

# User Guide Version 1.5



DBSophic LTD Givat Ram, P.O. Box 39158, Jerusalem 91391 ISRAEL Phone: +972.2.654.2742 | Fax: +972.77.444.8116 <u>www.dbsophic.com</u>

# About this Guide

This guide describes Qure Analyzer and its features. It provides detailed instructions on how to install and use Qure Analyzer to analyze your SQL Server workloads.

This guide includes the following chapters:

#### • Chapter 1: Introduction

Provides a general overview of Qure Analyzer, including its benefits and common use cases.

#### • Chapter 2: Installing Qure Analyzer

Describes how to download and install Qure Analyzer.

#### • Chapter 3: Planning the Workload Analysis

Describes the necessary steps and considerations for the process of capturing the workload for analysis using Qure Analyzer.

#### Chapter 4: Analyzing a Workload

Describes how to analyze workloads using Qure Analyzer.

#### • Chapter 5: Viewing the Workload Analysis Reports

Describes the two dynamic Qure Analyzer analysis reports. Shows how to view, navigate through and manipulate the analysis results to easily find the required analysis information.

#### Chapter 6: Comparing Workloads

Describes how to use Qure Analyzer to compare two workloads. Shows how to view, navigate through and manipulate the dynamic comparison reports to easily find the required workload comparison information.

#### Chapter 7: Managing Workload Analyses

Describes how to add traces to an existing workload analysis and how to delete workload analyses repositories from a database.

# **Table of Contents**

Ab	out t	his (	Guide1
Та	ble o	f Co	ntents 2
Lis	st of I	Figu	res5
1	Intr	odu	ction7
1	1.1	Wel	come to Qure Analyzer7
1	1.2	Use	Cases 7
	1.2.	1	Analyzing a Single Workload7
	1.2.	2	Comparing Two Workloads 8
1	1.3	Glos	ssary of Terms
	1.3.	1	Workload9
	1.3.	2	Workload Analysis9
	1.3.	3	Trace Event9
	1.3.	4	Batch (or 'Batch Template')9
	1.3.	5	Batch Instance 10
	1.3.	6	Resource Consumption Columns 10
	1.3.	7	Dimensions 10
2	Inst	tallir	ng Qure Analyzer11
ź	2.1	Gen	eral Guidelines11
ź	2.2	Inst	allation Prerequisites11
	2.2.	1	Hardware Requirements 11
	2.2.	2	Software Prerequisites
	2.2.	3	Supported SQL Server Trace Formats 11
ź	2.3	Inst	alling Qure Analyzer11
3	Plar	nnin	g the Workload Analysis14
	3.1	Plar	nning the Workload Capture14
	3.1.	1	Choosing the Applications and Databases for Analysis 14
	3.1.	2	Choosing the Data Columns for Analysis 14
	3.1.	3	Choosing the Time Periods for Analysis
	3.2	Сар	turing Workloads16
	3.2.	1	Using SQL Server Profiler
	3.2.	2	Using Server-Side Traces

3.3	Sav	ing Workload Traces1	.6
3.3	3.1	Saving to Files1	6
3.3	8.2	Saving to Tables 1	6
4 An	alyzi	ng a Workload1	7
4.1	Lau	Inching Qure Analyzer1	.7
4.1	1	Opening a Previously Analyzed Workload Report 1	.8
4.1	2	Using the 'Recent Analyses' List 2	0
4.2	Set	ting up the Workload Analysis2	21
4.2	2.1	Saving the Workload Analysis 2	1
4.2	2.2	Selecting the Analysis Workload Traces 2	3
4.2	2.3	Configuring Analysis Options 2	5
4.2	2.4	Reviewing Analysis Settings 2	7
4.3	Ana	alyzing the Workload2	27
5 Vie	ewing	g the Workload Analysis Reports2	8
5.1	Usi	ng the Workload Summary Report2	28
5.1	1	Analysis Information 2	9
5.1	2	Resource Consumption Statistics 2	9
5.2	Usi	ng the Workload Details Report3	0
5.2	2.1	Controlling the Workload View 3	1
5.2	2.2	Filtering 3	9
5.2	2.3	Changing the Resources Aggregation Mode 4	6
5.2	2.4	Workload Grid 4	6
5.2	2.5	Selection Details 4	8
6 Co	mpai	ring Workloads5	1
6.1	Ope	ening Multiple Workload Analyses5	51
6.2	Cor	nparing Workloads5	52
6.3	Usi	ng the Workload Comparison Details Report5	;3
6.3	8.1	Group by5	3
6.3	3.2	Filtering 5	3
6.3	3.3	Comparison Mode 5	4
6.3	8.4	Resource Aggregation Mode 5	;9
6.3	8.5	Resource Columns Sort Mode 5	;9
6.3	8.6	Workload Comparison Grid 6	0
6.3	8.7	Resource Consumption Details 6	1

7	Manag	ging Workload Analyses	62
7.	1 Ac	dding Traces to a Workload Analysis	62
7.	2 D	eleting a Workload Analysis Repository	62

# List of Figures

Figure 1 - Qure Analyzer Start Page	. 17
Figure 2 - Open Workload Analysis from Database	. 18
Figure 3 – Workload Summary Report	. 19
Figure 4 – Workload Details Report	. 20
Figure 5 – Start New Workload Analysis	. 21
Figure 6 - Save Workload Analysis to Database	. 22
Figure 7 - Save Analysis Results to File	. 23
Figure 8 – Select Workload Traces	. 23
Figure 9 - Select Trace Files	. 24
Figure 10 – Selected Workload Traces	. 24
Figure 11 - Select Trace Table	. 25
Figure 12 – Configure Analysis Options	. 25
Figure 13 – New Workload Analysis – Summary	. 27
Figure 14 – Workload Summary	. 28
Figure 15 – Sample Workload, All Events, All Columns	. 30
Figure 16 – Workload View Pane	. 31
Figure 17 – Conceptual Groupings Diagram	. 32
Figure 18 – Default Group by Settings	. 33
Figure 19 – Customized Group by Settings	. 33
Figure 20 – One Grouping Level with Two Columns	. 33
Figure 21 – Expanding one Grouping Level	. 34
Figure 22 – A Four-Level Grouping Hierarchy	. 34
Figure 23 - Add Top Grouping Level	. 35
Figure 24 - Add Application as Top Level Grouping	. 35
Figure 25 - Multiple Level Grouping Set	. 36
Figure 26 - Multiple Level Grouping Grid View	. 36
Figure 27 - Add Mid-level Grouping	. 36
Figure 28 – Add a grouping to the right	. 37
Figure 29 - Change Grouping Columns In-place	. 38
Figure 30 - Setting the Free Text Filter	. 39
Figure 31 – Free Text Filter Applied	. 39
Figure 32 – Free Text Filters within Batch Text	. 40

Figure 33 – Multiple Application Workload View	40
Figure 34 – Setting the Application Filter	41
Figure 35 –Application Filter Applied	41
Figure 36 - Setting Range Filters	42
Figure 37 – Filtering Example 1 – Top Applications by Duration	43
Figure 38 - Filtering Example 1 - Set Duration Filter	43
Figure 39 - Filtering Example 1 - Filtered View	43
Figure 40 - Filtering Example 1 - Empty Groups	44
Figure 41 - Filtering Example 2 - Viewing Average Duration	44
Figure 42 - Filtering Example 2 - Setting Duration Filter	45
Figure 43 - Filtering Example 2 - Filtered Duration	45
Figure 44 - Single Group Details	48
Figure 45 - Multiple Groups Details	49
Figure 46 - Switch to Batch Details Pane	49
Figure 47 - Batch Instance View	49
Figure 48 - Batch Template View	50
Figure 49 - Compare Workloads Dialog	52
Figure 50 - Comparison View	53
Figure 51 - Dual Filters Window	53
Figure 52 - Comparison Mode Control	54
Figure 53 – Example Workloads for Comparison	54
Figure 54 - Default Comparison (Include All)	55
Figure 55 - Matching Event Groups Comparison by Batch	55
Figure 56 - Matching Event Groups Comparison by Batch Instance	56
Figure 57 – Matching Event Groups Only Comparison	57
Figure 58 - Equalized Comparison	58
Figure 59 – Matching Event Groups, Equalized to Baseline	59
Figure 60 - Absolute Soring Mode	59
Figure 61 -Relative Sort Mode	60
Figure 62 - Workload Comparison Grid	60
Figure 63 – Resource Consumption Details	61
Figure 64 – Workload Analysis Management Buttons	62
Figure 65 - Workload Analysis Delete Confirmation	62

# 1 Introduction

# 1.1 Welcome to Qure Analyzer

Qure Analyzer helps you analyze database workloads comprised of one or more SQL traces. Using the intuitive graphical interface and multi-dimensional grouping, sorting and filtering, Qure Analyzer will help you easily analyze all aspects of your database workloads. For example, you can easily isolate the most resource-consuming SQL statements, applications, users, hosts, and much more.

Qure Analyzer introduces the ability to compare two database workloads. This may be needed before and after a hardware change, version upgrade or tuning effort, or to easily compare the workload performance characteristics of different environments. Even dissimilar workloads can be reliably compared. Thanks to Qure Analyzer's powerful equalization and weighting capabilities, there is no need to guess where the differences may be. Qure Analyzer not only summarizes overall differences, but pinpoints precise differences, at whatever level of granularity that you specify.

Qure Analyzer is brought to you 100% free of charge by <u>DBSophic</u>, an innovative provider of performance management products for database-centric applications. The Qure product suite also includes <u>Qure Optimizer</u> for SQL Server, the world's first <u>Workload Tuning</u> solution.

Your feedback will help us continuously improve Workload Analyzer for the entire SQL Server community. Contact <a href="mailto:support@dbsophic.com">support@dbsophic.com</a> with your comments, bug reports, and feature requests.

# 1.2 Use Cases

Qure Analyzer may be used to analyze a single, highly focused workload section to pin point a particular issue or as a high-level analytical aid to help you map your workload's hotspots and overall behavior. It can also be used to compare identical or similar workloads to find differences across multiple dimensions. If you save historical analyses of scheduled workload analyses you can use Qure Analyzer to observe trends over time and maintain highly detailed performance baseline metrics.

# 1.2.1 Analyzing a Single Workload

Use Qure Analyzer to closely explore your current production or test workloads to discover:

- Which SQL statements or batches...
  - Consume the most CPU?
  - Take the longest to run?
  - Generate the most writes?
  - Return the most rows of data?

- Why a certain batch occasionally exhibits performance issues? Is it a particular set of parameter values that causes it?
- What processes are running during that daily 14:30 slowdown and which of those processes is responsible for it?
- Which users consume most of the resources?
  - Which applications?
  - Which Hosts?
  - Which Databases?
- Which statements are executed most frequently?

## **1.2.2 Comparing Two Workloads**

Use Qure Analyzer in these types of situations:

- Your software engineers have been busy rewriting a critical set of procedures with the intention of speeding them up. Before moving these changes to production, the engineers can use Qure Analyzer to obtain accurate before-and-after comparisons by running the same workload using the original and modified set of procedures. Determine not only what improved, but also whether anything else in your entire workload suffered as a result.
- HP and Dell are competing for your new hardware budget. You can use Qure Analyzer to empirically determine which one runs your custom workload faster. To do so, use a load simulator to run your workload and capture SQL Server traces during the load period. Load the traces into Qure Analyzer and compare them to view the exact differences.
- You are about to migrate from SQL Server 2008 to SQL Server 2012. With Qure Analyzer you can accurately measure how much of an improvement (or degradation...) you'll experience after the upgrade for every process and overall. Better to know it before finding out the hard way, after the fact, in production.
- Your recent tuning effort resulted in compliments from your colleagues. Your manager wants to see empirical reports of the actual benefits. Use Qure Analyzer to compare production traces before and after applying the changes to generate detailed reports on all aspects of the improvement.
- Your recent consolidation project has caused the users of the HR application to complain about seemingly slower report generation times. Use Qure Analyzer to compare production workloads before and after the consolidation to quickly find out if they are correct, and to pinpoint the cause.

# 1.3 Glossary of Terms

Below are detailed descriptions of the terms used in this guide and in the Qure Analyzer user interface. Note that some terms, like *Batch* for example, are used in a slightly different manner than their common meaning.

#### 1.3.1 Workload

A *Workload* is a part (or all) of the application-to-database activity on your server. *Workloads* are comprised of a collection of one or more <u>SQL traces</u>. These traces contain a log of all selected events which were captured during the trace period.

#### 1.3.2 Workload Analysis

A Workload Analysis is the process of analyzing the workload traces by Qure Analyzer and saving the analysis results in a dedicated repository. This process includes reading the traces, extracting batch templates, preaggregating dimensions, optimizing storage space, indexing and more. You can later open the results of the Workload Analysis without needing to reprocess the traces. When used as a noun, the term Workload Analysis refers to the results of this process.

## **1.3.3** Trace Event

A *Trace Event* (or simply an *Event*) is a single row in one of the traces that comprise the analyzed workload. A *Trace Event* may be a single call to a procedure, a single execution of a SQL batch that includes multiple statements, a single execution of a SQL statement etc.

# **1.3.4** Batch (or 'Batch Template')

A *Batch* is a de-parameterized SQL query or a collection of commands without specific parameter values. For example:

SELECT \* FROM Employees WHERE EmployeeID = <value>

A *Batch* may be of any type of SQL query found in the workload. For example, a *Batch* may consist of a stored procedure call or an ad-hoc SQL statement.

## **1.3.5** Batch Instance

A *Batch Instance* is a SQL query including parameter values. It is comprised of a *Batch Template* plus a particular set of parameter values. For example:

SELECT \* FROM Employees WHERE EmployeeID = 346 SELECT \* FROM Employees WHERE EmployeeID = 9274

SELECT \* FROM Employees WHERE EmployeeID = 3720

To be more precise, a *Batch Instance* is the collection of all SQL queries found in the *Workload* that are identical both in structure and in parameter values. Suppose the *Workload* was found to contain 25 events whose *TextData* trace data column contains the following SQL query:

SELECT \* FROM Employees WHERE EmployeeID = 346

This set of 25 events would collectively be called a *Batch Instance*. Of course, each individual trace event of this *Batch Instance* may have different properties such as user names, start times, durations etc.

#### **1.3.6 Resource Consumption Columns**

*Resource Consumption Columns* (or *Resource Columns*) are numeric data columns in Qure Analyzer that present aggregated information about the resource consumption of the workload events for the particular group of events. The most commonly used are *Duration*, *CPU*, *Reads*, *Writes* and *Row Counts*.

#### 1.3.7 Dimensions

The data columns in Qure Analyzer that present information about the properties of the grouped events, as selected by the grouping control in the Qure Analyzer *Workload Analysis Report*. Some of these columns (*User Name* for example) are extracted from the workload trace events while others (*Batch Template* for example) are generated by Qure Analyzer during the workload analysis process.

# 2 Installing Qure Analyzer

# 2.1 General Guidelines

Qure Analyzer is available as a free download from the <u>DBSophic web site</u>. It may be used at no cost, and has no expiration date. DBSophic provides it as a goodwill gesture to the SQL Server community and as a way of introducing you to the power of holistic offline workload analysis. DBSophic's flagship product, Qure Optimizer, takes holistic offline workload analysis to the next level, and is able to automatically optimize millions of queries. Qure Optimizer is available free of charge in trial mode. <u>Download</u> your copy of <u>Qure</u> <u>Optimizer</u> today to experience <u>Workload Tuning</u> firsthand.

# 2.2 Installation Prerequisites

Qure Analyzer can be installed on any workstation or server. Saving the *Workload Analysis* to a SQL Server database requires SQL Server SysAdmin privileges on the instance where the results will be saved.

## 2.2.1 Hardware Requirements

- 2 GB RAM or more recommended.
- 50 MB free disk space for binary/executable/log files. Additional storage space will be required for each *Workload Analysis*.
- 2 GHz CPU. Dual core or better recommended.

# 2.2.2 Software Prerequisites

- Windows XP/Vista/7, Windows Server 2003/2008/2008R2, 32bit or 64bit.
- Microsoft .Net Framework 4.0 or later (<u>download here</u>).
- Microsoft SQL Server 2005/2008/2008R2/2012 client tools, 32bit or 64bit
  - Client tools are required for the analysis of trace files only.

#### 2.2.3 Supported SQL Server Trace Formats

- Microsoft SQL Server 2005/2008/2008R2/2012 trace files or tables.
- SQL Sentry Performance Advisor V6.0 or higher repository.

# 2.3 Installing Qure Analyzer

To install Qure Analyzer, follow these steps:

- **1** Download the Qure Analyzer installation file from the <u>Qure Analyzer</u> <u>download page</u> on the DBSophic web site.
- **2** Choose one of the following options:

• Save it to a local hard drive, and then double-click the Windows Installer file *QureAnalyzer.exe*.

OR

- From the Open File Dialog, click **Run**.
- **3** The installation process starts and the *Installation Wizard Welcome* dialog appears.



4 Click **Next**. The *License Agreement* dialog appears.



**5** Read the license agreement. If you accept it, select "*I accept the terms in the License Agreement"* and click **Next**. The *Select Installation Folder* dialog appears.

Select Installation Folder This is the folder where Qure	Analyzer will be installed.	DBSophic
To install in this folder, click "! "Browse".	Next". To install to a different	folder, enter it below or click
<u>F</u> older:		
C:\Program Files (x86)\DBSoph	ic\Qure Analyzer\	Browse

In the *Folder Text Box*, enter the path to an installation folder for Qure Analyzer or click **Browse** to navigate and select a different folder. The default installation folder is "%*Program Files*%\DBSophic\Qure Analyzer".

**Note**: The installation folder is used to store Qure Analyzer binary, executable and configuration files. Log files are saved to "*My Documents\Qure\Qure Analyzer\"*. Later you will select the location to store Qure Analyzer workload analysis results files, which may require significantly more disk space. For more information see section 4.2.1 – Saving the Workload Analysis.

6 Click **Next** to continue. The *Ready to Install* dialog appears.



**7** Click **Install**. The installation process starts. The installation process may take a few minutes. The *Installing Qure Analyzer* dialog appears.

Installing Qu	re Analyzer		DBSophic
Please wa minutes.	it while the Setup Wizard ins	talls Qure Analyzer.	This may take several
Status:	Registering product		
consed Testaller			

**8** When the installation is successfully completed, the *Completing the Qure Analyzer Setup* dialog appears.

O	Completing the Qure Analyzer Setup Wizard
Ċ	Click the "Finish" button to exit the Setup Wizard.
	V Launch Qure Analyzer
	< Back Finish Cancel

9 Click Finish. Qure Analyzer is now installed on your computer.

# **3** Planning the Workload Analysis

# 3.1 **Planning the Workload Capture**

Since all of the information Qure Analyzer will have about your database workload comes from SQL Server traces, you'll want to set up your traces of your production or test environment to capture all of the information you want to analyze. Since capturing trace information places a burden on the database server, you'll want to capture no more than what is necessary. Planning the workload capture has three main aspects:

- 1. Deciding which applications and databases you want to analyze
- 2. Deciding which events and data columns you want to analyze
- 3. Deciding which time periods you want to analyze

# 3.1.1 Choosing the Applications and Databases for Analysis

Qure Analyzer can analyze any number of databases at once, allowing you to examine cross-database activities. It can also analyze any number of applications that access those databases, because each access event can be traced. The key here is to minimize noise. Although you can filter out noise very effectively in Qure Analyzer afterwards, it is better to have a wellfocused workload to begin with. Doing so will place a smaller burden on your database server at the time of trace data collection, reduce the Qure Analyzer analysis duration, reduce the size of the analysis results repository, and improve the dynamic analysis report performance.

# 3.1.2 Choosing the Data Columns for Analysis

Qure Analyzer requires that you select at least one of the listed workload dimensions and all of the listed resource columns. *Row Count* is optional.

# Workload Dimensions

- Batch Instance (TextData)
- Server Name
- Database Name
- Event Class
- Host Name

- Application Name
- NT User Name
- Error
- Start Time
- End Time

# **Resource Consumption Columns:**

Duration

CPU

Reads

Row Count

Writes

Planning the Workload Analysis



**Tip**: Qure Analyzer provides you with recommended trace templates that you can use to capture your workload. You can find them in the *Trace Templates* folder in the Qure Analyzer installation folder. You can quickly access this folder by clicking on the *Open Templates Folder* link on the Qure Analyzer start page.

# 3.1.3 Choosing the Time Periods for Analysis

#### Analyzing One Workload

You may be analyzing the overall behavior of your database application or just part of it.

If you plan to analyze its overall behavior, obtain either a single long continuous trace or several shorter traces that together represent the full range of activities that the application and databases perform. For example, you may capture some traces during mid-day, evening, weekend and monthend timeframes.

If you are analyzing the behavior of a specific part of your database or application, capture traces during the timeframe when your particular events are likely to take place. For example, if you know that your application commonly performs poorly whenever a certain overnight batch process runs, capture traces during the entire runtime of that batch, and do so over several days. The set of traces can be provided to Qure Analyzer to represent a single workload that focuses on that one batch process. Qure Analyzer report provides a wide array of smart filters to pinpoint the issue.

#### Comparing Two Workloads

You may be analyzing the effect of a tuning effort on your production workload, and would like to capture one workload before the change and another workload after the change to reveal the precise size and nature of the impact. In this case, two considerations are important:

- 1. Plan to capture a '*before'* workload that has fairly similar characteristics to the '*after'* workload. For example, capture the same times of day, and capture times when the same scheduled jobs are running.
- 2. If possible, plan to capture a longer timeframe for both workloads than what you would capture if you were just analyzing one workload. The reason for this is that when selecting your comparison mode in Qure Analyzer, some of the comparison options will discard non-matching batches, and so you will effectively end up with fewer comparison points than if the two workloads contained identical events.
- 3. You may be performing a performance test to predict the impact of a server upgrade, or you may be testing the performance of two vendors' products. In this case, you will be capturing one production workload (or using a load simulator to simulate a realistic workload) and replaying it twice or more in a non-production environment. This is called a *Controlled Workload* comparison. Plan the capture of your one *Controlled Workload* as you would plan a single workload trace. Read more about controlled vs. uncontrolled workloads in chapter 6 Comparing Workloads.

# 3.2 Capturing Workloads

Qure Analyzer considers a set of traces to comprise a Workload.

## 3.2.1 Using SQL Server Profiler

MSDN has a complete <u>set of instructions</u> on how to use *SQL Server Profiler* to capture traces.

# 3.2.2 Using Server-Side Traces

For heavily loaded systems, *SQL Server Profiler* may not be the best choice as it may incur noticeable overhead. Server-side tracing can be performed at a much lower overhead. See this MSDN <u>article</u> for more information.

# 3.3 Saving Workload Traces

If you are using *SQL Server Profiler* to capture your traces, you can save them to files or to database tables. If you are using Server-side tracing, you can only save the workload to files. You can later load those files into trace tables by using *SQL Server Profiler* or the <u>sys.fn trace gettable</u> system function.



**Note**: It is highly recommended that you DO NOT check the "*Save to File"* or "*Save to Table"* options of SQL Server Profiler during the trace as it incurs a significant overhead. Instead, save the trace after you have completed and stopped the trace.

# 3.3.1 Saving to Files

The main advantage of saving traces to files is portability. When trace files are the source of trace data, Qure Analyzer requires SQL Server client tools to be installed in order to be able to decode the trace files and extract the trace information.

# 3.3.2 Saving to Tables

The main advantage of saving traces to tables is performance. A second advantage is that the data within the trace tables can be manually edited using UPDATE and DELETE SQL statements before you use them as inputs to Qure Analyzer. For example, you can replace non-intuitive stored procedure names with names people are more likely to recognize, making the analysis reports easier to understand. When trace tables are used as the source of trace data, Qure Analyzer does not require SQL Server client tools.



**Note**: If you encounter errors trying to open trace files for analysis in Qure Analyzer, we recommend that you open these trace files in profiler, save it to a trace table and use the trace table as the workload source for analysis. Alternatively, you can use the *sys.fn\_trace\_gettable* system function. For more information see the following <u>Books on Line article</u>.

# 4 Analyzing a Workload

# 4.1 Launching Qure Analyzer

In the Windows *All Programs* listing, find the DBSophic folder. Within it is the shortcut for Qure Analyzer. Click to open. Alternately, double-click the Qure Analyzer icon on your desktop.

Qure Analyzer's Start Page appears.



#### Figure 1 - Qure Analyzer Start Page

From this page, you may perform any of the following tasks:

#### **New Workload Analysis**

Start a new workload analysis. See section 4.2 - Setting up the Workload Analysis.

#### **Compare Two Workloads**

Launch a new window with instructions on how to compare two workloads. See chapter 6 - Comparing Workloads.

#### **Open Workload Analysis**

Open a previously analyzed workload analysis results. See section 4.1.1 - Opening a Previously Analyzed Workload.

# 4.1.1 Opening a Previously Analyzed Workload Report

There are two ways to open a previously saved Workload Analysis.

#### **Open Workload Analysis from File**

Use this option if the results of your previously analyzed workload are stored in a Qure Analyzer file. Qure Analyzer analysis results file uses the \*.qwan suffix.

- 1 Under the label Open Workload Analysis, click the From File link.
- 2 An Open File dialog will appear that will allow you to select an analysis results file. Multiple selection is not supported. However, you may repeat this procedure to open multiple workload analysis reports side by side.

#### **Open Workload Analysis from Database**

Use this option if the results of your previously analyzed workload are stored in a database.

- 1 Under the label *Open Workload Analysis*, click the **From Database** link.
- 2 An *Open Workload from Database* dialog will appear. It will allow you to select from existing workload analyses within one of your databases.

Open Workload	Analysis X
Open Workload	Analysis from Database
Server:	•
Authentication:	Windows Authentication 🔹
Database:	•
Available Workloa	ad analyses:
	OK Cancel

**Figure 2 - Open Workload Analysis from Database** 

Once you have selected an analysis results file or a SQL Server database analysis results repository, the workload analysis report will load and the *Workload Summary Report* page will be shown.

Analyzei							
art Page	Demo ×						
Summary	🔀 Details	Comp	are to Baselin	ne			Add Tra
		_					
emo \	Norkload	Sum	mary (s	how Details)			
alvsis In	formation						
ocation: C:\l	Jsers\Ami.DBSOPHI	iCHO\Docur	ments\Oure\C	Jure Analyzer\Demo\[	Demo.gwan		
Vorkload Tr	races						
Trace		Time	e Range		Event Count		
C\LIsers\na		Doci Unkr	nown		937		
source (	Consumption S	Statistics					
esource C	Consumption S urce Consumption	Statistics	5				
esource C overall Reso Resource	Consumption S Jurce Consumption Total	Statistics	5 Min	Max			
esource C overall Reso Resource Duration	Consumption S Furce Consumption Total 2.67 min	Statistics	Min ns 0 μs	Max 10.21 sec			
esource C overall Reso Resource Duration CPU	Consumption S Furce Consumption Total 2.67 min 3.4 min 7 704	Statistics Average 171.23 m 217 ms	Min ns 0 μs 0 μs	Max 10.21 sec 22.31 sec			
esource C everall Reso Resource Duration CPU Reads Writer	Consumption S aurce Consumption Total 2.67 min 3.4 min 7.79M 200	Average 171.23 m 217 ms 8.32K 0.42	5 Min ns 0 μs 0 μs 6	Max 10.21 sec 22.31 sec 443.67K 22			
esource C everall Reso Resource Duration CPU Reads Writes	Consumption S Total 2.67 min 3.4 min 7.79M 390	Average 171.23 m 217 ms 8.32K 0.42	<ul> <li>Min</li> <li>ns</li> <li>0 μs</li> <li>0 μs</li> <li>6</li> <li>0</li> </ul>	Max 10.21 sec 22.31 sec 443.67K 22			
esource C everall Reso Resource Duration CPU Reads Writes	Consumption 1 rurce Consumption 2.67 min 3.4 min 7.79M 390 ing Batches	Average 171.23 m 217 ms 8.32K 0.42	5 Min ns 0 μs 0 μs 6 0	Max 10.21 sec 22.31 sec 443.67K 22			
esource C verall Reso Duration CPU Reads Writes op Consum Resource	Consumption 9 rurce Consumption 2.67 min 3.4 min 7.79M 390 ing Batches Top Consumer	Statistics Average 171.23 m 217 ms 8.32K 0.42	Min     0 μs     0 μs     6     0  Event Count	Max 10.21 sec 22.31 sec 443.67K 22 Total	Average	More Information	
esource C verall Reso Resource Duration CPU Reads Writes op Consum Resource Duration	Consumption 5 Total 2.67 min 3.4 min 7.79M 390 ing Batches Top Consumer SELECT Sales Sale	Statistics Average 171.23 m 217 ms 8.32K 0.42 E sOrderH 1	Min ns 0 μs 0 μs 6 0	Max           10.21 sec           22.31 sec           43.67K           22           Total           47.16 sec	Average 3.63 sec	More Information Show Batches by Duration	
esource C verall Reso Duration CPU Reads Writes op Consum Resource Duration CPU	Consumption 5 Total 2.67 min 3.4 min 7.79M 390 ing Batches SELECT Sales.Sale SELECT Sales.Sale	Statistics Average 171.23 m 217 ms 8.32K 0.42 sOrderH 1 sOrderH 1	5 Min ns 0 μs 0 μs 6 0 2 Event Count .3 .3	Max           10.21 sec           22.31 sec           443.67K           22           Total           47.16 sec           2.17 min	Average 3.63 sec 10 sec	More Information Show Batches by Duration Show Batches by CPU	
Resource C Duration CPU Reads Writes op Consum Resource Duration CPU Reads	Consumer Total 2.67 min 3.4 min 7.79M 390 Img Battes SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale	Statistics Average 171.23 m 217 ms 8.32K 0.42 sOrderH 1 sOrderH 1 sOrderH 1	<ul> <li>Min Nin O μs O μs O μs O μs</li> <li>O μs O μs</li> <li>O μs O μs</li> <li>A S O μs&lt;</li></ul>	Max           10.21 sec           22.31 sec           443.67K           22           Total           2.17 min           2.55M	Average 3.63 sec 10 sec 159.48K	More Information Show Batches by Duration Show Batches by CPU Show Batches by Reads	
esource C verall Reso Duration CPU Reads Writes Op Consum Resource Duration CPU Reads Writes	Consumption 5 Total 2.67 min 3.4 min 7.79M 390 Top Consumer SELECT Sales.Sale SELECT Sales.Sale SELECT c.Employee	Statistics Average 171.23 m 217 ms 8.32K 0.42 sOrderH 1 sOrderH 1 sOrderH 1 sOrderH 1 eED, Nat 1	<ul> <li>Min Nin O μs G O 0</li> <li>Event Count 3.3</li> <li>.6 5.5</li> </ul>	Max           10.21 sec           22.31 sec           43.67K           22         Total         47.16 sec       2.17 min           255M           122	Average 3.63 sec 10 sec 159.48K 8.13	More Information Show Batches by Duration Show Batches by CPU Show Batches by Reds Show Batches by Writes	
esource C verall Reso Duration CPU Reads Writes op Consum Resource Duration CPU Reads Writes op Consum	Consumption 5 Total 2.67 min 3.4 min 7.79M 390 ing Batches Top Consumer SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Cales.Sale SELECT Cales.Sale SELECT Cales.Sale SELECT Cales.Sale SELECT Cales.Sale SELECT Cales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Sale.Sale SELECT Sale.Sale Sale SELECT Sale.Sale Sale Sale Sale Sale Sale Sale Sale	Statistics Average 171.23 m 217 ms 8.32K 0.42 sOrderH 1 sOrderH 1 sOrderH 1 sOrderH 1 sOrderH 1 sOrderH 1 sOrderH 1 sOrderH 1	5 Min ns 0 μs 6 0 0 Event Count .3 .6 .5	Max           10.21 sec           22.31 sec           22.31 sec           43.437K           22	Average 3.63 sec 10 sec 159.48K 8.13	More Information Show Batches by Ouration Show Batches by CPU Show Batches by Reads Show Batches by Writes	
esource C verall Reso Resource Duration CPU Reads Writes Op Consum Resource Duration CPU Reads Writes Op Consum Resource	Consumption 5 Total 2.67 min 3.4 min 7.79M 390 ing Batches Top Consumer SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale SELECT Sales.Sale	Statistics Average 171.23 m 8.32K 0.42 sOrderH 1 soOrderH 1 soOrderH 1 seID, Nat 1	Min           ns         0 μs           0 μs         6           0         0	Max           10.21 sec           22.31 sec           43.67K           22           Total           47.16 sec           2.17 min           2.55M           122           Total	Average 3.63 sec 10 sec 159.48K 8.13 Average	More Information Show Batches by Ouration Show Batches by CPU Show Batches by Reads Show Batches by Writes More Information	
esource C verall Reso Duration CPU Reads Writes op Consum Resource Duration CPU Reads Writes op Consum Resource Duration CPU Reads Writes	Consumption 5 Total 2.67 min 3.4 min 7.79M 390 Ing Batches Top Consumer SELECT Sales.Sale SELECT Sales.Sale SELECT Cales.Sale SELECT Cales.Sale SELECT Sales.Sale SALES SALE SALE	Statistics Average 171.23 m 8.32K 0.42 sorderH 1 sorderH 1 sorderH 1 sorderH 1 sorderH 1 sorderH 1 sorderH 1	Min     0 μs     0 μs     0 μs     0 μs     6     0      Event Count	Max           10.21 sec           23.31 sec           443.67K           22           Total           2.55M           122           Total           39.19 sec	Average 3.63 sec 10 sec 159.48K 8.13 Average 3.56 sec	More Information           Show Batches by Duration           Show Batches by CPU           Show Batches by Reads           Show Batches by Writes           More Information           Show Batch Instances by Duration	
esource C Duration CPU Reads Writes op Consum Resource Duration CPU Reads Writes op Consum Resource Duration CPU Duration CPU CPU CPU CPU Reads	Consumer Total 2.67 min 3.4 min 7.79M 390 ing Batches Top Consumer SELECT Sales Sale SELECT Sales Sale SELECT Sales Sale SELECT Sales Sale SELECT Sales Sale SELECT Sales Sale SELECT Sales Sale	Statistics Average 171.23 m 217 ms 8.32K 0.42 sorderH+ 1 sorderH+ 1 sorderH+ 1 sorderH+ 1 sorderH+ 1 sorderH+ 1 sorderH+ 1	Min     0 μs     0 μs     6     0      Event Count       Event Count       Event Count	Max           10.21 sec           22.31 sec           423.67 K           22           Total           47.16 sec           2.17 min           2.55M           122           Total           1.2	Average           3.63 sec           10 sec           159.48K           8.13           Average           3.56 sec           10.02 sec	More Information           Show Batches by Duration           Show Batches by CPU           Show Batches by Reads           Show Batches by Writes           More Information           Show Batch Instances by CPU           Show Batch Instances by CPU	

#### Figure 3 – Workload Summary Report

To use the *Summary Report*, see section 5.1 - Using the Workload Summary Report.

To review the detailed results of the *Workload Analysis*, click the **Details** icon on the top bar. The *Details Report* page appears.

Ostart Page         Demo *           Summary         Details         Compare to Baseline           Workload View         *         *         Batch         Duration         CPU         Reads         Writes         Event Col           * Save\Load Settings         8         SELECT Sales.SalesOrderHeader         47.16 sec         2.217 min         2.40M         1         8         1         1         8         1         1         8         1         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1	Add Trace
Summary         Details         Compare to Baseline           Workload View              • Batch               Batch               Duration               PPU               Reads               Writes               Event Co                  • Save\Load Settings               B StLECT Sales.SalesOrderHeader               4716 sec               2.237 min               2.434M               8               PU               8               BLCCT Sales.SalesOrderHeader               19.29 sec               14.36 sec               2.255M               0               0               Batch               0               8               StateSalesOrderHeader               14.38 sec               14.38 sec               1               0               0               0               1               0               0               1               0               0               0               0               0               0         <	Add Trace
Workload View         Image: Sales Sales OrderHeader         Duration         CPU         Reads         Writes         Event Cr           • SaveLload Settings         81         SELECT Sales Sales OrderHeader         4716 sec         2.17 min         2.49M         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         8         1         0         1         1         0         1         1         1         0         1         1         1         0         1         1         0         1         1         0         1         1         1         1         0         1         1         1         1         1         1         0         1 </th <th>I I E I I I I I I I I I I I I I I I I I</th>	I I E I I I I I I I I I I I I I I I I I
Save\Load Settings         80         SELECT Sales.SalesOrderHeader         4716 sec         2.17 min         2.49M         1         8         1           • Group By         ii         SELECT Sales.SalesOrderHeader         10.29 sec         1.136 sec         2.55M         1         0         1           • Batch • • • • • • • • • • • • • • • • • • •	
* Group By       * SELECT Sales.SalesOrderHeader       1929 sec       1436 sec       253M       0       9         * Batch * •       * SELECT Sales.SalesOrderHeader       1588 sec       1149 sec       644K       3       9         * Batch * •       * SELECT Sales.SalesOrder       663 sec       1436 sec       0458 st.       0       0         * Batch * •       * SELECT FROM Sales.SalesOrder       666 sec       1438 sec       0458 st.       0       0         * SELECT * FROM Sales.SalesOrder       666 sec       1438 sec       0428 st.       0       0         * SELECT * SOD.SalesOrderID, P.Na       584 sec       1       665 sec       1431 sec       140       0         * SELECT * SOD.SalesOrderID, P.Na       584 sec       1       665 sec       1317 sc.       40       0	
• Group By         • B SELECT Sales Sales Order Header         • 1588 sec         • 1149 sec         • 149 sec         • 63.4K         • 1         • 3         • SELECT * FROM Sales Sales Order         • 8.32 sec         • 1         3.52 sec         • 1         45.85K         • 1         • 0         • 1         • 3         SELECT * FROM Sales Sales Order         • 8.32 sec         • 1         45.85K         • 1         • 0         • 1         • 3         SELECT * FROM Sales Sales Order         • 6.66 sec         • 1         42.8 sec         • 92.86K         • 1         • 0         • 1         • 3         SELECT * FROM Sales Sales Order         • 5.84 sec         • 6.65 sec         • 1         • 3179K         • 40         •         •         • SELECT * SOD Sales OrderID, P.Na         • 5.84 sec         • 1         • 312 sec         • 1         • 31         • 0         •         • •	
************************************	
• Bach • • • • • • • • • • • • • • • • • • •	
Batch Instance              •             •	
SELECT SOD SalesOrderID, P.Na 492 sec   3.12 sec 1.89M   0	
	1
Apply 🖆 SELECT FirstName + <value>La 4.69 sec 1.94 sec 19.11K 11</value>	
SELECT P.Name, COUNT(*) FRO ■ 3.97 sec ■ 3.58 sec ■ 59.97K ■ 19	
▲ Filter By ④ SELECT * FROM Sales/Sales/Orde ■ 3.4 sec   1.8 sec   28.55K   1 ■	1
Add Filter	1
SELECT O.SalesOrderID, Custom 1 2.72 sec   2.12 sec   33.42K ■ 10	1
Kesource Aggregation Mode     SELECT e.EmployeeID, National 1 2.63 sec   1.51 sec   15.01K 122	1
● Total      SELECT ContactID, FirstName, L   2.55 sec   953 ms   13.68K ■ 49	1
ELECT Sales/Sales/OrderHeader 1 2.49 sec   538 ms   26.22K   0     ■	1
SELECT e.EmployeeID, National     245 sec   1.66 sec   4.51K   0	<b>I</b> 1
ELECT ContactID, FirstName, L 1 2.02 sec   34.78K   0	<b>I</b> 1
BELECT SOD.SalesOrderID, P.P I 201 sec   1.47 sec   15.86K ■ 11     ■	1
ELECT SalesOrderID, OrderDat   1.67 sec   1.21 sec   40.41K   0	I
Pressure Computer	
Resource consumption	*
Select one or more records to view their performance.	

#### Figure 4 – Workload Details Report

To use the *Workload Details Report*, see section 5.2. - Using the Workload Details Report.

**Tip**: You can open multiple *Workload Analysis Reports* in Qure Analyzer. Just go back to the *Start Page* by clicking the **Start Page** tab, and repeat these steps. For each workload, a new tab will open at the top of the Qure Analyzer ribbon. Each of these tabs has its own *Summary Report* and *Details Report* pages.



#### 4.1.2 Using the 'Recent Analyses' List

On the Qure Analyzer *Start Page* is a list of the most recent workload analyses. They are arranged with the most recently used ones at the top of the list.

- **1** Move the mouse cursor over any one workload analysis name to see the storage location of that workload analysis.
- 2 Click the **Workload Name** to open. Only one workload analysis can be clicked at a time. However, you may open more *Workload Analyses* by repeating this same procedure.
- **3** Once you have selected the name of an existing workload analysis, it will automatically load the *Workload Summary Report*.

# 4.2 Setting up the Workload Analysis

To begin a new Workload Analysis, click **New Workload Analysis** from the *Start Page*. The *Start New Workload Analysis* dialog box appears.

Start New Workloa	d Analysis						
Results of a workload analysis may be saved to a file or to a SQL Server database. Saving to a file provides better portability. Saving to a SQL Server database is recommended for better performance.							
Save to:							
SQL Server Database File							
Location:	Location:						
Server:	<b></b>						
Authentication:	Windows Authentication						
Database:	<b></b>						
Workload Name:							
	ОК Сапсе						

Figure 5 – Start New Workload Analysis

# 4.2.1 Saving the Workload Analysis

The results of Qure Workload Analysis are stored in a Qure Analyzer Analysis file or to a SQL Server database.



**Note**: You can save multiple workload analyses to the same SQL Server database. For each analysis, Qure Analyzer will create a separate schema that contains the analysis result tables. It will be named after the workload name that you provide in the new workload analysis wizard.

Each *Workload Analysis* file or schema holds the results of one workload analysis. For example, if you are comparing the performance of three servers being considered for purchase, and have captured traces from all three of them, the analyzed results will be stored in three result locations (three .qwan files or three schemas) for later review.



**Note**: The *Workload Comparison Report* is not stored in any repository. It is regenerated each time you perform a comparison of two workload analyses.

The dialog shown in Figure 5 – Start New Workload Analysis above allows you to specify that the results will be stored in either a SQL Server Database (the default option) or in a file.

The advantages of saving your results in a SQL Server database Repository are:

- Provides better performance
- Does not impose a 4GB size limit on the analysis results file
- Offers the option to later use the workload for comparison

This is the recommended option when analyzing large workloads or when using advanced analysis options such as *Save Individual Batch Parameters* or *Analyze Start and End Times*.

The main advantage of saving your analysis results in a *File Repository* is portability. Once the analysis is complete, the analysis file is completely portable, and can be easily shared with colleagues who also use Qure Analyzer, even if they have no connectivity to the database.

#### Saving the Analysis to a SQL Server Database

1 Select your server, authentication method, and database using the provided text boxes and pull-down menus. Enter a workload name. This will become the name of the schema that Qure Analyzer will create to store the *Workload Analysis*.

Start New Workload Results of a workload Server database. Sav a SQL Server databa	d Analysis d analysis may be saved to a file or to a SQL ing to a file provides better portability. Saving to se is recommended for better performance.				
Save to:	Save to: SQL Server Database File				
Server:	AMI-VAIO\sql2012				
Authentication:	Windows Authentication				
Database: Test_DB					
Workload Name: Demo					
	OK Cancel				

Figure 6 - Save Workload Analysis to Database

2 Click **OK** to create the *Workload Analysis* repository.

The dialog shown in Figure 8 within section 4.2.2 - Selecting the Analysis Workload Traces below will appear.

#### Saving the Analysis to a File

The dialog shown in Figure 5 – Start New Workload Analysis above, defaults to saving the analysis results in a SQL Server Database. To save to a file, follow these steps.

- 1 Under the option *Save to*, click **File.**
- 2 The *Location* text boxes will change to look like this:

Start New Workloa	d Analysis
Results of a workload Server database. Sav a SQL Server databas	d analysis may be saved to a file or to a SQL ing to a file provides better portability. Saving to se is recommended for better performance.
Save to:	Database 🎦 File
Location:	
Workload Name:	
Folder:	C:\Users\Ami.DBSOPHICHQ\Documents\Qu
	OK Cancel

Figure 7 - Save Analysis Results to File

- **3** Enter the name you would like to give your workload analysis and your .qwan file under *Workload Name*. You do not need to provide the .qwan suffix as Qure Analyzer will add it automatically. Enter a folder name or browse to select a folder into which the .qwan file will be created.
- 4 Click OK.

The dialog shown in Figure 8 within section 4.2.2 - Selecting the Analysis Workload Traces will appear.

#### 4.2.2 Selecting the Analysis Workload Traces

After you have specified where to save the results of the *Qure Workload Analysis*, the *New Workload Analysis* dialog appears. The wizard will guide you through the next steps. The steps are also listed on the left pane.

P New Workload Analysis	
<ul> <li>Setup Workload Analysis</li> <li>Select Workload Traces</li> <li>Configure Analysis Options</li> </ul>	Select Workload Traces Select one or more traces that comprise your workload
<ul> <li>Summary</li> <li>Analysis Progress</li> </ul>	
	Add Trace* Remove
	Next Cancel

#### Figure 8 – Select Workload Traces

As mentioned earlier, traces can be provided as trace files or as trace tables.

#### Adding Trace Files to the Workload

Click Add Trace. From the two selections that appear, select Add Trace File.

A *File Open* dialog will appear. Note that it defaults to show only *SQL Server Profiler* trace (\*.trc) files.

P Open	outer	Local Disk (C:)      Temp     Sample tra	ces ·	✓ 4 Search Sample traces
Organize • New folder				
☆ Favorites	-	Name	Date modified	
E Desktop		Bursting_1.trc	16/01/2011 18:14	
🐌 Downloads		Bursting_2.trc	16/01/2011 18:14	
👢 Dropbox		grocery1.trc	10/02/2010 12:02	
laces 😓 Recent Places	Ξ	ja grocery2.trc	10/02/2010 12:03	
🞾 Google Drive		📄 huncher1.trc	28/09/2010 10:01	
		📄 huncher2.trc	28/09/2010 10:01	Select a file to preview.
🥞 Libraries		Pare_EyeShare20100829.trc	29/08/2010 17:37	
locuments 🔍		Qure_EyeShare20100830.trc	30/08/2010 18:07	
🜛 Music		QWA_Demo_Baseline.trc	27/01/2011 15:21	
Sectores		📄 QWA_Demo_Evaluated.trc	27/01/2011 15:22	
Judeos 💐		📔 test_replay.trc	12/04/2011 11:08	
🚜 Homearoup	Ŧ	•	Þ	
File	nam	e:	-	SQL Trace Files (*.trc)
				Open 🔽 Cancel

#### Figure 9 - Select Trace Files

Select the one or more trace files that represent the workload (or portion of your entire workload) you would like to analyze.



**Tip:** You can select multiple files using the standard Windows selection keys (**<Ctrl>**, **<Shift>**, up or down arrows, or **<Ctrl>+A** to select all files in the current folder.

Once you have completed your selections, click **Open.** The traces selected so far are displayed for your confirmation.



Figure 10 – Selected Workload Traces

You may continue to add additional traces (from either files or tables) until your selections are complete. Once you have completed your selection, click **Next.** 

The dialog shown in Figure 12 within section 4.2.3 - Configuring Analysis Options will appear.

#### Adding Trace Tables to the Workload

Click Add Trace. From the two available selections, select Add Trace Table.

P

**Tip:** New in version 1.5 - You can select a <u>SQL Sentry</u> <u>Performance Advisor</u> repository trace tables as your workload source. Note that by default, only the top SQL in terms of duration are stored in these tables. You can reduce the threshold for collecting top SQL or perform quick-traces to collect a wider range of queries. See the <u>SQL Sentry</u> <u>Performance Advisor user guide</u> for details.

Select Trace Tab	le			
Select a table containing trace data.				
Server: Authentication:	Windows Authentication			
Database: Table:	<b></b>			
	OK Cancel			

Figure 11 - Select Trace Table

Select Server, Authentication method, Database Name, and the trace using the provided text boxes and pull down menus. Click  $\mathbf{OK}$ .

You may continue to add additional traces to your workload, from either trace files or tables. When you complete your selection of traces, click **Next**. The dialog shown in Figure 12 within section 4.2.3 - Configuring Analysis Options will appear.

# 4.2.3 Configuring Analysis Options

P New Workload Analysis	
Setup Workload Analysis     Select Workload Traces     Configure Analysis Options	Configure Analysis Options
<ul> <li>Summary</li> </ul>	Select Trace Data Columns
Analysis Progress	Batch     Addition     Andyze Execution Times     Save Individual Batch Parameters
	Previous Next Cancel

Figure 12 – Configure Analysis Options

This dialog in the wizard shows that you have completed selecting the workload traces, and are ready to continue to *Configure Analysis Options*.

#### Selecting Trace Data Columns

This section displays a list of check boxes showing all of the relevant data columns found in the trace files. The number of data columns depends on the data columns that are included in the trace (see section 3.1.2 - Choosing the Data Columns for Analysis).



**Note**: Qure Analyzer will only display a column for selection if it is present in all of the traces that comprise the workload.

Default columns used for workload analyses are pre-checked. Tailor the selections according to your own preferences. Qure Analyzer will only analyze the data columns you have selected.

#### Selecting Additional Analysis Options

In addition to analyzing the data columns you specified above, Qure Analyzer will also perform any or all of the optional analysis tasks. The number of additional options offered depends on what data columns are available for the workload.

• Analyze Execution Times

When you check this option, Qure Analyzer will include in the analysis the execution times of all events. Select this if you would like to analyze whether your processes perform differently at different times and days by grouping events by time, or if you want to be able to filter the results by time and day.



**Note**: The minimal resolution for the time dimension is one hour.

• Save Individual Batch Parameters

When you check this option, Qure Analyzer will provide a deeper level of analysis for each batch by retaining *Batch Instance* details.

When Qure Analyzer provides an analysis of a *Batch*, it sums the metrics (duration, reads, writes, CPU) of every execution of that *Batch*, regardless of parameter values, and provides a total view of the performance of that entire collection of *Batch Instances*.

When Qure Analyzer provides an analysis of *Batch Instances* (based on your selection of the option *Save Individual Batch Parameters*), it allows you to drill-down further. You'll be able to see how each *Batch Instance* performs on its own merit. That is, one set of performance numbers for each *Batch Instance*, in addition to the performance numbers at the higher *Batch* level. For more information on *Batch Instances*, see section 1.3.4 - Batch (or 'Batch Template').

If you have a SQL Statement that performs badly only when using certain sets of parameter values, we highly recommended that you select this option.

• Analyze Events with Errors

When you check this option, Qure Analyzer will include events that have not completed successfully (error > 0). When you don't check this checkbox, Qure Analyzer will exclude erroneous events from the analysis.

Click **Next**. The New Workload Analysis Wizard will advance to the *Summary Report* dialog (See Figure 13 within section 4.2.4 - Reviewing Analysis Settings).

# 4.2.4 Reviewing Analysis Settings



Figure 13 – New Workload Analysis – Summary

This dialog offers you a chance to review all settings before beginning the analysis. If you see something that needs to be changed, click **Previous**.

If you are satisfied with the settings, and would like to start the workload analysis process, click **Start**.

# 4.3 Analyzing the Workload

Once started, Qure Analyzer works completely unattended. Upon completion, it will present you with a new *Workload Analysis Report* tab. The report will be named after the workload name you provided when you first set up this workload analysis (see section 4.2.1 - Saving the Workload Analysis). Within this new tab, you will see the *Summary Report*, as shown in Figure 14 in section 5.1 - Using the Workload Summary Report.

# 5 Viewing the Workload Analysis Reports

Two reports are automatically generated upon completion of a Workload Analysis.

- 1. The Workload Summary Report
- 2. The Workload Details Report

# 5.1 Using the Workload Summary Report

The *Workload Summary Report* appears by default whenever a new workload analysis completes or whenever a previously analyzed workload is reopened.

If you are not already viewing the *Workload Summary Report* and want to do so, click the **Summary** button on the top ribbon, while working in any single workload analysis results tab.

Start Page	B1 ×							
Summary	Details	Comp	oare to Basel	ine			🗙 Delete Analysis	📑 Add Tra
81 Wor	kload Su	mma	ry (Show [	Details)				
nalysis In	formation							
Workload Tim Total Number Location: B1 c	e Range: 22/12/200 of Events: 115757 onBursting_Analysi	08 06:43 - 1 es	22/12/2008 (	)7:54				
Workload Tra	aces							
Trace		Tim	e Range		Event Count			
[Bursting_Tr	aces].[dbo].[Trace_A	A] 22/1	12/2008 06:4	3 - 22/12/2008 07:54	115757			
Overall Reso	urce Consumption	statistic	S					
Overall Resource C	urce Consumption	Average	s e Min	Max				
esource C Overall Reso Resource Duration	Total	Average 13.78 m	e Min Is 3μs	Max 1.23 min				
esource C Overall Resource Duration CPU	urce Consumption Total 26.58 min 18.01 min	Average 13.78 m 9 ms	e Min is 3 μs 0 μs	Max 1.23 min 28.24 sec				
esource C Overall Resource Duration CPU Reads	Total 26.58 min 18.01 min 119.57M	Average 13.78 m 9 ms 1.03K	<b>e Min</b> Is 3 μs 0 μs 0	Max 1.23 min 28.24 sec 3.15M				
esource C Overall Resor Resource Duration CPU Reads Writes	Total 26.58 min 18.01 min 119.57M 141.1K	Average 13.78 m 9 ms 1.03K 1.22	<b>e Min</b> is 3 μs 0 μs 0 0	Max 1.23 min 28.24 sec 3.15M 5.16K				
esource C Overall Resource Duration CPU Reads Writes Row Count	Total           26.58 min           18.01 min           119.57M           141.1K           29.65M	Average 13.78 m 9 ms 1.03K 1.22 256.14	e Min Is 3 μs 0 μs 0 0 0	Max 1.23 min 28.24 sec 3.15M 5.16K 2.31M				
Coverall Resource Resource Duration CPU Reads Writes Row Count Top Consumi	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M ing Batches	Average 13.78 m 9 ms 1.03K 1.22 256.14	e Min is 3 μs 0 μs 0 0 0	Max 1.23 min 28.24 sec 3.15M 5.16K 2.31M				
Overall Resource Duration CPU Reads Writes Row Count Top Consumi Resource	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M ing Batches Top Consumer	Average 13.78 m 9 ms 1.03K 1.22 256.14	<ul> <li>Min</li> <li>3 μs</li> <li>0 μs</li> <li>0</li> <li>0</li> </ul>	Max 1.23 min 28.24 sec 3.15M 5.16K 2.31M	Average	More Information		
CPU Reads Writes Row Count Top Consumi Resource Duration	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M ing Batches Top Consumer EXEC EDS_SENDEF	Average 13.78 m 9 ms 1.03K 1.22 256.14	e Min Is 3 μs 0 μs 0 0 Event Count 45	Max           1.23 min           28.24 sec           3.15M           5.16K           2.31M	Average 11.09 sec	More Information Show Batches by Duration		
CPU Resource Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M ing Batches Top Consumer EXEC EDS_SENDEF EXEC Admin_Eyeb	Average 13.78 m 9 ms 1.03K 1.22 256.14	<ul> <li>Min</li> <li>3 μs</li> <li>0 μs</li> <li>0 0</li> <li>0</li> </ul>	Max 1.23 min 28.24 sec 3.15M 5.16K 2.31M Total 8.32 min 8 min	Average 11.09 sec 4.85 sec	More Information Show Batches by Duration Show Batches by CPU		
Coverall Resource Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU Reads	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M Top Consumer EXEC Consumer EXEC CADS_SENDEF EXEC CADS_SENDEF EXEC CADS_SENDEF	Average 13.78 m 9 ms 1.03K 1.22 256.14 R_FETCH olasters_C 108	<ul> <li>Min 3 μs 0 μs</li></ul>	Max 1.23 min 28.24 sec 3.15M 5.16K 2.31M Total 8.32 min 8 min 27.31M	Average 11.09 sec 4.85 sec 64.86K	More Information Show Batches by Duration Show Batches by CPU Show Batches by Reads		
CPU Coverall Resource Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU Reads Writes	Insumption 3 Total 2658 min 18.01 min 119.57M 141.1K 29.65M Ing Batches Top Consumer EXEC Admin, Eyeb EXEC SS_SERDEF EXEC Admin, Eyeb EXEC SS_FERCH, EXEC SS_FERC	Average 13.78 m 9 ms 1.03K 1.22 256.14 R_FETCH elasters_C 10B CMD_UF	S Min is 3 μs 0 μs 0 0 0 Event Count 45 99 421 20368	Max           1.23 min           28.24 sec           3.15M           5.16K           2.31M           Total           8.32 min           8 min           27.31M           95.24K	Average 11.09 sec 4.85 sec 64.86K 4.68	More Information Show Batches by Duration Show Batches by CPU Show Batches by Whites		
esource C Verall Resource Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU Reads Writes Row Count	Total 26.58 min 18.01 min 19.57M 14.11K 29.65M ing Batches Top Consumer EXEC EDS_SENDEF EXEC CAS_SENDEF EXEC CAS_SENDEF EXEC CSS_FETCH_J EXEC EDS_SENDEF	Average 13.78 m 9 ms 1.03K 1.22 256.14 R_FETCH Masters_C IOB CMD_UF R_FETCH	<ul> <li>Min</li> <li>3 μs</li> <li>0 μs</li> <li>0 μs</li> <li>0</li> <li>0</li> </ul>	Max           1.23 min           28.24 sec           3.15M           5.16K           2.31M         Total       8 min           27.31M           95.24K           24.02M	Average 11.09 sec 4.85 sec 64.86K 4.68 533.78K	More Information Show Batches by Duration Show Batches by CPU Show Batches by Writes Show Batches by Writes Show Batches by Now Count		
esource C Verall Resource Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU Reads Writes Row Count Top Consumi	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M IND Batches Top Consumer EXEC ADS_SENDEF EXEC ADS_SENDEF EXEC ADS_SENDEF EXEC ADS_SENDEF EXEC ADS_SENDEF EXEC ADS_SENDEF	Average 13.78 m 9 ms 1.03K 1.22 256.14 R_FETCH iloba iloba R_FETCH iloba R_FETCH iloba R_FETCH ilobaa ilobaa il	<ul> <li>Min</li> <li>3 μs</li> <li>0 μs</li> <li>0 0</li> <li>0</li> </ul>	Max           1.23 min           28.24 sec           3.15M           5.16K           2.31M           Total           8 nin           27.31M           95.24K           24.02M	Average 11.09 sec 4.85 sec 64.86K 4.68 533.78K	More Information Show Batches by Duration Show Batches by CPU Show Batches by Reads Show Batches by Writes Show Batches by Row Count		
esource C Verall Resource Duration CPU Reads Writes Row Count Top Consumi CPU Reads Writes Row Count Top Consumi Resource Resource	Total 26.58 min 18.01 min 119.57M 141.1K 29.65M Ing Batches Top Consumer EXEC ADMIN.Symbol EXEC ADS_SENDEI EXEC ADS_SENDEI EXEC ADS_SENDEI Ing Batch Instance Top Consumer	Average 13.78 m 9 ms 1.03K 1.22 256.14 R_FETCH Values CMD_UF R_FETCH s	s         Min           is         3 μs         0 μs         0           0         0         0         0           203068         421         20368         45           Event Count         45         45         45	Max           1.23 min           28.24 sec           3.15M           5.16K           2.31M         Total           8.32 min           8 min           27.31M           95.24K           24.02M	Average 11.09 sec 4.85 sec 64.86K 4.68 533.78K Average	More Information Show Batches by Duration Show Batches by CPU Show Batches by Reads Show Batches by Reads Show Batches by Now Count		
esource C Verall Resor Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU Reads Writes Row Count Top Consumi Resource Duration CPU Reads CPU Reads CPU Reads CPU Reads CPU Resource CPU Reads CPU Reads CPU Reads CPU Reads CPU Resource CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU CPU Reads CPU CPU Reads CPU CPU Reads CPU CPU Reads CPU CPU CPU Reads CPU CPU Reads CPU CPU Reads CPU CPU Reads CPU CPU Reads CPU CPU Reads CPU Reads CPU Reads CPU CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU Reads CPU CPU Reads CPU CPU CPU CPU CPU CPU CPU CPU	Total 26.58 min 18.01 min 19.57M 141.1K 29.65M Top Consumer EXEC DS_SENDEF EXEC Admin.Eyeb EXEC Admin.Eyeb EXEC Admin.Eyeb EXEC Admin.Eyeb EXEC SS_FETCH_J Inst Batch Instance Top Consumer EXEC ES_SENDEF	Average 13.78 m 9 ms 1.03K 1.22 256.14 R_FETCH CMD_UF CMD_UF S R_FETCH	<ul> <li>Min</li> <li>3 μs</li> <li>0 μs</li> <li>0</li> <li>0</li> </ul> Event Count 45 20368 45 Event Count	Max           1.23 min           28.24 sec           3.15M           5.16K           2.31M         Total           8.32 min           8 min           27.31M           95.24K           24.02M	Average           11.09 sec           4.85 sec           64.86K           4.68           533.78K	More Information Show Batches by Duration Show Batches by CPU Show Batches by Reads Show Batches by Writes Show Batches by Row Count More Information Show Batch Instances by Duration		

#### Figure 14 – Workload Summary

The *Summary Report* offers a one-page high-level statistical analysis of the workload. It contains the following items:

# 5.1.1 Analysis Information

- **Workload Time Range** the times of the first and last events from all traces. This information is available only if you selected the *Analyze Execution Times* option during the workload analysis setup.
- **Total Number of Events** the summed count of all events found in all of your traces.
- **Location** the storage location for the results of this workload analysis.
- **Workload Traces** the full list of the traces that comprise this workload.

# 5.1.2 Resource Consumption Statistics

All *Consumption Statistics* list the following five resources:

• **Duration** - the total time between the start and end time of an event. Units of measure for duration are:

٠	day	= Days	• sec	= Seconds
•	hr	= Hours	• ms	= Milliseconds

- **min** = Minutes **µs** = Microseconds
- **CPU** the time spent by the CPU to execute an event.
- **Reads** the number of logical page reads. Units of measure for reads, writes, and row counts are:
  - **T** = Trillion (Tera) **M** = Million (Mega)
  - **G** = Billion (Giga) **K** = Thousand (Kilo)
- Writes the number of page writes.
- **Row Count** the number of rows affected by the event.

#### **Overall Resource Consumption**

- **Total** the sum of the measures of all of the events in the workload.
- **Average** the average measure for the entire workload.
- **Min** the lowest measure found on any process in the workload for this resource.
- **Max** the highest measure found on any process in the workload for this resource.

#### **Top Consuming Batches**

The term *Batch* is explained in detail in section 1.3 - Glossary of Terms.

- **Top Consumer** the Batch which consumed the highest amount of the resource within the workload. Recall that a Batch may represent one SQL execution or millions, so high resource consumption is not necessarily an indication of a poor performer. It could simply mean that this is a very commonly used SQL batch within the workload.
- **Event Count** the number of times this Batch was executed. Each event is a copy of the Batch, populated with specific parameter values. The different events can have different parameter values.

- **Total** the consumption statistic for this one *Batch* (which totals the measurements of all events that make up this *Batch*).
- Average the average consumption statistic for this *Batch*.
- **More Info** click the **Show...** link to see a list of all batches sorted from highest to lowest by one of the resource measures. Any time you click one of the **Show...** links, Qure Analyzer will take you to the *Details Report*, and show you the summed resource statistics for all events in the workload, grouped by that dimension, and sorted by a resource, such as by CPU. Once in the *Details Report*, you can group the events differently, filter the events (or groups of events) differently, or sort the groups of events differently. See section 5.2 Using the Workload Details Report.

#### **Top Consuming Batch Instances**

The term *Batch Instance* is explained in the section 1.3 - Glossary of Terms. The interpretation of the statistics in this section is similar to that of the above section, *Top Consuming Batches*.

#### **Top Consuming Traces**

If your workload is made up of multiple trace sources, this section identifies the trace which measures the highest on each particular resource. The interpretation of the statistics in this section is similar to that of the section *Top Consuming Batches*. Similarly, *Top consuming...* sections are provided for each available workload dimension.

# 5.2 Using the Workload Details Report

When analyzing a single workload, the *Workload Details Report* is where you will probably spend most of your time in Qure Analyzer. It provides the insight to what is going on within your workloads.

This sample table will be used to illustrate the behavior of the controls in the *Workload View Pane*. Imagine that this table represents all the events found in the workload. A real workload, of course, could have millions of events, and potentially many more data columns. A real workload would probably include additional metrics besides duration such as CPU, reads and writes.

Batch	Batch Instance	Trace Source	Start Time	Duration
SELECT * FROM TBL WHERE ID = <value></value>	SELECT * FROM TBL WHERE ID = 14	Month End	14:00	3 sec
SELECT * FROM TBL WHERE ID = <value></value>	SELECT * FROM TBL WHERE ID = 14	Month End	21:00	4 sec
SELECT * FROM TBL WHERE ID = <value></value>	SELECT * FROM TBL WHERE ID = 14	Mid-Month	04:00	5 sec
SELECT * FROM TBL WHERE ID = <value></value>	SELECT * FROM TBL WHERE ID = 26	Month End	01:00	6 sec
SELECT * FROM TBL WHERE ID = <value></value>	SELECT * FROM TBL WHERE ID = 26	Month End	14:00	7 sec
EXEC SP_unload <value></value>	EXEC SP_unload 'Account R'	Mid-Month	15:00	10 sec
EXEC SP_unload <value></value>	EXEC SP_unload 'Account R'	Weekend	13:00	20 sec
EXEC SP_unload <value></value>	EXEC SP_unload 'Account S'	Weekend	02:00	30 sec
EXEC SP_unload <value></value>	EXEC SP_unload 'Account T'	Weekend	05:00	40 sec
EXEC SP unload <value></value>	EXEC SP unload 'Account T'	Weekend	23:00	50 sec

Figure 15 – Sample Workload, All Events, All Columns

# 5.2.1 Controlling the Workload View

The *Workload View Pane* (see Figure 16) allows you to focus your attention on a particular subset of the events in the workload. It allows you to:

- Save and load your preferred settings (Save\Load Settings)
- Group the events on the grid (*Group by*)
- Filter events (or groups of events) to include in the analysis (*Filter By*)
- Switch between totals or averages (*Resources Aggregation Mode*)



**Note**: Note: to adjust the sort order, see section 5.2.4 - Workload Grid.

Workload View	🔺 🦘 🥐 🕂
<ul> <li>Save\Load Settings</li> </ul>	
← Group By	
+ Batch + +	<b>∦ ₩ ×</b>
+ Batch Instance • +	â 🎚 🗙
	Apply
<ul> <li>▲ Filter By</li> <li>Add Filter ▼</li> </ul>	
<ul> <li>▲ Resource Aggregation Mode</li> <li>⑥ Total</li> <li>⑦ Average</li> </ul>	

#### Figure 16 – Workload View Pane

#### Using the Save\Load Settings Control

The *Save\Load Settings* control allows you to quickly manage your preferred workload views. Configure the workload view settings as explained below to your preferred settings of grouping and sorting. Expand the *Save\Load Settings* control which is minimized by default, type a name for your configuration and click **Save**.

To select a previously saved configuration use the drop down list box, select your saved configuration and click **Load**.



**Note**: Only compatible configurations will be shown for selection. For example, a saved configuration that uses the application dimension will not be available to a workload analysis where the application dimension is not available.

To delete a previously saved configuration, select it from the drop down list box and click the red X (delete) button.

If you have multiple workload analyses open in Qure Analyzer, and you save your settings on one while the others are still open, your newly saved configuration will not be available on all other tabs until you click the **Refresh** button on the left side of the text box.



**Note**: saved *Workload View* configurations are user-specific and will not be available for other users using the same PC.

#### Using the Group by Control

The *Group by* control allows you to determine the groupings of the workload events. In this way, the groupings are similar in behavior to the SQL *GROUP BY* clause. These controls also determine which trace data columns will be used for grouping and appear in the grid as the grouping dimensions.

To illustrate the effect of the groupings, let's suppose we want to group all events in the sample workload first by *Batch*, and then by *Batch Instance* which happens to be the default grouping setting). The data on the grid would look like this (conceptually of course):

Batch		Duration	Event Count
SELECT * FROM TBL WHERE ID = <value></value>		25 sec	5
	Batch Instance	Duration	Event Count
SELECT * FROM TBL WHERE ID = 14		12 sec	3
	SELECT * FROM TBL WHERE ID = 26	13 sec	2

Batch		Duration	Event Count
EXEC SP_unload <value:< td=""><td>&gt;</td><td>150 sec</td><td>5</td></value:<>	>	150 sec	5
	Batch Instance	Duration	Event Count
EXEC SP_unload 'Account R'		30 sec	2
EXEC SP_unload 'Account S'		30 sec	1
	EXEC SP_unload 'Account T'	90 sec	2

#### Figure 17 – Conceptual Groupings Diagram

The above figure does not show the *Analysis Results Grid* as actually seen in Qure Analyzer. It is a conceptual view to show the relationship between each *Batch* and its child *Batch Instances*, whose durations sum to determine the duration of the *Batch*.

The duration represents the summed duration of all the events that share the specified group property. For example, the grouping of *Batch Instance* has a row whose property is the TextData SELECT \* FROM TBL WHERE ID = 14. There were 3 events in the workload that shared this characteristic. When the durations of those 3 events were summed, it was found to be 12 seconds. Therefore, the parent group *Batch* displays a total duration of 12 seconds.

To carry this example forward, Figure 18 shows what the *Group by* control would need to look like in order to make the grid contain the groupings shown above.

▲ Group By	
* Batch * *	1 V X
-	
+ Batch Instance - +	û 🕸 🗙
	Apply

#### **Figure 18 – Default Group by Settings**

The selections shown in this *Group by* control happen to be the default selections for all analyses in which the *Save Individual Batch Parameters* option has been selected. Of course you can adjust it at will.

When a new grouping is first applied, only the rows for the top-level grouping appear in the *Analysis Results Grid*. For example, when the highest level *Group by* setting is changed to *Application* as shown in Figure 19, the *Application* column would then appear first in the grid. All resource columns always appear to the right of the grouping columns.

Workload View	♠ *> <> ₽		Application	Duration	·	CPU		Reads	
Save\Load Settings		۰	.Net SqlClient Data Provider		12.37 min		5.28 min		57.07M
(		۰	AdminWeb		3.63 min		8.46 min		26.16M
▲ Group By		۰	EDSServe Application		3.39 min	1.00	23.44 sec	1	2.26M
		۰	Internet Information Services		3.02 min	•	1.34 min		14.1M
+ Application + +	<b>* * *</b>	۰	Microsoft SQL Server		2.86 min		1.75 min	•	8.66M
+ Batch • +		٠	AnalyticsWeb	1.00	30.24 sec	1	15.08 sec		10.17M
		٠	ACM	1	15.06 sec	1	5.66 sec	1	352.83K
+ Batch Instance	• • 👔 🕸 🗙	٠	ApplicationServicesSI	1	10.03 sec	1	5.85 sec	1	26.92K
-		٠	Microsoft SQL Server Manage	1	8.73 sec	1	10.76 sec	1	183.38K
	Apply	۰	CustomReportsSI	1	4.68 sec	1	3.51 sec	1	405.54K
+ Filter Du		۰	SQLAgent - TSQL JobStep (Job	1	2.99 sec	1	2.89 sec	1	9.1K
Add Filter =		۰	Microsoft SQL Server Manage	1	2.85 sec	1	281 ms	1	9.23K
Add Filler		-	and the second second second second		122				Proc. 224

#### Figure 19 – Customized Group by Settings

If the top-level grouping had been the pair of columns *Application* and *Host*, then the only two dimension columns that would initially appear in the grid would be those, as shown in Figure 20.

Norkload View 🏫 🖘 🕐 📮		Application	Host	Duration	•	CPU		Reads	4
Save\l oad Settings	۰	.Net SqlClient Data Provider	EDS01	-	8.39 min		2.47 min		7.52M
		EDSServe Application	EDS01	-	3.39 min	1.00	23.44 sec	1.1	2.26M
▲ Group By	۰	AdminWeb	ADMIN04	-	1.93 min		4.83 min		13.77M
	٠	AdminWeb	ADMIN03	•	1.69 min		3.63 min		12.39M
+ Application • + Host • + • • • •	٠	Internet Information Services	ADMIN03	•	1.58 min	•	34.15 sec		6.42M
+ Batch • +	٠	.Net SqlClient Data Provider	VEW02	•	1.52 min		1.06 min		13.75M
•	۰	.Net SqlClient Data Provider	VEW01	•	1.52 min		1.08 min		14.05M
+ Batch Instance - + 👔 🕸 🗙	۰	Microsoft SQL Server	CRM4DB	•	1.43 min		1.27 min		8.48M
-	۰	Microsoft SQL Server	RSDWH01	•	1.42 min	•	28.72 sec	1	154.42K
Apply	٠	Internet Information Services	ADMIN04	•	1.37 min		40.21 sec		7.21M
	٠	.Net SqlClient Data Provider	RPSVC01	1 - C	48.92 sec		37.87 sec		21.71M <b>4</b>
A Filter By	۰	AnalyticsWeb	ADMIN03	1	17.14 sec	1	9.11 sec		5.76M
Add Hiter	here	An al sine of the second		a prof	-	1.4	5 marco		din.

Figure 20 – One Grouping Level with Two Columns

In order to view or hide the next lower grouping for one of the rows, click the *Expand/Collapse* control. See the red highlight in Figure 21 – Expanding one Grouping Level below. In that figure, the *Expand/Collapse* control has been clicked on the fourth *Application/Host* grouping level, causing the first ten *Batch* rows associated within that *Application* and *Host* to be displayed.

Workload View	<b>•</b> • • 4		Application	Host		Duration		CPU		Reads	
<ul> <li>Save\Load Settings</li> </ul>		۰	.Net SqlClient Data Provider	EDS01		8.3	9 min		2.47 min		7.52N
g_		۰	EDSServe Application	EDS01		3.3	9 min		23.44 sec	1.00	2.26N
▲ Group By		-	AdminWeb	ADMIN04		1.9	3 min		4.83 min		13.77
+ Application + Host +	1 4 X	E	Batch		Dura	ition	CPU		Reads		X
•		×	EXEC Admin_Eyeblasters_Get_	AdsList_Dy		1.55 min		4.6 min		4.24M	
+ Batch - +	1 U X	۰	EXEC Admin_Accounts_Accou	ntsDS_Select	•	10.4 sec	1	8.16 sec		8.14M	I
		×	EXEC Admin_Campaigns_Get_	Campaign		7.54 sec	1	1.97 sec		1.16M	
+ Batch Instance • +		×	EXEC Admin_Flights_Get_Flights	tsList_Dyn	1	2.16 sec	1	733 ms	1	203.298	( )
		۰	EXEC Admin_PLC_QuickPlacer	nentDS_In	I.	962.72 ms	1	16 ms	1	195	
	Apply	×	EXEC BSA_SQLCMD_GET_SESS	SION_DATA	I.	919.52 ms	1	61 ms	1	962	1
▲ Filter By		×	EXEC BSA_SQLCMD_FETCH_A	CCOUNTS	I.	775.57 ms	1	2.66 sec	1	9.34K	ľ
Add Filter 🔻		×	EXEC BSA_SQLCMD_UPDATE_	FLIGHTS_C	I.	133.95 ms	1	0 µs	1	343	1
		×	EXEC Admin_Eyeblasters_Get_	AdsList_Se	I.	73.86 ms	1	77 ms	1	435	1
<ul> <li>Resource Aggregation Mode</li> </ul>		×	EXEC BSA_SQLCMD_SET_FLIG	HT_SERVIN	I.	54.55 ms	I.	0 µs	1	18	I
<ul> <li>Total</li> <li>Average</li> </ul>						<b>₩</b> M	ore				_

Figure 21 – Expanding one Grouping Level

In order to see an additional ten *Batch* rows within that grouping, click the **More** button at the bottom of the *Batches* list. Each click of this bar will load ten additional rows.

#### **Configure Multiple Level Groupings**

Figure 22 shows the flexibility of the *Group by* control. It allows you to:

- Establish a hierarchy several grouping levels deep. You can use as many dimensions as you have available in the workload.
- Arrange the groupings in any order.
- Add additional dimensions to a grouping level.

Workload View 🏫 🦘 🥐 🎙		Application	Host	Duration	CPU	Reads
<ul> <li>Save\Load Settings</li> </ul>	۰	.Net SqlClient Data Provider	EDS01	8.39 min	2.47 min	7.5
,	۰	EDSServe Application	EDS01	3.39 min	23.44 sec	2.26
▲ Group By		AdminWeb	ADMIN04	1.93 min	4.83 min	13.7
+ Application • + Host • +		Batch	Dur	ation CPU	Reads	
•		EXEC Admin_Eyeblasters_Get_/	AdsList_Dy	1.55 min	4.6 min	4.24M
+ Batch + + + + X		Batch Instance (TextData)	Duration	CPU	Reads	Writes
+ Batch Instance - + 👔 🖡 🗙		EXEC Admin_Eyeblasters_Get	9.5.	2 sec 26.5	58 sec 2.45N	
+	Ш	Hour	Duration	CPU	Reads	Writes R
		7:00 - 8:00	9.52 se	26.58 se	c 2.45M	1
Apply	Ш	EXEC Admin_Eyeblasters_Get	7.4	sec 22.2	28 sec 📔 117.6	5K
		EXEC Admin_Eyeblasters_Get	4.5	6 sec 14.1	L4 sec 178.5	7K
▲ Filter By		EXEC Admin_Eyeblasters_Get	4.4	2 sec 14.5	i6 sec   53.41	к і
Add Filter	<u>L</u> .	THE Anton Such		- 14.4	محمر <b>مسر ا محمد</b>	in manager

#### Figure 22 – A Four-Level Grouping Hierarchy

One row from each grouping level is shown expanded so as to reveal how the *Group by* control has adjusted the grid's column headers and rows to your preference.



**Note**: Only one sub-grouping level can be shown at a time for each parent grouping.

#### Adding a Top Level Grouping

Presume that that workload you analyzed contains events processed by four different *Applications*. To group by *Application* first, and then by *Batch*, and then by *Batch Instance*, follow these steps.



**Note**: These instructions work to insert any new grouping levels above existing groupings in the *Group by* control.

1 Click the top **Add Grouping Level** button (+).

Workload View 🔒 🕈 🍖 🖡
▼ Save\Load Settings
▲ Group By Add Grouping Level
+ Batch Instance - + 👔 🖡 🗙
*
Apply

Figure 23 - Add Top Grouping Level

2 Click Application or any dimension of your choice.

Wor	kload View	▲ う ぐ 早						
▼ Save\Load Settings								
-	Group By							
	Batch Instance	1 U X						
	Application							
	Host	• • 👔 🖡 🗙						
	User							
	Ever Class	April						

#### Figure 24 - Add Application as Top Level Grouping

**3** A grouping level by *Application* is added at the top level of the *Group by* hierarchy.

Workload View 🏫 🦘 🥐 🗜
▼ Save\Load Settings
• Batch • • 1 • X
Apply

**Figure 25 - Multiple Level Grouping Set** 

4 Click **Apply**. The grid will refresh and regroup the events so that those events that originated from the same application will be grouped together. Under *Application* will be *Batch* (not shown; collapsed), and under that, *Batch Instances* (not shown; collapsed).



Figure 26 - Multiple Level Grouping Grid View

#### Adding a Middle Level Grouping

To add a grouping between any two existing grouping levels in the *Group by* hierarchy, click an **Add Grouping Level** button between any two existing grouping levels.

Workload View	ŧ	•	¢	<b>д</b>
<ul> <li>Save\Load Settings</li> </ul>				
▲ Group By				
+ Application - +		î.	î X	•
Add Grouping Level		î ·	u ×	•
+ Batch Instance - +		î,	U X	•
*				
		A	ply	

Figure 27 - Add Mid-level Grouping

The rest of the steps are identical to the instructions for adding a higher level grouping as described above.

#### Adding a Grouping Column to a Level

One level of grouping can have two or more data columns. You can add new grouping columns wherever you see an **Add Grouping Column** button.

Workload View 🏫 🦘 🥐 🖡
▼ Save\Load Settings
▲ Group By
+ Application 👻 🛧 👔 🕸 🗶
+ Batch + +
+ Batch Instance 🔹 + 👔 🎚 🗙
Apply

#### Figure 28 – Add a grouping to the right

This way, you can also add a grouping between two side-by-side grouping dimensions on the same level, or to their left.



**Note**: Depending on your screen resolution and the sizing of the different panes, you may need to scroll the button for the *Add Grouping Column* icon to become fully visible.

#### Adding a Low Level Grouping

Use the same approach as explained above. Click the **Add Grouping Level** button on the very bottom of the *Group by* control.

#### Moving or Removing a Grouping Level

To remove one of the grouping levels, click the **Remove Grouping Level** (X) button to the right of the grouping.

To move one of the grouping levels up one level, click the **Move Up** (upward arrow) button to the right of the grouping level bar. To move one of the grouping levels down one level, click the **Move Down** (downward arrow) button to the right of the grouping level bar.

#### Changing a Grouping Column

To edit an existing grouping column (e.g. change *Application* to *Database*), click the grouping column to be changed. A list of available data columns appears for you to select from. Select one with a click.



#### Figure 29 - Change Grouping Columns In-place



**Note**: After all your changes to the *Group by* control are complete, you must click the **Apply** button to refresh the *Analysis Results Grid* to reflect your new settings.

#### **Using the Filter Control**

Qure Analyzer's powerful filtering capability allows you to narrowly target specific events or groups of events. For example, you may focus on events belonging to specifically named *Batches* when they run at specific times of day, on events for processes that take more than a certain number of seconds of CPU on a specific *Application*, or on events representing queries which certain users employed to retrieve very large numbers of rows. The flexibility of filtering is limited only by the types of data available within the workload.

The **Add Filter** drop-down will list every trace dimension data column found in the traces, with one exception. If a data column's value on all events shows no variation, then filtering on that data column will be meaningless, and therefore it will not be available. For example, if all events in your workload originated from only one host, then no filtering on host will be available.

Here are some common values you may see in the Filter list:

- Batches Users
- Applications
   Event Classes
- Hosts
   Hours
- Databases
   Start Time

In addition, the following Resource columns will appear in the **Add Filter** drop-down list.

- Duration
   Writes
- CPU
   Row Count
- Reads

After you've filtered down the events to those you are interested in analyzing, those events will be grouped according to your *Group by* preferences, aggregated according to your *Resources Aggregation Mode* preference, and sorted according to the current sorting column on the grid.

#### 5.2.2 Filtering

There are three types of filter controls: *Text Search, Discrete Values* and *Range Controls*. Below you will find one example of each type of filter type.

#### Filter by Batches (Free Text Filter)

This filter allows you to view only the *Batches* that contain certain text in their names or in their contents. For example, you may want to find all events whose *Batches* contain the text "*Update*".

- 1 Click the Add Filter button. The list of filter options appears.
- 2 Move the mouse cursor to *Batches*. The search text box appears.
- **3** Type or paste the text you'd like to find in any *Batch*.

-	Filter By Add Filter 💌	Internet Information     Microsoft SQL S
	Batches +	Include Batches With the Following Text:
-	Applications +	Lindate
	Hosts 🕨	opulic
	Users +	Apply Cancel
	Event Classes 🔸	Coldmant TSC
h	Hours	Microst S

#### Figure 30 - Setting the Free Text Filter

**4** Click **Apply**. The grid will refresh and display all *Batches* that contain the search text anywhere within the batch name or body or sample *Batch Instance* body, including variable and constant names.

In Figure 31 below, Qure Analyzer has found several *Batches* that have the term "*Update"* in their names.

Workload View 🔒 🦘 🕐 📮	_	Batch	Duration	•	CPU	-
<ul> <li>Save\Load Settings</li> </ul>	٠	EXEC EDS_SENDER_FETCH_PENDING_RESOURCE_UPDATE		8.32 min		2.46 mi
	۰	EDS_ENGINE_SQLCMD_UPDATE_LIGHT_IMPRESSIONS		2.48 min	1.00	15.67 sec
▼ Group By	٠	EDS_ENGINE_SQLCMP_UPDATE_EYEBLASTER_IMPRESSIONS	1. Alton	50.09 sec	1	5.72 sec
	٠	EXEC BSA_SQLCMD_INSERT_NEW_EYEBLATER_INTERACTIONS	1.00	34.24 sec	1	2.3 sec
▲ Filter By	۰	SELECT ContentID As <value> , ContentName As <value> , C</value></value>	1	3.68 sec	1	1.92 sec
Batches Update" ×	٠	EXEC EDS_SENDER_SET_RESOURCE UPDATE TO_COMPLETED	1	3.39 sec	1	78 ms
Add Eilmer	-	the Automation of the Automation of Street and Street			1	

#### Figure 31 – Free Text Filter Applied

Qure Analyzer also finds *Batches* that contain the desired search text within their bodies. For example, see Figure 32 – Free Text Filters within Batch Text below.

Workload View 🏫 🖘 🍖 📮		Batch	Duration		CPU	
Save\Load Settings	۰	EXEC EDS_SENDER_FETCH_PENDING_RESOURCE_UPDATES		8.32 min		2.46 min
	۰	EDS_ENGINE_SQLCMD_UPDATE_FLIGHT_IMPRESSIONS		2.48 min	1 - C	15.67 sec
▼ Group By	۰	EDS_ENGINE_SQLCMD_UPDATE_EYEBLASTER_IMPRESSIONS	1. A.	50.09 sec	1.00	5.72 sec
	۰	EXEC BSA_SQLCMD_INSERT_NEW_EYEBLATER_INTERACTIONS	1.00	34.24 sec	1	2.3 sec
▲ Filter By	٠	SELECT ContentID As <value> , ContentName As <value> , C</value></value>	1	3.68 sec	1	1.92 sec
Batches (Update) ×	٠	EXEC EDS SENDER SET RESOURCE UPDATE TO COMPLETED	1	3.39 sec		78 ms
Add Filter 🔻	Bato	h				
Resource Aggregation Mode     Otal     Average	SELE Cont Last Vide (Acc	<pre>CT ContentID As <value> , ContentName As <value> , C representer As <value> , FileVersjon As <value> , File Value&gt; , ContentFileName As <value> , C value&gt; , ContentFileName As <value> , C value&gt; , Calue&gt; , ExistsInPackage As <value> , FRC countID = ,Value&gt; ) AND (contentIume &lt;&gt; <value> ) AND ( proc</value></value></value></value></value></value></value></value></pre>	ontentTyp Size As < ntentType M BSA_CRE ContentTy	e As <val Value&gt; , As <valu ATIVE_ASS pe &lt;&gt; <va< td=""><td>ue&gt; , Cont Formatted e&gt; , Orig: ETS_VIEW N lue&gt;) AND</td><td>tentFolde Descripti inalFileS VITH(NOLC ContentF</td></va<></valu </val 	ue> , Cont Formatted e> , Orig: ETS_VIEW N lue>) AND	tentFolde Descripti inalFileS VITH(NOLC ContentF





**Feature**: Filters are applied for data columns, even if the column being filtered is not visible in the grid.

#### Filter by Applications (Discrete Values Filter)

This filter allows you to view only the events associated with the Applications you are interested in. For example, your workload may contain events from multiple applications, as seen in Figure 33.

Workload View	🛖 🦘 🥐 🖟		Application	Duration	*	CPU		Reads		Writes	Í
▼ Save\Load Settings		٠	.Net SqlClient Data Provider		12.37 min		5.28 min		57.07M	1	Ý
J J L3		٠	AdminWeb		3.63 min		8.46 min		26.16M	1	₹
← Group By		٠	EDSServe Application		3.39 min	1.00	23.44 sec	1	2.26M		₹
		٠	Internet Information Services		3.02 min	•	1.34 min		14.1M	$\mathbf{I}_{i} = \mathbf{I}_{i}$	9.5
+ Application + +	<b>* • •</b>	۰	Microsoft SQL Server		2.86 min	•	1.75 min	•	8.66M	1	X
+ Batch + +	1 U X	۰	AnalyticsWeb	1	30.24 sec	1	15.08 sec	•	10.17M	1	$\geq$
		۰	ACM	1	15.06 sec	1	5.66 sec	1	352.83K	1	ς.
+ Batch Instance - +	• 👔 🖡 🗙	۰	ApplicationServicesSI	1	10.03 sec	1	5.85 sec	1	26.92K	I.	¥.
		۰	Microsoft SQL Server Manage	1	8.73 sec	1	10.76 sec	1	183.38K	I.	5
	Apply	٠	CustomReportsSI	1	4.68 sec		3.51 sec	1	405.54K	I	Ś
	and a state of the		and the second second	-	have V					-	r -

#### Figure 33 – Multiple Application Workload View

You want to only examine the stats for events that were processed through the "Adminweb" application.

- 1 Click the **Add Filter** button. The list of filter options appears.
- **2** Move the mouse to *Applications*. A sub-menu appears listing the names of every application represented in your workload. By default, all are selected.

		<ul> <li>Internet Information Services</li> <li>3.02 min</li> </ul>
^	Filter By	Include the Following Applications:
	Batches +	EDSServe Application  K.Net SolClient Data Provider
-	Applications 🕨	ACM T
(	Hosts +	AdminSI AdminWeb
	Users +	AnalyticsWeb
	Event Classes	ApplicationServicesSI CustomReportsSI
	Hours •	✓ DashboardSI
	Start Time	<ul> <li>✓ Eyeblaster DCLPApp</li> <li>✓ GeneralWeb</li> <li>✓ ImportCampaign</li> </ul>
	Duration •	Internet Information Services
~	CPU	✓ IOBatch ✓ Microsoft SOL Server

#### Figure 34 – Setting the Application Filter

**3** Deselect the *Applications* you want to filter out or use the shortcut buttons at the bottom of the applications list.



**Shortcut**: The filter window includes three buttons labeled V, X and I. Use these buttons to speed up the selection process:

- V = Select all
- $\mathbf{X} = \text{Deselect all}$
- **I** = Invert selection
- 4 Click **Apply**. The grid will refresh and display only those groups that contain at least one event associated with the application that you selected.

Workload View 🔒 🕈 🥐 👎			Application	Duration		CPU	Reads		Write	s
▼ Save\Load Settings	L	-	AdminWeb	3.6	i3 min	8.46 mir	n 🗖	26.16M	-	34
			Batch		Durati	on	CPU		Reads	
<ul> <li>Group By</li> </ul>		٠	EXEC Admin_Eyeblasters_Get_Ad	dsList_Dyna		2.84 min		8 min	Writes Reads 6.25 17.4 2.11 3032 1.29 1.29 1.29 1.29 1.29 1.29 1.29 1.2	6.25
← Filter By ×		۰	EXEC Admin_Accounts_Accounts	sDS_Select		22.23 sec	1	17.77 sec		17.45
Applications (1 Included)		۰	EXEC Admin_Campaigns_Get_Ca	ImpaignsList	•	16.39 sec	1	3.89 sec	•	2.110
Add Filter 🔻	L	٠	EXEC Admin_Flights_Get_Flights	List_Dynami	1	3.58 sec	1	1.28 sec	1	303.9
		۰	EXEC BSA_SQLCMD_GET_SESSIO	N_DATA	1	1.77 sec	1	107 m93	1	1.89K
<ul> <li>Resource Aggregation Mode</li> </ul>		۰	EXEC BSA_SQLCMD_FETCH_ACC	OUNTS_BY	1	1.22 sec	1	4.06 sec	1	17.94
Total		۰	EXEC Admin_PLC_QuickPlaceme	ntDS_Insert	1	1.05 sec	1	32 ms	1	584
○ Average		۰	EXEC Admin_Eyeblasters_Get_Ad	dsList_Search	1	284.32 ms	1	201 ms	1	1.32K
		۰	EXEC BSA_SQLCMD_UPDATE_FL	IGHTS_COD	1	157.17 ms	1	16 ms	1	551
And the second design of the second		+7	and the second s	Children an	10-4		-	47.ms		

#### Figure 35 – Application Filter Applied

#### Filter by Duration (Range Filter)

This filter allows you to find events or groups whose duration is between two times.



**Note**: The maximum values shown on the range controls are the maximum value found in the entire workload, not just in the current set of data visible on the grid. So it's possible that when you ask to retain only those events with *Duration* greater than say, 1 minute, you may get no results at all if a second filter has already removed all of those events from view.

1 Click the **Add Filter** button. The list of filter options appears.

**2** Move the mouse to **Duration**. A sub-menu appears that allows you to specify the minimum and maximum durations to be included.

	Filter by Event / Group Duration
<ul> <li>▲ Filter By</li> <li>Add Filter ▼</li> </ul>	Event Duration
Batches  Applications	Between 7 sec  and 1.23 min
Hosts	Group Duration
Event Classes	Between 3 $\mu s$ and 26.58 min $\checkmark$
Hours	Note that group filters are applied at all grouping levels
Duration	Apply Cancel
	the and and a second

#### Figure 36 - Setting Range Filters

- **3** Slide the bookend controls to the desired value. The unit and numerical values dynamically adjust as you slide. Alternatively, you may enter specific values for the ranges by using the designated text boxes and the pull down controls for selecting the desired units.
- **4** *Event Duration* refers to the duration of the individual events in the workload. So if you'd like to analyze only those events that required more than a minute to process, use this control to restrict the minimum duration to one minute. Once selected, they will be grouped according to your *Group by* preferences.
- **5** *Group Duration* refers to the total (or average) duration of all events contained in one grouping level. One group filter applies to all groups at all levels of the *Group by* hierarchy.
- 6 Click **Apply**. The grid will refresh and retain only those rows (groupings at any level) whose durations match the specified range.



**Note**: The difference between *Event Duration* and *Group Duration* filters is similar to the difference between the WHERE and HAVING clauses of a GROUP BY query. In fact, they translate internally to exactly these clauses for the query that retrieves the data.

# Filtering Example 1 – Examine Groups whose total duration exceeds 1 minute

In this example, the highest level of the *Group by* hierarchy is the combination of the *Application* and *Hour* dimensions. All events which were executed both within this *Application* and within this *Hour* are grouped together. Their individual durations are summed, and their summed values are shown in the grouped row in the *Analysis Grid*. The summed duration of the row highlighted below (.Net SqlClient Data Provider between 06:00 and 07:00) is 3.1 minutes.

Workload View 🏫 🦘 🥐 👎	-	Application	Hour	Duration	CPU	Reads
<ul> <li>Save\Load Settings</li> </ul>	۰	.Net SqlClient Data Provider	7:00 - 8:00	9,27 min	3.94 min	44.7
	۰	.Net SqlClient Data Provider	6:00 - 7:00	3.1 min	1.33 min	12
▲ Group By	۰	EDSServe Application	7:00 - 8:00	2.93 min	19 sec	1.7
	۰	AdminWeb	7:00 - 8:00	2.93 min	6.41 min	22.8
+ Application + Hour + T + X	۰	Internet Information Services	7:00 - 8:00	2.75 min	1.19 min	12
+ Batch Instance • + 1	۰	Microsoft SQL Server	7:00 - 8:00	2.38 min	1.32 min	5.79
-	۰	AdminWeb	6:00 - 7:00	41.84 sec	2.06 min	3.35
integit, de		Microsoft SOL Server	6:02 7:00	28.29 sec	25.81 sec	2

#### Figure 37 – Filtering Example 1 – Top Applications by Duration

To narrow down to only those groups whose total duration exceeds 1 minute, add a filter as shown in Figure 38 - Filtering Example 1 - Set Duration Filter below.

Filter by Event / Group Duration
Event Duration
Between 3 µs • and 1.23 min •
Group Duration
Between 1 min • and 26.58 min •
Note that group filters are applied at all grouping levels
Apply Cancel

#### Figure 38 - Filtering Example 1 - Set Duration Filter

Click **Apply**. The grid refreshes and updates to show the following data:

Workload View 🏠 🔶 👎		Application	Hour	Duration	CPU	Reads	
<ul> <li>Save\Load Settings</li> </ul>	٠	.Net SqlClient Data Provider	7:00 - 8:00	9.27 min	3.94 min	44.	755
g_	۰	.Net SqlClient Data Provider	6:00 - 7:00	3.1 min	1.33 min	12.	₹
▲ Group By	٠	EDSServe Application	7:00 - 8:00	2.93 min	19 sec	1.7	71
	٠	AdminWeb	7:00 - 8:00	2.93 min	6.41 min	22.	81
+ Application • + Hour • + • • • •	٠	Internet Information Services	7:00 - 8:00	2.75 min	1.19 min	12.	39
+ Batch Instance - + 1	۰	Microsoft SQL Server	7:00 - 8:00	2.38 min	1.32 min	5.7	<pre>t</pre>
-							)
Apply							Ž
▲ Filter By							$\mathbf{b}$
Duration (Limited Range)							5
print Eller to and the second	ľ	and and a set of a set of	A	and a second second	a second of	and wanted	

**Figure 39 - Filtering Example 1 - Filtered View** 

We know that these groups have a total duration exceeding one minute because we can widen the *Duration* column to see the statistics, and each of them exceeds one minute.

**Important!** Group Duration filters affect all group levels. When you drill down to lower levels of groups, only subgroups with a total duration of over one minute are displayed. In some cases, you will click to expand one of the *Application/Host* group rows to find, perhaps with surprise, that there are no *Batch Instance* group rows beneath it. This just means that all of the *Batch Instance* group rows had a total duration of less than a minute, and so are not shown. However, their impact is still felt, because their summed durations totaled to over a minute. See example below.





# Filtering Example 2 – Find Events whose average duration exceeds 2 seconds.

Imagine (it won't be hard) that your support desk gets calls on occasion that at certain times of day on certain applications, the response time is painfully slow, let's say two seconds per click. How do you identify the patterns? Qure Analyzer can find them for you.

In this image, we can see that there is no combination of Application and Hour for which the average duration is greater and 0.5 seconds (the highest is 485.05ms). So far, that's no help.

Workload View 🏠 🔶 👎		Application	Hour	Duration		CPU		Reads	
<ul> <li>Save\Load Settings</li> </ul>	٠	Microsoft SQL Server Manageme	7:00 - 8:00	485.	05 ms	· · · · ·	598 ms		10.19K
	۰	.Net SqlClient Data Provider	6:00 - 7:00	217.	89 ms	•	94 ms		14.42K
▼ Group By	۰	CustomReportsSI	7:00 - 8:00	212.	85 ms		160 ms		18.43K
5	٠	Microsoft SQL Server Manageme	7:00 - 8:00	158.	48 ms	1	16 ms	1	512.8
▲ Filter By	٠	AdminWeb	7:00 - 8:00	136.	85 ms		299 ms		17.77K
Add Filter 🔻	۰	AdminWeb	6:00 - 7:00	120.	23 ms		354 ms		9.62K
Paraurca Aggregation Mode	٠	ApplicationServicesSI	7:00 - 8:00	101.	54 ms	1. Sec. 1	62 ms	1	270.16
	٠	ApplicationServicesSI	6:00 - 7:00	<b>1</b> 00.	4 ms	1 - C	49 ms	1	278.75
Average	۰	Microsoft SQL Server	7:00 - 8:00	92.5	4 ms	1 - C	51 ms		3.75K
	۰	SQLAgent - TSQL JobStep (Job 0	7:00 - 8:00	85.3	8 ms	1 - C	63 ms		11.87
man hand	-			- and		and the second s	-	( and the second	

#### Figure 41 - Filtering Example 2 - Viewing Average Duration

However, using a *Filter*, we can find all groups whose average event duration exceeds two seconds. Note that the *Resource Aggregation Mode* is already set to *Average* in Figure 41 - Filtering Example 2 - Viewing Average Duration above.

Filter by Ever	nt / Group	Duration	ı		
Event Dura	tion				
Between	2	sec 🔻	and	1.23	min 🔻
Group Dur	ation				
Between	3	µs 🔻	and	26.58	min 🔻
Note that	group filte	ers are ap	plied	at all gro	uping levels
				Арр	ly Cancel

#### Figure 42 - Filtering Example 2 - Setting Duration Filter

Click **Apply**. The grid refreshes and updates to show the following data:

Workload View 🏫 🦘 🥐 🤇	- I	Application	Hour		Duration	CPU		Reads	4
<ul> <li>Save\Load Settings</li> </ul>	٠	Microsoft SQL Server	7:00 - 8:00		34.73 sec		19.67 sec		4
		Microsoft SQL Server	6:00 - 7:00		26.19 sec		25.42 sec	Reads	2
▼ Group By		.Net SqlClient Data Provider	6:00 - 7:00		11.29 sec		3.51 sec	1.00	
		.Net SqlClient Data Provider	7:00 - 8:00		11.01 sec		3.19 sec	1	í
▲ Filter By		Internet Information Services	7:00 - 8:00		7.15 sec		2.87 sec		
Add Filter -	e	Batch Instance (TextData)		Dura	tion	CPU		Reads	-
	E	EXEC BSA_SQLCMD_INSERT_NEW_E	YEBLATER_I		25.01 sec		500 ms	1.1	4
<ul> <li>Resource Aggregation Mode</li> </ul>	S	ELECT CampaignID As 'CampaignII	D' , Campaig		17.78 sec	-	12.13 sec		
© Total	5	ELECT IOID As 'ID' , PayingAccount	As 'Paying		7.12 sec	⊢	6.8 sec		
Average	E	EXEC BSA_SQLCMD_INSERT_NEW_E	YEBLATER_I		5.3 sec		328 ms	1	1
	E	EXEC BSA_SQLCMD_FLIGHTS_DATA	_BY_CREATI		2.86 sec		1.78 sec		
		XEC BSA-SOLCMD_FETCH OUIDAT	TED DUDUTE	-	Zara sec	.t.	-31_ms		ſ

Figure 43 - Filtering Example 2 - Filtered Duration

We are now looking at every combination of *Application* and *Hour* that have an average event time over 2 seconds. If we expand any grouping, we see the precise culprits. Every *Batch Instance* whose average duration exceeds 2 seconds is named and ready to be fingerprinted.

#### **Using Multiple Filters**

More than one filter can be used concurrently. Qure Analyzer applies all filter conditions when determining which events or groups to include in the report. Qure Analyzer uses a logical AND condition between the filters.



**Note**: Even though filters generally act upon individual events, the Analysis grid never shows individual events. The lowest level visible in the grid is a grouping of multiple events based on at least one dimension.

#### **Modifying a Filter**

To modify any one existing filter setting, click its name. A dialog will appear to allow you to modify its values.

## 5.2.3 Changing the Resources Aggregation Mode

This control allows you to view statistics by *Total* or by *Average*. For example, if there are 100 executions of a *Batch* in the workload, each with an average of 45 reads, then if you select *Total*, the *Reads* column will show a total of 4500 reads. If you select *Average*, the *Reads* column will show an average of 45 reads.

# 5.2.4 Workload Grid

#### **Resizing the Grid**

There are two ways to make the grid larger or smaller.

#### **1. Move the Resize Bars**

Move the mouse cursor over one of the thick dark-grey bars that divide the panes in the *Details Report*. The cursor will change to a double-headed arrow. Click, hold, drag and release.

#### 2. Auto Hide the other Panes

Both the *Workload View* pane and the *Selection Details* pane offer a push-pin icon that lets you automatically hide a pane, allowing the *Analysis Grid* to grow to its maximum size. When the *Workload View* pane is hidden, its name still appears on a small tab on the left of your window to allow you to view the pane on command. The *Selection Details* pane behaves similarly at the bottom of the window.

When the push-pin icon is in the vertical position, its pane will stay fully open. When the push-pin icon is in the horizontal position, its pane will be automatically hidden after each viewing.

#### Viewing Resource Statistics in the grid

The columns always show the resource consumption bars.

Duration	CPU	Reads	
É		<b>1</b>	
	E. C.	1	

When the columns are wide enough, they also show resource consumption statistics numerically.

Duration	CPU		Read	s
59.03 sec		24.46 sec	1	45,48K
35.58 sec		8.52 sec	1	41.67K

#### **Colors Used on Resource Consumption Bars**



-46.23 sec

Duration

Blue indicates a *Measurement* for one resource. It is only seen when analyzing one workload.

Green indicates an *Improvement* comparing the resource measurement from the Evaluated Workload to the resource measurement of the baseline workload. That is, the Evaluated Workload consumed less of this resource.

Duratio	on
	+32.71 sec

**Red** indicates a *Degradation* comparing the resource measurement from the Evaluated Workload to the resource measurement of the baseline workload. That is, the Evaluated Workload consumed more of this resource.

#### **Repositioning Columns**

*Resource Consumption Columns* (Duration, CPU, Reads, Writes, etc.) can be repositioned left or right within the grid. Use drag and drop. Data grouping columns cannot. They must stay put.

#### **Expanding Groups**

All grid rows representing groups of events (except for the lowest level group) will have an *Expand/Collapse* control to its left. The **(+)** icon indicates that a lower-level group can be found within this group. Click it to expand that one group and to make its sub group rows visible. When a sub group appears, it will be indented slightly and surrounded by a blue border. You may repeat this for as many groups as are present in the grid.

Only one group row at each level can be expanded at any one time. If you click (+) to expand any other group row, it will expand, and will also close any previously expanded groups at that level. To collapse groups, click the (-) icon.

#### Viewing Additional Rows (Paging)

The grid, by default, shows only the first 50 rows in the highest-level grouping. To see additional rows, click the **More** button at the bottom of the grid.

#### Viewing More Rows in Sub Groups

Whenever you expand a group and can see its sub-groups, the initial display shows at most 10 sub-group rows. To see an additional 10 sub-group rows, click the **More** button at the bottom of the sub-level. You can probably guess what will happen if you click **More** again.

#### Sorting Grid Data

Each of the *Resource Consumption Columns* in the grid is sortable. Click the column header name to sort by that column in descending order. Click it a second time to sort that same column in ascending order. You can stay occupied for quite a long time in this fashion.

#### **Viewing Full Resource Metrics**

If the column is not wide enough to show numeric measurements and you don't want to widen the row, you can see the metric at the intersection of one group row and one resource by placing the mouse cursor over the blue bar. Its numerical measurement will appear after a second in a tool tip. To see all resource consumption metrics for one or many rows, see section 5.1.2 - Resource Consumption Statistics.

# 5.2.5 Selection Details

This pane gives you a more detailed look at selected rows. It provides details on resource consumption and details on the batch.

#### **Single Group Resource Consumption Details**

Select any single row, and all of its resource consumption measurements appear in the bottom *Resource Consumption* pane.

<b>A</b> .	Some Dand	mbaightor as contoon	/	-28,19, ec				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~	$\sim$	m
٠	SELECT CampaignID	As <value> , Campai</value>		17.78 sec		12.13 sec		3.15M		$\mathbf{I}_{i,j}$	$\mathbf{I}_{i} = \mathbf{I}_{i}$
٠	EXEC EDS_SENDER_	FETCH_PENDING_RES		11.09 sec		3.28 sec	1.00	166.85K	1	1	1
٠	SELECT IOID As <va< th=""><th>lue&gt; , PayingAccount</th><th>1. Sec. 1</th><th>7.12 sec</th><th></th><th>6.8 sec</th><th></th><th>873.59K</th><th>I.</th><th>1</th><th>1</th></va<>	lue> , PayingAccount	1. Sec. 1	7.12 sec		6.8 sec		873.59K	I.	1	1
٠	select * from SQL_FN	N_GET_CAMPAIGNS_VI	1	3.01 sec		2.64 sec	1	56.25K		I.	1
٠	select CampaignID,[	Name] from SQL_FN	1	2.23 sec	•	3.39 sec	1	60.68K		1	1
٠	EXEC BSA_SQLCMD	FLIGHTS_DATA_BY_C	1	2.18 sec	1. Sec. 1	1.69 sec	1.00	286.59K	1	1	۰. ۲
Res	Resource Consumption 7										
Du	iration	11.09 sec (0.7% of Wor	kload)								
CP	U	3.28 sec (0.3% of Work	load)								
Re	ads	166.85K (0.1% of Work	load)								
W	rites	0 (0.0% of Workload)									
Ro	w Count	533.78K (1.8% of Work	load)								
Ev	vent Count 45 (0.0% of Workload)										
Reso	ource Consumption	Batch									

#### Figure 44 - Single Group Details

These are the same measurements that appear graphically (as blue bars) in the *Analysis Results Grid*, but their statistics are all visible at once, and you can also see what percentage of the full workload is represented by each statistic.

#### **Multiple Groups Resource Consumption**

Select multiple rows, and the sum of the resource consumption measurements for these rows appears in this pane.

[	<b>Shortcut</b> : Use these familiar windows shortcuts to select rows:
~	<ul> <li>CTRL+Click = Select one more</li> <li>Click, followed by SHIFT+Click = Select a range</li> <li>Click, hold and Drag = Select a range</li> <li>CTRL+A = Select all</li> </ul>

In the screenshot below, five rows have been selected. The sums of their measurements are shown in the *Resource Consumption* pane at the bottom.

$\sim$	Anna	~ some ~ ~ ~	Ann	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~	non the second	mon	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~	$\sim$		~~
+	SELECT CampaignID	As <value> , Campai</value>		17.78 sec		12.13 sec		3.15M		$\mathbf{I}_{i,j}$	$\mathbf{I}_{i} = \mathbf{I}_{i}$	
÷	EXEC EDS_SENDER_F	ETCH_PENDING_RES		11.09 sec	• • • • • • • • • • • • • • • • • • •	3.28 sec	1.00	166.85K	1	1	1	
٠	SELECT IOID As <va< th=""><th>lue&gt; , PayingAccount</th><th>1.00</th><th>7.12 sec</th><th></th><th>6.8 sec</th><th></th><th>873.59K</th><th>1</th><th>1.1</th><th>1</th><th></th></va<>	lue> , PayingAccount	1.00	7.12 sec		6.8 sec		873.59K	1	1.1	1	
÷	select * from SQL_FN	GET_CAMPAIGNS_VI	1.00	3.01 sec	1.1	2.64 sec	1	56.25K	1	1	1	
÷	select CampaignID,[I	Nume] from SQL_FN	1	2.23 sec		3.39 sec	1	60.68K	1	1	1	
÷	EXEC BSA_SQLCMD_	FLIGHTS_DATA_BY_C	I.	2.18 sec	1	1.69 sec	1.00	286.59K	1	1	1	Ŧ
Tota	al Resource Consumpt	tion of 4 Selected Groups	;									ą
Du	ration	23.44 sec (1.5% of Wor	kload)									
СР	U	16.11 sec (1.5% of Wor	kload)									
Re	ads	1.16M (1.0% of Worklo	ad)									
Wr	ites	2 (0.0% of Workload)										
Ro	ow Count 861.82K (2.9% of Workload)											
Eve	ent Count	48 (0.0% of Workload)										
No.						· · · ·	<b>.</b>					_

**Figure 45 - Multiple Groups Details** 

Rows may be selected in any of the grouping levels. However, you may only select rows from one level at a time.

#### **Batch Details**

Click any one row underneath the column headers *Batch* or *Batch Instances*, located at any level, and you will be able to see its full text. Follow these steps:

- 1 Expand the hierarchy of the grid to reveal the level you are interested in. In this example, we want to see the text for *Batch Instances*.
- 2 Click to select a single row.
- **3** At the bottom of the *Selection Details* pane are two tabs. The tab on the right is labeled *Batch*. Click it.





Batch	8
SELECT * FROM Sales.SalesOrderDetail WHERE SalesOrderID IN ( SELECT SalesOrderID FROM Sales.SalesOrderHeader WHERE Freight > (1000 + (SELECT 5)) )	mar have
Show   Batch Instance Sample  Batch Template	
Resource Consumption Batch	2

**Figure 47 - Batch Instance View** 

**4** The contents of the *Selection Details* pane change to show, at your choice, the row's *Batch* or the complete text of the *Batch Instance*. Use the radio buttons at the bottom of the pane to toggle between them.



#### Figure 48 - Batch Template View



**Note**: When you've selected a *Batch* row, and you request to see its Batch Instance, you won't see all possible Batch Instances within that group. You'll see just one sample.



**Note**: To copy the text from either the *Batch* or the sample *Batch Instance* into the Windows Clipboard, click the **Copy** button on the right lower side.

# 6 Comparing Workloads

This chapter describes how to compare two workloads. The workloads can be compared whether they are *Controlled* or *Uncontrolled*.

*Controlled Workloads* contain an identical set of SQL queries and are replayed under two different conditions. Usually the testers modify only one aspect of the environment between replays. They may be evaluating a code change, a server upgrade or a different piece of hardware for example. Using *Controlled Workloads* offer one means to achieve an apples-to-apples comparison.

Uncontrolled Workloads on the other hand, usually contain a similar but nonidentical set of SQL queries. When you capture two different sets of traces in production for example, these workloads are usually uncontrolled, because in a production environment, transactions occur as business needs dictate, and so they are not predictable. Qure Analyzer can compare either Workload type.



**Important**: Qure Analyzer can only compare two workloads if at the time of creation (see section 4.2.1 – Saving the Workload Analysis) you opted to store the results of both workload analyses in a database, and on the same SQL Server instance. Workloads whose analysis results are stored in files or on different SQL Server instances cannot be compared.

# 6.1 **Opening Multiple Workload Analyses**

To open more than one workload analysis, repeat the instructions in sections 4.2. - Setting up the Workload Analysis or 4.1.1. - Opening a Previously Analyzed Workload as many times as you like. When two or more workload tabs are open, you are ready to proceed to the next step.

**Note**: You may open the same workload analysis multiple times. For each additional instance, Qure Analyzer will add an enumerator to distinguish the workloads. This allows you to compare different characteristics of the same workload using the compare feature and the dual filters. For example, you may analyze the differences of your workload's behavior between 7:00 - 8:00 compared to 8:00 - 9:00 by comparing the workload to itself and applying different filters for the baseline and for the evaluated workload analyses.

# 6.2 Comparing Workloads

Before putting your hands on the keyboard again, choose one of the two workloads to be the *Baseline Workload* (the '*before'* workload) and a second workload to be the *Evaluated Workload* (the '*after'* workload). When Qure Analyzer performs a comparison, it interprets lower resource consumptions in the '*after'* workload as being 'good' and shows that difference as an improvement using a green bar. Similarly, it interprets larger resource consumptions in the '*after'* workload as being '*bad'* and shows that difference as a degradation using a red bar. You'd probably hate to get that backwards for the big presentation to the boss.

#### Steps to Comparing Two Workloads:

- 1 Click on the light green tab showing the name of your Evaluated Workload.
- 2 Within that tab, click the button **Compare to Baseline**.
- **3** From the list of workloads that appears, select the workload that will serve as your *Baseline*.

P Compare Workloads
Compare Post-Tuning to
Select the baseline workload for comparison:
Pre-Tuning
Compare to Baseline Cancel

Figure 49 - Compare Workloads Dialog

- 4 Click the button Compare to Baseline.
- **5** Soon, the *Details Report* for the comparison analysis will be displayed. This window will be identified by a tab header showing the names of both workloads.

Qure Analyzer				-	_		A.a. 1				-
Q Start Page Pre-Tuning ×	Post-Tunin	g ×	Post-Tuning vs. Pre-Tuning ×								
Comparison View	*) (*) II		Batch	Duration	Ť	CPU		Reads		Writes	
<ul> <li>Save\Load Settings</li> </ul>		٠	EXEC EDS_SENDER_FETCH_PENDING		+2.11 min		+49.93 sec	-	+5.29M		Same
		٠	select "Tbl1004"."EyeblasterID" as "C		+1.42 min	•	+13.34 sec		+6.22M		Same
← Group By		٠	select "Tbl1006"."InteractionName"		-1.23 min		-28.24 sec	1	-120.87K		Same
-		٠	SELECT CampaignID As <value> , C</value>	-	+51.1 sec		+41 sec		+15.93M		+22.2
+ Batch +		٠	select "Tbl1006"."CampaignID" as "C	•	-29.57 sec		-22.15 sec	•	-2.77M	1	-1.15K
+ Batch Instance • +	1 U X	٠	EXEC JSS_FETCH_JOB	•	+26.28 sec		+39.24 sec		+16.12M	1	-3
-		٠	EXEC BSA_SQLCMD_FLIGHTS_DATA	•	+25.86 sec		+30.25 sec		+6.53M	•	+1.8K
6	Apply	٠	EDS_ENGINE_SQLCMD_UPDATE_FLI	•	+24.74 sec	1	+3.73 sec	1	+906.26K		+28.04
		٠	BSA_SQLCMD_CHECK_FOR_MOBILE	•	+20.1 sec	1	+1.09 sec	1	+26.76K		Same
▲ Filter By		٠	SELECT IOID As <value> , IOID As &lt;</value>	1	+13.92 sec	1	+441 ms	1	+149.07K		Same
Add Filter			and the set of the set	A server	A48.50mm	-	A	A. (			m l

Figure 50 - Comparison View

# 6.3 Using the Workload Comparison Details Report

If this report looks familiar, it is because it closely matches the *Workload Details Report*. Much of the *Comparison Details Report* works the same as the *Workload Details Report*. See section 5.2 – Using the Workload Details Report.

#### 6.3.1 Group by

This control works in much the same way as in the *Workload Details Report*. See section 5.2.1 - Using the Group by Control.

#### 6.3.2 Filtering

This works very similarly to the *Workload Details Report*. See section 5.2.1 - Using the Filter Control. However, in the comparison report the filter window is doubled to allow you to set different settings for each workload. For example, the *Users* filter will have two sections instead of one. The section on the left will be used to filter the baseline workload and the one on the right to filter the *Evaluated Workload*.



Figure 51 - Dual Filters Window

On the bottom ribbon, the lock button can be used to lock both sections in order to save you multiple clicks when you want to apply the filter to both workloads. Once in the locked position, any setting change you make on one workload will be duplicated to the other workload automatically.

# 6.3.3 Comparison Mode

In the *Workload View* pane is a section titled *Comparison Mode*. This gives you a full set of options for adjusting the data in two workloads so that they can be reliably compared.

#### Using the Include Control

The *Include* control gives you three options for specifying how closely defined any two event groups must be in the two workloads in order to consider them a comparable pair.

<ul> <li>Comparison Mode</li> </ul>	
Include	
All Event Groups	
Matching Event Groups Only	
Equalize Event Count to Baseline	

#### Figure 52 - Comparison Mode Control

The purpose and usage of these three options becomes obvious when looking at sample data. Consider the following two workloads being compared:

#### Baseline Workload

#### **Evaluated Workload**

Batch	Batch Instance	User	CPU	Batch	Batch Instance	User	С
EXEC SP1 <value></value>	EXEC SP1 @p=5	John	125ms	EXEC SP1 <value></value>	EXEC SP1 @p=5	John	1
EXEC SP1 <value></value>	EXEC SP1 @p=5	David	125ms	EXEC SP1 <value></value>	EXEC SP1 @p=5	John	1
				EXEC SP1 <value></value>	EXEC SP1 @p=5	Mary	1
EXEC SP1 <value></value>	EXEC SP1 @p=7	John	600ms				
				EXEC SP1 <value></value>	EXEC SP1 @p=8	John	6
EXEC SP2 <value></value>	EXEC SP2 @p='abc'	David	300ms				
				EXEC SP3 <value></value>	EXEC SP3 @p='2012'	Mary	6
EXEC SP3 <value></value>	EXEC SP3 @p='2013'	John	500ms	EXEC SP3 <value></value>	EXEC SP3 @p='2013'	John	4
EXEC SP3 <value></value>	EXEC SP3 @p='2014'	Mary	80ms				T
				EXEC SP4 <value></value>	EXEC SP4 @p=0.4	John	1

Figure 53 – Example Workloads for Comparison

#### All Event Groups

This option (the default setting) discards no events from either workload. All events are included and grouped according to your *Group by* settings. If one of the workloads does not have a group to match the other workload, an empty group will be created and resources will be given values of zero. Comparisons of these artificially created pairs will appear as 100% improvement (e.g. from 100ms to 0ms) or an infinite degradation (e.g. from 0ms to 100ms).

It's pretty evident this comparison mode is not very meaningful when the workloads are *Uncontrolled*. That is, when they vary substantially with regards to the events they contain. Here's what the *Comparison Report* would show if comparing only the *Batch* groups:

Batch	Baseline CPU	Event Count	Evaluation CPU	Event Count	Difference (Total)
EXEC SP1 <value></value>	125+125+600= 850ms	3	100+100+100+600=900ms	4	+50ms
EXEC SP2 <value></value>	300ms	1	0ms	0	-300ms
EXEC SP3 <value></value>	500+80=580ms	2	600+400=1000ms	2	+420ms
EXEC SP4 <value></value>	0ms	0	1200ms	1	+1200ms

#### Figure 54 - Default Comparison (Include All)

As you can see, these results report a significant improvement for the *Batch* EXEC SP2 **<Value>** and a significant degradation for the batch EXEC SP4 **<Value>** when in fact both are meaningless in this context, since both these *Batches* are only present in one of the two workloads.

For this reason, the *Include All Event Groups* option is typically useful only with *Controlled Workloads*, where you know that all events are identical.

# Matching Event Groups Only with Event Counts not Equalized to Baseline

This option discards events for which there is no match based on the setting of the current *Group by* settings.

For example, if the *Group by* control is set to group by *Batch* only, the EXEC SP1 <Value> and EXEC SP3 <Value> are retained because they are the only *Batches* that are found in both workloads. All other *Batches* will be discarded.

Batch	Baseline CPU	Event Count	Evaluation CPU	Event Count	Difference (Total)
EXEC SP1 <value></value>	125+125+600=850ms	3	100+100+100+600=900ms	4	+50ms
EXEC SP2 <value></value>	<del>300ms</del>	1	<del>0ms</del>	θ	<del>-300ms</del>
EXEC SP3 <value></value>	500+80=580ms	2	600+400=1000ms	2	+420ms
EXEC_SP4 <value></value>	<del>0ms</del>	θ	<del>1200ms</del>	÷	+1200ms

Figure 55 - Matching Event Groups Comparison by Datch	Figure 55 - Matching	<b>Event</b>	Groups	Comparison	by Batch
---	----------------------	--------------	--------	------------	----------

Note how the *Batch* EXEC SP3 **<Value>** appears to have degraded when considered in this fashion.

As another example, if the *Group by* control is set to group by *Batch* and *Batch Instance* (or just *Batch Instance*), only the following rows are retained because they are the only identical *Batch Instances* that are found in both workloads.

Batch	Batch Instance	Baseline CPU	Event Count	Evaluation CPU	Event Count	Difference (Total)
EXEC SP1 <value></value>	EXEC SP1 @p=5	125+125= 250ms	2	100+100+100= 300ms	3	+50ms
EXEC_SP1 <value></value>	EXEC SP1 @p=7	<del>600ms</del>				
EXEC_SP1 <value></value>	EXEC SP1 @p=8			<del>600ms</del>		
EXEC SP2 <value></value>	EXEC SP2 @p='abc'	<del>300ms</del>				
EXEC SP3 <value></value>	EXEC SP3 @p='2012'			<del>600ms</del>		
EXEC SP3 <value></value>	EXEC SP3 @p='2013'	500ms	1	400ms	1	-100ms
EXEC SP3 <value></value>	EXEC SP3 @p='2014'	<del>80ms</del>				
EXEC SP4 <value></value>	EXEC_SP4_@p=0.4			<del>1200ms</del>		

#### Figure 56 - Matching Event Groups Comparison by Batch Instance

Note how the *Batch* EXEC SP3 <Value> appears to have improved when considered in this fashion.

Note also how EXEC SP1 @p=5 appears to have degraded when considered in this fashion, even though we can see that each individual *Batch Instance* went from 125ms to 100ms. The reported total increase in CPU is due to the increase in the number of events in the *Evaluated Workload*.

#### A (semi) Real-Life Example

Below is a sample image of a workload that was compared using the *Include* option *Matching Event Groups Only*. The only Groups that appear in the *Analysis Grid* are those which appear in both workloads. One of them has been selected (highlighted) to show how the resource measurements compare. Because the *Resource Aggregation Mode* was set to *Total* (not shown), these measurements are sums. Note that the duration measurement is 25% higher in the *Evaluated Workload*. How is this best interpreted? Good or bad? It looks bad on the surface – red looks bad. But also note that the evaluation's version of this *Batch* group contained 42% more events in it. In that case, one may expect its total duration to be 42% higher if there were no tuning actions performed at all. So in this case, there seems to actually be a small improvement, but just by how much?

	Batch		Duration	n *	CPU		Reads		Writes		Row Co	Event	•
۰	EXEC EDS_SEM	DER_FETCH_PENDING_RES		+2.11 min		+49.93 sec		+5.29M		Same		1	Ξ
۰	select "Tbl100	4"."EyeblasterID" as "Col10		+1.42 min		+13.34 sec		+6.22M		Same		$\mathbf{I}_{i,j}$	
۰	select "Tbl100	6"."InteractionName" as "C		-1.23 min		-28.24 sec	1	-120.87K		Same		1	
۰	SELECT Camp	aignID As <value> , Campa</value>		+51.1 sec		+41 sec		+15.93M		+22.25K		$\mathbf{I}_{i,j}$	
٠	select "Tbl100	6"."CampaignID" as "Col10		-29.57 sec		-22.15 sec		-2.77M	1	-1.15K		1	
٠	EXEC JSS_FET	CH_JOB		+26.28 sec		+39.24 sec		+16.12M	1	-3		1	
٠	EXEC BSA_SQ	LCMD_FLIGHTS_DATA_BY_C		+25.86 sec		+30.25 sec		+6.53M	•	+1.8K		1	
٠	EDS_ENGINE_	SQLCMD_UPDATE_FLIGHT_I		+24.74 sec	1	+3.73 sec	1 - C	+906.26K		+28.04K			
٠	BSA_SQLCMD	_CHECK_FOR_MOBILE_FLIG	•	+20.1 sec	1	+1.09 sec	1	+26.76K		Same		1	
141				+13.92 sec	1	+441 ms	1	+149.07%	_	Same	-	1	Ŧ
Res	ource Consump	otion											ф
Du	uration	+2.11 min (25%)			_	8.32 r	min in Ba	iseline (Pre-Tuning) 3 min in Post-Tunin	a				
CF	υ	+49.93 sec (33%)		2.46 min in Baseline (Pre-Tuning)									
Re	ads	+5.29M (70%)		3.29 min in Post-Tuning 7.51M in Baseline (Pre-Tuning)									
w	rites	Same	0 in Baseline (Pre-Tuning) 0 in Baseline (Pre-Tuning)										
Rc	w Count	Same	I	0 in Baseline (Pr 0 in Post-Tuning	e-Tuning	1)							
Ev	ent Count	+19 (42%)		45 in Baseline (Pre-Tuning)									

Figure 57 – Matching Event Groups Only Comparison

Someone is bound to say, "Sure, it's nice to see all the details, but can we also get a simpler view that shows this as a green bar if in fact there's been an overall improvement?" Yes. The next comparison option provides that punch-line type view.

# Matching Event Groups Only with Event Counts Equalized to Baseline

This option factors out the impact of differences in event count. It equalizes the event count (and all resource measures along with it) of the *Evaluated Workload* with this simple formula:

#### Equalized Resource Measure =

(Total Metric of Evaluation Group) X (Event Count of Baseline Group)

Event Count of Evaluated Workload Group

Batch	Batch Instance	Baseline CPU	Event Count	Evaluation CPU	Event Count	Event Count Equalization	Equalized Evaluation CPU	Difference (Total)
EXEC SP1 <value></value>	EXEC SP1 @p=5	125+125= 250ms	2	100+100+100= 300ms	3	X 2/3	200ms	-50ms
EXEC SP1 <value></value>	EXEC_SP1 @p=7	<del>600ms</del>						
EXEC_SP1 <value></value>	EXEC_SP1 @p=8			<del>600ms</del>				
EXEC_SP2 <value></value>	EXEC_SP2 @p='abc'	<del>300ms</del>						
EXEC_SP3 <value></value>	EXEC_SP3 @p='2012'			<del>600ms</del>				
EXEC SP3 <value></value>	EXEC SP3 @p='2013'	500ms	1	400ms	1	X 1/1	400ms	-100ms
EXEC_SP3 <value></value>	EXEC_SP3 @p='2014'	<del>80ms</del>						
EXEC SP4 <value></value>	EXEC SP4 @p=0.4			<del>1200ms</del>				

Here is how it would look using our example workloads:

#### Figure 58 - Equalized Comparison

Note how EXEC SP1 @p=5 now realistically reports that the CPU time has gotten shorter on average by 50ms. This passes the gut-check because we can see that each individual *Batch Instance* went from 125ms to 100ms. In both cases we see a 20% better CPU measurement.

#### A (semi) Real-Life Example

Below is a sample image of the same workload we saw in Figure 57. However, this image shows the *Resource Comparison Details* after using the *Include* option *Matching Event Groups, Equalized to Baseline*. The same *Batch Instance* has been selected (highlighted) to show how the resource measurements compare now. Note that the *Duration* measurement is 11% improved in the *Evaluated Workload*. Why the difference? It's because the *Duration* measurements have been adjusted so that their event counts are equivalent to the event counts of the *Baseline* group. Once that's done, then it's clear that the *Duration* measurements really are improved, and the red bar from Figure 57 turns to green in Figure 59.

	Batch		Duration	1	CPU		Reads		Writes		Row Co	Event	*
۰	EXEC EDS_SEM	NDER_FETCH_PENDING_RES		-59.11 sec		-8.74 sec		+1.49M		Same			Ξ
۰	EXEC JSS_FET	CH_JOB		-19.75 sec	-	-4 sec		+4.16M	1	-10.71			
۰	EXEC Admin_	Eyeblasters_Get_AdsList_Dy		-9.45 sec		+14.81 sec	-	+1.41M		Same			
٠	EDS_ENGINE_	SQLCMD_UPDATE_FLIGHT_I		-8.59 sec	1	-519 ms	1	+85.6K		-2.5K			
۰	EXEC RS_Repo	ortingServices_SelectImmedi		-8.31 sec		-1.93 sec		+3.36M		Same			
٠	EXEC BSA_SQ	LCMD_FLIGHTS_DATA_BY_C		-5.93 sec	I.	+140 ms	-	+807.21K	1	-13.92			
٠	EXEC BSA_SQ	LCMD_INSERT_NEW_EYEBL		-5.69 sec	I.	+379 ms	1	+70.27K	1	-б			
۰	SELECT Camp	aignID As <value> , Campa</value>	1	-4 sec	•	-1.5 sec	•	+667.04K	•	+321.2			
۰	EXEC Admin_G	Campaigns_Get_Campaigns	1	-3.08 sec	1	-152 ms	•	+283.11K	1	-0.67			
-	EDS ENIGINE			-3 sec	1	-23 ms	1	+26 39K		-2 33K		_	Ŧ
Res	ource Consump	otion											ф,
Du	uration	-59.11 sec (11%)				7.	8.32 min 33 min in Po	in Baseline (Pre- st-Tuning (Equa	-Tuning) lized from	10.43 min)			
CF	υ	-8.74 sec (5%)		2.46 min in Baseline (Pre-Tuning)									
Re	ads	+1.49M (19%)		2.32 min in Post-Turning (Equalized from 3.29 min) 7.51M in Baseline (Pre-Turning) 9M in Post-Turning (Equalized from 12.8M)									
W	rites	Same	0 in Baseline (Pre-Tuning) 0 in Post-Tuning										
Rc	w Count	Same	I	0 in Baseline (Pre-Tuning) 0 in Post-Tuning									
Ev	45 in Baseline (Pre-Tuning) 45 in Post-Tuning (Equalized from 64)												

Figure 59 – Matching Event Groups, Equalized to Baseline

#### 6.3.4 Resource Aggregation Mode

This acts the same as in the *Workload Details Report*. See section 5.2.35.2.3 - Changing the Resources Aggregation Mode.

#### 6.3.5 Resource Columns Sort Mode

In the Comparison View pane is a section titled *Resource Columns Sort Mode*. It offers the following two sort options:

#### Largest to Smallest Difference (Absolute)

When this option is selected, the groupings with the greatest change (regardless of whether it was a good or bad change) are sorted to the top. The groupings with the smallest change are sorted to the bottom. The entire top contains wide difference bars that may be green or red. In this view you can quickly see whether the *Evaluated Workload* is more heavily weighted towards green (improvements) or red (degradations) for the particular selected grouping.

Duration	*
R.	-2.11 min
6)	-1.42 min
	+1.23 min
	-51.1 sec
	+29.57 sec
	-26.28 sec
	-25.86 sec
	-24.74 sec
	-20.1 sec
	-13.92 sec
	-12.59 sec

#### Figure 60 - Absolute Soring Mode

To sort in the other direction (smallest difference on top), click the column header of the resource type (e.g. Duration, CPU). Each click toggles its sort direction.

#### Most Improved to Most Degraded (Relative)

When this option is selected, the most improved groupings are sorted to the top of the column, and the most degraded are sorted to the bottom. The entire top looks green and the entire bottom looks red.

uration	
	-2.11 min
	-1.42 min
	-51.1 sec
	-26.28 sec
	-25.86 sec
	-24.74 sec
	-20.1 sec
	-13.92 sec
	-12.59 sec
	-12.43 sec
	-10.64 sec

#### Figure 61 -Relative Sort Mode

To sort in the other direction (most degraded on top), click the column header of the resource measure (e.g. Duration, CPU). Each click toggles its sort direction.

#### 6.3.6 Workload Comparison Grid

The Workload Comparison Grid behaves in much the same way as the grid of the Workload Details Report. The main difference is that the red and green difference bars show the difference in resource consumption from the Baseline Workload to the Evaluated Workload. If the bar is green, the Evaluated Workload has a better performance metric than the Baseline Workload (lower resource consumption). The numbers that appear to the right of each bar is the size of the difference between the two measurements. This number will represent a difference in the totals or a difference in the averages, according to how you set the Resource Aggregation Mode.

In the image below, the *Resource Aggregation Mode* was set to total (not shown here).

	Batch	Duration	•	CPU		Reads		Writes	-5
÷	EXEC EDS_SENDER_FETCH_PENDING_RES		+2.11 min		+49.93 sec		+5.29M		Same
÷	select "Tbl1004"."EyeblasterID" as "Col10		+1.42 min		+13.34 sec		+6.22M		Sam
÷	select "Tbl1006"."InteractionName" as "C		-1.23 min		-28.24 sec	1	-120.87K		Saf
+	SELECT CampaignID As <value> , Campa</value>	-	+51.1 sec		+41 sec		+15.93M		+22.
÷	select "Tbl1006"."CampaignID" as "Col10		-29.57 sec		-22.15 sec		-2.77M	1	-1.15
+	EXEC JSS_FETCH_JOB	-	+26.28 sec		+39.24 sec		+16.12M	1	-3 \
+	EXEC BSA_SQLCMD_FLIGHTS_DATA_BY_C	-	+25.86 sec		+30.25 sec		+6.53M	•	+1.8
+	EDS_ENGINE_SQLCMD_UPDATE_FLIGHT_I	-	+24.74 sec	1.00	+3.73 sec	1.00	+906.26K		+28.0
÷	BSA_SQLCMD_CHECK_FOR_MOBILE_FLIG	•	+20.1 sec	1	+1.09 sec	1	+26.76K		Sa
÷	SELECT IOID As <value> , IOID As <valu< th=""><th></th><th>+13.92 sec</th><th>1</th><th>+441 ms</th><th>1</th><th>+149.07K</th><th></th><th>Same</th></valu<></value>		+13.92 sec	1	+441 ms	1	+149.07K		Same
+	EXEC BSA_SQLCMD_DELETE_FLIGHT_LIST	•	+12.59 sec	1	+1.27 sec	1	+79.75K	1	+97
+	EXEC RS_ReportingServices_SelectImmedi	•	+12.43 sec		+14.93 sec		+16.06M		Sam
+	SELECT CampaignID As <value> , Campa</value>		+10.64 sec		+9.55 sec		+3.8M		+4
+	EDS_ENGINE_SQLCMD_UPDATE_EYEBLAS		+9.54 sec	1	+1.31 sec	1	+278.11K	-	+3
+	select "Tbl1004"."EyeblasterID" as "Col11	1	+8.98 sec	1	+2.08 sec		+30.85K		San

#### Figure 62 - Workload Comparison Grid

#### Examples

- The top *Batch* has degraded in duration. The evaluated group of *Batches* took 2.11 more minutes to process all the events in its group than did the baseline.
- The top three *Batches* show that the numbers of writes, as measured in both the evaluated and baseline workloads are the same.

## 6.3.7 **Resource Consumption Details**

The resource consumption details pane behaves much the same as it does for a single workload analysis. See section 5.2.5 - Selection Details.

However, instead of showing a single bar, it shows two bars: A grey bar for the measurement of the *Baseline's Workload* group and a green, red, or grey bar for the measurement of the *Evaluated Workload's* group. Also, for each bar, the measurement and the name of the workload is listed alongside.

On the left side of the *Resource Consumption Details* pane is the label for each resource type available within your traces, and two representations of the difference – numeric and percentage.



Figure 63 – Resource Consumption Details

# 7 Managing Workload Analyses

This chapter describes how to add traces to an existing analysis and how to delete a workload analysis.





# 7.1 Adding Traces to a Workload Analysis

Qure Analyzer allows you to add additional traces to a workload analysis even after it has completed. Simply click the **Add Trace** button on the top right section of the ribbon and follow the wizard which is the same one you used in section 4.2.2 - Selecting the Analysis Workload Traces to setup the original workload analysis.

# 7.2 Deleting a Workload Analysis Repository

When a workload analysis is saved to a database repository, the tables that comprise the repository are created in a schema which is named after the workload name. This allows you to save multiple workload analyses to the same database but makes deleting a particular analysis cumbersome.

To easily delete a workload analysis schema and all of its tables, simply click the **Delete Analysis** button on the top right section of the ribbon.

The confirmation dialog will appear.

Delete Wo	rkload Analysis 📡
8	Deleting a workload analysis will delete all associated tables from the database. Unless you've backed up the repository database, this action is irreversible. Are you sure you want to completely delete this workload analysis?
	Yes No

#### Figure 65 - Workload Analysis Delete Confirmation

Click Yes to delete the workload analysis repository.

Note that this option is not available for workload analyses that are saved in files. To delete those, simple delete the containing file from your hard disk.



July 2012

http://www.dbsophic.com