

The CLUE tool usage guide

By Clint Huffman ([clinth@microsoft.com](mailto:clinth@microsoft.com))

Contents

[What is CLUE tool? 3](#_Toc45126430)

[Features 3](#_Toc45126431)

[Prerequisites 3](#_Toc45126432)

[Download and Installation (administrator rights required) 3](#_Toc45126433)

[Silent installation 7](#_Toc45126434)

[Uninstallation (administrator rights required) 7](#_Toc45126435)

[CLUE Tasks in Task Scheduler 7](#_Toc45126436)

[CLUE Performance Monitor Data Collectors 9](#_Toc45126437)

[Config XML file 9](#_Toc45126438)

[PalCollector 10](#_Toc45126439)

[User initiated data collection 10](#_Toc45126440)

[Frequently Asked Questions 11](#_Toc45126441)

[Feedback and Support 11](#_Toc45126442)

## What is CLUE tool?

C.L.U.E. (Collection of Logs and the User Experience) is a fully automate data collection tool for performance problems that may cause system delays. It is best used on Microsoft Windows where the end user does not have administrator rights or to capture difficult problems on Windows Server.

## Features

* Triggered data collection during critical events with no human intervention.
* Configurable to collect any data through Powershell.
* Survives reboots. In the event that the server is rebooted, the Task Scheduler service automatically reinitializes CLUE data collectors.
* Gathers the optimal performance counters for Performance Analysis of Logs (PAL) analysis (<http://github.com/clinthuffman/pal>).
* Allows users without administrator rights to initiate data collection.

## Prerequisites

This tool is supported on x64 (64-bit) versions of Windows 10.

* Powershell 2.0 or later must be installed and functional. The “ByPass” feature is used to by-pass the Powershell execution policy. This means that there is no need to change the system’s Powershell execution policy, but administrator rights is needed during initial installation.
* Windows Task Scheduler
* Windows Performance Monitor

## Download and Installation (administrator rights required)

The latest version of the CLUE tool is available at <http://aka.ms/ClueTool>.

1. Download the latest CLUE\_\*.zip file.
2. After download, unblock the zip file. Right-click on the zip file and go to Properties. Check **Unblock**, and click OK.
3. Extract the contents of the zip file. This can be a network share ([\\server\share](file:///\\server\share)) or a local folder on the target system.
4. Optionally, edit the first line of config.xml to pre-populate setup answers. More information below.
5. Log on to the target system with administrator rights and run Setup.bat with administrator rights.

The setup might prompt you for an installation directory and/or an output directory if they are not already defined in config.xml.

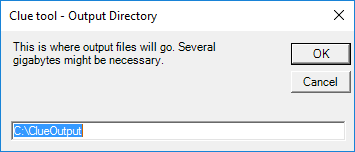
**Installation Directory**

This is where all of the CLUE related files will be copied. All CLUE operations will run from this directory. Once installed, the installation files are no longer needed. The size of this directory is relatively small (less than 100 MB). This is set to **C:\Program Files\Clue** by default, but can be changed (if needed) by setting or removing the **InstallationDirectory** attribute on the first line of the config.xml prior to running setup.

**Output Directory**

This is where all of the collected data will be saved. Files such as ETL and BLG files can be very large (often larger than 500 MB). Ensure that the logical disk hosting the Output Directory has enough free space to accommodate the potential data. Several gigabytes of disk capacity is recommended.

This folder is set to **C:\ClueOutput** by default. This default value can be changed in the config.xml file.

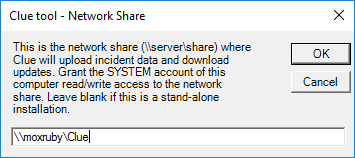


For a silent installation, edit config.xml and set the value of **OutputDirectory** to the file system location you desire.

**[OPTIONAL] Network Share**

To help with data collection from several computer systems, specify a network share for the data to be moved to as a convenience. This copies the collected data (\*.zip files) and deletes it from the local PC. This both frees up disk space on the PC and makes it easier for the network administrator to collect all of the data in once place.

This is the network share where the **\Microsoft\Windows\Clue\IncidentFolderManagement** scheduled task will move collected data periodically. It is transferred using Robocopy.exe at a throttled speed of roughly 100 KB/s in order to reduce network overhead.



For a silent installation, edit config.xml and set the value of **UploadNetworkShare** to the network share path that you desire.

The scheduled task is running as the System account. If a network share is used, then it is recommended to grant the computer account (SYSTEM) read/write permissions to the network share.

If a domain\user account is needed (not recommended due to how the credentials are stored), then after the CLUE tool is installed, use the following commands to change the credentials for the IncidentFolderManagement job. Use /S parameter to apply it to a remote system:

**schtasks /Change /TN \Microsoft\Windows\Clue\IncidentFolderManagement /RU MyDomain\MyUser /RP "*My password*"**

Where **“*My password*”** is the password of the account.

***WARNING****: Adding credentials to a scheduled job is a* ***high security risk*** *due to how the password is stored by the Task Scheduler. It is recommended to set the network share to allow the SYSTEM account write access to the share.*

The scheduled task must be restarted whenever a credentials change is made. To stop and restart the job, run the following commands:

**schtasks /End /TN \Microsoft\Windows\Clue\IncidentFolderManagement**

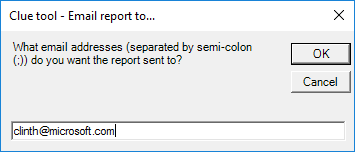
**schtasks /Run /TN \Microsoft\Windows\Clue\IncidentFolderManagement**

And then, check its status:

**schtasks /Query /TN \Microsoft\Windows\Clue\IncidentFolderManagement**

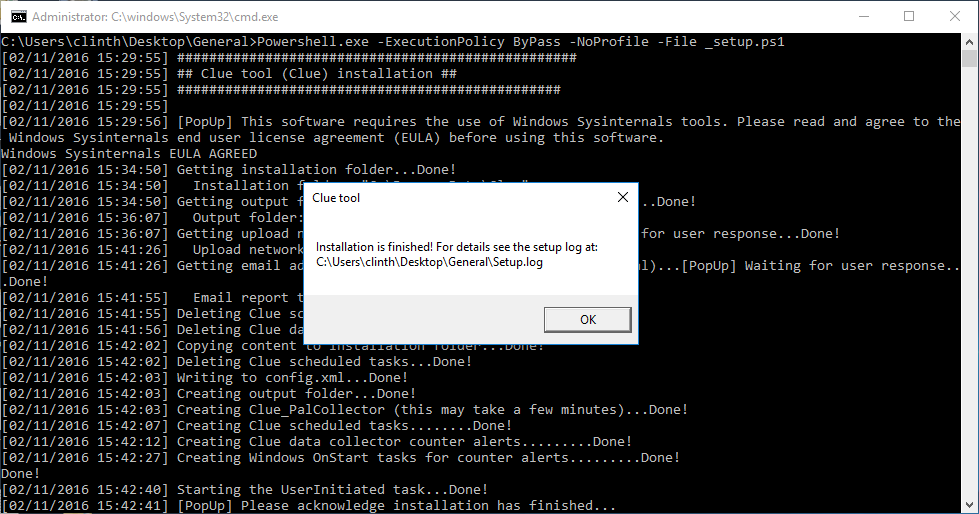
**EmailReportTo**

This is an optional field. This is the email address or email addresses (separated by semi-colons) that want to be emailed information if the data is analyzed. This only applies to the data collected from the local PC.



For a silent installation, edit config.xml and set the value of **EmailReportTo** to the email addresses you desire.

In less than a minute, a dialog box should show that installation was successful.



If Setup.bat fails to run from Windows Explorer, then open an administrator command prompt and run Setup.bat. Administrator rights is required in order to create performance counter data collectors and scheduled tasks.

If an error occurs, then look at Setup.log in the folder where Setup.bat was executed. If help is still needed, then report the error to the GitHub.com project at <http://github.com/clinthuffman/Clue> (preferred) or email me at [clinth@microsoft.com](mailto:clinth@microsoft.com) with a screenshot or detailed description of the exact error.

Once installation is complete, you may log off of the system. Data collection will be automatic. Periodically, watch the Output folder and/or Network Share folder for “incident” zip files.

**CollectionLevel**

CollectionLevel is how intensive that data collection is permitted to be.

**0** = No data collection

**1** = Minimal. Mostly just the performance counter log and other low overhead queries.

**2** = Moderate. This permit some low overhead ETW traces and other queries.

**3** = Full (recommended, default). Unrestricted data collection is permitted. A lot of effort went into reducing the amount of overhead while collecting the necessary data for root cause, this level might still impact the user experience. I recommend starting with this mode and reducing to 2 if the overhead is not acceptable.

Setup will set the registry key: **HKEY\_LOCAL\_MACHINE\SOFTWARE\Clue\CollectionLevel**

After setup, the registry key can be changed at anytime to a new level and will take effect immediately. For example, if done collecting data, but don’t want to uninstall CLUE, then set the registry key to 0.

## Silent installation

A “silent” installation will not prompt for anything. If installation fails such as not filling out config.xml, then see setup.log at **%temp%\ClueSetup** or the CLUE installation folder (**%programdata%\Clue** by default).

The silent option assumes that installation questions (attributes in the first node) have been answered in config.xml in the installation folder such as, but not limited to:

* InstallationDirectory
* OutputDirectory
* UploadNetworkShare
* EmailReportTo

The following is the first line of config.xml:

<Configuration InstallationDirectory="%ProgramFiles%\Clue" OutputDirectory="C:\ClueOutput" UploadNetworkShare="" EmailReportTo="clinth@microsoft.com" WptFolderPath="C:\WINDOWS\System32" CollectionLevel="3">

To install CLUE silently, fill in the attribute fields in config.xml and then run SetupSilent.bat with administrator rights. Check setup.log for success or failure.

## Uninstallation (administrator rights required)

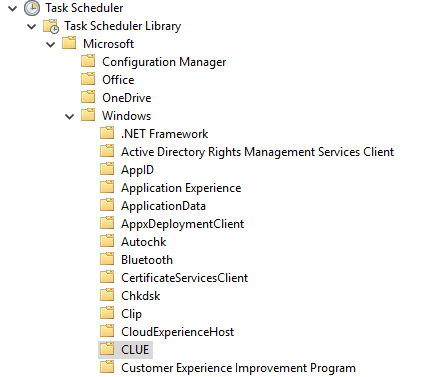
To remove or uninstall CLUE, run Uninstall.bat with administrator rights from the original zip file or from the CLUE installation folder. This action removes the Performance Monitor data collectors, scheduled tasks, and installation folder.

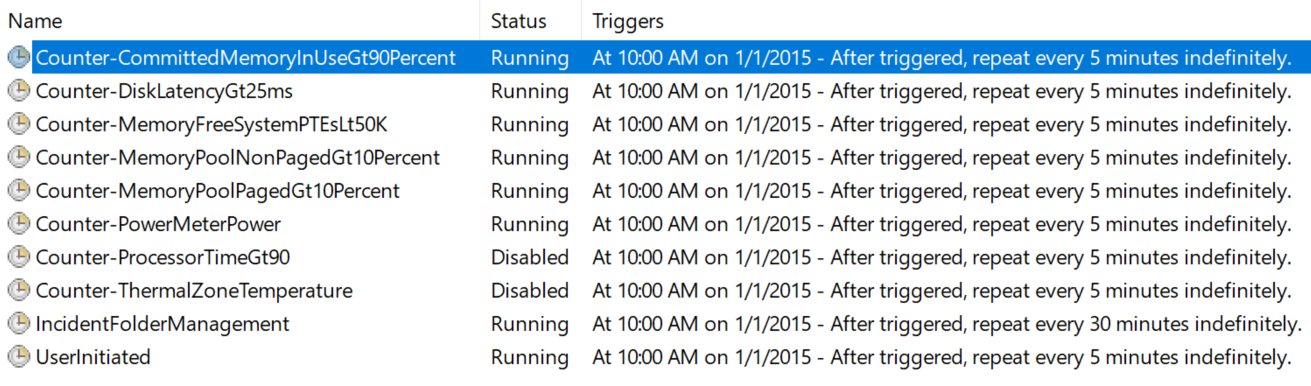
Note: Data in the Output folder will remain untouched to prevent data loss. This could be a significant amount of disk usage, so check it while uninstalling.

## CLUE Tasks in Task Scheduler

CLUE uses the Task Scheduler almost exclusively to collect data and survive reboots.

Open Windows Task Manager (part of the operating system) and navigate to **\Microsoft\Windows\Clue** folder.





The installation of CLUE creates tasks at this location only.

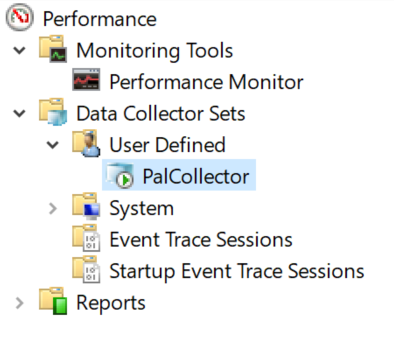
All scheduled tasks should be in a Ready or Running state except for **IncidentFolderManagement** which starts up 5 minutes after boot and any other rule that is not set to start immediately.

**IncidentFolderManagement** compresses resultant incident folders into zip files and moves completed CLUE incident data to the designated network share, if applicable. Otherwise, the data remains in the output folder.

**UserInitiated** watches **%public%\Documents\ClueUserInitiated.txt** for content changes. When any content is added to the text file, then it start data collection and then deletes the content within the file.

## CLUE Performance Monitor Data Collectors

CLUE creates a Performance Monitor data collector called, PalCollector, to collect performance counter data. To see it, open **Performance Monitor** by running “perfmon.exe”.



**PalCollector** is a performance counter data collector set with the performance counters detected on the local system and optimized for analysis with the [PAL](http://github.com/clinthuffman/pal) tool. This means that if the system has Microsoft Exchange Server and a named instance of SQL Server installed, then performance counters for the operating system, and both products will automatically be added. In addition, the data collector is configured to be a binary circular log file of 50 MB – meaning it will never be larger than 50 MB. Furthermore, a scheduled task is created to automatically start this data collector set upon the start of Windows allowing it to “survive” reboots.

## Config XML file

The config.xml file is the heart of the tool in which the rest of the scripts and tasks follow.

Changes to Config.xml must be done in the setup folder and setup must be ran anytime changes are made to config.xml. Directly editing config.xml in the installation folder is permitted, but all of the CLUE related Scheduled Tasks will require restarting.

**Rule node:** This is a rule which defines what to collect and how to collect it.

* **Type:** Currently only Counter type is implemented. Counter type collect performance counter data. Other types such as Event Log, WMI, and ETW event subscriptions are being considered as new features.
* **Enabled:** Must be True or False. If this value is not True, then it is skipped during installation. If set to True, or any other value changes in the rule then setup will need to be ran again.
* **Name:** This is the name of the Rule used in the naming convention of the output zip files. This must be unique compared to all of the other rule names.
* **Ran:** Set to zero (0) by default. This is used to track how many times this rule has been executed and is used to compare against the RunLimit.
* **RunLimit:** This is how many times this rule is permitted to run. If Ran reaches the RunLimit, then the scheduled task associated with this rule is Disabled. ReEnabling the rule will reset the Ran value to zero (0).
* **CounterPath:** This is the counter definition path of which counter instances to collect. It must be defined as **\*Object*(*Instance*)\*Counter*** where instance can explicitly be the name of a single counter instance, all counter instances (\*), or counter instances that meet the filter such as **\Process(WmiPrvSE\*)\% Processor Time** which will collect all counter instances that start with **WmiPrvSE**.
* **Exclude:** If all counter instances (\*) is defined, then this field will filter out any instances that match this name. This is often used to remove the **\_Total** instance.
* **SampleInterval:** This is how often in seconds that the defined counters will be collected and then compared to the threshold.
* **MaxSamples:** This is how many samples of the counters will be collected before the collected data is compared to the threshold. For example, if the sample interval is 1 second and MaxSamples is 5, then the defined counters are collected every 1 second for 5 seconds, then compared to the threshold as a group.
* **Operator:** This uses the Powershell operators (Example: “gt” is greater than) comparing the values of the defined counters to the value of the Threshold attribute.
* **Threshold:** This must be a valid double value and must be a positive number. This value is compared against the collected values and the operator to determine if the threshold has been broken. If any of the instances exceeds the threshold, then the action nodes are executed.
* **OnStartActions:** The names of each Action nodes separated by commas (,) to execute immediately.
* **OnEndActions:** The names of each Action nodes separated by commas (,) to execute after all of the counter instances are below the threshold or the **MaxTraceTimeInSeconds** time has been reached, whichever comes first.
* **MaxTraceTimeInSeconds:** This is the time in seconds that the **OnStartActions** are permitted to run. If this time is exceeded, then **OnEndActions** are executed.
* **StartImmediately:** If set to True, then the scheduled task associated with this rule is started immediately after setup. Note: This does not apply to a reboot. The scheduled tasks are setup to run 5 min after boot or 5 min if the task crashes.

## PalCollector

The Performance Analysis of Logs (PAL) tool is a popular open source project that is analyzes performance counter logs. CLUE uses the threshold files of the PAL tool to create a Performance Monitor data collector that optimizes the analysis of the PAL tool by collection every performance counter that could be analyzed in the PAL tool.

During setup of CLUE, the PalCollector.ps1 script executes and creates a performance counter data collector as a binary circular log file – meaning this data collector can run indefinitely and not cause the system to run out of disk space because it is hard coded to a specific size.

For more information, see the PAL tool at <http://pal.codeplex.com>.

## User initiated data collection

CLUE has automated data collectors, but there may be an application “hang” situation that needs to be captured. Any user such as non-administrator users can initiate general data collection. Simply run the batch file that should be on the Desktop folder. It will modify the **%public%\Documents\ClueUserInitiated.txt** which will trigger the **UserInitiated** rule to execute.

Once the change is saved, data collection should be immediate. Check the output folder for an incident folder with **UserInitiated** in the folder name.

## Frequently Asked Questions

**If the PC is rebooted, will CLUE continue to function?**

Yes, CLUE creates scheduled tasks to restart itself in the event of a reboot.

**How do I remove this tool when done?**

Run **Uninstall.bat** using an administrator command prompt.

**How do I prevent CLUE from creating too many files?**

Set the RunLimit attribute on the respective Action nodes to the number of times you wish it to run. This is reset when the respective scheduled task is started or restarted.

## Feedback and Support

CLUE is not supported by Microsoft. Please contact me directly for support at [clinth@microsoft.com](mailto:clinth@microsoft.com).