最大 STCK 差異-SQL 更新
232	E8	字元	8	WRITTCUW	累加 STCK 差異-執行緒寫入
240	F0	字元	8	WRITTMXW	最大 STCK 差異-執行緒寫入
248	F8	字元	8	WRITSCUW	累加 STCK 差異-SQL 寫入
256	100	字元	8	WRITSMXW	最大 STCK 差異-SQL 寫入
264	108	字元	8	SCSSTCUW	累加 STCK 差異-執行緒選取
272	110	字元	8	SCSSTMXW	最大 STCK 差異-執行緒選取
280	118	字元	8	SCSSSCUW	累加 STCK 差異-SQL select
288	120	字元	8	SCSSSMXW	最大 STCK 差異-SQL select
296	128	字元	8	SCSITCUW	累加 STCK 差異-執行緒插入
304	130	字元	8	SCSITMXW	最大 STCK 差異-執行緒插入
312	138	字元	8	SCSISCUW	累加 STCK 差異-SQL 插入

表 36: Db2 統計資料記錄 (Q5ST) (繼續)

偏移:Dec	偏移: 十六進位	類型	Len	姓名	說明
320	140	字元	8	SCSISMXW	最大 STCK 差異-SQL 插入
328	148	字元	8	SCSUTCW	累加 STCK 差異-執行緒更新
336	150	字元	8	SCSUTMXW	最大 STCK 差異-執行緒更新
344	158	字元	8	SCSUSCUW	累加 STCK 差異-SQL 更新
352	160	字元	8	SCSUSMXW	最大 STCK 差異-SQL 更新
360	168	字元	8	SCSDTCW	累加 STCK 差異-執行緒刪除
368	170	字元	8	SCSDTMXW	最大 STCK 差異-執行緒刪除
376	178	字元	8	SCSDSCW	累加 STCK 差異-SQL 刪除
384	180	字元	8	SCSDSMXW	最大 STCK 差異-SQL 刪除
392	188	字元	8	SSKSTCW	累加 STCK 差異-執行緒選取
400	190	字元	8	SSKSTMXW	最大 STCK 差異-執行緒選取
408	198	字元	8	SSKSSCW	累加 STCK 差異-SQL select
416	1A0	字元	8	SSKSSMXW	最大 STCK 差異-SQL select
424	1A8	字元	8	SSKITCW	累加 STCK 差異-執行緒插入
432	1B0	字元	8	SSKITMXW	最大 STCK 差異-執行緒插入
440	1B8	字元	8	SSKISCW	累加 STCK 差異-SQL 插入
448	1C0	字元	8	SSKISMXW	最大 STCK 差異-SQL 插入
456	1C8	字元	8	SSKDTCW	累加 STCK 差異-執行緒刪除
464	1D0	字元	8	SSKDTMXW	最大 STCK 差異-執行緒刪除
472	1D8	字元	8	SSKDSCW	累加 STCK 差異-SQL 刪除
480	1E0	字元	8	SSKDSMXW	最大 STCK 差異-SQL 刪除
488	1E8	整數	4	LMSSEL	Db2 BLOB 讀取要求數
492	1EC	整數	4	LMSINS	Db2 BLOB 插入要求數
496	1F0	整數	4	LMSUPD	Db2 BLOB 更新要求數
500	1F4	整數	4	LMSDEL	Db2 BLOB 刪除要求數
504	1F8	整數	4	LMSLIS	Db2 BLOB 清單要求數
508	1FC	64 位元整數	8	LMSSTCW	所有執行緒讀取 BLOB 要求的經歷時間總計
516	204	64 位元整數	8	LMSSTMXW	執行緒讀取 BLOB 要求的最長經歷時間
524	20C	64 位元整數	8	LMSSSCW	所有 SQL 讀取 BLOB 要求的經歷時間總計
532	214	64 位元整數	8	LMSSSMXW	SQL 讀取 BLOB 要求的最長經歷時間

表 36: Db2 統計資料記錄 (Q5ST) (繼續)

偏移:Dec	偏移: 十六進 位	類型	Len	姓名	說明
540	21C	64 位元整數	8	LMSITCUW	所有執行緒插入 BLOB 要求的經歷時間總計
548	224	64 位元整數	8	LMSITMXW	執行緒插入 BLOB 要求的最長經歷時間
556	22C	64 位元整數	8	LMSISCUW	所有 SQL 插入 BLOB 要求的經歷時間總計
564	234	64 位元整數	8	LMSISMXW	SQL 插入 BLOB 要求的最長經歷時間
572	23C	64 位元整數	8	LMSUTCW	所有執行緒更新 BLOB 要求的經歷時間總計
580	244	64 位元整數	8	LMSUTMXW	執行緒更新 BLOB 要求的最長經歷時間
588	24C	64 位元整數	8	LMSUSCUW	所有 SQL 更新 BLOB 要求的經歷時間總計
596	254	64 位元整數	8	LMSUSMXW	SQL 更新 BLOB 要求的最長經歷時間
604	25C	64 位元整數	8	LMSDTCW	所有執行緒刪除 BLOB 要求的經歷時間總計
612	264	64 位元整數	8	LMSDTMXW	執行緒刪除 BLOB 要求的最長經歷時間
620	26C	64 位元整數	8	LMSDSCW	所有 SQL 刪除 BLOB 要求的經歷時間總計
628	274	64 位元整數	8	LMSDSMXW	SQL 刪除 BLOB 要求的最長經歷時間
636	27C	64 位元整數	8	LMSLTCW	所有執行緒清單 BLOB 要求的經歷時間總計
644	284	64 位元整數	8	LMSLTMXW	執行緒清單 BLOB 要求的最長經歷時間
652	28C	64 位元整數	8	LMSLSCW	所有 SQL 清單 BLOB 要求的經歷時間總計
660	294	64 位元整數	8	LMSLSMXW	SQL 清單 BLOB 要求的最長經歷時間

資料包含 Db2 資源管理程式支援的每一種要求類型的計數。對於這些要求類型，會保留下列項目的最長及累積經歷時間：

- 在整個 Db2 資源管理程式中所花費的時間 (稱為執行緒時間)。
- 執行要求的 RRSAF 和 SQL 組件所花費的時間 (稱為 SQL 時間的執行緒時間子集)。

還提供下列資訊：

- 連接的伺服器作業數。
- 任何伺服器作業的整體要求深度上限。
- 任何伺服器作業要求異常終止的次數。

如果異常終止計數不是零，則會提供重新排入佇列的計數，指出因異常終止而重新排入其他伺服器作業的佇列要求數。

如果平均執行緒時間明顯大於平均 SQL 時間，這可能表示執行緒要求花費過多時間，等待伺服器作業處理要求的 SQL 部分。如果是這種情況，請檢查 DHIGMAX 欄位，如果值大於 1，請考慮增加 CSQ6SYSP 系統參數巨集的 QSGDATA 參數中指定的 Db2 伺服器作業數目。

## **z/OS** 連結機能管理程式資料記錄

請使用本主題作為連結機能管理程式資料記錄格式的參照。

下表及組譯器巨集 thlqual 中說明連結機能管理程式統計資料記錄的格式。SCSQMACS (CSQDQEST) 及 C 標頭檔 thlqual.SCSQC370(CSQDSMFC)。C 中的欄位名稱都是小寫，例如 qest、qestid。

如果佇列管理程式未啟動為佇列共用群組的成員，則不會在此記錄中記錄任何資料。

偏移:Dec	偏移:十六進位	類型	Len	姓名	說明
0	0	結構	4104	QEST	CF 管理程式統計資料
0	0	Bitstring	2	QESTID	控制區塊 ID
2	2	整數	2	QESTLL	控制區塊長度
4	4	字元	4	QESTEYEC	控制塊捕眼器
8	8	字元	4096	QESTZERO	偶爾清除 QEST 組件
8	8	字元	64	QESTUC (0:63)	陣列 (每個結構一個項目)
8	8	字元	12	QESTR	結構名稱
20	14	整數	4	QESTSTRN	結構號碼
24	18	整數	4	QESTCSEC	IXLLSTE 呼叫數
28	1C	整數	4	QESTCMEC	IXLLSTM 呼叫數
32	20	字元	8	QESTSSTC	執行 IXLLSTE 呼叫所花費的時間
40	28	字元	8	QESTMSTC	執行 IXLLSTM 呼叫所花費的時間
48	30	整數	4	QESTRSEC	IXLLSTE 磁碟機數目
52	34	整數	4	QESTRMEC	IXLLSTM 磁碟機數目
56	38	整數	4	QESTSFUL	結構完整數目
60	3C	整數	4	QESTMNUS	使用中的項目數上限
64	40	整數	4	QESTMLUS	使用中元素數目上限
68	44	字元	4	*	已保留
4104	1008	字元	0	*	控制區塊結束

資料包含佇列管理程式在統計資料間隔期間可以連接的每一個連結機能清單結構 (包括 CSQ\_ADMIN 結構) 的資訊。每一個結構的資訊包括下列各項：

- IXLLSTE 及 IXLLSTM 要求的經歷時間數目及累計時間。
- 由於逾時而必須重試要求的次數。
- 發生「結構已滿」狀況的次數。

## z/OS 主題管理程式資料記錄

請使用本主題作為主題管理程式資料記錄格式的參照。

下表及組譯器巨集 `thlqual.SCSQMACS(CSQDQTST)` 及 C 標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 中說明主題管理程式統計資料記錄的格式。C 中的欄位名稱都是小寫，例如 `qtst`、`qtstid`。

偏移 :Dec	偏移: 十六進位	類型	Len	姓名	說明
0	0	結構	96	QTST	主題管理程式統計資料
0	0	Bitstring	2	QTSTID	控制區塊 ID
2	2	整數	2	QTSTLL	控制區塊長度
4	4	字元	4	TESTEYEC	控制塊捕眼器
8	8	字元	88	QTSTZERO	偶爾清除 QTST 組件
8	8	整數	4	QTSTSTOT	訂閱要求總數
12	0C	整數	4	QTSTSDUR	可延續訂閱要求
16	10	整數	4	QTSTSHIG (1: 3)	訂閱高臨界值陣列 (API、ADMIN、PROXY)
28	1C	整數	4	QTSTSLOW (1: 3)	訂閱低臨界值陣列 (API、ADMIN、PROXY)
40	28	整數	4	QTSTSEXP	訂閱已過期
44	2C	整數	4	QTSTTMSG	放入子佇列的訊息總數
48	30	整數	4	QTSTSPHW	單一發佈訂閱者高臨界值
52	34	整數	4	QTSTPTOT (1: 3)	發佈要求總數 (API、ADMIN、PROXY)
64	40	整數	4	QTSTPTHI	發佈高臨界值總計
68	44	整數	4	QTSTPTLO	發佈低臨界值總計
72	48	整數	4	QTSTPNOS	發佈至無訂閱者的計數
76	4C	整數	4	*	已保留
80	50	Bitstring	8	QTSTETHW	發佈時經歷時間 HW
88	58	Bitstring	8	QTSTETTO	發佈時經歷時間總計

## z/OS 連結機能管理程式 SMDS 資料記錄

請利用這個主題來參照連結機能管理程式共用訊息資料集 (SMDS) 資料記錄的格式。

連結機能管理程式共用訊息資料集 (SMDS) 統計資料記錄的格式在組譯器巨集 `thlqual.SCSQMACS(CSQDQESD)` 及 C 標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 中有說明。

統計資料提供所擁有共用訊息資料集的使用率、共用訊息資料集群組的 I/O 活動，以及 SMDS 緩衝區使用率的相關資訊。

如果佇列管理程式未啟動為佇列共用群組的成員，則不會在此記錄中記錄任何資料。

## z/OS 通道起始程式 SMF 類型 115 記錄的佈置

本主題說明通道起始程式統計資料 (SMF 類型 115 子類型 231) 記錄的佈置。

## 自行定義區段

通道起始程式統計資料的自行定義區段遵循標準 SMF 標頭。它以標準三元組格式建構。三項值的格式在結構 qwsx 的 C 程式設計語言標頭檔 thlqual.SCSQC370(CSQDSMFC)及組譯器巨集 thlqual.SCSQMACS(CSQDQWSX)中有說明。

第 310 頁的表 39 顯示自我定義區段的格式。

偏移:Dec	偏移:十六進位	類型	長度	姓名	說明
0	0	整數	4	QWSX0PSO	從 SMF 記錄開始到第一個檢測標準標頭 (QWHS) 的偏移
4	4	整數	2	QWSX0PSL	QWHS 的長度
6	6	整數	2	QWSX0PSN	QWHS 實例數
8	8	整數	4	QWSX0R1O	從 SMF 記錄開始到第一個通道起始程式控制資訊區塊 (QCCT) 的偏移
12	C	整數	2	QWSX0R1L	QCCT 的長度
14	E	整數	2	QWSX0R1N	QCCT 實例數
16	10	整數	4	QWSX0R2O	從 SMF 記錄開始到第一個分派器作業區塊 (QCT_DSP) 的偏移
20	14	整數	2	QWSX0R2L	QCT_DSP 的長度
22	16	整數	2	QWSX0R2N	QCT_DSP 的實例數
24	18	整數	4	QWSX0R3O	從 SMF 記錄開始到第一個配接卡作業區塊的偏移 (QCT_ADP)
28	1C	整數	2	QWSX0R3L	QCT_ADP 的長度
30	1E	整數	2	QWSX0R3N	QCT_ADP 的實例數
32	20	整數	4	QWSX0R4O	從 SMF 記錄開始到第一個 SSL 作業區塊 (QCT_SSL) 的偏移
36	24	整數	2	QWSX0R4L	QCT_SSL 的長度
38	26	整數	2	QWSX0R4N	QCT_SSL 實例數
40	28	整數	4	QWSX0R5O	從 SMF 記錄開始到第一個 DNS 作業區塊 (QCT_DNS) 的偏移
44	2C	整數	2	QWSX0R5L	QCT_DNS 的長度
46	2E	整數	2	QWSX0R5N	QCT_DNS 的實例數

通常一個記錄包含所有資料。如果有大量分派器、配接器或 SSL 作業，則會將資料分割為多筆記錄。

如果發生此情況，則部分類型作業的實例計數可能為零，且一組作業的相關資訊可以分散在多筆記錄中。通道起始程式控制資訊區塊 (QCCT) 僅存在於第一筆記錄中。例如，資料可以在兩個 SMF 記錄之間分割，如下所示：

計數	第一筆記錄	最後一個記錄
QWHS	1	1

計數	第一筆記錄	最後一個記錄
QCCT	1	0
QCT_DSP	50	5
QCT_ADP	0	10
QCT_SSL	0	3
QCT_DNS	0	1

此範例顯示在 SMF 間隔期間，有 55 個分派器 TCB 在執行中。

## 設備測試標準標頭 (QWHS)

QWHS 的格式在結構 qwhs (位於 C 程式設計語言標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 中) 及組譯器巨集 `thlqual.SCSQMACS(CSQDQWHS)` 中有說明。它包含下列與通道起始程式 SMF 115 記錄相關的索引鍵欄位:

姓名	長度	說明
QWHSNDA	1 個位元組	自行定義區段數
QWHSSSID	4 個位元組	子系統名稱
QWHS SMFC	1 位元	指出是否有多筆 SMF 記錄包含此間隔的資訊。如果此位元已開啟，則會在進一步 SMF 記錄中繼續此間隔的資訊。如果此位元關閉，則這是最後一個或唯一記錄。QWHSSSID 中的子系統 ID，以及 QWHSTIME 中的 SMF 間隔開始時間，可用來分組相同間隔的多筆記錄。
QWHSTIME	8 個位元組	間隔開始的當地時間 (STCK 格式)
QWHS DURN	8 個位元組	從間隔開始到間隔結束的持續時間 (STCK 格式)
QWHS STCK	8 個位元組	以 UTC 格式表示的間隔結束 (STCK 格式)

## 通道起始程式統計資料記錄

請使用本主題作為通道起始程式統計資料資料記錄的參照。

通道起始程式統計資料記錄的格式包含兩個部分:

- 第一部分是通道起始程式控制資訊區塊，在組譯器巨集 `thlqual.SCSQMACS(CSQDQCCT)` 中說明。如需進一步資訊，請參閱第 312 頁的『通道起始程式控制資訊區塊』。
- 第二個部分是通道起始程式作業區塊，如組譯器巨集 `thlqual.SCSQMACS(CSQDQCTA)` 中所說明。

通道起始程式作業區塊包含 CHINIT 內四種作業類型的相關資訊。如需進一步資訊，請參閱:

- 第 312 頁的『分派器作業』
- 第 313 頁的『配接器作業』
- 第 314 頁的『網域名稱伺服器 (DNS) 作業』
- 第 315 頁的『SSL 作業』

每一項作業包括:

- 作業在間隔內處理要求所花費的經歷時間 (*qcteltm*)
- 作業在間隔中使用的 CPU 時間，由處理要求時使用的 CPU 及要求之間使用的 CPU 組成 (*qctcptm*)

- 此作業在間隔內的等待時間總計 (*qctwttm*)
- 間隔中的要求數 (*qctreqn*)

您可以使用此資訊來查看作業的忙碌程度，並根據分析判斷是否需要新增更多作業。

對於 TLS 及 DNS 作業，還包括最長要求 (*qctlgdu*, *qctlsdu*) 的持續時間，以及發生此狀況的當日時間 (*qctlgdm*, *qctlsdm*)。

這些可能有助於識別通道要求花費較長時間的時間。例如，前往企業外部伺服器的 DNS 查閱要求會花費秒而非毫秒。

CPU 時間 (*qctcptm*) 值包括作業所耗用的所有 CPU，包括處理要求及處理要求之間的 CPU。經歷時間 (*qctelmt*) 值僅包括處理要求時的時間。這表示 CPU 時間可能大於經歷時間。

下列作業中的範例統計資料已使用 IBM MQ SupportPac MP1B 進行格式化。

這兩個組件也會在 C 程式設計語言標頭檔 `th1qua1.SCSQC370(CSQDSMFC)` 中說明。請注意，C 中的欄位名稱都是小寫，例如 *qcct*、*qct\_adp*。

### 通道起始程式控制資訊區塊

請使用本主題作為通道起始程式控制資訊區塊的參照。

通道起始程式控制資訊區塊包含此 CHINIT 的基本資訊，包括：

- CHINIT 工作名稱 (*qcctjobn*)
- QSG 名稱 (如果它位於佇列共用群組中) (*qcctqsgn*)
- 已使用的現行通道尖峰數 (*qcctnoccc*)
- 作用中通道的尖峰使用數 (*qcctnoac*)
- MAXCHL-允許的現行通道數上限 (*qcctmxcc*)
- ACTCHL-允許的作用中通道數上限 (*qcctmxac*)
- TCPCHL-允許的 TCP/IP 通道上限 (*qcctmxtp*)
- LU62CHL -允許的 LU62 通道上限 (*qcctmxlu*)
- 延伸專用區域中 CHINIT 使用的儲存體 (*qcctstus*)。CHINIT 工作日誌中的 [CSQX004I](#) 訊息也會提供此資訊。

通道起始程式控制資訊區塊的格式在結構 *qcct* 的 C 程式設計語言標頭檔 `th1qua1.SCSQC370(CSQDSMFC)` 中及組譯器巨集 `th1qua1.SCSQMACS(CSQDQCCS)` 中有說明。

您可以使用此資訊來查看作用中通道數是否接近所配置的最大值。請注意，現行及作用中通道的數目是建立記錄時的值。因此，在兩個間隔之間，作用中的通道數可能超過此數目。

## SMF 資料的通道資訊

以下是 SMF 資料的通道資訊範例：

```
MVCA,MQPV,2014/03/18,13:00:00,VRM:800,
From 2014/03/18,12:45:00.015222 to 2014/03/18,13:00:00.083630 duration 900.068408 seconds
Peak number used of current channels..... 1
Peak number used of active channels ..... 1
MAXCHL. Max allowed current channels.....9999
ACTCHL. Max allowed active channels.....9999
TCPCHL. Max allowed TCP/IP channels.....9999
LU62CHL. Max allowed LU62 channels..... 200
Storage used by Chinit..... 436MB
```

您可以監視儲存體用量，並查看值是否呈上升趨勢。如果使用的總計接近可用的儲存體總計，則您可能已用完儲存體，因此可能無法支援更多通道。

如果作用中現行通道數趨於通道數上限，您可能需要增加通道數上限。

### 分派器作業

本主題包含分派器作業統計資料的範例資料，以及如何解譯資料的相關資訊。

分派器作業區塊的格式在結構 `qct_dsp` 的 C 程式設計語言標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 及組譯器巨集 `thlqual.SCSQMACS(CSQDQCTA)` 中有說明。

## 範例資料

Task	Type	Requests	Busy %	CPU used Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	DISP	26587	0.4	0.592463	0.1	22	127
1	DISP	26963	0.3	0.588092	0.1	22	112
2	DISP	864329	2.7	2.545668	0.3	3	28
3	DISP	26875	0.4	0.590825	0.1	22	120
4	DISP	26874	0.4	0.603285	0.1	22	123
Summ	DISP	971628	0.8	4.920332	0.1	5	38

範例資料顯示有五個分派器。通道在啟動時與分派器相關聯。將通道配置給分派器時，通道起始程式會嘗試將工作分散到所有分派器。此範例顯示一個分派器正在處理比其他分派器更多的要求。這是正常的，因為部分通道可能會停止，因此分派器會處理較少的通道，而部分通道可能比其他通道更忙碌。

- 分派器已使用 CPU 4.9 秒。
- CPU 的平均要求已使用 5 微秒，並花費 38 微秒經歷時間。
- 分派器是用來透過通訊網路傳送及接收資料，這通常不會相依於外部事件。因此，平均經歷時間應該接近所使用的平均 CPU 時間。CPU 時間 (`qctcptm`) 值包括作業所耗用的所有 CPU，包括處理要求及處理要求之間的 CPU。

經歷時間 (`qcteltm`) 值僅包括處理要求時的時間。這表示 CPU 時間可能大於經歷時間。如果 CHINIT 因缺少 CPU 而延遲，則與未延遲 CPU 的 CHINIT 相比，平均經歷時間與平均 CPU 時間的比例會大得多。

- 每個要求使用的平均 CPU 取決於訊息資料流量。例如，較大的訊息所使用的 CPU 比較小的訊息所使用的 CPU 更多。

欄位的計算方式如下：

- 持續時間: `qwhs.qwhsdurn`
- 要求: `qctreqn`
- 忙碌%: `qcteltm` 及持續時間
- 使用的 CPU: `qctcptm`
- CPU%: `qctcptm` 及持續時間
- 平均 CPU: `qctcptm` 及 `qctreqn`
- 平均 ET: `qcteltm` 及 `qctreqn`

通常，分派器數目應該小於或等於 LPAR 中的處理器數目。如果您在 LPAR 中具有比處理器更多的分派器，則它們可能會競爭 CPU 資源。如需調整系統的相關資訊，請參閱 [SupportPac MP16](#)。

通道與分派器具有親緣性，因此您可能會發現部分分派器處理比另一個分派器更多的要求。

您可以使用 `ALTER QMGR CHIDISPS ()` 指令來變更使用的分派器數目。下次啟動通道起始程式時，任何變更都會生效。

## 配接器作業

本主題包含配接器作業統計資料的範例資料，以及如何解譯資料的相關資訊。

配接器作業區塊的格式在結構 `qct_adp` 的 C 程式設計語言標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 中，以及在組譯器巨集 `thlqual.SCSQMACS(CSQDQCTA)` 中有說明。

## 範例資料

Task	Type	Requests	Busy %	CPU used Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	ADAP	470297	10.2	41.290670	4.6	88	194
1	ADAP	13907	0.6	1.589428	0.2	114	365
2	ADAP	2517	0.2	0.185325	0.0	74	746

3, ADAP,	1095,	0.1,	0.085774,	0.0,	78,	907
4, ADAP,	535,	0.1,	0.040743,	0.0,	76,	947
5, ADAP,	220,	0.0,	0.016228,	0.0,	74,	1175
6, ADAP,	82,	0.0,	0.005521,	0.0,	67,	1786
7, ADAP,	80,	0.0,	0.004248,	0.0,	53,	1160
Summ, ADAP,	488733,	1.4,	43.217938,	0.6,	88,	205

欄位的計算方式如下:

- 持續時間: *qwhs.qwhsdurn*
- 要求: *qctreqn*
- 忙碌%: *qcteltm* 及持續時間
- 已使用的 CPU: *qctcptm*
- CPU%: *qctcptm* 及持續時間
- 平均 CPU: *qctcptm* 及 *qctreqn* 平均值
- ET: *qcteltm* 和 *qctreqn*

此範例顯示有八個配接器作業。

### 配接卡號碼 0

- 已處理大部分要求 (488733 項中的 470297 項)
- 在間隔中忙碌 10.2%
- 已用 CPU 41.3 秒

### 整體

每個要求的平均 CPU 是 CPU 的 88 微秒，並花費 205 微秒

配接器會處理 IBM MQ 要求。其中部分要求可能等待 (例如，在確定期間等待日誌 I/O)，因此每個要求的平均「經歷時間」意義不大。

CPU 時間 (*qctcptm*) 值包括作業所耗用的所有 CPU，包括處理要求及處理要求之間的 CPU。經歷時間 (*qcteltm*) 值僅包括處理要求時的時間。這表示 CPU 時間可能大於經歷時間。

當發出 IBM MQ 要求時，會使用第一個可用的配接器作業。

- 如果至少有一個配接卡很少使用 (小於 1%) 忙碌，則您有足夠的配接卡。
- 如果未使用至少一個配接卡，則您已定義足夠的配接卡。
- 如果已使用所有配接卡，則可能需要配置更多配接卡。
- 如果所有配接卡都已使用，而且大部分間隔都在忙碌中，則您需要配置更多配接卡。

您可以使用 ALTER QMGR CHIADAPS () 指令來變更使用的配接卡數目。下次啟動通道起始程式時，任何變更都會生效。



**小心:** 如果有太多配接卡在一小組佇列上運作，您可能會在佇列管理程式內取得競用。

### 相關參考

ALTER QMGR

### z/OS 網域名稱伺服器 (DNS) 作業

本主題包含 DNS 作業統計資料的範例資料，以及如何解譯資料的相關資訊。

DNS 作業區塊的格式在結構 *qct\_dns* (位於 C 程式設計語言標頭檔 *thlqual.SCSQC370(CSQDSMFC)* 中) 及組譯器巨集 *thlqual.SCSQMACS(CSQDQCTA)* 中有說明。

### 範例資料

Task,	Type,	Requests,	Busy %,	CPU used,	CPU %,	"avg CPU",	"avg ET",	longest,
date,		time		Seconds,		uSeconds,	uSeconds,	uSeconds,
0,	DNS,	14002,	0.0,	0.122578,	0.0,	9,	11,	463,
								2014/03/18,

```
12:56:33.987671
Summ, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
```

通道起始程式使用單一 DNS 作業。此範例顯示作業已處理 14002 要求，平均而言，該要求已使用 9 微秒的 CPU，並花費 11 微秒的經歷時間。

最長 DNS 要求花費 463 微秒經歷時間，此發生在當地時間 12:56:33。

欄位的計算方式如下：

- 持續時間: *qwhs.qwhsdurn*
- 要求: *qctreqn*
- 忙碌%: *qcteltm* 及持續時間
- 已使用的 CPU: *qctcptm*
- CPU%: *qctcptm* 及持續時間
- 平均 CPU: *qctcptm* 及 *qctreqn*
- 平均 ET: *qcteltm* 和 *qctreqn*
- 最長: *qctlgdu*
- 最長時間: *qctlgtm*

DNS 作業可以離開您的企業，以查閱與名稱相關聯的 IP 位址。如果平均經歷時間明顯超過所使用的平均 CPU 時間，您可能會有一些長要求。

如果最長要求時間的值無法接受，您應該與網路團隊合作，以調查您有長要求的原因。可能是您的連線中有無效名稱。

如果 DNS 作業忙碌達 25% 的持續時間，請考慮進一步調查原因。

CPU 時間 (*qctcptm*) 值包括作業所耗用的所有 CPU，包括處理要求及處理要求之間的 CPU。經歷時間 (*qcteltm*) 值僅包括處理要求時的時間。這表示 CPU 時間可能大於經歷時間。

註：對 DNS 作業的要求不是 DNS 查閱，因此您可能有大於零的要求數目，但沒有最長的要求資訊。

## z/OS SSL 作業

本主題包含 SSL 作業統計資料的範例資料，以及如何解譯資料的相關資訊。

SSL 作業區塊的格式在結構 *qct\_ssl* 的 C 程式設計語言標頭檔 *thlqual.SCSQC370* (CSQDSMFC) 中，以及在組譯器巨集 *thlqual.SCSQMACS* (CSQDQCTA) 中有說明。

## 範例資料

```
Task, Type, Requests, Busy %, CPU used, CPU %, "avg CPU", "avg ET", longest,
date, time
0, SSL, 3112, 1.2, 0.248538, 0.3, 80, 362, 8864, 2014/03/18,
12:46:40.237697
1, SSL, 3070, 1.2, 0.245433, 0.3, 80, 359, 4714, 2014/03/18,
12:46:18.938022
2, SSL, 3170, 1.2, 0.255557, 0.3, 81, 362, 7273, 2014/03/18,
12:46:35.358145
3, SSL, 3060, 1.2, 0.246542, 0.3, 81, 365, 13164, 2014/03/18,
12:46:44.514045
4, SSL, 3120, 1.3, 0.251927, 0.3, 81, 373, 22438, 2014/03/18,
12:46:22.134123
Summ, SSL, 15532, 1.2, 1.247998, 0.3, 80, 364, 22438, 2014/03/18,
12:46:22.134123
```

此範例資料顯示平均要求花費 364 微秒。最長要求是針對 SSL 作業 4，耗時 22,438 微秒，並在當地時間 12:46:22.134123 發生。

欄位的計算方式如下：

- 持續時間: *qwhs.qwhsdurn*

- 要求: *qctreqn*
- 忙碌%: *qcteltm* 及持續時間
- 已使用的 CPU: *qctcptm*
- CPU%: *qctcptm* 及持續時間
- 平均 CPU: *qctcptm* 及 *qctreqn*
- 平均 ET: *qcteltm* 和 *qctreqn*
- 最長: *qctlsdu* 最長時間: *qctlstm*

執行中通道與 SSL 作業相關聯，與通道與分派器相關聯的方式類似。SSL 作業可以使用 LPAR 可用的加密輔助處理器。因此，經歷時間可以包括花費在輔助處理器上的時間。您應該監視一整天的平均經歷時間。如果此時間在尖峰期間大幅增加，您應該與 z/OS 系統程式設計師一起工作，因為您的輔助處理器可能過度使用。

如果 SSL 作業忙碌的時間佔間隔的很大比例，增加 SSL 作業數目可能會有幫助。如果 SSL 作業正在等待外部資源 (例如輔助處理器)，則增加 SSL 作業數目幾乎沒有作用。

您可以使用 ALTER QMGR SSLTASKS () 指令來變更使用的 SSL 作業數。下次啟動通道起始程式時，任何變更都會生效。

CPU 時間 (*qctcptm*) 值包括作業所耗用的所有 CPU，包括處理要求及處理要求之間的 CPU。經歷時間 (*qcteltm*) 值僅包括處理要求時的時間。這表示 CPU 時間可能大於經歷時間。

## 相關參考

[ALTER QMGR](#)

## 解譯 IBM MQ 帳戶資料

IBM MQ 帳戶資料會寫入 SMF 類型 116 記錄。請利用這個主題來參照不同類型的統計資料記錄。

可以收集下列子類型的 IBM MQ 帳戶資訊：

**0**

訊息管理程式結算記錄 (處理 IBM MQ API 呼叫所花費的處理器時間，以及 MQPUT 和 MQGET 呼叫數)。當具名作業與 IBM MQ 中斷連線時，會產生此資訊，因此記錄內包含的資訊可能涵蓋數小時。

**1**

每一個作業的帳戶資料 (在執行緒及佇列層次)。

**2**

其他佇列層次結算資料 (如果作業所使用的佇列數超過子類型 1 記錄中可容納的佇列數)。

**10**

通道的帳戶資料。

**註：** STATCHL 通道屬性及 STATACLS 佇列管理程式屬性可以啟用或抑制特定通道的結算資訊。

請注意：

- 使用追蹤類別 (1) 產生子類型 0
- 使用追蹤類別 (3) 產生子類型 1 及 2
- 使用帳戶追蹤類別 (4) 產生子類型 10

子類型

## SMF 類型 116 記錄的佈置

請利用這個主題來參照 SMF 類型記錄的格式。

SMF 記錄的標準佈置包含三個部分：

### SMF 標頭

提供記錄本身的格式、識別及時間和日期資訊。

### 自行定義區段

定義 SMF 記錄內個別資料記錄的位置和大小。

## 資料記錄

您要分析來自 IBM MQ 的實際資料。

如需 SMF 記錄格式的相關資訊，請參閱 [z/OS MVS 系統管理機能 \(SMF\)](#)。

## SMF 標頭

第 317 頁的表 42 顯示 SMF 記錄標頭的格式 (SM116)。

偏移:Dec	偏移:十六進位	類型	Len	名稱	說明	範例
0	0	結構	28	SM116	SMF 記錄標頭。	
0	0	整數	2	SM116LEN	SMF 記錄長度。	01A4
2	2		2		保留。	
4	4	整數	1	SM116FLG	系統指示器。	5E
5	5	整數	1	SM116RTY	記錄類型。SMF 記錄類型，對於 IBM MQ 帳戶記錄，這一律是 116 (X'74')。	74
6	6	整數	4	SM116TME	SMF 移動記錄的時間。	00356124
10	A	整數	4	SM116DTE	SMF 移動記錄的日期。	0100223F
14	E	字元	4	SM116SID	z/OS 子系統 ID。定義收集記錄所在的 z/OS 子系統。	D4E5F4F1 (MV41)
18	12	字元	4	SM116SSI	IBM MQ 子系統 ID。	D4D8F0F7 (MQ07)
22	16	整數	2	SM116STF	記錄子類型。	0000
24	18	字元	3	SM116REL	IBM MQ 版本。	F9F1F0 (910)
27	1B		1		保留。	
28	1C	字元	0	SM116END	SMF 標頭的結尾和自我定義區段的開頭。	

註: 右側直欄中的 (十六進位) 值與 [第 319 頁的圖 22](#) 相關。

## 自行定義區段

SMF 記錄的自行定義區段會告訴您尋找帳戶記錄的位置、時間長度，以及該記錄類型的重複次數 (具有不同的值)。自我定義區段會在標頭後面，從 SMF 記錄開始算起的固定偏移。

每一個自行定義區段都指向帳戶相關資料。第 317 頁的表 43 彙總從 SMF 記錄標頭開始的偏移。

記錄子類型 (SMF116STF)	會計資料來源	自行定義區段的偏移		請參閱...
		十二月	十六進位碼	
全部	共用標頭	28	X'1C'	第 320 頁的『共用 IBM MQ SMF 標頭』

表 43: 自我定義區段的偏移 (繼續)

記錄子類型 (SMF116STF)	會計資料來源	自行定義區段的偏移		請參閱...
		十二月	十六進位碼	
0	訊息管理程式	44	X'2C'	第 321 頁的『訊息管理程式資料記錄』
1	執行緒識別記錄	36	X'24'	第 322 頁的『執行緒層次及佇列層次資料記錄』
1	執行緒層次帳戶	44	X'2C'	第 322 頁的『執行緒層次及佇列層次資料記錄』
1	佇列層次帳戶	52	X'34'	第 322 頁的『執行緒層次及佇列層次資料記錄』。只有在作業相關資訊 (WTAS) 結構中的 WTASWQCT 欄位為非零時，才會呈現此區段。
2	執行緒識別記錄	36	X'24'	第 322 頁的『執行緒層次及佇列層次資料記錄』
2	佇列層次帳戶	44	X'2C'	第 322 頁的『執行緒層次及佇列層次資料記錄』
10	通道統計作業			第 325 頁的『通道統計資料記錄』

註: 其他自行定義區段是指僅供 IBM 使用的資料。

每一個自我定義區段都有兩個完整單字，且具有下列格式:

```
sssssssl1111nnnn
```

其中:

**SSSSSSS**

包含 SMF 記錄開頭偏移的全字組。

**利爾**

提供此資料記錄長度的半字組。

**nnnn**

提供此 SMF 記錄中資料記錄數的半字。

第 319 頁的圖 22 顯示 SMF 類型 116 記錄的組件範例。左側直欄中的數字代表從記錄開始算起的偏移 (十六進位)。每一行對應於 16 個位元組的資料，其中每一個位元組是兩個十六進位字元，例如 0C。右側直欄中的字元代表每一個位元組的可列印字元。不可列印字元由句點 (.) 字元顯示。

在此範例中，SMF 標頭中的替代欄位會 畫底線，以協助您查看它們；請參閱第 317 頁的表 42 以識別它們。其中一個訊息管理程式結算資料記錄的自行定義區段 (位於第 317 頁的表 43 中給定的偏移) 以 **粗體** 顯示。

```

000000 01A40000 5E740035 61240100 223FD4E5 *...;.../.....MV*
000000 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600.....*
000000 00700001 00000054 00B00001 00000104 *.....*
000000 00300001 00000000 00000000 00000000 *.....*
000000 00000000 00000000 00000000 00000000 *.....*

```

圖 22: SMF 記錄 116 的一部分，顯示標頭和自我定義區段

訊息管理程式結算資料類型的自行定義區段位於 SMF 記錄開頭的偏移 X'2C' 處，並包含下列資訊：

- 訊息管理程式帳戶資料的偏移位於從 SMF 記錄開始算起的 X'00000104' 位元組。
- 此訊息管理程式記錄長度為 X'0030' 位元組。
- 有一筆記錄 (X'0001')。

註：一律在自行定義區段中使用偏移來尋找帳戶記錄。

### z/OS 正在處理類型 116 SMF 記錄

請使用本主題作為處理類型統計記錄格式的參照。

必須處理您從 SMF 收集的任何帳戶資料，才能擷取有用資訊。當您處理資料時，請驗證記錄來自 IBM MQ，且它們是您預期的記錄。

驗證下列欄位的值：

- SM116RTY，SMF 記錄號碼 = X'74' (116)
- SM116STF(記錄子類型) 必須是 0000、0001、0002 或 0010

不支援從作用中 SMF 資料集 (或 SMF 日誌串流) 讀取。您必須使用 SMF 程式 IFASMFDP (如果正在使用日誌串流，則為 IFASMF DL)，將 SMF 記錄傾出至循序資料集，以便可以處理它們。如需相關資訊，請參閱第 292 頁的『使用系統管理機能』。

如需結構及欄位的詳細資料，請參閱 IBM MQ SupportPac MP1B。

有一個稱為 CSQ4SMFD 的 C 範例程式，它會從循序資料集列印 SMF 類型 115 和 116 記錄的內容。程式在 thlqual.SCSQC37S 中以原始檔形式提供，在 thlqual.SCSQLOAD 中以執行檔格式提供。thlqual.SCSQPROC(CSQ4SMFJ) 中提供範例 JCL。

您需要以 SMF 資料集名稱來更新 SMFIN DD 卡。請使用 z/OS 指令 '/D SMF' 來顯示資料集的名稱，且您需要以輸出資料集的名稱來更新 DUMPOUT DD 卡。

您也需要指定所需的 START 及 END 時間。

下列範例 JCL 會從 SMF 資料集擷取 SMF 記錄，並將它們傾出至 SMFOUT 資料集：

```

//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=
//SMFIN DD DSN=xxxxxxx.MANA,DISP=SHR
//SMFOUT DD DSN=xxxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
INDD(SMFIN,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*

```

下列範例 JCL 會從 LSNAME 中指定的 SMF 日誌串流擷取 SMF 記錄，並將它們傾出至 SMFOUT 資料集：

```

//SMFDUMP EXEC PGM=IFASMF DL,REGION=0M
//SYSPRINT DD SYSOUT=*
//SMFOUT DD DSN=xxxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
LSNAME(IFASMF.MQ,OPTIONS(DUMP))

```

OUTDD(SMFOUT,TYPE(116))  
 OUTDD(SMFOUT,TYPE(115))  
 START(1159) END(1210)  
 /\*

### z/OS 共用 IBM MQ SMF 標頭

請利用這個主題來參照一般 IBM MQ SMF 標頭類型統計記錄。

此記錄的格式在 第 320 頁的表 44 及組譯器巨集 thlqual.SCSQMACS(CSQDQWHS) 和 thlqual.SCSQMACS(CSQDQWHC) 及 C 標頭檔 thlqual.SCSQC370(CSQDSMFC) 中有說明。C 中的欄位名稱都是小寫，例如 qwhs、qwhsnsda。

QWHS 資料包括子系統名稱。對於子類型 1 記錄，它也會顯示是否存在佇列層次帳戶記錄。如果 QWHSNSDA 欄位為 3 或更小，則沒有，且未設定對應的自我定義區段 (偏移 X'34')。

QWHC 資料提供使用者的相關資訊 (例如，使用者 ID (QWHCAID) 及應用程式類型 (QWHCATYP))。僅針對子類型 0 記錄完成 Q 西隧區段。在子類型 1 及 2 記錄的執行緒識別記錄中存在對等資訊。

表 44: 一般 IBM MQ SMF 標頭記錄 QWHS 的結構

偏移:Dec	偏移:十六進位	類型	長度	名稱	說明
0	0	結構	128	QWHS	
0	0		6		已保留
6	6	字元	1	QWHSNSDA	SMF 記錄中自行定義區段的數目
7	7		5		已保留
12	C	字元	4	QWHSSSID	子系統名稱
16	10		24		已保留
40	28	字元	8	QWHCAID	與 z/OS 工作相關聯的使用者 ID
48	30	字元	12	QWHCCV	執行緒交互參照
60	3C	字元	8	QWHCCN	連線名稱
68	44		8		已保留
76	4C	字元	8	QWHCOPID	與交易相關聯的使用者 ID
84	54	整數	4	QWHCATYP	連接系統的類型 (1 =CICS、2=Batch 或 TSO、3 =IMS 控制區域、4 =IMS MPP 或 BMP、5=Command 伺服器、6=Channel 起始程式、7=RRS 批次)
88	58	字元	22	QWHCTOKN	帳戶記號設定為使用者的 z/OS 帳戶資訊
110	6E	字元	16	QWHCNID	網路 ID
126	7E		2		已保留

### z/OS 結合 CICS 與 IBM MQ 效能資料

請利用這個主題來參照 IBM MQ 與 CICS 效能資料的組合。

使用一般 IBM MQ SMF 標頭類型統計記錄區段 QWHCTOKN，將 CICS 類型 110 SMF 記錄與 IBM MQ 類型 116 SMF 記錄產生關聯。

CICS 會針對每一個 CICS 作業產生 LU6.2 工作單元記號。記號用來產生在子類型零記錄的相關性標頭中寫入 QWHCTOKN 的帳戶記號。

詳細資料也會寫入子類型 1 及 2 記錄中的 WTIDACCT 區段。帳戶記號會啟用交易的 CICS 與 IBM MQ 效能資料之間的相關性。

## **z/OS** 執行緒交互參照資料

請利用這個主題來參照執行緒交互參照類型帳戶記錄的格式。

執行緒交互參照 (QWHCCV) 欄位中資料的解譯有所不同。這取決於資料與下列項目的相關：

- CICS 連線 (QWHCATYP=1)-請參閱 [第 321 頁的表 45](#)
- IMS 連線 (QWHCATYP=3 或 4)-請參閱 [第 321 頁的表 46](#)
- 批次連線 (QWHCATYP=2 或 7)-此欄位由二進位零組成
- 其他-沒有有意義的資料

偏移 :Dec	偏移: 十六進位	類型	長度	說明
48	30	字元	4	CICS 執行緒號碼。
52	34	字元	4	CICS 交易名稱。
56	38	整數	4	CICS 作業號碼。

部分項目包含空白字元。這些適用於作業，而非特定交易。

偏移 :Dec	偏移: 十六進位	類型	長度	說明
48	30	字元	4	IMS 分割區規格表 (PST) 區域 ID。
52	34	字元	8	IMS 程式規格區塊 (PSB) 名稱。

## **z/OS** 訊息管理程式資料記錄

請利用這個主題來參照訊息管理程式帳戶記錄的格式。

訊息管理程式是 IBM MQ 的元件，可處理所有 API 要求。訊息管理程式帳戶記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDQMAC) 中說明。

QMAC 資料提供處理 IBM MQ 呼叫所花費處理器時間的相關資訊，以及不同大小訊息的 MQPUT 及 MQGET 要求數的計數。

**註：**單一 IMS 應用程式可能會寫入兩筆 SMF 記錄。在此情況下，請新增兩筆記錄中的數字，以提供 IMS 應用程式的正確總計。

### 包含零處理器時間的記錄

有時會產生在 QMACCPUT 欄位中包含零處理器時間的記錄。當識別為 IBM MQ 的長時間執行作業終止或被停止帳戶追蹤提示輸出帳戶記錄時，即會發生這些記錄。這類作業存在於 CICS 配接卡及通道起始程式中 (適用於分散式佇列作業)。這些具有零處理器時間的作業數目取決於系統中有多少活動：

- 對於 CICS 配接卡，這可能導致最多九筆記錄的處理器時間為零。
- 對於通道起始程式，處理器時間為零的記錄數最多可以達到 Adapters + Dispatchers + 6 的總和，如佇列管理程式屬性中所定義。

這些記錄反映在作業下完成的工作量，可以忽略。

## z/OS 樣本子類型零統計記錄

請利用這個主題來參照子類型零帳戶記錄的格式。

第 322 頁的圖 23 顯示類型 116，子類型零 SMF 記錄。在此圖中，SMF 記錄標頭和 QMAC 帳戶資料記錄會畫底線。自我定義區段以粗體顯示。

```
000000 01A40000 5E740035 61240100 223FD4E5 *...;.../.....MV*
000010 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000020 00700001 00000054 00B00001 00000104 *.....*
000030 00300001 00000000 00000000 00000000 *.....*
000040 00000000 00000000 00000000 00000000 *.....*
000050 00000000 B478AB43 9C6C2280 B478AB47 *.....%.....*
000060 9DB47E02 00000000 04C0F631 00000001 *..=.....}6.....*
000070 9880E72D 00000000 014D9540 00000000 *..X.....(. ..*
000080 08480C80 00000010 40404040 40404040 *.....*
000090 00000000 00000000 00000051 00000000 *.....*
0000A0 00000000 00000000 00000000 00000000 *.....*
0000B0 00000000 00000000 00000000 00000000 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 00000000 *.....*
000100 00000000 D4140030 D8D4C1C3 00000000 *...M...QMAC...*
000110 689C738D 00000050 00000000 00000050 *.....&.....&*
000120 0000000A 00000000 00000000 00000000 *.....*
000130 00000000 0024011A 00030710 02DAACF0 *.....0*
```

圖 23: SMF 類型 116 子類型零記錄範例

## z/OS 執行緒層次及佇列層次資料記錄

請利用這個主題來參照執行緒層次和佇列層次統計記錄的格式。

使用 IBM MQ 來收集每一個作業的執行緒層次統計記錄。對於每一項作業，當作業完成時，會將執行緒層次統計資料記錄寫入 SMF。若為長時間執行的作業，也會以 CSQ6SYSP 系統參數巨集 (或由系統 SMF 統計資料播送) 的 STATIME 參數所設定的統計資料間隔來寫入資料，前提是作業正在執行前次收集統計資料的時間。此外，還會收集作業開啟之每一個佇列的帳戶資訊。自前次寫入執行緒層次帳戶記錄後，會針對作業已使用的每一個佇列寫入佇列層次帳戶記錄。

如果您在啟動帳戶追蹤時指定類別 3，則會產生執行緒層次及佇列層次帳戶記錄。例如，使用下列指令：

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
```

執行緒層次統計資訊會寫入 SMF 類型 116 子類型 1 記錄，後面接著佇列層次記錄。如果作業開啟許多佇列，則會將進一步的佇列資訊寫入一或多個 SMF 類型 116 子類型 2 記錄。執行緒識別控制區塊包含在每一個子類型 1 和 2 記錄中，可讓您將每一個記錄與正確的作業相關聯。一般而言，每一個 SMF 記錄中的佇列層次記錄數上限約為 45。

執行緒層次統計記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDWTAS) 中說明。佇列層次統計記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDWQ) 中說明。執行緒識別記錄的格式在組譯器巨集 thlqual.SCSQMACS(CSQDWTID) 中說明。所有這些記錄也在 C 標頭檔 thlqual.SCSQC370(CSQDSMFC) 中說明。C 中的欄位名稱都是小寫，例如 wtas、wtasshex。

## z/OS 通道名稱的意義

請使用本主題作為通道名稱意義的參照。

WTID 中的通道名稱如下列範例所示建構。在此範例中，存在從佇列管理程式 QM1 到佇列管理程式 QM2 的傳送端通道。

下表說明通道名稱的意義。

表 47: 通道名稱的意義

欄位名稱	意義	範例
對於佇列管理程式 QM1，傳送端通道已設定下列欄位:		
WTIDCCN	工作名稱	QM1CHIN
WTIDCHL	通道名稱	QM1.QM2
WTIDCHLC	這定義在通道的 CONNAME 中	WINMVS2B(2162)
對於佇列管理程式 QM2，接收端通道已設定下列欄位:		
WTIDCCN	工作名稱	QM2CHIN
WTIDCHL	通道名稱	QM1.QM2
WTIDCHLC	頻道的來源	9.20.101.14

**z/OS** 樣本子類型 1 和子類型 2 記錄

請利用這個主題來參照子類型 1 和子類型 2 帳戶記錄的格式。

第 323 頁的圖 24 和 第 323 頁的圖 25 顯示 SMF 類型 116、子類型 1 和子類型 2 記錄的範例。已為開啟 80 個佇列的批次工作建立這兩個帳戶記錄。因為開啟了許多佇列，所以子類型 2 記錄必須包含產生的所有資訊。

```

000000  703C0000 5E74002D 983B0100 229FD4E5  *...;.....MV*
000010  F4F1D4D8 F0F70001 F6F0F000 00006FCC  *41MQ07..600...?.*
000020  00700001 0000003C 00D00001 0000010C  *.....}*
000030  02C00001 000003CC 02400030 F70000D0  *..7..}*
000040  E6E3C9C4 00000000 00000000 00000040  *WTID..... *
.
.
000100  00000000 00000000 7F4A4BB8 F70102C0  *....."...7..}*
000110  E6E3C1E2 B4802373 0BF07885 7F4AE718  *WTAS.....0..".X.*

```

圖 24: SMF 類型 116、子類型 1 記錄範例

第一個自我定義區段從 X'24' 開始，在範例中為 **粗體**；X'0000003C' 是 WTID 資料記錄的偏移，X'00D0' 是 WTID 記錄的長度，而 X'0001' 是 WTID 記錄的數目。

第二個自行定義區段從 X'2C' 開始，且為 **斜體**；X'0000010C' 是 WTAS 資料記錄的偏移，X'02C0' 是 WTAS 記錄的長度，而 X'0001' 是 WTAS 記錄的數目。

第三個自行定義區段從 X'34' 開始，在範例中為 **粗體**；X'000003CC' 是第一個 WQST 資料記錄的偏移，X'0240' 是 WQST 記錄的長度，而 X'0030' 是 WQST 記錄的數目。

第 323 頁的圖 25 顯示 SMF 類型 116 子類型 2 記錄的範例。

```

000000  49740000 5E74002D 983B0100 229FD4E5  *...;.....MV*
000010  F4F1D4D8 F0F70002 F6F0F000 00004904  *41MQ07..600....*
000020  00700001 00000034 00D00001 00000104  *.....}*
000030  02400020 F70000D0 E6E3C9C4 00000002  *..7..}*WTID....*
.
.
000100  7F4A4BB8 F7020240 E6D8E2E3 00000001  *"...7.. WQST....*

```

圖 25: SMF 類型 116，子類型 2 記錄範例

第一個自我定義區段從 X'24' 開始，在範例中為 **粗體**；X'00000034' 是 WTID 資料記錄的偏移，X'00D0' 是 WTID 記錄的長度，而 X'0001' 是 WTID 記錄的數目。

第二個自行定義區段從 X'2C' 開始，且為 **斜體**；X'00000104' 是第一個 WQST 資料記錄的偏移，X'0240' 是 WQST 記錄的長度，而 X'0020' 是 WQST 記錄的數目。

第 324 頁的圖 26 顯示 SMF 類型 116 子類型 1 記錄的範例，其中未開啟任何佇列，因此沒有 WQST 記錄的自我定義區段。

```

000000      5E740039 4E9B0104 344FD4E5 * .....|MV*
000010  F4F1D4D8 F0F70001 F6F0F000 000003DC *41MQ07..600.....*
000020  00800001 00000034 00D00001 00000104 *.....*
000030  02D80001 F7000000 E6E3C9C4 00000002 *.Q..7...WTID...*
000040  C1F8C5C1 C4C5D740 C1F8C5C1 C4C54040 *A8EADEP A8EADE *
000050  40404040 40404040 00000000 00000000 * .....*
000060  40404040 40404040 4040          *          *

```

圖 26: 沒有 WQST 資料記錄的 SMF 類型 116 子類型 1 記錄範例

第一個自我定義區段從 X'24' 開始，在範例中為 **粗體**；X'00000034' 是 WTID 資料記錄的偏移，X'00D0' 是 WTID 記錄的長度，而 X'0001' 是 WTID 記錄的數目。

第二個自行定義區段從 X'2C' 開始，且為 **斜體**；X'0000010C' 是 WTAS 資料記錄的偏移，X'02D8' 是 WTAS 記錄的長度，而 X'0001' 是 WTAS 記錄的數目。

沒有說明 WQST 資料記錄的自行定義區段，相當於第 323 頁的圖 24 中的第三個自行定義區段。

## 通道起始程式 SMF 類型 116 記錄的佈置

本主題說明通道結算資料 (SMF 類型 116、子類型 10) 記錄的佈置。

### 自行定義區段

通道統計資料的自行定義區段遵循標準 SMF 標頭。它以標準三元組格式建構。三項值的格式在結構 qws5 的 C 程式設計語言標頭檔 thlqual.SCSQC370 (CSQDSMFC) 及組譯器巨集 thlqual.SCSQMACS (CSQDQWS5) 中有說明。

第 324 頁的表 48 顯示自我定義區段的格式。

偏移:Dec	偏移:十六進位	類型	長度	姓名	說明
0	0	整數	4	QWS50PSO	從 SMF 記錄開始到第一個檢測標準標頭 (QWHS) 的偏移
4	4	整數	2	QWS50PSL	QWHS 的長度
6	6	整數	2	QWS50PSN	QWHS 實例數
8	8	整數	4	QWS50R1O	從 SMF 記錄開始到第一個通道統計資料記錄 (QCST) 的偏移
12	C	整數	2	QWS50R1L	QCST 的長度
14	E	整數	2	QWS50R1N	QCST 實例數

## 設備測試標準標頭 (QWHS)

QWHS 的格式在結構 `qwhs` (位於 C 程式設計語言標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 中) 及組譯器巨集 `thlqual.SCSQMACS(CSQDQWHS)` 中有說明。它包含下列與通道起始程式 SMF 116 記錄相關的索引鍵欄位:

姓名	長度	說明
QWHSNDA	1 個位元組	自行定義區段數
QWHSSSID	4 個位元組	子系統名稱
QWHS SMFC	1 位元	指出是否有多筆 SMF 記錄包含此間隔的資訊。如果此位元已開啟,則會在進一步 SMF 記錄中繼續此間隔的資訊。如果此位元關閉,則這是最後一個或唯一記錄。
QWHSTIME	8 個位元組	間隔開始的當地時間 (STCK 格式)
QWHS DURN	8 個位元組	從間隔開始到間隔結束的持續時間 (STCK 格式)
QWHS STCK	8 個位元組	以 UTC 格式表示的間隔結束 (STCK 格式)

## z/OS 通道統計資料記錄

請使用本主題作為通道統計資料記錄的參照。

通道統計資料記錄的格式在組譯器巨集 `thlqual.SCSQMACS(CSQDQCST)` 中說明。該格式也在 C 程式設計語言標頭檔 `thlqual.SCSQC370(CSQDSMFC)` 中說明。請注意, C 中的欄位名稱都是小寫,例如 `qcst`。

通道結算資料提供每一個通道實例的狀態及統計資料相關資訊,包括:

- 平均網路時間 (`qcstntav`)
- 結束時的平均時間 (`qcstetav`)
- 通道批次資料限制 (`qcstcbd`)
- 通道批次間隔 (`qcstcbit`)
- 通道批次大小 (`qcstcbsz`)
- 通道分派器號碼 (`qcstdspn`)
- 通道處置 (`qcstchdp`)
- 通道名稱 (`qcstchnm`)
- 通道狀態 (`qcstchst`)
- 通道啟動時間 (`qcststrt`)
- 通道狀態收集時間 (`qcstcltm`)
- 通道停止時間 (`qcstludt`)
- 通道類型 (`qcstchty`)
- SSLCERTI (`qcstslcn`) 中的通用名稱 (CN)
- 壓縮率 (`qcstcpa`)
- 連線名稱 (`qcstcnm`)
- 現行共用交談數 (`qcstcscv`)
- DNS 解析時間 (`qcstdnrt`)
- STATCHL 參數的有效值 (`qcststcl`)
- 前次訊息時間 (`qcstlmst`)
- 網路時間上限 (`qcstntmx`)
- 結束時的時間上限 (`qcstetmx`)

- 最短網路時間 (*qcstntmn*)
- 結束時的最短時間 (*qcstetmn*)
- 遠端佇列管理程式或應用程式的名稱 (*qcstrqmn*)
- 批次數 (*qcstbatc*)
- 訊息資料的位元組數 (*qcstnbyt*)
- 持續訊息資料的位元組數 (*qcstnpby*)
- 針對訊息資料及控制資訊所接收的位元組數 (*qcstbyrc*)
- 針對訊息資料及控制資訊所傳送的位元組數 (*qcstbyst*)
- 完整批次數 (*qcstfuba*)
- 訊息數或 MQI 呼叫數 (*qcstnmsg*)
- 持續訊息數 (*qcstnmsg*)
- 放置重試次數 (*qcstptrc*)
- 變成空的傳輸佇列數 (*qcstqetc*)
- 接收的傳輸緩衝區數目 (**qcstbfrc**)
- 傳送的傳輸緩衝區數目 (*qcstbfst*)
- SSLPEER 中的序號 (*qcstslsn*)
- SSL CipherSpec (零表示未使用 TLS) (*qcstslcs*)
- 網路時間上限的日期和時間 (*qcstntdt*)
- 結束時的時間上限日期和時間 (*qcstetdt*)

請注意，對於通道統計欄位 *qcstetmn* (結束時的最短時間) 和 *qcstntmn* (網路時間下限)，這兩個欄位在未用時將起始設定為十六進位值 8FFFFFFF。

您可以使用此資訊來查看通道的傳輸量，如果實際批次接近限制、網路延遲、遠端系統的相關資訊、使用者結束程式的效能等等。

以下是已使用 IBM MQ SupportPac MP1B 格式化的通道統計資料範例。

可用的欄位是根據顯示通道狀態指令 (DIS CHS) 及平台上 IBM MQ (z/OS 除外) 的通道統計資料，以及一些其他欄位。

```
The data and time of the start and end of the record in local time, and the duration
SMF interval start      2014/03/26,02:30:00
SMF interval end       2014/03/26,02:45:00
SMF interval duration   899.997759 seconds
```

Information about the channel

```
Connection name      9.20.4.159
Channel disp         PRIVATE
Channel type         RECEIVER
Channel status       CLOSING
Channel STATCHL     HIGH
```

```
Start date & time      2014/03/26,02:44:58
Channel status collect time 2014/03/26,02:45:00
Last status changed   1900/01/01,00:00:00
Last msg time         2014/03/26,02:44:59
```

```
Batch size           50
Messages/batch       3.3
Number of messages   1,102
Number of persistent messages 1,102
Number of batches    335
Number of full batches 0
Number of partial batches 335
Buffers sent         337
Buffers received     1,272
Message data         5,038,344  4 MB
Persistent message data 5,038,344  4 MB
```

```

Non persistent message data      0    0 B
Total bytes sent                 9,852  9 KB
Total bytes received            5,043,520  4 MB
Bytes received/Batch            15,055  14 KB
Bytes sent/Batch                 29    29 B
Batches/Second                   1
Bytes received/message          4,576  4 KB
Bytes sent/message               8     8 B
Bytes received/second           28,019  27 KB/sec
Bytes sent/second               54    54 B/sec
Compression rate                 0

The name of the queue manager at the remote end of the connection
Remote qmgr/app                  MQPH
Put retry count                  0

```

## 調整 IBM MQ 網路

請使用本節中的調整提示來協助改善佇列管理程式網路的效能。

### 調整用戶端及伺服器連線通道

在 IBM WebSphere MQ 7.0 中，用戶端與伺服器連線通道的預設值已變更為使用共用交談。然後在 IBM MQ 8.0 中引進分散式伺服器的效能加強功能。若要受益於與共用交談一起引進的新增特性，而對分散式伺服器沒有效能影響，請將 IBM MQ 8.0 或更新版本伺服器連線通道上的 **SHARECNV** 設為 1。

從 IBM WebSphere MQ 7.0 開始，依預設會定義每個通道實例最多執行 10 個用戶端交談。在 IBM WebSphere MQ 7.0 之前，每個交談都配置給不同的通道實例。IBM WebSphere MQ 7.0 中新增的加強功能也包括下列特性：

- 雙向活動訊號
- 管理者停止/靜止
- 先讀
- 非同步-由用戶端應用程式使用

對於部分配置，使用共用交談會帶來重大好處。不過，對於分散式伺服器，在使用 10 個共用交談的預設配置的通道上處理訊息，平均比在不使用共用交談的通道上處理訊息慢 15%。在共用交談的 MQI 通道實例上，相同執行緒會接收 Socket 上的所有交談。如果共用 Socket 的交談都在忙碌中，則交談式執行緒會彼此競爭，以使用接收端執行緒。競用會導致延遲，在此狀況下使用較少的共用交談數會比較好。

您可以使用 **SHARECNV** 參數來指定要在特定 TCP/IP 用戶端通道實例上共用的交談數上限。如需所有可能值以及 IBM WebSphere MQ 7.0 中所新增特性的詳細資料，請參閱 [MQI 用戶端: 用戶端連線及伺服器連線的預設行為](#)。如果您不需要共用交談，則有兩個設定可在 IBM MQ 8.0 或更新版本中提供最佳效能：

- **SHARECNV(1)**。請盡可能使用此設定。它可消除使用接收端執行緒的競用，而且您的用戶端應用程式可以利用 IBM WebSphere MQ 7.0 中新增的特性。對於此設定，分散式伺服器效能可在 IBM MQ 8.0 或更新版本中已顯著改善。效能改進適用於發出非先讀同步取得等待呼叫的 IBM MQ 8.0 或更新版本用戶端應用程式；例如 C 用戶端 **MQGET** 等待呼叫。當連接這些用戶端應用程式時，分散式伺服器會使用較少的執行緒及較少的記憶體，並增加傳輸量。
- **SHARECNV(0)**。通道實例的行為與它是 IBM WebSphere MQ 6.0 伺服器或用戶端連線通道的行為完全相同，而且您無法取得額外特性(例如雙向活動訊號)，當您將 **SHARECNV** 設為 1 或更高版本時，即可使用這些雙向活動訊號。只有當您將 **SHARECNV** 設為 1 或以上時，如果現有的用戶端應用程式未正確執行，請使用 0 值。

#### 附註：

- 如果伺服器連接的用戶端是透過 Socket 來共用交談，且您將共用交談設定從 **SHARECNV(10)** 減少至 **SHARECNV(1)**，則會有下列效果：
  - 增加伺服器上的 Socket 使用情形。
  - 伺服器上增加的通道實例。

在此情況下，您也可以選擇增加 **MaxChannels** 和 **MaxActiveChannels** 的設定。

- 當針對非重新進入的程式庫鏈結或編譯應用程式時，即使在 `CLNTCONN` 和 `SVRCONN` 中設定了較高的值，也會協議 `CURSHCNV(0)` 值。

為了與舊版保持一致，尚未更新預設 `SVRCONN` 通道，因此您需要明確將 `SHARECNV` 設為 1 或 0。

#### 相關概念

MQI 用戶端: 用戶端連線及伺服器連線的預設行為

## 調整分散式發佈/訂閱網路

請使用本節中的調整提示來協助改善 IBM MQ 分散式發佈/訂閱叢集和階層的效能。

#### 相關概念

第 282 頁的『監視叢集』

在叢集內，您可以監視應用程式訊息、控制訊息及日誌。當叢集在佇列的兩個以上實例之間進行負載平衡時，有一些特殊監視考量。

## 直接遞送發佈/訂閱叢集效能

在直接遞送的發佈/訂閱叢集中，不論所有叢集佇列管理程式是否主動參與發佈/訂閱傳訊，都會將叢集主題和 Proxy 訂閱之類的資訊推送至叢集的所有成員。此處理程序可能會在系統上產生大量額外負載。若要減少叢集管理對效能的影響，您可以在離峰時間執行更新，請定義發佈/訂閱所涉及的較小佇列管理程式子集，並使其成為「重疊」叢集，或切換至使用主題主機遞送。

發佈/訂閱叢集中的佇列管理程式有兩個工作量來源:

- 直接處理應用程式的訊息。
- 處理管理叢集所需的訊息及通道。

在一般點對點叢集中，叢集系統工作量很大程度上受限於叢集成員視需要明確要求的資訊。因此，除了非常大的點對點叢集(例如包含數千個佇列管理程式的叢集)之外，您可以大幅降低管理叢集的效能效果。不過，在直接遞送的發佈/訂閱叢集中，不論所有叢集佇列管理程式是否主動參與發佈/訂閱傳訊，都會將叢集主題、佇列管理程式成員資格及 Proxy 訂閱之類的資訊推送至叢集的所有成員。這可能會在系統上產生大量額外負載。因此，您需要考量叢集管理對佇列管理程式效能的影響，包括其計時及其大小。

## 直接遞送叢集的效能性質

比較點對點叢集與核心管理作業的直接遞送發佈/訂閱叢集。

首先，點對點叢集:

1. 當定義新的叢集佇列時，會將目的地資訊推送至完整儲存庫佇列管理程式，且只有在其他叢集成員第一次參照叢集佇列時(例如，當應用程式嘗試開啟它時)，才會將目的地資訊傳送至其他叢集成員。然後佇列管理程式會在本端快取此資訊，以移除每次存取佇列時從遠端擷取資訊的需求。
2. 將佇列管理程式新增至叢集不會直接影響其他佇列管理程式的負載。新佇列管理程式的相關資訊會推送至完整儲存庫，但只有在資料流量開始流入或流出新佇列管理程式時，才會建立及啟動從叢集中其他佇列管理程式到新佇列管理程式的通道。

總而言之，點對點叢集中佇列管理程式的負載與它為應用程式處理的訊息資料流量相關，且與叢集大小沒有直接相關。

其次，直接遞送的發佈/訂閱叢集:

1. 定義新的叢集主題時，會將資訊推送至完整儲存庫佇列管理程式，並直接從該處推送至叢集的所有成員，這會導致通道從完整儲存庫啟動至叢集的每一個成員(如果尚未啟動的話)。如果這是第一個直接叢集主題，則會向每一個佇列管理程式成員傳送叢集中所有其他佇列管理程式成員的相關資訊。
2. 在新主題字串上建立叢集主題的訂閱時，會立即將資訊從該佇列管理程式直接推送至叢集的所有其他成員，這會導致通道從該佇列管理程式啟動至叢集的每一個成員(如果尚未啟動的話)。
3. 當新的佇列管理程式加入現有的叢集時，會將所有叢集主題(如果定義直接叢集主題，則會將所有佇列管理程式成員)的相關資訊從完整儲存庫佇列管理程式推送至新的佇列管理程式。然後，新的佇列管理程式會將叢集中叢集主題所有訂閱的知識與叢集的所有成員同步。

總之，直接遞送發佈/訂閱叢集中任何佇列管理程式的叢集管理負載，會隨著佇列管理程式數目、叢集主題及叢集內不同主題字串上的訂閱變更而增加，而不論每一個佇列管理程式上那些叢集主題的本端使用。

在大型叢集中，或在訂閱變更率較高的叢集中，此層次的叢集管理可能是所有佇列管理程式的重大額外負擔。

## 減少直接遞送發佈/訂閱對效能的影響

若要減少叢集管理對直接遞送發佈/訂閱叢集效能的影響，請考量下列選項：

- 在一天中的離峰時間執行叢集、主題及訂閱更新。
- 定義涉及發佈/訂閱的較小佇列管理程式子集，並使其成為「重疊」叢集。此叢集是定義叢集主題的叢集。雖然部分佇列管理程式現在位於兩個叢集中，但發佈/訂閱的整體效果會減少：
  - 發佈/訂閱叢集的大小較小。
  - 不在發佈/訂閱叢集中的佇列管理程式受叢集管理資料流量影響較小。

如果先前的選項無法充分解決您的效能問題，請考慮改用主題主機遞送發佈/訂閱叢集。如需發佈/訂閱叢集中直接遞送與主題主機遞送的詳細比較，請參閱 [設計發佈/訂閱叢集](#)。

### 相關概念

#### 主題主機遞送發佈/訂閱叢集效能

主題主機遞送的發佈/訂閱叢集可讓您精確控制哪些佇列管理程式管理每一個主題。這些主題主機會變成主題樹狀結構該分支的遞送佇列管理程式。此外，沒有訂閱或發佈者的佇列管理程式不需要連接主題主機。此配置可大幅減少叢集中佇列管理程式之間的連線數目，以及在佇列管理程式之間傳遞的資訊量。

#### 在發佈/訂閱網路中平衡生產者和消費者

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡，否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

#### 發佈/訂閱網路中的訂閱效能

IBM MQ 中的分散式發佈/訂閱的運作方式是傳播佇列管理程式網路中已建立不同主題字串的訂閱所在的知識。這可讓發佈訊息的佇列管理程式識別哪些其他佇列管理程式需要已發佈訊息的副本，以符合其訂閱。

## 主題主機遞送發佈/訂閱叢集效能

主題主機遞送的發佈/訂閱叢集可讓您精確控制哪些佇列管理程式管理每一個主題。這些主題主機會變成主題樹狀結構該分支的遞送佇列管理程式。此外，沒有訂閱或發佈者的佇列管理程式不需要連接主題主機。此配置可大幅減少叢集中佇列管理程式之間的連線數目，以及在佇列管理程式之間傳遞的資訊量。

主題主機遞送的發佈/訂閱叢集行為如下：

- 主題在叢集中的個別主題主機佇列管理程式上手動定義。
- 在叢集佇列管理程式上進行訂閱時，只會在主題主機上建立 Proxy 訂閱。
- 當應用程式將資訊發佈至主題時，接收端佇列管理程式會將發佈轉遞至管理主題的佇列管理程式。然後主題主機會將發佈資訊傳送至叢集中對主題具有有效訂閱的所有佇列管理程式。

如需主題主機遞送的更詳細簡介，請參閱 [叢集中的主題主機遞送](#)。

對於許多配置，主題主機遞送是比直接遞送更適當的拓撲，因為它提供下列好處：

- 已改進較大叢集的可調整性。只有主題主機佇列管理程式才需要能夠連接到叢集中的所有其他佇列管理程式。因此，佇列管理程式之間的通道更少，佇列管理程式間的發佈/訂閱管理資料流量會少於直接遞送。佇列管理程式上的訂閱變更時，只需要通知主題主機佇列管理程式。
- 對實體配置有更多控制。使用直接遞送時，所有佇列管理程式將擔任所有角色，因此全都需要同等能力。使用主題主機遞送時，您可明確選擇主題主機佇列管理程式。因此，您可以確保這些佇列管理程式是在充足的設備上執行，並且可對其他佇列管理程式使用功能差一點的系統。

不過，主題主機遞送還是會對您的系統加諸一些限制：

- 系統配置和維護需要比直接遞送更多的規劃。您需要決定要在主題樹狀結構中形成叢集的時間點，以及主題定義在叢集的位置。

- 就像直接遞送主題一樣，在定義新的主題主機遞送主題時，該資訊會推送到完整儲存庫佇列管理程式，再從中導向至叢集的所有成員。此事件會導致從完整儲存庫啟動通往叢集每一個成員的通道（如果尚未啟動）。
- 一律從非主機佇列管理程式將這些發佈資訊傳送至主機佇列管理程式，即使叢集中沒有訂閱也一樣。因此，當一般預期會有訂閱存在時，或當廣域連線功能和知識的超載量大於額外發佈資料流量的風險時，您應使用遞送主題。
- 在非主機佇列管理程式上發佈的訊息不會直接進入管理訂閱的佇列管理程式，而是一律透過主題主機佇列管理程式來遞送。此方法可能會增加叢集的總超載量，因而增加訊息延遲時間並降低效能。  
**註：**對於某些配置，您可以如 [使用集中式發佈者或訂閱者的主題主機遞送](#) 中所述，有效地移除此限制。
- 使用單一主題主機佇列管理程式，為發佈到某個主題的所有訊息建立單一失敗點。您可以透過定義多個主題主機來移除這個單一失敗點。不過，擁有多個主機將影響訂閱接收已發佈訊息的順序。
- 主題主機佇列管理程式將產生額外訊息負載，因為它們需要處理來自多個佇列管理程式的發佈資料流量。可以減輕這項負載：將多個主題主機用於單一主題（在此情況下，不會維護訊息順序），或使用不同的佇列管理程式來管理主題樹狀結構的不同分支的遞送主題。

### 具有集中式發佈者或訂閱者的主題主機遞送

如果要移除在一律透過主題主機佇列管理程式將發佈遞送至訂閱時所產生的額外「躍點」，請在管理主題的相同佇列管理程式上配置發佈者或訂閱。在下列兩種情況下，此方法可帶來最大效能好處：

- 具有許多發佈者及少量訂閱的主題。在此情況下，請在主題主機佇列管理程式上管理訂閱。
- 具有較少發佈者及許多訂閱的主題。在此情況下，請在主題主機佇列管理程式上管理發佈者。

下圖顯示也管理訂閱的主題主機佇列管理程式。此方法會移除發佈者與訂閱者之間的額外「中繼站」，並減少叢集所有成員之間不必要的訂閱知識共用：

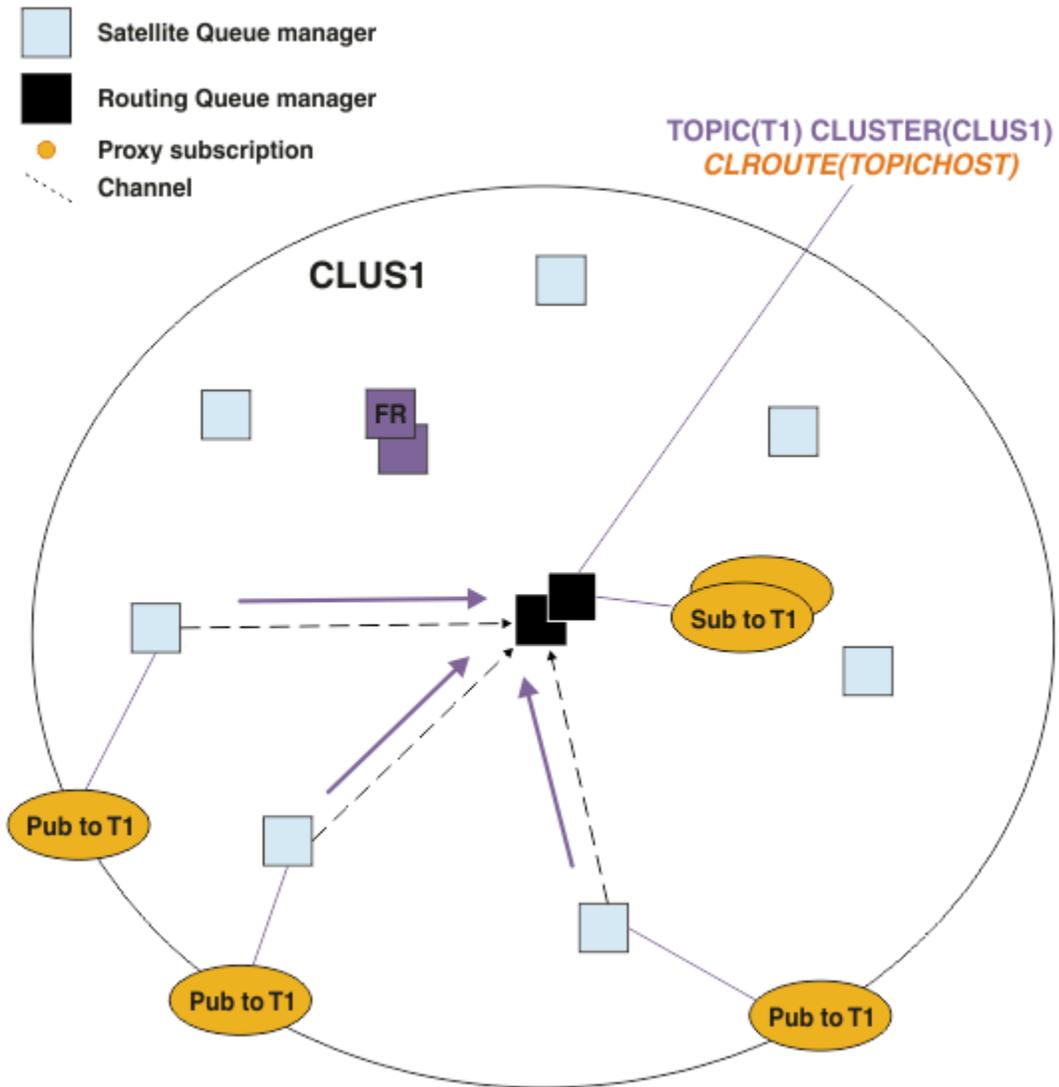


圖 27: 在主題主機佇列管理程式上管理訂閱

下圖顯示同時管理發佈者的主題主機佇列管理程式。此方法會移除發佈者與訂閱者之間的額外「中繼站」，並減少叢集所有成員之間不必要的訂閱知識共用：

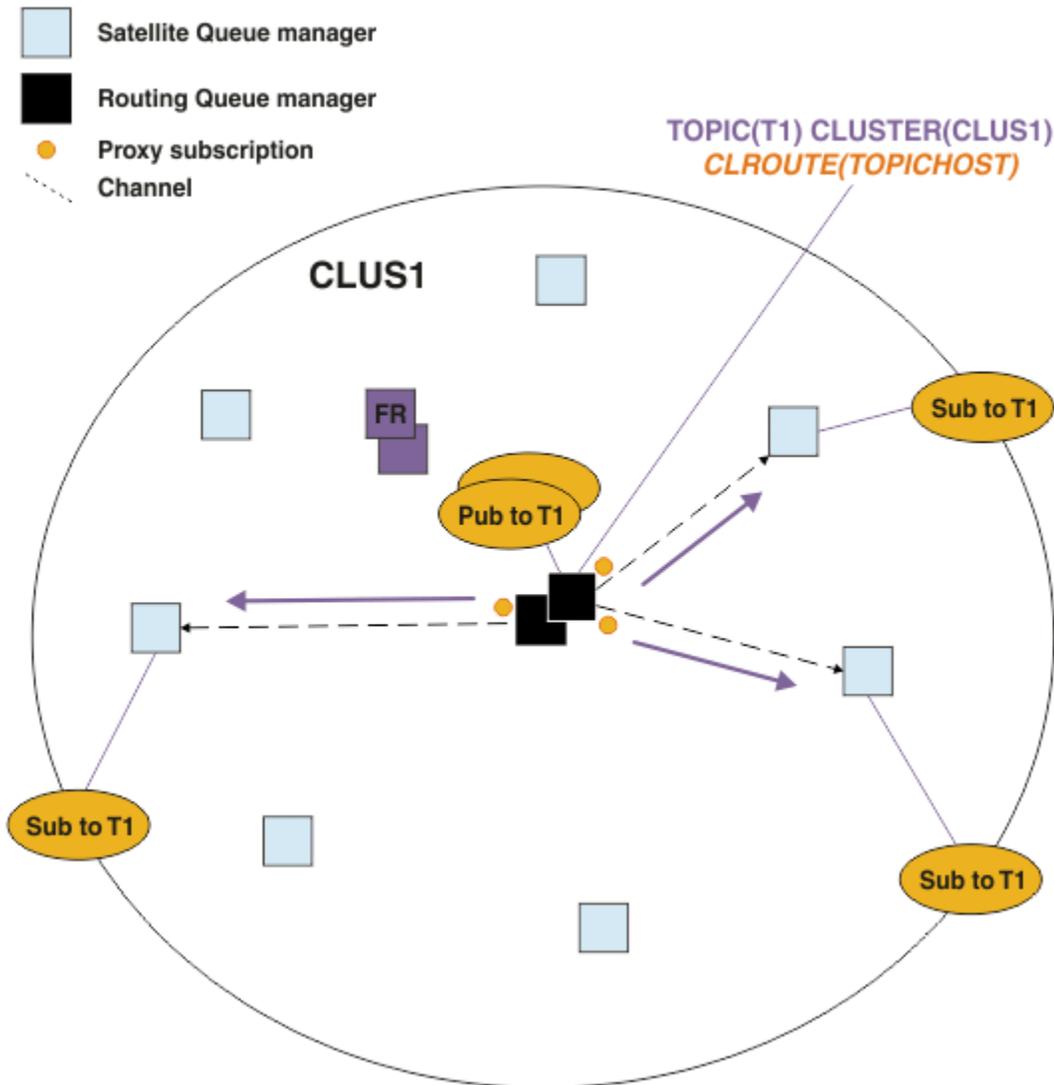


圖 28: 在主題主機佇列管理程式上管理發佈

### 相關概念

#### 直接遞送發佈/訂閱叢集效能

在直接遞送的發佈/訂閱叢集中，不論所有叢集佇列管理程式是否主動參與發佈/訂閱傳訊，都會將叢集主題和 Proxy 訂閱之類的資訊推送至叢集的所有成員。此處理程序可能會在系統上產生大量額外負載。若要減少叢集管理對效能的影響，您可以在離峰時間執行更新，請定義發佈/訂閱所涉及的較小佇列管理程式子集，並使其成為「重疊」叢集，或切換至使用主題主機遞送。

#### 在發佈/訂閱網路中平衡生產者和消費者

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡，否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

#### 發佈/訂閱網路中的訂閱效能

IBM MQ 中的分散式發佈/訂閱的運作方式是傳播佇列管理程式網路中已建立不同主題字串的訂閱所在的知識。這可讓發佈訊息的佇列管理程式識別哪些其他佇列管理程式需要已發佈訊息的副本，以符合其訂閱。

### 在發佈/訂閱網路中平衡生產者和消費者

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡，否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

在點對點傳訊拓撲中，訊息消費者與訊息產生者之間的關係很容易理解。您可以取得訊息正式作業及耗用的估計值，依佇列、依通道的通道。如果缺乏平衡，則會立即找出瓶頸，然後加以補救。

在發佈/訂閱拓撲中，很難找出發佈者和訂閱者是否平衡。從每一個訂閱開始，然後回到具有主題發佈者的佇列管理程式。計算從每一個佇列管理程式流向每一個訂閱者的發佈數。

每一個符合遠端佇列管理程式上的訂閱 (根據 Proxy 訂閱) 的發佈都會放入傳輸佇列中。如果多個遠端佇列管理程式具有該發佈的 Proxy 訂閱，則會將多個訊息副本放置到傳輸佇列，每個副本都以不同的傳送端通道為目標。

在發佈/訂閱叢集中，這些發佈會以管理訂閱之遠端佇列管理程式上的 SYSTEM.INTER.QMGR.PUBS 佇列為目標。在階層中，每個發佈資訊都以 SYSTEM.BROKER.DEFAULT.STREAM 佇列為目標，或以遠端佇列管理程式上「SYSTEM.QPUBSUB.QUEUE.NAMELIST」中列出的任何其他串流佇列為目標。每一個佇列管理程式都會處理到達該佇列的訊息，並將它們遞送至該佇列管理程式上的正確訂閱。

基於此原因，請在可能出現瓶頸的下列點監視負載：

- 監視個別訂閱佇列的負載。
  - 這個瓶頸意味著訂閱端應用程式不會像發佈時一樣快速使用發佈。
- 監視 SYSTEM.INTER.QMGR.PUBS 佇列或串流佇列的負載。
  - 此瓶頸意味著佇列管理程式從一或多個遠端佇列管理程式接收發佈的速度，比它將它們配送至本端訂閱的速度還要快。
  - 在叢集中使用主題主機遞送時，如果在主題主機佇列管理程式上看到的話，請考量建立其他佇列管理程式主題主機，以容許在它們之間平衡發佈工作量。不過，這會影響發佈之間的訊息排序。請參閱 [針對單一主題使用多個主題主機的主題主機遞送](#)。
- 監視發佈佇列管理程式與訂閱佇列管理程式之間通道的負載，這些通道由發佈佇列管理程式上的傳輸佇列提供。
  - 這個瓶頸意味著一或多個通道不在執行中，或訊息發佈至本端佇列管理程式的速度比通道遞送至遠端佇列管理程式的速度更快。
  - 當您使用發佈/訂閱叢集時，請考量在目標佇列管理程式上定義其他叢集接收端通道。這可讓發佈工作量在它們之間達到平衡。不過，這會影響發佈之間的訊息排序。另外，也請考慮移至多重叢集傳輸佇列配置，因為在某些情況下，這可以增進效能。
- 如果發佈應用程式正在使用已排入佇列的發佈/訂閱介面，請監視 (a) SYSTEM.BROKER.DEFAULT.STREAM 佇列的負載，以及 SYSTEM.QPUBSUB.QUEUE.NAMELIST 中列出的任何其他串流佇列；及 (b) SYSTEM.BROKER.DEFAULT.SUBPOINT 佇列，以及 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST 中列出的任何其他子點佇列。
  - 這個瓶頸意味著本端發佈應用程式放置訊息的速度比本端佇列管理程式處理訊息的速度還快。

## 相關概念

### 直接遞送發佈/訂閱叢集效能

在直接遞送的發佈/訂閱叢集中，不論所有叢集佇列管理程式是否主動參與發佈/訂閱傳訊，都會將叢集主題和 Proxy 訂閱之類的資訊推送至叢集的所有成員。此處理程序可能會在系統上產生大量額外負載。若要減少叢集管理對效能的影響，您可以在離峰時間執行更新，請定義發佈/訂閱所涉及的較小佇列管理程式子集，並使其成為「重疊」叢集，或切換至使用主題主機遞送。

### 主題主機遞送發佈/訂閱叢集效能

主題主機遞送的發佈/訂閱叢集可讓您精確控制哪些佇列管理程式管理每一個主題。這些主題主機會變成主題樹狀結構該分支的遞送佇列管理程式。此外，沒有訂閱或發佈者的佇列管理程式不需要連接主題主機。此配置可大幅減少叢集中佇列管理程式之間的連線數目，以及在佇列管理程式之間傳遞的資訊量。

### 發佈/訂閱網路中的訂閱效能

IBM MQ 中的分散式發佈/訂閱的運作方式是傳播佇列管理程式網路中已建立不同主題字串的訂閱所在的知識。這可讓發佈訊息的佇列管理程式識別哪些其他佇列管理程式需要已發佈訊息的副本，以符合其訂閱。

### 第 282 頁的『監視叢集』

在叢集內，您可以監視應用程式訊息、控制訊息及日誌。當叢集在佇列的兩個以上實例之間進行負載平衡時，有一些特殊監視考量。

## 發佈/訂閱網路中的訂閱效能

IBM MQ 中的分散式發佈/訂閱的運作方式是傳播佇列管理程式網路中已建立不同主題字串的訂閱所在的知識。這可讓發佈訊息的佇列管理程式識別哪些其他佇列管理程式需要已發佈訊息的副本，以符合其訂閱。

此方法可將將已發佈訊息傳送至沒有相符訂閱的佇列管理程式的作業減至最少。不過，當訂閱的主題字串數目偏高，且透過頻繁建立及刪除訂閱而持續變更時，訂閱知識的傳播可能會變成重大額外負擔。

您可以調整發佈/訂閱網路周圍的發佈和訂閱傳送方式，以影響效能。如果您的網路資料流量只有幾個發佈，以及快速建立、刪除或變更訂閱，您可以停止將訂閱資訊傳送至所有佇列管理程式，並改為將所有發佈轉遞至網路中的所有佇列管理程式。您也可以連接的佇列管理程式之間限制給定主題的 Proxy 訂閱和發佈流程、限制包含萬用字元的 Proxy 訂閱流程，以及減少主題字串的數目和暫時性。

## 個別訂閱傳播及隨處發佈

隨處發佈是個別訂閱傳播的替代方案。利用個別傳播，只有在佇列管理程式上具有相符訂閱的發佈才會轉遞至該佇列管理程式。使用隨處發佈，所有發佈都會轉遞至網路中的所有佇列管理程式。然後，接收端佇列管理程式會遞送符合本端訂閱的那些發佈。

### 個別訂閱傳播

此機制會產生最少的佇列管理程式間發佈資料流量，因為只會傳送符合佇列管理程式上訂閱的那些發佈。

不過：

- 對於所訂閱的每一個個別主題字串，會將 Proxy 訂閱傳送至發佈/訂閱拓撲中的其他佇列管理程式。佇列管理程式集取決於所使用的遞送模型，如規劃分散式發佈/訂閱網路中所述。
  - 如果要建立或刪除數千個訂閱 (例如，在重新啟動佇列管理程式之後重建所有不可延續的訂閱)，或如果訂閱集快速變更，且每一個訂閱都是不同的主題字串，則此傳訊額外負擔可能很重要。
  - 將 Proxy 訂閱延伸到的佇列管理程式數目也會影響額外負擔的規模。
- Proxy 訂閱會使用非同步傳訊傳送至其他佇列管理程式。這具有下列效果：
  - 在建立訂閱與其他佇列管理程式建立、遞送及處理 Proxy 訂閱之間，會有延遲。
  - 在該間隔中發佈在那些佇列管理程式的訊息不會遞送至遠端訂閱。

### 隨處發佈

使用此機制，系統上不會有每個主題字串 Proxy 訂閱額外負擔。這表示快速建立、刪除或變更訂閱不會導致網路負載及處理程序增加。

建立訂閱與傳送至佇列管理程式的發佈之間也沒有延遲，因為所有發佈都會傳送至所有佇列管理程式。因此，沒有視窗未將發佈遞送至新建立的遠端訂閱。

不過：

- 將所有發佈資訊傳送至發佈/訂閱拓撲中的所有佇列管理程式，可能會導致網路資料流量過多，其中發佈資訊在每一個佇列管理程式上沒有相符的訂閱。
  - 拓撲中的佇列管理程式數目越多，額外負擔就越大。

當您預期發佈會從相當比例的佇列管理程式訂閱，或 Proxy 訂閱因為訂閱變更頻率而額外需要太多時，您應該考慮使用隨處發佈機制。當發佈傳送至所有佇列管理程式時，如果您遇到傳訊資料流量增加的情況，您應該使用個別 Proxy 訂閱轉遞，而不是傳送至具有相符訂閱的佇列管理程式。

您可以在主題樹狀結構內的任何層次設定隨處發佈行為。若要啟用隨處發佈，請將高階主題物件的 **PROXYSUB** 參數設為 **FORCE**。這會產生單一萬用字元 Proxy 訂閱，以符合主題樹狀結構中此主題物件下的所有主題。在叢集主題物件上設定它時，**PROXYSUB(FORCE)** 屬性會延伸到網路中的每個佇列管理程式，而不只是在其中定義主題的佇列管理程式。

註：在階層中使用時，您可以在每一個佇列管理程式上個別設定 **PROXYSUB(FORCE)**，因此拓撲機制自然會限制通道數目。不過，在叢集中使用時，可能會啟動許多其他通道：

- 在主題主機遞送叢集中，會啟動從每一個佇列管理程式到每一個主題主機佇列管理程式的通道。
- 在直接遞送叢集中，通道會從每一個佇列管理程式啟動至每一個其他佇列管理程式。

啟動許多通道的額外負擔在直接遞送叢集中最為明顯，且可能導致效能問題。請參閱 [第 328 頁的『直接遞送發佈/訂閱叢集效能』](#)。

## 限制已連接佇列管理程式之間 Proxy 訂閱及發佈流程的其他方式

### 合併主題字串

在連接發佈者或訂閱的系統中，使用許多不同的暫時性主題字串會在每一個佇列管理程式上引進某種層次的管理額外負擔。您應該定期評量主題字串的使用，以查看它們是否可以合併。減少主題字串的數目和暫時性，從而減少它們的發佈者和訂閱，減少對系統的影響。

### 限制發佈及訂閱範圍

對於給定主題，您可以使用 [發佈範圍](#) 及 [訂閱範圍](#) 設定，將發佈及訂閱保留在定義它們的佇列管理程式本端。

### 封鎖對萬用字元主題所做的訂閱

您可以將 **Topic** 屬性 **萬用字元** 設為 BLOCK，以限制包含萬用字元的 Proxy 訂閱流程。請參閱 [Proxy 訂閱中的萬用字元](#)。

另請參閱 [第 332 頁的『在發佈/訂閱網路中平衡生產者和消費者』](#)

## 監視叢集中的 Proxy 訂閱資料流量

考量來自 Proxy 訂閱資料流量的系統負載時，除了監視 [第 332 頁的『在發佈/訂閱網路中平衡生產者和消費者』](#) 中列出的佇列之外，還請監視下列叢集佇列：

- SYSTEM.INTER.QMGR.FANREQ 佇列。
- SYSTEM.INTER.QMGR.CONTROL 佇列。

這些佇列上的任何重要訊息待辦事項都意味著訂閱變更率對系統而言太大，或佇列管理程式在叢集中無法正確運作。如果您懷疑問題在於特定的佇列管理程式，請檢查該佇列管理程式未停用發佈/訂閱支援。請參閱 [ALTER QMGR](#) 中的 **PSMODE**。

### 相關概念

#### 直接遞送發佈/訂閱叢集效能

在直接遞送的發佈/訂閱叢集中，不論所有叢集佇列管理程式是否主動參與發佈/訂閱傳訊，都會將叢集主題和 Proxy 訂閱之類的資訊推送至叢集的所有成員。此處理程序可能會在系統上產生大量額外負載。若要減少叢集管理對效能的影響，您可以在離峰時間執行更新，請定義發佈/訂閱所涉及的較小佇列管理程式子集，並使其成為「重疊」叢集，或切換至使用主題主機遞送。

#### 主題主機遞送發佈/訂閱叢集效能

主題主機遞送的發佈/訂閱叢集可讓您精確控制哪些佇列管理程式管理每一個主題。這些主題主機會變成主題樹狀結構該分支的遞送佇列管理程式。此外，沒有訂閱或發佈者的佇列管理程式不需要連接主題主機。此配置可大幅減少叢集中佇列管理程式之間的連線數目，以及在佇列管理程式之間傳遞的資訊量。

#### 在發佈/訂閱網路中平衡生產者和消費者

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡，否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

#### 發佈/訂閱網路中的 Proxy 訂閱

## 減少主題樹狀結構中不想要的主题數目

透過減少主題樹狀結構中不想要的主题數目，可改善發佈/訂閱系統的效能。什麼是不想要的主题，以及如何移除它們？

您可以建立大量主题，而不會對效能產生不利影響。不過，某些使用發佈/訂閱的方式會導致持續展開主题樹狀結構。異常大量的主题會建立一次，且不再使用。越來越多的主题可能會變成效能問題。

如何避免設計導致大量且不斷增加的不想要主题？您可以做什麼來協助佇列管理程式從主题樹狀結構中移除不想要的主题？

佇列管理程式可辨識不想要的主题，因為它已 30 分鐘未使用。佇列管理程式會為您從主题樹狀結構中移除未用的主题。您可以變更佇列管理程式屬性 **TREELIFE** 來變更 30 分鐘的持續時間。您可以協助佇列管理

程式移除不想要的主題，方法是確定該主題對佇列管理程式顯示為未用。第 336 頁的『何謂未使用的主題?』一節說明未使用的主題為何。

程式設計師，設計任何應用程式，特別是設計長時間執行的應用程式，會考量其資源用量：程式需要多少資源，是否有任何無限制的需求，以及任何資源洩漏？主題是發佈/訂閱程式使用的資源。如同程式所使用的任何其他資源一樣，仔細檢查主題的使用。

## 何謂未使用的主題？

在定義未使用的主題之前，哪些確實是主題？

當主題字串 (例如 USA/Alabama/Auburn) 轉換為主題時，該主題會新增至主題樹狀結構。必要的話，會在樹狀結構中建立其他主題節點及其對應的主題。主題字串 USA/Alabama/Auburn 會轉換成具有三個主題的樹狀結構。

- USA
- USA/Alabama
- USA/Alabama/Auburn

若要顯示主題樹狀結構中的所有主題，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('#') TYPE(TOPIC)`。

主題樹狀結構中未用的主題具有下列內容。

### 它未與主題物件相關聯

管理主題物件具有將其與主題相關聯的主題字串。當您定義主題物件 Alabama 時，如果要關聯的主題 USA/Alabama 不存在，則會從主題字串建立主題。如果主題確實存在，則會使用主題字串將主題物件與主題關聯在一起。

### 它沒有保留的發佈

具有保留發佈資訊的主題是來自發佈者使用 MQPMO\_RETAIN 選項將訊息放置到主題。

使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama') RETAINED` 來檢查 USA/Alabama 是否具有保留的發佈資訊。回應為 YES 或 NO。

使用 **runmqsc** 指令 `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)`，從 USA/Alabama 中移除保留的發佈資訊。

### 它沒有子主題

USA/Alabama/Auburn 是沒有子主題的主題。USA/Alabama/Auburn 是 USA/Alabama 的直屬子主題。

使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama+')` 顯示 USA/Alabama 的直屬子項。

### 節點沒有作用中的發佈者

節點的作用中發佈者是開啟主題以供輸出的應用程式。

例如，應用程式以開啟選項 MQ00\_OUTPUT 開啟名為 Alabama 的主題物件。

若要顯示 USA/Alabama 及其所有子項的作用中發佈者，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN`。

### 節點沒有作用中的訂閱者

作用中訂閱者可以是可延續訂閱，或是已向 MQSUB 登錄主題訂閱且未關閉它的應用程式。

若要顯示 USA/Alabama 的作用中訂閱，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN`。

若要顯示 USA/Alabama 及其所有子項的作用中訂閱，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN`。

## 管理主題樹狀結構中的主題數

總之，有許多方法可以管理主題樹狀結構中的主題數。

## 顯示 TPCOUNT

定期使用 **runmqsc** 指令 DISPLAY PUBSUB ALL 來顯示 **TPCOUNT** 內容。這是主題樹狀結構中的主題節點數目。如果數目不斷增加，可能表示需要較短的 **TREELIFE**，或需要重新設計主題本身。

## 修改 TREELIFE

依預設，未用的主題的生命期限為 30 分鐘。您可以減少未用主題的生命期限。

例如，**runmqsc** 指令 ALTER QMGR TREELIFE(900) 會將未用主題的生命期限從 30 分鐘減少至 15 分鐘。

## 異常地重新啟動佇列管理程式

當佇列管理程式重新啟動時，會從主題物件、具有保留發佈資訊的節點及可延續訂閱重新起始設定主題樹狀結構。已刪除發佈者及訂閱者程式作業所建立的主題。

作為最後手段，如果不想要的主題中的成長已成為過去效能問題的原因，請重新啟動佇列管理程式。

## 相關概念

[主題樹狀結構](#)

V 9.1.4

MQ Adv.

Linux

MQ Adv. VUE

## Aspera gateway 可以改善高延遲網路的

### 效能

IBM Aspera fasp.io Gateway 提供快速 TCP/IP 通道，可大幅增加 IBM MQ 的網路傳輸量。

Aspera gateway 可用來改善佇列管理程式通道的效能。如果網路具有高延遲或容易遺失封包，且通常用來加速不同資料中心的佇列管理程式之間的連線，則特別有效。

不過，對於不會遺失封包的快速網路，使用 Aspera gateway 時效能會降低，因此在定義 Aspera gateway 連線之前及之後檢查網路效能是很重要的。

在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat® 上或 Ubuntu Linux。

如需相關資訊，請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。



## 注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、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，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權  
法務部與智慧財產權法律  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國：** International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation  
軟體交互作業能力協調程式，部門 49XA  
3605 公路 52 N  
Rochester, MN 55901  
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。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。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本書的使用者應依自己的特定環境，查證適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。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。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

## 程式設計介面資訊

---

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

**重要：**請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

## 商標

---

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

UNIX 是 The Open Group 在美國及/或其他國家/地區的註冊商標。

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。





產品編號:

(1P) P/N: