

7.5

*IBM WebSphere 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 259 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® WebSphere MQ 7.5 版及所有後續版次與修訂。

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>
事件監視.....	5
檢測事件.....	5
效能事件.....	17
配置事件.....	32
指令事件.....	35
日誌程式事件.....	37
監視檢測事件的範例程式.....	42
訊息監視.....	49
活動和業務.....	49
訊息遞送技術.....	50
活動記錄中.....	52
追蹤路徑傳訊.....	56
IBM WebSphere MQ 顯示路徑應用程式.....	67
活動報告參照.....	82
追蹤路徑訊息參照.....	108
追蹤路徑回覆訊息參照.....	117
帳戶及統計資料訊息.....	120
帳戶訊息.....	120
統計資料訊息.....	123
顯示帳戶及統計資料資訊.....	127
帳戶及統計資料訊息參照.....	132
應用程式活動追蹤.....	177
收集應用程式活動追蹤資訊.....	177
amqsact 範例程式.....	184
應用程式活動追蹤訊息參照.....	185
即時監視.....	247
控制即時監視的屬性.....	248
顯示佇列及通道監視資料.....	249
監視佇列.....	250
監視頻道.....	253
Windows 效能監視器.....	258
<b>注意事項</b> .....	<b>259</b>
程式設計介面資訊.....	260
商標.....	260



## 監視及效能

---

IBM WebSphere MQ 中提供許多監視技術，以取得關於佇列管理程式網路執行方式的統計資料及其他特定資訊。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。

視佇列管理程式網路的大小及複雜性而定，您可以從監視佇列管理程式網路取得一系列資訊。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。

### 相關工作

[配置](#)

[管理 WebSphere MQ](#)

## 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 WebSphere 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

[事件訊息參照](#)

[第 7 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 WebSphere 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 6 頁的圖 1 說明檢測事件的概念。

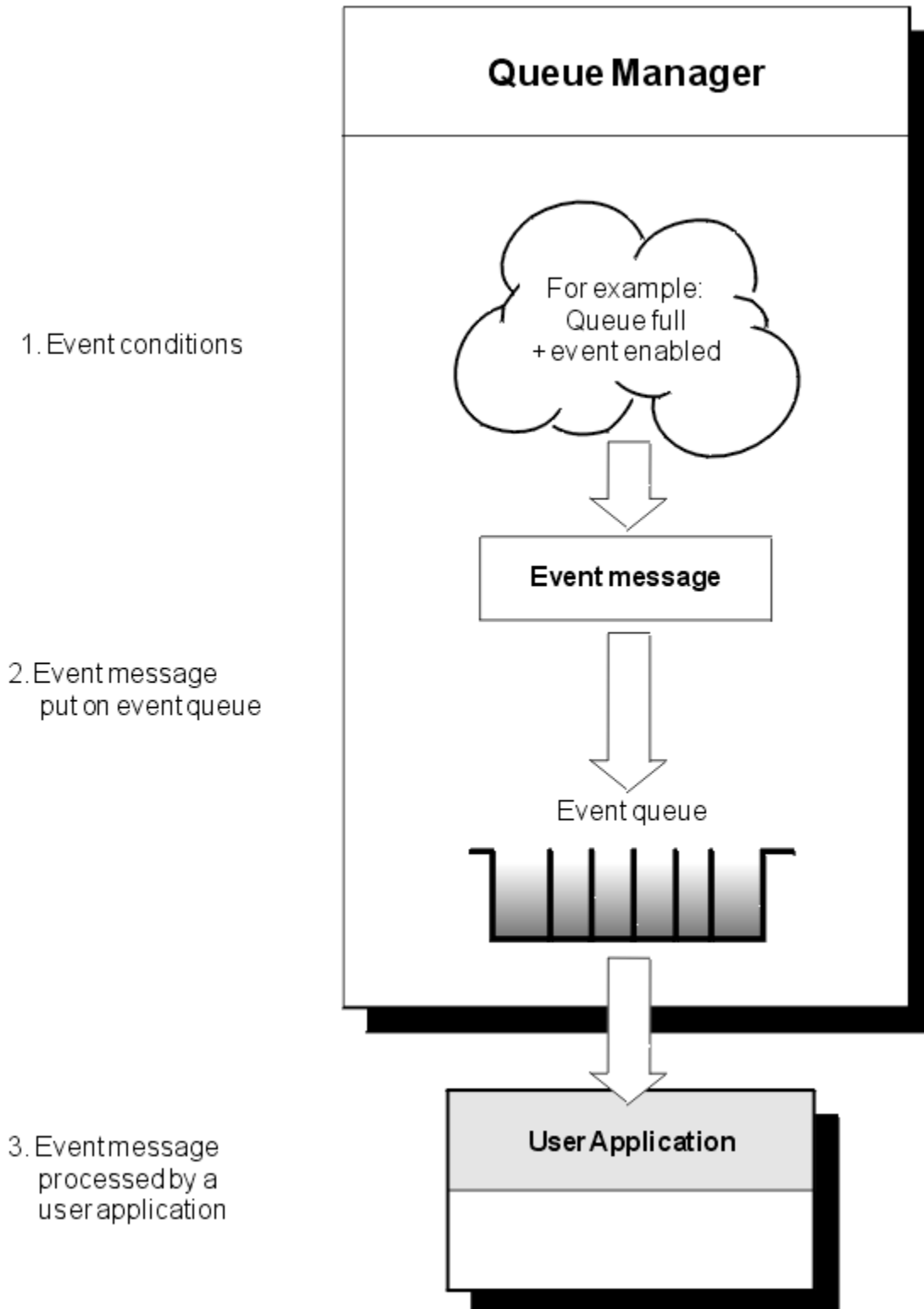


圖 1: 瞭解檢測事件

### 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 WebSphere 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 17 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 42 頁的『監視檢測事件的範例程式』

請利用這個頁面來檢視監視檢測事件的範例 C 程式

## 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 WebSphere 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

SYSTEM.ADMIN.PERFM.EVENT

### 包含來自下列的訊息：

佇列管理程式事件

通道事件

效能事件

**此事件佇列:**

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:**

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 WebSphere MQ 應用程式的活動。特別是您可以從單一節點 (針對支援 IBM WebSphere MQ 事件的節點) 監視系統中的所有節點，如 [第 8 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。

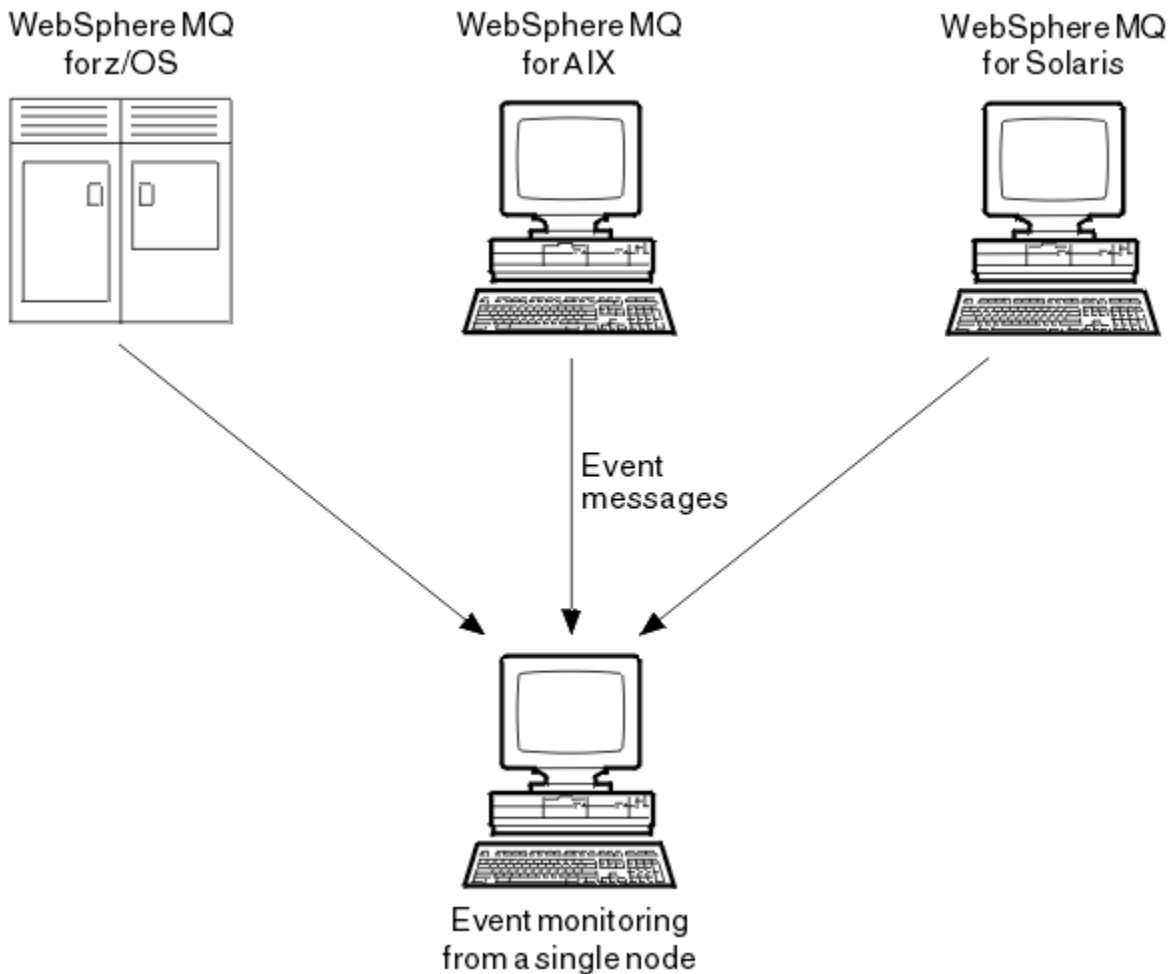


圖 2: 在單一節點上跨不同平台監視佇列管理程式

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。

下列範例是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。

WebSphere 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 系統)

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 \(類型 6\)](#)

所有權限事件僅在 Windows 及 UNIX 系統上有效。

#### 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

本端事件指出應用程式 (或佇列管理程式) 無法存取本端佇列或其他本端物件。例如，應用程式可能嘗試存取尚未定義的物件。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

遠端事件指出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，可能無法正確定義要使用的傳輸佇列。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 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](#)

## SSL 事件

唯一的 Secure Sockets Layer (SSL 或 TLS) 事件是「通道 SSL 錯誤」事件。當使用 SSL 或 TLS 的通道無法建立 SSL 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。

#### 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](#)。

#### 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時，會報告日誌程式事件。

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	<a href="#">指令</a>
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>
IMS Bridge 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>

事件類型	請參閱下列主題
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 WebSphere MQ Script 指令 (MQSC)。
- 對應的 IBM WebSphere MQ PCF 指令。
- IBM WebSphere MQ 檔案總管。

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 5 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。

### 相關工作

[自動化管理作業](#)

[使用可程式指令格式](#)

### 相關參考

[第 7 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 14 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。

表 1: 使用 MQSC 指令啟用佇列管理程式事件	
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 14 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。

表 2: 使用 MQSC 指令啟用通道及橋接器事件	
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 MQRQ\_CHANNEL\_ACTIVATED
- MQRQ\_CHANNEL\_CONV\_ERROR
- 已啟動 MQRQ\_CHANNEL\_NOT\_ACTIVATED
- MQRQ\_CHANNEL\_STOPPED
  - 具有下列 ReasonQualifiers:
    - MQRQ\_CHANNEL\_STOPPED\_ERROR
    - MQRQ\_CHANNEL\_STOPPED\_RETRY
    - MQRQ\_CHANNEL\_STOPPED\_DISABLED
- MQRQ\_CHANNEL\_STOPPED\_BY\_USER
- MQRQ\_CHANNEL\_BLOCKED
  - 具有下列 ReasonQualifiers:
    - MQRQ\_CHANNEL\_BLOCKED\_NOACCESS
    - MQRQ\_CHANNEL\_BLOCKED\_USERID
    - MQRQ\_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 QSVICINT 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、別名佇列或遠端佇列的本端定義。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。

## 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 13 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 16 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 WebSphere MQ 訊息一樣，事件訊息有兩個部分：訊息描述子和訊息資料。

[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 WebSphere 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 84 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 88 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 89 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 WebSphere MQ 指令 (MQSC) 來設定佇列屬性。在上，您也可以使用佇列管理程式的作業和控制項畫面來設定佇列屬性。

### 相關參考

[第 7 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 17 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」。

### 相關概念

[第 17 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 19 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 20 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 20 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](#)

## 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。

在共用佇列上 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 19 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。

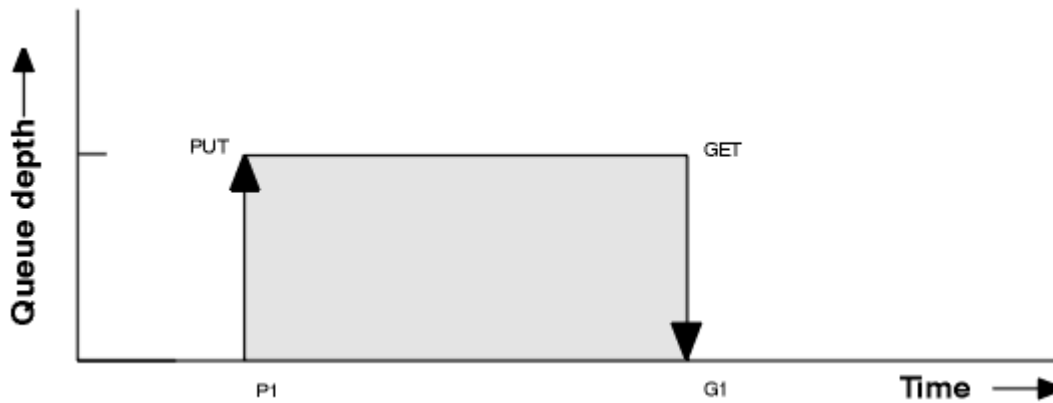


圖 3: 瞭解佇列服務間隔事件

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 20 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 \(MQLONG\)](#)

[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 20 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 QSVCINT 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') QSVCINT(10000) QSVCI EV(HIGH)
```

## 佇列服務間隔事件範例

使用這些範例來瞭解您可以從佇列服務間隔事件取得的資訊

這三個範例逐步提供使用佇列服務間隔事件的更複雜圖解。

範例隨附的數字具有相同的結構：

- 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。

**記注：**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 18 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 19 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。

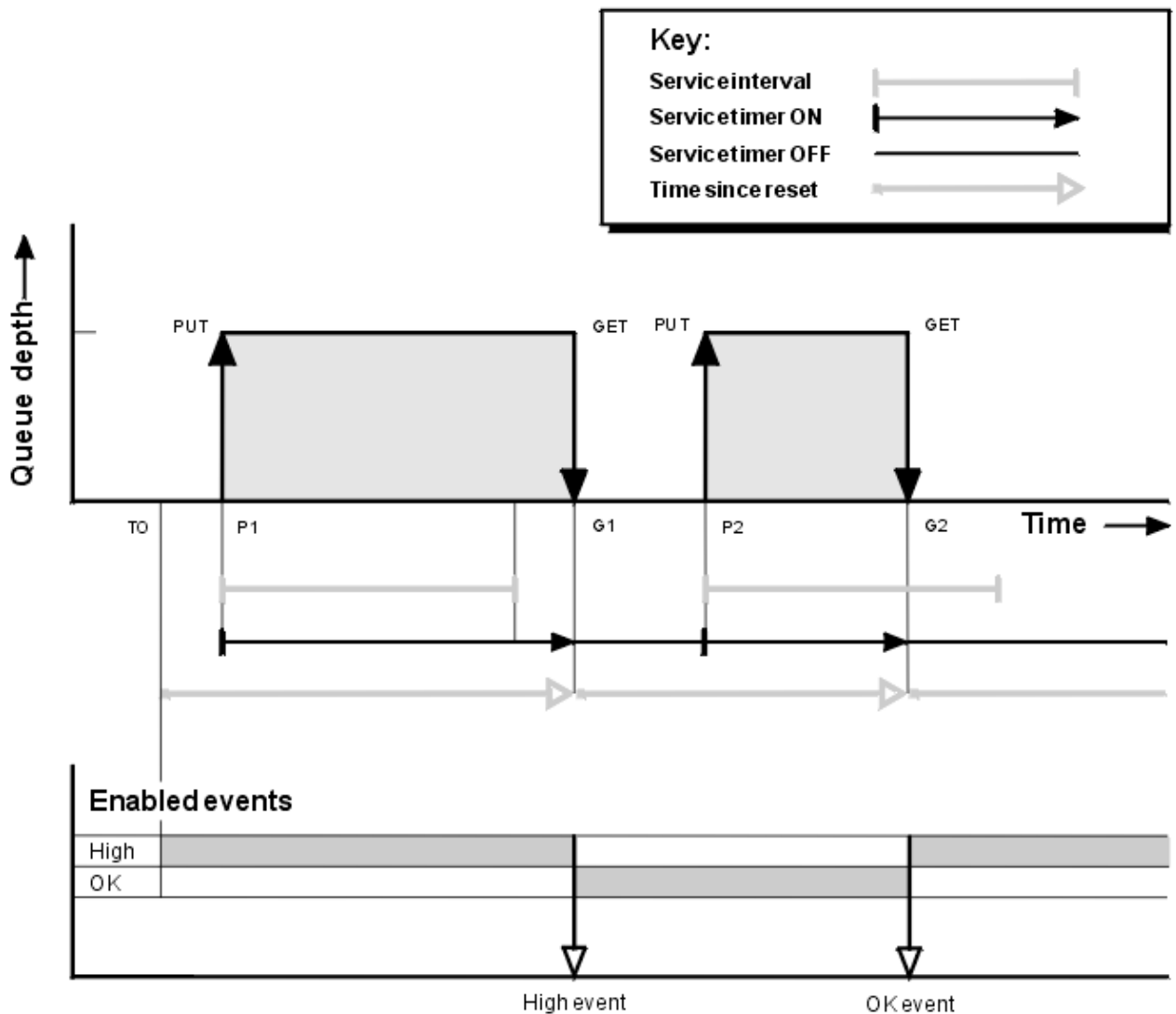


圖 4: 佇列服務間隔事件-範例 1

### 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 23 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		
	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 22 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。

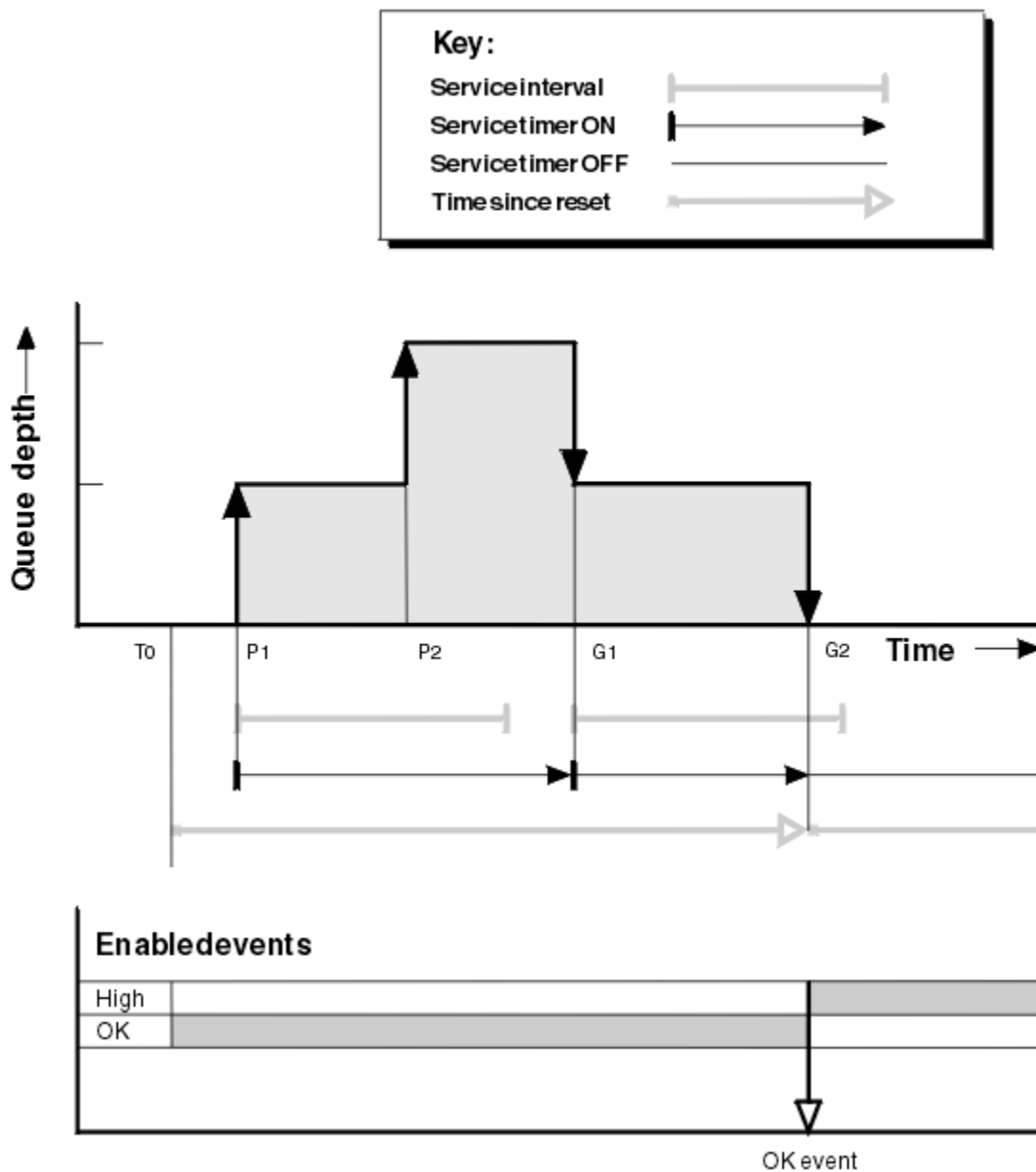


圖 5: 佇列服務間隔事件-範例 2

### 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 25 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	
	<b>事件 2</b>
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。

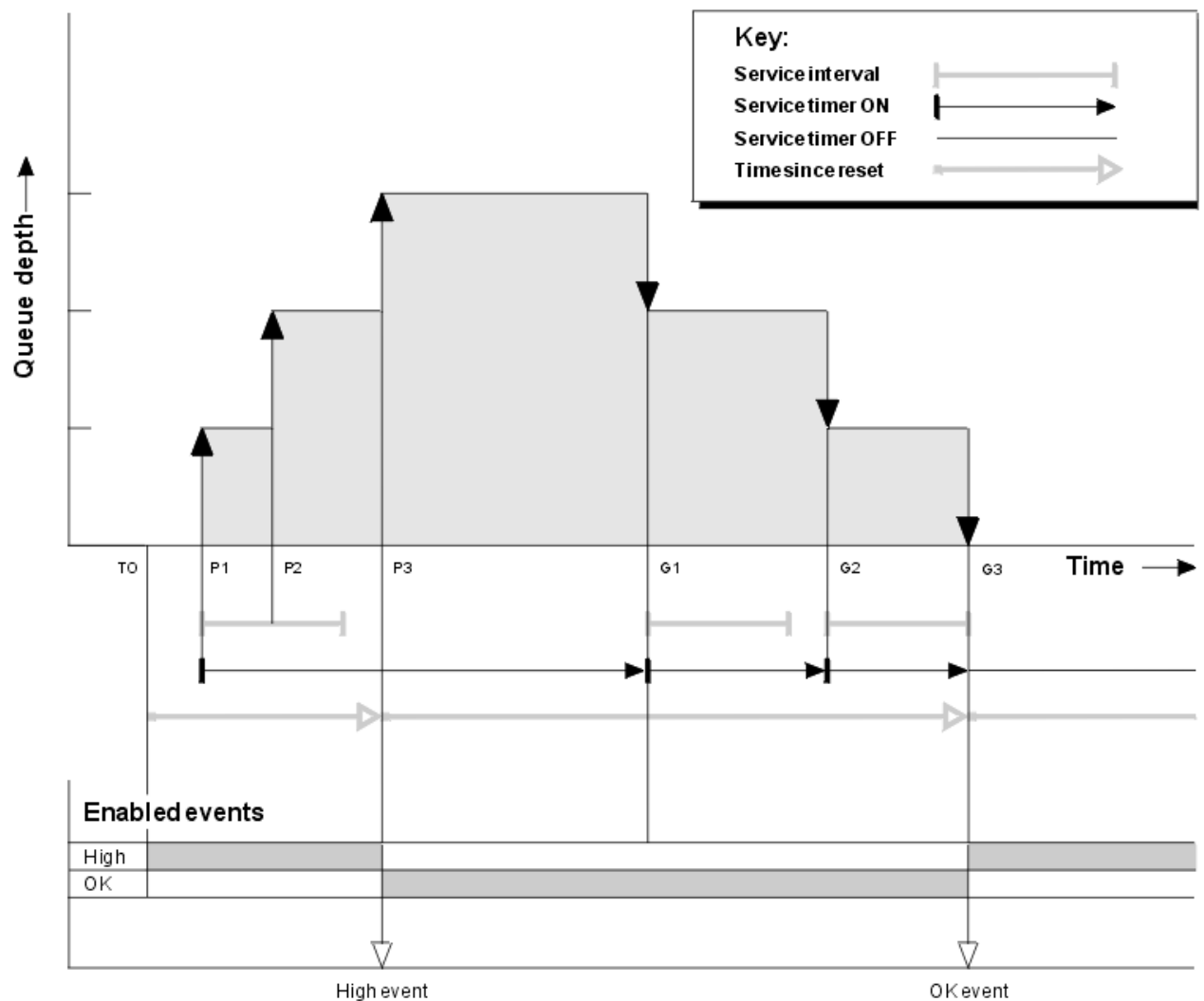


圖 6: 佇列服務間隔事件-範例 3

### 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 26 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		
	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 WebSphere MQ 應用程式中，佇列不得已滿。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，WebSphere 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 28 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：

- 將訊息放入佇列時，會產生「佇列深度高值」事件，導致佇列深度大於或等於「佇列深度高值」限制所決定的值。
  - 「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。
- 當取得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「佇列深度低值」限制所決定的值。
  - 「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。
- 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。
2. 設定下列其中一個屬性，以在必要佇列上啟用事件：
  - `QDepthHighEvent` (MQSC 中的 `QDPHIEV`)
  - `QDepthLowEvent` (MQSC 中的 `QDPLOEV`)
  - `QDepthMaxEvent` (MQSC 中的 `QDPMADEV`)
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：
  - `QDepthHighLimit` (MQSC 中的 `QDEPTHHI`)
  - `QDepthLowLimit` (MQSC 中的 `QDEPTHLO`)

**限制：** `QDEPTHHI` 不得小於 `QDEPTHLO`。

如果 `QDEPTHHI` 等於 `QDEPTHLO`，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：

當取得作業從佇列中移除過期訊息，導致佇列深度小於或等於「佇列深度低限制」所決定的值時，不會產生「佇列深度低值」事件。

只有在順利完成取得作業期間，IBM WebSphere MQ 才會產生低事件訊息。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，不會重設佇列深度高事件及佇列深度低事件。

## 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)
```

## 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。

要配置此佇列的 WebSphere 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 26 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 27 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 29 頁的圖 7](#) 顯示隨時間變化的佇列深度。

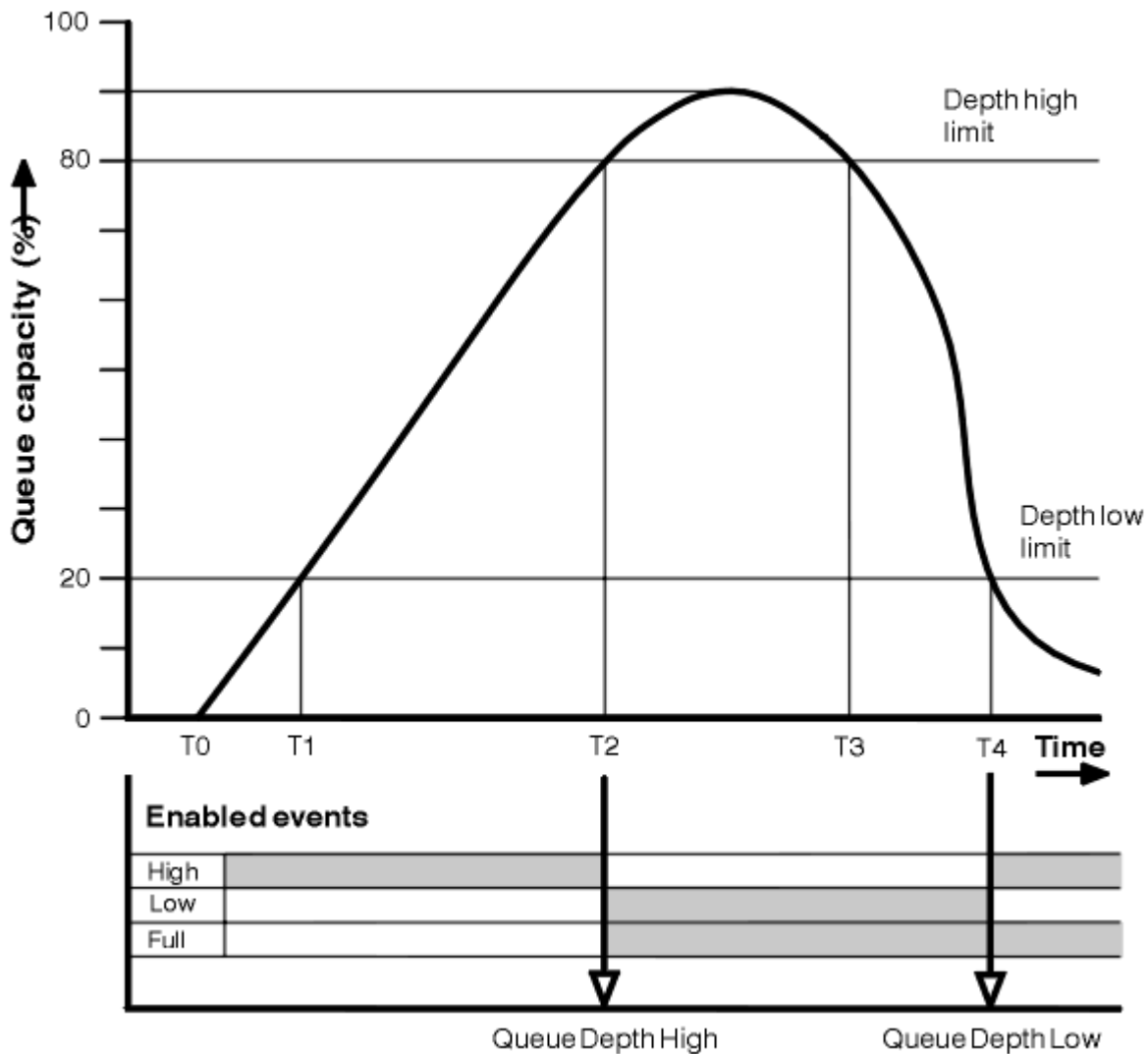


圖 7: 佇列深度事件 (1)

### 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 30 頁的表 8 彙總佇列事件統計資料，並 第 30 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 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 31 頁的圖 8 顯示隨時間變化的佇列深度。

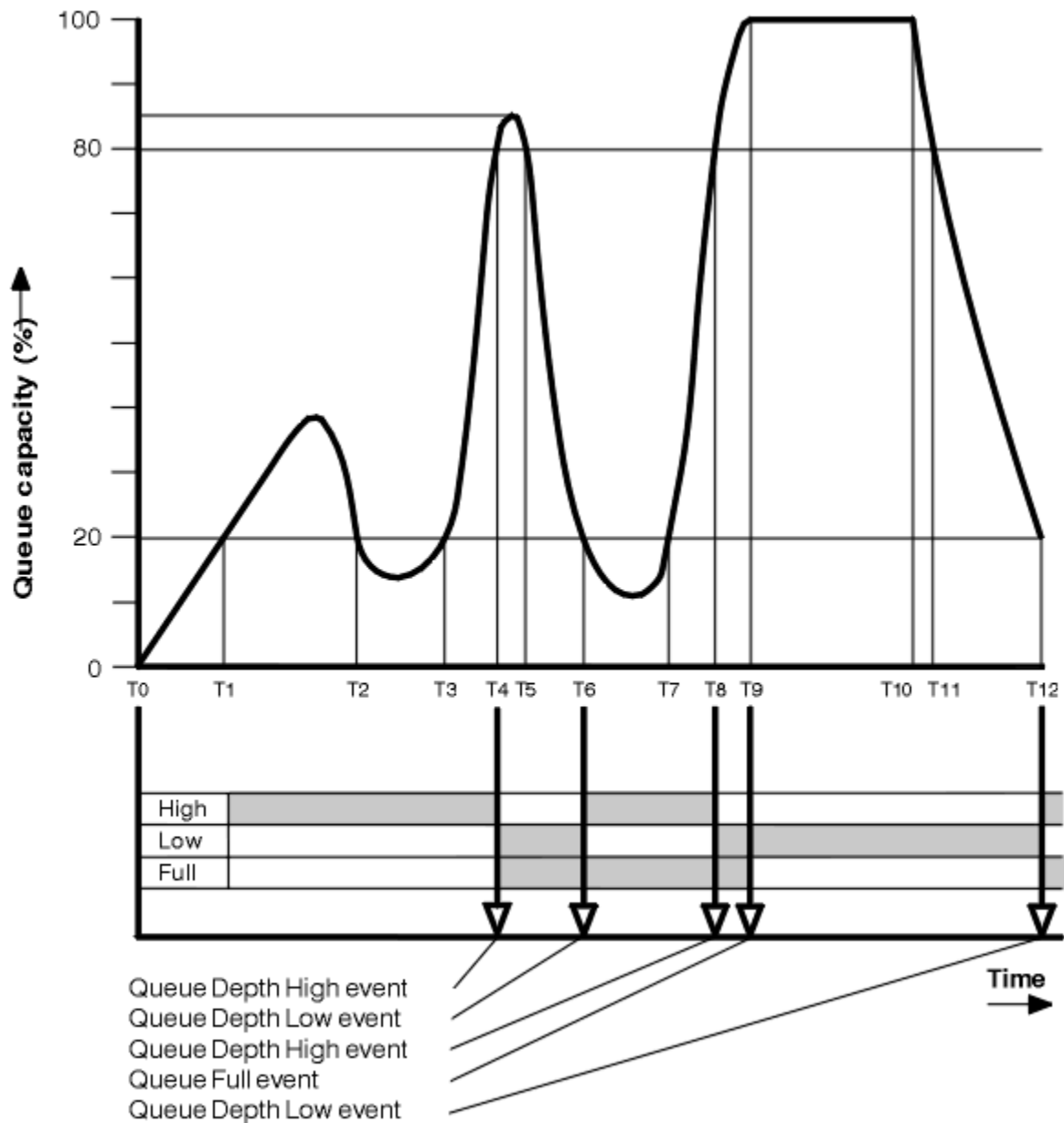


圖 8: 佇列深度事件 (2)

### 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

在此範例中, [第 32 頁的表 10](#) 彙總佇列事件統計資料, 並 [第 32 頁的表 11](#) 彙總在不同時間啟用哪些事件。

	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。

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。

如果是變更物件事件, 則會產生兩則訊息, 其中一則含有變更之前的資訊, 另一則含有變更之後的資訊。

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 11 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 7 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 MAXSMSGS
  - 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

[可程式指令格式簡介](#)

[第 32 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](#)

[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 32 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](#)

## 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 32 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，在分散式平台上，即使原始要求是 MQSC 指令，指令資料格式一律為 PCF 格式。

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 7 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

### 未產生指令事件時

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 35 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](#)

[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 35 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 36 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](#)

## 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的通知。

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。

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 7 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

## 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。

**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 37 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的通知。

## 相關參考

[LoggerEvent \(MQLONG\)](#)

[RESET QMGR](#)

## 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 37 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的通知。

## 相關參考

[LoggerEvent \(MQLONG\)](#)

第 37 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

## 監視日誌程式事件佇列的範例程式

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: AMQSLOG0.C
/*
/* Description: Sample C program to monitor the logger event queue and output
/* a message to stdout when a logger event occurs
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (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.
/* <NOC_COPYRIGHT>
/*
/* Function: AMQSLOG is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and puts the contents
/* of the message to stdout.
/*
/*
/* AMQSLOG 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_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"};

/*****
/* 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;
    PMQCHAR pBuffer = NULL;

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

    /*****/
    /* Parse any command line options */
    /*****/

    if (argc > 1)
        stncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);

    pBuffer = (char *)malloc(MAX_MESSAGE_LENGTH);
    if (!pBuffer)
    {
        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 != MQCC_OK)
    {
        printf("Error in call to MQCONN, Reason %d, CompCode %d\n", Reason,
              CompCode);
        goto MOD_EXIT;
    }

    /* Open the logger event queue for input */

    stncpy(ObjDesc.ObjectQMgrName, QMName, MQ_Q_MGR_NAME_LENGTH);
    stncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

```

```

MQOPEN( hConn,
        &ObjDesc,
        MQOO_INPUT_EXCLUSIVE,
        &hEventQueue,
        &CompCode,
        &Reason);
if (Reason)
{
    printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %d\n",
           ObjDesc.ObjectQMgrName,
           ObjDesc.ObjectName,
           Reason);

    goto MOD_EXIT;
}
else
{
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:

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

/*****
/* 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 msg*/
/*
/* Output Parameters: None */
/*
/* Logic: Wait for messages to appear on the logger event queue and display */
/* their 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 Reason = 0;
    MQLONG CompCode;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    /* Set timeout value */
    Gmo.Options |= MQGMO_WAIT;
    Gmo.Options |= MQGMO_CONVERT;
    Gmo.WaitInterval = MQWI_UNLIMITED;
    /*****
    /* Process response Queue */
    /*****
    while (Reason == MQCC_OK)
    {
        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 != MQCC_OK)
{
    switch(Reason)
    {
        case MQRC_NO_MSG_AVAILABLE:
            printf("Timed out");
            break;

        default:
            printf("MQGET failed RC(%d)\n", Reason);
            break;
    }
    goto MOD_EXIT;
}

/*****
/* Only expect PCF event messages on this queue */
/*****
if (memcmp(Mqmd.Format, MQFMT_EVENT, sizeof(Mqmd.Format)))
{
    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)
{
    printf("-----\n");
    printf("Event Message Received\n");

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

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

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

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

            fwrite( pCfst -> String, pCfst -> StringLength, 1, stdout);
            pPCFCmd += pCfst -> StrucLength;
            break;
        default:
            printf("Unrecognised datatype %d returned\n",pCfst->Type);

```

```

        goto MOD_EXIT;
    }
    putchar('\n');
}
printf("-----\n");
}
MOD_EXIT:
    return;
}

/*****
/* Function: ParmToString
/*****
/*
/* Input Parameters: Parameter for which to get string description
/*
/* Output Parameters: None
/*
/* Logic: Takes a parameter as input and returns a pointer to a string
/* description for that parameter, or NULL if the parameter does not
/* have an associated string description
/*****

static PMQCHAR ParmToString(MQLONG Parameter){
    long i;
    for (i=0 ; i< sizeof(ParmTable)/sizeof(ParmTableEntry); i++)
    {
        if (ParmTable[i].ConstVal == Parameter ParmTable[i].Desc)
            return ParmTable[i].Desc;
    }
    return NULL;
}

```

## 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 38 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 36 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 37 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

## 監視檢測事件的範例程式

請利用這個頁面來檢視監視檢測事件的範例 C 程式

此範例程式不是任何 IBM WebSphere MQ 產品的一部分，因此未提供作為實際實體項目。此範例不完整，因為它未列舉所指定動作的所有可能結果。不過，您可以使用此範例作為您自己使用事件之程式的基礎，特別是事件訊息中使用的 PCF 格式。不過，您必須先修改此程式，然後才能在您自己的系統上執行它。

```

/*****/
/*
/* Program name: EVMON
/*
/* Description: C program that acts as an event monitor
/*
/*
/*
/*****/
/*
/* Function:
/*
/*
/*
/* EVMON is a C program that acts as an event monitor - reads an
/* event queue and tells you if anything appears on it
/*
/*
/* Its first parameter is the queue manager name, the second is
/* the event queue name. If these are not supplied it uses the
/* defaults.
/*
/*
/*****/
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifndef min
#define min(a,b) ((a) < (b)) ? (a) : (b)
#endif

/*****/
/* includes for MQI
/*
/*****/
#include <cmqc.h>
#include <cmqcf.h>
void printfmqcfst(MQCFST* pmqcfst);
void printfmqcfm(MQCFIN* pmqcfst);
void printreas(MQLONG reason);

#define PRINTREAS(param) \
    case param: \
        printf("Reason = %s\n",#param); \
        break;

/*****/
/* global variable
/*
/*****/
MQCFH evtmsg; /* evtmsg message buffer */

int main(int argc, char **argv)
{
/*****/
/* declare variables
/*
/*****/
int i; /* auxiliary counter
/*
/*****/
/* Declare MQI structures needed
/*
/*****/
MQOD od = {MQOD_DEFAULT}; /* Object Descriptor
/*
MQMD md = {MQMD_DEFAULT}; /* Message Descriptor
/*
MQGMO gmo = {MQGMO_DEFAULT}; /* get message options
/*
/*****/
/* note, uses defaults where it can
/*
/*****/

MQHCONN Hcon; /* connection handle
/*
MQHOBJ Hobj; /* object handle
/*
MQLONG Q_options; /* MQOPEN options
/*
MQLONG C_options; /* MQCLOSE options
/*
MQLONG CompCode; /* completion code
/*
MQLONG OpenCode; /* MQOPEN completion code
/*
MQLONG Reason; /* reason code
/*
MQLONG CReason; /* reason code for MQCONN
/*
MQLONG buflen; /* buffer length
/*
MQLONG evtmsglen; /* message length received
/*

```

```

MQCHAR  command[1100];          /* call command string ... */
MQCHAR  p1[600];                /* ApplId insert           */
MQCHAR  p2[900];                /* evtmsg insert           */
MQCHAR  p3[600];                /* Environment insert      */
MQLONG  mytype;                 /* saved application type  */
char     QMName[50];            /* queue manager name      */
MQCFST  *paras;                 /* the parameters          */
int      counter;               /* loop counter            */
time_t   ltime;

/*****
/* Connect to queue manager
*****/
QMName[0] = 0;                  /* default queue manager */
if (argc > 1)
    strcpy(QMName, argv[1]);
MQCONN(QMName,                  /* queue manager          */
        &Hcon,                   /* connection handle     */
        &CompCode,              /* completion code       */
        &CReason);             /* reason code           */

/*****
/* Initialize object descriptor for subject queue
*****/
strcpy(od.ObjectName, "SYSTEM.ADMIN.QMGR.EVENT");
if (argc > 2)
    strcpy(od.ObjectName, argv[2]);

/*****
/* Open the event queue for input; exclusive or shared. Use of
/* the queue is controlled by the queue definition here
*****/

O_options = MQOO_INPUT_AS_Q_DEF /* open queue for input  */
            + MQOO_FAIL_IF_QUIESCING /* but not if qmgr stopping */
            + MQOO_BROWSE;
MQOPEN(Hcon,                    /* connection handle     */
        &od,                    /* object descriptor for queue*/
        O_options,              /* open options          */
        &Hobj,                 /* object handle         */
        &CompCode,             /* completion code       */
        &Reason);             /* reason code           */

/*****
/* Get messages from the message queue
*****/
while (CompCode != MQCC_FAILED)
{
    /*****
    /* I don't know how big this message is so just get the
    /* descriptor first
    *****/
    gmo.Options = MQGMO_WAIT + MQGMO_LOCK
                 + MQGMO_BROWSE_FIRST + MQGMO_ACCEPT_TRUNCATED_MSG;
    gmo.WaitInterval = MQWI_UNLIMITED; /* no time limit */
    buflen = 0; /* amount of message to get */

    /*****
    /* clear selectors to get messages in sequence
    *****/
    memcpy(md.MsgId, MQMI_NONE, sizeof(md.MsgId));
    memcpy(md.CorrelId, MQCI_NONE, sizeof(md.CorrelId));

    /*****
    /* wait for event message
    *****/
    printf("...>\n");
    MQGET(Hcon,                  /* connection handle     */
          Hobj,                  /* object handle         */
          &md,                  /* message descriptor    */
          &gmo,                 /* get message options  */
          buflen,                /* buffer length         */
          evtmsg,                /* evtmsg message buffer */
          &evtmglen,           /* message length       */
          &CompCode,           /* completion code       */
          &Reason);           /* reason code           */
}

```

```

/*****
/* report reason, if any */
/*****
if (Reason != MQRC_NONE && Reason != MQRC_TRUNCATED_MSG_ACCEPTED)
{
    printf("MQGET ==> %ld\n", Reason);
}
else
{
    gmo.Options = MQGMO_NO_WAIT + MQGMO_MSG_UNDER_CURSOR;
    buflen = evtmsglen; /* amount of message to get */
    evtmsg = malloc(buflen);
    if (evtmsg != NULL)
    {
        /*****
        /* clear selectors to get messages in sequence */
        /*****
        memcpy(md.MsgId, MQMI_NONE, sizeof(md.MsgId));
        memcpy(md.CorrelId, MQCI_NONE, sizeof(md.CorrelId));

        /*****
        /* get the event message */
        /*****
        printf("...>\n");
        MQGET(Hcon, /* connection handle */
            Hobj, /* object handle */
            &md, /* message descriptor */
            &gmo, /* get message options */
            buflen, /* buffer length */
            evtmsg, /* evtmsg message buffer */
            &evtmsglen, /* message length */
            &CompCode, /* completion code */
            &Reason); /* reason code */

        /*****
        /* report reason, if any */
        /*****
        if (Reason != MQRC_NONE)
        {
            printf("MQGET ==> %ld\n", Reason);
        }
    }
    else
    {
        CompCode = MQCC_FAILED;
    }
}
}

/*****
/* . . . process each message received */
/*****
if (CompCode != MQCC_FAILED)
{
    /*****
    /* announce a message */
    /*****
    printf("\a\a\a\a\a");
    time(&ltime);
    printf(ctime(&ltime));

    if (evtmsglen != buflen)
        printf("DataLength = %ld?\n", evtmsglen);
    else
    {
        /*****
        /* right let's look at the data */
        /*****
        if (evtmsg->Type != MQCFT_EVENT)
        {
            printf("Something's wrong this isn't an event message,"
                " its type is %ld\n", evtmsg->Type);
        }
        else
        {
            if (evtmsg->Command == MQCMD_Q_MGR_EVENT)
            {
                printf("Queue Manager event: ");
            }
            else
            if (evtmsg->Command == MQCMD_CHANNEL_EVENT)

```

```

    {
        printf("Channel event: ");
    }
    else
    :

    {
        printf("Unknown Event message, %ld.",
            evtmsg->Command);
    }

    if      (evtmsg->CompCode == MQCC_OK)
        printf("CompCode(OK)\n");
    else if (evtmsg->CompCode == MQCC_WARNING)
        printf("CompCode(WARNING)\n");
    else if (evtmsg->CompCode == MQCC_FAILED)
        printf("CompCode(FAILED)\n");
    else
        printf("* CompCode wrong * (%ld)\n",
            evtmsg->CompCode);

    if (evtmsg->StrucLength != MQCFH_STRUC_LENGTH)
    {
        printf("it's the wrong length, %ld\n", evtmsg->StrucLength);
    }

    if (evtmsg->Version != MQCFH_VERSION_1)
    {
        printf("it's the wrong version, %ld\n", evtmsg->Version);
    }

    if (evtmsg->MsgSeqNumber != 1)
    {
        printf("it's the wrong sequence number, %ld\n",
            evtmsg->MsgSeqNumber);
    }

    if (evtmsg->Control != MQCFC_LAST)
    {
        printf("it's the wrong control option, %ld\n",
            evtmsg->Control);
    }

    printreas(evtmsg->Reason);
    printf("parameter count is %ld\n", evtmsg->ParameterCount);
    /******
    /* get a pointer to the start of the parameters */
    /******

    paras = (MQCFST *) (evtmsg + 1);
    counter = 1;
    while (counter <= evtmsg->ParameterCount)
    {
        switch (paras->Type)
        {
            case MQCFT_STRING:
                printfmqfst(paras);
                paras = (MQCFST *) ((char *) paras
                    + paras->StrucLength);
                break;
            case MQCFT_INTEGER:
                printfmqfin((MQCFIN*) paras);
                paras = (MQCFST *) ((char *) paras
                    + paras->StrucLength);
                break;
            default:
                printf("unknown parameter type, %ld\n",
                    paras->Type);
                counter = evtmsg->ParameterCount;
                break;
        }
        counter++;
    }
}
} /* end evtmsg action */
free(evtmsg);
evtmsg = NULL;
} /* end process for successful GET */
} /* end message processing loop */

```

```

/*****
/* close the event queue - if it was opened */
/*****
if (OpenCode != MQCC_FAILED)
{
    C_options = 0; /* no close options */
    MQCLOSE(Hcon, /* connection handle */
            &Hobj, /* object handle */
            C_options,
            &CompCode, /* completion code */
            &Reason); /* reason code */
/*****
/* Disconnect from queue manager (unless previously connected) */
/*****
if (CReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&Hcon, /* connection handle */
          &CompCode, /* completion code */
          &Reason); /* reason code */

/*****
/*
/* END OF EVMON */
/*
/*****
}

#define PRINTPARAM(param) //
    case param: //
    { //
        char *p = #param; //
        strncpy(thestring,pmqcfst->String,min(sizeof(thestring), //
            pmqcfst->StringLength)); //
        printf("%s %s\n",p,thestring); //
    } //
    break; //

#define PRINTAT(param) //
    case param: //
        printf("MQIA_APPL_TYPE = %s\n",#param); //
    break; //

void printfmqcfst(MQCFST* pmqcfst)
{
    char thestring[100];

    switch (pmqcfst->Parameter)
    {
        PRINTPARAM(MQCA_BASE_Q_NAME)
        PRINTPARAM(MQCA_PROCESS_NAME)
        PRINTPARAM(MQCA_Q_MGR_NAME)
        PRINTPARAM(MQCA_Q_NAME)
        PRINTPARAM(MQCA_XMIT_Q_NAME)
        PRINTPARAM(MQCACF_APPL_NAME)
        :
        default:
            printf("Invalid parameter, %ld\n",pmqcfst->Parameter);
            break;
    }
}

void printfmqcfin(MQCFIN* pmqcfst)
{
    switch (pmqcfst->Parameter)
    {
        case MQIA_APPL_TYPE:
            switch (pmqcfst->Value)
            {
                PRINTAT(MQAT_UNKNOWN)
                PRINTAT(MQAT_OS2)
                PRINTAT(MQAT_DOS)
                PRINTAT(MQAT_UNIX)
                PRINTAT(MQAT_QMGR)
                PRINTAT(MQAT_OS400)
                PRINTAT(MQAT_WINDOWS)
                PRINTAT(MQAT_CICS_VSE)
                PRINTAT(MQAT_VMS)
            }
        }
    }
}

```

```

        PRINTAT(MQAT_GUARDIAN)
        PRINTAT(MQAT_VOS)
    }
    break;
case MQIA_Q_TYPE:
    if (pmqcfst->Value == MQQT_ALIAS)
    {
        printf("MQIA_Q_TYPE is MQQT_ALIAS\n");
    }
    else
    {
        if (pmqcfst->Value == MQQT_REMOTE)
        {
            printf("MQIA_Q_TYPE is MQQT_REMOTE\n");
            if (evtmsg->Reason == MQR_ALIAS_BASE_Q_TYPE_ERROR)
            {
                printf("but remote is not valid here\n");
            }
        }
        else
        {
            printf("MQIA_Q_TYPE is wrong, %ld\n",pmqcfst->Value);
        }
    }
    break;

    case MQIACF_REASON_QUALIFIER:
    printf("MQIACF_REASON_QUALIFIER %ld\n",pmqcfst->Value);
    break;

case MQIACF_ERROR_IDENTIFIER:
    printf("MQIACF_ERROR_IDENTIFIER %ld (X'%lX')\n",
        pmqcfst->Value,pmqcfst->Value);
    break;

case MQIACF_AUX_ERROR_DATA_INT_1:
    printf("MQIACF_AUX_ERROR_DATA_INT_1 %ld (X'%lX')\n",
        pmqcfst->Value,pmqcfst->Value);
    break;

case MQIACF_AUX_ERROR_DATA_INT_2:
    printf("MQIACF_AUX_ERROR_DATA_INT_2 %ld (X'%lX')\n",
        pmqcfst->Value,pmqcfst->Value);
    break;
:
default :
    printf("Invalid parameter, %ld\n",pmqcfst->Parameter);
    break;
}
}

void printreas(MQLONG reason)
{
    switch (reason)
    {
        PRINTREAS(MQRCCF_CFH_TYPE_ERROR)
        PRINTREAS(MQRCCF_CFH_LENGTH_ERROR)
        PRINTREAS(MQRCCF_CFH_VERSION_ERROR)
        PRINTREAS(MQRCCF_CFH_MSG_SEQ_NUMBER_ERR)
        :
        PRINTREAS(MQRC_NO_MSG_LOCKED)
        PRINTREAS(MQRC_CONNECTION_NOT_AUTHORIZED)
        PRINTREAS(MQRC_MSG_TOO_BIG_FOR_CHANNEL)
        PRINTREAS(MQRC_CALL_IN_PROGRESS)
        default:
            printf("It's an unknown reason, %ld\n",
                reason);
            break;
    }
}
}

```

## 相關概念

[第 5 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。

[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。

### 相關參考

[C 程式設計](#)

[第 38 頁的『監視日誌程式事件佇列的範例程式』](#)

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 WebSphere 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。

### 相關概念

[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。
- WebSphere 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，WebSphere MQ 發佈/訂閱引擎處理訊息的活動可能包含下列多項作業：

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，您可能會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。

WebSphere 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 49 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 50 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 56 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
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：

#### WebSphere MQ 6.0 及後續版本中的佇列管理程式

MCA 及連接至佇列管理程式 (WebSphere MQ 6.0 版或後續版本) 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是由 WebSphere MQ 6.0 或後續版本的佇列管理程式所組成，則可以判斷完整的訊息路徑。

#### WebSphere MQ 版本之前的佇列管理程式 6.0

連接至 WebSphere MQ 6.0 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任何 6.0 版之前的 WebSphere MQ 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

WebSphere 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 52 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 56 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 60 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 90 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 WebSphere MQ 顯示遞送應用程式，將活動記錄與追蹤遞送訊息一起使用。

### 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 WebSphere 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。

如果您不想要針對透過佇列管理程式遞送的訊息產生活動報告，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 WebSphere 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 `send` 的 *OperationType*，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 開始執行 `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` 並擷取具有符合原始訊息之 *MsgId* 的 *CorrelId* 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 6.0 版的 WebSphere MQ 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 WebSphere 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追蹤路徑傳訊中，PCF 群組稱為 *TraceRoute* PCF 群組。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。其他資訊會在 *TraceRoute* PCF 群組中傳回。*TraceRoute* PCF 群組中包含的資訊僅在產生活動報告時才正確。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 56 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 參數可控制追蹤路徑訊息是否在到達時放置在目標佇列上，或是否捨棄。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 WebSphere 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 MQRROUTE\_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 WebSphere 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 MQRROUTE\_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 53 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 59 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 59 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 WebSphere MQ 顯示路徑應用程式。

## 相關工作

第 53 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。值可以為：

### 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，且下一個要對追蹤路徑訊息執行的活動：

- 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)
- 會導致在追蹤路徑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，且下一個要對追蹤路徑訊息執行的活動：

- 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)
- 會導致在追蹤路徑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。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 在重新啟動通道之前不會偵測變更。

## 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。

如果所執行活動的明細層次不小於 明細 參數指定的明細層次，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\_ACUMULATE\_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 *Deliver* 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 WebSphere 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，其中 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 群組及一個以上 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。活動 當 MCA 或使用者撰寫的應用程式代表其執行活動時，PCF 群組會寫入追蹤路徑訊息的訊息資料。

## WebSphere MQ 顯示路徑應用程式

使用 WebSphere 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 *Format* 參數設為 MQFMT\_ADMIN。您無法將使用者資料新增至 WebSphere MQ 顯示遞送應用程式所產生的追蹤路徑訊息。

**限制:** 在 WebSphere MQ 6.0 版之前，或在 WebSphere MQ for z/OS 佇列管理程式上，無法發出 dspmqrte。如果您要第一個佇列管理程式，追蹤路徑訊息會透過遞送至此類型的佇列管理程式，請使用選用參數 -c，以 WebSphere MQ 6.0 版或更新版本用戶端的身分連接至佇列管理程式。

## 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:** WebSphere 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:** WebSphere 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，則可以新增其他參數，如第 64 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：

### 詳細資料

指定要記錄之活動資訊的詳細層次。值可以為：

#### **MQROUTE\_DETAIL\_LOW**

只會記錄使用者應用程式所執行的活動。

#### **MQROUTE\_DETAIL\_MEDIUM**

應該記錄 MQROUTE\_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參數明細指定的明細層次。
- 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* 會依 1 遞增。

### **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。

## **MQRROUTE\_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。值可以為：

### **MQRROUTE\_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。

### **MQRROUTE\_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：

- WebSphere MQ 第 6 版 (或更新版本) 佇列管理程式會捨棄追蹤路徑訊息。
- WebSphere MQ 第 6 版 (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。

#### **MQRROUTE\_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

指定追蹤路徑訊息可以轉遞至何處。值可以為：

#### **MQRROUTE\_FORWARD\_IF\_SUPPORTED**

追蹤路徑訊息只會轉遞至佇列管理程式，以允許使用 *TraceRoute* 群組中 *Deliver* 參數的值。

#### **MQRROUTE\_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 62 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 62 頁的『2』。
2. 判定 *TraceRoute* 群組中的 遞送 參數是否包含 MQRROUTE\_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 62 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 MQRROUTE\_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 MQRROUTE\_DELIVER\_NO，則演算法會繼續執行步驟 第 62 頁的『4』。
4. 判定 *TraceRoute* 群組中的 *Forward* 參數是否包含 MQRROUTE\_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 62 頁的『5』。
5. 從追蹤路徑訊息中的 *TraceRoute* PCF 群組，決定參數 *Forward* 的值。
  - 如果將 轉遞 指定為 MQRROUTE\_FORWARD\_IF\_SUPPORTED，則會拒絕追蹤路徑訊息，並提供意見 MQFB\_NOT\_FORWARDED。
  - 如果 轉遞 指定為 MQRROUTE\_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。值可以為：

#### **MQRROUTE\_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。

#### **MQRROUTE\_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 WebSphere 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，您可以找到追蹤路徑訊息，該訊息必須在 *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 *TraceRoute* PCF 群組中具有指定為 `MQROUTE_ACCUMULATE_IN_MSG` 或 `MQROUTE_ACCUMULATE_AND_REPLY` 的累計參數時，才能從追蹤路徑訊息獲得活動資訊。

對於要遞送至目標佇列的追蹤路徑訊息，*TraceRoute* PCF 群組中的遞送參數必須指定為 `MQROUTE_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 WebSphere 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，可以手動排序它們，或使用 WebSphere MQ 顯示遞送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 WebSphere 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 WebSphere 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 <b>IBM WebSphere MQ</b> 顯示路徑應用程式顯示的名稱，其會將 <b>ParameterValue</b> 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:	GroupName.
值:	要顯示的名稱。

## ParameterValue

表 15: 參數值	
說明	包含要由 <b>IBM WebSphere MQ</b> 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:	GroupName.
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 WebSphere 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。

### 範例 1

使用者應用程式會以 WebSphere MQ 顯示遞送應用程式無法辨識參數 ID 的格式來記錄其他活動資訊。

1. WebSphere 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，WebSphere 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 WebSphere 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 WebSphere MQ 顯示路徑應用程式。其他活動資訊顯示如下：

```
65536: 'Red'
```

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

如需 WebSphere 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 66 頁的『範例 2』](#)。

## 範例 2

使用者應用程式會以參數 ID 可由 IBM WebSphere MQ 顯示路徑應用程式辨識的格式來記錄其他活動資訊。

1. IBM WebSphere 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 65 頁的『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 WebSphere 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。
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 IBM WebSphere MQ 顯示路徑應用程式顯示的值。
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 參數是以 IBM WebSphere MQ 顯示路徑應用程式可辨識參數 ID 是 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 WebSphere MQ 顯示路徑應用程式。其他活動資訊顯示如下：

```
Color: 'Red'
```

IBM WebSphere 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。

## WebSphere MQ 顯示路徑應用程式

使用 WebSphere MQ 顯示畫面遞送應用程式 (**dspmqrte**)，利用指令行介面來使用追蹤遞送訊息及與追蹤遞送訊息相關的活動資訊。

**註：**若要對佇列管理程式執行「用戶端應用程式」，必須安裝「用戶端附件」特性。

基於下列目的，您可以使用 WebSphere MQ 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 WebSphere 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，WebSphere MQ 顯示畫面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 WebSphere 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，包括在透過佇列管理程式網路遞送時如何處理它。

### 相關參考

[dspmqrte](#)

## 佇列管理程式連線

請利用這個頁面來指定 WebSphere MQ 顯示畫面路徑應用程式所連接的佇列管理程式

**-c**

指定 WebSphere MQ 顯示遞送應用程式連接為用戶端應用程式。

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

**-m QMgrName**

WebSphere 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 WebSphere 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 TargetTopicString**

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 WebSphere MQ 顯示畫面遞送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 WebSphere MQ 顯示路徑應用程式的追蹤路徑訊息主題字串

### **-ts TopicName**

指定主題字串，供 WebSphere MQ 顯示路徑應用程式發佈追蹤路徑訊息，並將這個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 WebSphere 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 WebSphere 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 Persistence**

指定所產生追蹤路徑訊息的持續性。持續性的可能值如下：

是

產生的追蹤路徑訊息是持續的。(MQPER\_PERSISTENT)。

否

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 Priority**

指定追蹤路徑訊息的優先順序。優先順序的值大於或等於 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 Expiry**

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。

#### **到期**

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 WebSphere 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：

### **activity 報告選項，使用 -ro 指定**

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，則活動資訊 **不會** 在追蹤路徑訊息內累計。

### **-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 WebSphere MQ 佇列管理程式會捨棄追蹤路徑訊息。
- IBM WebSphere MQ 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，則活動資訊 **不會** 在追蹤路徑訊息內累計。

**-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，則追蹤路徑訊息 **不會** 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 Detail**

指定記錄的活動。明細的可能值如下：

**低**

僅記錄使用者定義應用程式所執行的活動。

**中型**

記錄 **low** 中指定的活動。此外，還會記錄 MCA 所執行的發佈活動和活動。

**high**

記錄 **low** 及 **medium** 中指定的活動。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 WebSphere 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 WebSphere 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 WebSphere 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 ReplyToQMgr**

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 WebSphere 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 Deliver**

指定是否在到達時將追蹤路徑訊息遞送至目標佇列。遞送可能的值如下：

是

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。

否

到達時，追蹤路徑訊息 **不會** 放置到目標佇列。

如果您未指定此參數，則追蹤路徑訊息 **不會** 放置到目標佇列。

**-f Forward**

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 60 頁的『TraceRoute PCF 群組』。向前的可能值如下：

## 全部

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 6.0 版的 IBM WebSphere MQ 佇列管理程式，則不管 *-d Deliver* 參數值為何，追蹤路徑訊息將無法辨識且可遞送至本端佇列。

## 支援的

追蹤路徑訊息只會轉遞至佇列管理程式，以允許使用 *TraceRoute* PCF 群組中的 *Deliver* 參數

如果您未指定此參數，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 Activities**

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 Expiry**

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：

#### 是

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。

#### 否

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。

### **discard 報告選項，使用 -ro 指定**

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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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。

#### **全部**

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**

畫面上會顯示 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**

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**

即會顯示 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 WebSphere 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 64 頁的『其他活動資訊』中所述。

## WebSphere MQ 顯示路徑應用程式範例

下列範例顯示如何使用 WebSphere 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，WebSphere MQ 顯示畫面路徑應用程式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。使用活動報告傳回至 WebSphere MQ 顯示畫面遞送應用程式的活動資訊會依序排列並顯示。

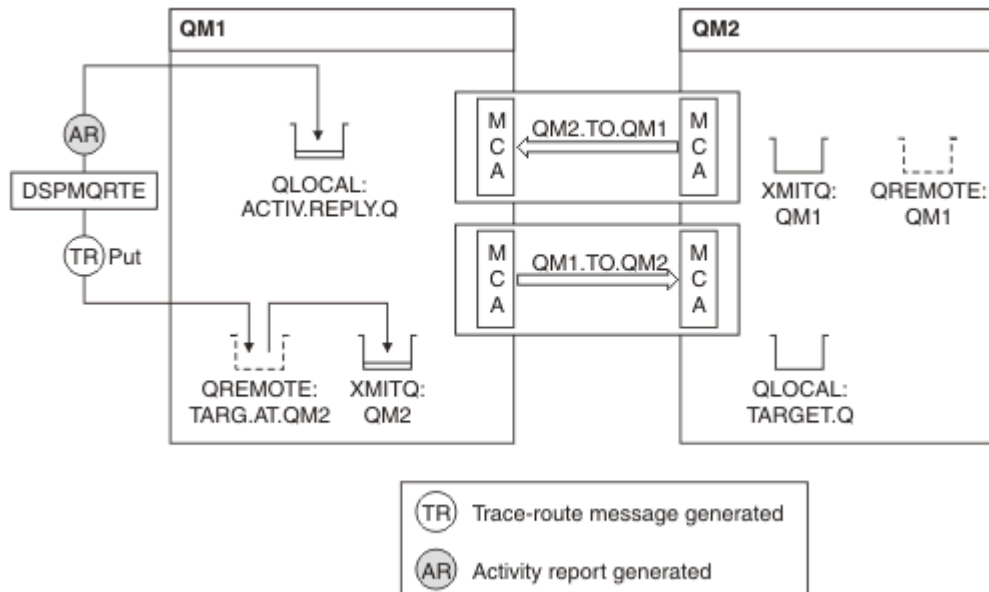


圖 9: 要求活動報告，圖 1

- 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 WebSphere MQ 顯示畫面遞送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。

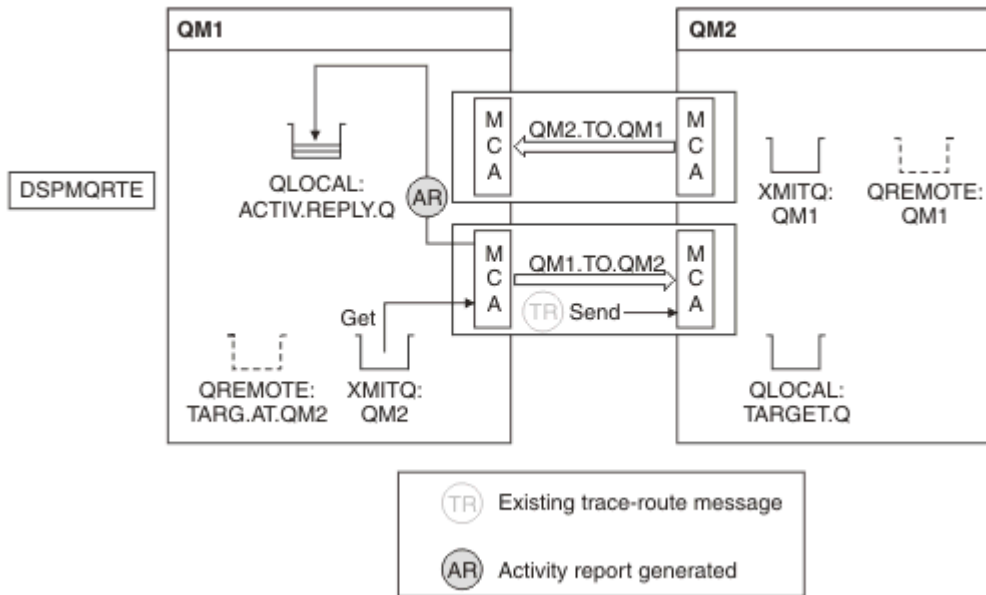


圖 10: 要求活動報告, 圖表 2

- 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 62 頁的『1』](#)、[第 62 頁的『4』](#) 及 [第 62 頁的『5』](#)), 且 MCA 會選擇傳送訊息。
- 然後 MCA 會產生活動報告, 並將它放在回覆佇列 (ACTIV.REPLY.Q)。

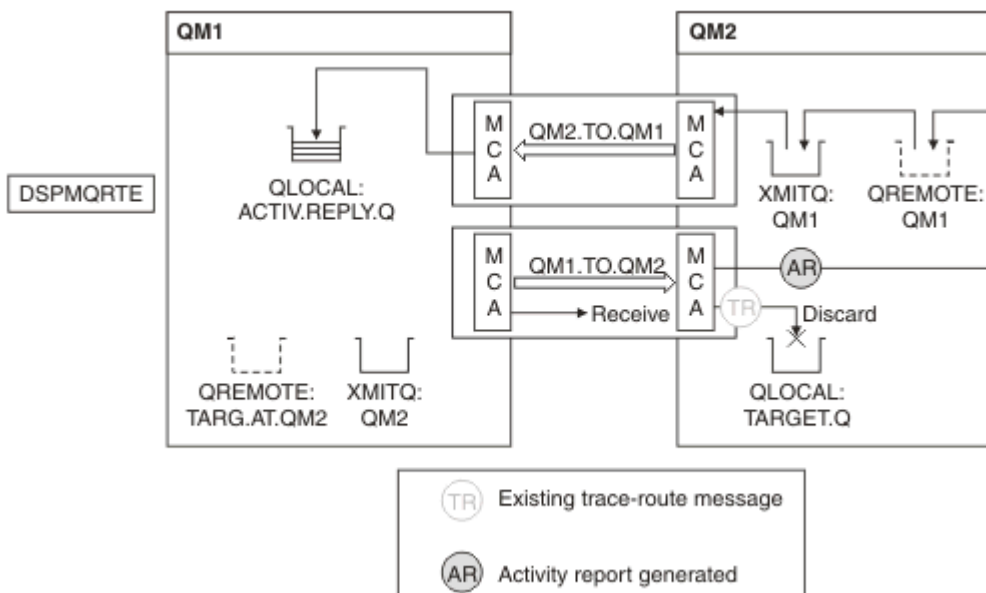


圖 11: 要求活動報告, 圖表 3

- 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。

- 如果追蹤路徑訊息的來源佇列管理程式是 5.3.1 版或更舊版本，則 MCA 會將 TraceRoute PCF 的 DiscontinuityCount 參數增加 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)。

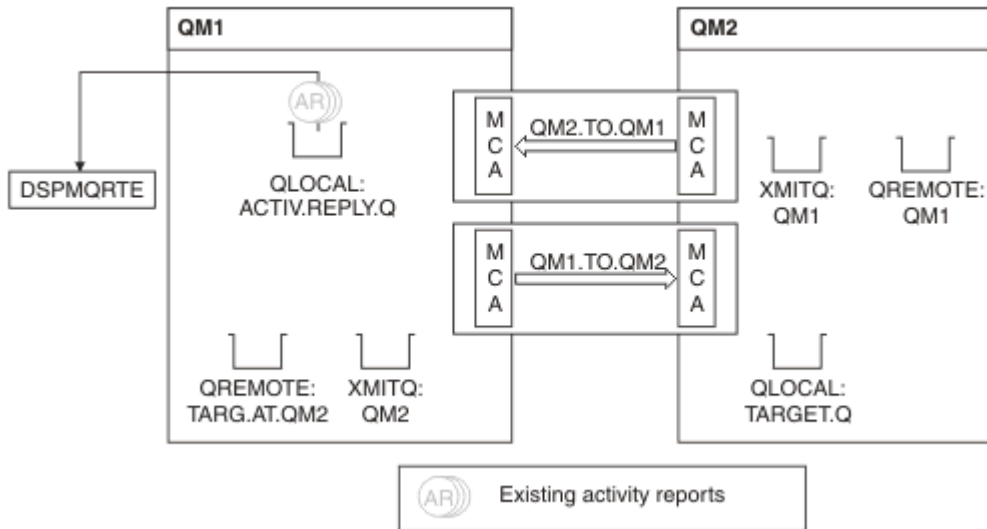


圖 12: 要求活動報告，圖 4

- 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，WebSphere MQ 顯示畫面路徑應用程式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。

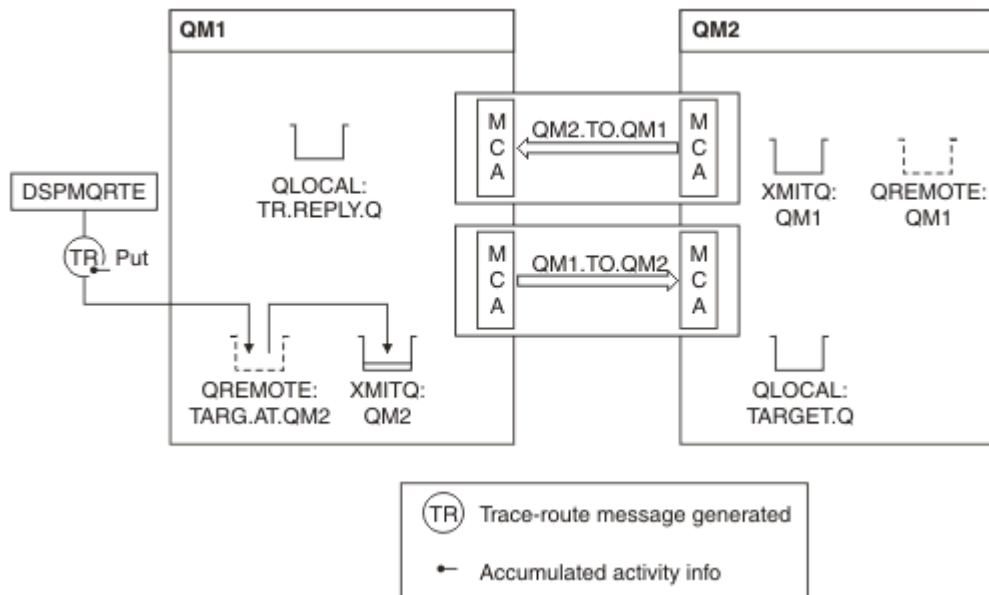


圖 13: 要求追蹤路徑回覆訊息，圖表 1

- 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 WebSphere MQ 顯示畫面遞送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。

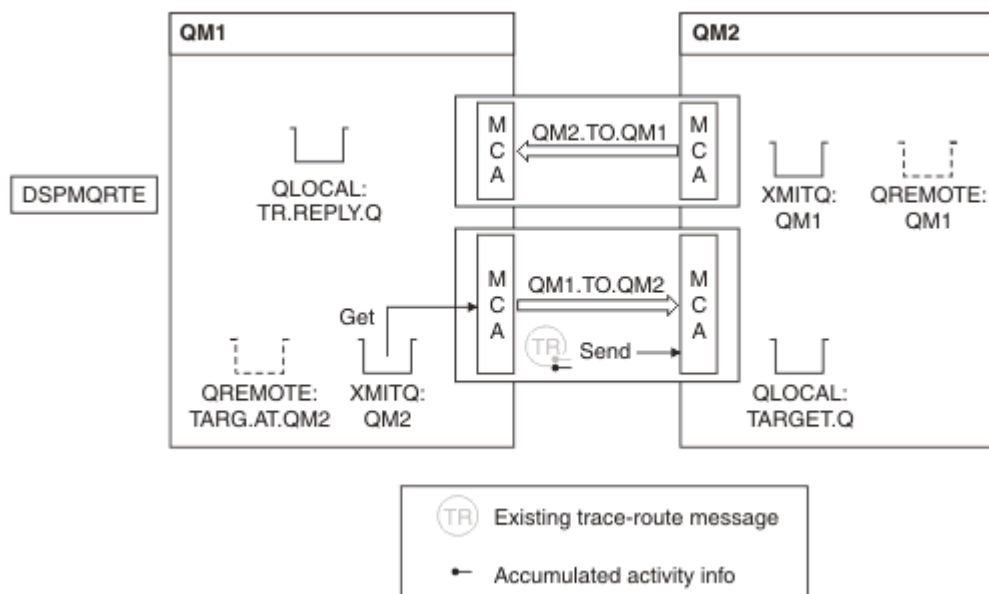


圖 14: 要求追蹤路徑回覆訊息，圖 2

- 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。

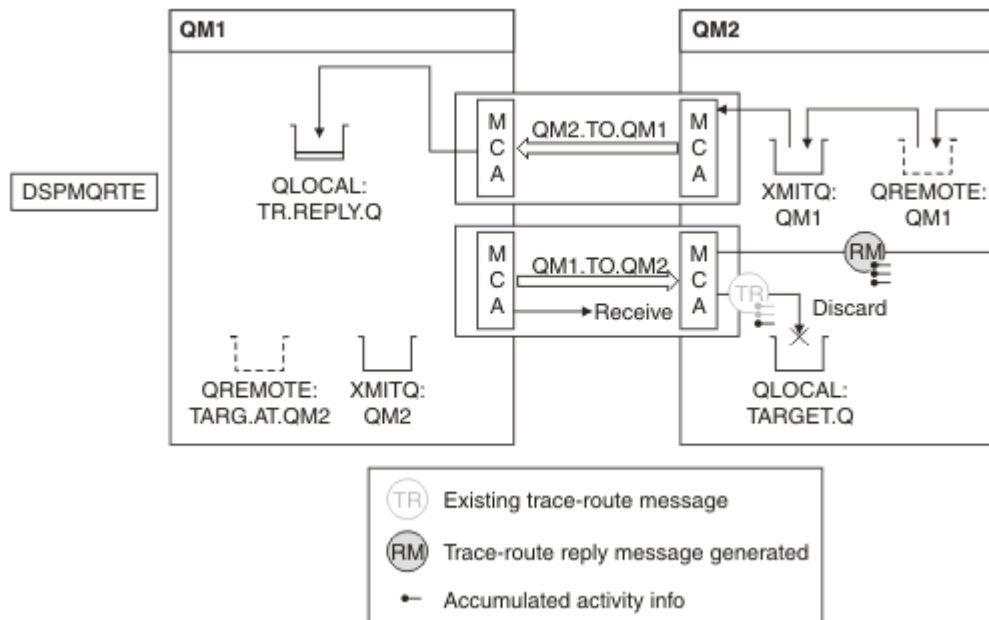


圖 15: 要求追蹤路徑回覆訊息，圖 3

- 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。

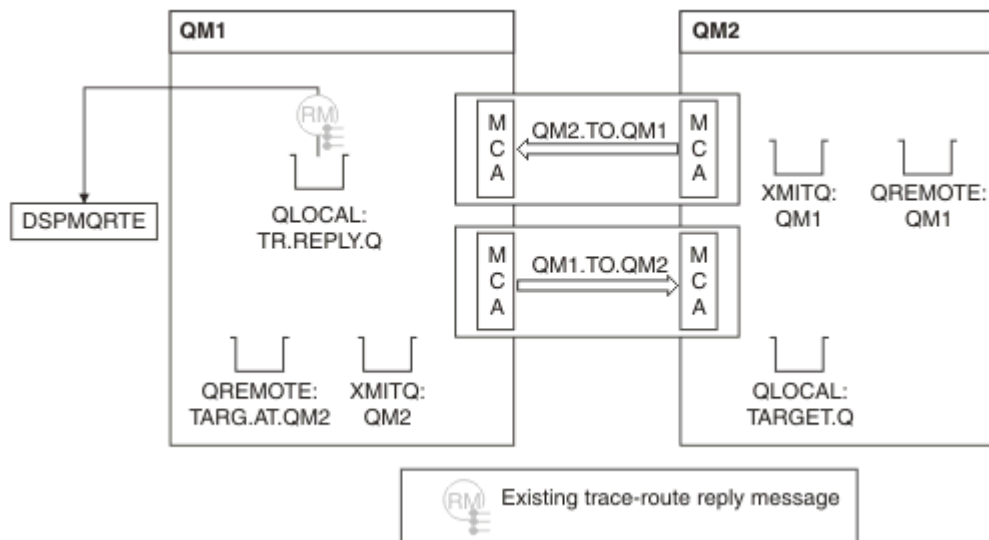


圖 16: 要求追蹤路徑回覆訊息，圖 4

- 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 WebSphere MQ 顯示畫面遞送應用程式從其他佇列讀取活動報告。

此範例與第 73 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。偵測到之後，WebSphere MQ 顯示路徑應用程式會用來從另一個佇列讀取活動報告。

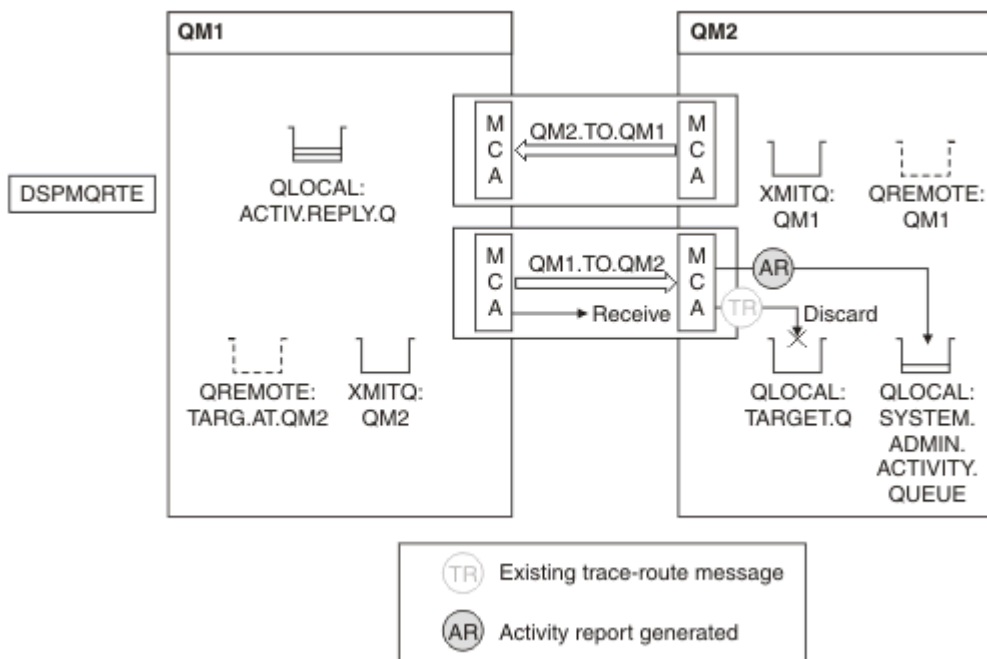


圖 17: 將活動報告遞送至系統佇列，圖 1

- 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)。

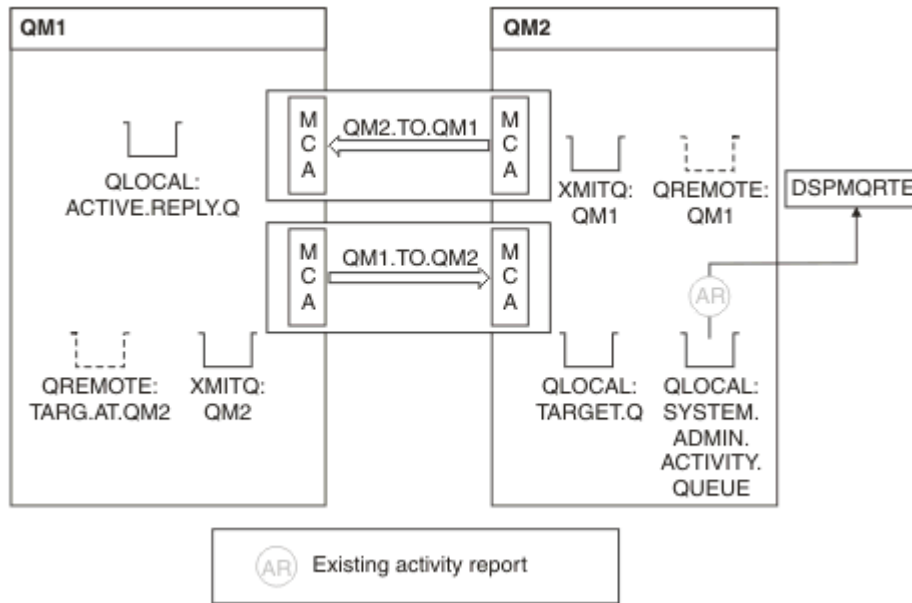


圖 18: 將活動報告遞送至系統佇列，圖 2

- 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'414D51204C4152474551202020202020A3C9154220001505'
  CorrelId: X'414D51204C4152474551202020202020A3C9154220001502'
```

```
EmbeddedMQMD:
  MsgId: X'414D51204C4152474551202020202020A3C9154220001502'
  CorrelId: X'414D51204C4152474551202020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'
```

```
Operation:
  OperationType: Send
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C4152474551202020202020A3C9154220001502'
  CorrelId: X'414D51204C4152474551202020202020A3C9154220001503'
  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 414D51204C4152474551202020202020A3C9154220001502 -v outline
```

其中 414D51204C4152474551202020202020A3C9154220001502 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 414D51204C4152474551202020202020A3C915420001502 -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，WebSphere MQ 顯示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。

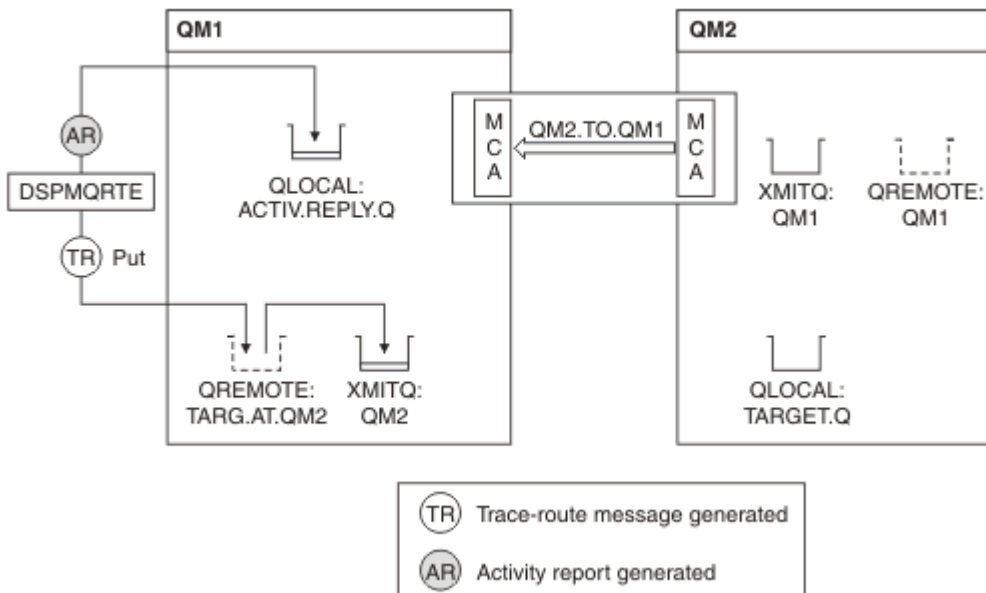


圖 19: 診斷通道問題

- 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 WebSphere 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。

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

第 83 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> <sup>2 3</sup> <sup>7</sup> 已解析佇列名稱 <sup>1</sup> 版 <sup>7</sup> 遠端佇列名稱 <sup>3</sup> <sup>7</sup> 遠端佇列管理程式名稱 <sup>2</sup> <sup>3 4</sup> <sup>5 版</sup> <sup>7</sup> 訂閱層次 <sup>9</sup> 訂閱 ID <sup>9</sup> 意見 <sup>2</sup> <sup>10</sup> 通道名稱 <sup>4</sup> <sup>5</sup> 通道類型 <sup>4</sup> <sup>5</sup> 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 AIX、HP-UX、z/OS、IBM i、Solaris、Linux、Windows 及所有連接至這些系統的 WebSphere MQ MQI 用戶端上受支援。

### **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該值。  
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\_CC SID\_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的參數字集。

### **PCFHeader**

- 說明： 可程式指令格式標頭
- 資料類型： MQCFH。
- 值： 請參閱第 89 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 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。

活動報告訊息資料包含 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)。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:      *ActivityApplName*  
                                 *ActivityApplType*  
                                 *ActivityDescription*  
                                 *Operation*  
                                 *TraceRoute*

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\_QUEUE*。  
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:	<i>MQMD</i> 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:	<i>MQMD</i> 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:	<i>MQMD</i> 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:	<i>MQMD</i> 或 <i>EmbeddedMQMD</i> 。
已傳回:	一律，但「排除的發佈作業」及 <i>MQMD</i> 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:	<i>MQMD</i> 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:	執行活動之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 WebSphere 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:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
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 (已取得或已瀏覽佇列上的訊息)，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 意見回饋 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\_DISCARD\_PUBLISH/MQOPER\_EXCLUDED\_PUBLISH) 作業類型 (已遞送、捨棄或排除發佈/訂閱訊息) 傳回的其他活動報告訊息資料參數。

#### **SubId**

說明: 訂閱 ID。

ID: MQBACF\_SUB\_ID。  
資料類型: MQCFBS。  
包含在 PCF 群組  
中: 作業。  
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  
中: 作業。  
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  
中: 作業。  
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置到佇列) 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 群組中: 作業。  
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 群組中: 作業。  
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 群組中: 作業。  
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 群組中: 作業。  
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 作業。  
中：  
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 作業。  
中：  
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 作業。  
中：  
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 作業。  
中：  
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 WebSphere 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，其中格式欄位設為 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的訊息資料 (如果格式設為 MQFMT\_EMBEDDED\_PCF)。

當使用 WebSphere 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\_ACUMULATE\_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。

表 20: 追蹤路徑訊息格式		
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 , 或者如果 報告 具有任何報告產生選項集, 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：  
**MQEPH\_NONE**  
未指定任何旗標。  
**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的參數字集。

### **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</a> 的『追蹤路徑訊息 MQCFH (PCF 標頭)』。

### **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 *TraceRoute* PCF 群組部分追蹤路徑訊息資料的參數

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:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>

### **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: *TraceRoute.*

值: **MQROUTE\_DETAIL\_LOW**  
記錄使用者撰寫的應用程式所執行的活動。

**MQROUTE\_DETAIL\_MEDIUM**  
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。

**MQROUTE\_DETAIL\_HIGH**  
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: *TraceRoute.*

### **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: *TraceRoute.*

### **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: *TraceRoute.*

### **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: *TraceRoute.*

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:	<i>TraceRoute</i> 。
值:	<b>MQROUTE_ACCUMULATE_NONE</b> 活動資訊不會累積在追蹤路徑訊息的訊息資料中。 <b>MQROUTE_ACCUMULATE_IN_MSG</b> 活動資訊會累計在追蹤路徑訊息的訊息資料中。 <b>MQROUTE_ACCUMULATE_AND_REPLY</b> 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 <u>轉遞</u> 中說明的演算法。
ID:	MQIACF_ROUTE_FORWARDING。
資料類型:	MQCFIN。
包含在 PCF 群組中:	<i>TraceRoute</i> 。
值:	<b>MQROUTE_FORWARD_IF_SUPPORTED</b> 追蹤路徑訊息只會轉遞至佇列管理程式，以允許使用 <i>TraceRoute</i> 群組中 <i>Deliver</i> 參數的值。 <b>MQROUTE_FORWARD_ALL</b> 不論是否允許使用 <i>遞送</i> 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:	<i>TraceRoute</i> 。
值:	<b>MQROUTE_DELIVER_YES</b> 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。 <b>MQROUTE_DELIVER_NO</b> 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 WebSphere 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 90 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 WebSphere MQ 應用程式所執行 MQI 作業的相關資訊，或記錄 IBM WebSphere 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 WebSphere 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 WebSphere MQ 系統中發生之活動的相關資訊，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。

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。

## 帳戶訊息

結算訊息會記錄 WebSphere 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)。若為長時間執行的 WebSphere 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 135 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 145 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。

對於已收集結算資料的每個佇列，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 MQCONN 選項來控制連線層次的集合。

### MQI 帳戶資訊

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。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)
```

## MQCONNX 選項

在 MQCONNX 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)
```

## 帳戶訊息產生

當應用程式與佇列管理程式中斷連線時，會產生結算訊息。中間結算訊息也會針對長時間執行的 WebSphere MQ 應用程式撰寫。

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

中間結算訊息是針對長時間執行 WebSphere MQ 應用程式而撰寫的，自建立連線以來的間隔，或自前次寫入中間結算訊息以來的間隔超出配置的時間。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 WebSphere 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『MQI 統計資料資訊』](#)

使用佇列管理程式屬性 STATMQI 可控制 MQI 統計資料資訊的收集。

[第 124 頁的『佇列統計資料資訊』](#)

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『通道統計資料資訊』](#)

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。這些屬性適用於分散式平台及 IBM i。

## 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，範圍介於 1 到 100 *QStatisticsData* PCF 群組。

在間隔中，每個佇列都有一個作用中的 *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由許多 PCF 參數組成，範圍為 1 到 100 *ChlStatistics* 資料 PCF 群組。

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

### MQI 統計資料資訊

使用佇列管理程式屬性 STATMQI 可控制 MQI 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)
```

### 佇列統計資料資訊

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 WebSphere 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。

#### 關閉

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

#### 關閉

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。

#### 無

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)
```

### **distributed** 通道統計資料資訊

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。這些屬性適用於分散式平台及 IBM i。

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 WebSphere 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	用法
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	用法
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。

**中**

通道統計資料資訊是以中等詳細程度來收集。

**高**

通道統計資料資訊是以高階詳細資料來收集。

**關閉**

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。

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。

**中**

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。

**高**

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。

**關閉**

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。

**無**

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。

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。

**中**

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。

**高**

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。

**關閉**

統計資料資訊不適用於自動定義的叢集傳送端通道。

**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。

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)。預設統計資料時間隔為 1800 秒 (30 分鐘)。若要變更統計資料時間隔，請使用 MQSC 指令 ALTER QMGR 並指定 STATINT 參數。例如，若要將統計資料時間隔變更為 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時間隔到期之前，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時間隔。

## 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

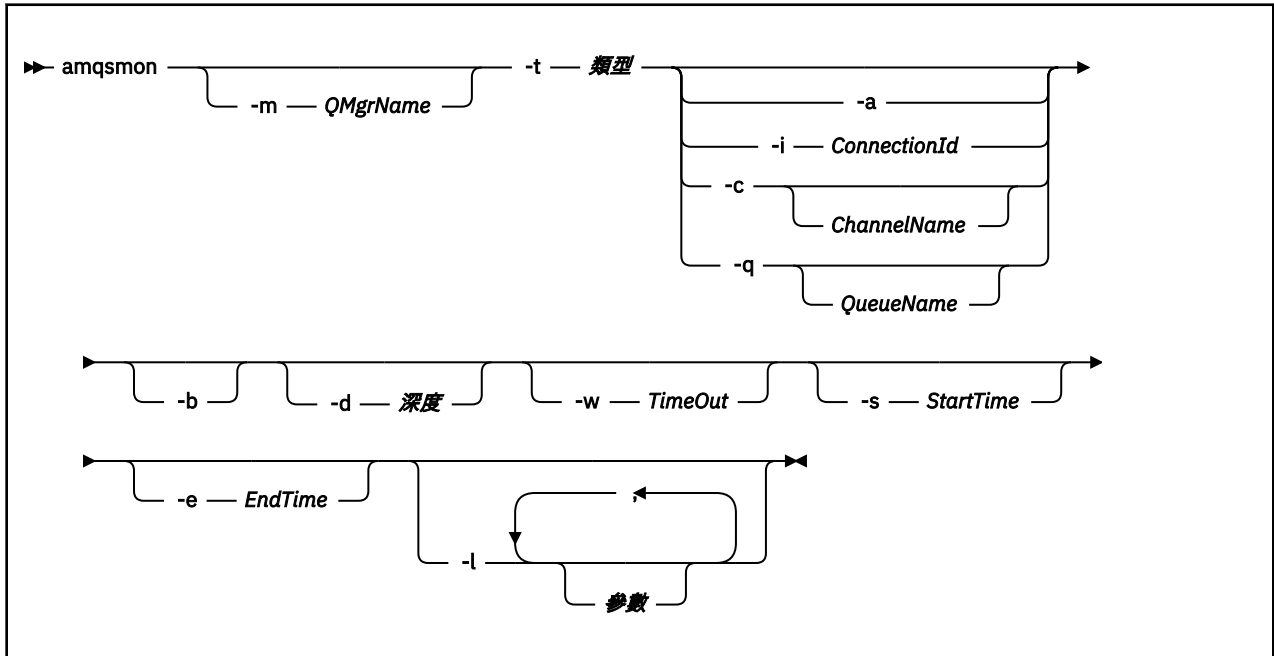
帳戶和統計資料訊息會寫入系統帳戶和統計資料佇列。**amqsmon** 是隨 WebSphere MQ 一起提供的範例程式，可處理結算及統計資料佇列中的訊息，並以可讀格式將資訊顯示至畫面。

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* 所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 Depth**

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。

如果您沒有指定這個參數, 一旦沒有其他訊息要處理, *amqsmn* 就會結束。

#### **-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 Parameter**

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 [amqsmn 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。

## amqsmn 範例

請利用這個頁面來檢視執行 *amqsmn* (顯示格式化監視資訊) 範例程式的範例

1. 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:

```
amqsmn -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, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [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]
CloseFailCount: [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]
InqFailCount: [0, 31, 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]
SetFailCount: [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. 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, 0]
OpenFailCount: [0, 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, 0]
CloseFailCount: [0, 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 -a -l 7006,3024
```

此指令的輸出如下:

```

ConnectionId: x'414d514354524556312020202020208d0b374203090020'
ApplicationName: 'runmqsc'

ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
ApplicationName: 'amqsput'

ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
ApplicationName: 'runmqsc'

ConnectionId: x'414d514354524556312020202020208d0b3742010d0020'
ApplicationName: 'amqsput'

ConnectionId: x'414d514354524556312020202020208d0b3742150d0020'
ApplicationName: 'amqsget'

5 Records Processed.

```

## 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 WebSphere MQ 訊息。訊息資料包含 WebSphere MQ 應用程式所執行 MQI 作業的相關資訊，或 WebSphere 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，範圍介於 1 到 100 *QStatisticsData* PCF 群組。

## 通道統計資料訊息

通道統計資料訊息資料由許多 PCF 參數組成，範圍為 1 到 100 *ChlStatistics* 資料 PCF 群組。

## 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 WebSphere 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: WebSphere 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 WebSphere 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

### ***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: <b>MQDISCONNECT_NORMAL</b> 由應用程式要求 <b>MQDISCONNECT_IMPLICIT</b> 應用程式異常終止 <b>MQDISCONNECT_Q_MGR</b> 佇列管理程式中斷連線
已傳回:	可用時

### **OpenCount**

說明:	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 <a href="#">參照附註 1</a> 。
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 [參照附註 2](#)。

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 [參照附註 2](#)。

ID: MQIAMO\_PUT1S  
資料類型: MQCFIL  
包含在 PCF 群組中: *QAccountingData*  
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 [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 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 <a href="#">參照附註 2</a> 。
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： 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\_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。

### **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 WebSphere 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:	WebSphere 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 WebSphere 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: 在此間隔中應用程式開啟此佇列的次數  
ID: MQIamo\_opens  
資料類型: MQCFIL  
包含在 PCF 群組中: *QAccountingData*  
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 <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>QAccountingData</i>



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

### **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。

### **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

## **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 WebSphere 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: 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: 順利關閉物件的嘗試次數。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\_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 <a href="#">參照附註 2</a>
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 <a href="#">參照附註 2</a>
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 <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。
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 <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。
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 <a href="#">參照附註 2</a>
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 WebSphere 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

- MQQDT\_PREDEFINED
- MQQDT\_PERMANENT\_DYNAMIC
- MQQDT\_TEMPORARY\_DYNAMIC

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: *QStatisticsData*  
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 WebSphere 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

### **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。

## **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 WebSphere 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。

### ***ChIStatisticsData***

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: 通道的名稱。

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。

### ***IncmplBatchCount***

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 135 頁的『MQI 結算訊息資料』](#)
- [第 145 頁的『佇列帳戶訊息資料』](#)
- [第 155 頁的『MQI 統計資料訊息資料』](#)
- [第 166 頁的『佇列統計資料訊息資料』](#)
- [第 172 頁的『通道統計資料訊息資料』](#)

1. 此參數與 WebSphere 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 WebSphere MQ 訊息相關。此參數是由下列常數檢索的值陣列 (MQCFIL 或 MQCFIL64):

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 WebSphere 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。

## 收集應用程式活動追蹤資訊

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 MQCONN 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 183 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 184 頁的『amqsact 範例程式』](#)。

## 程序

1. [第 178 頁的『設定 ACTVTRC 以控制活動追蹤資訊的收集』](#)。
2. [第 179 頁的『設定 MQCONN 選項以控制活動追蹤資訊的收集』](#)。
3. [第 179 頁的『使用 mqat.ini 來配置活動追蹤行為』](#)。
4. [第 183 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

已開啟 API 活動追蹤收集

### 關閉

已關閉 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 179 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 184 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 183 頁的『調整應用程式活動追蹤的效能影響』](#)。

## 設定 MQCONNX 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 MQCONNX 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 `mqt.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 `mqt.ini` 所定義。

2. 將 MQCONNX 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：

### **MQCNO\_ACTIVITY\_TRACE\_DISABLED**

已關閉連線的活動追蹤。

### **MQCNO\_活動\_追蹤\_已啟用**

已開啟連線的活動追蹤。

註：如果應用程式同時選取 **MQCNO\_活動\_追蹤\_已啟用** 及 **MQCNO\_ACTIVITY\_TRACE\_DISABLED** for MQCONNX，則呼叫會失敗，原因碼為 MQRC\_OPTIONS\_ERROR。

3. 請檢查活動追蹤配置檔 `mqt.ini` 中的設定是否未置換這些活動追蹤設定。

請參閱 [第 179 頁的『使用 mqt.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 184 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 183 頁的『調整應用程式活動追蹤的效能影響』](#)。

## 使用 mqt.ini 來配置活動追蹤行為

活動追蹤行為是使用稱為 `mqt.ini` 的配置檔來配置。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。

## 關於這項作業

  在 UNIX and Linux 系統上，`mqt.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\mqgrs\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 Version 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。

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 (**ApplName** 及 **DebugPath**) 區分大小寫。

### 段落索引鍵

配置檔中容許兩種段落索引鍵類型：「AllActivity 追蹤」段落及 ApplicationTrace 段落

#### AllActivity 追蹤段落

除非置換，否則「AllActivity 追蹤」段落會定義套用至所有 IBM WebSphere 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 段落中更具體的資訊置換。

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 追蹤中遺漏的參數採用預設值。會忽略來自先前 AllActivity 追蹤段落的參數及值

#### ApplicationTrace 段落

ApplicationTrace 段落定義可套用至 IBM WebSphere MQ 連線的特定名稱及/或類型的設定。

此段落包括 ApplName 及 ApplClass 值，根據「連線比對規則」中定義的比對規則來使用這些值，以判定段落是否適用於特定連線。

#### 參數/值配對

下表列出可在活動追蹤配置檔中使用的參數/值配對。

姓名	段落類型	值 (以粗體類型表示的預設值)	說明
追蹤	ApplicationTrace	ON/ <b>OFF</b>	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 26: 可在活動追蹤配置檔中使用的參數/值配對 (繼續)

姓名	段落類型	值 (以粗體類型表示的預設值)	說明
ActivityInterval	AllActivity 追蹤 ApplicationTrace	<b>0-99999999 (0=off)</b>	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，因此不會在經歷時間的確切時刻寫入追蹤訊息，而是在經歷時間間隔之後執行第一個 MQI 作業時寫入追蹤訊息。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	AllActivity 追蹤 ApplicationTrace	<b>0-99999999 (0=off)</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	AllActivity 追蹤 ApplicationTrace	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	AllActivity 追蹤 ApplicationTrace	<b>0-104 857 600 (100Mb)</b>	MQGET、MQPUT、MQPUT1 及回呼作業所追蹤的訊息資料量 (以位元組為單位)
ApplName	ApplicationTrace	字串 (必要參數-無預設值)	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。若為分散式平台，則只有 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。對於 z/OS 應用程式，這是整個 MQAXC.ApplName 符合段落中的值。ApplName 值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。

表 26: 可在活動追蹤配置檔中使用的參數/值配對 (繼續)

姓名	段落類型	值 (以粗體類型表示的預設值)	說明
ApplClass	ApplicationTrace	<b>使用者 /MCA/內部/全部</b>	應用程式的類別。如需 ApplType 值如何對應至 IBM WebSphere MQ 連線的說明, 請參閱下表

下表顯示 ApplClass 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 27: Applclass 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)
內部	MQXACT_EXTERNAL	MQXE_COMMAND_SERVER MQXE_MQSC	'runmqsc' 及指令伺服器
內部	MQXACT_INTERNAL	(任何值)	"trusted" 及內部應用程式與處理程序; 例如, amqzdmaa
ALL	(任何值)	(任何值)	追蹤所有使用者及內部連線



**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 wildcarded)
                                              # (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
                                              # Values: 0-99999999 (0=off)
                                              # Default: 0
ActivityCount=1                               # Number of operations between trace msgs
                                              # Values: 0-99999999 (0=off)
                                              # Default: 0
TraceLevel=MEDIUM                           # Amount of data traced for each operation
                                              # Values: LOW | MEDIUM | HIGH
                                              # Default: MEDIUM
TraceMessageData=1000                        # Amount of message data traced
                                              # Values: 0-100000000
                                              # Default: 0
    
```

### 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 AllActivity 追蹤段落中指定的值, 除非該值也出現在 ApplicationTrace 段落中, 且該段落符合點 2、3 及 4 中所說明連線的比對準則。
2. ApplClass 符合 IBM WebSphere MQ 連線的類型。如果 ApplClass 不符合連線類型, 則會忽略此連線的段落。
3. 段落中的 ApplName 值符合連線之 API 結束程式環境定義結構 (MQAXC) 中 ApplName 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 ApplName 包含萬用字元 (\*), 則只會將萬用字元左側的字元與連線 ApplName 中相等的字元數相互比較。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 ApplName 值不符合連線 ApplName，則此連線會忽略該段落。

4. 如果多個段落符合連線 ApplName 及 ApplClass，則會使用具有最特定 ApplName 的段落。最明確的 ApplName 定義為使用最多字元來符合連線 ApplName 的名稱。例如，如果 ini 檔案包含一個具有 ApplName= "FRE\*" 的段落，以及另一個具有 ApplName= "FREE\*" 的段落，則會選擇 ApplName= "FREE\*" 的段落作為與 ApplName= "path/FREEDOM" 連線的最佳相符項，因為它符合四個字元 (而 ApplName= "FRE\*" 只符合三個字元)。
5. 在點 2、3 和 4 中套用規則之後，如果有多個段落符合連線 ApplName 和 ApplClass，則會使用最後一個相符項中的值，並忽略所有其他段落。

## 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。此範例在 C samples 目錄 (與 amqsact.c file 相同的目錄) 中以稱為 mqat.ini 的範例提供。

```
AllActivityTrace:
  ActivityInterval=0          # Time interval between trace messages
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  ActivityCount=0           # Number of operations between trace msgs
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  TraceLevel=MEDIUM        # Amount of data traced for each operation
                              # Values: LOW | MEDIUM | HIGH
                              # Default: MEDIUM
  TraceMessageData=0       # Amount of message data traced
                              # Values: 0-100000000
                              # Default: 0

ApplicationTrace:
  ApplClass=USER            # Application type
                              # Values: (USER | MCA | INTERNAL | ALL)
                              # Default: USER
  ApplName=AppName*        # Application name (may be wildcarded)
                              # (matched to app name without path)
                              # Default: *
  Trace=OFF                # Activity trace switch for application
                              # Values: ( ON | OFF )
                              # Default: OFF
  ActivityInterval=0       # Time interval between trace messages
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  ActivityCount=0         # Number of operations between trace msgs
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  TraceLevel=MEDIUM      # Amount of data traced for each operation
                              # Values: LOW | MEDIUM | HIGH
                              # Default: MEDIUM
  TraceMessageData=0     # Amount of message data traced
                              # Values: 0-100000000
                              # Default: 0
```

## 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 183 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 179 頁的『使用 mqat.ini 來配置活動追蹤行為』和第 179 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 179 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

### amqsact 範例程式

**amqsact** 會為您格式化「應用程式活動追蹤」訊息，並隨 WebSphere MQ 提供。

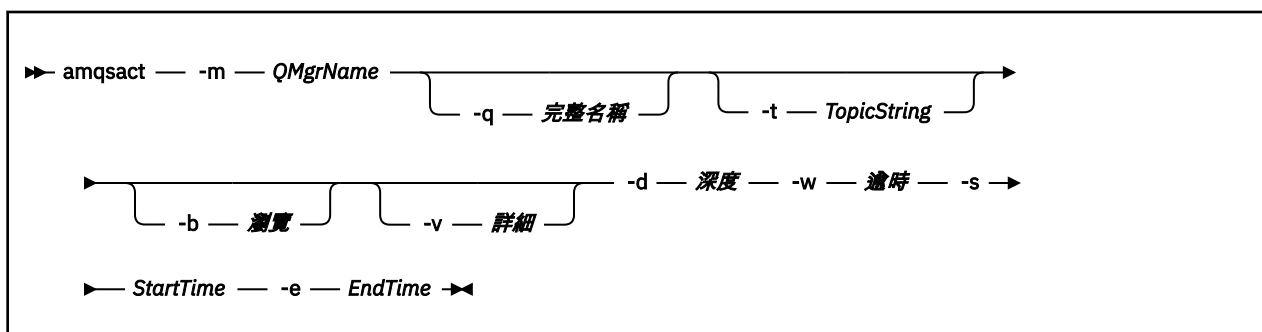
已編譯的程式位於 samples 目錄中：

- 在 UNIX and Linux 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 Depth**  
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 QName**

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: '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
```

**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 WebSphere MQ 訊息。 訊息資料包含 IBM WebSphere MQ 應用程式所執行 MQI 作業的相關資訊，或 IBM WebSphere 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****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 WebSphere 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\_XX\_ENCODE\_CASE\_ONE other  
MQXE\_MCA MQXE\_MCA\_SVRCONN MQXE\_COMMAND\_SERVER MQXE\_MQSC  
ID: MQIACF\_API\_環境  
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

### ***Pointer Size***

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。 值是其中一個 MQPL\_\* 值。  
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 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。

## **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

## **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)。

### **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)

### **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)

## **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

#### **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

## **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 MQCONNX  
PCF 參數: MQIACF\_CONNECT\_OPTIONS  
追蹤層次: 2  
類型: MQCFIN

### **ConnectionOptionsStructure**

說明: MQCNO 結構。  
註: 僅限 MQCONNX)  
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)

### **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

### **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 APPTTRACE 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: MQIAMO64\_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)

### **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

### **IntAttrs**

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

### **CharAttrs**

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

## **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: 用來開放物件的選項  
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>12</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>12</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>12</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>12</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 MQOD.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

#### 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。Ap-pActivityDistList PCF 群組會

- 
- 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。

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：

表 28: AppActivityDistList 群組 MQCFGR 結構		
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

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)

#### ***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 AppActivityDistList PCF 群組。對於配送清單中的每一個佇列，請參閱第 214 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 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，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 及 Reason):

---

<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。

### **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

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)

### ***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 214 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

<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。

### **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

### **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

## **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)。

## **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

## **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)

## **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

### **XACOMLETE**

應用程式已啟動 XACOMLETE 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: 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

### **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:	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

### **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

## **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 WebSphere 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 WebSphere 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 248 頁的表 29 彙總可用於即時監視通道的監視層次：

水準	說明	用法
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 247 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。

[使用佇列管理程式](#)

## 相關工作

[第 249 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 WebSphere MQ 探險家」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。

[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 WebSphere MQ 探險家」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 WebSphere MQ 探險家或 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 WebSphere MQ 探險家或 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 247 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. Then issue the following command, inserting in <appltag> the APPLTAG value from the output of the previous command:

```
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。  
另請確定通道已指定重試屬性的值: *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)
```

6. 選擇性的: 如果 STATUS 欄位的值是 RETRY, 請執行下列步驟:
- 請檢查錯誤日誌以識別錯誤, 然後更正問題。
  - 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:
- 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:

- 在某些情況下, 可能只有通道一端有子狀態。
  - 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 254 頁的表 30](#) 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 256 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 256 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 256 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	SSL 信號交換是由許多傳送和接收所組成。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。

附註:

- 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。
- 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 251 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：

#### MSGGS

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。

#### CURMSGGS

如果是傳送端通道，則這是已在現行批次中傳送的訊息數。如果是接收端通道，則這是已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。

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. 如果通道已啟動尚未完成的批次 (如 CURMSGGS 中的非零值所指示)，通道可能正在等待通道另一端確認批次。查看輸出中的 SUBSTATE 欄位，請參閱 [第 255 頁的表 31](#)：

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。
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

請利用這個頁面來檢視批次可能需要很長時間才能完成的部分原因。

## 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。下列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。  
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。
- 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 251 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

1. 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：
  - a) 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
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 COMPMSG 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 253 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。

### 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 WebSphere MQ 可讓您使用 `runswch1` 指令來切換一個以上未執行的通道的傳輸佇列。

### 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
- .
- 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，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。

## Windows 效能監視器

在 WebSphere MQ 7.0 版及更早版本中，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 WebSphere MQ 7.1 版開始，此效能監視方法不再可用。

您可以使用 [第 247 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。

## 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。本文件的使用者應驗證其特定環境適用的資料。

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 IBM 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: