

Kofax Communication Server

TC/PerfLog Technical Manual

Version: 10.3.0

Date: 2019-12-13

The KOFAX logo is rendered in a bold, blue, sans-serif typeface. The letters are thick and closely spaced, with a consistent weight throughout the word.

© 2019 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

Table of Contents

Chapter 1: Preface	4
Chapter 2: Installation	5
Prerequisites.....	5
Installation with KCS.....	5
Manual Installation.....	6
Chapter 3: Software Components / Resources Belonging to TC/PerfLog	8
Chapter 4: Functionality	9
Mode-1: Logging by Custom Logging Task Supervising.....	9
Mode-0: Creating / Writing / Supervising Log Files by TC/PerfLog.....	10
Chapter 5: Usage	12
Chapter 6: Configuration	15
Configuration in the Registry.....	15
Counter Configuration File (counters.txt).....	17
Chapter 7: Troubleshooting	21
Checking the Proper Operation.....	21
Chapter 8: Remarks	22

Chapter 1

Preface

TC/PerfLog is a component of Kofax Communication Server that controls the process of collecting + logging of performance counters: once the parameters for performance counter logging defined, TC/PerfLog provides permanent logging (e.g. automatic log-restart after machine reboot) and log file size control (=keeping only the most recent log files according to the LogSizeTotalMax setting) .

TC/PerfLog can use 2 different ways to collect the performance data:

- **Mode-0:** TC/PerfLog creates, writes, and supervises the performance log files.
- **Mode-1:** TC/PerfLog creates and supervises an according Window Performance Monitor “custom logging task”. The performance log files will be created by this task. (Default mode.)

The two modes are doing basically the same. The slight advantage of Mode-1 is, that the Windows Server 2008 “Reliability and Performance Monitor” application provides direct display of the custom logging task log files (by navigating on the left-tree-view to the “Reports” / “User Defined” folder) without an extra log-file-load step.

Important The Kofax Communication Server (KCS) and its components formerly used the name **TOPCALL**. The abbreviation **TCOSS** was used earlier as alternative name for the **TOPCALL** server. Names **TOPCALL** and **TCOSS** appear even now in some registry settings and path names. Also, the frequently used prefix ‘TC’ refers to the old name **TOPCALL**.

Chapter 2

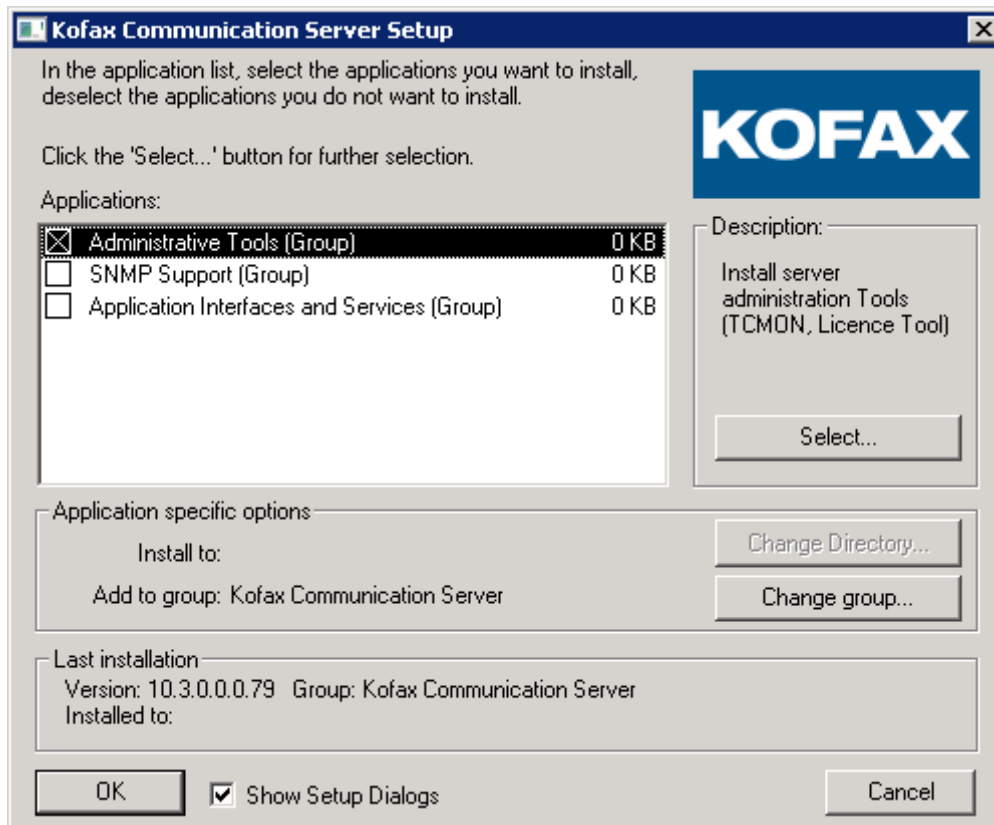
Installation

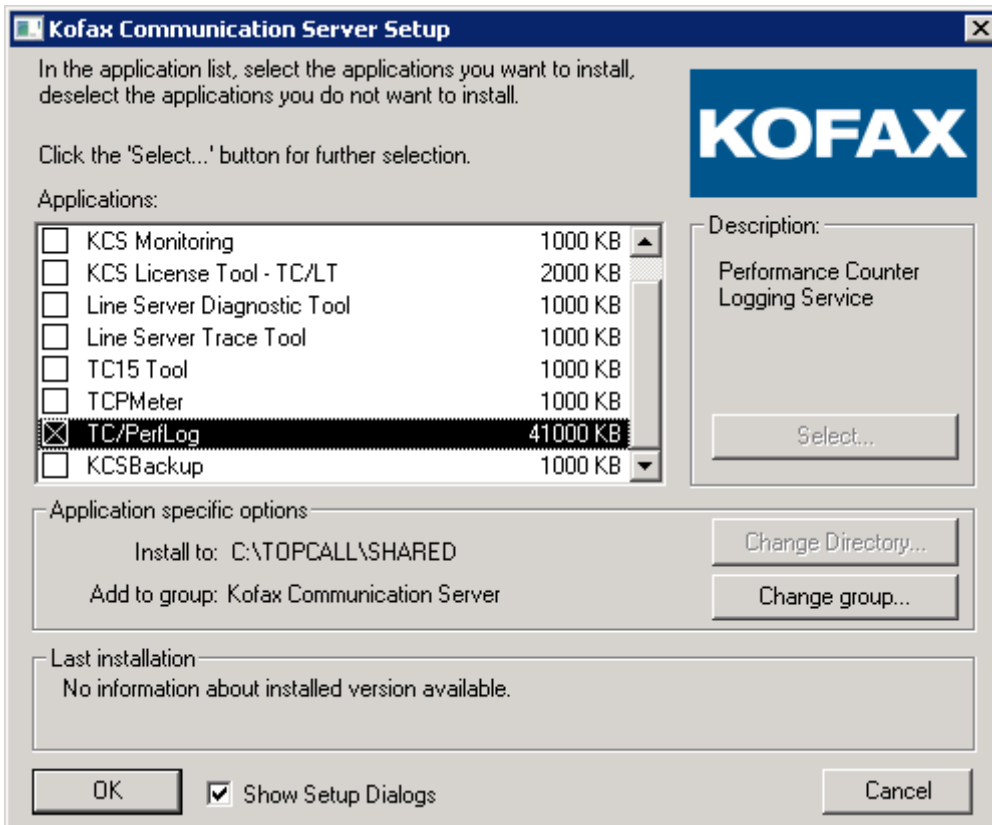
This section describes the installation of TC/PerfLog.

Prerequisites

TC/PerfLog can be installed on Windows Server 2008 and 2012.

Installation with KCS





1. Select **TC/PerfLog** in Common > Administrative Tools group of KCS setup.
2. Click **Install**.
3. (Optional) Modify the TPCERFLOG registry settings (HKLM\SOFTWARE\TOPCALL\TPCERFLOG\Config) and/or counters.txt, if you want other settings/counters as the default.
4. Restart the server with the KCS Monitor to activate the TPCERFLOG process.
5. Check if the performance counter logging / supervising is running properly. See chapter [Checking the Proper Operation](#).

Remarks:

- The setup will perform exactly the steps 1 to 6 that are described in section [Manual Installation](#).
- If the file counters.txt already exists, then this installation never overwrites it (even if it is older than the shipped one). If you want change your existing counters.txt with the shipped one, please delete your existing counters.txt prior to installation.

Manual Installation

Typically, to install TC/PerfLog, proceed as described in chapter [Installation with KCS](#).

If KCS is already installed on the machine and you have acquired tperflog.exe + counters.txt + this documentation from Kofax, but you do not have the KCS setup version that supports TC/PerfLog installation option, then you must install TC/PerfLog manually as described in this chapter.

1. Copy tperflog.exe to c:\topcall\shared.
2. Create the directory c:\tcoss\perflog.
3. Create a c:\tcoss\perflog\counters.txt file with the performance counter name list (plain text, one line = one counter name, see more below.)
4. Create the registry key ..\TOPCALL\TCPERFLOG.
5. Create the registry value ..\TOPCALL\TCPERFLOG\CommandLine (REG_SZ) = "C:\TOPCALL\SHARED\TCPERFLOG.EXE TCMONITOR".
6. Add TCMONITOR to the ..\TOPCALL\Boot\Startup setting.
7. Restart the server with the KCS Monitor. So, also the TCMONITOR process will be started. Now, tperflog.exe creates automatically the registry settings in ..\TOPCALL\TCPERFLOG.
8. Before modifying the registry settings stop TCMONITOR using the KCS Monitor. Make the changes and restart TCMONITOR using the KCS Monitor.

Chapter 3

Software Components / Resources Belonging to TC/PerfLog

Component/resource name	Type	Location	Function
tcterflog.exe	file	C:\topcall\shared	Process module for installing / operating / supervising of performance counter logging. Normally, this process is started by TCSR.V.
counters.txt	file	C:\tcoss\perfmon	Configuration file, plain text, contains the list of counters to be logged.
c:\tcoss\perflog	directory	-	Dedicated directory where the performance log files will be created. This directory will be created automatically by tcterflog.exe.
..\TOPCALL\TCPERFLOG	registry key & entries	HKLM\SOFTWARE\TOPCALL	Contains the configuration settings for tcterflog.exe
TCPERFLOG (Only Mode-1)	Windows "Performance logs and alerts" <i>custom logging task</i>		<i>Custom logging task</i> definition on the Windows "Performance Logs and Alerts" utility, created automatically by tcterflog.exe.
..\TOPCALL\Boot\Startup => TCPERFLOG	registry entry value	..\TOPCALL\Boot\Startup	TCSR.V configuration setting. It directs TCSR.V to start/supervise tcterflog.exe automatically.

Chapter 4

Functionality

TC/PerfLog provides in both modes (Mode-0, Mode-1) permanent performance counter logging and log file size supervising. However, this basic functionality will be achieved in slightly different ways, as described below.

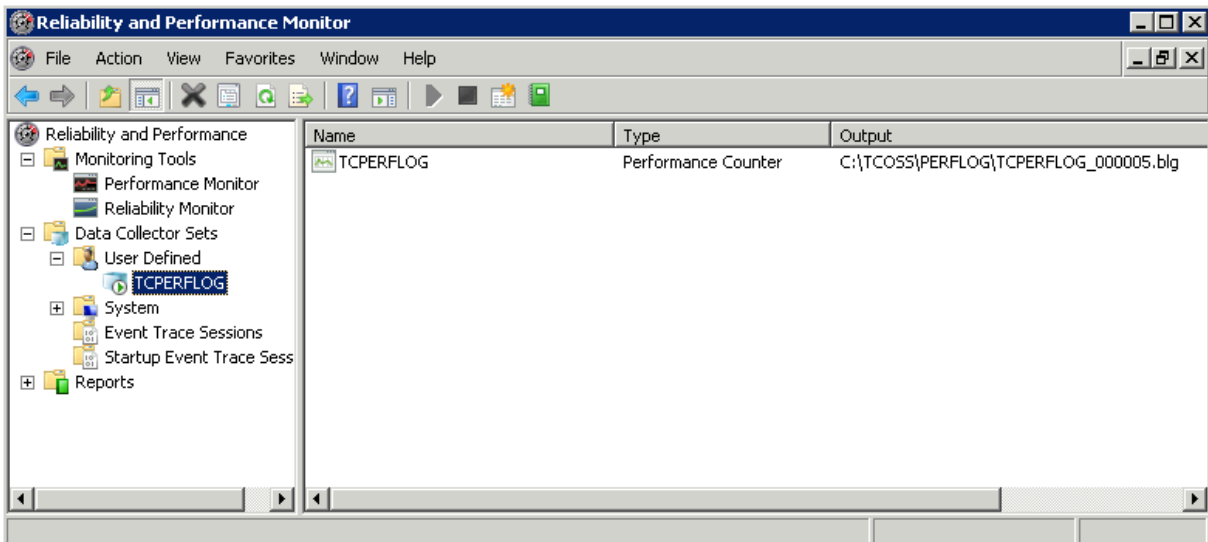
Mode-1: Logging by Custom Logging Task Supervising

In Mode-1 (default), TC/PerfLog controls the process of collecting and logging performance counters. TC/PerfLog itself does not write the performance log files, it manages/supervises only the logging process. It uses the standard Windows performance logging service to make create the log files. Users can define their own data collector sets (formerly called custom logging tasks). A custom logging task has its own configurable performance counter set, poll-period, log file directory/name, log file type, max. log file size, etc.

TC/PerfLog uses this configuration option of the Windows performance logging service to operate (create / (re)configure / start / stop) such a custom logging task for our logging purposes. (Note: Internally, TC/PerfLog uses the logman.exe command line interface to control the performance logging Windows service. logman.exe is part of the Windows, thus, there is no need to install it separately.)

With the Windows Reliability and Performance Monitor you can browse the user-defined data collector sets (custom logging tasks).

Windows Server 2008 32-bit:



TC/PerfLog offers the following functionality in Mode=1:

- On start, TC/PerfLog creates or updates (if it already exists) our custom logging task according to the configuration in the registry key `..\TOPCALL\TCPERFLOG` and in file counters.txt (this is the list of counters to be logged). After that, it starts this custom logging task.
TCPERFLOG (tcpperflog.exe) is started automatically by TCSRVR or manually by KCS Monitor. If the computer restarts, TC/PerfLog and the custom logging task restart as well.
- TC/PerfLog runs continually and checks from time to time (registry setting "SuperviseCycle" in seconds) whether our custom logging task is running. If not, TC/PerfLog tries to restart the task. Also, it checks the total size of created log files on the disk. When defined limit is exceeded (registry setting "LogSizeTotalMax" in MB), then the oldest logs are deleted. TC/PerfLog displays the current supervising status on the KCS Monitor display.
- If TC/PerfLog is stopped manually with KCS Monitor, then TC/PerfLog stops automatically the custom logging task. (This is necessary for changing the custom logging task configuration and also to view the current log file.)

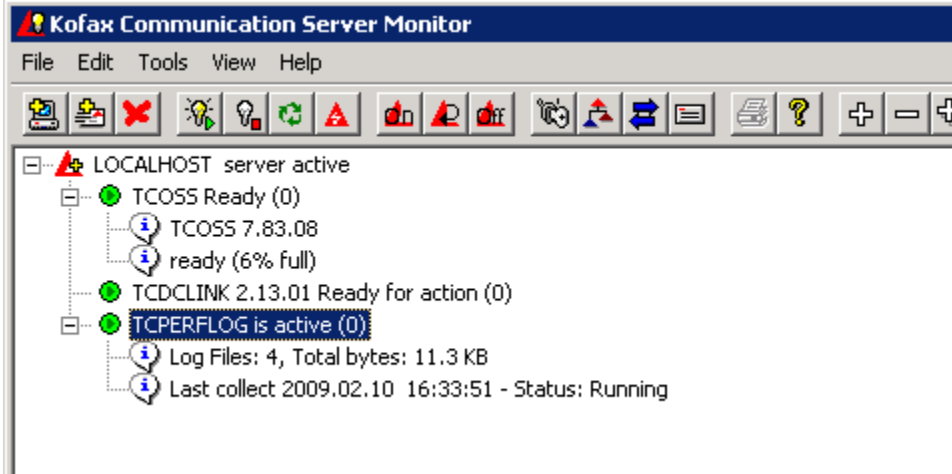
Mode-0: Creating / Writing / Supervising Log Files by TC/PerfLog

In Mode=0, TC/PerfLog directly creates and writes the performance log. Windows Performance Monitor does not show any custom logging tasks. The running state can be checked on the KCS Monitor display (but not in Windows Performance Monitor).

TC/PerfLog offers the following functionality in Mode=0:

- On start, TC/PerfLog starts collecting performance data and writing it to log file. (On every new start a new performance log file will be created, differing from the previous log file only by the 6 digit name-postfix, e.g. `TCPERFLOG_000001.blg => TCPERFLOG_000002.blg`).
TC/PerfLog (tcpperflog.exe) is started automatically by TCSRVR or manually by KCS Monitor. If the computer restarts, then TC/PerfLog restart automatically with TCSRVR.
- TC/PerfLog runs (and collects performance data) continually and checks from time to time (registry setting "SuperviseCycle") the total size of created log files on the disk. When defined limit is exceeded (registry setting "LogSizeTotalMax" in MB), then the oldest logs are deleted. TC/PerfLog displays its current collect status in the KCS Monitor.

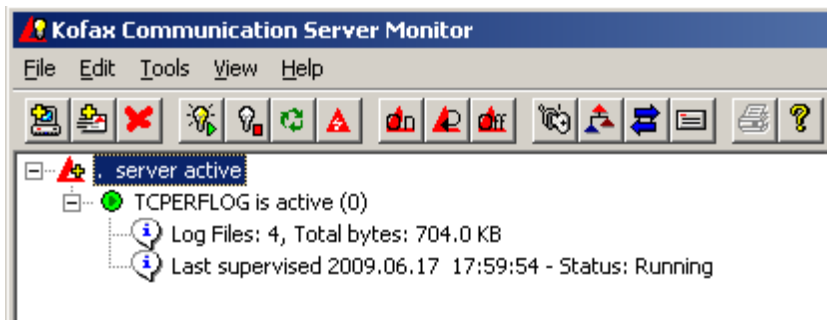
- If TC/PerfLog is stopped manually with KCS Monitor, then TC/PerfLog stops the logging and the log file is closed.



Chapter 5

Usage

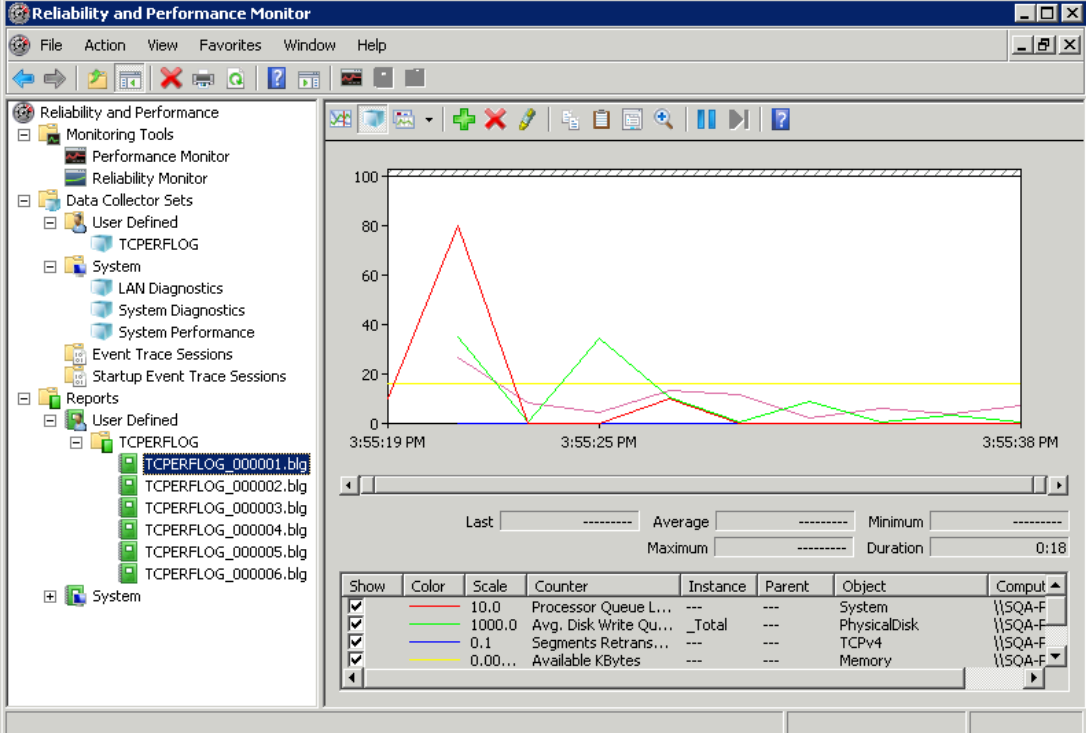
1. Install KCS with the TC/PerfLog option on the server computer.
2. Enter the counter names you want to log in the plain text file C:\tcoss\perfmon\counters.txt, one counter name in each line.
3. Start the computer with KCS Monitor. TCSRv process will start TC/PerfLog automatically. In Mode-1, TC/PerfLog installs (if not already exists) and starts the custom logging task automatically. After that, TC/PerfLog runs continuously and keeps the custom logging task alive, even when the computer restarts. The current supervising status is displayed on the KCS Monitor display.



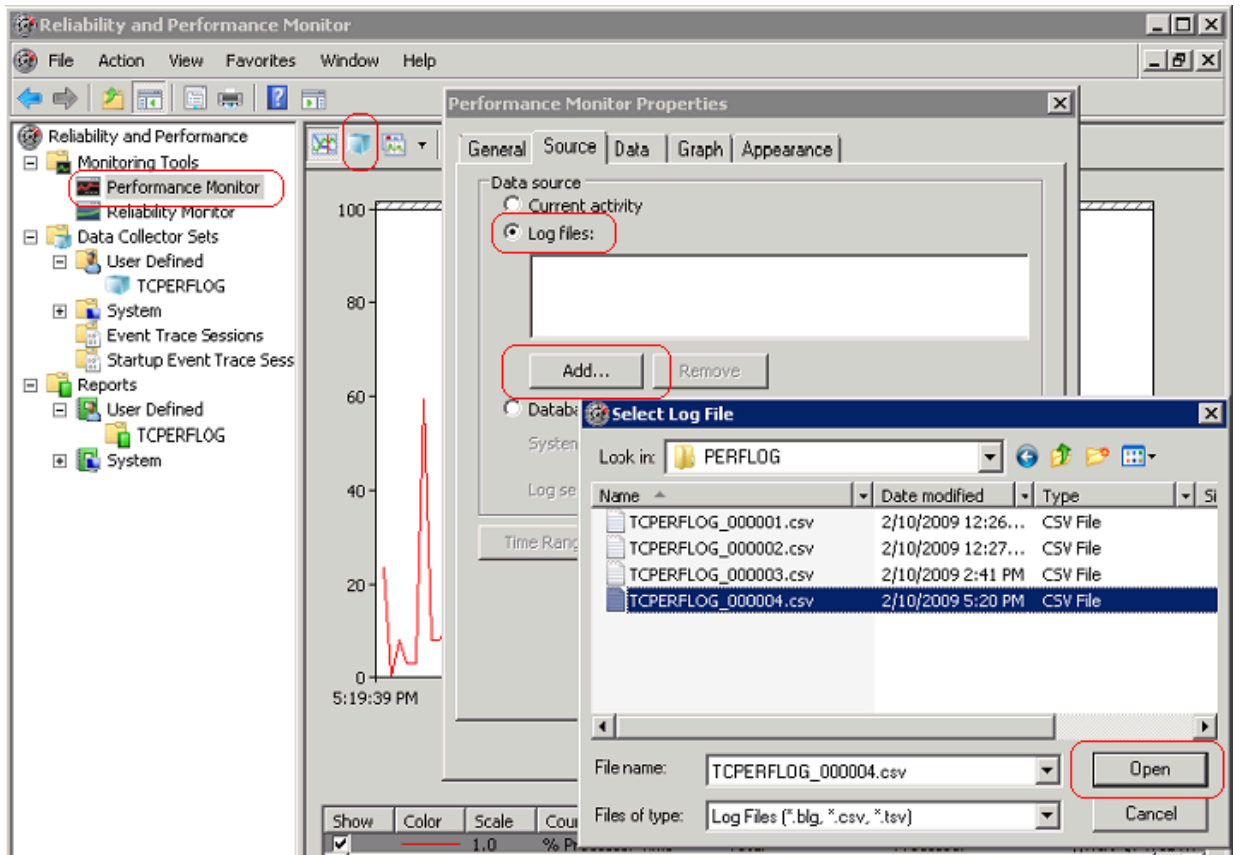
In Mode-0, the TC/PerfLog starts writing periodically the log file according the parameters in the registry and in the counter definition file.

4. Stop the TC/PerfLOG process with KCS Monitor to save all collected counters to the current log file.

- 5. You can use the standard Windows Performance Monitor utility to display the collected log file: Windows Server 2008 32-bit (only Mode-1):



Windows Server 2008 32-bit (both Mode-0 and Mode-1):



Chapter 6

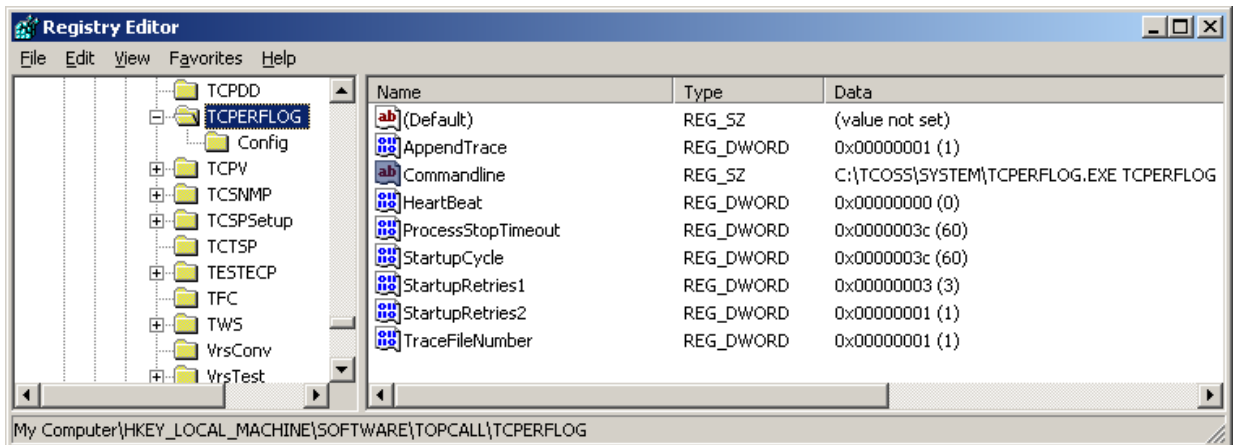
Configuration

The configuration occurs in 2 places: in the registry and in the counter configuration file (c:\tcoss\perfmon\counters.txt).

(Mode-1 and Mode-0 use the same parameters. There is no difference in the configuration.)

Configuration in the Registry

1. Registry setting ..\TOPCALL\Boot\Startup: "TCPERFLOG" must be entered here. The TCSP installation (if TCPERFLOG option was selected) enters it automatically, so normally no manual configuration is necessary here.
2. Settings under ..\TOPCALL\TCPERFLOG =>



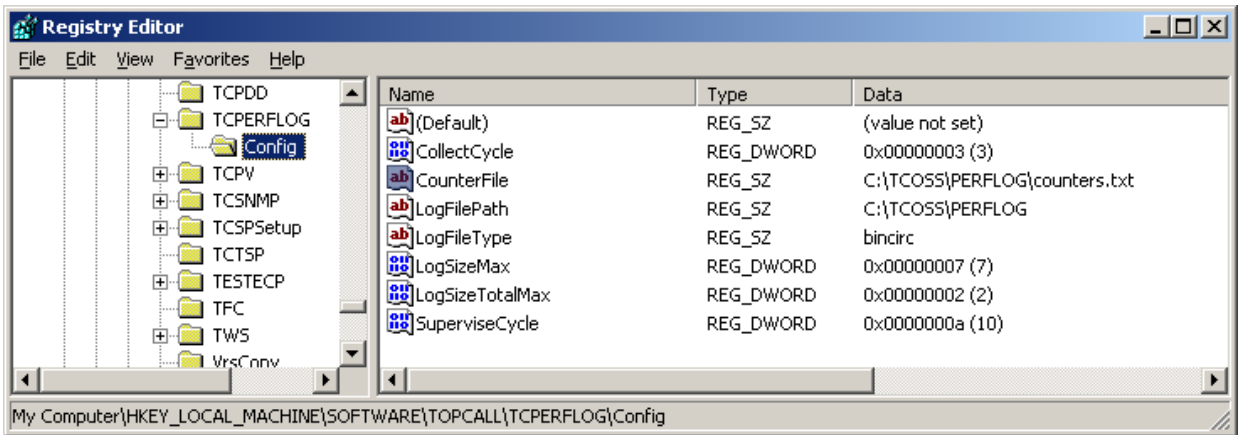
These are the standard TCSRv and TCLIB and settings.

Note The KCS "instance name" (default: "TCPERFLOG") is not fixed. You can use another name, e.g., "TCPERFLOG1". In this case, the following three changes must be made (manually) in the registry:

- Rename the registry key belonging to the instance to ..\TOPCALL\TCPERFLOG1
- TCSRv start configuration: in ..\TOPCALL\Boot\Startup must now contain "TCPERFLOG1"
- The ..\TOPCALL\TCPERFLOG1\CommandLine must now contain:
"c:\tcoss\system\tcperlog.exe TCPERFLOG1"

In this case, tcperlog.exe creates the custom logging task with the name "TCPERFLOG1".

3. Settings under ..\TOPCALL\TCPERFLOG\Config =>



Setting	Default	Explanation
CollectCycle (REG_DWORD)	30 (second)	Specifies sample intervals for performance counter collection (in seconds). TCMERFLOG passes this configuration value to Windows "Performance Log and Alerts".
CounterFile (REG_SZ)	C:\TCOSS \ PERFLOG \counters.txt	The counter configuration file which defines the performance counters to be logged.
LogFilePath (REG_SZ)	C:\TCOSS \PERFLOG	Directory, where the performance log file will be created. TCMERFLOG passes this configuration value to Windows "Performance Log and Alerts".
LogFileType (REG_SZ)	bincirc	Specifies the file format used for collecting performance counter. Possible values: <i>bin</i> – binary <i>bincirc</i> – binary circular <i>csv</i> – comma delimited text file <i>tsv</i> – tab delimited text file TCMERFLOG passes this configuration value to Windows "Performance Log and Alerts".
LogSizeMax (REG_DWORD)	4 (MB)	Size limit for the performance log file (in MB). TCMERFLOG passes this configuration value to Windows "Performance Log and Alerts".
LogSizeTotalMax (REG_DWORD)	40 (MB)	Limit for the total size of all log files. (Every new start of the <i>custom logging task</i> generates a new log file, so there can be many log files in the performance log directory. The total size of these files is limited to avoid disk overflow.)
Mode	1	0: (Mode-0) Logging by writing the log file by TC/PerfMon itself. 1: (Mode-1) Logging by Windows "custom logging task" supervising.
ShowLogmanStdOut		For Windows Server 2012 R2, set ShowLogmanStdOut registry value to 1.

Setting	Default	Explanation
SuperviseCycle (REG_DWORD)	120 (second)	TCPERFLOG checks in every <i>SuperviseCycle</i> seconds whether the "Performance Log and Alerts" <i>custom logging task</i> is running and whether the total size of all performance log files exceeds the configured LogSizeTotalMax value.

Counter Configuration File (counters.txt)

This configuration file defines the performance counters to be logged. The path / name of this file is configured in the registry setting `..\TOPCALL\TCPERFLOG\Config\CounterFile`. By default, this is: `C:\TCOSS\PERFLOG\counters.txt`

It is installed with a predefined set of counters. (Some general counters plus the KCS counters.)

If you want to log other performance counters, you can enter the names of these counters into `counters.txt`. The syntax is very simple:

1. Each line must contain exactly one performance counter name. (Leading / terminating white space characters will be ignored.)
2. Using the # character as the first non-white space character in a line, the whole line will be ignored. This method can be used to comment out some counters temporarily or insert explanatory comments.
3. All content after an empty line is ignored and can be used for comments.
4. The syntax required for the performance counter names are the same as the `logman.exe` requires in its `-c` parameter, that is: (excerpt from the `logman.exe` manual)

"http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/nt_command_logman.msp?mfr=true"

The general format for counter paths is as follows: [\\Computer]object[parent/instance#index]counter] where:
The parent, instance, index, and counter components of the format may contain either a valid name or a wildcard character. The computer, parent, instance, and index components are not necessary for all counters.

You determine the counter paths to use based on the counter itself. For example, the LogicalDisk object has an instance index, so you must provide the #index or a wildcard. Therefore, you could use the following format:

```
\\LogicalDisk(*/*#*)*
```

In comparison, the Process object does not require an instance index. Therefore, you could use the following format:

- \\Process(*)\\ID Process
- The following is a list of the possible formats:
- \\machine\\object(parent/instance#index)\\counter
- \\machine\\object(parent/instance)\\counter
- \\machine\\object(instance#index)\\counter
- \\machine\\object(instance)\\counter
- \\machine\\object\\counter
- \\object(parent/instance#index)\\counter
- \\object(parent/instance)\\counter
- \\object(instance#index)\\counter
- \\object(instance)\\counter
- \\object\\counter

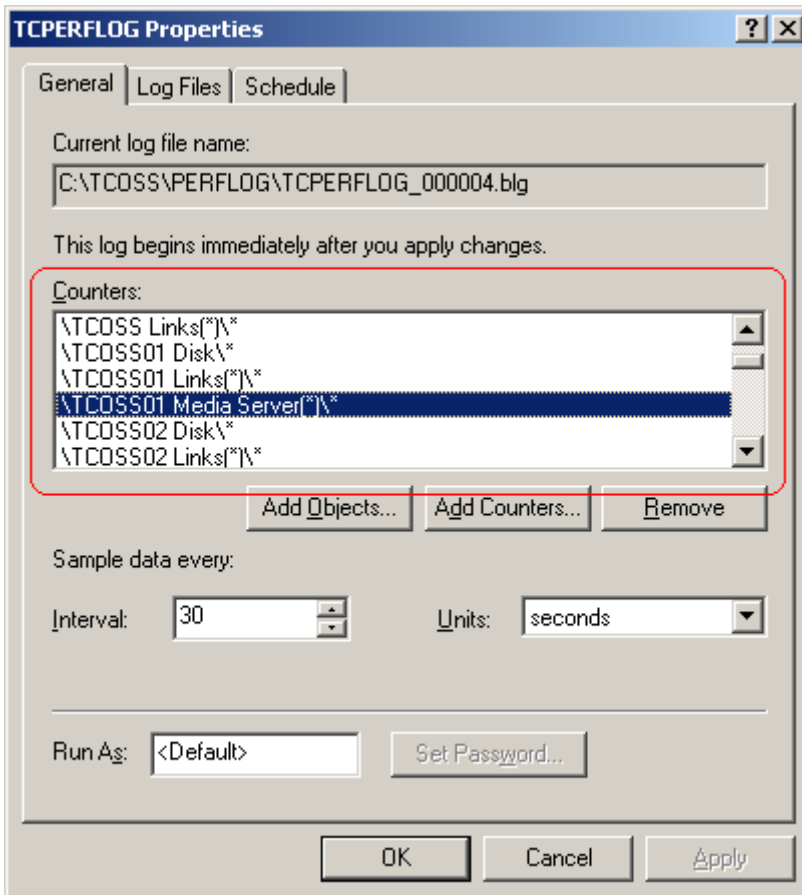
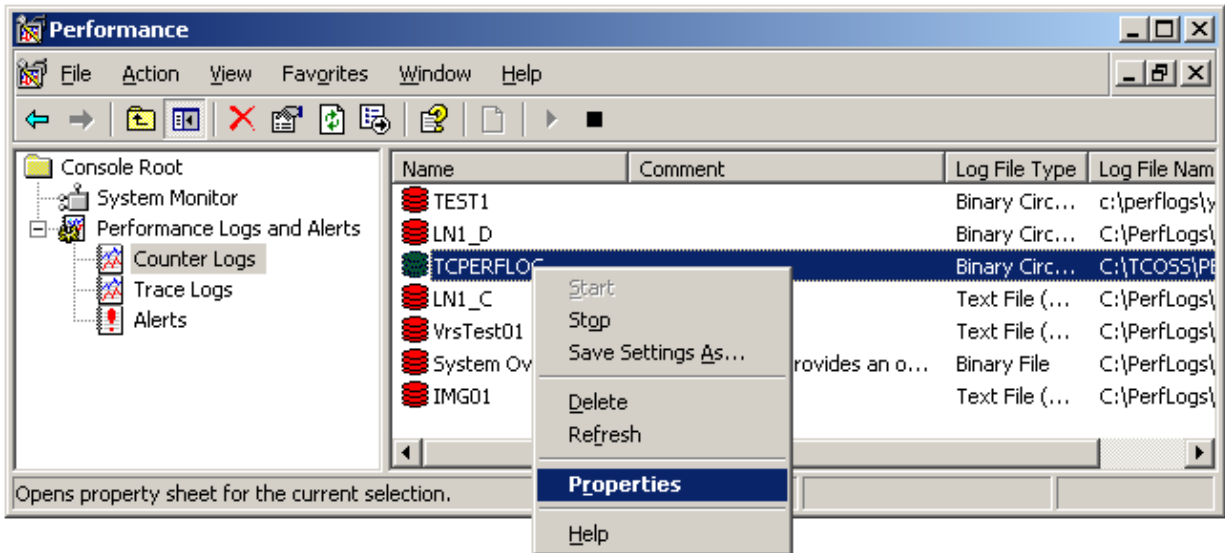
If a wildcard character is specified in the parent name, all instances of the specified object that match the specified instance and counter fields will be returned.

If a wildcard character is specified in the instance name, all instances of the specified object and parent object will be returned if all instance names corresponding to the specified index match the wildcard character.

If a wildcard character is specified in the counter name, all counters of the specified object are returned.

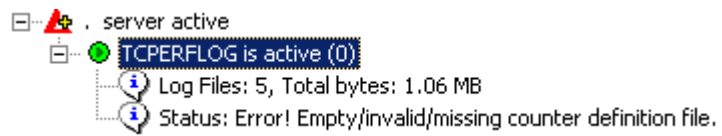
Partial counter path string matches (for example, pro*) are not supported.

Tip If you are not sure about the valid syntax of a certain performance counter name, then you can test it with the Windows Performance Monitor (perfmon.exe) tool. The list box “Counters” (see picture below) will contain the performance counter names with the desired syntax. So, if you make some tests via the “Add counters...” operation, then you can see in this list, what is the correct counter name syntax. #



Troubleshooting hint:

If the counter configuration file is missing/inaccessible/empty, then KCS Monitor displays the following error:



Chapter 7

Troubleshooting

This section describes how to troubleshoot TCMERFLOG.

Checking the Proper Operation

- Check in the KCS Monitor if the TCMERFLOG instance is started and “Status: Running” displayed.
- Check in the Windows Performance Monitor if the *custom logging task* “TCMERFLOG” exists and is running (green icon). Check the properties of the *custom logging task* “TCMERFLOG” (configured counter-set, file name, poll-cycle, log file name, log file type, max. file size) if it is the same as configured in the registry.
- Check the existence of performance log files in the C:\TCOSS\PERFLOG directory.
- Check the content of the standard KCS trace file belonging to the tcmperflog.exe process in C:\TCOSS\TRACE.

Chapter 8

Remarks

(Mode-1) The performance counter values are initially collected in the memory and it is not controlled, when exactly the writing to the disk occurs. Thus, the current log file size and content on the disk do not have to mirror the current collected size / content.