

## IBM Rational Tester for SOA Quality 的測試執行與效能報告 追蹤服務效能輕而易舉

級別：中級

顧問 [Michael Kelly \(Mike@MichaelDKelly.com\)](mailto:Mike@MichaelDKelly.com)，[www.MichaelDKelly.com](http://www.MichaelDKelly.com)

2007 年 3 月 27 日

服務導向架構是許多現代運算基礎架構的根本，如果您的商店正是以服務為導向，便需要一套簡易一致的方法，針對您所部署的 Web 服務追蹤品質。請進一步瞭解 Rational Tester for SOA Quality 如何協助您分析服務的狀況，幫助您突破營運瓶頸。

IBM® Rational® Tester for SOA Quality 可針對服務導向架構 (SOA) 應用程式，自動建立、執行與分析功能及回歸測試。您可透過本文，瞭解 IBM Rational Tester for SOA Quality 的報告內容，尤其如何使用 Test Log 檢視器，以及 Web 服務效能報告適用的各種選項細節。

The Rational Tester for SOA Quality 產品是 Rational Performance Tester 應用程式的延伸。如果您對 Rational Tester for SOA Quality 或 Rational Performance Tester 不熟悉，建議您花點時間閱讀下方[資源](#)部分的相關介紹。

### 測試設定

本文在撰寫時使用的軟體包括 IBM Rational Performance Tester 7.0.0 版、IBM Rational Tester for SOA Quality 7.0.0 版公開測試版、Microsoft Windows® 2000 Professional SP2，以及現行（截至本文初版發佈日期為止）的 Google Web API。

本文將使用適用於 Google Web API 的 Web 服務測試，您可以在下方[資源](#)部分找到這項服務的 WSDL。在本文的測試套組中，我們提供三項測試，API 中每項作業都有適用的測試；測試的內容顯示如下方圖 1。

**圖 1：GoogleAPI 測試套組的測試內容**

## Test Contents

This section shows the test contents

The screenshot shows a tree view of test contents under the 'GoogleAPI' folder. The tree structure is as follows:

- GoogleAPI
  - doGetCachedPage( "ijdWdfBQFHJS...", "http://www-1..." )
    - doGetCachedPage( "PG1ldGEgaHR0..." )
      - Equal Verification Point
  - doGoogleSearch( "ijdWdfBQFHJS...", "developerWorks", "0", "10", "true", ... )
    - doGoogleSearch( " ", "true", "10", "false", "113000", ... )
      - Contain verification point
  - doSpellingSuggestion( "ijdWdfBQFHJS...", "IBM Rationla..." )
    - doSpellingSuggestion( "IBM Rational..." )
      - Contain verification point

On the right side of the tree view, there is a vertical toolbar with the following buttons: Add, Insert, Remove, Up, Down, Prev, and Next.

### 測試一：doGetCachedPage()

第一項測試是呼叫 doGetCachedPage() 作業，會傳入 API Key 以及 IBM developerWorks 的 URL (<http://www-128.ibm.com/developerworks/>)。在這個測試案例中，我們有一個相等驗證點 (equal verification point)，會尋找完全相符的回應訊息。我將這個測試案例設定為驗證點失敗 (Fail)。

### 測試二：doGoogleSearch()

第二項測試是呼叫 doGoogleSearch() 作業，會傳入 API Key、"developerWorks" 的 q 值、開始值 0、maxResults 值 10，以及其他所有項目的預設值。在這個測試案例中，我們有一個包含驗證點 (contain verification point)，只會尋找 developerWorks 中的以下片段："An online collection of tutorials, sample code, standards, and other resources <br> provided experts at IBM to assist software developers using open standards <b>...</b>"。我將這個測試案例設定為驗證點通過 (Pass)。

### 測試三：doSpellingSuggestion()

第三項測試是呼叫 doSpellingSuggestion() 作業，會傳入 API Key 與 "IBM Rationla Performance Tester" 詞組。在這個測試案例中，我再次使用包含驗證點 (contain verification point)，會尋找所產生的正確拼字。我將這個測試案例設定為驗證點通過 (Pass)。

設定以上三項測試後，便可探討不同類型的測試結果。

## Test Log 檢視器

要瞭解測試的相關細節，Test Log 檢視器可謂最佳來源。就 Web 服務功能測試而言，Test Log 檢視器是查看結果的第一首選；也可以利用 Test Log 檢視器，查看 Web 服務效能測試的相關細節。這個檢視器包含 Overview（概觀）與 Events（事件）兩個視圖。在 Overview 視圖中，您可以查看一般資訊與共用內容。一般資訊包括測試套組的名稱、說明，以及相關路徑；共用內容則包含開始與停止的時間、測試執行裁決 (verdict)，以及測試套組的類型等等。此外，這個視圖還包含其中指出整體測試執行結果的裁決。表 1 說明您可能會看到的裁決類型。

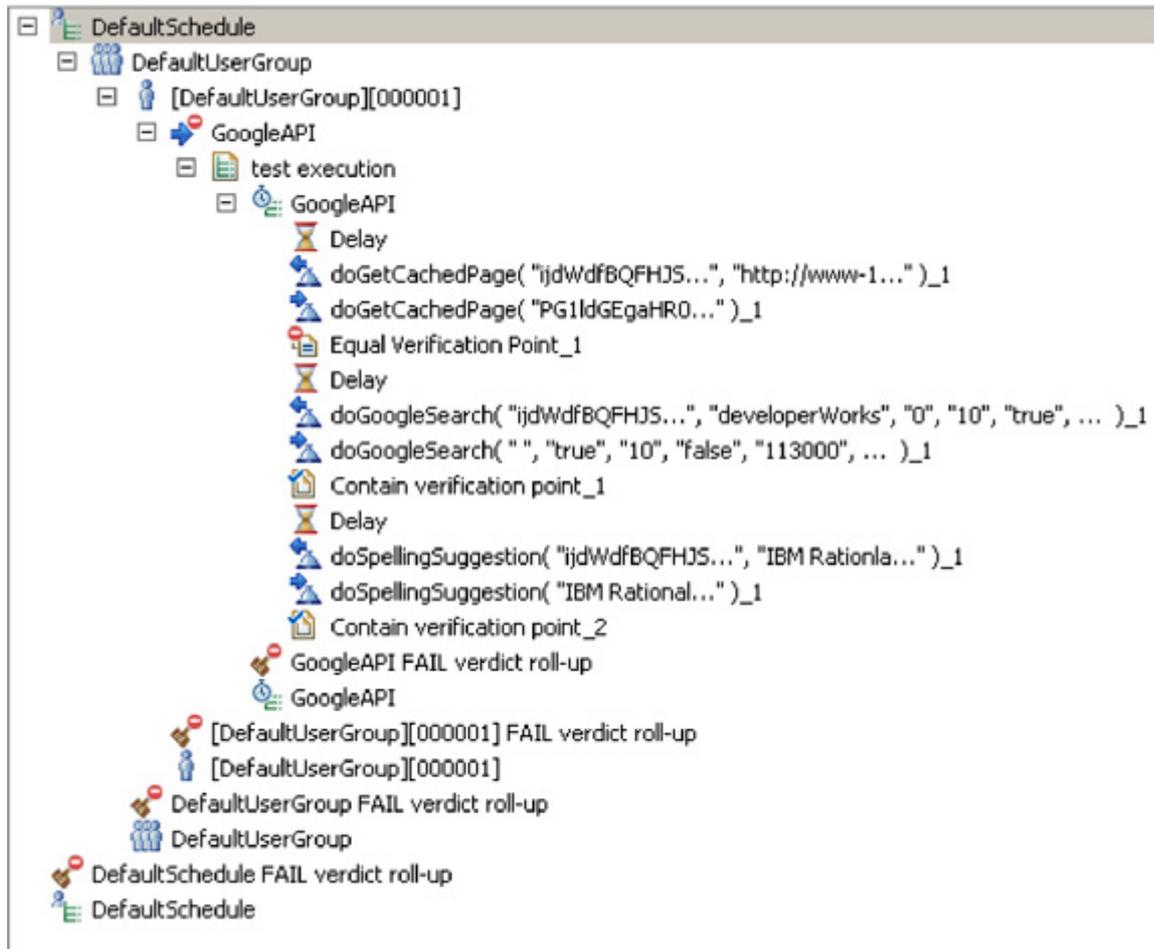
表 1 可能的裁決與裁決圖示

裁決	圖示	說明
Error（錯誤）		表示未能成功傳送主要要求給伺服器、並未從伺服器收到任何回應，或者回應不完整或無法剖析。
Fail（失敗）		表示驗證點與預期回應不相符，或沒有收到預期回應。
Inconclusive（沒有結論）		表示您所提供的自訂程式碼將裁決定義為沒有結論。
Pass（通過）		表示驗證點相符，或已收到預期回應。

在 Events 視圖中，您可以查看測試執行的相關細節。Events 樹狀結構會列出所有測試執行事件，如 start and end、loop、invocation、message 或 verdict 等 script。在 Events 樹狀結構中選取物件後，選定的事件或物件會顯示於 Common Properties 與 Detailed Properties。Common Properties 會針對 Events 樹狀結構中的選定事件，顯示其 Time 與 Text。按一下 Properties 下的元素名稱，即可開啓 Events 樹狀結構中選定事件的測試套組、測試案例或測試行爲。Text 欄位則會針對您在 Events 樹狀結構中選取的事件，顯示事件執行的相關訊息。Defects 區段則已與 ClearQuest 整合，您可記載或檢視與選定元素相關聯的問題。

現在，讓我們進一步探討 Google Web API 的測試套組。圖 2 說明自動產生時程的 Events 範例。

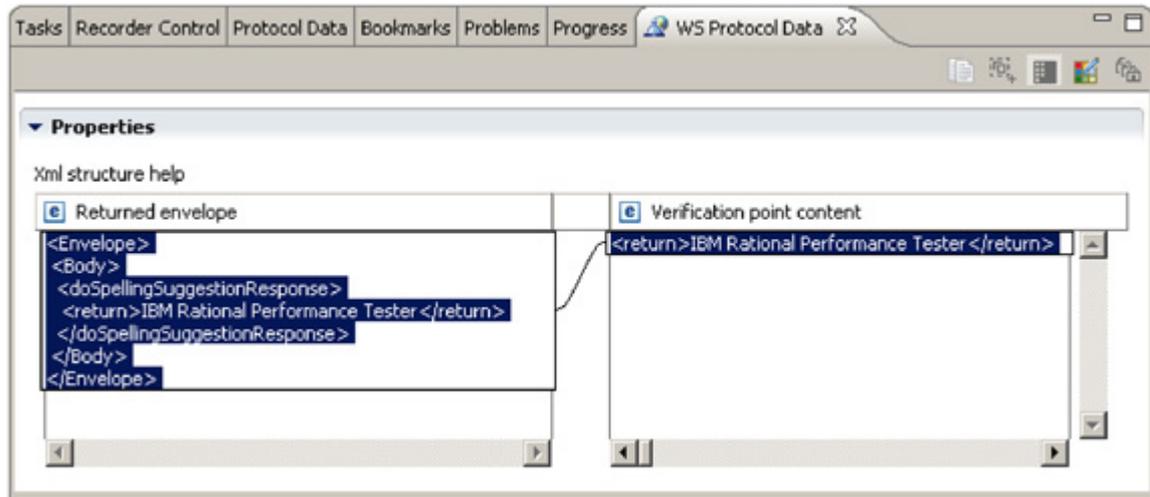
圖 2：Google Web API 測試套組的 Events 樹狀結構



在樹狀結構中移動（從頂端開始向下移動），您將會看到，樹狀結構各個層級都有裁決彙總；Google API 套組元素顯示彙總的裁決，而每個驗證點則顯示本身的裁決。每個 Web 服務呼叫之間的延遲是自動新增的；其預設值為零毫秒。最後，每個呼叫在列出時都會包含其對應要求與驗證點。

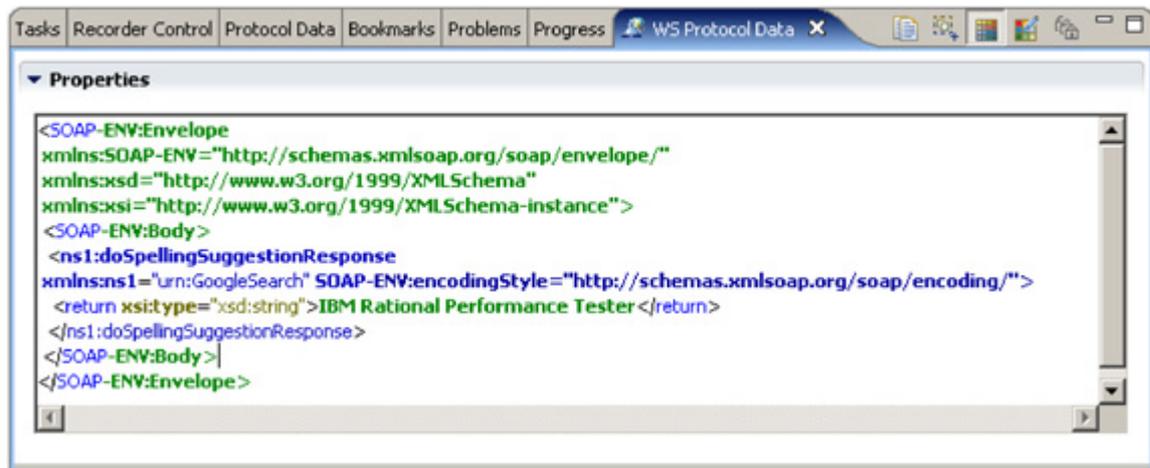
如需驗證點的詳細資料，您可以使用 Web Service Protocol 視圖。您可以利用這個視圖，查看所傳回訊息封以及驗證點的詳細資料。基於某些原因，這個視圖預設為不顯示，因此您需要先選取 **Window > Show View > Other**，然後在 Show View 視窗中選取 **Test > WS Protocol Data**，才能開啓視圖。圖 3 說明我們用於 doSpellingSuggestion() 的驗證點。

圖 3：Web Service Protocol 視圖



在 Web Service Protocol 視圖中，您還可以看到用來要求或回應 Web 服務呼叫的 XML。圖 4 說明用於 `doSpellingSuggestion()` 回應的 XML。

圖 4：doSpellingSuggestion() 的回應 XML



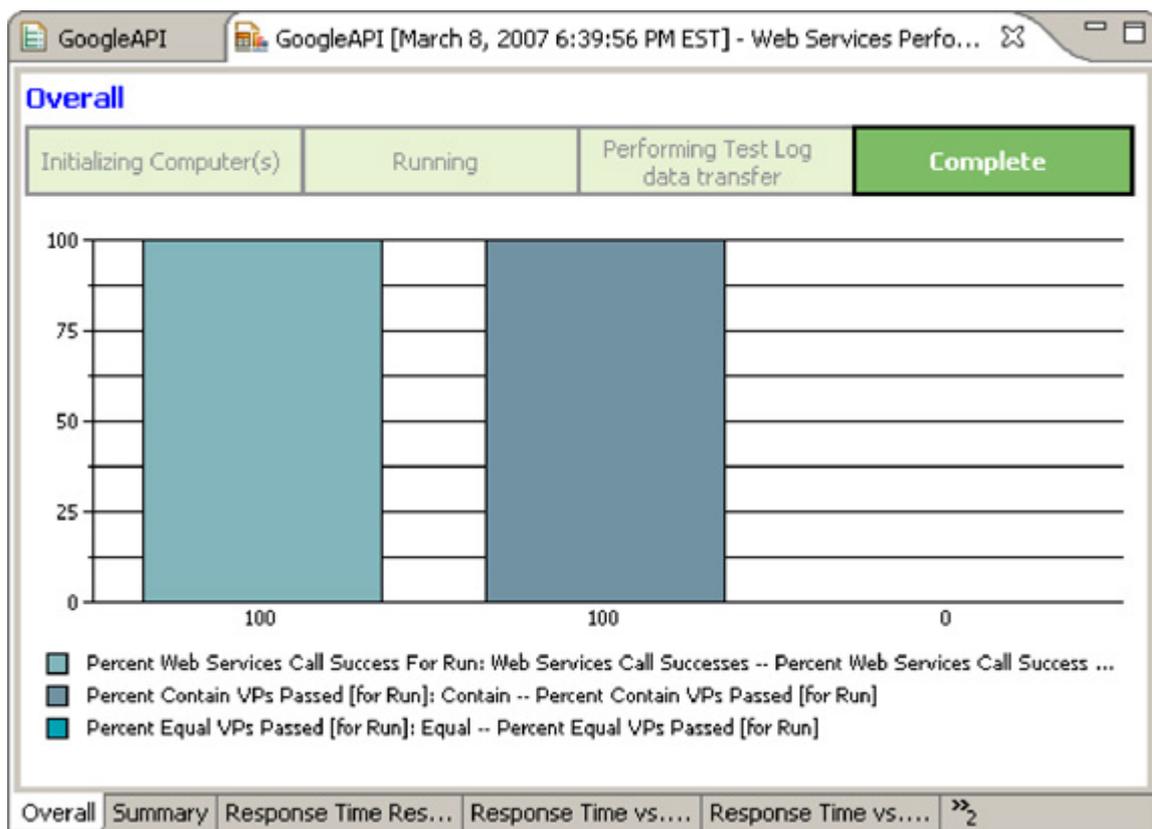
## Web 服務效能報告

無論您需要的是 Web 服務功能測試摘要資訊，還是 Web 服務效能測試詳細資訊，Web 服務效能報告都可滿足需求。Web 服務效能報告最大的優點，是您可以自訂摘要資訊，測試結束時隨即顯示。

## 自訂 Web 服務的報告

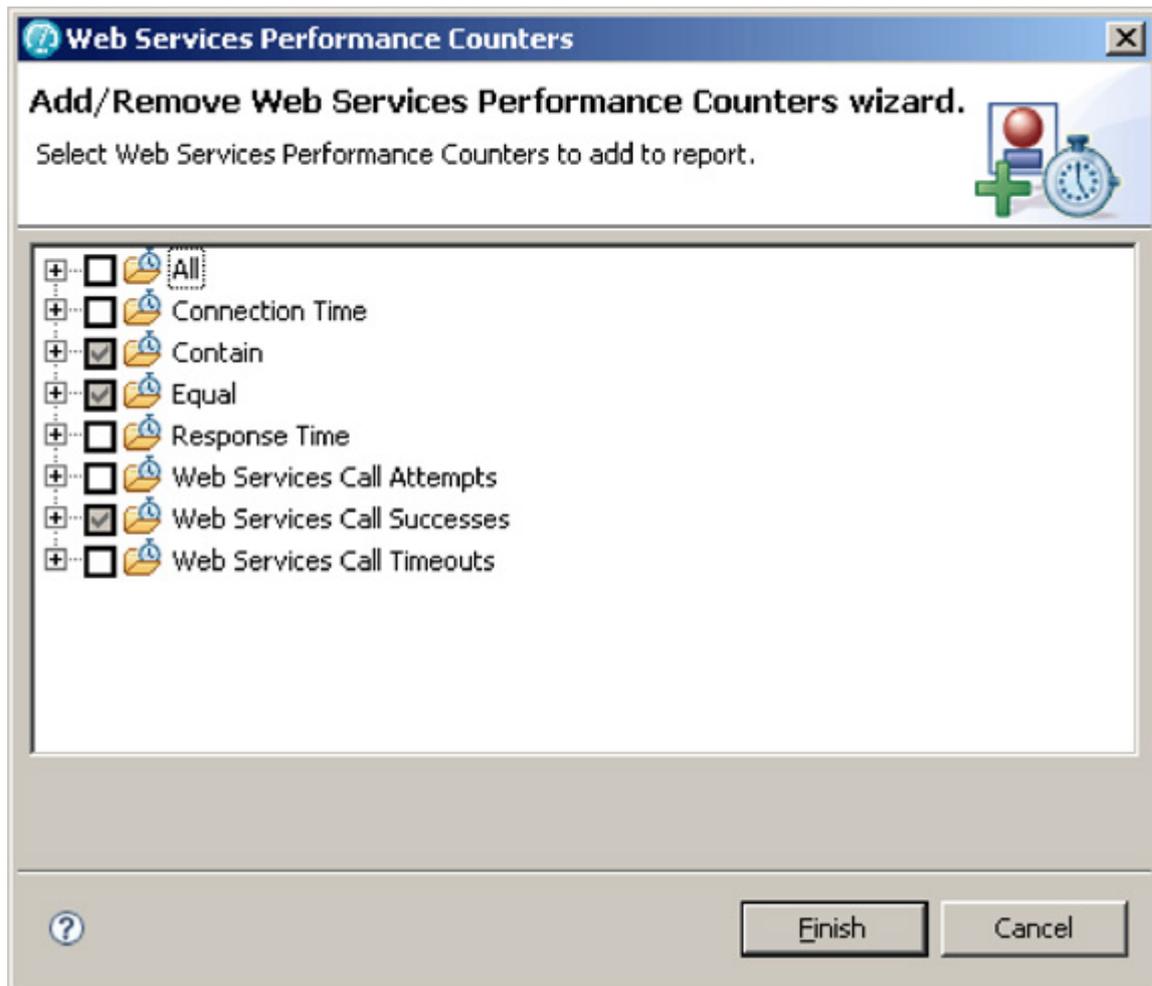
執行測試時，您所看到的第一份報告就是 Overall 報告；圖 5 為範例說明。依照預設，這份報告會顯示 Web 服務呼叫的成功率，以及您在套組中設定的每種驗證點之過關率。

圖 5：Web 服務的整體效能報告



有了整體效能報告，您就可以放手進行各項工作。要查看可用選項，您可在任意處用滑鼠右鍵按一下報告，然後選取 **Add/Remove Performance Counters > Web Services Performance Counter...**，即可開啓 Add/Remove Web Services Performance Counter 精靈，見圖 6。

圖 6：Add/Remove Web Services Performance Counter 精靈

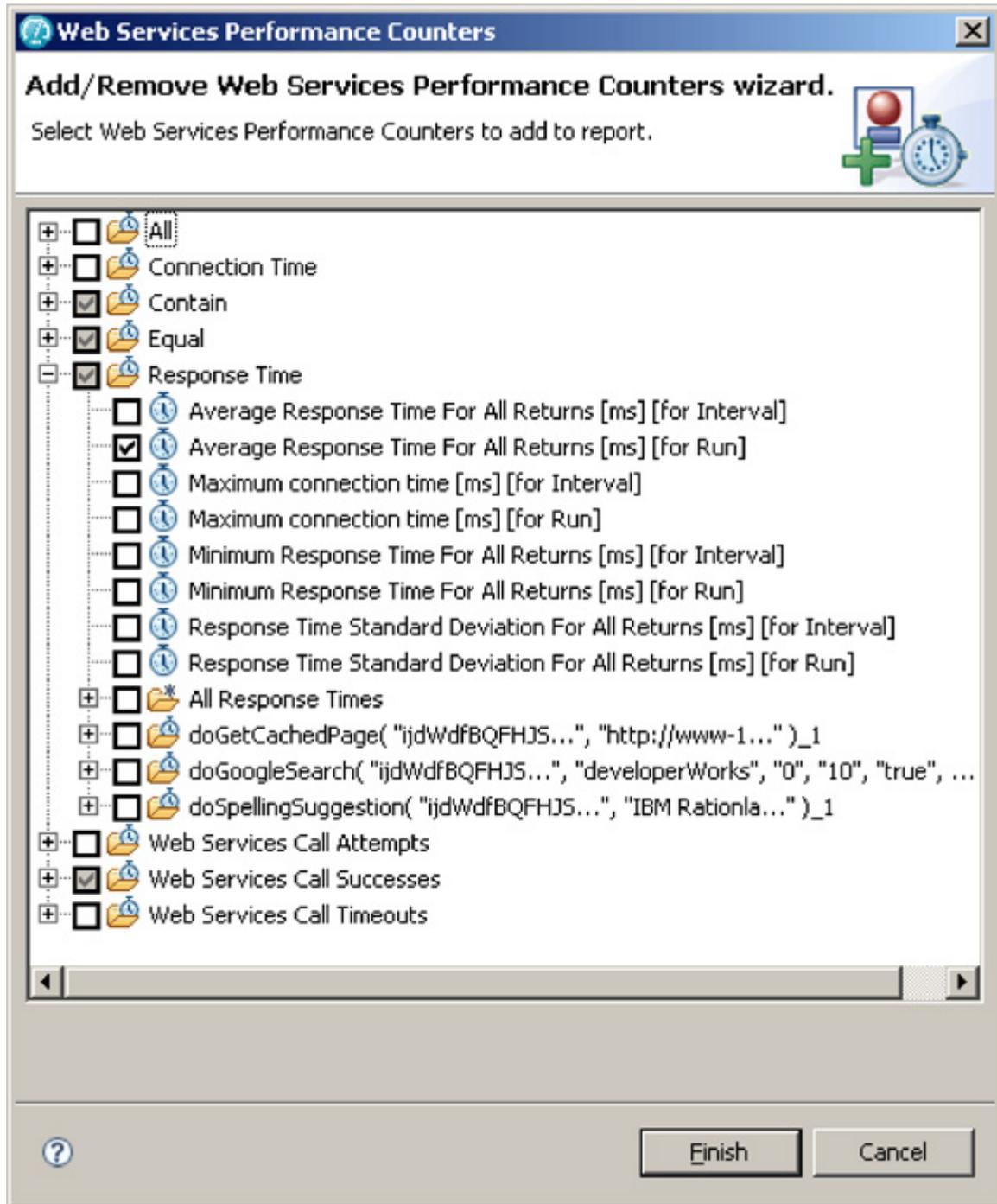


展開任何計數器類型，您將會看到各種不同的計數器，例如：

- 成功百分比
- 回應時間或連線時間的平均值、最小值、最大值與標準差
- 驗證點總數
- 包含驗證點的總數與百分比

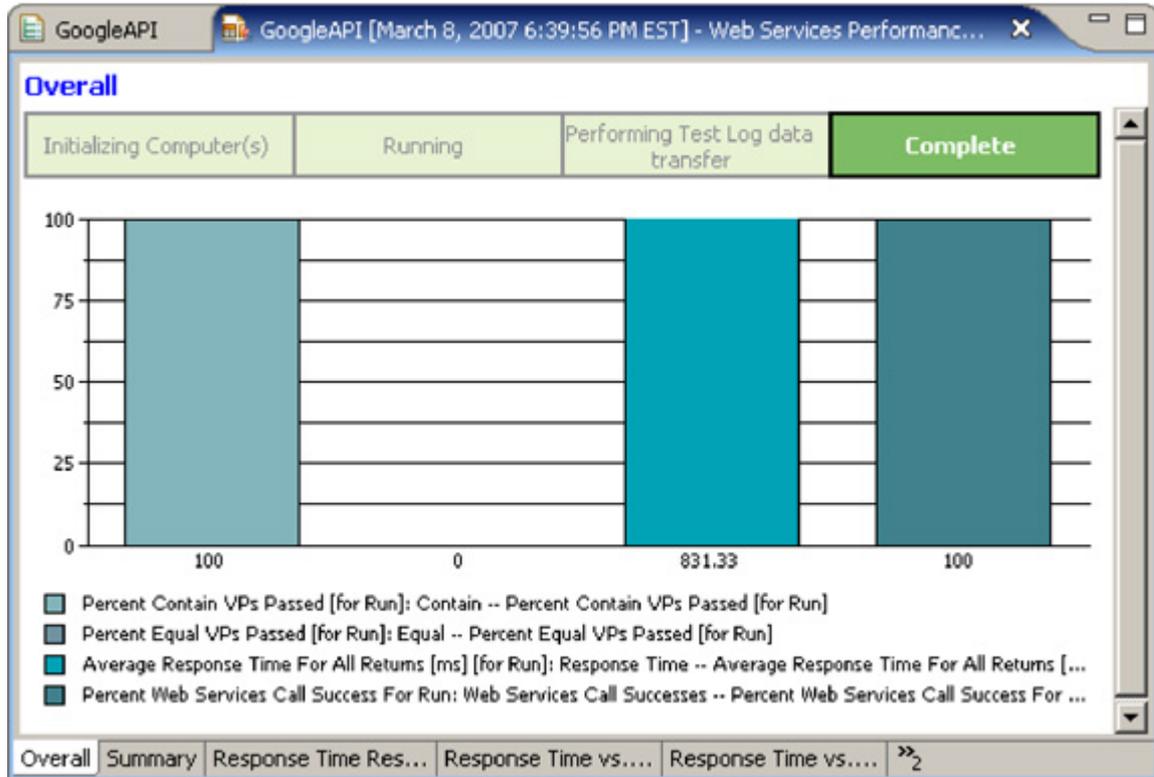
請參閱以下特別範例。展開 **Response Time**(回應時間)，您可以為所有的 **Web 服務** 回傳，新增平均回應時間之計數器，如圖 7 所示。

**圖 7：為所有回傳新增平均回應時間之計數器**



新增此計數器後，Overall 報告將隨之變更，如您在圖 8 所見。

圖 8：為所有回傳新增平均回應時間之計數器的 Overall 報告



在此範例中，您可以看到 Web 服務回傳的平均回應時間為 831.33 毫秒。為了確保看到正確的計數器／值，您可以切換至 Response Time vs. Time Detail 報告（在 Web Service Performance Report 視圖底端的另一個標籤中），並且查看每個 Web 服務呼叫的回應時間，如圖 9 所示。

**圖 9：Response Time vs. Time Detail 報告的 Performance Summary（效能摘要表）**

	Response Time -- Minimum [for Run]	Response Time -- Average [for Run]	Response Time -- Maximum [for Run]	Response Time -- Standard Deviation [for Run]
doGetCachedPage()_1	1,573	1,573	1,573	0
doSpellingSuggestion()_1	420	420	420	0
doGoogleSearch()_1	501	501	501	0

如果您為這三個數字計算平均值，將會得出 831.33。我第一次為報告新增某個計數器時，通常會按兩下計數器；主要是因為我不相信自己，而不是不信任這個工具。這個檢查點可讓我確定所做和所想的一致，我不希望在效能報告中出錯。基於某些原因，管理階層總是希望第一次出擊就正確無誤。

您可以使用剛剛為所有回傳新增平均回應時間計數器的方法，為所有報告新增或移除計數器。開啓 **Add/Remove Web Services Performance Counter** 精靈，然後查看您為每種報告設定的選項。您可以嘗試新增各種不同的計數器，看看會發生何種結果。試過各種查看資料的方式後，您就會找到最適合自己的方法。

如果您想關閉正在使用的報告，系統會詢問是否要儲存變更。如果希望日後的 **Web Services Performance Report** 納入您所做的變更，便可逕行儲存。然而，如果不確定是否每次都需要該項資訊，請勿執行儲存，而等到有需要時，再新增資訊（或建立新的報告類型）。

## Web 服務驗證點報告

**Web Service Verification Point** 報告內含測試中所有驗證點的詳細資料。這些報告雖無法提供許多新資料，卻是很好的驗證點資訊摘要。此外，如果您的套組中含有大量驗證點，這些報告便可提昇尋找結果的效率。

若要顯示 **Web 服務驗證點報告**，請用滑鼠右鍵按一下日誌，然後選取 **Web Services Reports > Web Services Verification Point Report**。報告隨即開啓，並且以 **Summary** 視圖做為開始畫面，如圖 10 所示。

圖 10：Web 服務驗證點報告中的 Summary (摘要)

Verification Point Summary Tab	
Percent All VPs Passed [for Run]	66.667
Total All VPs Attempted [for Run]	3
Total All VPs Failed [for Run]	1
Total All VPs Error [for Run]	0
Total All VPs Passed [for Run]	2

這份報告中的其他視圖，會顯示套組中各種驗證點的詳細資料。舉例而言，您可以透過圖 11 中的 Return Contain Verification Points 視圖，查看套組中兩個包含驗證點的結果。

圖 11：Return Contain Verification Points 視圖

	Pass -- Count [for Run]	Fail -- Count [for Run]	Error -- Count [for Run]	Inconclusive -- Count [for Run]	Contain -- Percent Pass
Contain verification point_1	1	0	0	0	100
Contain verification point_2	1	0	0	0	100

您可以使用先前在 Overall Web Service Performance 報告中用過的方法，為這些報告新增或移除計數器。

### 要訣和技巧

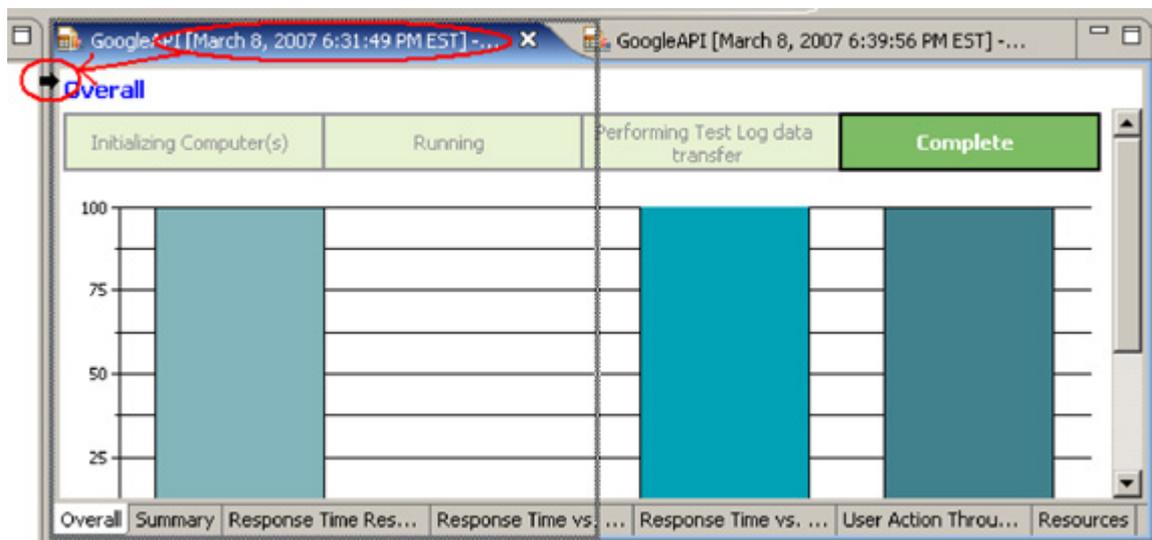
您花費在分析結果的時間越多，越希望瞭解如何以更快的速度，尋找所需要的資

訊。本節所提供的要訣和技巧，能協助您以更快的速度前往目的地。

### 同時檢視多份報告

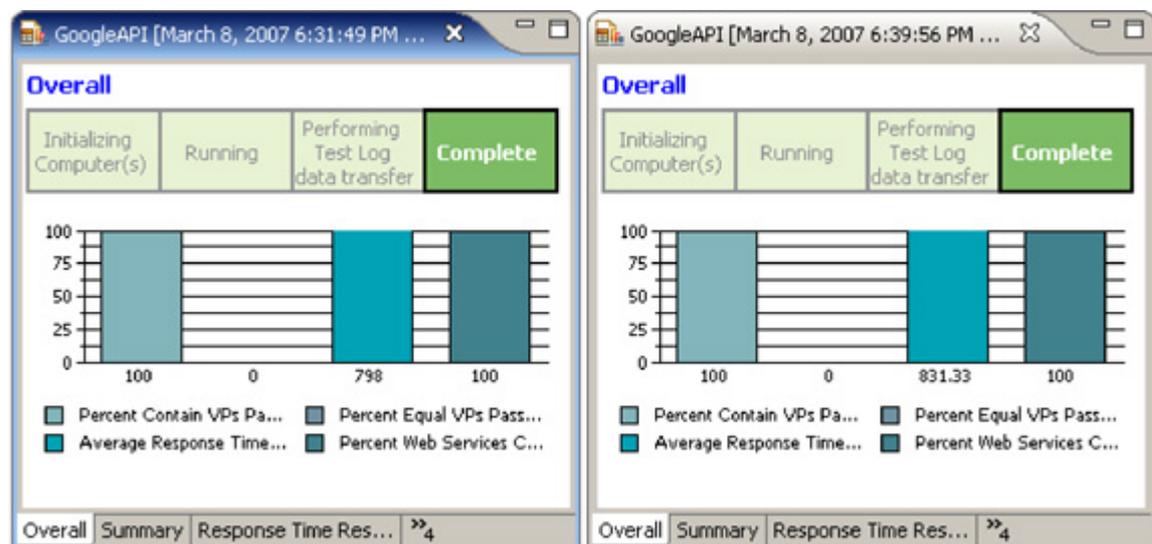
您可能常希望可以同時檢視兩份報告，或許是爲了進行比較，也可能只爲了獲得更多資訊。若要同時檢視兩份報告（事實上是多份報告；您想檢視多少報告都可以），只要簡單按一下報告的標題，將游標拖曳至檢視器左邊後定置，如圖 12 所示。

圖 12：拖曳報告



游標會變成黑色箭頭，這兩份執行中的報告會並列顯示，如圖 13 所示。

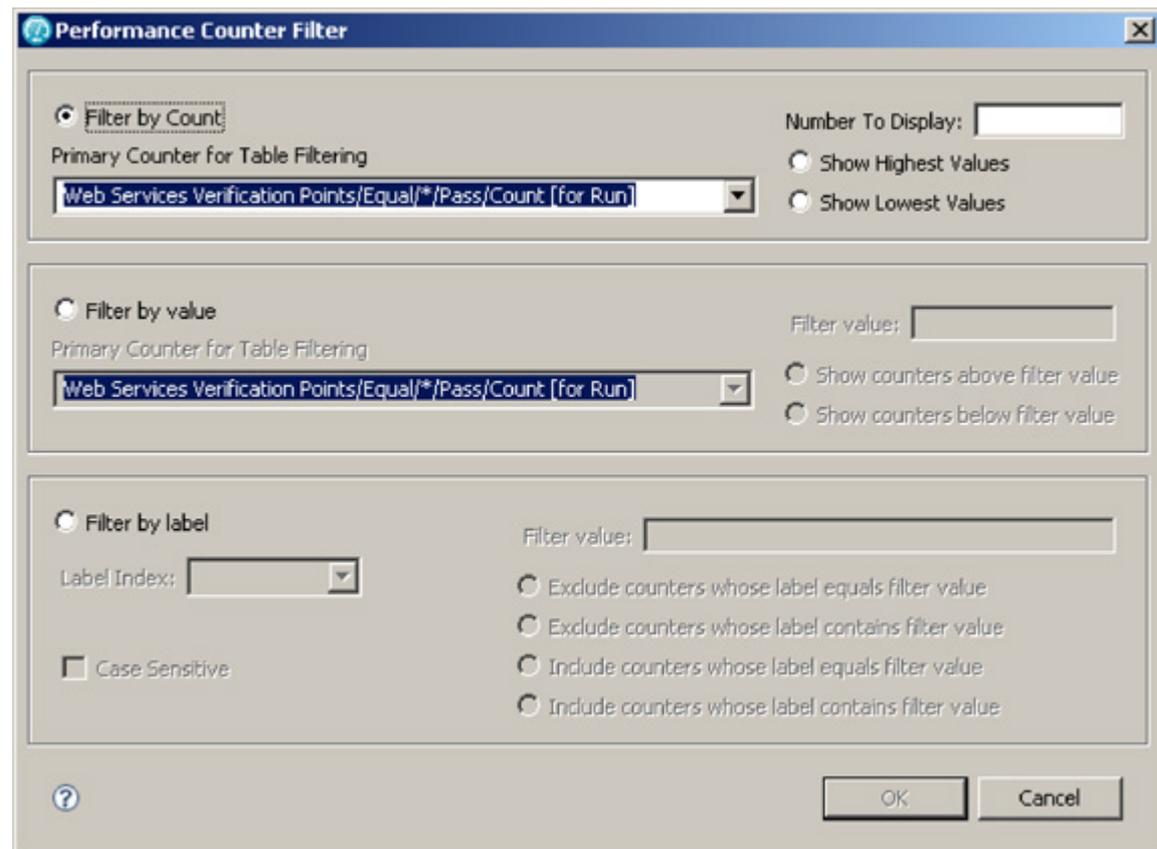
圖 13：兩份報告並列顯示



## 過濾結果

在顯示的報告上執行過濾，可移除不必要的資料，而專注於有意義的資料。就我目前所知，幾乎所有報告都可以進行過濾。只要用滑鼠右鍵按一下報告（如同新增計數器），然後選取 **Apply Filter, Performance Counter Filter** 對話框隨即開啓，如圖 14 所示。

圖 14：Performance Counter Filter 對話框



以下三種可用選項的簡短摘要：

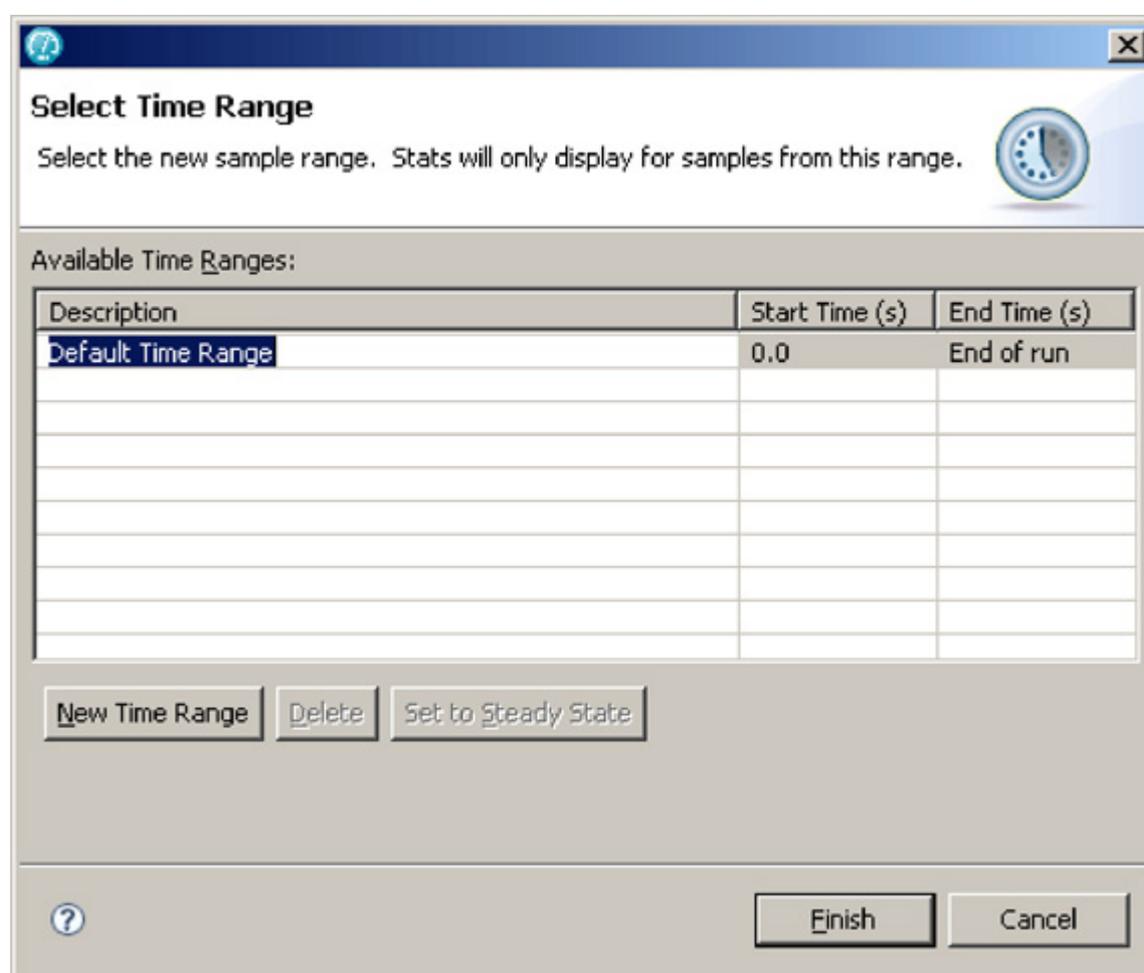
- **Filter by count (依計數過濾)**：顯示指定數目的項目。舉例而言，如果您選取這個選項並輸入 15，報告將會顯示 15 個含最高值或最低值（視您選取的圓鈕而定）的項目。
- **Filter by value (依值過濾)**：根據與指定值的比較而顯示項目。舉例而言，如果您選取這個選項並輸入 15，報告將會顯示所有大於或小於 15（視您選取的圓鈕而定）的項目。
- **Filter by label (依標籤過濾)**：顯示符合特定標籤的項目。如果您對表格進行過濾，標籤通常列於左欄的頁面；如果您對圖形進行過濾，標籤會是圖形中的圖例。

## 針對指定的時間範圍評估結果

另一個選項類似過濾功能，可縮短測試的時間範圍。若要針對特定開始及停止時間重新計算結果，您可以指定報告的時間範圍。您可以輸入自訂的開始及停止時間，以便針對延長或縮短時間的測試，執行過濾資料。集中於特定時間範圍可提供多項功能，例如可以只查看最多虛擬使用者呼叫 Web 服務期間的測試結果。或許這段期間驗證點的通過／失敗率，正是您最關切的。系統重新計算彙總結果時，只會考量在指定時間範圍內所收集的資料。

要變更效能報告的時間範圍，請用滑鼠右鍵按一下，然後選取 **Change Time Range...**，即可開啓 **Select Time Range**（選取時間範圍）對話框，如圖 15 所示。

圖 15：Select Time Range（選取時間範圍）對話框



按一下 **New Time Range**，為 **Available Time Ranges**（可用時間範圍）清單新增時間範圍。按一下 **Finish** 後，報告就會更新；只要縮放時間軸，即可顯示指定時間範圍內的資料。重新計算彙總結果時，只會反映選定時間範圍內的資料。請注意，新指定的時間範圍，會隨報告儲存。要返回完整報告，請用滑鼠右鍵按一

下報告並選取 **Change Time Range...**，再從 **Available Time Ranges** 清單選取 **Overall Time Range**。

### 將結果匯出成 CSV、XML 或 HTML 檔

您可以將測試結果匯出成 CSV、XML 或 HTML 檔，這項功能在許多情況下相當有用，最常見的情況是，您希望透過匯出功能彙總測試資料與其他工具所收集的資料，又或者單純爲了歸檔與報告之用。

您可以整個測試執行結果或特定部分匯出成 CSV 檔案，以供進一步分析。若要匯出測試執行：

1. 選擇 **File > Export**。
2. 在 **Export** 視窗中按一下 **Performance test run statistics**，然後按一下 **Next**。
3. 輸入 CSV 檔的名稱（含 .CSV 副檔名），然後按一下 **Next**。
4. 選取要匯出的測試執行，再按一下 **Next**。測試執行會依時間先後順序列出，其中最近一次的測試執行，會列在清單的最底端。
5. 此時如有需要，您可以選取希望匯出的資訊類型，完成動作後按一下 **Finish**。

您可以匯出 XML 格式的測試日誌，以供進一步分析之用。若要匯出 XML 格式的測試日誌：

1. 選擇 **File > Export**。
2. 在 **Export** 視窗中按一下 **Test execution history**，然後按一下 **Next**。
3. 在 **Export Test Execution History** 視窗中，瀏覽並選定您想儲存日誌的資料夾，然後按一下 **Next**。如果您只輸入檔名，日誌便會匯出至安裝資料夾。
4. 選取要匯出的日誌，再按一下 **Finish**。

您也可以將整份報告或報告的某個標籤，匯出成 HTML 格式。若要將報告匯出成 HTML 檔：

1. 在 **Performance Test runs** 視圖中，用滑鼠右鍵按一下想匯出的報告或標籤，然後選取 **Export to HTML**。
2. 在 **Specify file path for HTML exported file** 中選取要儲存新建報告的資料夾，然後按一下 **Next**。現行專案爲預設，但您通常會在專案以外的資料夾儲存所匯出的報告。

3. 按一下 **Finish**。
4. 另外，您可以選擇將匯出報告貼至試算表程式，供進一步分析之用。

## 回應時間分析與資源監控資料

這部分本文並不細究（一來因為篇幅有限，二來這方面還有比筆者更專精的人士），不過您也可以納入擷取資訊，或匯入回應時間分析資料與資源監控資料，資源監控資料是由定期收集的連續觀察資料所組成，您可以收集即時資料，或從 IBM Tivoli Enterprise™ Monitoring Server 擷取資料。除了回應時間分析資料外，資源監控資料可讓您完整檢視系統，從而判斷問題所在。以下是您可收集與分析的部分資料種類：

- CPU 使用情況（總計、個別處理器，或甚至個別處理程序）
- 可用記憶體
- 磁碟使用情況
- TCP/IP 與網路產量

您可利用這項功能完整檢視您的 Web 服務，將問題區別開來。您可以使用 IBM Tivoli Monitoring 代理程式或 Rational Performance Tester，監控受測系統（或代理程式）。要檢視資源監控資料，您可以使用 Eclipse 的 Profiling and Logging 視景。

回應時間分析將顯示系統運作時，測試程式碼在系統各部分所耗費的時間量。回應時間分析視圖，與特定測試或排程執行的 Web 服務呼叫相關聯。您可以使用回應時間分析來執行下列作業：

- 識別程式碼問題
- 查看哪一個伺服器上的哪項應用程式，面臨效能瓶頸
- 進一步深入分析是哪個套件、類別或方法造成問題

要擷取回應時間分析資料，您必須在測試或排程中啟用回應時間分析，並配置想擷取的資料量。資料收集基礎架構（可見於安裝 Rational Performance Tester 工具之時）負責收集回應時間分析資料，每部執行收集資料應用程式的主機上，都必須安裝及執行資料收集基礎架構。此外，您必須配置（或檢測）每個應用程式伺服器，以便使用資料收集基礎架構。

## 總結

瞭解如何進入報告、進行變更之後，請花一點時間試用各種可用的計數器及報告；請務必嘗試將資料匯出成各種格式（特別是 CSV）。此外，爲了大型測試之用，請練習使用過濾功能並變更時間範圍，看看您能否爲測試結果精簡資訊。

## 資源

### Learn

- Visit the [Rational Tester for SOA Quality](#) resource page on developerWorks for technical articles, free tutorials, and more.
- [Introduction to IBM Rational Tester for SOA Quality V7.0 and IBM Rational Performance Tester Extension for SOA Quality V7.0](#) (developerWorks, Mar 2007): Get an overview of the functionality of these new offerings and see a real example of testing a Web service.
- SOA and Web services: Visit the [SOA area on developerWorks](#) to get the resources you need to advance your knowledge and skills.
- Browse the [technology bookstore](#) for books on these and other technical topics.
- "[Introduction to IBM Rational Performance Tester V7.0](#)," Michael Kelly (developerWorks, January 2007): Get an introduction to this product.
- [developerWorks Rational Performance Tester resource page](#): See more dW resources on Performance Tester.

### Get products and technologies

- [Google Web API WSDL](#): The WSDL for the API used in this article.
- Get a trial download of [IBM Rational Tester for SOA Quality V7.0](#).
- Get a trial download of [IBM Rational Performance Tester V7.0](#).

### Discuss

- Get involved in the [developerWorks Rational community](#) by participating in the forums and more.

## 作者簡介

Mike Kelly 是一位獨立顧問，目前定居於美國中西部。他也針對軟體測試相關主題，撰寫文章，進行演講。您可以透過他的部落格：[www.MichaelDKelly.com](http://www.MichaelDKelly.com)，閱讀他最新發表的文章。