Citrix CDFMonitor
Reference Guide

TABLE OF CONTENTS

Description 4
Requirements 5
Security 6
Install/uninstall 7
Activity 8
Modules 9
  Module Source Options 10
  Module Source Filtering 11
  Creating a control file (.ctl) 12
  Enable Modules by Filter 13
Logging 14
  Logging Options 15
  CDFMonitor Log File 16
  Output Trace File 17
  Log File Server 18
Network 20
  Network Options 22
  Monitoring CDFMonitor network clients 23
  Monitoring CDFMonitor client trace messages 24
  CDFMonitor client tracing to network 26
Match 28
  Matching Options 29
  TMF (Trace Message Format) files 30
  RegexPattern 31
Action 34
  Execute on match 35
  Startup and Shutdown Commands 37
Notify 39
  Example Notify Configuration 40
Upload 41
  Example Upload Configuration 42

Remote 43
Remote Options 44
Example Remote Deploy Configuration 45
Edit Remote Config 46
Options 48
Options Local 49
Options Advanced 50
Options Miscellaneous 52
Register File Type Association (FTA) 54
Output 57
Display Filter 58
Syntax Highlighting 59
Show Stats / Activity Summary 60
Command Line Console Usage 63
Example output from TraceToEtl 64
Example output from RegexParseToCsv 65
Command line display help ‘/?’ for operators 66
Operators 67
Operator Definitions 67
Configuration File 69
Config File Definitions 70
Default Configuration 78
Central Configuration 81
Walkthroughs 82
Walkthrough: Capturing a trace to .csv file 82
Walkthrough: Capturing a trace to .etl file 85
Walkthrough: Using events to capture a trace to .etl file 87
Walkthrough: Capture remote trace over network 91
Walkthrough: Formatting a native .etl file into .csv 93
Walkthrough: Using console / command line mode 95
Walkthrough: Formatting multiple native .etl files into .csv 96
Walkthrough: Creating a remote configuration 99
Walkthrough: Deploying a remote configuration 104

Walkthrough: Gathering data from a remote configuration 107
Walkthrough: Undeploying a remote configuration 109

DESCRIPTION

CDFMonitor is a small, lightweight, multipurpose utility used in conjunction with Citrix Diagnostics Facility (CDF) tracing to troubleshoot as well as monitor Citrix products. Some of the features are the ability to monitor for a certain trace message, execute an action for a certain trace message, and notify when a message occurs. It can run as a user process or as a service from either a console or from a GUI. Built-in to the utility is the ability to parse and view captured trace files saved in .etl native format using Citrix TMF URL. For viewing trace messages real-time or post capture, the Output tab optionally displays all messages and can be filtered.

All configuration information is stored in a .net exe.config file for ease of setup and deployment. The utility has many possible configurations but only the appropriate ones are enabled based on the activity that is currently selected. There are two views, simple and advanced, as depending on the activity not all options are necessary.

For logging of traces, there are many options including local, remote, or network. A permanent service can be installed using a ring buffer as a long term logging solution for example. Some configurations have requirements that restrict or prohibit data logging on the local device. Using CdfMonitor can alleviate these issues by filtering what data gets logged in addition to sending trace information directly over the network to a 'server' instance.

REQUIREMENTS

The software requirements for CDFMonitor are:

- Windows OS:
  - XP
  - 2003
  - Vista
  - 2008
  - 7
  - 8
  - 2012
- Microsoft .Net 3.5 or greater
- Microsoft WMI (for remote operations)
- Access to admin$ share (for remote operations)

SECURITY

The information below contains information about CDFMonitor from a Security perspective.

- CDFMonitor by default does not modify the operating system or file structure.
- CDFMonitor requires administrative rights to run due to Microsoft Event Tracing for Windows (ETW) requirements which CDF is based on.
- CDFMonitor can optionally be configured to run as a service with default credentials set to ‘NT_AUTHORITY\SYSTEM’.
    - This is because of administrative rights required for ETW/CDF tracing.
    - The service can optionally be installed to use alternate credentials supplied for **Use Credentials** if **Use Service Credentials is checked**. See Options section for additional information.
    - The service can also be modified to use alternate credentials manually but will still need to have rights to trace.
    - Service can be uninstalled locally remotely from utility.
- Remote connections and ALL remote management commands are made using Windows Management Instrumentation (WMI) calls including service creation and service logon.
- Remote management copying of files to and from remote machines running CDFMonitor is performed using the Windows ‘Admin$’ share.
- Traces can optionally be sent over the network using UDP protocol but does not do this by default. Traces are sent clear text as GUI states.
- No credentials are stored in config file EXCEPT optionally SMTP and ftp/http URL upload credentials as GUI states.
- All other alternate credentials would come from Windows Credential Manager.
- CDFMonitor can optionally use a stored ‘Utility’ credential that is stored in Windows Credential Manager under name 'CDFMonitor'. It does not do this by default and only if user chooses to do so will this occur. This credential can be removed manually from Windows Credential Manager or by unchecking **Use Credentials** on Options tab.
- SSL connection/authentication does not verify cert as an issue would not be known until needed and extra maintenance/debugging. This does allow to take advantage of encryption however just assumes cert and therefore server is correct.
- No information is stored in the registry except optional File type association. This information can be added or removed via **Register FTA** buttons on Option tab.

INSTALL/UNINSTALL

CDFMonitor is an executable with a configuration file that can be started directly and therefore has no installer. By default, nothing is modified on the machine. See Security for additional information. Optionally CDFMonitor can be installed as a service and file type association can be configured for .etl files which both of which modify the registry.

See Options Local for additional information.

To Install CDFMonitor as a service:

1. Start CDFMonitor.exe
2. Select Options tab
3. Select **Install Service**

To Uninstall CDFMonitor as a service:

1. Start CDFMonitor.exe
2. Select Options tab
3. Select **Uninstall Service**

To Register File Type Association

1. Start CDFMonitor.exe
2. Select Options tab and maximize window
3. Select **Register FTA**

To Unregister File Type Association

1. Start CDFMonitor.exe
2. Select Options tab and maximize window
3. Select **Unregister FTA**

ACTIVITY

The Activity tab is the main tab that is shown when CDFMonitor starts. An activity should be selected first before modifying other properties. Based on selected activity, different options in the utility will be available.

- **Client: Capture local CDF trace to .etl and parse later for best performance** should be used by default as it has best performance if monitoring or viewing trace messages real-time is not required. Traces will have to be formatted before viewing.
- **Client: Capture local CDF trace to .csv and parse real-time for trace message monitoring** is used to view trace messages real-time or 'matching' trace messages for actions or notifications. Parsing traces real-time does incur additional overhead.
- **Server: Capture remote trace messages and/or monitor remote cdfmonitor clients** is a 'server' activity used to listen on network Udp port for incoming Udp messages from remote CdfMonitor 'clients'.
- **Server: Remote management of cdfmonitor clients** is an advanced option that enables the ability to manage remote instances of CDFMonitor utility. Through the use of WMI and SMB, remote instances can be deployed, data gathered, and un-deployed from this console. See Remote for additional information.
- **Format trace: Parse existing .etl trace to .csv and optionally filter for trace message** is used to parse .etl traces which is the native file format into csv format for analysis. To parse etl files, TMF files are required unless it is a .net module.

NOTE: Always test performance of configuration before deploying to environment. Configuration for example module selection, resources on the machine, and current activities being performed on that machine can all affect performance.

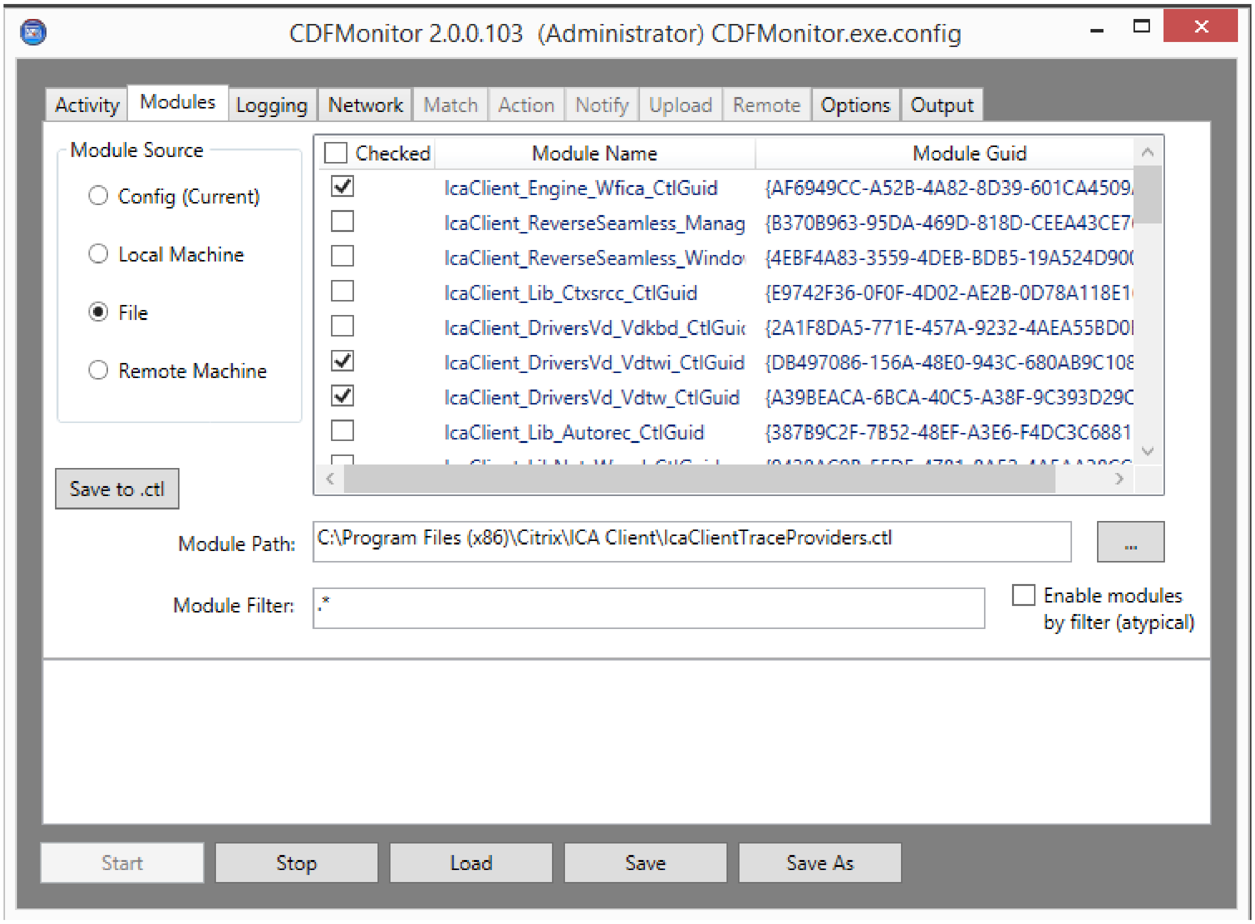
Detailed Activity Walkthrough information:

- Walkthrough: Capturing a trace to .csv file
- Walkthrough: Capturing a trace to .etl file
- Walkthrough: Capture remote trace over network
- Walkthrough: Formatting a native .etl file into .csv
- Walkthrough: Formatting multiple native .etl files into .csv

MODULES

The Modules tab is only used when capturing real-time traces and is not used for parsing or server activities. The modules list by default enumerates Citrix modules out of the registry for components installed on the local machine. Not all Citrix components use the registry to store module GUIDs. A .ctl file containing the module GUID and module name can also be specified. Depending on the type of information to be gathered, certain or possibly all modules would be selected. Guidance on selection of these modules is outside the scope of this document.

- **Module Source** is used to select the source from where to enumerate the list of module GUIds for the two capture activities. See Module Source Options
- **Save to .ctl** will save the checked 'Module Name' and 'Module GUID' into specified .ctl text file. See Creating a control file (.ctl)
- **Module Path** is used for file name and path when '**File**' is selected for Module Source. **Module Path** is used for IP address, FQDN, or name of remote machine when '**Remote Machine**' is selected.
- **Module Filter** is used to filter current module view window. See Enable Modules by Filter
- **Enable modules by filter** is used for selection of modules by **Module Filter**



Module Source
○ Config (Current)
○ Local Machine
● File
○ Remote Machine

□ Checked	Module Name	Module Guid
☑	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82-8D39-601CA4509
□	IcaClient_ReverseSeamless_Manag	{B370B963-95DA-469D-818D-CEEA43CE7
□	IcaClient_ReverseSeamless_Windo	{4EBF4A83-3559-4DEB-BDB5-19A524D90
□	IcaClient_Lib_Ctxsrcc_CtlGuid	{E9742F36-0F0F-4D02-AE2B-0D78A118E1
□	IcaClient_DriversVd_Vdkbd_CtlGui	{2A1F8DA5-771E-457A-9232-4AEA55BD0
☑	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-943C-680AB9C10
☑	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5-A38F-9C393D29
□	IcaClient_Lib_Autorec_CtlGuid	{387B9C2F-7B52-48EF-A3E6-F4DC3C6881

Module Path: C:\Program Files (x86)\Citrix\ICA Client\IcaClientTraceProviders.ctl

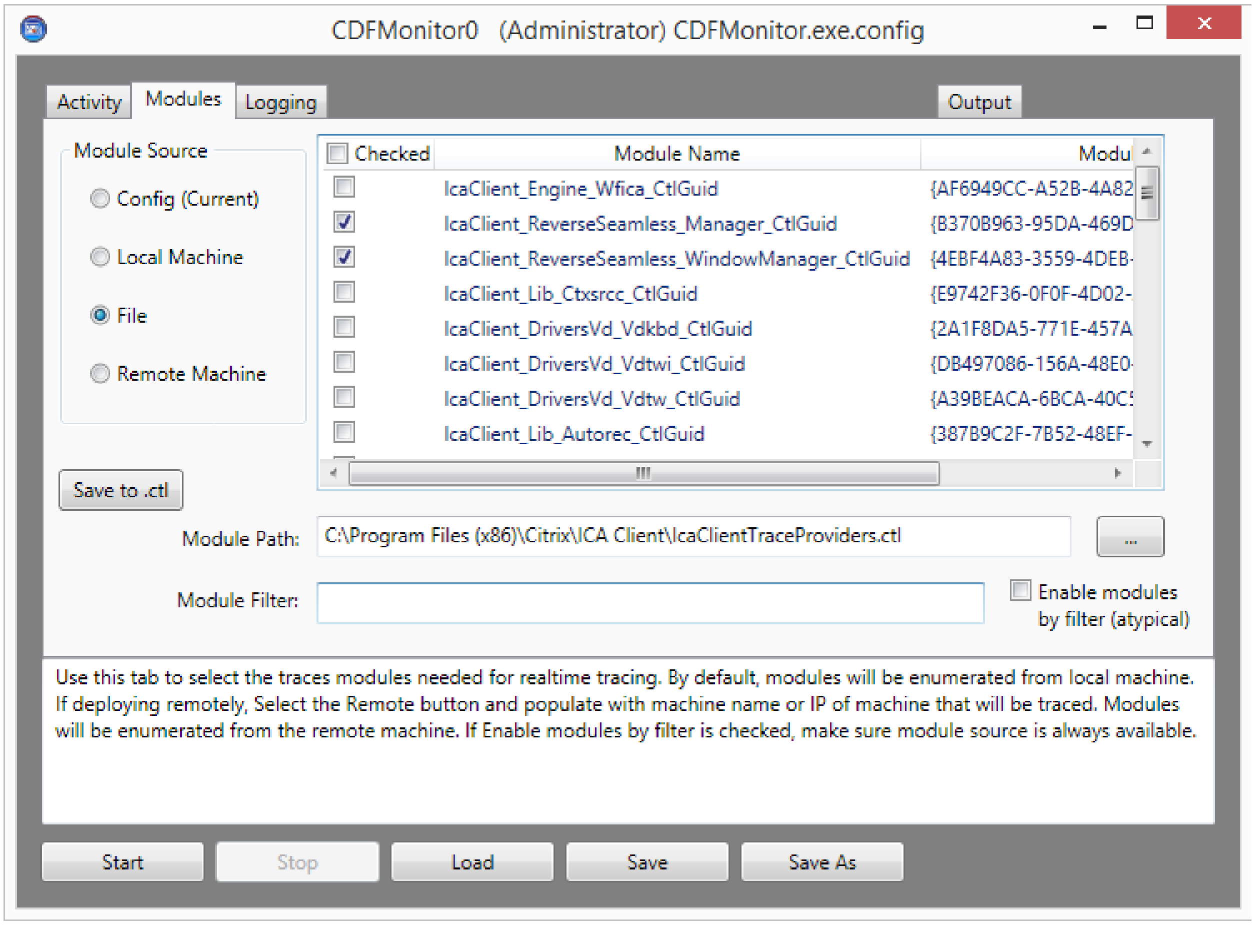
Module Filter: .*

□ Enable modules by filter (atypical)

Module Source Options

Module Source Options are the radio buttons on the left that enumerates the different available module sources. Modules can be loaded from configuration file, registry, file, or remote machine registry.

- **Config (Current)** is the module list currently saved in the config file. When a new configuration is saved, regardless of source, the module source will be switched to Config (Current) as it is now the current configuration.
- **Local Machine** is the registered Citrix module list for installed components on the local machine. This information is enumerated from the registry. Not all Citrix components register their modules in the registry.
- **File** is to load module list from a file. This is typically a .ctl file in %GUID% %name% format.
- **Remote Machine** is the registered Citrix module list on the specified machine in the Module Path field. This information is enumerated through WMI. Alternate credentials can be enabled from the options tab.



Module Source

○ Config (Current)
○ Local Machine
● File
○ Remote Machine

☐ Checked	Module Name	Modu
☐	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82
☑	IcaClient_ReverseSeamless_Manager_CtlGuid	{B370B963-95DA-469D
☑	IcaClient_ReverseSeamless_WindowManager_CtlGuid	{4EBF4A83-3559-4DEB-
☐	IcaClient_Lib_Ctxsrcc_CtlGuid	{E9742F36-0F0F-4D02-
☐	IcaClient_DriversVd_Vdkbd_CtlGuid	{2A1F8DA5-771E-457A
☐	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-
☐	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5
☐	IcaClient_Lib_Autorec_CtlGuid	{387B9C2F-7B52-48EF-

Module Path: C:\Program Files (x86)\Citrix\ICA Client\IcaClientTraceProviders.ctl

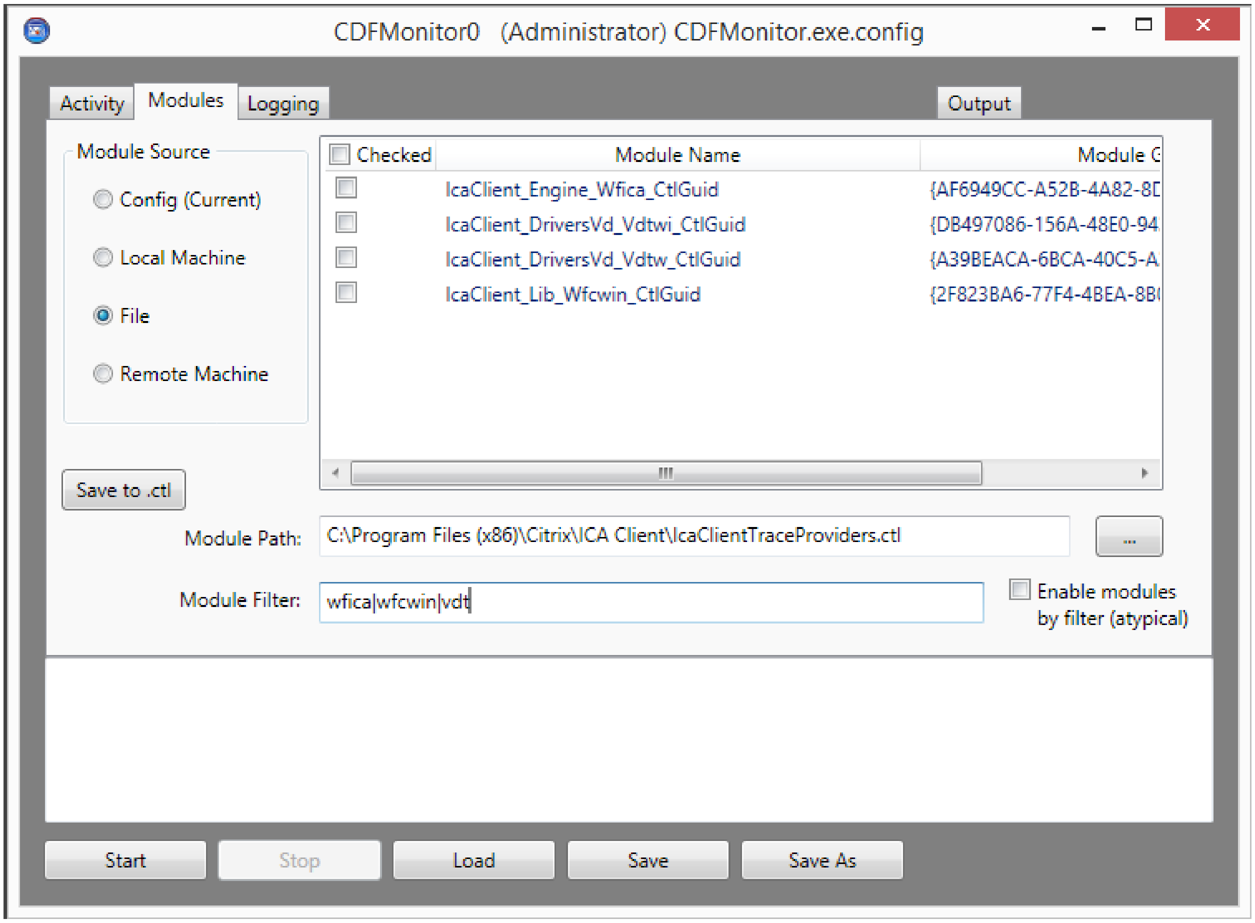
Module Filter:

☐ Enable modules by filter (atypical)

Use this tab to select the traces modules needed for realtime tracing. By default, modules will be enumerated from local machine. If deploying remotely, Select the Remote button and populate with machine name or IP of machine that will be traced. Modules will be enumerated from the remote machine. If Enable modules by filter is checked, make sure module source is always available.

Module Source Filtering

Module Filter is not required but is helpful when selecting like named modules out of a large list. Module Filter will filter the values in the displayed module list using regular expressions. Only values that are visible in the displayed module list will be saved inside the configuration file and optionally enabled. Module filtering can optionally be used to enable modules as well. See Enable Modules by Filter for additional information.



CDFMonitor0 (Administrator) CDFMonitor.exe.config

Module Source

○ Config (Current)
○ Local Machine
● File
○ Remote Machine

□ Checked	Module Name	Module G
□	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82-8D
□	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-94
□	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5-A
□	IcaClient_Lib_Wfcwin_CtlGuid	{2F823BA6-77F4-4BEA-8B

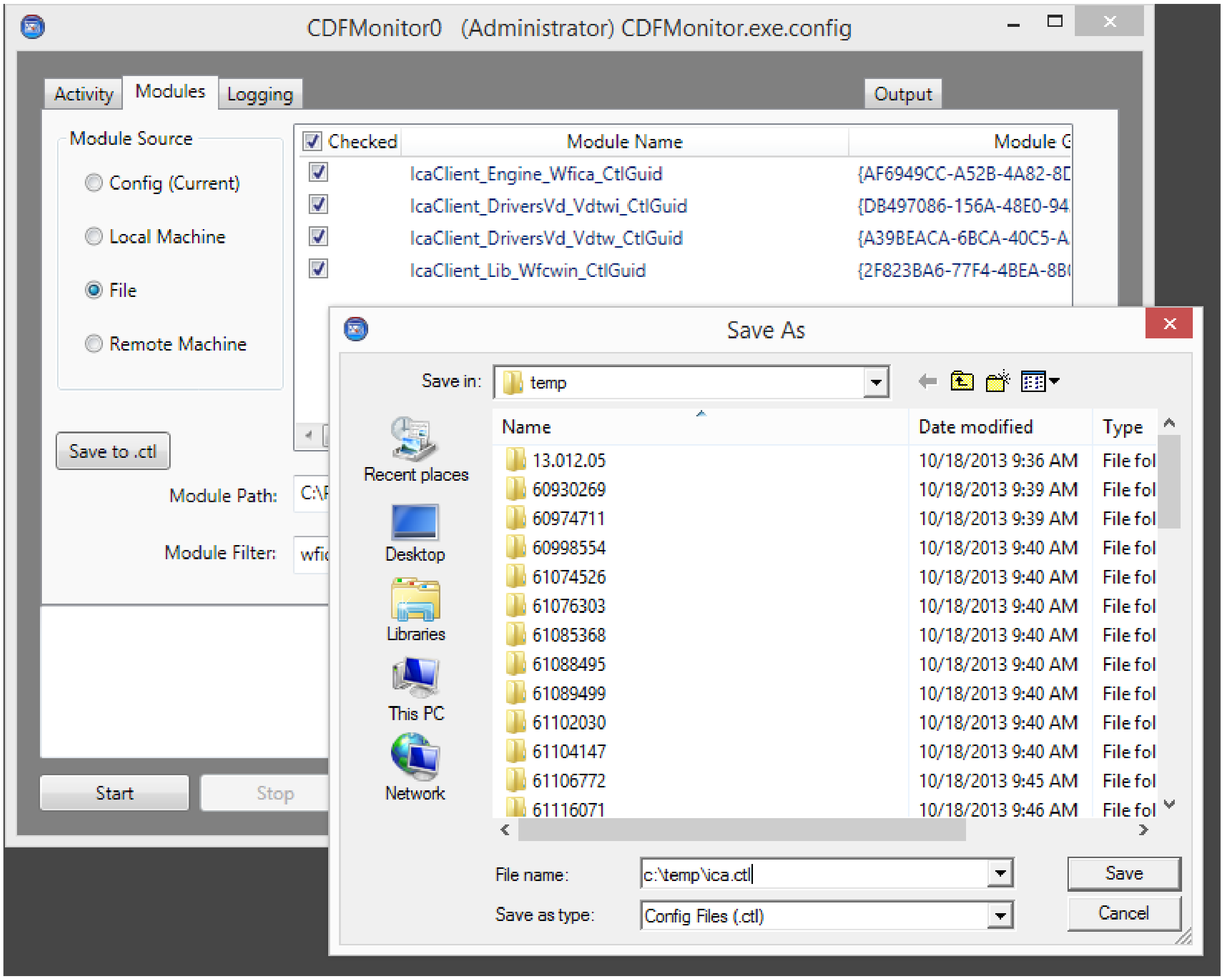
Module Path: C:\Program Files (x86)\Citrix\ICA Client\IcaClientTraceProviders.ctl

Module Filter: wfica|wfcwin|vdt

□ Enable modules by filter (atypical)

Creating a control file (.ctl)

Creating a control file is useful if there is a need to store the module list in an external file outside of the config file. The config file can point to the .ctl file and use the modules from that file directly. Creating a .ctl file is also useful if tracing from other ETW utilities for example CDFControl as it is a standard format. Only the checked modules will be saved as everything listed in the .ctl file would typically be enabled. When selecting **Save to .ctl** button, a dialog prompt will be displayed for path and name.



Module Source

○ Config (Current)

○ Local Machine

● File

○ Remote Machine

☑ Checked	Module Name	Module G
☑	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82-8D
☑	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-94
☑	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5-A
☑	IcaClient_Lib_Wfcwin_CtlGuid	{2F823BA6-77F4-4BEA-8B

Module Path: C:\F

Module Filter: wfic

Save in: temp

Name	Date modified	Type
13.012.05	10/18/2013 9:36 AM	File fol
60930269	10/18/2013 9:39 AM	File fol
60974711	10/18/2013 9:39 AM	File fol
60998554	10/18/2013 9:40 AM	File fol
61074526	10/18/2013 9:40 AM	File fol
61076303	10/18/2013 9:40 AM	File fol
61085368	10/18/2013 9:40 AM	File fol
61088495	10/18/2013 9:40 AM	File fol
61089499	10/18/2013 9:40 AM	File fol
61102030	10/18/2013 9:40 AM	File fol
61104147	10/18/2013 9:40 AM	File fol
61106772	10/18/2013 9:45 AM	File fol
61116071	10/18/2013 9:46 AM	File fol

File name: c:\temp\ica.ctl

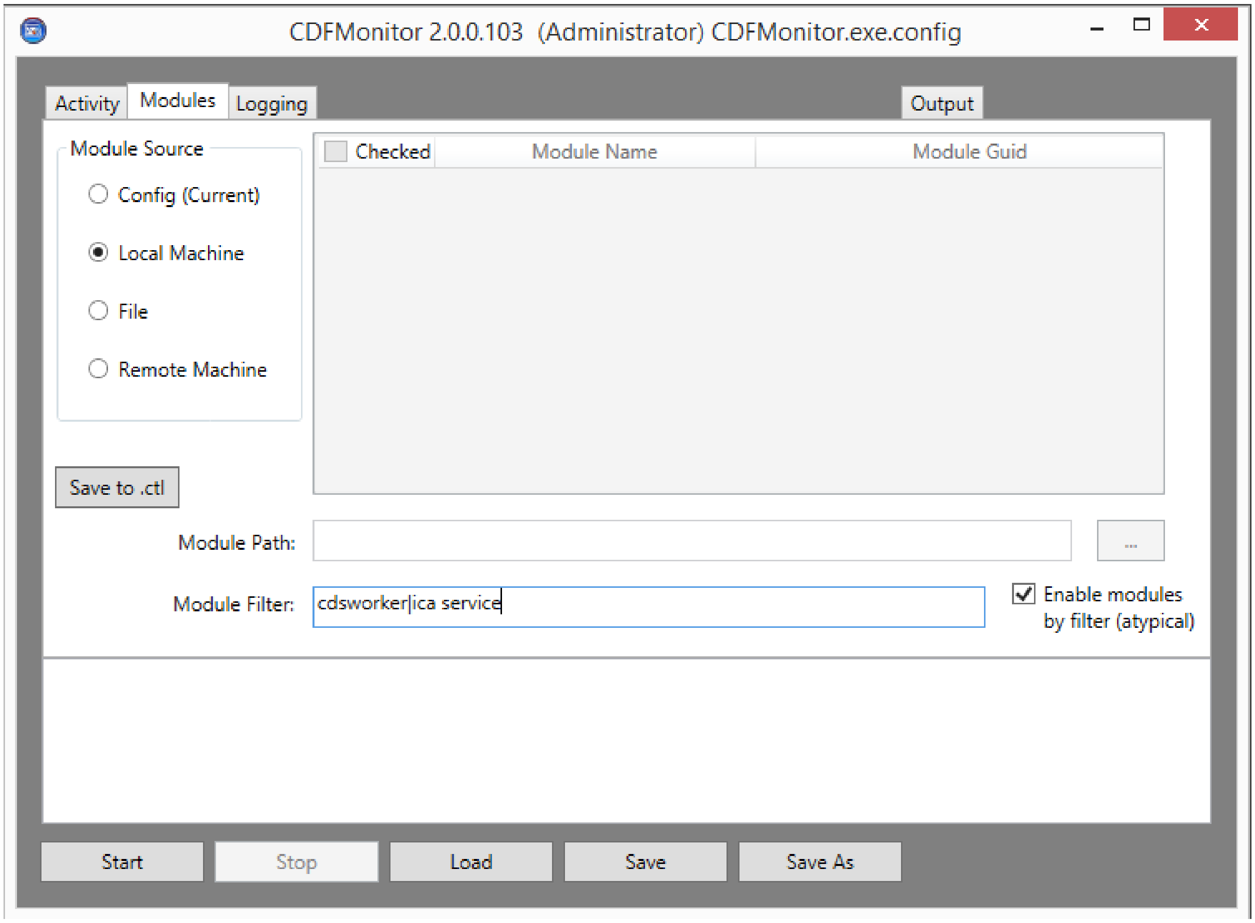
Save as type: Config Files (.ctl)

Enable Modules by Filter

Enable Modules by Filter if selected will default all modules matching the module filter to be enabled. What this provides is the ability of not having to specifically select each individual module. This is useful for using the same config file across multiple versions of a product for example where module names might slightly change. Another example is if module cdsworker is to be enabled, select **Local Machine** as the module source and cdsworker as the **Module Filter**.

For remote configurations, selecting **Local Machine** would be the correct module source in most circumstances when modifying configuration which is not intuitive. The reason being the remote configuration file needs to have **Local Machine** as the value for **Module Source** and not **Remote Machine**.

When enabling this feature, the modules view will be disabled and may or may not be the final module list provided to the remote machine. This list does not matter as keep in mind the list will be coming from the remote machines registry.



CDFMonitor 2.0.0.103 (Administrator) CDFMonitor.exe.config

Module Source

○ Config (Current)

● Local Machine

○ File

○ Remote Machine

□ Checked	Module Name	Module Guid

Module Path:

Module Filter: cdsworker|ica service

☑ Enable modules by filter (atypical)

LOGGING

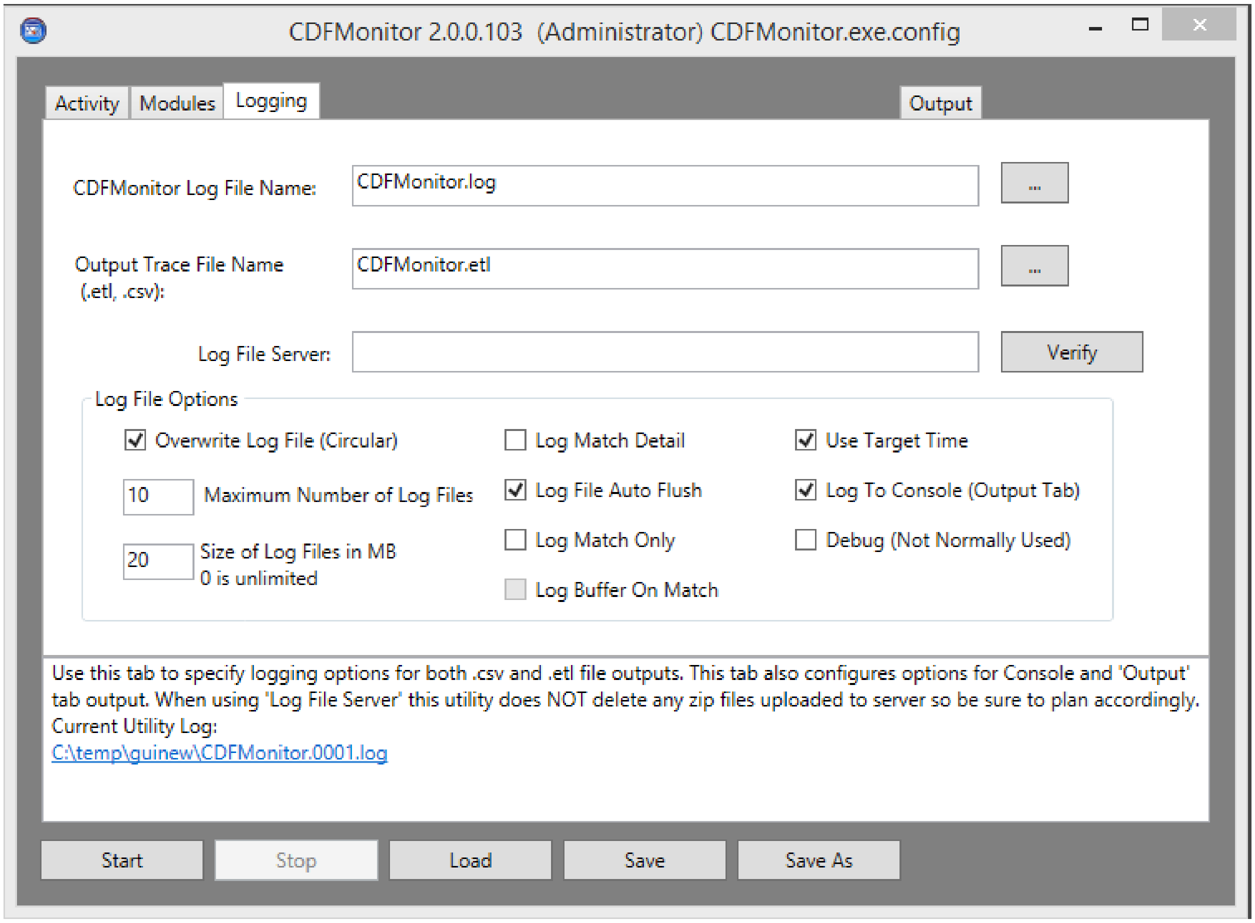
Logging tab contains all of the logging and tracing output options.

In general, the default settings as shown below are sufficient for normal tracing or when using CDFMonitor for remote management or server functions. The default settings are to have a ring buffer of 10 files with file sizes of 20 megabytes for a total of 200 MB. Using the default configuration prevents the drive from filling up while keeping a manageable file size and accounting for reboots or corruption. For most configurations logging is not necessary and values can be cleared to disable.

NOTE: Items to account for when setting configuration for logging is the impact on environment in terms of resources used on the machines being traced. Resources such as CPU, IO, Memory, and Storage should all be considered. If not using console output, unchecking **Log To Console** can decrease resource utilization.

Verify Button for Logging will verify the:

- Path for **Log File Name** is accessible.
- Path for **Trace File Name** is accessible.
- Extension of trace file name corresponds to activity of trace to etl which would have an .etl file extension or trace to csv which would have a .csv file extension. Verify will correct the extension automatically.
- **Log File Server** path if populated.



CDFMonitor 2.0.0.103 (Administrator) CDFMonitor.exe.config

CDFMonitor Log File Name: CDFMonitor.log

Output Trace File Name (.etl, .csv): CDFMonitor.etl

Log File Server:

Log File Options

☑ Overwrite Log File (Circular)

10 Maximum Number of Log Files

20 Size of Log Files in MB
0 is unlimited

☐ Log Match Detail

☑ Log File Auto Flush

☐ Log Match Only

☐ Log Buffer On Match

☑ Use Target Time

☑ Log To Console (Output Tab)

☐ Debug (Not Normally Used)

Use this tab to specify logging options for both .csv and .etl file outputs. This tab also configures options for Console and 'Output' tab output. When using 'Log File Server' this utility does NOT delete any zip files uploaded to server so be sure to plan accordingly.
Current Utility Log:
C:\temp\guinew\CDFMonitor.0001.log

Logging Options

Log File Options controls the output to both the utility and trace file as well as console output.

- **Overwrite Log File** if checked allows the trace/utility file to be overwritten if needed according to log file max count.
- **Maximum Number of Log Files** - (0 - 9999) is the maximum number of log files before the activity stops or if Overwrite is checked before oldest log file gets deleted. Setting to 0 is unlimited but this is not recommended.
- **Size of Log Files** (0 - 1024) is the maximum size in megabytes log file can be before new file is created. Setting to 0 is unlimited but this is not recommended.
- **Log Match Detail** if checked will log additional information if needed.
  - All trace statements will have additional fields; TMF, thread, process id, module, …
  - At trace capture start and end a detailed running process list will also be traced
  - If enabled, output will have this format:
    - [Match],[Date],[Time],[TMFGUId],[ThreadID],[ProcessID],[Module],[Source],[Function],[Level],[Class],[TraceMessage]
  - If not enabled, output will have this format:
    - [Match],[Date],[Time],[Level],[TraceMessage]
  - See TMF (Trace Message Format) files for additional information
- **Log File Auto Flush** if checked will write the trace/utility line immediately. Disabing improves writing directly over wan.
- **Log Match Only** if checked will only write trace information if it is a match from configuration on 'Match' tab. This is useful if looking for only a particular message. Additionally a buffer can be dumped on match.
- **Log Buffer On Match** if checked and Log Match Only is checked will write the last 1000 trace statements in buffer to output trace/utility.
- **Use Target Time** if checked will modify trace time to the local time of the machine that gathered the trace instead of the default of the machine that is parsing the trace.
- **Log To Console** if checked will send all output to console or output tab. If this information is not needed, better performance can be obtained by unchecking.
- **Debug** if checked is used to see additional logging used for debugging the utility itself. It may be useful if utility is not working as expected.

CDFMonitor Log File

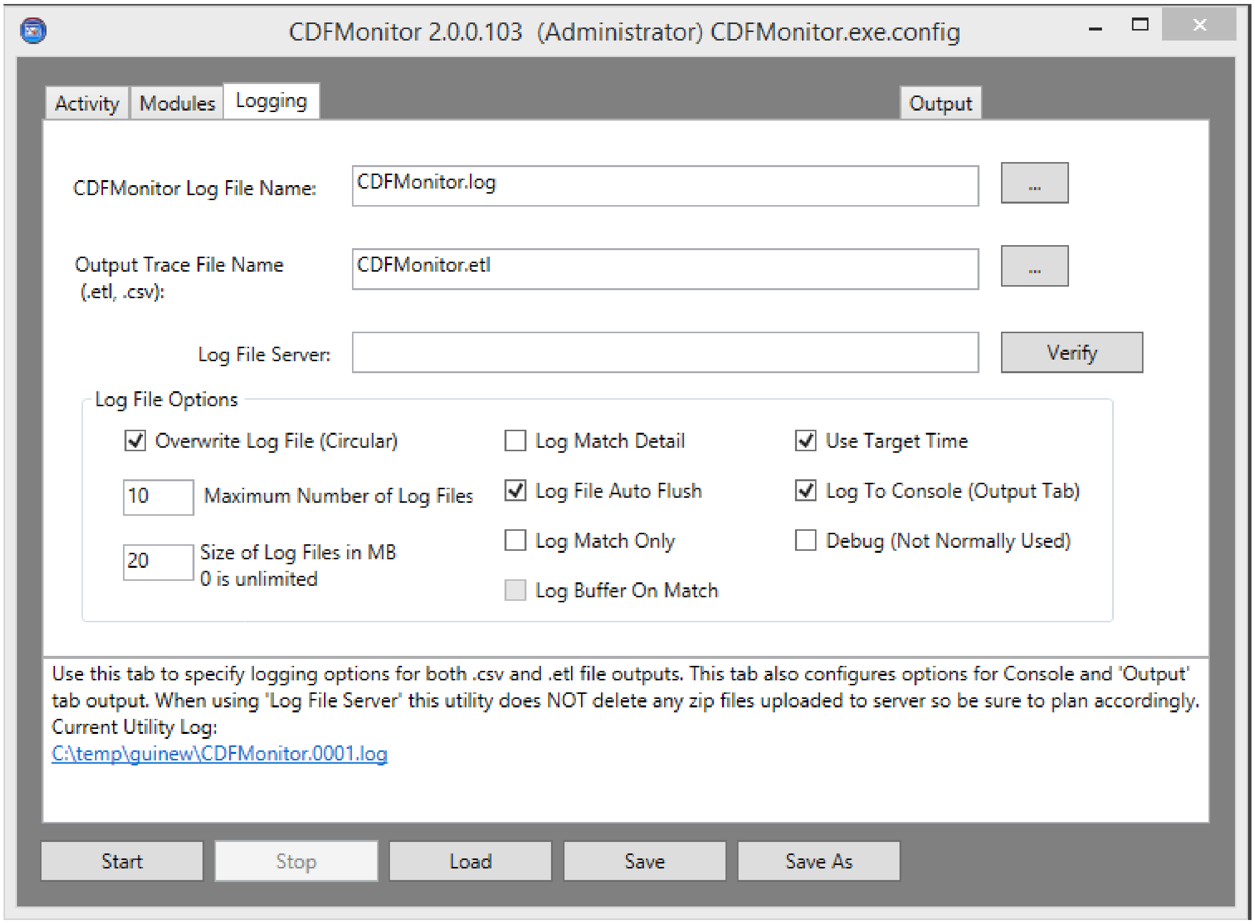
CDFMonitor Log File if populated, will capture all output that is not a CDF trace statement. All trace statements if configured are logged in the separate Trace File. For each activity, all configuration properties will be logged in addition to CDFMonitor log messages.

The log file value can be:

- Empty (to disable logging)
- A file name (which will be created in the working directory)
- A local path and file name
- A remote path and file name unc or mapped drive
  - **NOTE:** Consider authentication when setting up remote paths especially if CDFMonitor is configured to run as a service.

The **CDFMonitor Log File** and **Output Trace File** both share the same log file options. Specifically the settings: **Overwrite Log File**, **Size of Log Files**, and **Maximum Number of Log Files** apply to both files.

For default **Log File Name** as shown below using multiple files, the configured utility log file name will be modified to have a 4 digit index number inserted. For example, cdfmonitor.log actual name will be cdfmonitor.0001.log until conditions are met to create a new file. The next file will have same file name with the index number incremented; cdfmonitor.0002.log.



CDFMonitor Log File Name: CDFMonitor.log

Output Trace File Name (.etl, .csv): CDFMonitor.etl

Log File Server:

Log File Options

☑ Overwrite Log File (Circular)

10 Maximum Number of Log Files

20 Size of Log Files in MB
0 is unlimited

☐ Log Match Detail

☑ Log File Auto Flush

☐ Log Match Only

☐ Log Buffer On Match

☑ Use Target Time

☑ Log To Console (Output Tab)

☐ Debug (Not Normally Used)

Use this tab to specify logging options for both .csv and .etl file outputs. This tab also configures options for Console and 'Output' tab output. When using 'Log File Server' this utility does NOT delete any zip files uploaded to server so be sure to plan accordingly.
Current Utility Log:
C:\temp\guinew\CDFMonitor.0001.log

Output Trace File

Output Trace File Name if populated, will capture all output that is a CDF trace statement. This also includes all traces captured over network when using utility as a server. If activity is **Client: Capture local CDF trace to .csv** the trace file will have a .csv extension and will contain parsed trace statements. If activity is **Client: Capture local CDF trace to .etl** the trace file will have an .etl extension and will not be parsed as it is captured.

The trace file value can be:

- Empty (to disable logging of traces ONLY if activity is **Client: Capture local CDF trace to .csv**)
- A file name (which will be created in the working directory)
- A local path and file name
- A remote path and file name unc or mapped drive ONLY if activity is **Client: Capture local CDF trace to .csv**
    - **NOTE:** Consider authentication when setting up remote paths especially if CDFMonitor is configured to run as a service.

The **CDFMonitor Log File** and **Output Trace File** both share the same log file options. Specifically the settings: **Overwrite Log File**, **Size of Log Files**, and **Maximum Number of Log Files** apply to both files.

For default **Output Trace File Name** as shown in above figure using multiple files, the configured trace file name will be modified to have a 4 digit index number inserted. For example, cdfmonitor.etl actual name will be cdfmonitor.0001.etl until conditions are met to create a new file. The next file will have same file name with the index number incremented; cdfmonitor.0002.etl. The same is true when parsing real-time to .csv file.

Depending on configuration **Log Match Detail**, the trace log will also contain detailed trace message information and a currently running process list.

Log File Server

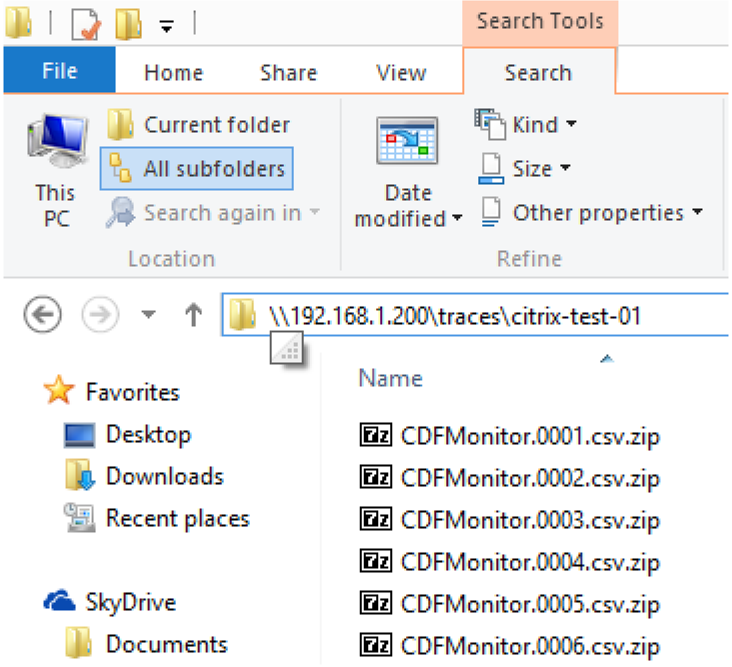
Log File Server if populated will be used to permanently store log and trace files as they are finalized. These files will be zipped, time stamped, and uploaded to a subdirectory named with the machines' host name. When activity is started, every 60 seconds all trace and log files that are configured will be checked. Any files that are not currently being written to will be zipped for upload. After upload, the local zip will be deleted. The local files that are uploaded will be locally as normal according to Logging configuration.

NOTE: The uploaded zip files will NOT be deleted by CDFMonitor on the remote share.

Log File Server value can be:

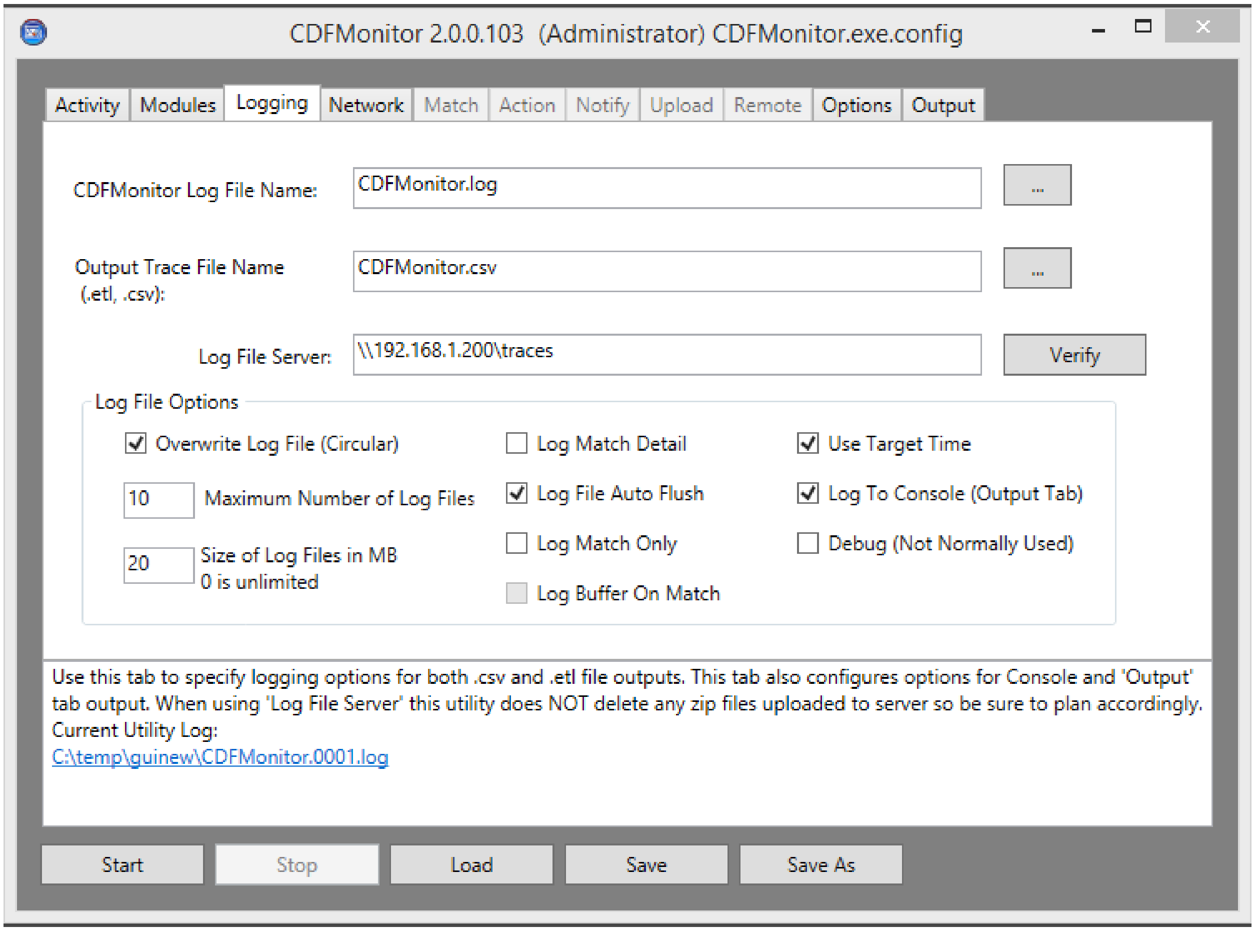
- Empty (to disable logging to a log file server)
- A remote unc path and file name.
    - **NOTE:** Consider authentication when setting up remote paths especially if CDFMonitor is configured to run as a service.

In the following figures, **\\192.168.1.200\traces** is configured for **Log File Server**. The machines' name running this instance of CDFMonitor is citrix-test-01.



\\192.168.1.200\traces\citrix-test-01

CDFMonitor.0001.csv.zip
CDFMonitor.0002.csv.zip
CDFMonitor.0003.csv.zip
CDFMonitor.0004.csv.zip
CDFMonitor.0005.csv.zip
CDFMonitor.0006.csv.zip



CDFMonitor Log File Name: CDFMonitor.log

Output Trace File Name (.etl, .csv): CDFMonitor.csv

Log File Server: \\192.168.1.200\traces

Log File Options

☑ Overwrite Log File (Circular)

10 Maximum Number of Log Files

20 Size of Log Files in MB
0 is unlimited

☐ Log Match Detail

☑ Log File Auto Flush

☐ Log Match Only

☐ Log Buffer On Match

☑ Use Target Time

☑ Log To Console (Output Tab)

☐ Debug (Not Normally Used)

Use this tab to specify logging options for both .csv and .etl file outputs. This tab also configures options for Console and 'Output' tab output. When using 'Log File Server' this utility does NOT delete any zip files uploaded to server so be sure to plan accordingly.
Current Utility Log:
C:\temp\guinew\CDFMonitor.0001.log

NETWORK

Network tab is and advanced option that can be used by both Client and Server activities. For real-time client activities such as trace to csv, enabling Network options can send messages to another CDFMonitor instance running as a server. If Firewalls are enabled, they would need exception rules added for the configured UDP ports.

For server activity, any **Udp Ping** information being sent to this instance will be displayed in the **Udp clients connecting to this server** view pane shown in figure below on the right. **Udp Client** trace messages will be shown on the server instance Output tab.

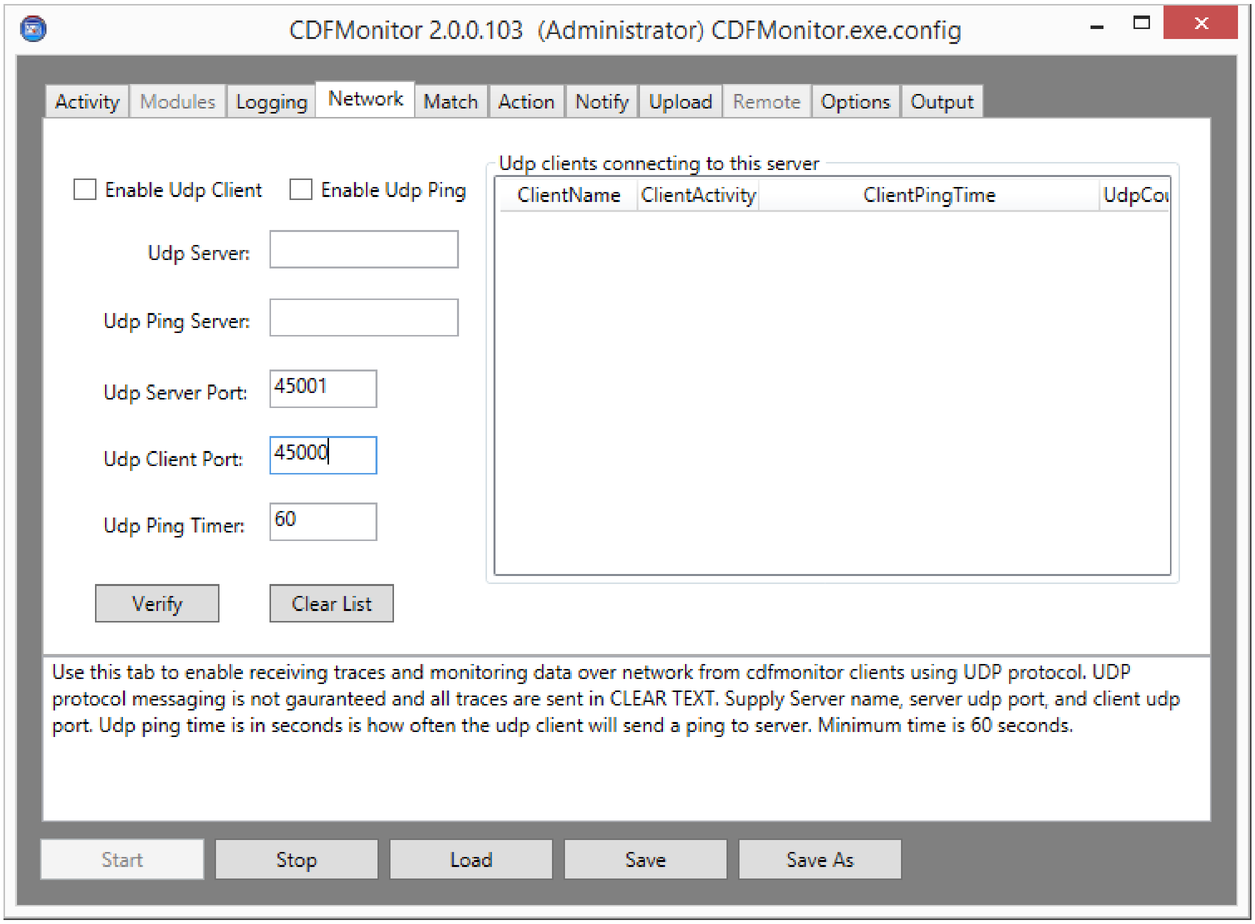
Each trace message as seen in Output tab would be sent as an individual UDP packet to value specified for **Udp Server**. **Udp Client** trace messages can be sent to any UDP listener like Splunk for example as data is sent parsed and in csv string format. **Udp Ping** can only be sent to CDFMonitor server instances.

NOTE: All information sent over network will be in clear text. This would be the same as viewing the trace locally in the Output tab.

NOTE: Consider the amount of information being sent over network and test before deploying. Some Citrix products if configured to trace every module can generate over 20,000 traces per second under load. Selecting only needed modules and\or using filtering on Match tab can reduce the amount of traffic and overall utilization.

Verify button will:

- Check ports are in valid range and are able to be bound.
- Check timer range is at least 60 seconds.
- Ping **Udp Server** and **Udp Ping Server** with a UDP Ping packet that will be displayed on the server instance if server activity has been started.



□ Enable Udp Client □ Enable Udp Ping

Udp Server:

Udp Ping Server:

Udp Server Port: 45001

Udp Client Port: 45000

Udp Ping Timer: 60

Udp clients connecting to this server

ClientName	ClientActivity	ClientPingTime	UdpCo

Use this tab to enable receiving traces and monitoring data over network from cdfmonitor clients using UDP protocol. UDP protocol messaging is not gauranteed and all traces are sent in CLEAR TEXT. Supply Server name, server udp port, and client udp port. Udp ping time is in seconds is how often the udp client will send a ping to server. Minimum time is 60 seconds.

Network Options

Network Options are used for both Client and Server tracing activities.

- **Enable Udp Client** if checked will enable sending trace messages to the configured 'Udp Server' from the client running the trace.
- **Enable Udp Ping** if checked will enable sending 'ping' packets to the 'Udp Ping Server' or 'Udp Server' if **Udp Ping Server** is not populated from the client running the trace.
- **Udp Server** is the IP address or FQDN of the server to receive UDP trace messages.
- **Udp Ping server** is the IP address or FQDN of the server to receive the UDP ping packet.
- **Udp Server Port** the UDP network port the 'server' will listen on and 'client' should send to.
- **Udp Client Port** the UDP network port the 'client' will send from.
- **Udp Ping timer** time in seconds above 60 if **Udp Ping** is enabled client will send 'ping' packet with trace session metadata info (as partially shown below in the listview) to the **Udp Ping Server** or **Udp Server**.

Monitoring CDFMonitor network clients

For Server configuration when running **Server: Capture remote trace messages** activity, this tab can be used to see all of the clients ping packet information being sent to this configured instance. Server instances can be 'clients' of other servers.

NOTE: Pings are sent clear text.

Udp Server configuration for server capture of remote pings or traces:

- If empty will listen on all adapters on port configured for **Udp Server Port** as shown below.
- If a local IP address will listen on that adapter on port configured for **Udp Server Port**.
- If a remote IP address/FQDN, CDFMonitor will be a client and send/forward messages to the specified IP address or FQDN using specified ports if enabled. This includes captured traces. Traces will be prepended with local 'server' hostname.

This ping information shown in view below is not saved and therefore lost when utility is closed however you can select all and copy information out of view. It will then paste as csv.

The following information is provided from the client in a ping packet:

- **Client name** - machine name of CDFMonitor client.
- **Client Activity** - activity CDFMonitor is performing. TraceToCsv, TraceToEtl
- **Client Ping Time** - time of last ping from CDFMonitor client.
- **Udp Counter -** number of packets that have been sent from CDFMonitor client since tracing started.
- **Traces Per Second** – number of traces that are being processed per second on client.
- **Matched Events** – number of events that have been matched from client since trace started.
- **Missed Matched Events -** number of events that have been matched but no action taken due to throttle.
- **Processed Events** - number of events that have been processed from client since trace started.
- **Average Process Cpu -** average process utilization for CDFMonitor process.
- **Current Process Cpu** - current process utilization for CDFMonitor process.
- **Current Machine Cpu** - current process utilization for all processes.
- **Duration** - time since client trace started.

Monitoring CDFMonitor client trace messages

CDFMonitor Server activity instances can listen for both ping and trace messages. All captured messages will be processed, and if configured, will take an action on any matches. All messages will be sent to output console and logged to trace file if configured.

Traces captured over network will be prepended with the client CDFMonitor machine name and UDP counter from client. See figure below.

Traces are sent as csv string one packet per trace message. Incoming traces are parsed as a local trace would be for matches and actions to be performed.

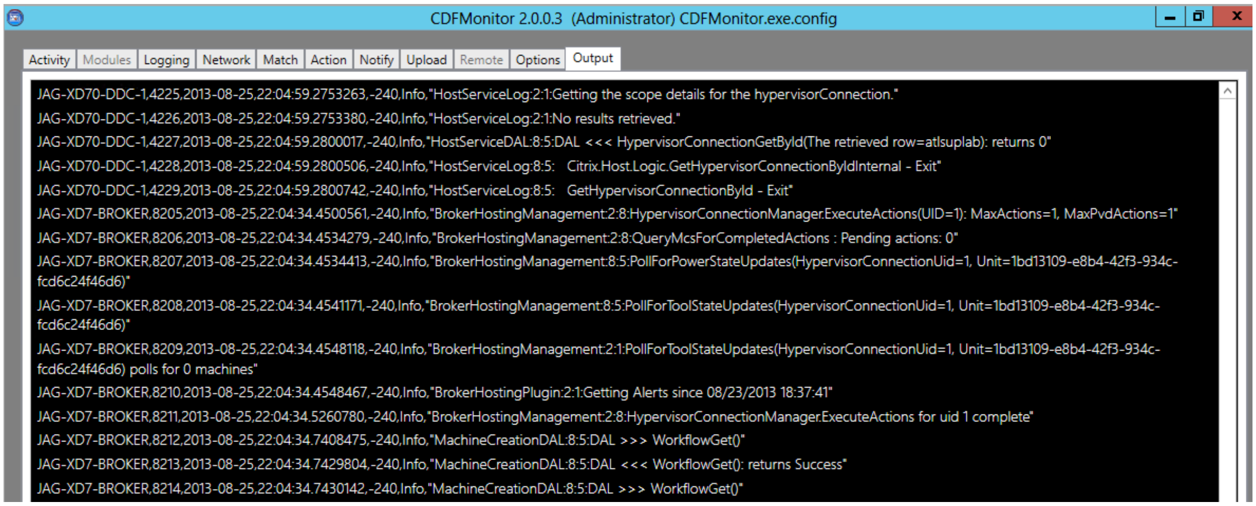
NOTE: Traces are sent clear text.

NOTE: Always test performance of configuration before deploying to environment. Configuration for example module selection, resources on the machine, and current activities being performed on that machine can all affect performance.

Udp Server configuration for server capture remote ping/trace:

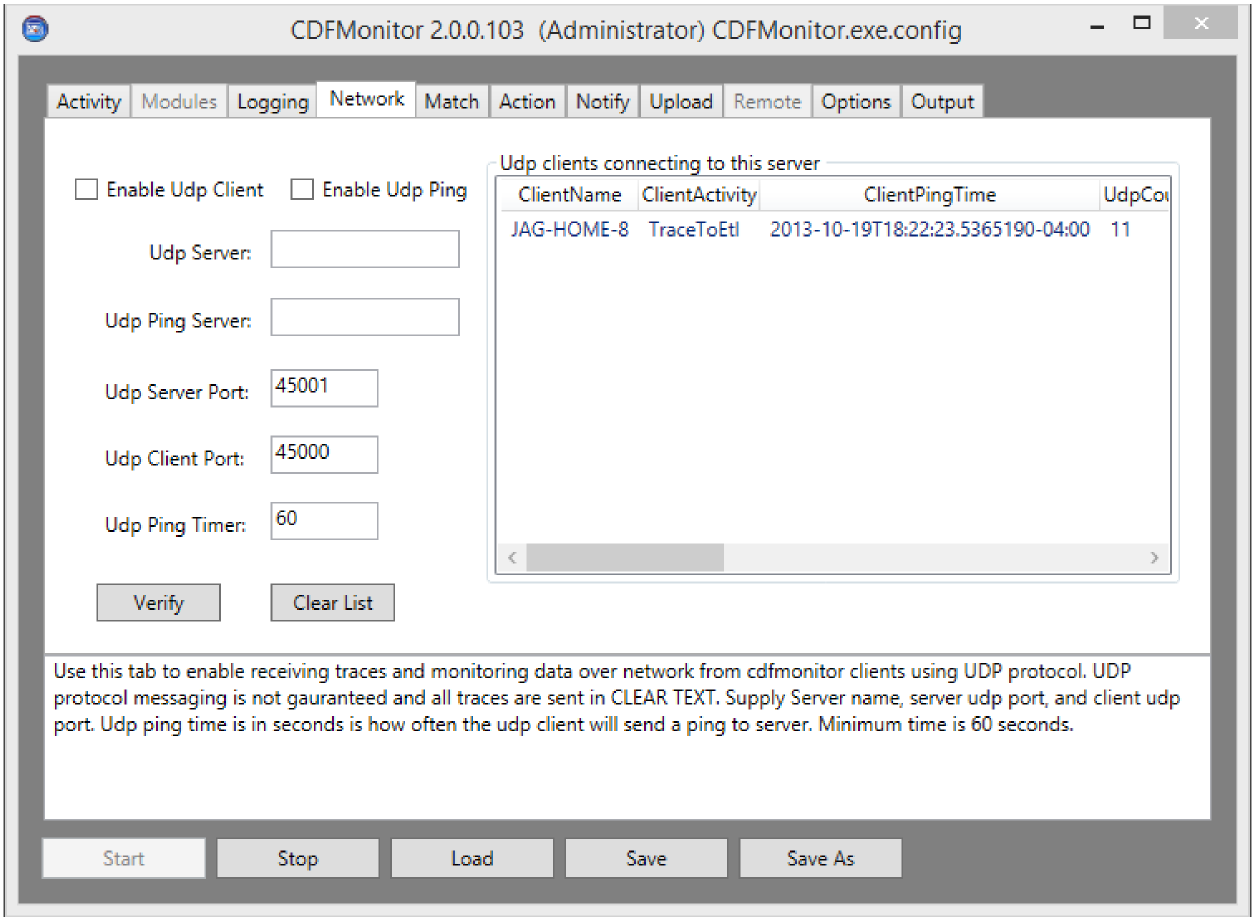
- If empty will listen on all adapters on port configured for Udp Server Port.
- If a local IP address will listen on that adapter on port configured for Udp Server Port as shown below.
- If a remote IP address/FQDN, the utility will act as a client and send/forward messages to IP Address or FQDN using specified ports if enabled. This includes captured traces. Traces will be prepended with local 'server' hostname.

Below shows an example of a server trace capturing from two CDFMonitor clients 'jag-xd70-ddc-1' and 'jag-xd7-broker':



```
JAG-XD70-DDC-1,4225,2013-08-25,22:04:59.2753263,-240,Info,"HostServiceLog:2:1:Getting the scope details for the hypervisorConnection."
JAG-XD70-DDC-1,4226,2013-08-25,22:04:59.2753380,-240,Info,"HostServiceLog:2:1:No results retrieved."
JAG-XD70-DDC-1,4227,2013-08-25,22:04:59.2800017,-240,Info,"HostServiceDAL:8:5:DAL <<< HypervisorConnectionGetById(The retrieved row=atlsuplab): returns 0"
JAG-XD70-DDC-1,4228,2013-08-25,22:04:59.2800506,-240,Info,"HostServiceLog:8:5:   Citrix.Host.Logic.GetHypervisorConnectionByIdInternal - Exit"
JAG-XD70-DDC-1,4229,2013-08-25,22:04:59.2800742,-240,Info,"HostServiceLog:8:5:   GetHypervisorConnectionById - Exit"
JAG-XD7-BROKER,8205,2013-08-25,22:04:34.4500561,-240,Info,"BrokerHostingManagement:2:8:HypervisorConnectionManager.ExecuteActions(UID=1): MaxActions=1, MaxPvdActions=1"
JAG-XD7-BROKER,8206,2013-08-25,22:04:34.4534279,-240,Info,"BrokerHostingManagement:2:8:QueryMcsForCompletedActions : Pending actions: 0"
JAG-XD7-BROKER,8207,2013-08-25,22:04:34.4534413,-240,Info,"BrokerHostingManagement:8:5:PollForPowerStateUpdates(HypervisorConnectionUid=1, Unit=1bd13109-e8b4-42f3-934c-
fcd6c24f46d6)"
JAG-XD7-BROKER,8208,2013-08-25,22:04:34.4541171,-240,Info,"BrokerHostingManagement:8:5:PollForToolStateUpdates(HypervisorConnectionUid=1, Unit=1bd13109-e8b4-42f3-934c-
fcd6c24f46d6)"
JAG-XD7-BROKER,8209,2013-08-25,22:04:34.4548118,-240,Info,"BrokerHostingManagement:2:1:PollForToolStateUpdates(HypervisorConnectionUid=1, Unit=1bd13109-e8b4-42f3-934c-
fcd6c24f46d6) polls for 0 machines"
JAG-XD7-BROKER,8210,2013-08-25,22:04:34.4548467,-240,Info,"BrokerHostingPlugin:2:1:Getting Alerts since 08/23/2013 18:37:41"
JAG-XD7-BROKER,8211,2013-08-25,22:04:34.5260780,-240,Info,"BrokerHostingManagement:2:8:HypervisorConnectionManager.ExecuteActions for uid 1 complete"
JAG-XD7-BROKER,8212,2013-08-25,22:04:34.7408475,-240,Info,"MachineCreationDAL:8:5:DAL >>> WorkflowGet()"
JAG-XD7-BROKER,8213,2013-08-25,22:04:34.7429804,-240,Info,"MachineCreationDAL:8:5:DAL <<< WorkflowGet(): returns Success"
JAG-XD7-BROKER,8214,2013-08-25,22:04:34.7430142,-240,Info,"MachineCreationDAL:8:5:DAL >>> WorkflowGet()"
```

Example Network Server configuration:



☐ Enable Udp Client ☐ Enable Udp Ping

Udp Server:

Udp Ping Server:

Udp Server Port: 45001

Udp Client Port: 45000

Udp Ping Timer: 60

Udp clients connecting to this server

ClientName	ClientActivity	ClientPingTime	UdpCou
JAG-HOME-8	TraceToEtl	2013-10-19T18:22:23.5365190-04:00	11

Use this tab to enable receiving traces and monitoring data over network from cdfmonitor clients using UDP protocol. UDP protocol messaging is not gauranteed and all traces are sent in CLEAR TEXT. Supply Server name, server udp port, and client udp port. Udp ping time is in seconds is how often the udp client will send a ping to server. Minimum time is 60 seconds.

CDFMonitor client tracing to network

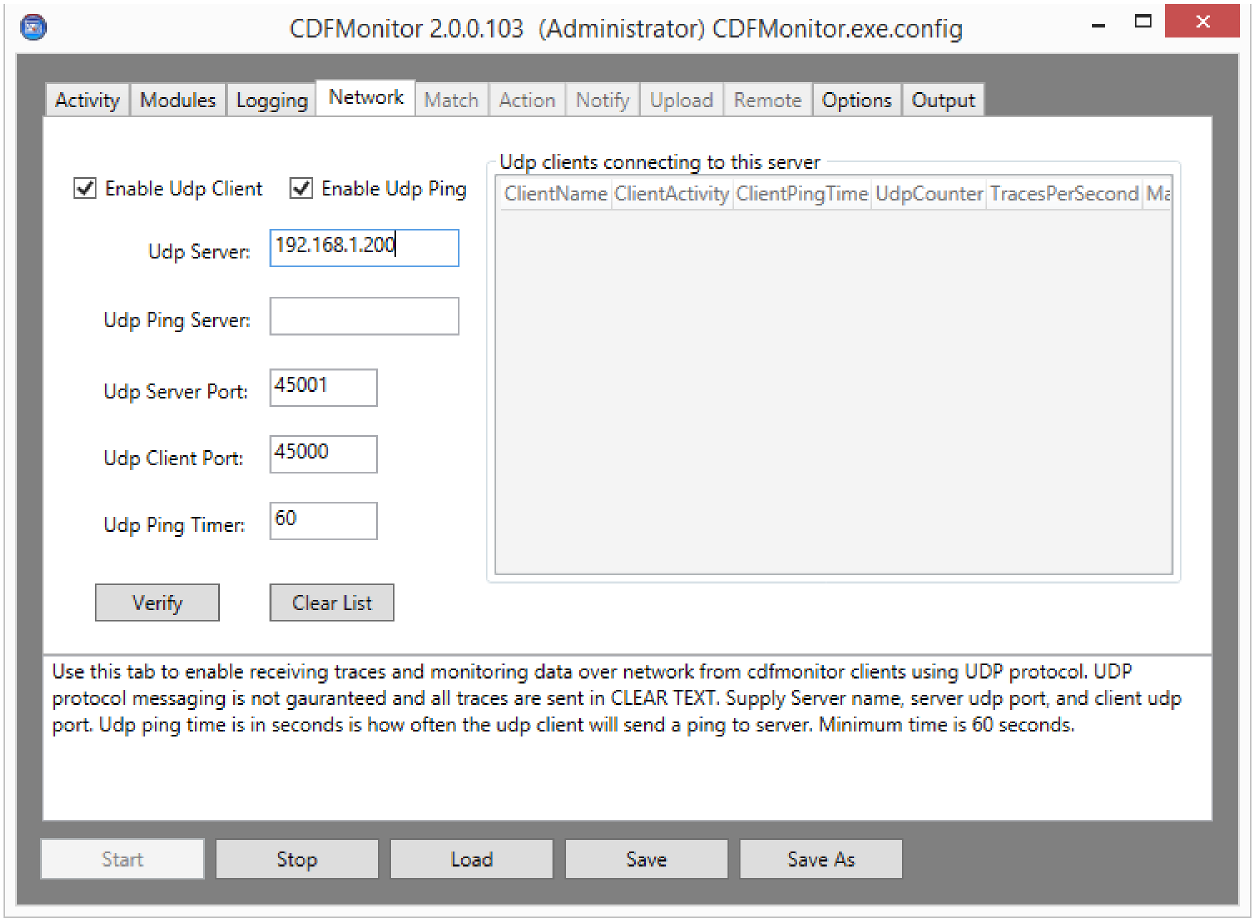
For client configuration, optionally sending trace messages and ping messages over network using UDP protocol can be enabled. There are two types of packets. The first type is a trace packet to an **Udp Server**. Additionally, at a specified interval, a 'ping' packet can be sent to an **Udp Ping Server** that contains metadata information about the client being traced. Both **Udp Server** and **Udp Ping Server** can be other instances of CDFMonitor running the **Server: Capture remote trace messages** activity. For **Udp Server**, it can also be any other UDP listener type server for example Splunk. If Firewalls are enabled, rules will have to be added for the UDP ports configured.

Sending messages over network can help overcome obstacles of having to store trace information locally on the client machine being traced. Examples of this are doing long term monitoring of the environment for health or for a specific intermittent issue.

NOTE: All messages are sent clear text.

NOTE: Filtering traces may be preferred to keep from sending every message over network. Using Match tab to filter unwanted messages and logging options log match only can reduce amount of traffic. See Match and Logging sections for more information.

Example Network Client configuration:



CDFMonitor 2.0.0.103 (Administrator) CDFMonitor.exe.config

☑ Enable Udp Client ☑ Enable Udp Ping

Udp Server: 192.168.1.200

Udp Ping Server:

Udp Server Port: 45001

Udp Client Port: 45000

Udp Ping Timer: 60

Use this tab to enable receiving traces and monitoring data over network from cdfmonitor clients using UDP protocol. UDP protocol messaging is not gauranteed and all traces are sent in CLEAR TEXT. Supply Server name, server udp port, and client udp port. Udp ping time is in seconds is how often the udp client will send a ping to server. Minimum time is 60 seconds.

MATCH

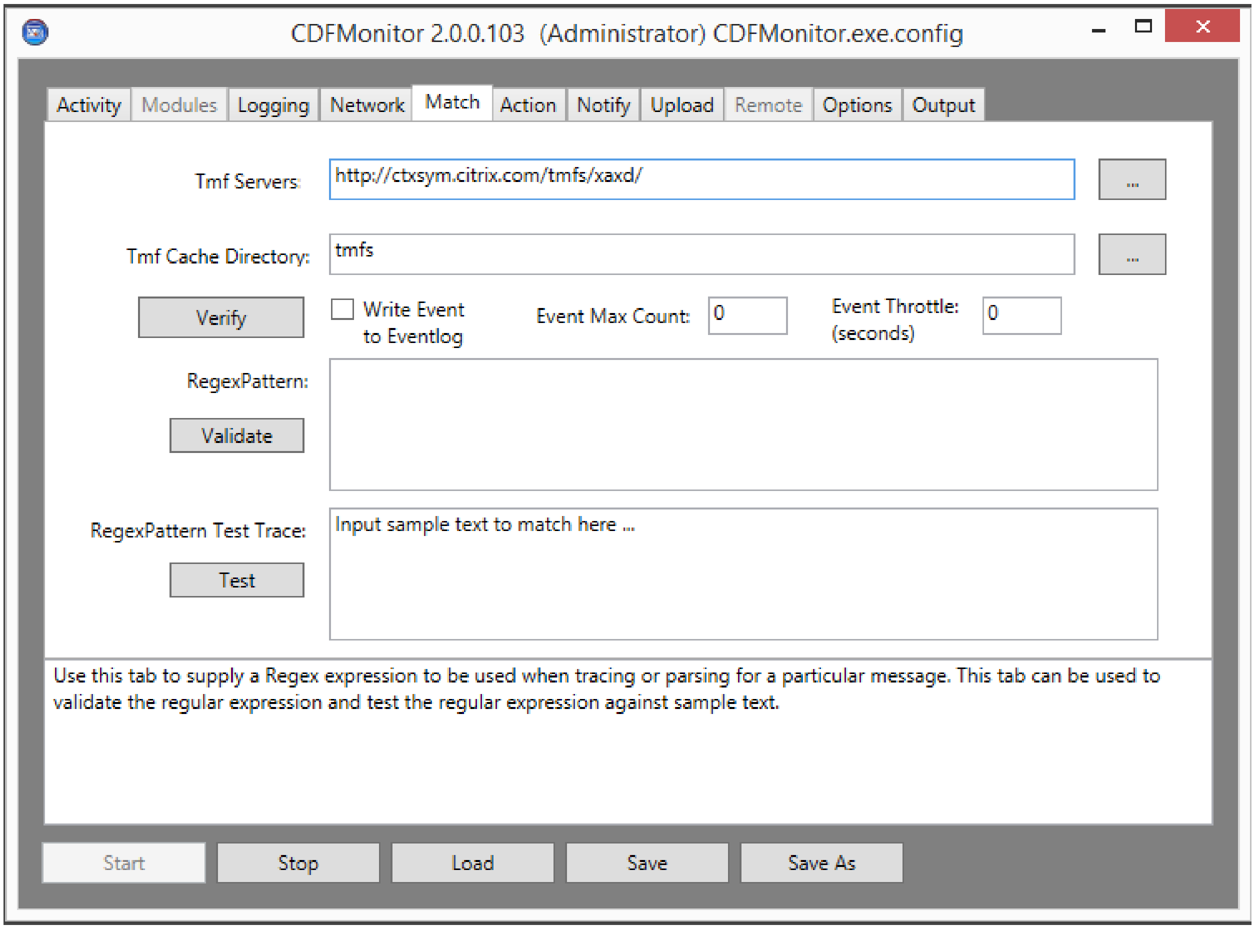
Matching is used for troubleshooting intermittent issues or for monitoring environments in general. Configuring Matching Tab is an advanced option that can be used for certain activities to take an action or notification when a matched event occurs. Client tracing to csv, Server capturing remote traces, or formatting a trace are the activities that can use this capability.

NOTE: For best performance when tracing trace to .etl file should be used. This is because there is no real-time parsing of the raw trace message from ETW. The consequence of this the inability to perform any matching. If matching or real-time parsing is not desired then tracing to .etl file would be best.

Verify button for Matching will verify:

- TMF file server path if populated
- TMF cache directory path if populated
- Send test event to Event Log if Write Event to Event Log is checked

See Show Stats / Activity Summary for additional information.



Tmf Servers: http://ctxsym.citrix.com/tmfs/xaxd/

Tmf Cache Directory: tmfs

☐ Write Event to Eventlog

Event Max Count: 0

Event Throttle: (seconds) 0

RegexPattern:

RegexPattern Test Trace: Input sample text to match here ...

Use this tab to supply a Regex expression to be used when tracing or parsing for a particular message. This tab can be used to validate the regular expression and test the regular expression against sample text.

Matching Options

The following options are available for Matching on the Matching Tab that control some notification, how many matches, and how often matches will be acted upon (throttling). Other options on the Action, Notify, and Upload tabs can be used in conjunction with option on the Match tab but are not necessary.

NOTE: If using **Notify** or **Upload**: to prevent a misconfigured match pattern notifications from continually being sent, especially when testing, use **Event Max Count** and **Event throttle** options.

To configure utility to perform an action on a match for example, run an external process, see Action section.

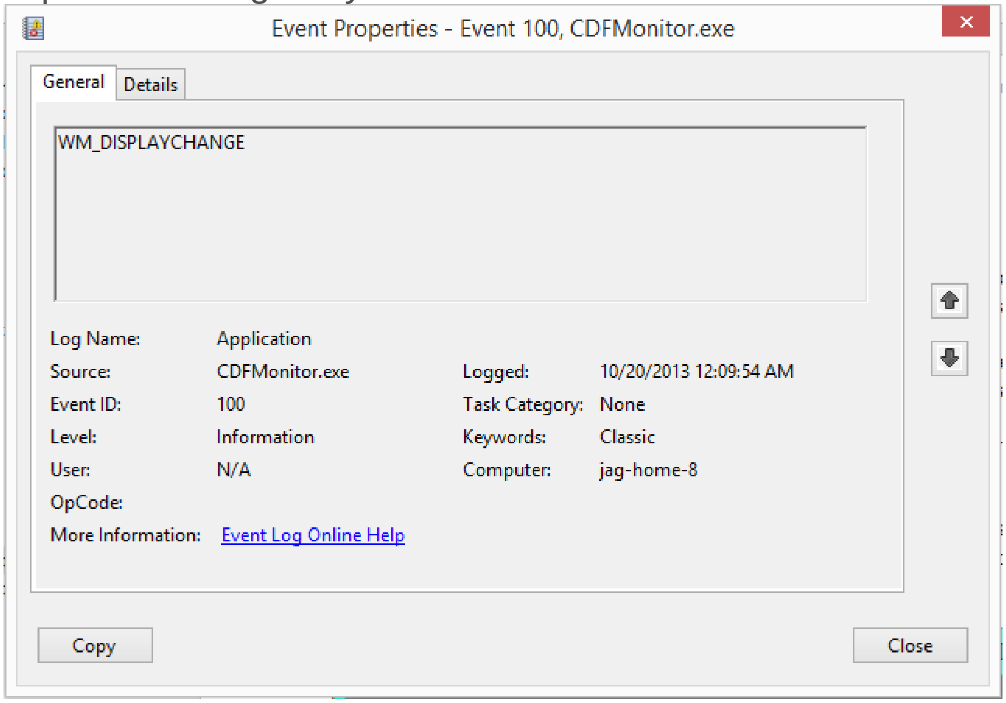
To configure utility for notification on a match multiple options are available. For an email notification see Notify section. Additionally writing an event to the Event Log when a match occurs can be used for notification as well as most environments have Event Log monitoring capability.

To configure utility to upload logging and other information see Upload section. This will give the ability to specify an ftp upload URL that zipped files should be uploaded to when a match occurs.

Matching options definitions:

- **Write Event to Event Log** if selected will write an event (Event ID 100) to the Application Event Log every time a match occurs unless throttled.
  - As mentioned above, writing to Event Log can allow other environment mechanisms already in place to take action or monitor for this event.
- **Event Max Count** if populated will stop tracing once the specified number of traces have been matched
- **Event Throttle** if populated will prevent any subsequent match from starting an action, notification, or upload until configured seconds elapse. This throttled event match will be logged as 'bypassing event due to throttle'. At the end of a trace in trace summary information there is also a counter for 'throttled events'. The Matched Events counter will still be incremented even if throttled.

Example Event Log entry from CDFMonitor:



WM_DISPLAYCHANGE

Log Name: Application
Source: CDFMonitor.exe
Logged: 10/20/2013 12:09:54 AM
Event ID: 100
Task Category: None
Level: Information
Keywords: Classic
User: N/A
Computer: jag-home-8
OpCode:
More Information: Event Log Online Help

TMF (Trace Message Format) files

Trace Message Format files (TMFs) are readable text files that are required for parsing unmanaged (C/C++) traces.

The format raw trace bytes from a program into a readable format. Citrix .net traces use a generic TMF file for parsing but this file is built into both CDFControl and CDFMonitor so for all intents a TMF file is not required for .net real-time tracing or parsing. Depending on the product and component, tracing maybe all .net, all unmanaged, or both. Some examples are XD7 broker is all .net, but XD7 VDA has both.

If not tracing real-time trace to csv or parsing unmanaged code then **TMF Servers** configuration is not required.

- **TMF Servers** if populated will be used when parsing or real-time tracing to csv unmanaged code. Input can have multiple paths separated by semicolon. Paths can be drive, unc, or URL.
    - Default Citrix TMF server URL: **http://ctxsym.citrix.com/TMFs/xaxd/**
- **TMF Cache Directory** if populated will store all TMFs downloaded from a TMF server locally for future trace sessions. Only one path can be specified.

Trace Message Output

See section **Log File Options** for **Log Match Detail** information

- If **Log Match Detail** is enabled, output will have this format:
    - [Match],[Date],[Time],[TMFGUId],[ThreadID],[ProcessID],[Module],[Source],[Function],[Level],[Class],[TraceMessage]
- If **Log Match Detail** is not enabled, output will have this format:
    - [Match],[Date],[Time],[Level],[TraceMessage]
- Example managed code (.net) trace from Citrix CDF with **Log Match Detail** checked:
    - NOMATCH,2013-04-04,08:49:36.6765410,-240,1c7af7a9-d188-4604-a430-9c378087ca5d,2628,2576,,unknown1,,Info,,"ICA Service:8:9:Citrix.Portica.Utils.CtxThreadPool.CalculateMovingAverage"
    - 1c7af7a9-d188-4604-a430-9c378087ca5d = generic .net TMF used for ALL Citrix CDF .net tracing
    - This module is built into utility and therefore utility does not need access to TMF server for .net tracing
    - Notice on .net trace, no module, source, or class information will ever be traced.
- Example unmanaged code trace from Citrix CDF with **Log Match Detail** checked:
    - cdfmonitor.0003.etl.csv	223877	MATCH,2013-05-07,15:18:42.5596100,-300,d8359128-5162-47f1-d4f0-12193deb7e39,1840,4400,,unknown2551,,"Hook_SHQueryValueExW: entry"

RegexPattern

For an event match to occur for action, documentation, notification or upload, a regular expression has to be configured for **RegexPattern**. A regular expression can be as simple as literal strings or complex with named grouping. For CDFMonitor any valid regular expression is supported.

Validate button will verify the given regular expression is a valid regular expression

- Output is similar to: The supplied regular expression is valid. Use sample text to verify pattern will capture the desired match.

Regular expression named grouping can be used to associate one or many regular expressions to one or more actions.

NOTE: Pattern complexity along with selected Modules configuration can affect performance. Keeping the selected module list and **RegexPattern** as simple as possible and testing before deployment should always be performed.

Regular expressions Reference:

- **www.myregextester.com** - good site for building and testing regular expressions.
- **http://www.regular-expressions.info/** - good site for documentation

RegexPattern Testing

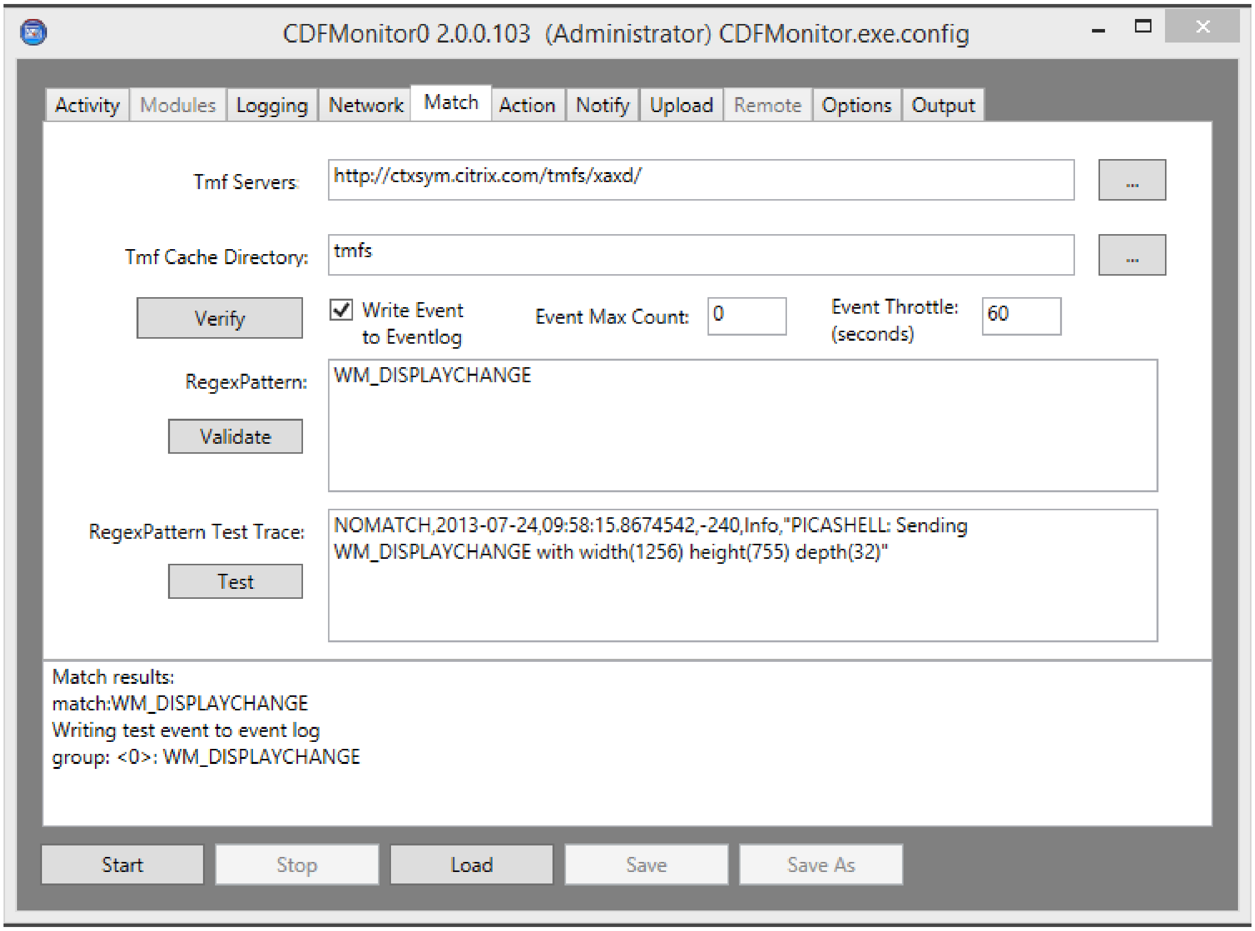
RegexPattern Test trace can be populated with an example trace to be matched to help build out and verify RegexPattern for both trace matching and for their configured actions on Action tab.

To use:

- Paste in an example trace that the RegexPattern should match.
- Build RegexPattern and **Validate.**
- Select **Test** button and view results in Status window.
    - Example results:
        - Match results:
        - match:WM_DISPLAYCHANGE
        - Writing test event to event log
        - group: <0>: WM_DISPLAYCHANGE

See Action Execute on match for additional information.

Example simple text match with results from **Test** button:



CDFMonitor0 2.0.0.103 (Administrator) CDFMonitor.exe.config

Tmf Servers: http://ctxsym.citrix.com/tmfs/xaxd/

Tmf Cache Directory: tmfs

☑ Write Event to Eventlog

Event Max Count: 0

Event Throttle: (seconds) 60

RegexPattern: WM_DISPLAYCHANGE

RegexPattern Test Trace: NOMATCH,2013-07-24,09:58:15.8674542,-240,Info,"PICASHELL: Sending WM_DISPLAYCHANGE with width(1256) height(755) depth(32)"

```
Match results:
match:WM_DISPLAYCHANGE
Writing test event to event log
group: <0>: WM_DISPLAYCHANGE
```

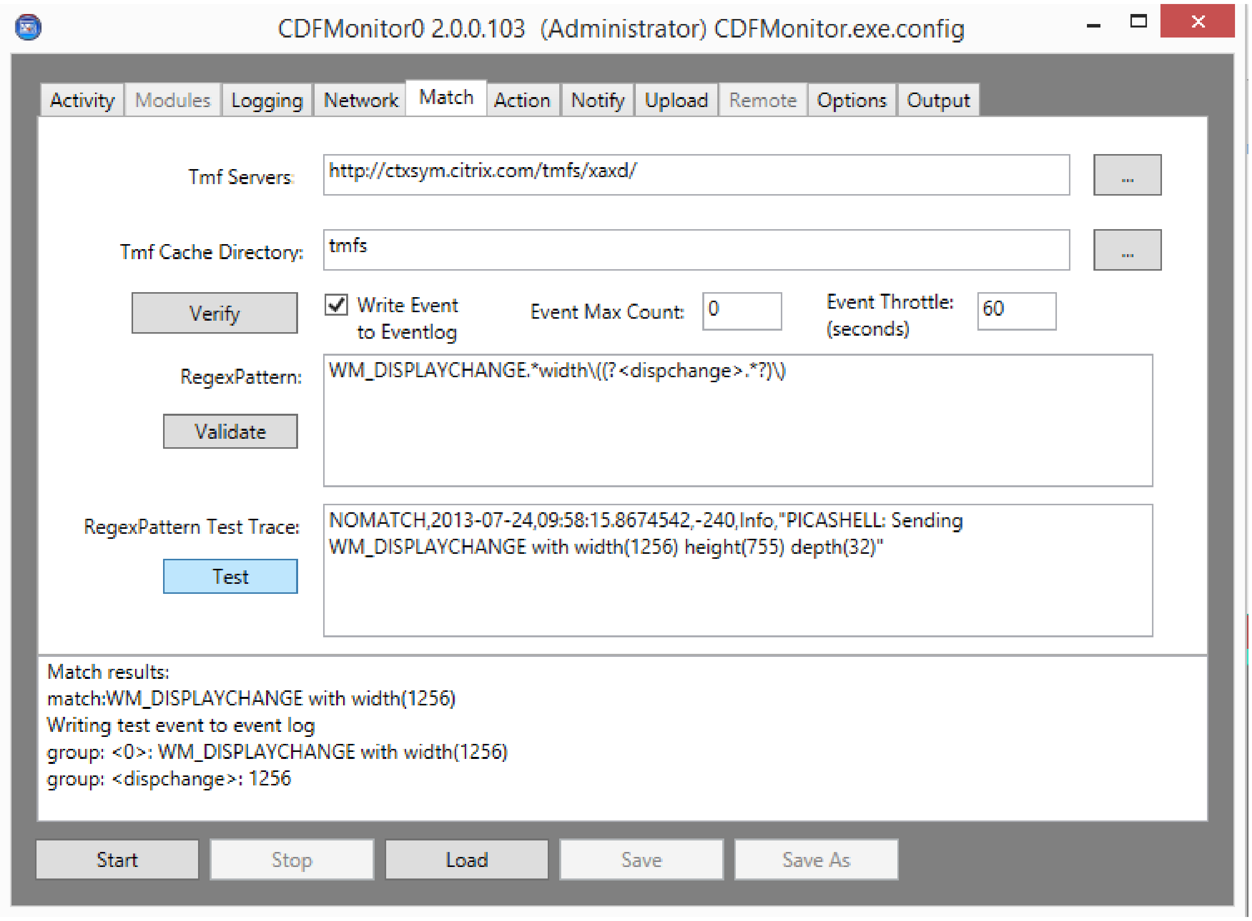
Regular Expression Named Grouping

Using normal Regular expression named grouping syntax, each regex match can be associated to a specific action by the group name. The named group placeholder can be used to specify both the command(s) to run and for passing variables to those commands from the trace statement itself.

See RegexPattern Testing for additional information on 'Test' results as shown below

See Action for complete configuration using same example as below

Example multiple named group match configuration.



Tmf Servers: http://ctxsym.citrix.com/tmfs/xaxd/

Tmf Cache Directory: tmfs

☑ Write Event to Eventlog   Event Max Count: 0   Event Throttle: (seconds) 60

RegexPattern: WM_DISPLAYCHANGE.*width\((?<dispchange>.*?)\)

RegexPattern Test Trace: NOMATCH,2013-07-24,09:58:15.8674542,-240,Info,"PICASHELL: Sending WM_DISPLAYCHANGE with width(1256) height(755) depth(32)"

```
Match results:
match:WM_DISPLAYCHANGE with width(1256)
Writing test event to event log
group: <0>: WM_DISPLAYCHANGE with width(1256)
group: <dispchange>: 1256
```

ACTION

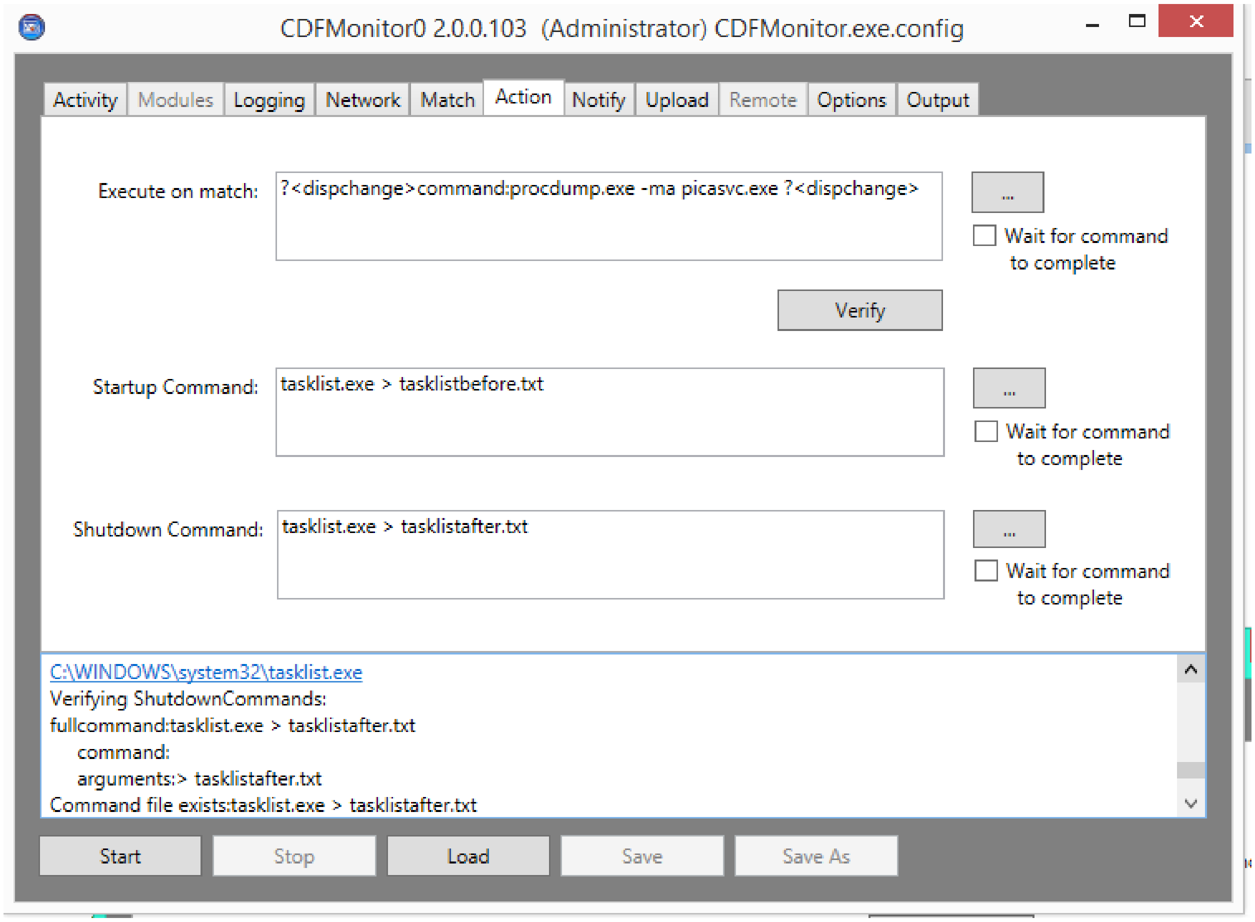
Action is an advanced option that if configured allows for commands to be executed either when a match occurs, on startup, or on shutdown. This can be in any combination.

Wait for command to complete will wait for command to complete to log results. Waiting for the results will not stop the currently running activity.

NOTE: Shutdown command should not be used unless necessary as this can have undesired results if machine is shutting down. It should be a short lived process.

Verify button for Action will simulate a matched event but will not actually run the command. If **RegexPattern** and **RegexPattern Test Trace** are populated with named groups, Verify will simulate argument replacements as well as shown in the below screenshot. Verify will try to split the command from the arguments and find a file matching command specified. Verify will search current directory and all paths in %PATH% environment variable.

Example below shows actions after selecting **Verify**.



Execute on match: ?<dispchange>command:procdump.exe -ma picasvc.exe ?<dispchange>

□ Wait for command to complete

Startup Command: tasklist.exe > tasklistbefore.txt

□ Wait for command to complete

Shutdown Command: tasklist.exe > tasklistafter.txt

□ Wait for command to complete

```
C:\WINDOWS\system32\tasklist.exe
Verifying ShutdownCommands:
fullcommand:tasklist.exe > tasklistafter.txt
    command:
    arguments:> tasklistafter.txt
Command file exists:tasklist.exe > tasklistafter.txt
```

Execute on match

Execute on match if populated will execute the provided command(s). If configuring multiple commands for multiple matches, commands are separated by semicolon (;) and can be prefaced with a Regular expression named group for specific match to command association. Prefacing the command is not required and if not prefaced, all commands will be run on match. See below screenshot for example.

Command syntax to preface command: %named_group%command:.

- Example: ?<dispchange>command:procdump.exe -ma picasvc.exe

The regular expression named group also passes variables from matched trace statement if configured. To insert the variables

Command syntax to insert variable into command: %command% %named_group%

- Example: procdump.exe -ma picasvc.exe ?<dispchange>

Below is from output tab after pressing 'Verify'. Notice '1280' variable from trace string is populated into the execution string

- 9/2/2013 8:51:56 AM:fullcommand:?<dispchange>command:procdump.exe -ma picasvc.exe ?<dispchange>

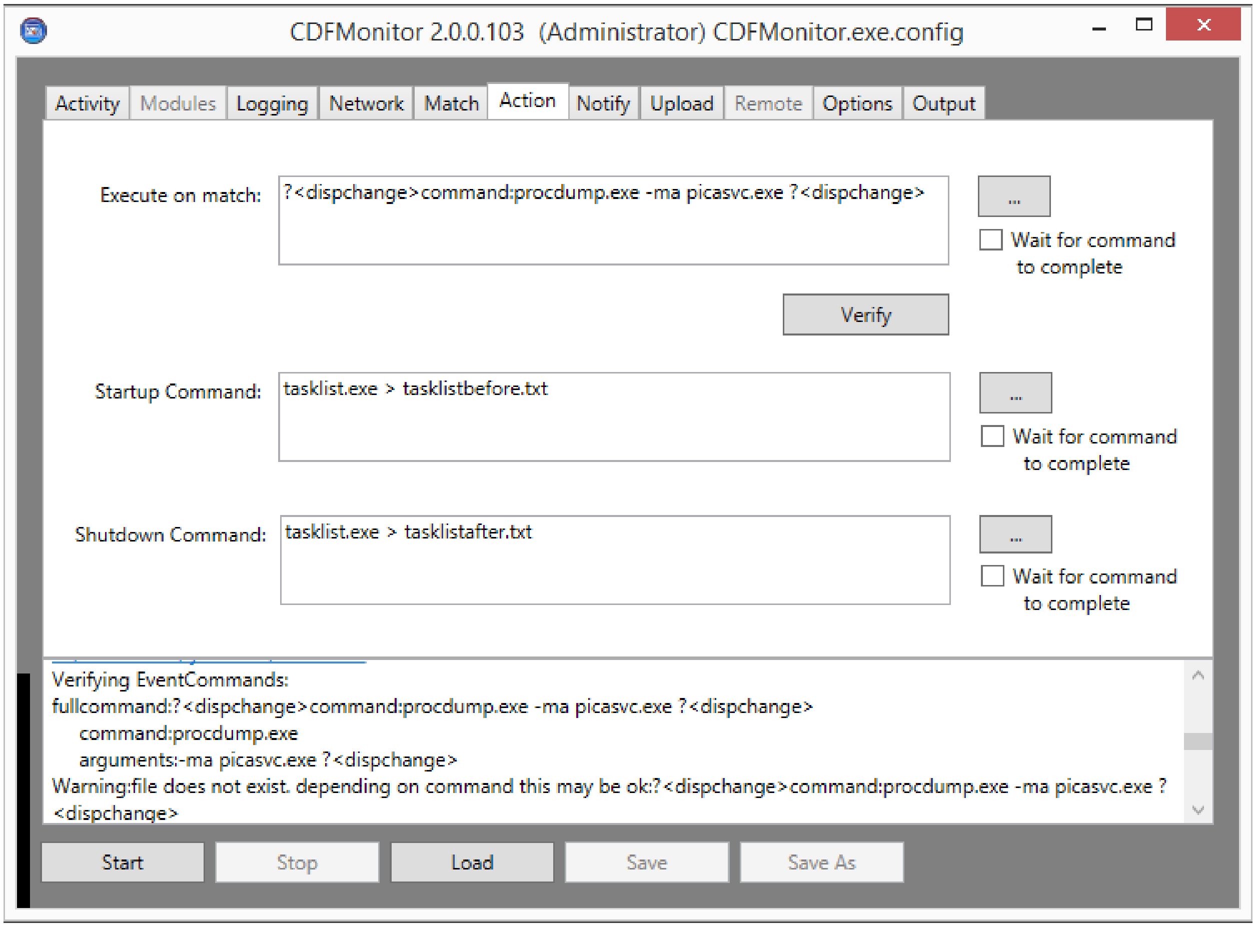
  command:procdump.exe

  arguments:-ma picasvc.exe ?<dispchange>
- Warning:file does not exist. depending on command this may be ok:?<dispchange>command:procdump.exe -ma picasvc.exe ?<dispchange>
- 9/2/2013 8:51:56 AM:starting command procdump.exe -ma picasvc.exe 1256 with wait=False

See Regular Expression Named Grouping for the Match configuration.

See RegexPattern Testing for verifying an example trace for match and action.

Example Action configuration using examples from Match documentation



Execute on match: ?<dispchange>command:procdump.exe -ma picasvc.exe ?<dispchange>

□ Wait for command to complete

Startup Command: tasklist.exe > tasklistbefore.txt

□ Wait for command to complete

Shutdown Command: tasklist.exe > tasklistafter.txt

□ Wait for command to complete

```
Verifying EventCommands:
fullcommand:?<dispchange>command:procdump.exe -ma picasvc.exe ?<dispchange>
    command:procdump.exe
    arguments:-ma picasvc.exe ?<dispchange>
Warning:file does not exist. depending on command this may be ok:?<dispchange>command:procdump.exe -ma picasvc.exe ?
<dispchange>
```

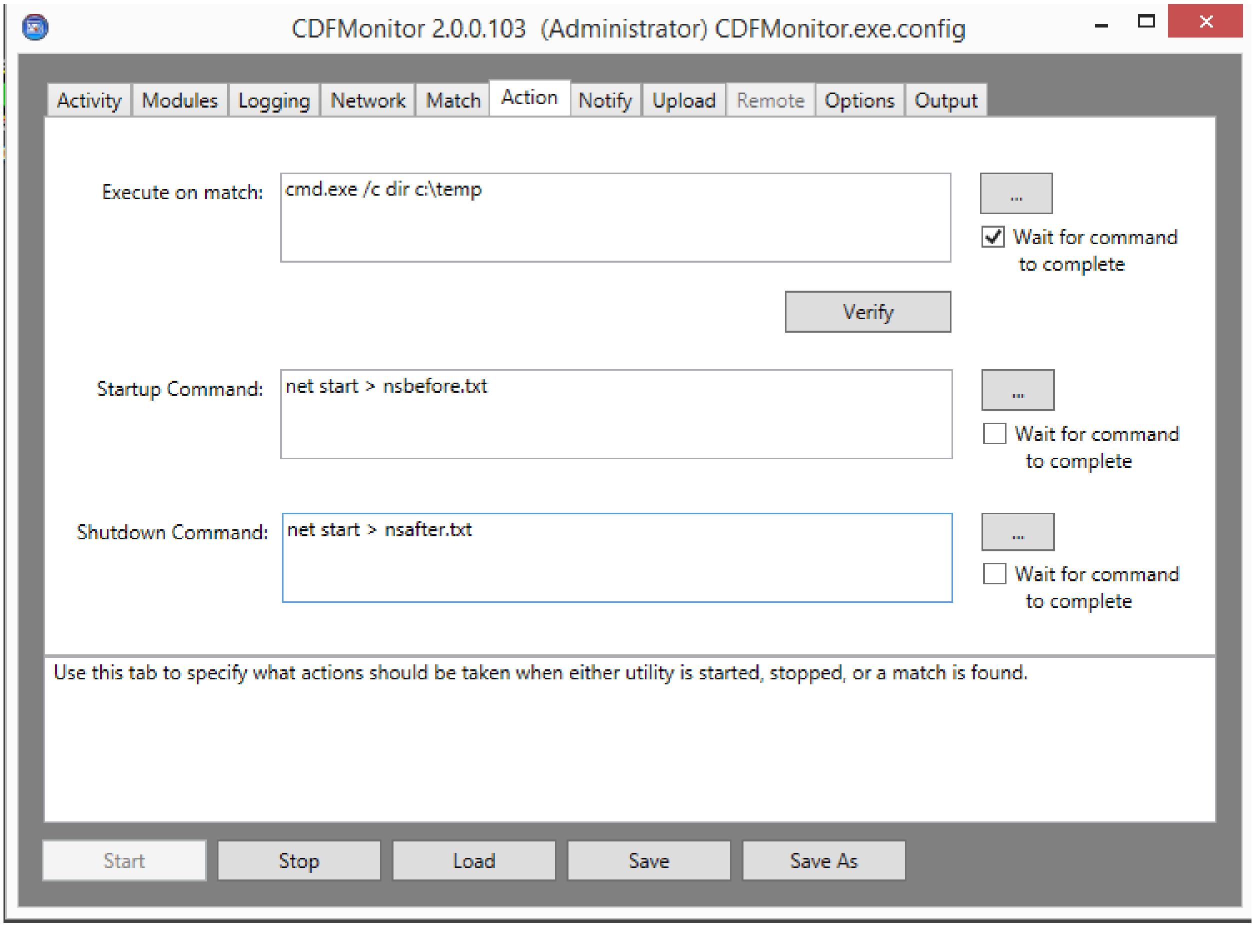
Startup and Shutdown Commands

Startup and Shutdown commands if populated will run the startup command(s) at process or service startup and shutdown command(s) at process or service shutdown. Multiple commands can be specified separated by a semicolon ';'.

As a general rule, if the command can run from Start -> Run command box then it should work in this utility. Use **Verify** and **Debug** on Logging tab to assist in troubleshooting. It may be easier to put the commands into a batch file or call a PowerShell script for further execution.

NOTE: As noted prior, specifically for Shutdown command shouldn't be used unless necessary especially with 'Wait for command to complete'. If the OS is shutting down for example, there are timers that could expire in OS to force this utility and child processes closed potentially creating unknown results or cause an issue.

If **Wait for command to complete** is selected, the command output will be traced out into console and log. Additionally an event with command results will be written to the Event Log with Event ID 101 if enabled on Match tab.



CDFMonitor 2.0.0.103 (Administrator) CDFMonitor.exe.config

Execute on match: cmd.exe /c dir c:\temp

☑ Wait for command to complete

Startup Command: net start > nsbefore.txt

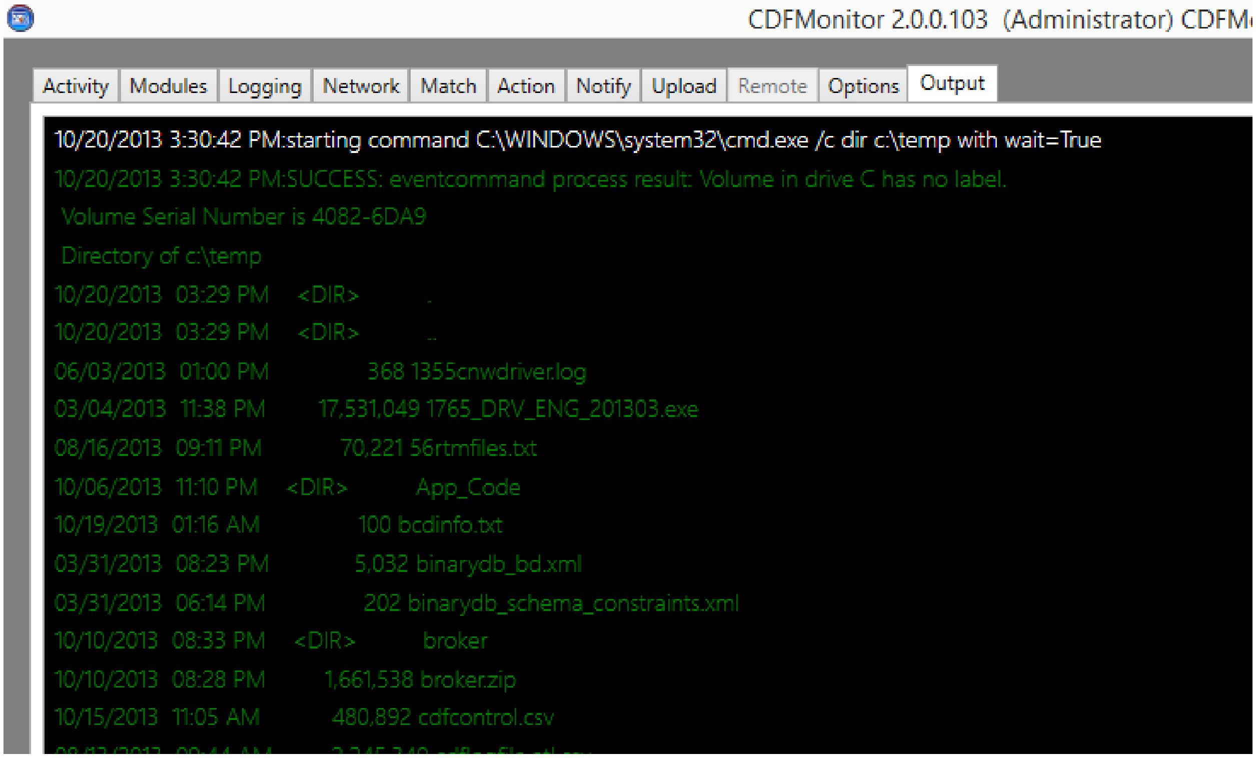
□ Wait for command to complete

Shutdown Command: net start > nsafter.txt

□ Wait for command to complete

Use this tab to specify what actions should be taken when either utility is started, stopped, or a match is found.

Example command wait results shown in console after a Match and Action completes



```
10/20/2013 3:30:42 PM:starting command C:\WINDOWS\system32\cmd.exe /c dir c:\temp with wait=True
10/20/2013 3:30:42 PM:SUCCESS: eventcommand process result: Volume in drive C has no label.
 Volume Serial Number is 4082-6DA9
 Directory of c:\temp
10/20/2013  03:29 PM    <DIR>          .
10/20/2013  03:29 PM    <DIR>          ..
06/03/2013  01:00 PM               368 1355cnwdriver.log
03/04/2013  11:38 PM        17,531,049 1765_DRV_ENG_201303.exe
08/16/2013  09:11 PM            70,221 56rtmfiles.txt
10/06/2013  11:10 PM    <DIR>          App_Code
10/19/2013  01:16 AM               100 bcdinfo.txt
03/31/2013  08:23 PM             5,032 binarydb_bd.xml
03/31/2013  06:14 PM               202 binarydb_schema_constraints.xml
10/10/2013  08:33 PM    <DIR>          broker
10/10/2013  08:28 PM         1,661,538 broker.zip
10/15/2013  11:05 AM           480,892 cdfcontrol.csv
```

NOTIFY

Notify is an advanced option that allows an SMTP email to be sent when a match occurs if below mandatory settings are properly populated and a RegexPattern is configured.

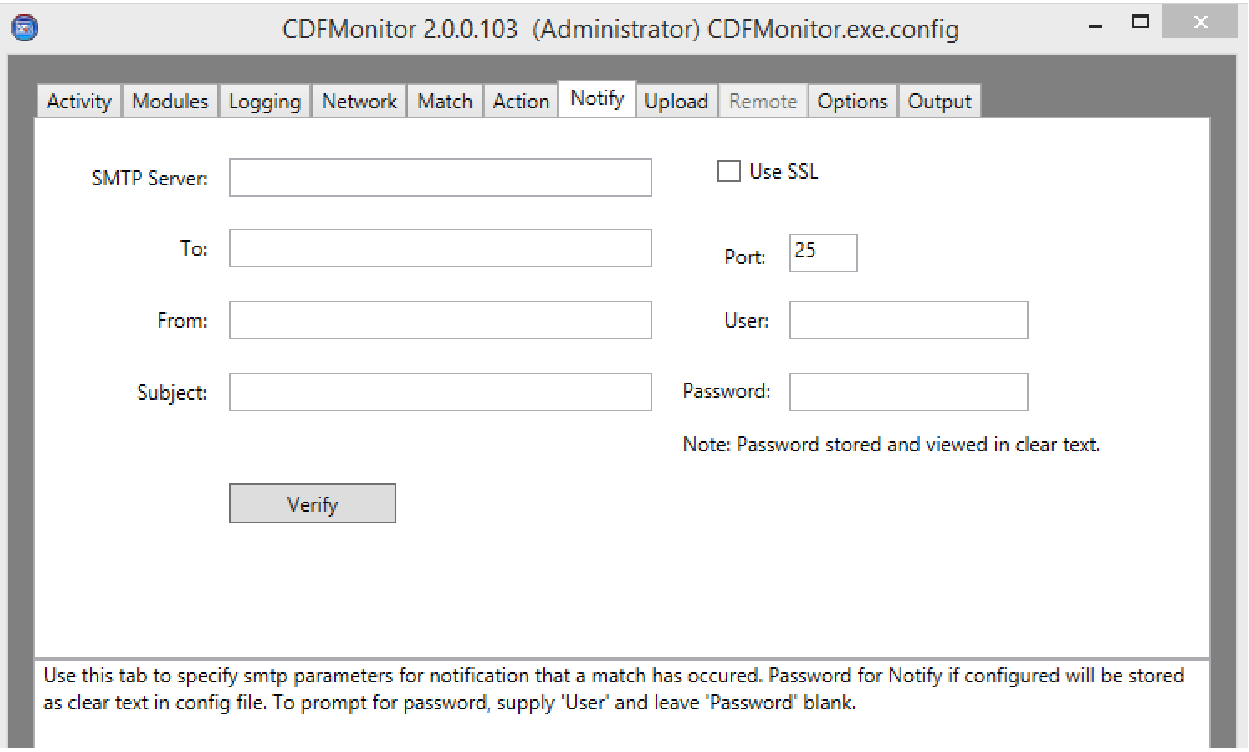
Verify button will try to send a test message using the provided configuration and display results. See screenshot below for example configuration and results from Verify.

NOTE: Password for this configuration if specified will be stored in clear text in config file.

NOTE: When setting up notification, in most use cases enabling **Event Throttle** on Match tab should be considered to prevent spamming from misconfiguration or too many matched events.

Notify Options:

- **SMTP Server -** mandatory setting used to specify FQDN or IP address of SMTP server to send email to for routing.
- **To -** mandatory setting used to specify email addresses separated by comma ',' to send notification to.
- **From** - mandatory setting used to specify where notification came from.
- **Subject** - not required but if specified will be appended to mandatory subject with machine name specified.
- **Use Ssl** - not required but if specified will connect to SMTP server using SSL.
- **Port** - not required but if specified will used given port number, else will use default port of 25.
- **User** - not required but if specified will be used when authenticating to SMTP.
- **Password** - not required but if specified will be sent with user. If user is specified and password is not specified, and an SMTP message is queued, a dialog prompt will be displayed to enter password so that it does not have to be stored in clear text.



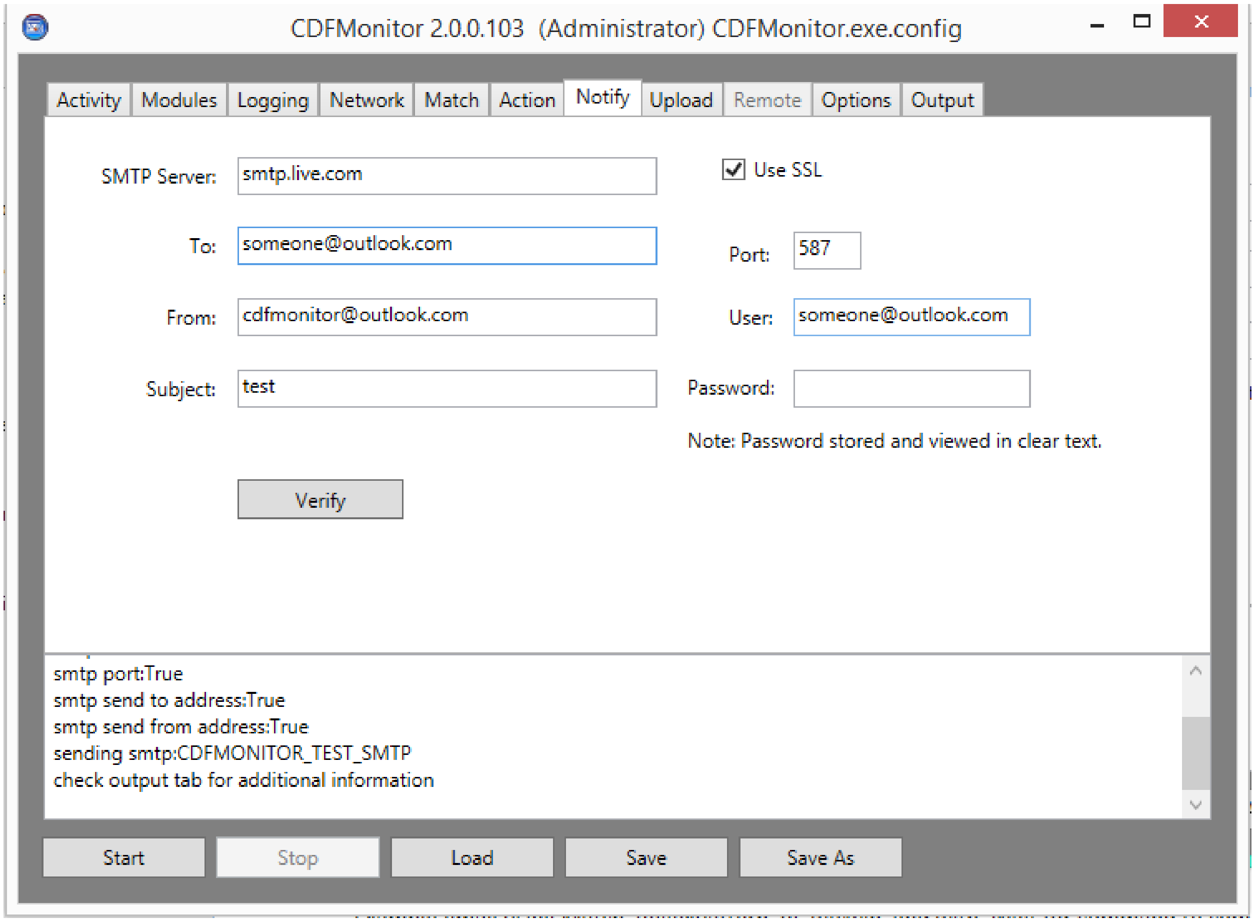
Port: 25

Note: Password stored and viewed in clear text.

Use this tab to specify smtp parameters for notification that a match has occured. Password for Notify if configured will be stored as clear text in config file. To prompt for password, supply 'User' and leave 'Password' blank.

Example Notify Configuration

The below screenshot shows an example configuration with **Verify** button results in the Status window at bottom.



SMTP Server: smtp.live.com

☑ Use SSL

To: someone@outlook.com

Port: 587

From: cdfmonitor@outlook.com

User: someone@outlook.com

Subject: test

Password:

Note: Password stored and viewed in clear text.

```
smtp port:True
smtp send to address:True
smtp send from address:True
sending smtp:CDFMONITOR_TEST_SMTP
check output tab for additional information
```

Example email from selecting **Verify** button:

From: cdfmonitor@outlook.com
To: someone@outlook.com
Date: Mon, 21 Oct 2013 15:57:33 -0400
Subject: test citrix-test-01

CDFMONITOR_TEST_SMTP

UPLOAD

Upload is an advanced option that, if mandatory settings are properly populated along with Match RegexPattern, will gather configured log files and URL files. It will then zip, and upload to ftp, http, or https sites. Log and trace files will be gathered automatically if enabled and do not have to be specified in **URL Files**.

Format of zip file name will be CDFMonitor-yyyy-mm-dd--hh-mm-ss.zip.

- Example: CFMonitor-2013-9-7--16-45-2.zip

Verify button will attempt to verify authentication to URL and will send step for test if SMTP is enabled.

NOTE: Password for this configuration if specified will be stored in clear text in config file.

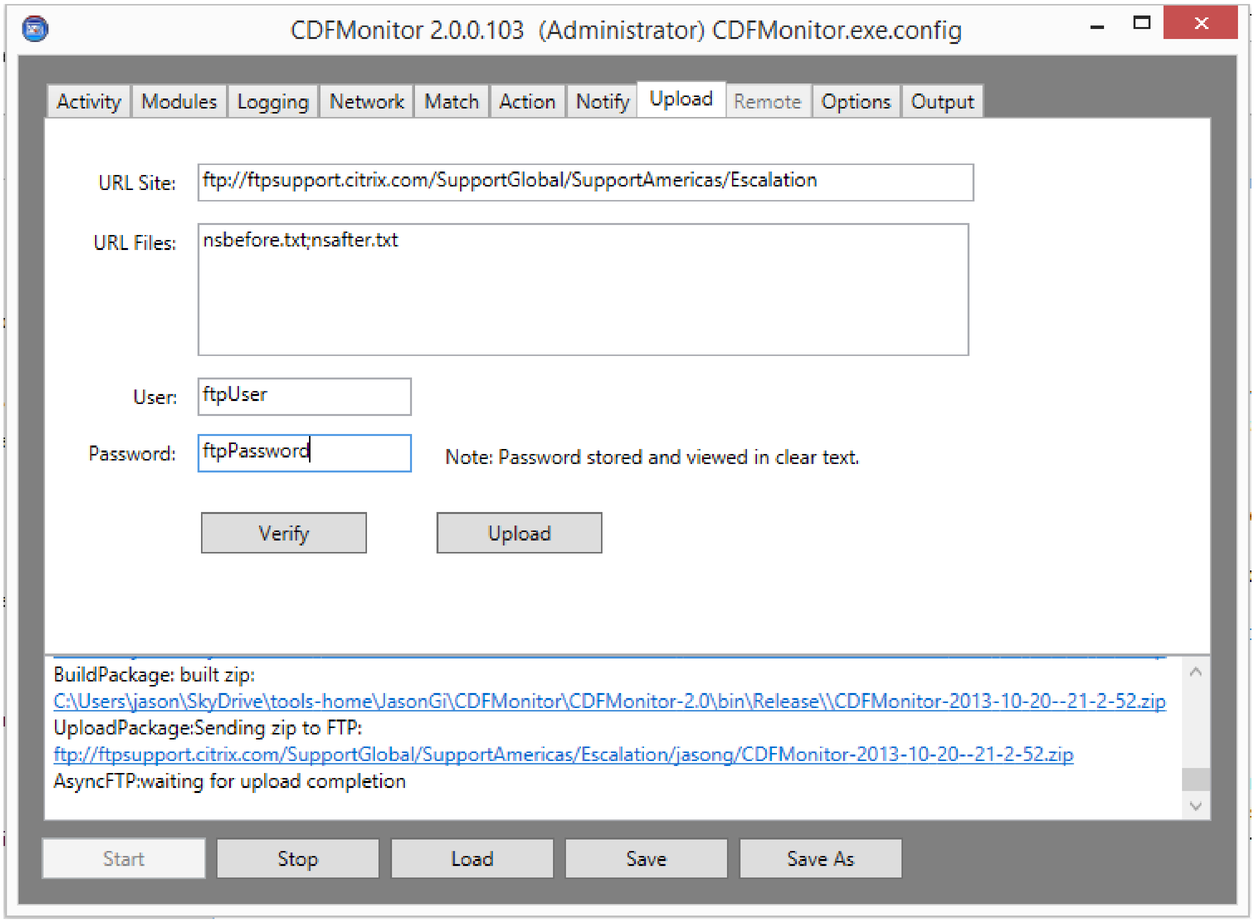
NOTE: When setting up upload, in most use cases enabling **Event Throttle** on Match tab should be considered to prevent spamming from misconfiguration or too many matched events.

Upload options:

- **URL Site** - mandatory setting that specifies the destination ftp, http, or https URL.
- **URL Files** - optional setting to configure a semicolon ';' separated list of any files to be included into zip file for upload. Full or relative path can be given. Note: this setting can be used to configure files to be gathered when using remote management activities even though 'Upload' is not desired or configured. To use this way, leave all other settings blank on this page except for 'URL Files'.
- **User** - not required but if specified will be used for authentication to URL site.
- **Password** - not required but if specified will be sent with user. If user is specified and password is not specified, and an upload is queued, a dialog prompt will be displayed to enter password so that it does not have to be stored in clear text.
- **Upload** will zip logs, traces, and configured URL files and will upload to configured **URL Site** the same as if a match had occurred.

Example Upload Configuration

This example shows an upload configuration with successful ftp **Upload** output in Status window:



URL Site: ftp://ftpsupport.citrix.com/SupportGlobal/SupportAmericas/Escalation

URL Files: nsbefore.txt;nsafter.txt

User: ftpUser

Password: ftpPassword

Note: Password stored and viewed in clear text.

```
BuildPackage: built zip:
C:\Users\jason\SkyDrive\tools-home\JasonGi\CDFMonitor\CDFMonitor-2.0\bin\Release\\CDFMonitor-2013-10-20--21-2-52.zip
UploadPackage:Sending zip to FTP:
ftp://ftpsupport.citrix.com/SupportGlobal/SupportAmericas/Escalation/jasong/CDFMonitor-2013-10-20--21-2-52.zip
AsyncFTP:waiting for upload completion
```

REMOTE

Remote management is an advanced option that is enabled when selecting **Server: Remote Management of clients** activity. Through the Remote tab as shown below, remote instances of CDFMonitor can be managed. CDFMonitor can be deployed, data can be gathered, and then undeployed all remotely.

Setup/Verify button will perform checks based on the Remote Action that is selected. **Setup/Verify** will also prompt to create/repair setup for that Remote Action. For all actions **Setup/Verify** will ping entries in the **Remote Machines** list. For **Deploy**, **Setup/Verify** will optionally create 'deploy' folder if there is no value in **Deploy Source Path** or repair it by copying CDFMonitor.exe and config file into the 'deploy' folder. For **Gather, Setup/Verify** will generate a 'gather' folder if **Gather Destination Path** is not configured or if path does not exist.

NOTE: For remote management, access to WMI on remote machine is required as well as access to the admin$ share. Alternate credentials can be enabled from the Options tab. See Options and Security for additional information.

NOTE: For performance reasons, before deploying a configuration to an environment, make sure to test first and verify configuration and resource utilization. It is best to deploy at least initially to a small subset of machines to verify performance and functionality.

Detailed Remote walkthroughs are available here:

- Walkthrough: Creating a remote configuration
- Walkthrough: Deploying a remote configuration
- Walkthrough: Gathering data from a remote configuration
- Walkthrough: Undeploying a remote configuration

Remote Options

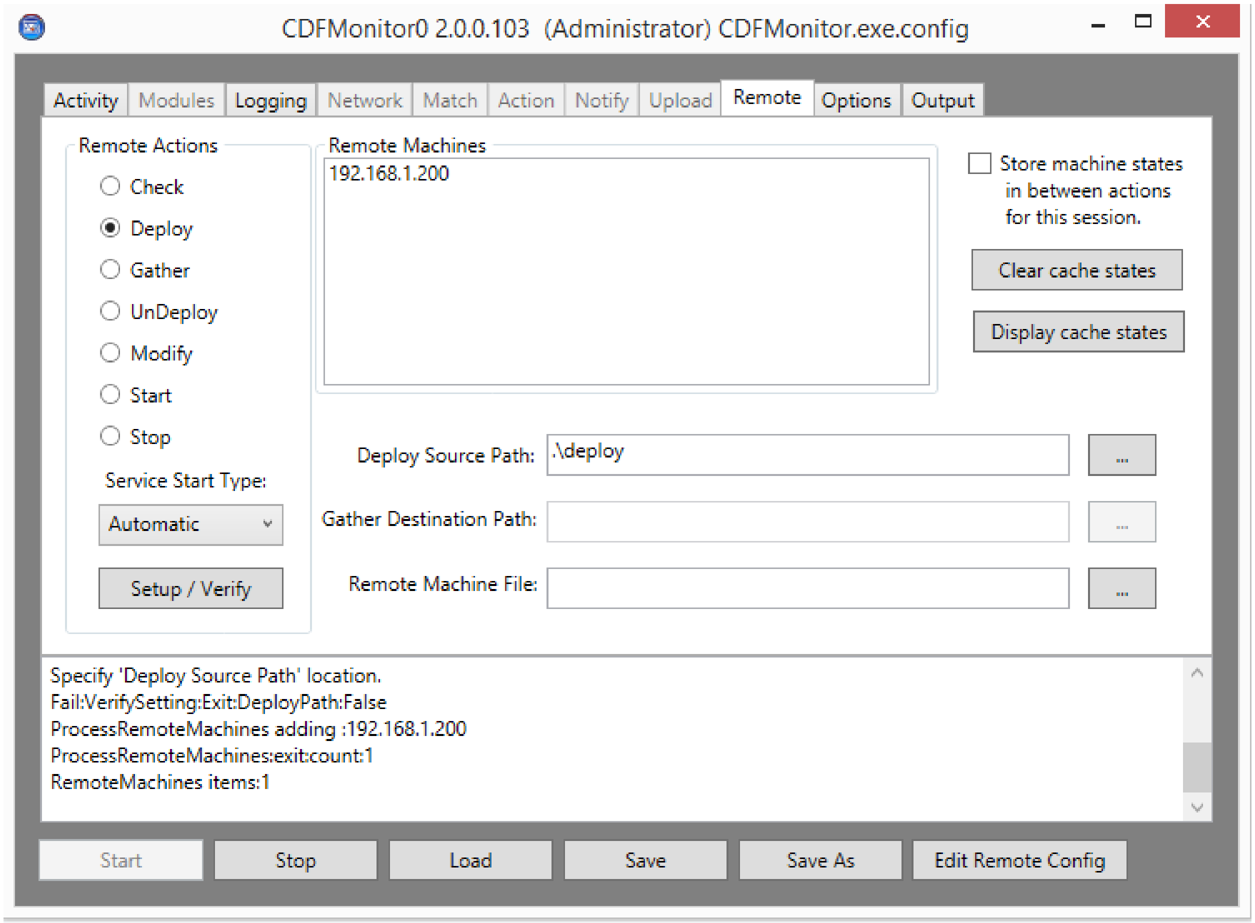
Below lists all the properties on the Remote tab for Remote machine management and their meanings.

- **Store machine states in between actions for this session** if selected, will keep the last known state for all machines listed while CDFMonitor stays running. It is not saved in config file. This is useful if running **Remote Actions** across multiple machines where some may have failed for example by being powered off at the time, will allow only those machines to be processed again the next time an action is started.
- **Clear cache states -** will clear out all remote machines cached states. The next **Remote Action** started will be processed across all remote machines in list.
- **Display cache states -** will enumerate all machines names and cached state to the Status window at bottom of window.

Remote Actions:

- **Check -** to contact all machines in **Remote Machines** list to see if CDFMonitor service is installed and if so the state of the service.
- **Deploy -** to deploy CDFMonitor and the config file located in the path specified in **Deploy Source Path** to all machines in **Remote Machines** list. The service will be deployed to %systemroot%\cdfmonitor via 'admin$' share. If the service already exists, it will be stopped for the source exe and config file to be copied and then the service will be started back. CDFMonitor running as a service will stay deployed until service is 'Undeployed' even across boots.
- **Gather -** to gather log files, trace files, and URL files from all machines in **Remote Machines** list. From each machine, gather will stop the service if started, read config file to determine files to zip, zip the files, and set service back to original state. The zip file will be created remotely and copied back to the path specified in **Gather Destination Path**. Each machines’ files will be staged in a machine named subdirectory. Format of zip file name will be CDFMonitor-yyyy-mm-dd--hh-mm-ss.zip. Example: CFMonitor-2013-9-7--16-45-2.zip.
- **UnDeploy -** to undeploy CDFMonitor from all machines in **Remote Machines** list and clean up the directories. This will not save any trace or log information. Use **Gather** first if logs are needed. If re-deploying a new configuration or exe file does not require **Undeploy** first.
- **Modify** will set all machines CDFMonitor service start type to the value configured for **Service Start Type** in **Remote Machines** list. If **Service Start Type** value is switched to 'Manual' or 'Disabled', service will be stopped. If **Service Start Type** is setup to 'Automatic', the service will be started. Modify does NOT modify the remote config, exe, or trace/log files.
- **Start** will start all machines CDFMonitor service in **Remote Machines** list.
- **Stop** will stop all machines CDFMonitor service in **Remote Machines** list.

Example Remote Deploy Configuration



Remote Machines: 192.168.1.200

Service Start Type: Automatic

Deploy Source Path: .\deploy

Gather Destination Path:

Remote Machine File:

```
Specify 'Deploy Source Path' location.
Fail:VerifySetting:Exit:DeployPath:False
ProcessRemoteMachines adding :192.168.1.200
ProcessRemoteMachines:exit:count:1
RemoteMachines items:1
```

Edit Remote Config

With Remote Management as the selected activity, all of the management options are on the Remote tab. Before deploying a configuration to a remote machine, a remote config file has to be created or specified.

NOTE: A default **Deploy Source Path** of 'deploy' will created in working directory of utility with exe and config file automatically by selecting **Setup / Verify** or **Edit Remote Config** buttons. See Walkthrough: Creating a remote configuration.

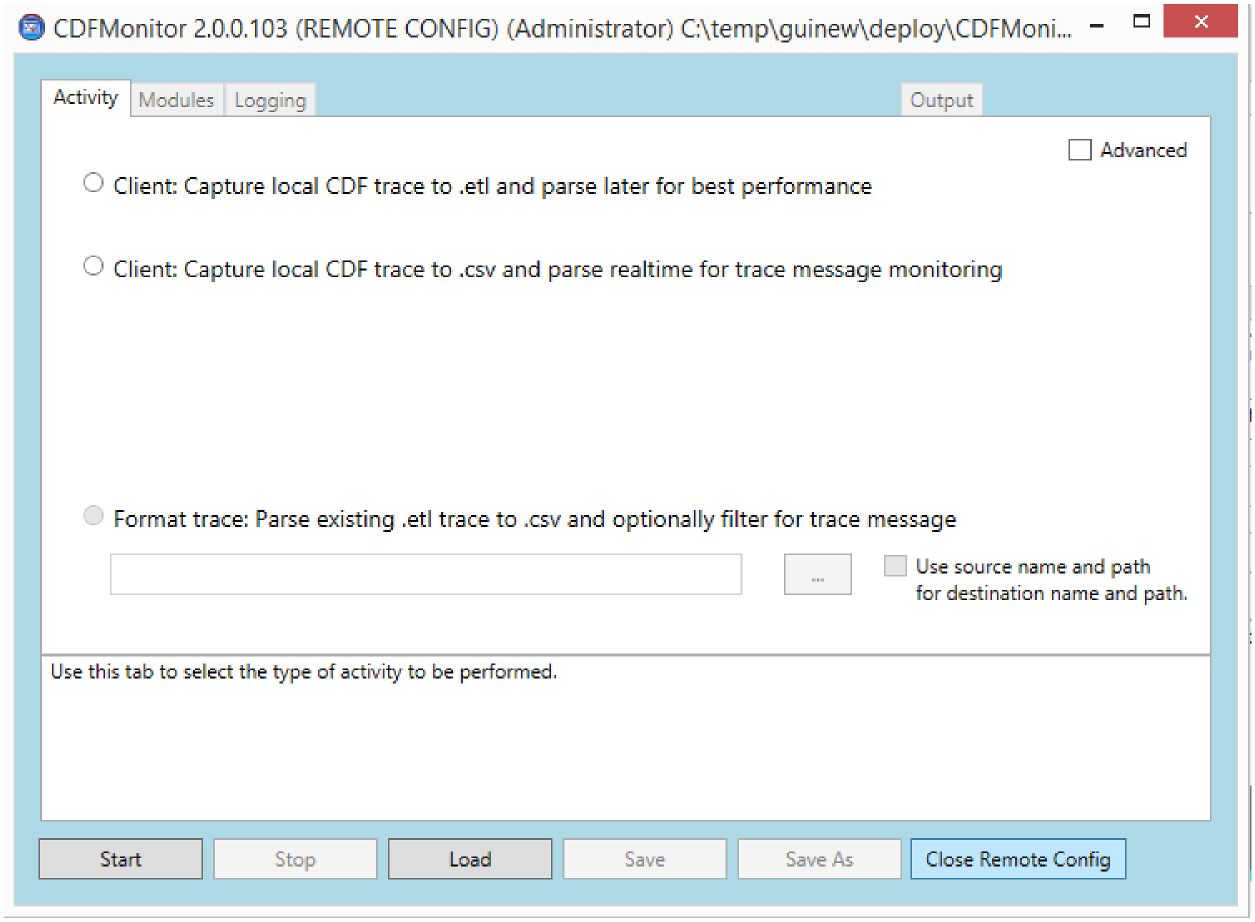
NOTE: This is editing the cdfmonitor.exe.config file in the folder configured in **Deploy Source Path** and is NOT editing a file on a remote machine.

General Guidelines for modifying a remote config:

- Select Remote Actions **Deploy** to modify remote configuration deploy source path. The **Deploy Source Path** configures the folder name containing cdfmonitor.exe and cdfmonitor.exe.config files that should be deployed.
- To edit an existing remote config file for deployment, configure **Deploy Source Path** with the existing folder name with configuration file.
- To create a new remote configuration file for deployment, configure **Deploy Source Path** with folder name where cdfmonitor.exe and new default cdfmonitor.exe.config will be generated and stored.
- When editing remote config, having at least one machine specified in **Remote Machines** will auto-populate the modules list in the remote config file as a selection if available. See Walkthrough: Creating a remote configuration.
- After selecting **Edit Remote Config**, the form will turn blue and the title will have (REMOTE CONFIG) in it as shown in screenshot below.
- Modify the remote configuration for deployment, **Save**, and **Close Remote Config**.

Example Remote Config Activity tab

The background changes and Title Bar will have ‘(REMOTE CONFIG)’ when viewing a remote configuration file.



CDFMonitor 2.0.0.103 (REMOTE CONFIG) (Administrator) C:\temp\guinew\deploy\CDFMoni...

□ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

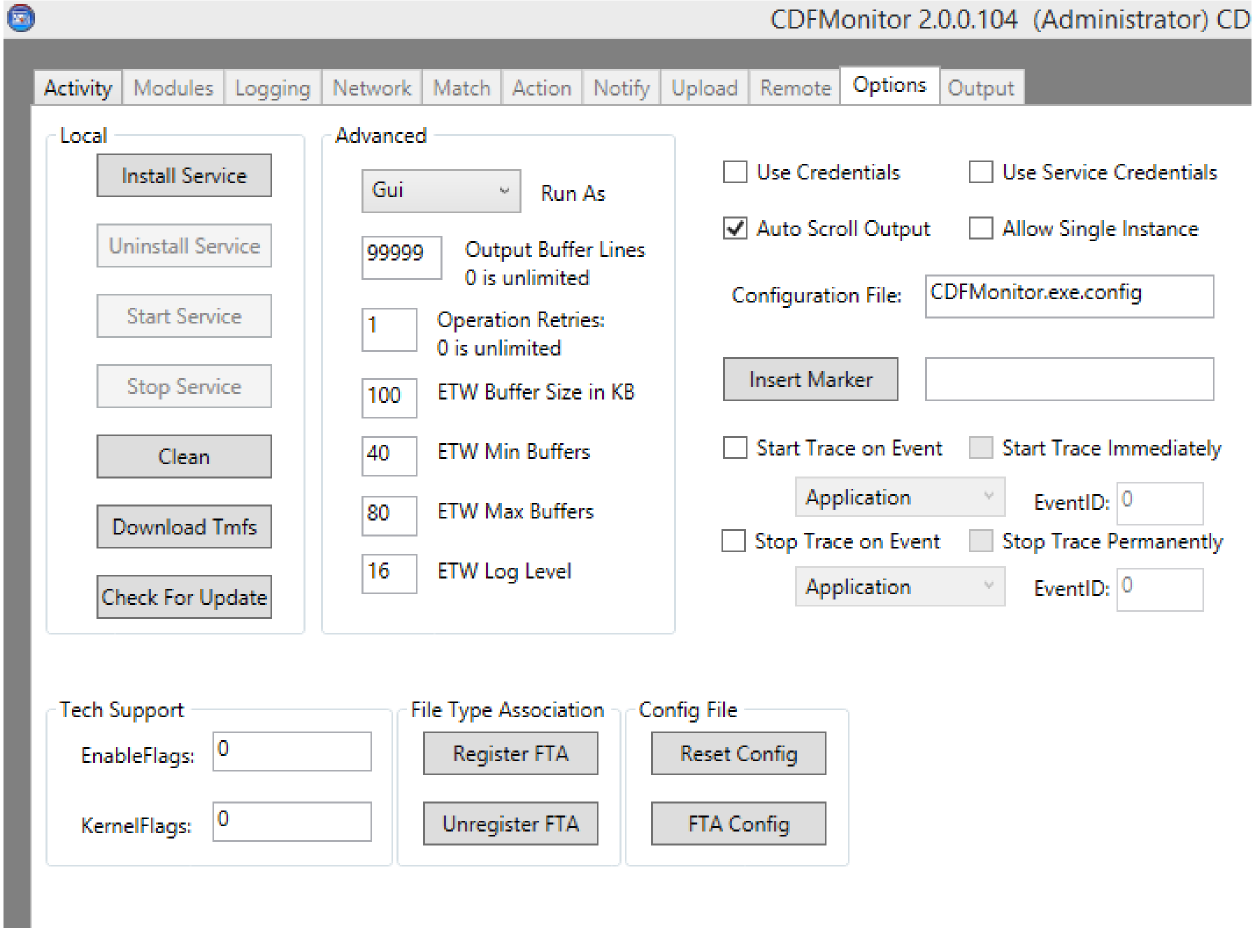
○ Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

□ Use source name and path for destination name and path.

Use this tab to select the type of activity to be performed.

OPTIONS

The Options tab is available when checking 'Advanced' on the Activity tab. The Options tab contains options for local service management, advanced CDFMonitor and ETW settings, alternate credentials, and Windows Event Log Event ID options.



Gui Run As

99999 Output Buffer Lines 0 is unlimited

1 Operation Retries: 0 is unlimited

100 ETW Buffer Size in KB

40 ETW Min Buffers

80 ETW Max Buffers

16 ETW Log Level

☐ Use Credentials ☐ Use Service Credentials

☑ Auto Scroll Output ☐ Allow Single Instance

Configuration File: CDFMonitor.exe.config

☐ Start Trace on Event ☐ Start Trace Immediately

Application EventID: 0

☐ Stop Trace on Event ☐ Stop Trace Permanently

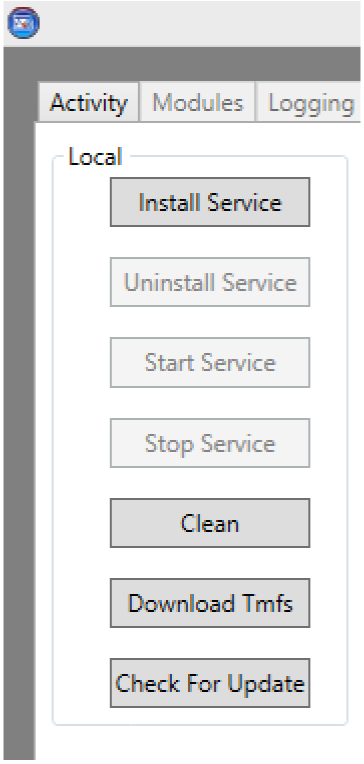
Application EventID: 0

EnableFlags: 0

KernelFlags: 0

Options Local

Local actions available on the Options tab are used to manage CDFMonitor on local machine. When switching to Options tab, 'Local' buttons enablement will reflect the current local CDFMonitor service state.

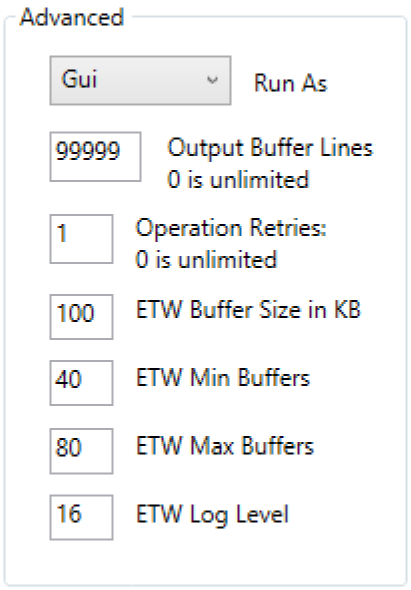
- **Install Service** - if enabled and selected will install CDFMonitor as a service on the local machine. Install Service will copy CDFMonitor.exe and config file to %systemroot%\cdfmonitor, install service, and start service. Once service is installed, be sure to 'Load' the configuration file located in %systemroot%\cdfmonitor as this is not done automatically. Install service can be used in conjunction with 'Remote management' of CDFMonitor services. See Remote for additional information.
- **Uninstall Service** - if enabled and selected will uninstall CDFMonitor service on the local machine. Uninstall Service will stop and uninstall CDFMonitor service, delete CDFMonitor.exe and config file from %systemroot%\cdfmonitor. **Uninstall Service** will not save any log or trace files, this needs to be done manually before uninstalling.
- **Start Service** - if enabled and selected will start CDFMonitor service on the local machine.
- **Stop Service** - if enabled and selected will stop CDFMonitor service on the local machine.
- Clean - if selected will clean the local machine CDFMonitor files and configurations. If CDFMonitor did not terminate correctly prior or is not functioning correctly, using 'Clean' can resolve some issues. Clean will clear all 'CDFMonitorx' sessions from ETW, delete TMF cache directory, and will delete all log and trace files.
- **Download TMFs** - if selected will download all TMFs from path specified in **TMF Servers** on **Match** tab. This is useful if setting up an internal TMF server. An example would be staging with CDFMonitor installs that do not have access to internet.
- **Check For Update** - if selected will check to see if there is a newer version of CDFMonitor available for download.
  - 

Options Advanced

Advanced settings on the Options tab are sufficient for most use cases. A couple of notable exceptions are **Output Buffer Lines** and **Run As**. If using CDFMonitor for parsing or real-time tracing, using filtering on the Output tab is useful to find traces as they occur. **Output Buffer Lines** controls size of buffer for screen output on buffer tab and directly affects CPU and memory. If there are no resource bottlenecks, the buffer can be increased to retain a larger amount of traces in Output for filtering. **Run As** controls how the window is displayed and if CDFMonitor is to be used as a console application instead of a GUI similar to previous versions.

The following are the configurable advanced options:

- **Run As - GUI** is default and should only be changed if using the GUI interface is not desired. Options are:
    - **Unknown** - should not be used.
    - **Console** - should be selected when using command prompt console instead of the GUI is desired. Once this is set, to view the GUI again, edit configuration file manually and set **RunAs** to GUI.
    - **Hidden** - used to run CDFMonitor as a process as opposed to a service with no console or GUI. The process will only show in process list / task manager and may have to be terminated from there.
    - **Service** - does not modify functionality of CDFMonitor but is set when creating/modifying remote config files for deployment and documentation purposes.
- **Output Buffer Lines** - (0 - 999999) configures the number of lines stored in Output buffer for Output tab display. Increasing this number increases amount of CPU and memory required when performing real-time tracing or parsing traces.
- **Operation Retries** - (0+) is the number of times a remote operation is retried if first try fails.
- **NOTE:** Increasing the below settings can help with 'Missed Controller Events' due from the amount of traces being generated is causing bottlenecks in performance at the expense of memory.
- **ETW Buffer Size** - (0 - 1024) is the size of ETW buffer configured when performing real-time tracing (trace to csv or trace to etl).
- **ETW Min Buffers** - 2x #Processors is the minimum number of buffers configured for real-time tracing (trace to csv or trace to etl).
- **ETW Max Buffers** - (Greater than Min Buffers) is the maximum number of buffers configured for real-time tracing (trace to csv or trace to etl).
- **ETW Log Level** - (0 - 16) is the ETW Log Level configured for real-time tracing (trace to csv or trace to etl). The higher the number the more verbose the trace is. Not all Citrix products use multiple tracing levels.



Advanced

Gui Run As

99999 Output Buffer Lines
0 is unlimited

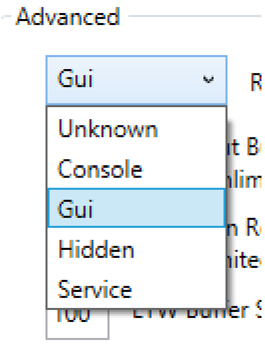
1 Operation Retries:
0 is unlimited

100 ETW Buffer Size in KB

40 ETW Min Buffers

80 ETW Max Buffers

16 ETW Log Level

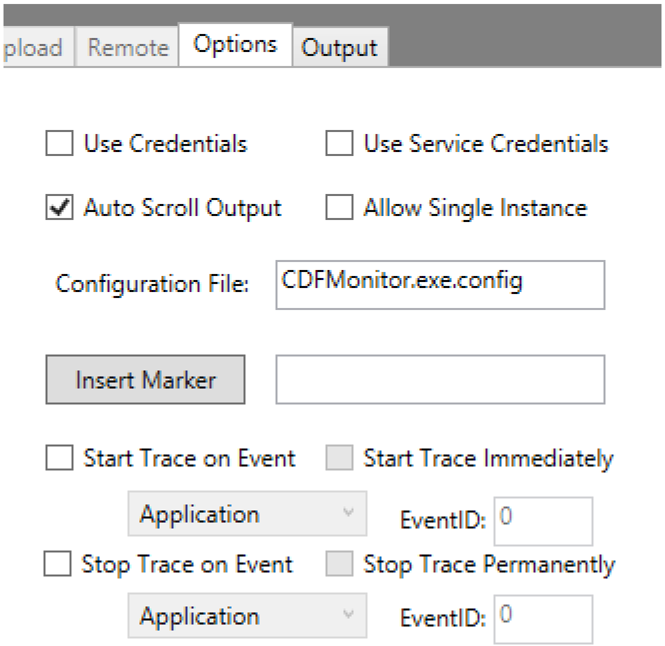


Advanced

Gui

Options Miscellaneous

- **Use Credentials** - allows the use of alternate credentials for remote management and for file and directory management. Currently logged on user credentials will be tried first, then specified alternate credentials stored in Windows 'Credential Manager' as 'CDFMonitor' (see below), then finally will prompt for credentials. When prompting for credentials, if credentials for that resource has been stored in Credential Manager then those credentials will be used.
- **Use Service Credentials** - uses same credentials stored for 'Use Credentials' and is only used when deploying CDFMonitor remotely using remote management. When this is selected, the alternate credentials will be used for the service startup/logon authentication. This is useful if CDFMonitor running as a service needs to connect to a remote file or directory. By default when installing CDFMonitor to run as a service, the service uses LOCAL_SYSTEM account due to administrative requirements for ETW/CDF tracing.
- **NOTE:** By default current logged on credentials are used. If 'Use Credentials' is checked, alternate credentials will be used for network activities including Remote Management. If 'Use Service Credentials' is also checked, when the Remote Management Activity is 'Deploy', the alternate credentials will be used for the remote service 'Log On As' authentication. By default this is set to 'LOCAL_SYSTEM' due to admin requirements for ETW activities.
- **Auto Scroll Output** - if checked (default) will auto-scroll trace and message output on the output tab.
- **Allow Single Instance** - if checked will not allow multiple instances of CDFMonitor to be running.
- **Configuration File** - if not default or empty will be used to override the default (cdfmonitor.exe.config) file on process startup. Configuration File can point to a local or remote path. This value can also be a URL which is useful if wanting to maintain a central configuration file. This works by checking this value before processing any other values in config, if different, the new config will be saved and thus overwrite the initial config, then processed and used. Note: The 'Central' config file 'Configuration File' value should point to itself for the above process to work.
- **Insert Marker** - will write a CDFMonitor Marker event to output and directly into the etl file being traced with an incrementing number. An optional string value can be added to the trace message.
- **Start Trace on Event** - if selected will start tracing if not already started when specified Event ID has been written to the specified Windows Event Log.
- **Start Trace Immediately** - if selected, will start tracing as soon as process/service starts if 'Start Trace on Event' is selected. Otherwise, if not selected, initial tracing will not start until first event log event is written matching **EventID**.
- **Stop Trace on Event** - if selected, will stop tracing if not already stopped when specified Event ID has been written to the specified Windows Event Log.
- **Stop Trace Permanently** - if selected, will stop process/service and will not restart tracing if a Start Trace Event ID is configured. Otherwise if not selected, Stop Trace can be used in conjunction with Start Trace to start and stop tracing multiple times without restarting process/service.
- For more information on events see Walkthrough: Using events to capture a trace to .etl file.



☐ Use Credentials ☐ Use Service Credentials

☑ Auto Scroll Output ☐ Allow Single Instance

Configuration File: CDFMonitor.exe.config

☐ Start Trace on Event ☐ Start Trace Immediately

Application EventID: 0

☐ Stop Trace on Event ☐ Stop Trace Permanently

Application EventID: 0

Register File Type Association (FTA)

Setting optional file type association (FTA) for .etl files for CDFMonitor enables parsing of .etl files to .csv format using specified configuration. Performing the following will enable automatic parsing of .etl files into csv using the same name and path for destination. Up to 8 .etl files can be selected at a time.

The following settings will be set initially but can later be overridden:

Activity = Regex Parse To Csv

RunAs = Console

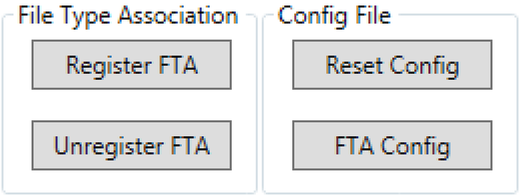
Use Trace Source For Destination = true

Log To Console = false

Log File Auto Flush = false

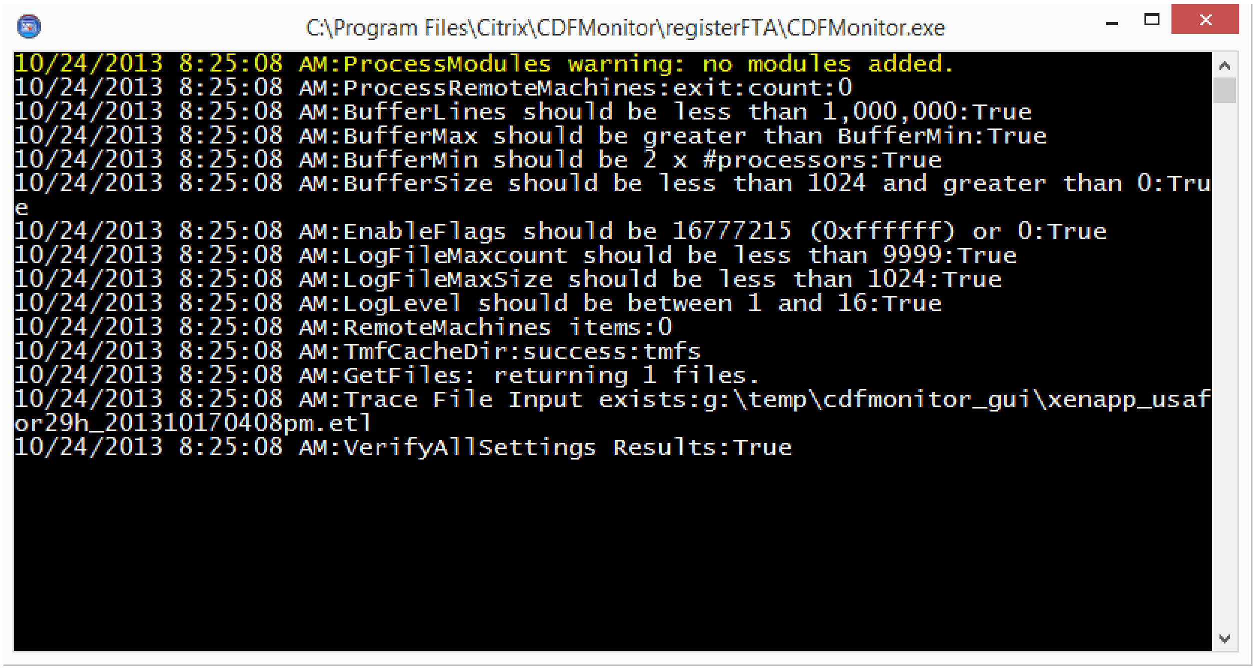
Use the following steps to setup file type association.

- Open CDFMonitor
- Select: Options tab and Maximize window to view the buttons below.



- Optional: Select **FTA Config**
    - **NOTE:** If using **FTA Config**, select **Reset Config** first.
    - **FTA Config** will set the following optimal settings:
        - **Runas** = **Console**
        - **Use Trace Source For Destination** = **True**
        - **Log To Console** = **False**
        - **Log File Auto Flush** = **False**
        - **Log Match Detail** = **True**
        - **Activity** = **RegexParseToCsv**
        - **Log File Max Count** = **0**
        - **Log File Max Size** = **0**
        - **Log File Overwrite** = False
    - After selecting **FTA Config**, any other custom settings can be applied in the current local config file before selecting **Register FTA.**
    - If using these settings, when parsing a trace, CDFMonitor will look like the figure shown below. Additionally selecting ‘s’ and ‘enter’ will show the status information.
- Select: **Register FTA**
    - This will copy cdfmonitor.exe and config file to '%ProgramFiles%\Citrix\cdfmonitor\registerfta'
    - **NOTE:** Use command line to register FTA from a custom directory
- After selecting **Register FTA**, to modify FTA configuration, modify the config in the register FTA directory by opening CDFMonitor from that folder. Another option is to **Unregister FTA**, make changes to current local configuration, then selecting **Register FTA** to re-register the file type association. Etl files can be parsed directly now by double clicking the .etl file.

Selecting **FTA Config** and double clicking an etl file to be parsed will start CDFMonitor in console as shown below.

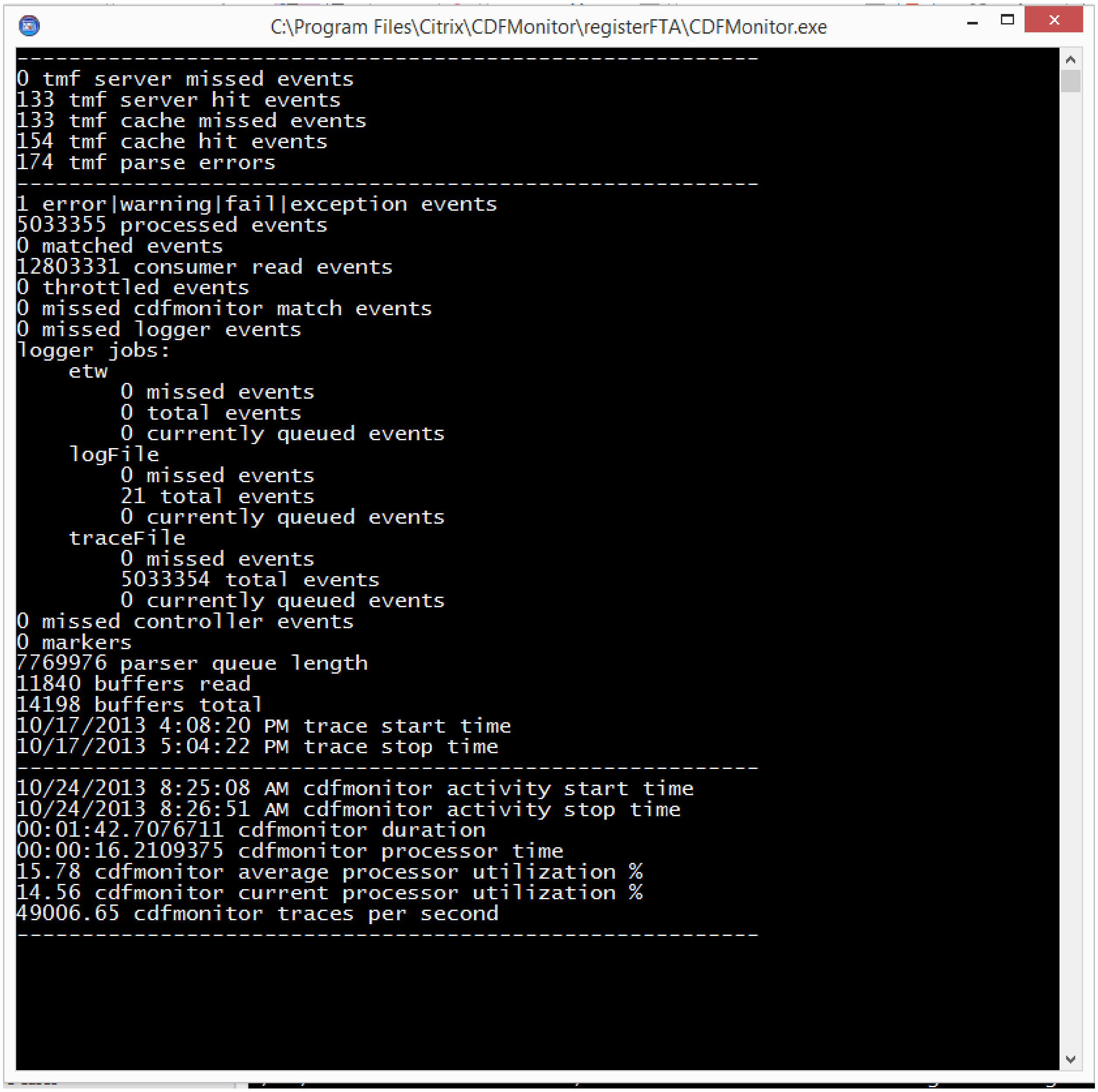


C:\Program Files\Citrix\CDFMonitor\registerFTA\CDFMonitor.exe

```
10/24/2013 8:25:08 AM:ProcessModules warning: no modules added.
10/24/2013 8:25:08 AM:ProcessRemoteMachines:exit:count:0
10/24/2013 8:25:08 AM:BufferLines should be less than 1,000,000:True
10/24/2013 8:25:08 AM:BufferMax should be greater than BufferMin:True
10/24/2013 8:25:08 AM:BufferMin should be 2 x #processors:True
10/24/2013 8:25:08 AM:BufferSize should be less than 1024 and greater than 0:Tru
e
10/24/2013 8:25:08 AM:EnableFlags should be 16777215 (0xffffff) or 0:True
10/24/2013 8:25:08 AM:LogFileMaxcount should be less than 9999:True
10/24/2013 8:25:08 AM:LogFileMaxSize should be less than 1024:True
10/24/2013 8:25:08 AM:LogLevel should be between 1 and 16:True
10/24/2013 8:25:08 AM:RemoteMachines items:0
10/24/2013 8:25:08 AM:TmfCacheDir:success:tmfs
10/24/2013 8:25:08 AM:GetFiles: returning 1 files.
10/24/2013 8:25:08 AM:Trace File Input exists:g:\temp\cdfmonitor_gui\xenapp_usaf
or29h_201310170408pm.etl
10/24/2013 8:25:08 AM:VerifyAllSettings Results:True
```

Example console window with CDFMonitor Optimized for parsing.

Selecting ‘s’ and ‘enter’ in console will show this real-time status output while parsing if **FTA Config** was used. See Show Stats / Activity Summary for additional information.


C:\Program Files\Citrix\CDFMonitor\registerFTA\CDFMonitor.exe

```
---------------------------------------------------------
0 tmf server missed events
133 tmf server hit events
133 tmf cache missed events
154 tmf cache hit events
174 tmf parse errors
---------------------------------------------------------
1 error|warning|fail|exception events
5033355 processed events
0 matched events
12803331 consumer read events
0 throttled events
0 missed cdfmonitor match events
0 missed logger events
logger jobs:
    etw
        0 missed events
        0 total events
        0 currently queued events
    logFile
        0 missed events
        21 total events
        0 currently queued events
    traceFile
        0 missed events
        5033354 total events
        0 currently queued events
0 missed controller events
0 markers
7769976 parser queue length
11840 buffers read
14198 buffers total
10/17/2013 4:08:20 PM trace start time
10/17/2013 5:04:22 PM trace stop time
---------------------------------------------------------
10/24/2013 8:25:08 AM cdfmonitor activity start time
10/24/2013 8:26:51 AM cdfmonitor activity stop time
00:01:42.7076711 cdfmonitor duration
00:00:16.2109375 cdfmonitor processor time
15.78 cdfmonitor average processor utilization %
14.56 cdfmonitor current processor utilization %
49006.65 cdfmonitor traces per second
---------------------------------------------------------
```

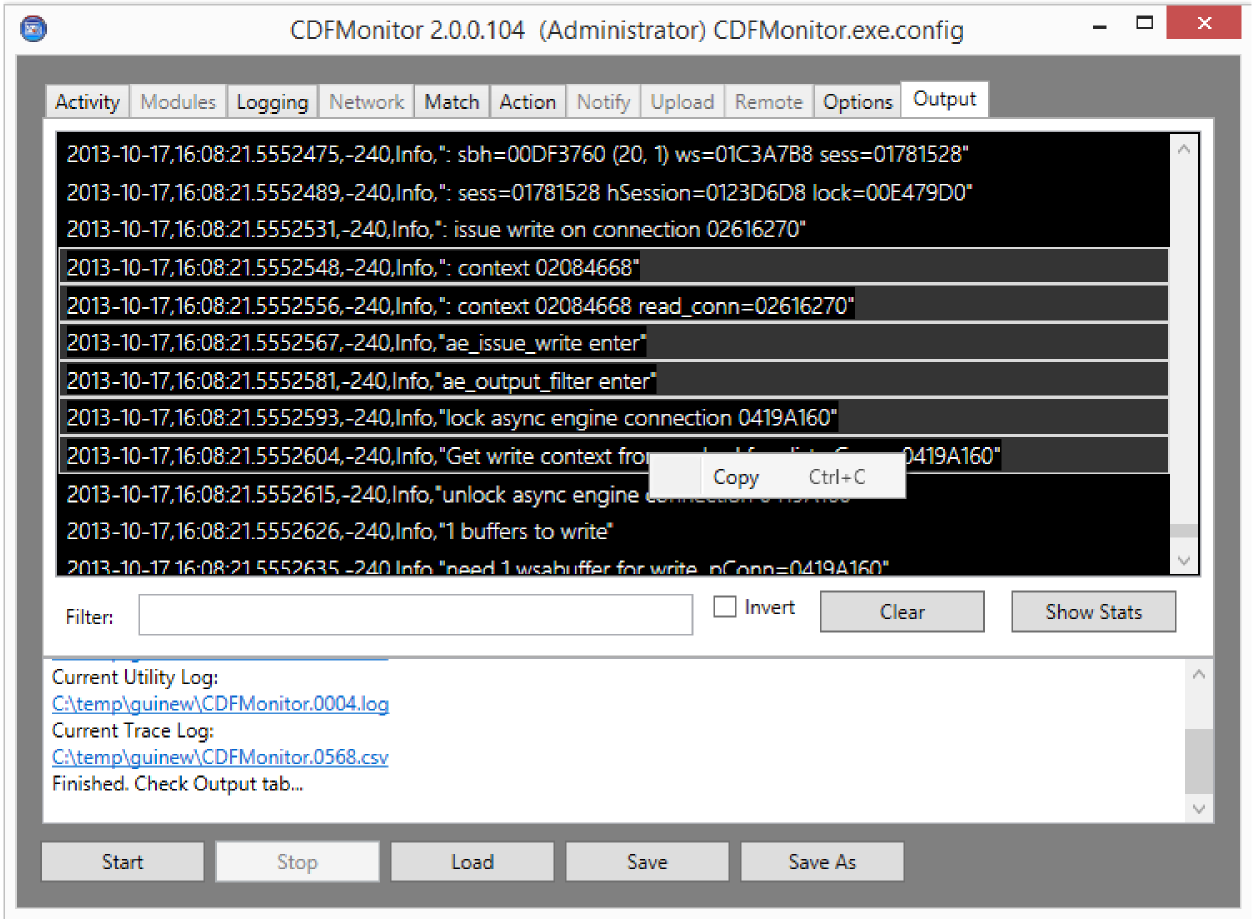
OUTPUT

Output Tab will optionally show all output from utility. This includes, CdfMonitor messages, Trace messages, and debug messages. Output enablement is controlled by the **Log To Console** check box on the Logging tab which is enabled by default. Everything displayed in the Output console will be written to a log or trace file if those files are configured.

The buffer for the Output console is controlled by the **Buffer Lines** setting on the Options tab. By default this is set to 99999. Setting to 0 will let buffer grow unlimited and this is not recommended. Performance will degrade with a bigger buffer. The one advantage for having a larger buffer though is for using the **Filter** option to search/filter output.

NOTE: If processing large amounts of data, better performance can be achieved by unchecking **Log To Console** on the Logging tab to disable all output to Output tab.

Multiple messages can be copied out of the Output window in the shown text format by using standard methods. Both ‘Ctrl-C’ and right clicking the message can be used for selection.



```
2013-10-17,16:08:21.5552475,-240,Info,": sbh=00DF3760 (20, 1) ws=01C3A7B8 sess=01781528"
2013-10-17,16:08:21.5552489,-240,Info,": sess=01781528 hSession=0123D6D8 lock=00E479D0"
2013-10-17,16:08:21.5552531,-240,Info,": issue write on connection 02616270"
2013-10-17,16:08:21.5552548,-240,Info,": context 02084668"
2013-10-17,16:08:21.5552556,-240,Info,": context 02084668 read_conn=02616270"
2013-10-17,16:08:21.5552567,-240,Info,"ae_issue_write enter"
2013-10-17,16:08:21.5552581,-240,Info,"ae_output_filter enter"
2013-10-17,16:08:21.5552593,-240,Info,"lock async engine connection 0419A160"
2013-10-17,16:08:21.5552604,-240,Info,"Get write context fro                    0419A160"
2013-10-17,16:08:21.5552615,-240,Info,"unlock async engine
2013-10-17,16:08:21.5552626,-240,Info,"1 buffers to write"
2013-10-17,16:08:21.5552635,-240,Info,"need 1 wsabuffer for write, pConn=0419A160"
```

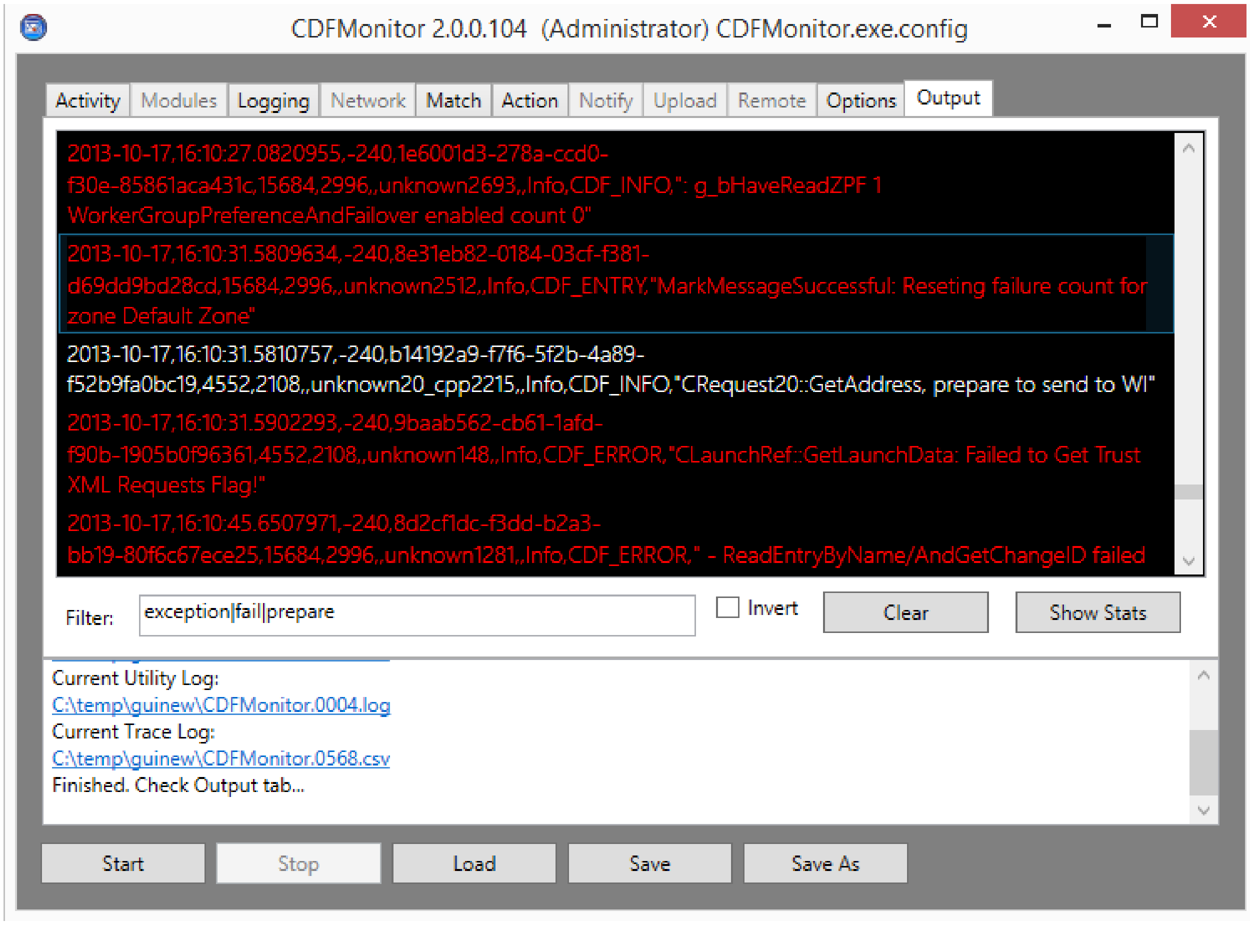
Current Utility Log:
C:\temp\guinew\CDFMonitor.0004.log
Current Trace Log:
C:\temp\guinew\CDFMonitor.0568.csv
Finished. Check Output tab...

Display Filter

Display filter is an easy and fast way to search through output post processing or real-time tracing. The filter does not affect what is being written to log, only what is shown on the console. The filter will filter all items in the buffer until they are forced out.

Filter is regex based and can be saved in the config file like other fields. Search patterns can be simple strings separated by pipe '|'. Wildcards are '.' for any single character and '*' for any number of matches. For regex, the escape character is backslash '\'. Pattern matching is not case sensitive. Any valid regex search command should work in the filter.

Invert will display all items in buffer except what is matched in the **Display Filter** regex search pattern.



```
2013-10-17,16:10:27.0820955,-240,1e6001d3-278a-ccd0-f30e-85861aca431c,15684,2996,,unknown2693,,Info,CDF_INFO,": g_bHaveReadZPF 1 WorkerGroupPreferenceAndFailover enabled count 0"
2013-10-17,16:10:31.5809634,-240,8e31eb82-0184-03cf-f381-d69dd9bd28cd,15684,2996,,unknown2512,,Info,CDF_ENTRY,"MarkMessageSuccessful: Reseting failure count for zone Default Zone"
2013-10-17,16:10:31.5810757,-240,b14192a9-f7f6-5f2b-4a89-f52b9fa0bc19,4552,2108,,unknown20_cpp2215,,Info,CDF_INFO,"CRequest20::GetAddress, prepare to send to WI"
2013-10-17,16:10:31.5902293,-240,9baab562-cb61-1afd-f90b-1905b0f96361,4552,2108,,unknown148,,Info,CDF_ERROR,"CLaunchRef::GetLaunchData: Failed to Get Trust XML Requests Flag!"
2013-10-17,16:10:45.6507971,-240,8d2cf1dc-f3dd-b2a3-bb19-80f6c67ece25,15684,2996,,unknown1281,,Info,CDF_ERROR," - ReadEntryByName/AndGetChangeID failed
```

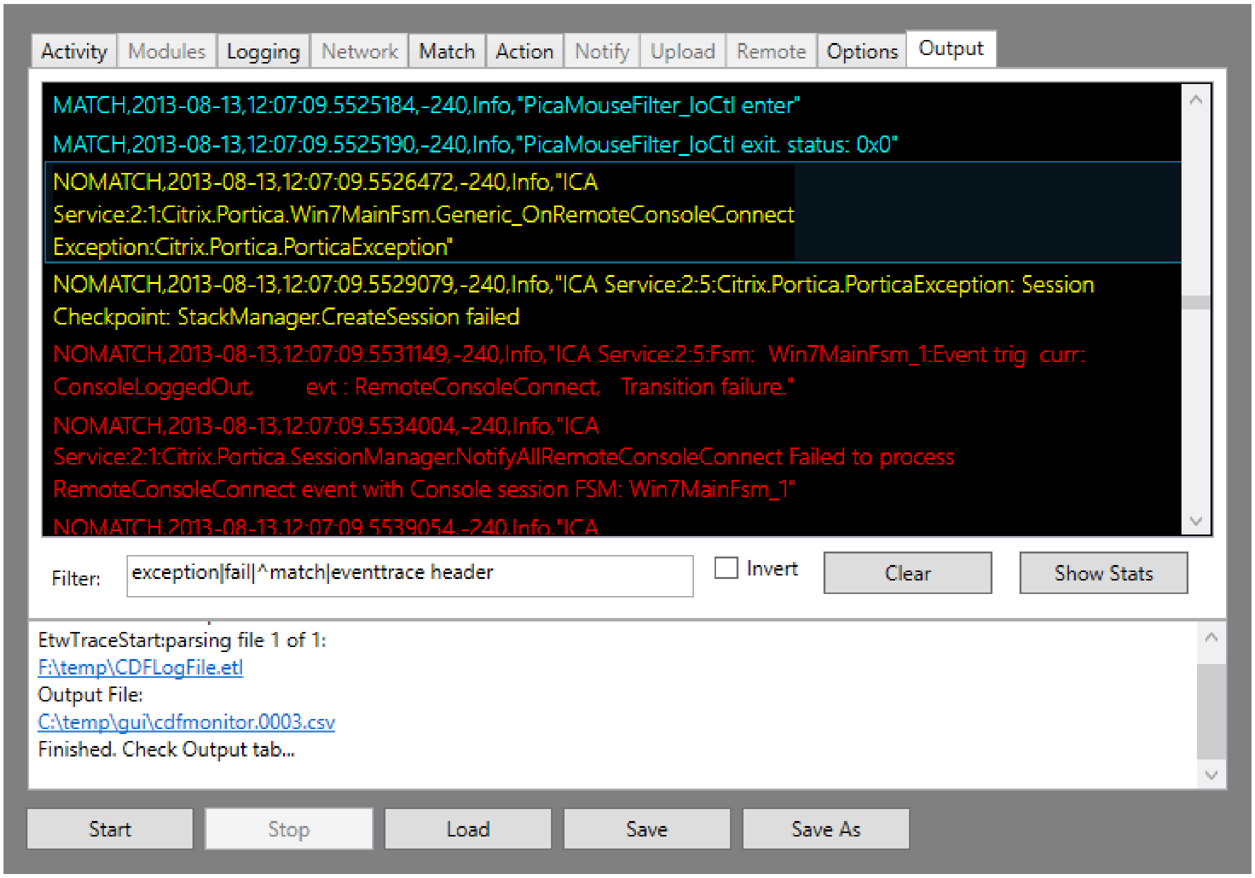
Filter: exception|fail|prepare ☐ Invert

```
Current Utility Log:
C:\temp\guinew\CDFMonitor.0004.log
Current Trace Log:
C:\temp\guinew\CDFMonitor.0568.csv
Finished. Check Output tab...
```

Syntax Highlighting

In the Output console there are multiple highlighting colors for different items. These cannot be disabled or configured.

- Cyan - Console item contains regex match from Match tab.
- Gray - Debug statement from CDFMonitor.
- Green - Action being processed or Action results.
- Magenta - Kernel trace messages and start of trace.
- Red - string contains fail or error.
- Yellow - string contains warning or exception.
- White - default color



```
MATCH,2013-08-13,12:07:09.5525184,-240,Info,"PicaMouseFilter_IoCtl enter"
MATCH,2013-08-13,12:07:09.5525190,-240,Info,"PicaMouseFilter_IoCtl exit. status: 0x0"
NOMATCH,2013-08-13,12:07:09.5526472,-240,Info,"ICA
Service:2:1:Citrix.Portica.Win7MainFsm.Generic_OnRemoteConsoleConnect
Exception:Citrix.Portica.PorticaException"
NOMATCH,2013-08-13,12:07:09.5529079,-240,Info,"ICA Service:2:5:Citrix.Portica.PorticaException: Session
Checkpoint: StackManager.CreateSession failed
NOMATCH,2013-08-13,12:07:09.5531149,-240,Info,"ICA Service:2:5:Fsm:  Win7MainFsm_1:Event trig  curr:
ConsoleLoggedOut,        evt : RemoteConsoleConnect,    Transition failure."
NOMATCH,2013-08-13,12:07:09.5534004,-240,Info,"ICA
Service:2:1:Citrix.Portica.SessionManager.NotifyAllRemoteConsoleConnect Failed to process
RemoteConsoleConnect event with Console session FSM: Win7MainFsm_1"
NOMATCH,2013-08-13,12:07:09.5539054,-240,Info,"ICA
```

Filter: exception|fail|^match|eventtrace header ☐ Invert

```
EtwTraceStart:parsing file 1 of 1:
F:\temp\CDFLogFile.etl
Output File:
C:\temp\gui\cdfmonitor.0003.csv
Finished. Check Output tab...
```

Show Stats / Activity Summary

Show Stats will display the current counts being maintained. This output is the same that will also be seen when an activity is finished. Output will be displayed in the Status window. If an activity is taking longer than expected for example parsing a trace, this view can help determine if there is an issue.

- **TMF server missed events -** number of unique TMF file requests that were not resolved from configured TMF servers.
- **TMF server hit events** - number of unique TMF file requests that were resolved from configured TMF servers.
- **TMF cache missed events** - number of unique TMF file requests that were not resolved from cache directory.
- **TMF cache hit events** - number of unique TMF file requests that were resolved from cache directory
- **TMF parse errors -** number of TMF file parse errors one per bad trace. (Missing ID in Existing TMF)
- **Error|warning|fail|exception events -** number of trace messages containing either error, warning, fail, or exception in message.
- **Processed events** - number of trace messages processed.
- **Matched events** - number of matched trace messages.
- **Consumer read events -** number of events read from call back from ETW but not processed.
- **Throttled events** - number of matched events that were matched but no action taken due to configured throttle.
- **Missed cdfmonitor matched events -** number of events dropped internally once read. Should be 0.
- **Missed logger events -** number of events dropped internally by the logger queue. Should be 0.
- **Logger jobs** - different output jobs for CdfMonitor log messages and CDF/ETW trace messages.
  - **Etw** - number of messages sent from CdfMonitor as a trace provider into .etl during realtime tracing to etl. Currently this is only marker events and running process lists.
  - **GUI** - number of messages sent to 'Output' console.
  - **Logfile** - number of messages sent to CdfMonitor log.
  - **Tracefile** - number of trace messages sent to Trace log file.
  - **Udp** - number of messages sent over network using Udp protocol.
- **Missed controller events -** number of missed events as documented by ETW Controller when writing the .etl file.
- **Markers** - number of CdfMonitor markers in trace.
- **Parser queue length -** current queue length of raw messages waiting to be formatted and parsed.
- **Buffers read** - number of buffers that have been read according to ETW Controller.
- **Buffers total** - total number of buffers.
- **Trace start time -** time from .etl file of when trace capture was started.
- **Trace stop time -** time from .etl file of when trace capture was stopped.
- **Cdfmonitor activity start time -** start time of current CdfMonitor activity.
- **Cdfmonitor activity stop time -** stop time of current CdfMonitor activity.
- **Cdfmonitor duration -** length of time for current CdfMonitor activity.
- **Cdfmonitor processor time -** total processor time used by CdfMonitor.
- **Cdfmonitor average processor utilization -** average processor utilization used by CdfMonitor.
- **Cdfmonitor current processor utilization -** total processor utilization used by CdfMonitor.
- **Cdfmonitor traces per second** - average number of traces being processed per second by CdfMonitor.

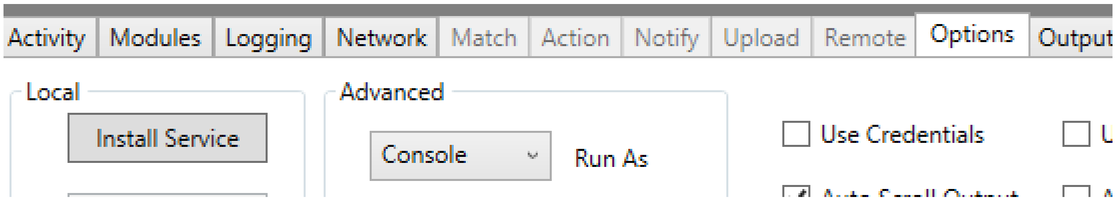
Example console window with CDFMonitor Optimized for parsing.

```
9/1/2013 7:46:41 AM:-------------------------------------------------------
10 TMF server missed events
0 TMF server hit events
10 TMF cache missed events
0 TMF cache hit events
1904 TMF parse errors
-------------------------------------------------------
1 error|warning|fail|exception events
1905 processed events
1904 matched events
1905 consumer read events
1903 throttled events
0 missed cdfmonitor match events
0 missed logger events
logger jobs:
        etw
                0 missed events
                1 total events
                0 currently queued events
        GUI
                0 missed events
                3868 total events
                0 currently queued events
        logFile
                0 missed events
                1926 total events
                0 currently queued events
        traceFile
                0 missed events
                1905 total events
                0 currently queued events
0 missed controller events
0 markers
0 parser queue length
10 buffers read
```

```
10 buffers total
7/11/2013 11:35:09 AM trace start time
7/11/2013 11:37:23 AM trace stop time
--------------------------------------------------------
9/1/2013 7:46:39 AM cdfmonitor activity start time
9/1/2013 7:46:41 AM cdfmonitor activity stop time
00:00:01.3621617 cdfmonitor duration
00:00:04.6605298 cdfmonitor processor time
342.14 cdfmonitor average processor utilization %
2.47 cdfmonitor current processor utilization %
1398.51 cdfmonitor traces per second
--------------------------------------------------------
```

COMMAND LINE CONSOLE USAGE

CDFMonitor can be used in console only mode similar to previous releases from a command prompt window. To configure for command line usage, set the following in GUI as shown below or by setting 'RunAs' in the configuration file to 'Console'. After configuration file RunAs is set to Console, whatever activity type configured will be started when cdfmonitor starts. To enable GUI, edit configuration file and set 'RunAs' to 'GUI'   <add key="RunAs" value="GUI" />



Console ⌄ Run As

□ Use Credentials

The syntax of passing arguments however has changed in addition to some of the parameter names. Using the command line instead of the GUI can be useful in environments where only console access is available or when scripting setup or deployment. All parameters available in the GUI are also configurable on the command line.

See Walkthrough: Using console / command line mode for additional examples of syntax and usage.

The current syntax used for passing parameters on the command line is:

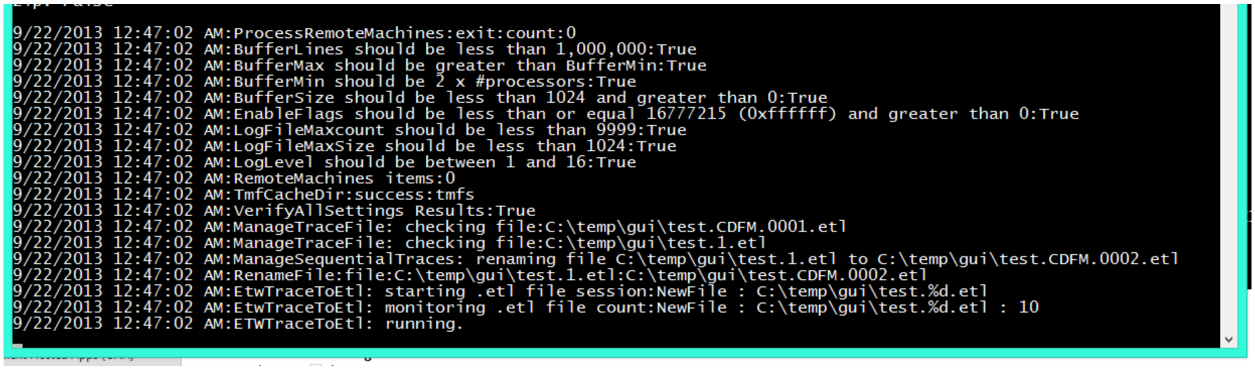
- CDFMonitor.exe /%parameter%:%value%
    - Example: **cdfmonitor.exe /configfile:c:\temp\cdfmonitor.exe.config**
- **NOTE:** In prior versions of CDFMonitor the syntax was slightly different. Prior versions used syntax of /%parameter%: %value% where there was a space ‘ ‘ between the colon ‘:’ and the value. In the current version there is no space between colon and value.

Passing operators via command line is the same as in previous versions. The syntax is:

- CDFMonitor.exe /%operator%
    - Example: **cdfmonitor.exe /registerfta**
- Command line operators can be displayed by using ‘/?’ on command line as shown in the following section.

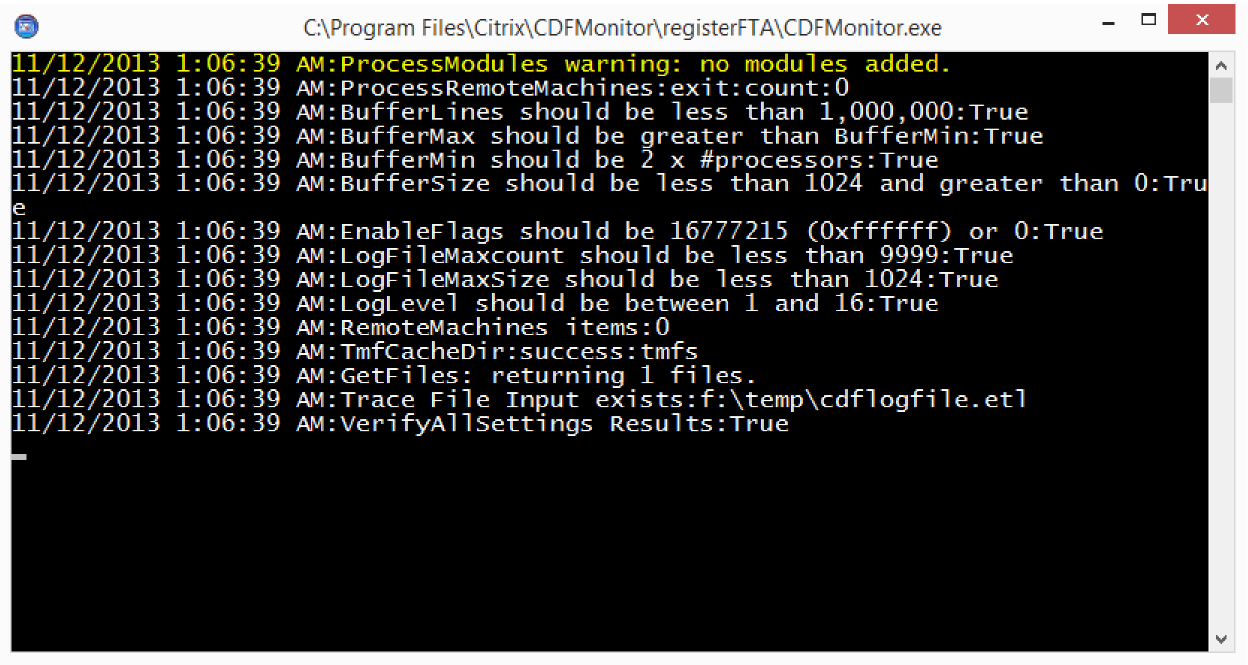
NOTE: Start command prompt with Administrative credentials. See Security for additional information. Failure to do so may unexpectedly close the command prompt or fail to perform the action requested.

Example output from TraceToEtl



```
9/22/2013 12:47:02 AM:ProcessRemoteMachines:exit:count:0
9/22/2013 12:47:02 AM:BufferLines should be less than 1,000,000:True
9/22/2013 12:47:02 AM:BufferMax should be greater than BufferMin:True
9/22/2013 12:47:02 AM:BufferMin should be 2 x #processors:True
9/22/2013 12:47:02 AM:BufferSize should be less than 1024 and greater than 0:True
9/22/2013 12:47:02 AM:EnableFlags should be less than or equal 16777215 (0xffffff) and greater than 0:True
9/22/2013 12:47:02 AM:LogFileMaxcount should be less than 9999:True
9/22/2013 12:47:02 AM:LogFileMaxSize should be less than 1024:True
9/22/2013 12:47:02 AM:LogLevel should be between 1 and 16:True
9/22/2013 12:47:02 AM:RemoteMachines items:0
9/22/2013 12:47:02 AM:TmfCacheDir:success:tmfs
9/22/2013 12:47:02 AM:VerifyAllSettings Results:True
9/22/2013 12:47:02 AM:ManageTraceFile: checking file:C:\temp\gui\test.CDFM.0001.etl
9/22/2013 12:47:02 AM:ManageTraceFile: checking file:C:\temp\gui\test.1.etl
9/22/2013 12:47:02 AM:ManageSequentialTraces: renaming file C:\temp\gui\test.1.etl to C:\temp\gui\test.CDFM.0002.etl
9/22/2013 12:47:02 AM:RenameFile:file:C:\temp\gui\test.1.etl:C:\temp\gui\test.CDFM.0002.etl
9/22/2013 12:47:02 AM:EtwTraceToEtl: starting .etl file session:NewFile : C:\temp\gui\test.%d.etl
9/22/2013 12:47:02 AM:EtwTraceToEtl: monitoring .etl file count:NewFile : C:\temp\gui\test.%d.etl : 10
9/22/2013 12:47:02 AM:ETWTraceToEtl: running.
```

Example output from RegexParseToCsv



C:\Program Files\Citrix\CDFMonitor\registerFTA\CDFMonitor.exe

```
11/12/2013 1:06:39 AM:ProcessModules warning: no modules added.
11/12/2013 1:06:39 AM:ProcessRemoteMachines:exit:count:0
11/12/2013 1:06:39 AM:BufferLines should be less than 1,000,000:True
11/12/2013 1:06:39 AM:BufferMax should be greater than BufferMin:True
11/12/2013 1:06:39 AM:BufferMin should be 2 x #processors:True
11/12/2013 1:06:39 AM:BufferSize should be less than 1024 and greater than 0:Tru
e
11/12/2013 1:06:39 AM:EnableFlags should be 16777215 (0xffffff) or 0:True
11/12/2013 1:06:39 AM:LogFileMaxcount should be less than 9999:True
11/12/2013 1:06:39 AM:LogFileMaxSize should be less than 1024:True
11/12/2013 1:06:39 AM:LogLevel should be between 1 and 16:True
11/12/2013 1:06:39 AM:RemoteMachines items:0
11/12/2013 1:06:39 AM:TmfCacheDir:success:tmfs
11/12/2013 1:06:39 AM:GetFiles: returning 1 files.
11/12/2013 1:06:39 AM:Trace File Input exists:f:\temp\cdflogfile.etl
11/12/2013 1:06:39 AM:VerifyAllSettings Results:True
```

Command line display help '/?' for operators

When running CDFMonitor via command line, passing '/?' will show the following help for operators. Most of these operators are available in the GUI but not all. See section Operators for additional information and definitions.

```
10/24/2013 9:30:16 AM:This utility manages CDF (ETW) tracing for Citrix components.
10/24/2013 9:30:16 AM:All properties are in the 'cdfMonitor.exe.config' file.
10/24/2013 9:30:16 AM:All properties can also be passed via commandline using format /%property%:%value%
10/24/2013 9:30:16 AM:     Example: cdfmonitor.exe /configfile:c:\temp\new.config
10/24/2013 9:30:16 AM:Optional operators:
10/24/2013 9:30:16 AM:  '/check' checks cdfmonitor service state and status of a remote machine.
10/24/2013 9:30:16 AM:  '/clean' to cleanup ETW sessions and files if utility terminated unexpectedly.
10/24/2013 9:30:16 AM:  '/configfile:%configFile%' used to specify alternate utility configuration file to read from.
10/24/2013 9:30:16 AM:  '/deploy' deploys cdfmonitor service to a remote machine.
10/24/2013 9:30:16 AM:  '/downloadconfigs' to download configuration files from server
10/24/2013 9:30:16 AM:  '/downloadTMFs' to download TMF files from server
10/24/2013 9:30:16 AM:  '/enummodules' to enumerate GUIdString list from registry
10/24/2013 9:30:16 AM:  '/gather' gathers cdfmonitor service traces from a remote machine.
10/24/2013 9:30:16 AM:  '/installservice' installs cdfmonitor as a service on local machine.
10/24/2013 9:30:16 AM:  '/modify' modifies cdfmonitor service state on a remote machine.
10/24/2013 9:30:16 AM:  '/registerfta' registers .etl file type association with this utility.
10/24/2013 9:30:16 AM:  '/resetconfig' resets config file values to default.
10/24/2013 9:30:16 AM:  '/startservice' starts cdfmonitor service on local machine.
10/24/2013 9:30:16 AM:  '/startremote' starts cdfmonitor service on remote machine.
10/24/2013 9:30:16 AM:  '/stop' kills all cdfmonitor instances on local machine.
10/24/2013 9:30:16 AM:  '/stopservice' stops cdfmonitor service on local machine.
10/24/2013 9:30:16 AM:  '/stopremote' stops cdfmonitor service on remote machine.
10/24/2013 9:30:16 AM:  '/undeploy' uninstalls cdfmonitor service from remote machine.
10/24/2013 9:30:16 AM:  '/uninstall' uninstalls cdfmonitor service from local machine.
10/24/2013 9:30:16 AM:  '/unregisterfta' unregisters .etl file type association from this utility.
10/24/2013 9:30:16 AM:  '/update' downloads latest version of utility
10/24/2013 9:30:16 AM:  '/upload' uploads URLfiles to URLserver.
10/24/2013 9:30:16 AM:  '/zip' zips log files and URL files into zip file.
10/24/2013 9:30:16 AM:For additional information please search support.citrix.com for 'CDFMonitor' or CTX129537.
10/24/2013 9:30:16 AM:current version:2.0.0.104
10/24/2013 9:30:16 AM:No update
```

OPERATORS

Operators unlike Configuration settings, perform certain tasks. Most of the commonly used operators are available in the GUI. All operators are available from the command line. The table below defines all the operators and whether they are available in the GUI.

See Walkthrough: Using console / command line mode and Command line display help '/?' for operators for additional information.

Operator Definitions

Operator	GUI	Description
Check	Remote Tab	Checks the CDFMonitor service state of the specified remote machines.
CheckService	Options Tab (Indirectly)	Checks the service state of CDFMonitor service on the local machine. This is not directly available in the GUI but when selecting the **Options** tab, this is performed in the background to enable and disable the appropriate **Local** service buttons.
Clean	Options Tab	Cleans up all CDFMonitor ETW sessions, removes all log and trace files and deletes the TMF cache directory if one exists.
Deploy	Remote Tab	Deploys CDFMonitor as a service with specified configuration to the specified remote machines.
DownloadConfigs	N/A	Reserved for future use to download template configurations.
DownloadTMFs	Options Tab	Will download all TMF files located in path specified in **Tmf Servers**. This process can take a considerable amount of time and should not be necessary unless staging the TMF files is necessary for example offline use.
EnumModules	Modules Tab	Enumerates all Citrix CDF modules from the local machines registry.
Gather	Remote Tab	Gathers CDFMonitor service log, trace, and URL files from specified remote machines.
InstallService	Options Tab	Installs CDFMonitor as a service on the local machine.
Kill	N/A	Kills all instances of CDFMonitor process on the local machine.
Modify	Remote Tab	Modifies the CDFMonitor service states on the specified remote machines.
RegisterFta	Options Tab	Registers .etl File Type Association to use CDFMonitor.exe.
ResetConfig	Options Tab	Resets all configuration settings to their default values in the config file.
Start	N/A	Will start only an existing CDFMonitor ETW session if one exists and is stopped.
StartRemote	Remote Tab	Starts the CDFMonitor service on the specified remote machines.
StartService	Options Tab	Starts the CDFMonitor service if installed on the local machine.

Operator	GUI	Description
Stop	N/A	Will stop only an existing CDFMonitor ETW session if one is started. It will not remove the ETW session.
StopRemote	Remote Tab	Stops the CDFMonitor service on the specified remote machines.
StopService	Options Tab	Stops the CDFMonitor service if running on the local machine.
UnDeploy	Remote Tab	Uninstalls the CDFMonitor service from the specified remote machines.
UninstallService	Options Tab	Uninstalls the CDFMonitor service from the local machine.
UnregisterFta	Options Tab	Unregisters the .etl File Type Association from CDFMonitor process.
Update	N/A	Checks for new version of CDFMonitor from Citrix site.
Upload	Upload Tab	Uploads log, trace, and url files from local CDFMonitor instance to specified **URL Site.**
Zip	N/A	Zips log, trace, and url files from local CDFMonitor instance.

CONFIGURATION FILE

The configuration file 'cdfmonitor.exe.config' is critical for utility operation and is required. It contains all the variables that configure CdfMonitor. No information is stored in the registry except optional File Type Association.

By default, on startup, CDFMonitor will attempt to load the default config file named 'cdfmonitor.exe.config'. This can be overridden by using command line argument /configfile:%new config file name%.

This file can be edited manually but almost all configuration is available in the GUI which helps prevent misconfiguration.

On some systems depending on .net configuration, if config file is missing, it will attempt to generate a new one.

NOTE: Configuration files from CdfMonitor 1.x are not compatible with this version.

Config File Definitions

Setting	Type	Default	Location	Description
AdvancedOptions	True/False	False	Activity Tab	If enabled will show additional tabs and options.
Activity	String	Unknown	Activity Tab	Determines activity being performed. RegexParseToCsv, Remote, TraceToEtl, TraceToCsv, Unknown, Server.
AllowSingleInstance	True/False	False	Options Tab	If enabled will only allow one instance of CdfMonitor to perform an Activity at a time.
Annoyance	True/False	True	N/A	If enabled will prompt to save config file if config modified and GUI exiting. Can be disabled in MessageBox using 'Cancel'.
AutoScroll	True/False	True	Options Tab	If enabled will AutoScroll Output console when new messages arrive.
BufferLines	Integer	99999	Options Tab	Number of lines kept in Output console.
BufferMax	Integer	80	Options Tab	Maximum number of allocated buffers for tracing.
BufferMin	Integer	40	Options Tab	Minimum number of allocated buffers for tracing.
BufferSize	Integer	100	Options Tab	Size of buffer in kilobytes.
ConfigFile	String	Cdfmonitor.exe.config	Options Tab	Path and name of CdfMonitor configuration file.
Debug	True/False	False	Logging Tab	If enabled allows Debug messages from CdfMonitor to Output console.
DeployPath	String		Remote Tab	Configures the Deploy Path that contains the configuration being used for remote deployment of CDFMonitor.
DisplayFilter	String		Output Tab	If populated filters output console using given regex pattern.

Setting	Type	Default	Location	Description
EnableFlags	Integer	0	Options Tab	Used by tech support to enable kernel flags for tracing.
EventCommand	String		Action Tab	If populated stores command string to run when a matched event occurs.
EventCommandWait	True/False	False	Action Tab	If enabled, process will wait for results from command and send to logger queue.
EventMaxCount	Integer	0	Match Tab	Maximum number of matches before trace session stops.
EventThrottle	Integer	0	Match Tab	Number of seconds to disable Actions if a subsequent match occurs.
GatherPath	String		Remote Tab	Configures the Gather Path that will be used when gathering traces from remotes machines.
LogBufferOnMatch	True/False	False	Logging Tab	If enabled and 'Log Match Only' is enabled, when a match occurs, the current buffer will be logged including non-matched events.
LogFileAutoFlush	True/False	True	Logging Tab	If enabled will write each trace message to file as it is processed. If disabled messages will be queued and written in chunks for better WAN performance.
LogFileMaxCount	Integer	20	Logging Tab	Number of log and traces files counted separately to maintain before discarding.
LogFileMaxSize	Integer	10	Logging Tab	Max size of each log or trace file.
LogFileName	String	CdfMonitor.log	Logging Tab	If populated will write all CdfMonitor messages to the specified log file.
LogFileOverwrite	True/False	True	Logging Tab	If enabled will overwrite oldest log or trace file when maximum file count is reached.

Setting	Type	Default	Location	Description
LogFileServer	String		Logging Tab	If populated will be used to permanently store all log and trace files generated.
LogLevel	Integer	16	Options Tab	Defines granularity of logging 16 being the highest.
LogMatchDetail	True/False	False	Logging Tab	If enabled will populate additional trace fields for output.
LogMatchOnly	True/False	False	Logging Tab	If enabled will only log matched events.
LogToConsole	True/False	True	Logging Tab	If enabled will send all configured output to console.
ModuleEnableByFilter	True/False	False	Modules Tab	If enabled will use 'Module Filter' to generate module list at runtime from 'Local Machine' registry list.
ModuleFilter	String		Modules Tab	If populated filters module list. Can be used in conjunction with ModuleEnableByFilter for module selection.
ModuleListViewItems	String		Modules Tab	If populated is a semicolon ';' separated list of Modules for 'Current Configuration' view.
ModulePath	String		Modules Tab	If populated will be used for 'Remote Machine' and 'File' configurations.
ModuleSource	String	Configuration	Modules Tab	Current source being used for Module list. Unknown, Configuration, LocalMachine, RemoteMachine, File.
RegexPattern	String		Match Tab	If populated will be used as a regular expression for comparison against processed trace messages for a match.
RemoteActivity	String	Unknown	Remote Tab	Determines the currently selected 'Remote Activity'. Check, Deploy, Gather, UnDeploy, Modify, Start, Stop, Unknown.

Setting	Type	Default	Location	Description
RemoteMachines	String		Remote Tab	List of machines separated by semicolon ';' for Remote Management.
RemoteMachinesPath	String		Remote Tab	If populated will be used to configure file and path location of file containing list of machines to be used for 'Remote Machines'.
RemoteUseMachinesCache	True/False	False	Remote Tab	If enabled will use current session cache of machine states when processing new 'Remote Activity'.
Retries	Integer	1	Options Tab	Configures the number of retries attempted on a failed 'Remote Activity'.
RunAs	String	GUI	Options Tab	Configures how CdfMonitor process runs. Unknown, Console, GUI, Hidden, Service. (Service option is for documentation only. To RunAs Service, use GUI or console commands to install service.
ServiceStartMode	String	Automatic	Remote Tab	Sets the service start mode when deploying or modifying remote CdfMonitor service instance. Automatic, Manual, Disabled.
ShutdownCommand	String		Activity Tab	If populated is a semicolon ';' separated list of commands to run on process/service shutdown.
ShutdowncommandWait	True/False	False	Activity Tab	If enabled will configure CdfMonitor to wait for ShutdownCommand process completion for logging and optional notification of results.
SmtpPassword	String		Notify Tab	If configured will be saved as CLEAR TEXT in config file for use when connecting to SMTP server. Populate SmtpUser and leave SmtpPassword blank for prompt.

Setting	Type	Default	Location	Description
SmtpPort	Integer	25	Notify Tab	TCP port to use when connecting to SMTP server.
SmtpSendFrom	String		Notify Tab	Conforming email address used to identify source of email notification.
SmtpSendTo	String		Notify Tab	Comma separated ',' list of conforming email addresses to send notification to when a Match occurs.
SmtpServer	String		Notify Tab	IP Address or FQDN that If populated will be used to route SMTP notification email.
SmtpSsl	True/False	False	Notify Tab	If enabled will use SSL protocol for SMTP server communication.
SmtpSubject	String	%computername%	Notify Tab	If populated will be prepended to machine name of CdfMonitor instance Subject line of a SMTP email notification message.
SmtpUser	String		Notify Tab	If populated will be used for authentication to SmtpServer.
StartEventEnabled	True/False	False	Options Tab	If enabled manages when start of tracing will occur based off of configured Windows Event Id being logged.
StartEventEnabledImmediately	True/False	False	Options Tab	If enabled will start tracing immediately instead of waiting for first Windows Event Event ID to be logged. This is only functional if StopEventDisabledPermanently is disabled.
StartEventId	Integer	0	Options Tab	Windows Event ID to monitor for to start tracing.
StartEventSource	String	Application	Options Tab	Used to specify which Windows Event Log to monitor for configured Event ID.

Setting	Type	Default	Location	Description
StartupCommand	String		Activity Tab	If populated is a semicolon ';' separated list of commands to run on process/service startup.
StartupCommandWait	True/False	False	Activity Tab	If enabled will configure CdfMonitor to wait for StartupCommand process completion for logging and optional notification of results.
StopEventDisabledPermanently	True/False	False	Options Tab	If enabled will stop tracing permanently when configured Windows Event Log Event ID is logged.
StopEventEnabled	True/False	False	Options Tab	If enabled manages when stop of tracing will occur based off of configured Windows Event Id being logged.
StopEventID	Integer	0	Options Tab	Windows Event ID to monitor for to stop tracing.
StopEventSource	String	Application	Options Tab	Used to specify which Windows Event Log to monitor for configured Event ID.
TMFCacheDir	String	TMFs	Match Tab	If populated will be used to store TMFs that have been downloaded from 'TMF Servers' for future use.
TMFServers	String	http://ctxsym.citrix.com/TMFs/xaxd/	Match Tab	If populated is a semicolon ';' separated list of paths to be searched when downloading a TMF file for .etl file parsing.
TraceFileInput	String		Activity Tab	If populated is used to configure source .etl file and path location of file being parsed.
TraceFileOutput	String		Logging Tab	If populated is used to configure destination file (typically .etl or .csv) when tracing or parsing.
UdpClientEnabled	True/False	False	Network Tab	If enabled will send all outputted messages from CdfMonitor over network using Udp protocol.

Setting	Type	Default	Location	Description
UdpClientPort	Integer	45000	Network Tab	Udp port number used by 'client' (not server) instances of CdfMonitor.
UdpPingEnabled	True/False	False	Network Tab	If enabled will send a 'ping' information packet at configured UdpPingTimer intervals to UdpPingServer.
UdpPingServer	String		Network Tab	If populated is IP or FQDN of CdfMonitor 'server' configured to listen for 'ping' information messages from CdfMonitor 'clients'.
UdpServer	String		Network Tab	If populated is IP or FQDN of CdfMonitor 'server' configured to listen for trace messages from CdfMonitor 'clients'.
UdpServerPort	Integer	45001	Network Tab	Udp port number used by 'server' (not client) instances of CdfMonitor.
URLFiles	String		Upload Tab	If populated is a semicolon ';' separated list of file and path locations of files to be gathered and\or uploaded.
URLPassword	String		Upload Tab	If configured will be saved as CLEAR TEXT in config file for use when connecting to URLServer. Populate URLUser and leave URLPassword blank for prompt.
URLSite	String		Upload Tab	If configured is the destination URL for uploaded package.
URLUser	String		Upload Tab	If populated will be used for authentication to URLServer.
UseCredentials	True/False	False	Options Tab	If enabled will prompt and optionally store alternate credentials for remote activities and remote paths.
UseServiceCredentials	True/False	False	Options Tab	If enabled will use UseCredentials alternate credentials for remote service 'Logon Authentication'.

Setting	Type	Default	Location	Description
UseTargetTime	True/False	True	Logging Tab	If enabled will adjust trace time messages to trace source local time zone.
UseTraceSourceForDestination	True/False	False	Activity Tab	If enabled, overrides TraceFileOutput with TraceFileInput file and path name with '.csv' appended.
Version	String		N/A	Populated with current version of CdfMonitor when config file is saved.
WriteEvent	True/False	False	Match Tab	If enabled will write Match and Activity information to Windows Application Event log.

Default Configuration

The following is the default configuration file included in the zip download from the CTX article. Using **Reset Config** on the Options tab will generate the same values.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!--CdfMonitor:2.0.0.104:JAG-HOME-8-->
<configuration>
  <startup>
    <supportedRuntime version="v4.5" />
    <supportedRuntime version="v4.0.30319" />
    <supportedRuntime version="v2.0.50727" />
  </startup>
  <appSettings>
    <add key="Activity" value="Unknown" />
    <add key="AdvancedOptions" value="False" />
    <add key="AllowSingleInstance" value="False" />
    <add key="Annoyance" value="True" />
    <add key="AutoScroll" value="True" />
    <add key="BufferLines" value="99999" />
    <add key="BufferMax" value="80" />
    <add key="BufferMin" value="40" />
    <add key="BufferSize" value="100" />
    <add key="ConfigFile" value="CDFMonitor.exe.config" />
    <add key="Debug" value="False" />
    <add key="DeployPath" value="" />
    <add key="DisplayFilter" value="" />
    <add key="EnableFlags" value="0" />
    <add key="EventCommand" value="" />
    <add key="EventCommandWait" value="False" />
    <add key="EventMaxCount" value="0" />
    <add key="EventThrottle" value="0" />
    <add key="GatherPath" value="" />
    <add key="KernelFlags" value="0" />
    <add key="LogBufferOnMatch" value="False" />
    <add key="LogFileAutoFlush" value="True" />
    <add key="LogFileMaxCount" value="10" />
    <add key="LogFileMaxSize" value="20" />
    <add key="LogFileName" value="CDFMonitor.log" />
```

```
<add key="LogFileOverWrite" value="True" />
<add key="LogFileServer" value="" />
<add key="LogLevel" value="16" />
<add key="LogMatchDetail" value="False" />
<add key="LogMatchOnly" value="False" />
<add key="LogToConsole" value="True" />
<add key="ModuleEnableByFilter" value="False" />
<add key="ModuleFilter" value="" />
<add key="ModuleListViewItems" value="" />
<add key="ModulePath" value="" />
<add key="ModuleSource" value="Configuration" />
<add key="RegexPattern" value="" />
<add key="RemoteActivity" value="Unknown" />
<add key="RemoteMachines" value="" />
<add key="RemoteMachinesPath" value="" />
<add key="RemoteUseMachinesCache" value="False" />
<add key="Retries" value="1" />
<add key="RunAs" value="GUI" />
<add key="ServiceStartMode" value="Automatic" />
<add key="ShutdownCommand" value="" />
<add key="ShutdownCommandWait" value="False" />
<add key="SmtpPassword" value="" />
<add key="SmtpPort" value="25" />
<add key="SmtpSendFrom" value="" />
<add key="SmtpSendTo" value="" />
<add key="SmtpServer" value="" />
<add key="SmtpSsl" value="False" />
<add key="SmtpSubject" value="" />
<add key="SmtpUser" value="" />
<add key="StartEventEnabled" value="False" />
<add key="StartEventEnabledImmediately" value="False" />
<add key="StartEventID" value="0" />
<add key="StartEventSource" value="Application" />
<add key="StartupCommand" value="" />
<add key="StartupCommandWait" value="False" />
<add key="StopEventDisabledPermanently" value="False" />
<add key="StopEventEnabled" value="False" />
```

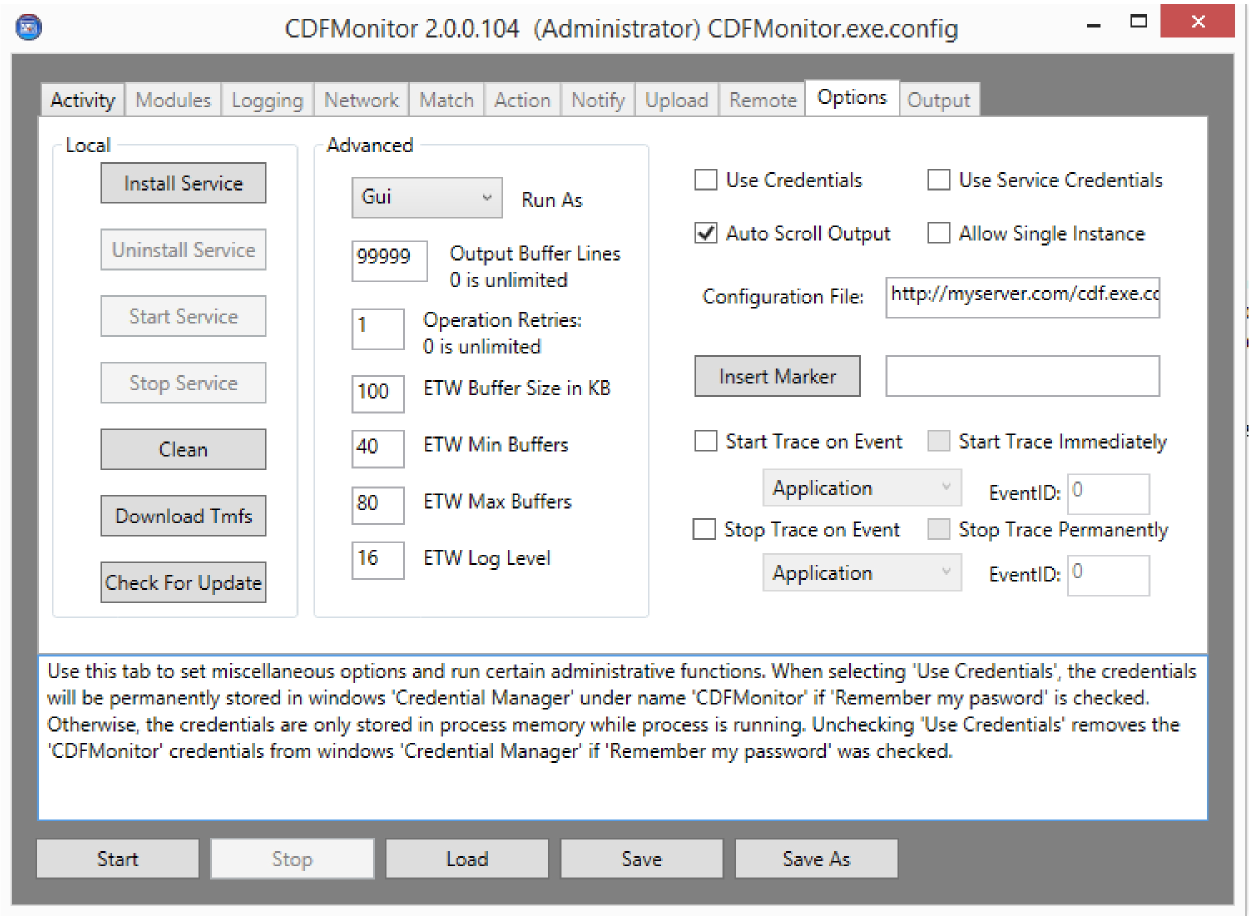
```
    <add key="StopEventID" value="0" />
    <add key="StopEventSource" value="Application" />
    <add key="TMFCacheDir" value="TMFs" />
    <add key="TMFServers" value="http://ctxsym.citrix.com/TMFs/xaxd/" />
    <add key="TraceFileInput" value="" />
    <add key="TraceFileOutput" value="CDFMonitor.etl" />
    <add key="UdpClientEnabled" value="False" />
    <add key="UdpClientPort" value="45000" />
    <add key="UdpPingEnabled" value="False" />
    <add key="UdpPingServer" value="" />
    <add key="UdpPingTimer" value="60" />
    <add key="UdpServer" value="" />
    <add key="UdpServerPort" value="45001" />
    <add key="URLFiles" value="" />
    <add key="URLPassword" value="" />
    <add key="URLSite" value="" />
    <add key="URLUser" value="" />
    <add key="UseCredentials" value="False" />
    <add key="UseServiceCredentials" value="False" />
    <add key="UseTargetTime" value="True" />
    <add key="UseTraceSourceForDestination" value="False" />
    <add key="Version" value="" />
    <add key="WriteEvent" value="False" />
  </appSettings>
</configuration>
```

Central Configuration

Central Configuration is an optional feature that allows the use of a centrally managed configuration file for multiple machines. Central Configuration uses **Configuration File** property on the Options tab. This value can be an UNC, URL, or local path.

On process/service startup, 'cdfmonitor.exe.config' file is read. If the value for 'ConfigFile' does not match the current value, the utility will attempt to load from the path specified in 'ConfigFile'. If file exists, it will be copied locally overwriting 'cdfmonitor.exe.config'. New values from central config will be used. This process only occurs during startup. If file is not accessible, the existing config file will be used.

NOTE: Make sure the config file that is centrally located **Configuration File** value points to itself so on next process start the same path will be checked again for a new version.



Gui Run As

99999 Output Buffer Lines
0 is unlimited

1 Operation Retries:
0 is unlimited

100 ETW Buffer Size in KB

40 ETW Min Buffers

80 ETW Max Buffers

16 ETW Log Level

☐ Use Credentials ☐ Use Service Credentials

☑ Auto Scroll Output ☐ Allow Single Instance

Configuration File: http://myserver.com/cdf.exe.co

☐ Start Trace on Event ☐ Start Trace Immediately

Application EventID: 0

☐ Stop Trace on Event ☐ Stop Trace Permanently

Application EventID: 0

Use this tab to set miscellaneous options and run certain administrative functions. When selecting 'Use Credentials', the credentials will be permanently stored in windows 'Credential Manager' under name 'CDFMonitor' if 'Remember my pasword' is checked. Otherwise, the credentials are only stored in process memory while process is running. Unchecking 'Use Credentials' removes the 'CDFMonitor' credentials from windows 'Credential Manager' if 'Remember my password' was checked.

WALKTHROUGHS

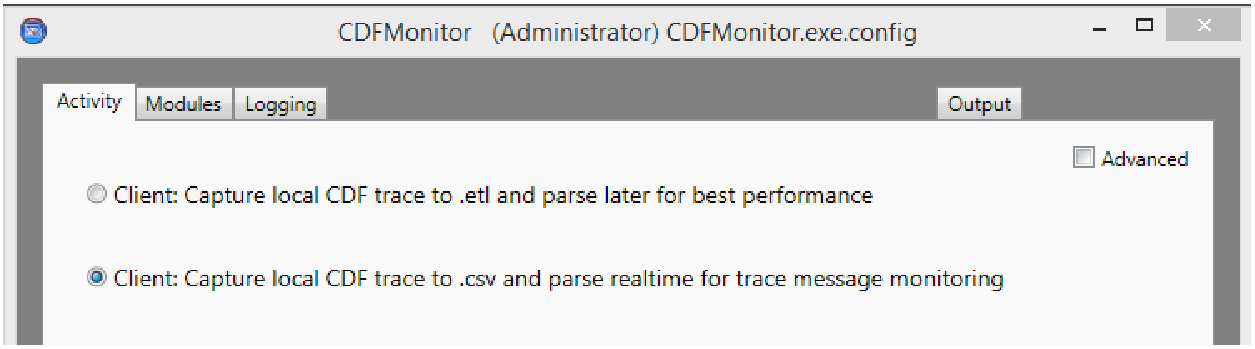
The Walkthrough sections below give short examples of how to use different activities in CDFMonitor.

Walkthrough: Capturing a trace to .csv file

The steps below demonstrate how to capture a trace to a formatted .csv file.

NOTE: Always test performance of configuration before deploying to environment. For example configuration, module selection, resources on the machine, and current activities being performed on that machine can all affect performance.

- Open CDFMonitor
- **Activity Tab**
  - Select: **Client: Capture local CDF trace to .csv and parse real-time for trace message monitoring**

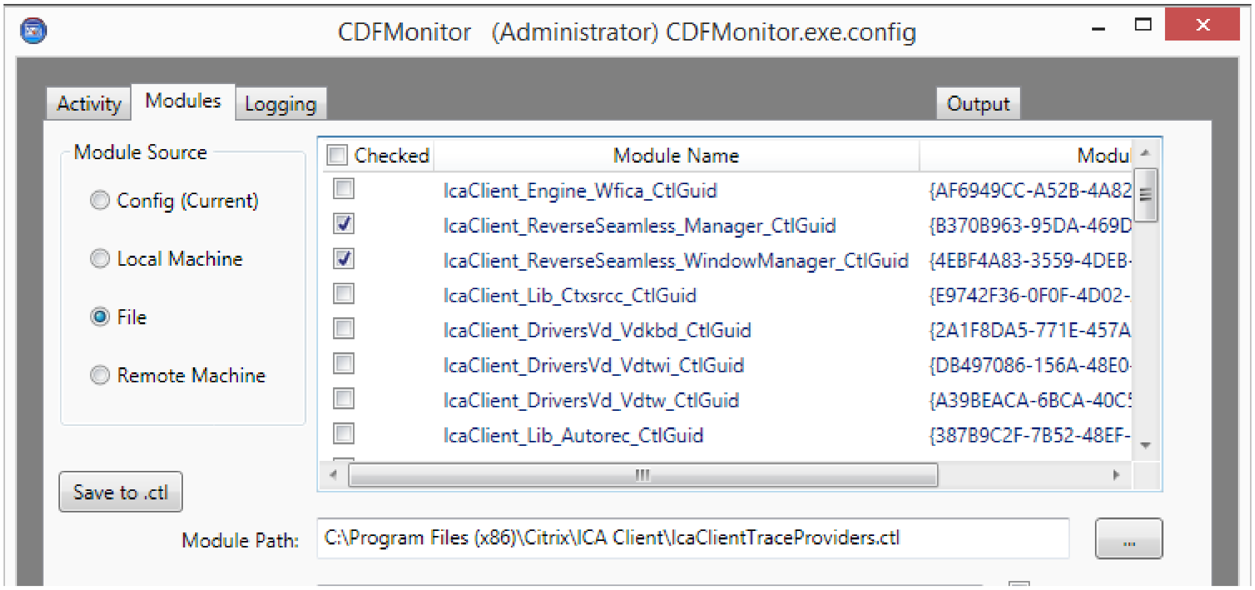


CDFMonitor (Administrator) CDFMonitor.exe.config

○ Client: Capture local CDF trace to .etl and parse later for best performance

● Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

- **Modules Tab**
  - Select: the source location to load the Module GUIds from. This can be from the current configuration file, the local machines registry key, a control .ctl file, or from a remote machines registry key.
  - After selecting the module source, select the modules to be traced.



Module Source

○ Config (Current)

○ Local Machine

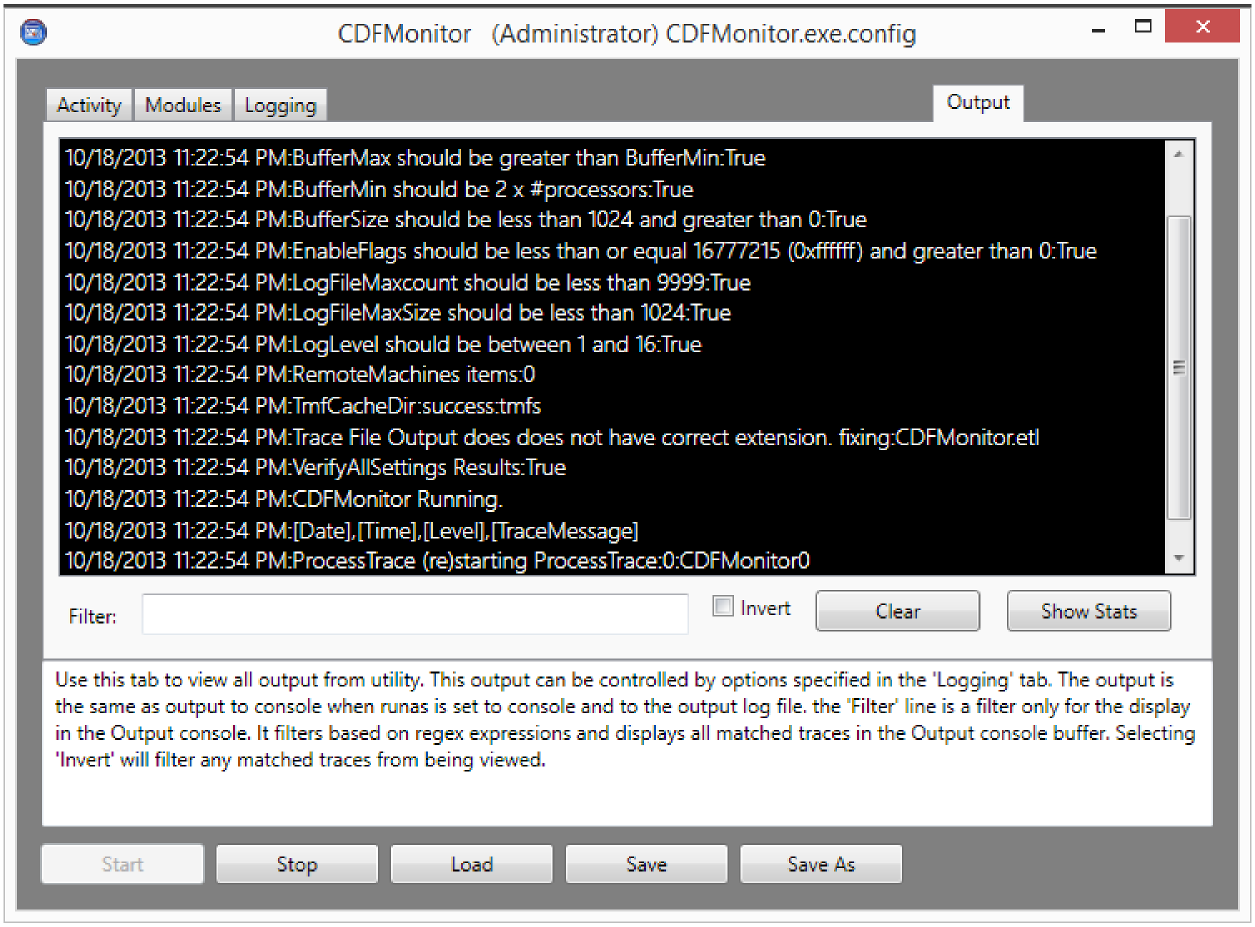
● File

○ Remote Machine

□ Checked	Module Name	Modu
□	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82
☑	IcaClient_ReverseSeamless_Manager_CtlGuid	{B370B963-95DA-469D
☑	IcaClient_ReverseSeamless_WindowManager_CtlGuid	{4EBF4A83-3559-4DEB-
□	IcaClient_Lib_Ctxsrcc_CtlGuid	{E9742F36-0F0F-4D02-
□	IcaClient_DriversVd_Vdkbd_CtlGuid	{2A1F8DA5-771E-457A
□	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-
□	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5
□	IcaClient_Lib_Autorec_CtlGuid	{387B9C2F-7B52-48EF-

Module Path: C:\Program Files (x86)\Citrix\ICA Client\IcaClientTraceProviders.ctl

- **Logging Tab**
  - Use defaults for most use cases.
- **Output Tab**
  - View output and verify trace is enabled.



```
10/18/2013 11:22:54 PM:BufferMax should be greater than BufferMin:True
10/18/2013 11:22:54 PM:BufferMin should be 2 x #processors:True
10/18/2013 11:22:54 PM:BufferSize should be less than 1024 and greater than 0:True
10/18/2013 11:22:54 PM:EnableFlags should be less than or equal 16777215 (0xffffff) and greater than 0:True
10/18/2013 11:22:54 PM:LogFileMaxcount should be less than 9999:True
10/18/2013 11:22:54 PM:LogFileMaxSize should be less than 1024:True
10/18/2013 11:22:54 PM:LogLevel should be between 1 and 16:True
10/18/2013 11:22:54 PM:RemoteMachines items:0
10/18/2013 11:22:54 PM:TmfCacheDir:success:tmfs
10/18/2013 11:22:54 PM:Trace File Output does does not have correct extension. fixing:CDFMonitor.etl
10/18/2013 11:22:54 PM:VerifyAllSettings Results:True
10/18/2013 11:22:54 PM:CDFMonitor Running.
10/18/2013 11:22:54 PM:[Date],[Time],[Level],[TraceMessage]
10/18/2013 11:22:54 PM:ProcessTrace (re)starting ProcessTrace:0:CDFMonitor0
```

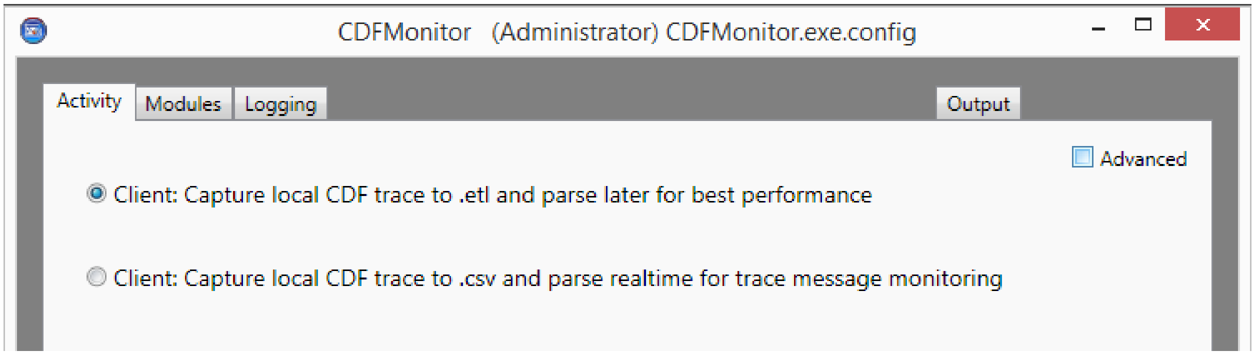
Use this tab to view all output from utility. This output can be controlled by options specified in the 'Logging' tab. The output is the same as output to console when runas is set to console and to the output log file. the 'Filter' line is a filter only for the display in the Output console. It filters based on regex expressions and displays all matched traces in the Output console buffer. Selecting 'Invert' will filter any matched traces from being viewed.

- Select: '**Start**' button to start activity which will enable tracing to a formatted .csv file.

Walkthrough: Capturing a trace to .etl file

The steps below demonstrate how to capture a trace to native .etl file.

- Open CDFMonitor
- **Activity Tab**
  - Select: **Client: Capture local CDF trace to .etl and parse later for best performance**

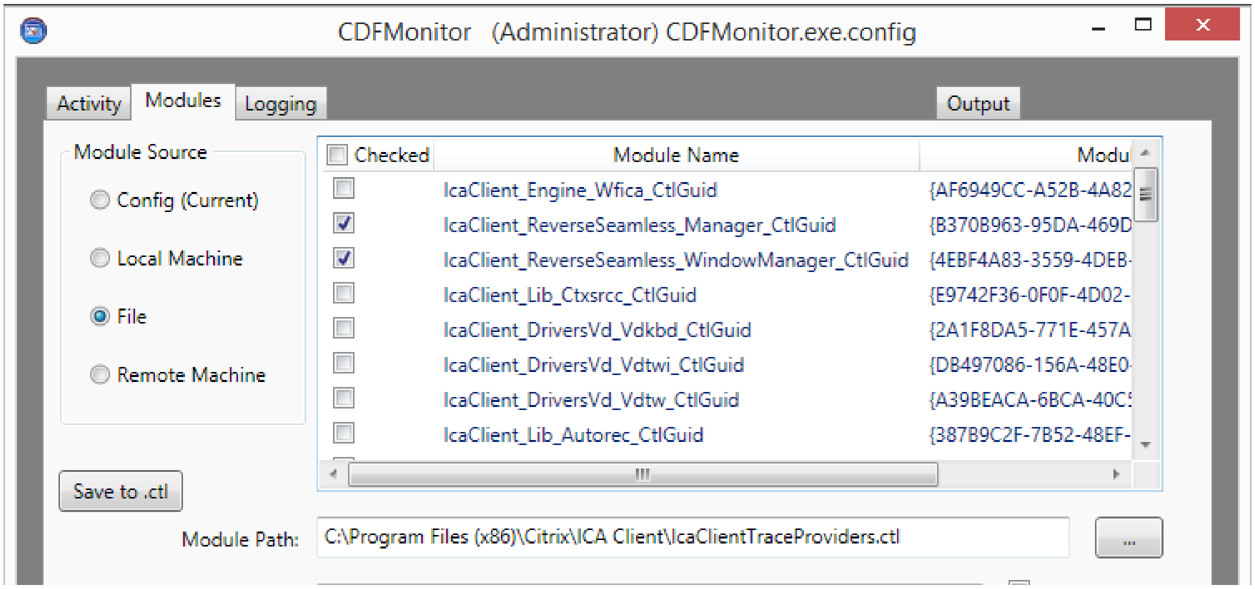


□ Advanced

● Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

- **Modules Tab**
  - Select the source location to load the Module GUIds from. This can be from the current configuration file, the local machines registry key, a control .ctl file, or from a remote machines registry key.
  - After selecting the module source, select the modules to be traced.



Module Source

○ Config (Current)

○ Local Machine

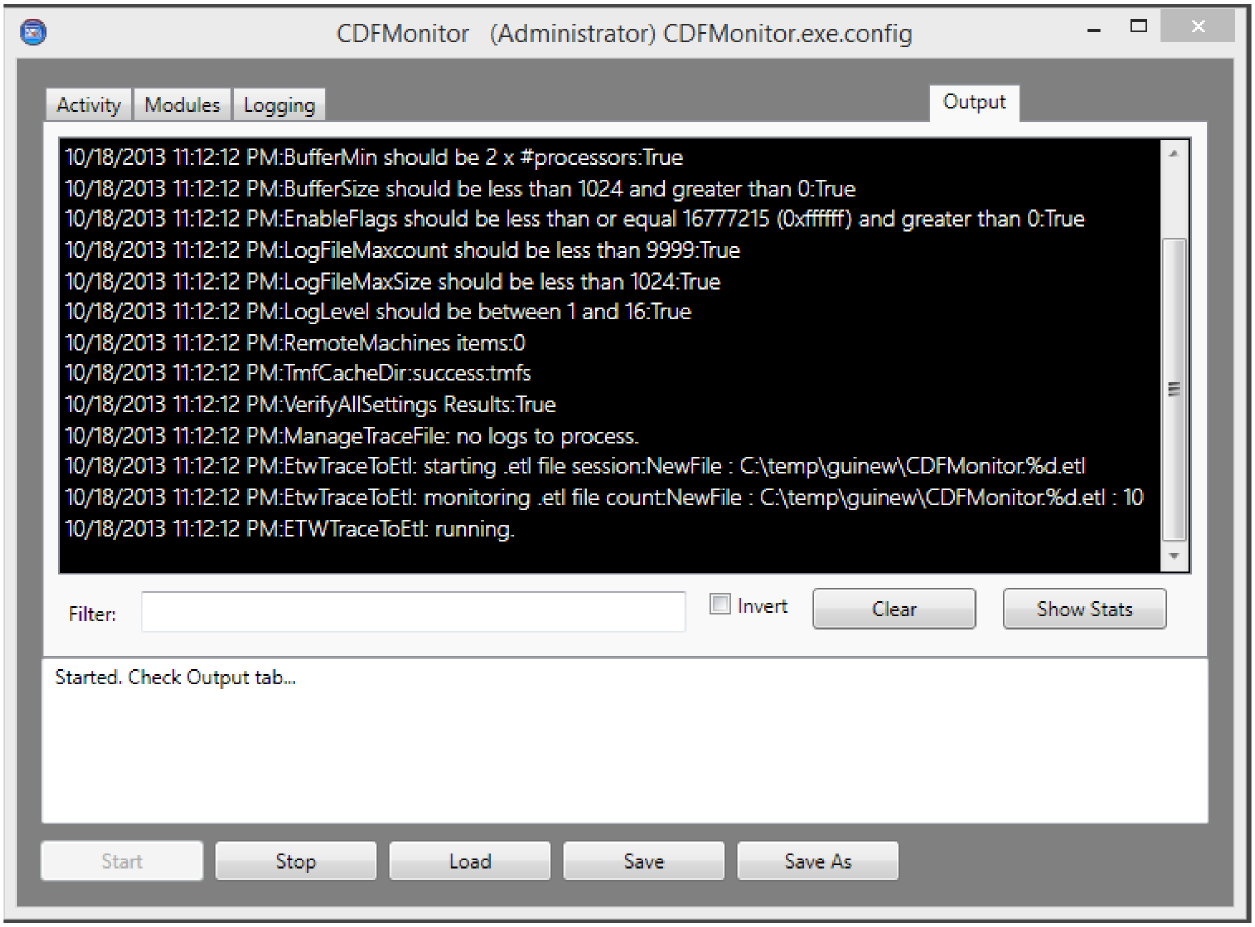
● File

○ Remote Machine

□ Checked	Module Name	Modu
□	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82
☑	IcaClient_ReverseSeamless_Manager_CtlGuid	{B370B963-95DA-469D
☑	IcaClient_ReverseSeamless_WindowManager_CtlGuid	{4EBF4A83-3559-4DEB-
□	IcaClient_Lib_Ctxsrcc_CtlGuid	{E9742F36-0F0F-4D02-
□	IcaClient_DriversVd_Vdkbd_CtlGuid	{2A1F8DA5-771E-457A
□	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-
□	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5
□	IcaClient_Lib_Autorec_CtlGuid	{387B9C2F-7B52-48EF-

Module Path: C:\Program Files (x86)\Citrix\ICA Client\IcaClientTraceProviders.ctl

- **Logging Tab**
  - Use defaults for most use cases.
- **Output Tab**
  - View output and verify trace is enabled.



```
10/18/2013 11:12:12 PM:BufferMin should be 2 x #processors:True
10/18/2013 11:12:12 PM:BufferSize should be less than 1024 and greater than 0:True
10/18/2013 11:12:12 PM:EnableFlags should be less than or equal 16777215 (0xffffff) and greater than 0:True
10/18/2013 11:12:12 PM:LogFileMaxcount should be less than 9999:True
10/18/2013 11:12:12 PM:LogFileMaxSize should be less than 1024:True
10/18/2013 11:12:12 PM:LogLevel should be between 1 and 16:True
10/18/2013 11:12:12 PM:RemoteMachines items:0
10/18/2013 11:12:12 PM:TmfCacheDir:success:tmfs
10/18/2013 11:12:12 PM:VerifyAllSettings Results:True
10/18/2013 11:12:12 PM:ManageTraceFile: no logs to process.
10/18/2013 11:12:12 PM:EtwTraceToEtl: starting .etl file session:NewFile : C:\temp\guinew\CDFMonitor.%d.etl
10/18/2013 11:12:12 PM:EtwTraceToEtl: monitoring .etl file count:NewFile : C:\temp\guinew\CDFMonitor.%d.etl : 10
10/18/2013 11:12:12 PM:ETWTraceToEtl: running.
```

Filter: □ Invert

Started. Check Output tab...

- Select ‘**Start**’ button to start activity which will enable tracing to native .etl file.

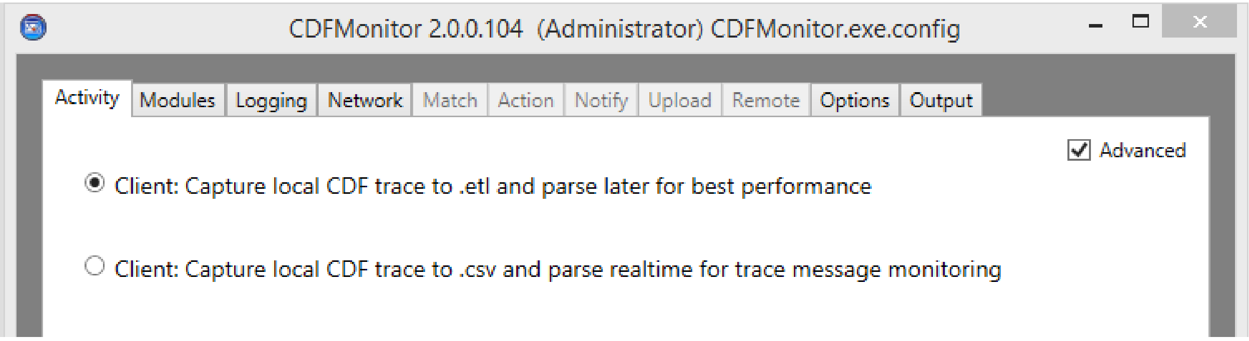
Walkthrough: Using events to capture a trace to .etl file

The steps below demonstrate how to capture a trace to native .etl file based on Windows Events being written to the Windows Event Log.

Using the optional built-in Windows Event listening functionality can be useful in situations when tracing is only needed during a certain function being performed on a machine. An example of this is capturing a trace only during a logon to save space and resources on the machine.

Traces can be started immediately and stopped permanently (one-time use) or be set to start and stop continually based on the Event Log events.

- Open CDFMonitor
- **Activity Tab**
  - Check: **Advanced**
  - Select: Appropriate activity
    - **Client: Capture local CDF trace to .csv and parse real-time for trace message monitoring**
    - **Client: Capture local CDF trace to .etl and parse later for best performance**

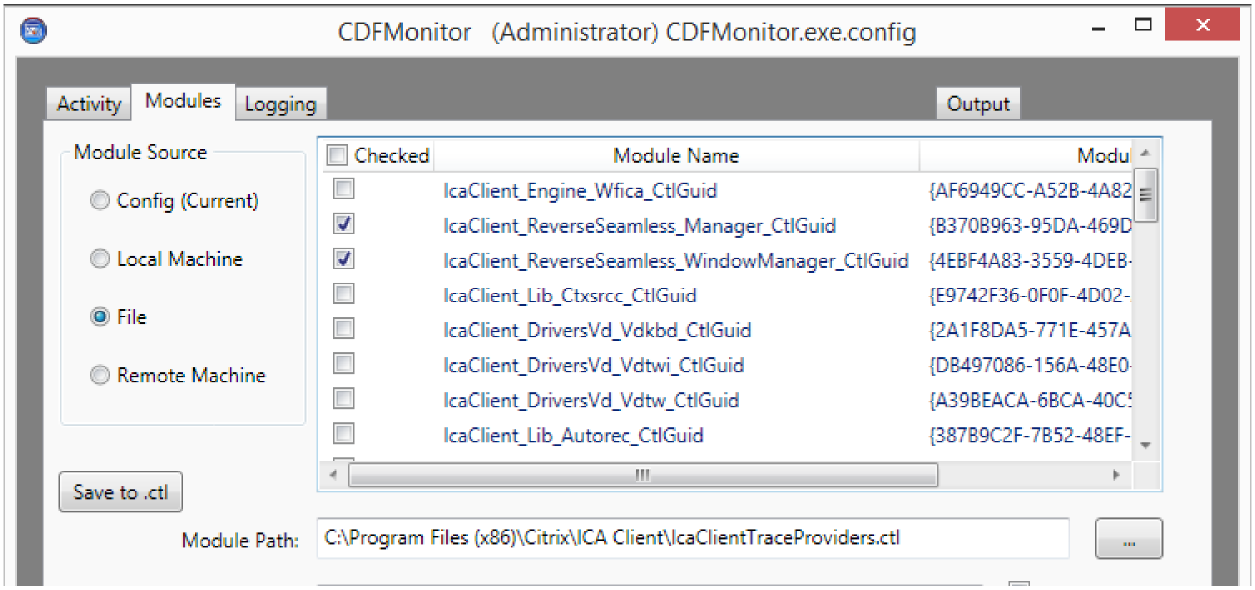


☑ Advanced

● Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

- **Modules Tab**
  - Select the source location to load the Module GUIDs from. This can be from the current configuration file, the local machines registry key, a control .ctl file, or from a remote machines registry key.
  - After selecting the module source, select the modules to be traced.



Module Source

○ Config (Current)

○ Local Machine

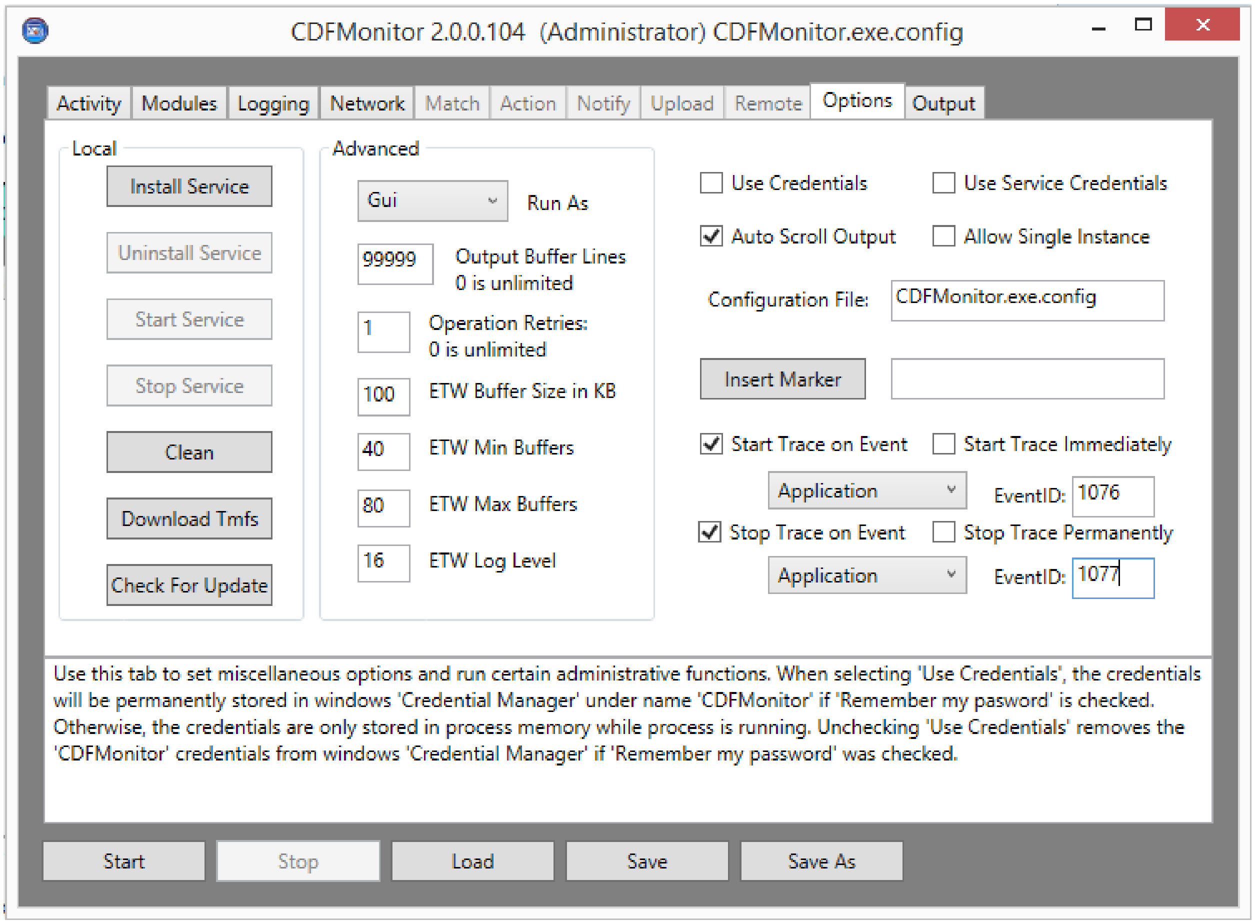
● File

○ Remote Machine

□ Checked	Module Name	Modu
□	IcaClient_Engine_Wfica_CtlGuid	{AF6949CC-A52B-4A82
☑	IcaClient_ReverseSeamless_Manager_CtlGuid	{B370B963-95DA-469D
☑	IcaClient_ReverseSeamless_WindowManager_CtlGuid	{4EBF4A83-3559-4DEB-
□	IcaClient_Lib_Ctxsrcc_CtlGuid	{E9742F36-0F0F-4D02-
□	IcaClient_DriversVd_Vdkbd_CtlGuid	{2A1F8DA5-771E-457A
□	IcaClient_DriversVd_Vdtwi_CtlGuid	{DB497086-156A-48E0-
□	IcaClient_DriversVd_Vdtw_CtlGuid	{A39BEACA-6BCA-40C5
□	IcaClient_Lib_Autorec_CtlGuid	{387B9C2F-7B52-48EF-

Module Path: C:\Program Files (x86)\Citrix\ICA Client\IcaClientTraceProviders.ctl

- **Logging Tab**
  - Use defaults for most use cases.
- **Options Tab**
  - Contains all the Windows Event configuration.
  - Refer to Options Miscellaneous for definitions of the event configuration settings.
  - In this example, when CDFMonitor process/service is started, it will not start tracing until Event ID 1076 in the Application Event Log is written. Tracing will continue until Application Event 1077 is written to the Event Log. This configuration will continue to start / stop tracing until CDFMonitor is stopped.



CDFMonitor 2.0.0.104 (Administrator) CDFMonitor.exe.config

Gui Run As

99999 Output Buffer Lines
0 is unlimited

1 Operation Retries:
0 is unlimited

100 ETW Buffer Size in KB

40 ETW Min Buffers

80 ETW Max Buffers

16 ETW Log Level

□ Use Credentials □ Use Service Credentials

☑ Auto Scroll Output □ Allow Single Instance

Configuration File: CDFMonitor.exe.config

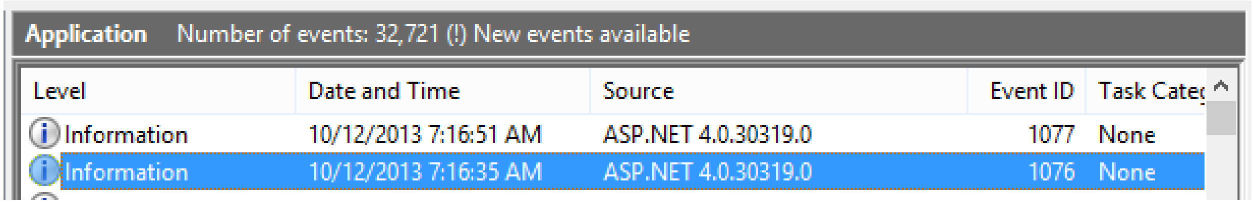
☑ Start Trace on Event □ Start Trace Immediately

Application EventID: 1076

☑ Stop Trace on Event □ Stop Trace Permanently

Application EventID: 1077

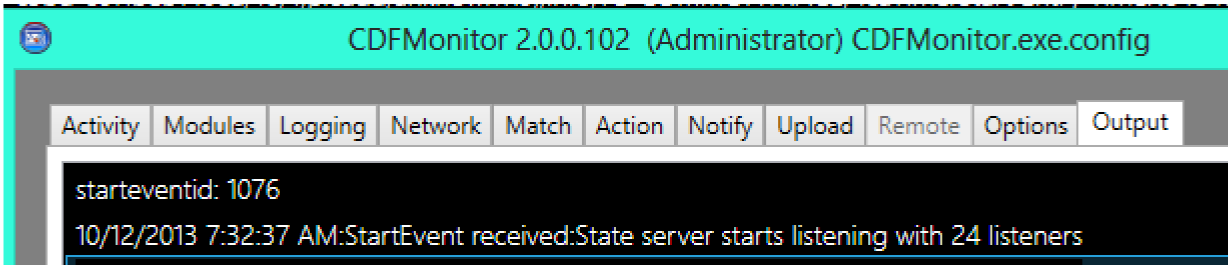
Use this tab to set miscellaneous options and run certain administrative functions. When selecting 'Use Credentials', the credentials will be permanently stored in windows 'Credential Manager' under name 'CDFMonitor' if 'Remember my pasword' is checked. Otherwise, the credentials are only stored in process memory while process is running. Unchecking 'Use Credentials' removes the 'CDFMonitor' credentials from windows 'Credential Manager' if 'Remember my password' was checked.



Application Number of events: 32,721 (!) New events available

Level	Date and Time	Source	Event ID	Task Categ
Information	10/12/2013 7:16:51 AM	ASP.NET 4.0.30319.0	1077	None
Information	10/12/2013 7:16:35 AM	ASP.NET 4.0.30319.0	1076	None

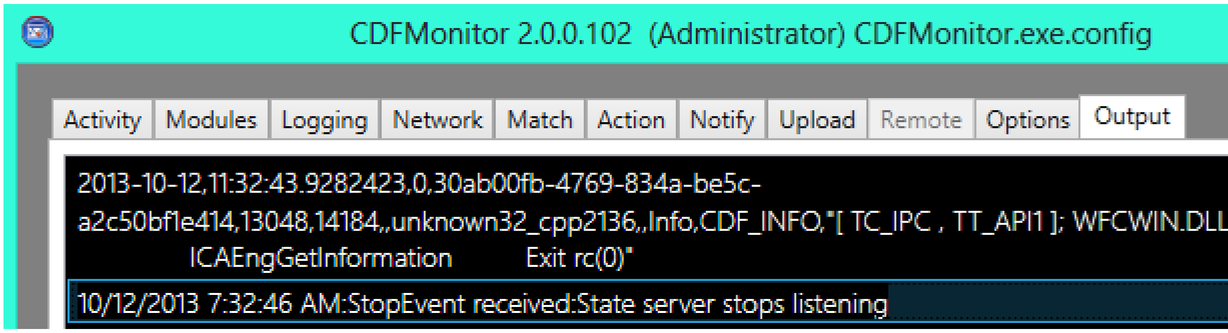
- **Output Tab**
    - When process is started with this configuration, Output will look like this:
        - Last entry will show 'CDFMonitor waiting for startup event'
        - When start event is written, CDFMonitor will write 'StartEvent received:'



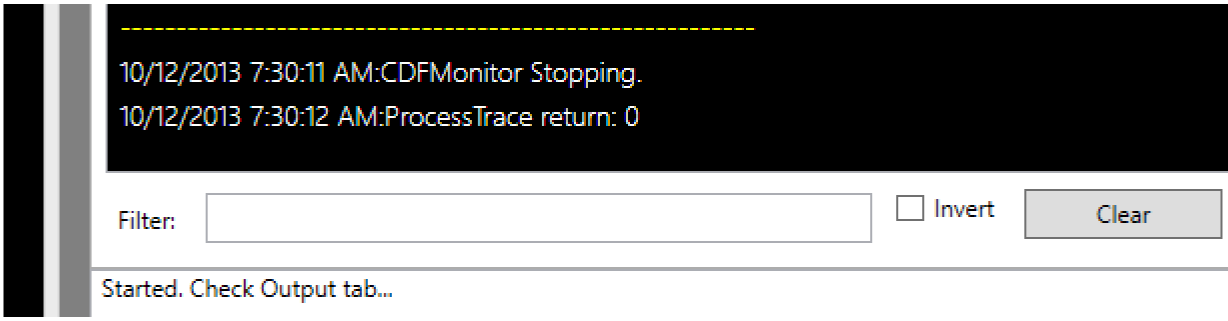
CDFMonitor 2.0.0.102 (Administrator) CDFMonitor.exe.config

```
starteventid: 1076
10/12/2013 7:32:37 AM:StartEvent received:State server starts listening with 24 listeners
```

- When start event is written to Event Log, tracing will start and when stop event is written, tracing will stop.
  - When stop event is written, CDFMonitor will write 'StopEvent received:'.



```
2013-10-12,11:32:43.9282423,0,30ab00fb-4769-834a-be5c-
a2c50bf1e414,13048,14184,,unknown32_cpp2136,,Info,CDF_INFO,"[ TC_IPC , TT_API1 ]; WFCWIN.DLL
          ICAEngGetInformation        Exit rc(0)"
10/12/2013 7:32:46 AM:StopEvent received:State server stops listening
```


```
------------------------------------------------------------
10/12/2013 7:30:11 AM:CDFMonitor Stopping.
10/12/2013 7:30:12 AM:ProcessTrace return: 0
```

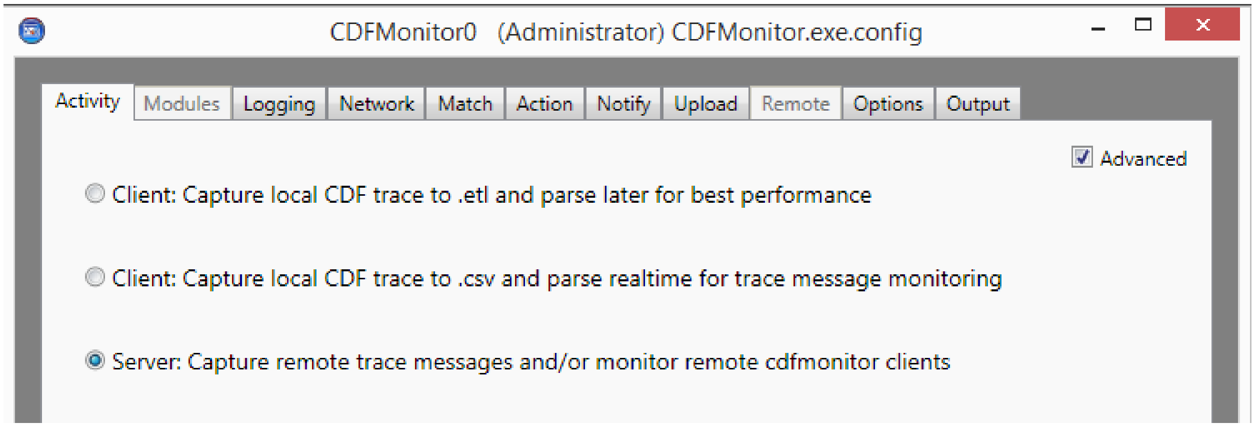
Started. Check Output tab...

Walkthrough: Capture remote trace over network

The steps below demonstrate how to capture a remote trace over the network from other CDFMonitor instances.

NOTE: Always test performance of configuration before deploying to environment. Configuration for example module selection, resources on the machine, and current activities being performed on that machine can all affect performance.

- Open CDFMonitor
- **Activity Tab**
  - Select '**Advanced**' checkbox.
  - Select: **Server: Capture remote trace messages and/or monitor remote cdfmonitor clients**



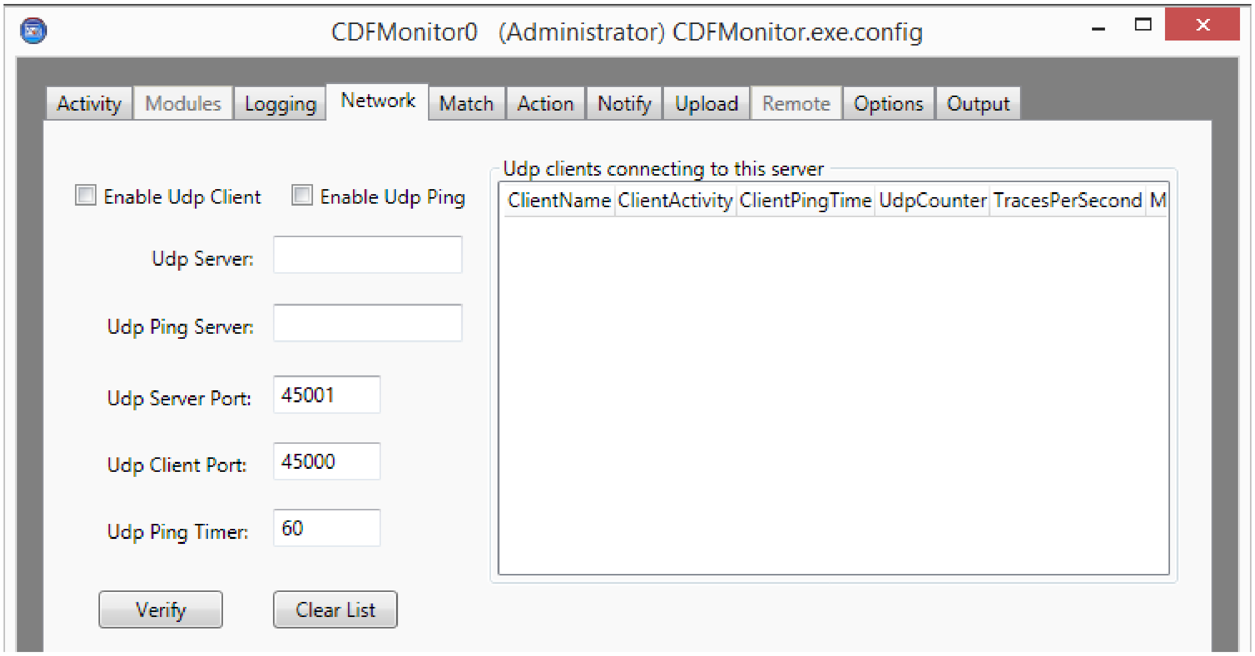
☑ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

● Server: Capture remote trace messages and/or monitor remote cdfmonitor clients

- **Logging Tab**
  - Use defaults for most use cases.
- **Network Tab**
  - **Udp Server** can be set to a specific IP Address, otherwise the utility will listen on all addresses.
  - **Udp Server Port** verify default ok.
  - **Udp Client Port** verify default ok.



□ Enable Udp Client □ Enable Udp Ping

Udp Server:

Udp Ping Server:

Udp Server Port: 45001

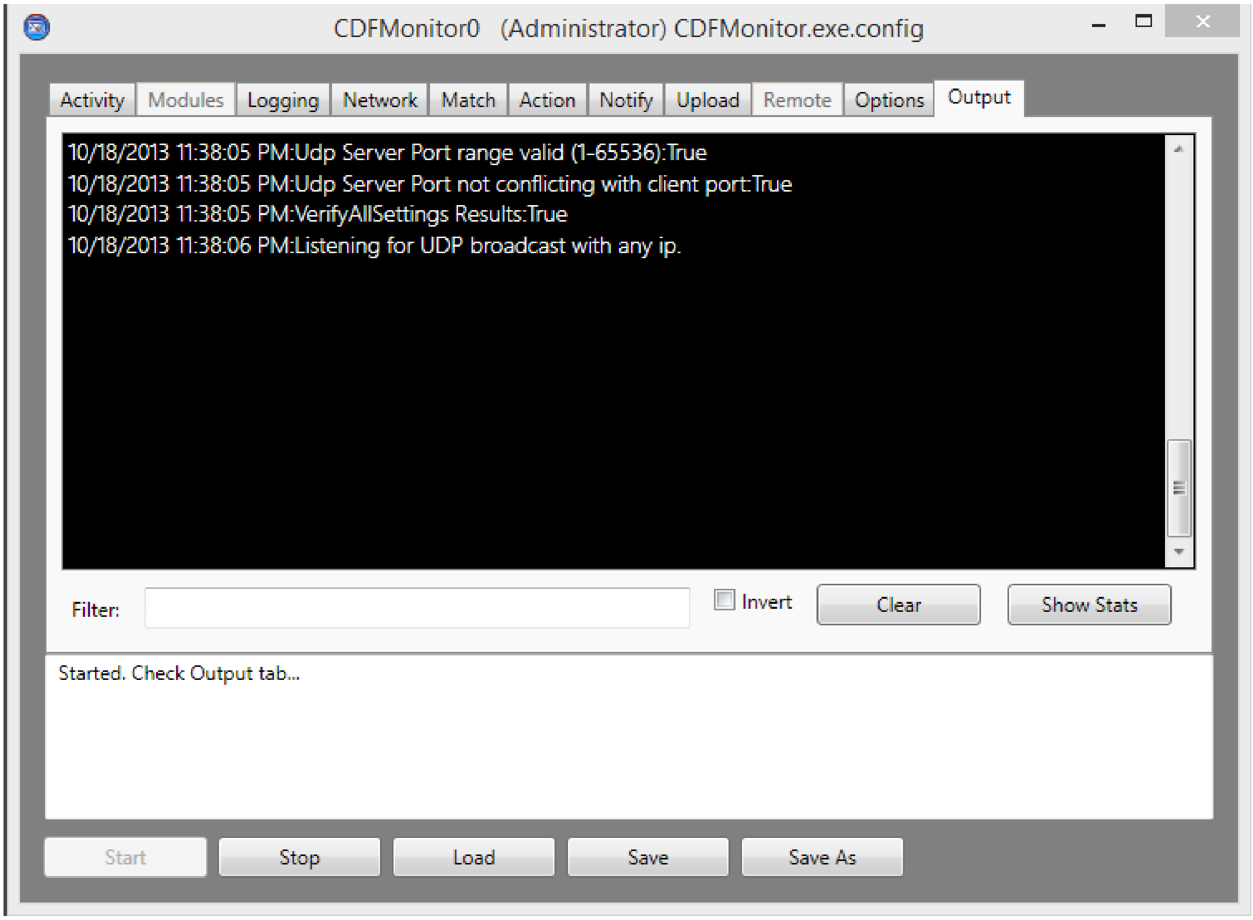
Udp Client Port: 45000

Udp Ping Timer: 60

Udp clients connecting to this server

ClientName	ClientActivity	ClientPingTime	UdpCounter	TracesPerSecond	M

- **Output Tab**
  - View output and verify server is listening.



```
10/18/2013 11:38:05 PM:Udp Server Port range valid (1-65536):True
10/18/2013 11:38:05 PM:Udp Server Port not conflicting with client port:True
10/18/2013 11:38:05 PM:VerifyAllSettings Results:True
10/18/2013 11:38:06 PM:Listening for UDP broadcast with any ip.
```

Started. Check Output tab...

- Select: '**Start**' button to start activity which will enable UDP listener to capture tracing over network to formatted .csv file.

Walkthrough: Formatting a native .etl file into .csv

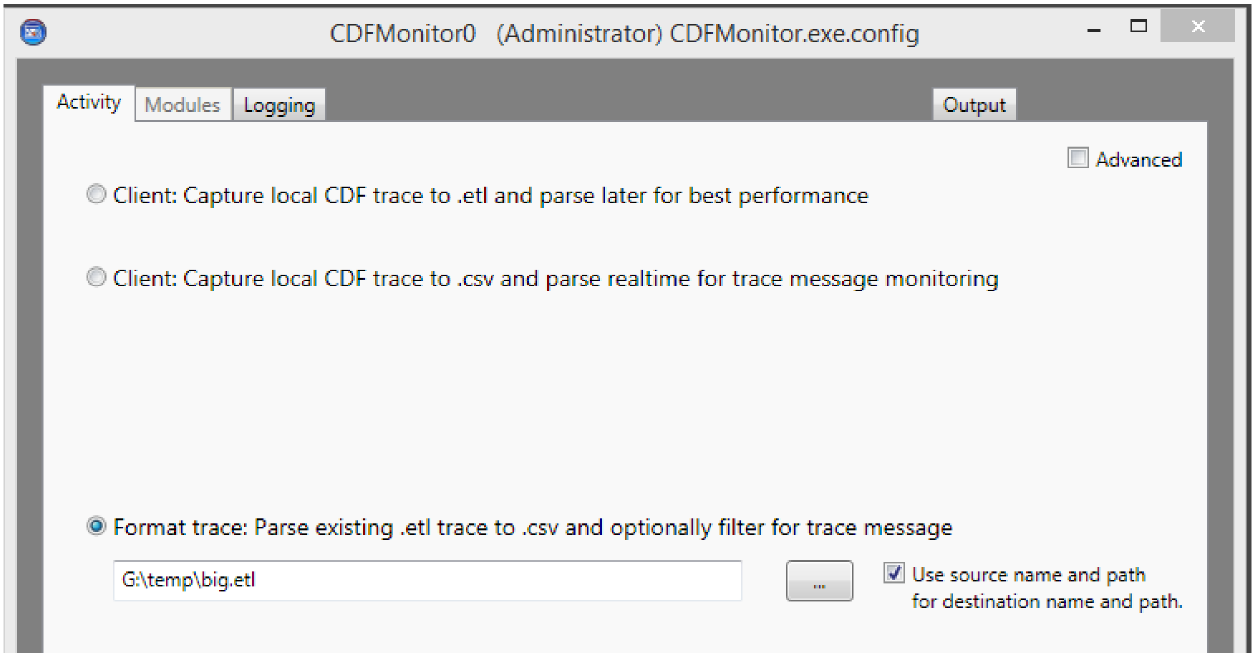
The steps below demonstrate how to format a previously gathered trace file in native .etl format into a comma separated formatted file.

In addition to specifying one file, another option is to pass a wildcard '*' for file. Example: c:\temp\*.etl will process all .etl files in specified directory and subdirectories.

NOTE: Configuring '**TMF Servers**' with a path containing TMFs for the .etl file being parsed is required for unmanaged code. For 'TMF Servers', verify the default value of **http://ctxsym.citrix.com/TMFs/xaxd/** is ok. Setting up a local '**TMF Cache Directory**' will have better performance.

See Register File Type Association (FTA) for additional information.

- Open CDFMonitor
- **Activity Tab**
  - Select: **Format trace: Parse existing .etl trace to .csv and optionally filter for trace message**
  - (OPTIONAL) Check: **Use source name and path for destination name and path**
  - **NOTE:** This option saves the output from .etl parse in the same directory and using the same name as the .etl file with .csv appended.



○ Client: Capture local CDF trace to .etl and parse later for best performance

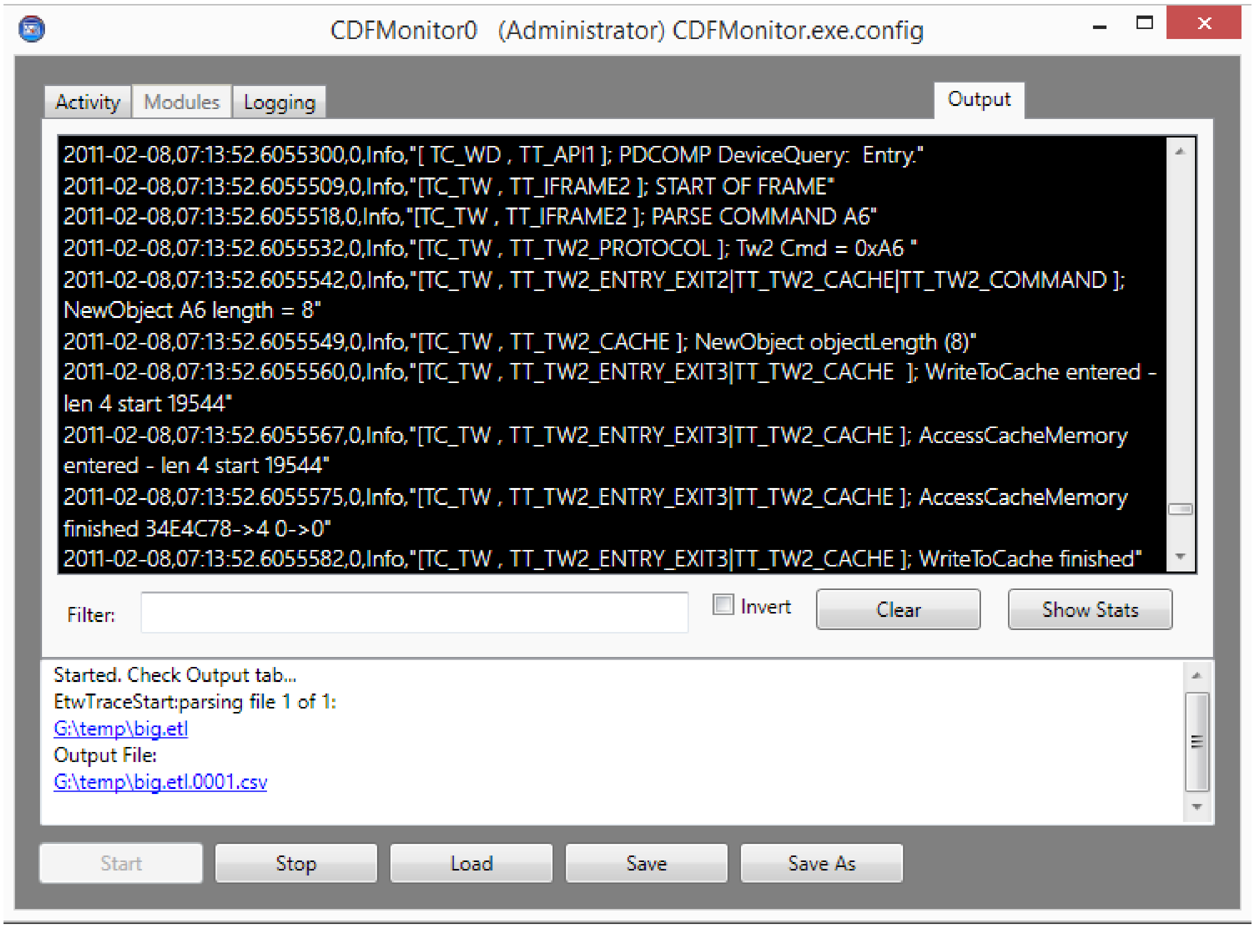
○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

● Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

G:\temp\big.etl

☑ Use source name and path for destination name and path.

- **Logging Tab**
  - Use defaults for most use cases.
- **Output Tab**
  - View output and verify server is listening.



```
2011-02-08,07:13:52.6055300,0,Info,"[ TC_WD , TT_API1 ]; PDCOMP DeviceQuery:  Entry."
2011-02-08,07:13:52.6055509,0,Info,"[TC_TW , TT_IFRAME2 ]; START OF FRAME"
2011-02-08,07:13:52.6055518,0,Info,"[TC_TW , TT_IFRAME2 ]; PARSE COMMAND A6"
2011-02-08,07:13:52.6055532,0,Info,"[TC_TW , TT_TW2_PROTOCOL ]; Tw2 Cmd = 0xA6 "
2011-02-08,07:13:52.6055542,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT2|TT_TW2_CACHE|TT_TW2_COMMAND ];
NewObject A6 length = 8"
2011-02-08,07:13:52.6055549,0,Info,"[TC_TW , TT_TW2_CACHE ]; NewObject objectLength (8)"
2011-02-08,07:13:52.6055560,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE  ]; WriteToCache entered -
len 4 start 19544"
2011-02-08,07:13:52.6055567,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE ]; AccessCacheMemory
entered - len 4 start 19544"
2011-02-08,07:13:52.6055575,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE ]; AccessCacheMemory
finished 34E4C78->4 0->0"
2011-02-08,07:13:52.6055582,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE ]; WriteToCache finished"
```

```
Started. Check Output tab...
EtwTraceStart:parsing file 1 of 1:
G:\temp\big.etl
Output File:
G:\temp\big.etl.0001.csv
```

- Select: ‘**Start**’ button to start activity which will enable parsing of .etl file to formatted .csv file.

Walkthrough: Using console / command line mode

All configuration variables available in the configuration file can also be passed as parameters on the command line.
Passing '/?' on command line will give limited help.

Properties are passed using format '/%property%:%value%'. Example: cdfmonitor.exe /configfile:c:\temp\test.exe.config

Operators are passed using format '/%operator%'. Example: cdfmonitor.exe /clean

NOTE: Start Command Prompt in Administrator mode.

The following assumes a default configuration file to show parameter passing though the parameters below could be set in the configuration file.

Example Commands:

- **Trace To Etl** - >CDFMonitor.exe /runas:console /activity:tracetoetl /modulesource:localmachine /moduleenablebyfilter:true /modulefilter:.* /tracefileoutput:test.etl
- **Trace To Csv** - >CDFMonitor.exe /runas:console /activity:tracetocsv /modulesource:localmachine /moduleenablebyfilter:true /modulefilter:.* /tracefileoutput:test.csv
- **Deploy remote config** (assuming deploy path has remote exe and config)**-** >cdfmonitor.exe /runas:console /activity:remote /remoteactivity:deploy /deploypath:c:\temp\GUI\deploy /remotemachines:127.0.0.1
- **Gather remote config** - >cdfmonitor.exe /runas:console /activity:remote /remoteactivity:gather /gatherpath:c:\temp\GUI\gather /remotemachines:127.0.0.1
- **Undeploy remote config -** >cdfmonitor.exe /runas:console /activity:remote /remoteactivity:undeploy /remotemachines:127.0.0.1
- **Parse etl file -** >CDFMonitor.exe /RunAs:Console /Activity:RegexParseToCsv /TraceFileInput:F:\temp\Cdflogfile.etl /UseTraceSourceForDestination:True

Walkthrough: Formatting multiple native .etl files into .csv

The steps below demonstrate how to format a previously gathered trace files in native .etl format into comma separated formatted files.

Parsing multiple files can be performed by using either *.etl or *.zip wildcards. Using either of those will start a search from the configured directory and all subdirectories.

If a .zip file is specified, a search for all zips will occur first. Each zip that contains a file named '[Content_Types].xml' will be searched for .etl files. Any .etl found in the zip files are extracted. A new search for *.etl files will be performed and will subsequently be parsed.

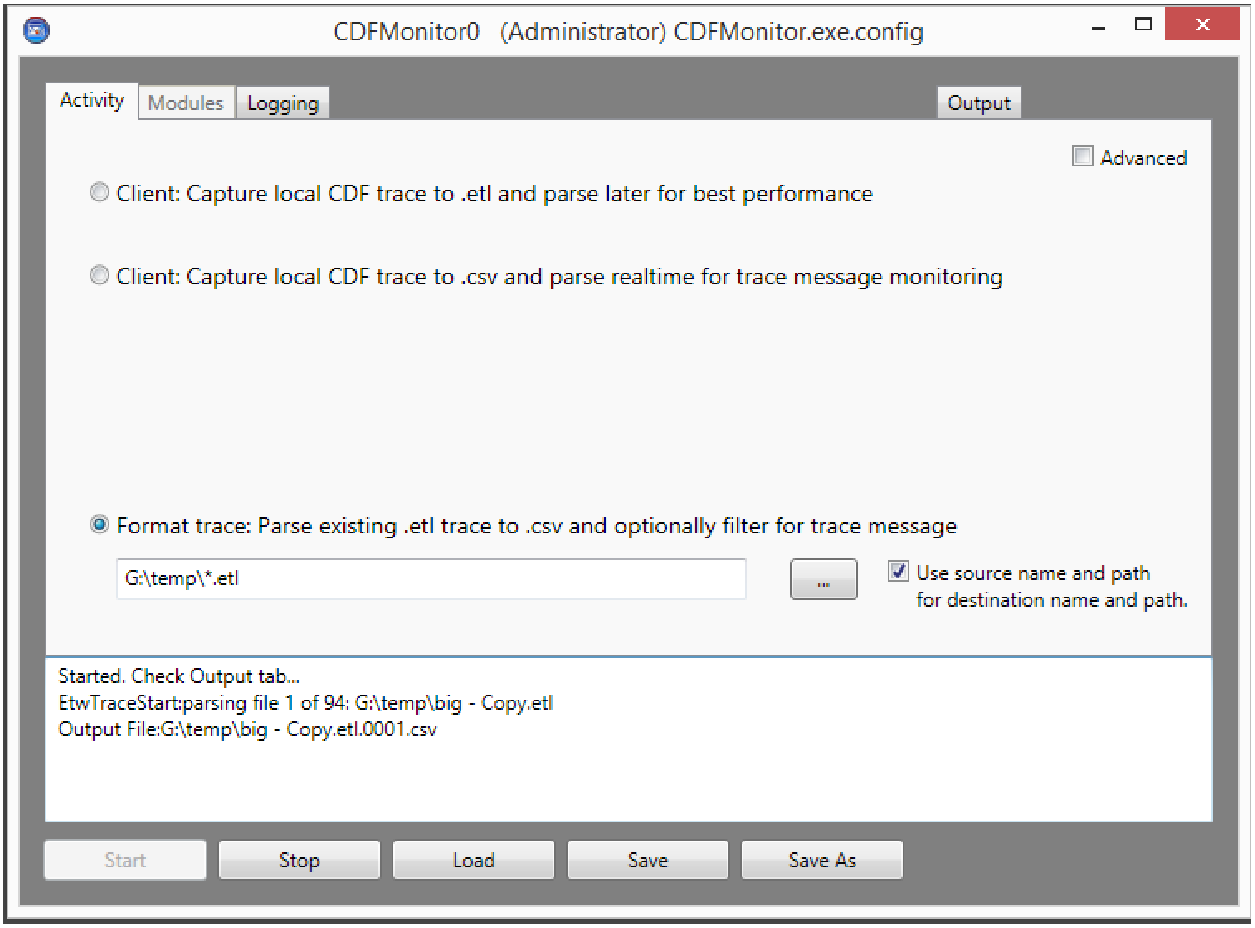
If an .etl file is specified, a search for all etl files in configured directory and all sub directories will be performed. Each .etl file will then be parsed

Microsoft .Net creates packages using Open XML format (XPS) which is what creates and uses '[Content_Types].xml. Any zip created by CDFMonitor can be read by CDFMonitor.

NOTE: Configuring '**TMF Servers**' with a path containing TMFs for the .etl file being parsed is required for unmanaged code. For 'TMF Servers', verify the default value of **http://ctxsym.citrix.com/TMFs/xaxd/** is ok. Setting up a local '**TMF Cache Directory**' will have better performance.

See Register File Type Association (FTA) for additional information.

- Open CDFMonitor
- **Activity Tab**
    - Select: **Format trace: Parse existing .etl trace to .csv and optionally filter for trace message**
    - (OPTIONAL) Check: **Use source name and path for destination name and path**
    - **NOTE:** this option saves the output from .etl parse in the same directory and using the same name as the .etl file with .csv appended.



○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

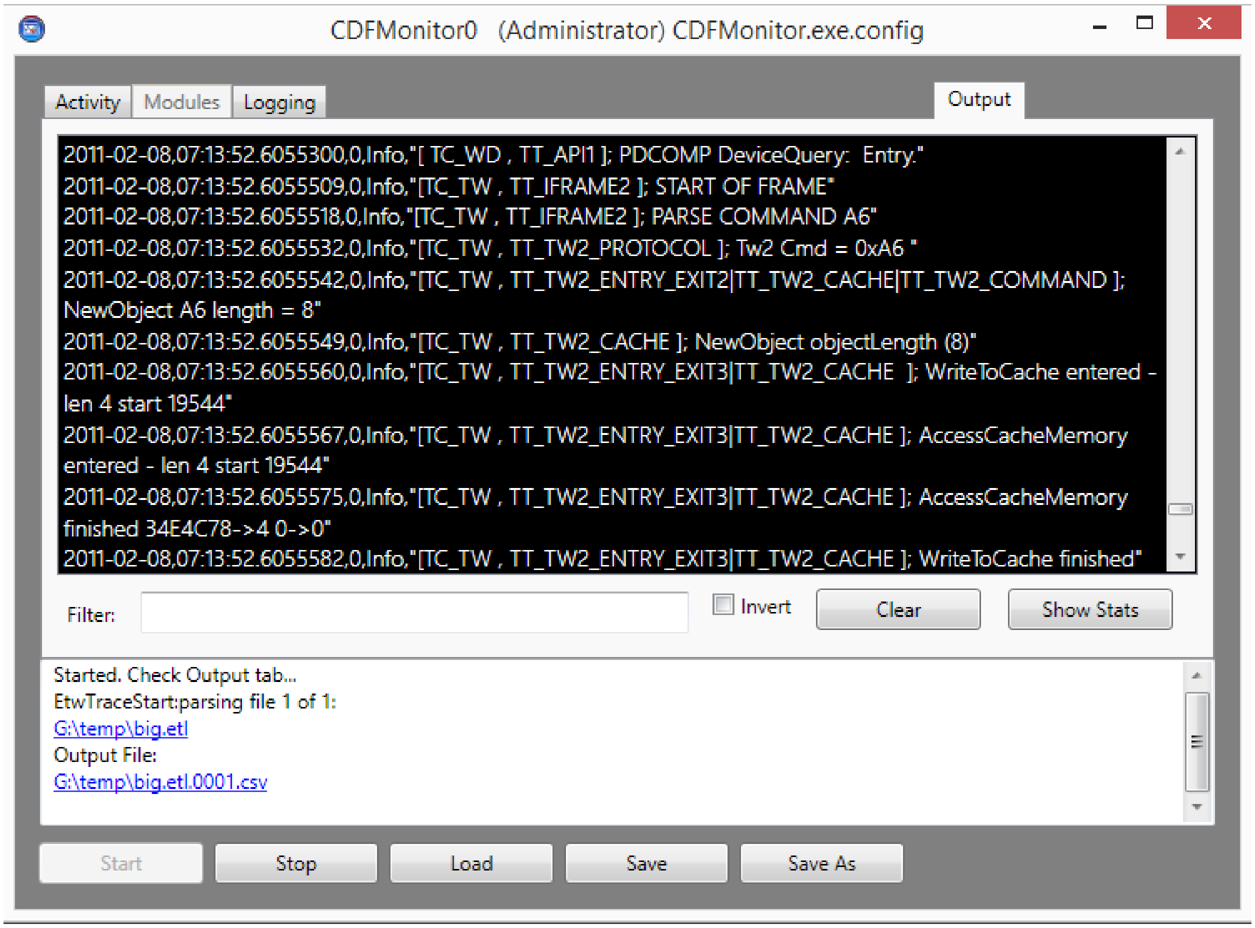
● Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

G:\temp\*.etl

☑ Use source name and path for destination name and path.

```
Started. Check Output tab...
EtwTraceStart:parsing file 1 of 94: G:\temp\big - Copy.etl
Output File:G:\temp\big - Copy.etl.0001.csv
```

- **Logging Tab**
  - Use defaults for most use cases.
- **Output Tab**
  - View output and verify server is listening.



```
2011-02-08,07:13:52.6055300,0,Info,"[ TC_WD , TT_API1 ]; PDCOMP DeviceQuery:  Entry."
2011-02-08,07:13:52.6055509,0,Info,"[TC_TW , TT_IFRAME2 ]; START OF FRAME"
2011-02-08,07:13:52.6055518,0,Info,"[TC_TW , TT_IFRAME2 ]; PARSE COMMAND A6"
2011-02-08,07:13:52.6055532,0,Info,"[TC_TW , TT_TW2_PROTOCOL ]; Tw2 Cmd = 0xA6 "
2011-02-08,07:13:52.6055542,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT2|TT_TW2_CACHE|TT_TW2_COMMAND ];
NewObject A6 length = 8"
2011-02-08,07:13:52.6055549,0,Info,"[TC_TW , TT_TW2_CACHE ]; NewObject objectLength (8)"
2011-02-08,07:13:52.6055560,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE  ]; WriteToCache entered -
len 4 start 19544"
2011-02-08,07:13:52.6055567,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE ]; AccessCacheMemory
entered - len 4 start 19544"
2011-02-08,07:13:52.6055575,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE ]; AccessCacheMemory
finished 34E4C78->4 0->0"
2011-02-08,07:13:52.6055582,0,Info,"[TC_TW , TT_TW2_ENTRY_EXIT3|TT_TW2_CACHE ]; WriteToCache finished"
```

```
Started. Check Output tab...
EtwTraceStart:parsing file 1 of 1:
G:\temp\big.etl
Output File:
G:\temp\big.etl.0001.csv
```

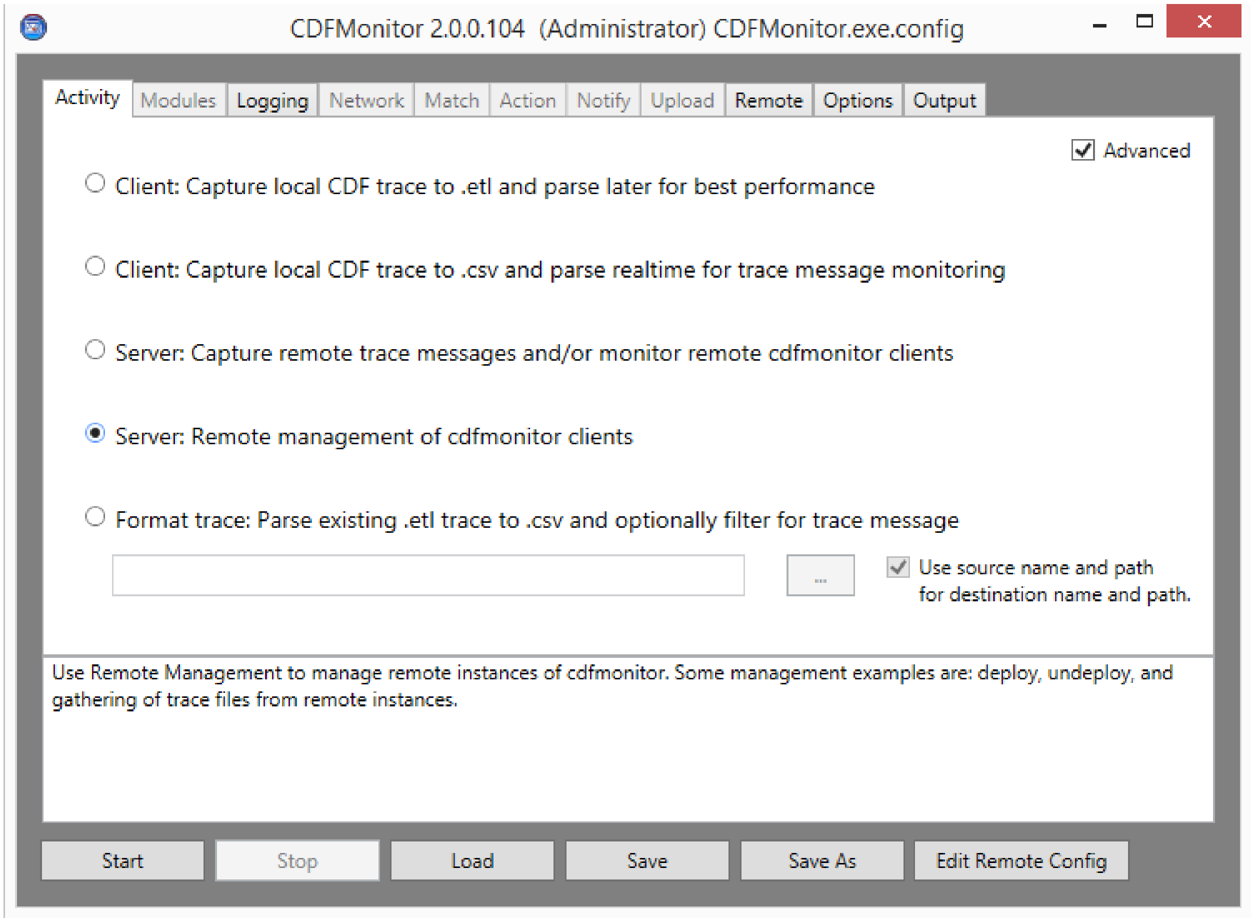
- ○ Select: '**Start**' button to start activity which will enable parsing of all enumerated .etl files to formatted .csv files.

Walkthrough: Creating a remote configuration

The steps below demonstrate how to create a remote configuration file to be used for deployment from the Remote tab.

NOTE: This current gray configuration/window is NOT the configuration being deployed. All the configurations in this window are for the configuration of the currently running CDFMonitor instance remote management activity and not for the remote configuration file.

- Open CDFMonitor
- **Activity Tab**
    - Check: **Advanced**
    - Select: **Server: Remote management of cdfmonitor clients**



CDFMonitor 2.0.0.104 (Administrator) CDFMonitor.exe.config

☑ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

○ Server: Capture remote trace messages and/or monitor remote cdfmonitor clients

● Server: Remote management of cdfmonitor clients

○ Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

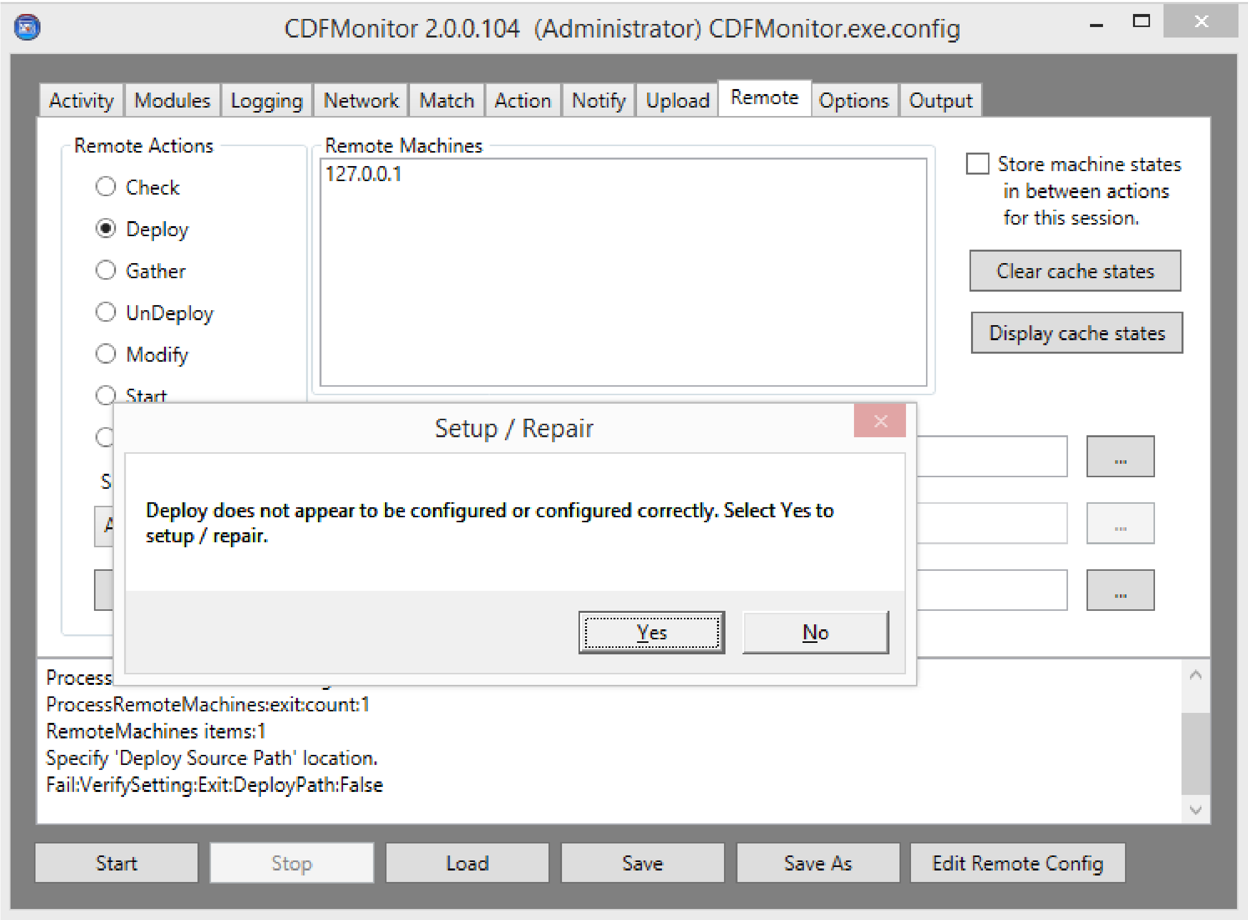
☑ Use source name and path for destination name and path.

Use Remote Management to manage remote instances of cdfmonitor. Some management examples are: deploy, undeploy, and gathering of trace files from remote instances.

- **Logging Tab**
    - Defaults are sufficient for remote management requirements.
- **Options Tab**
    - Use defaults for most use cases.
    - **NOTE:** if Alternate credentials are needed for connecting to remote machines WMI, selecting **Use Credentials** on Options tab will enable the use of alternate credentials for remote connectivity.

  - See Options Section for additional information.

- **Remote Tab**
  - Enter machine(s) IP address, FQDN, or name in the **Remote Machines** list.
    - **NOTE:** Though not required for creating or editing a remote configuration file, having one machine in the **Remote Machines** list as a reference can help automate the module selection as it will be passed into the remote config Modules tab **Module Path.**
  - Select: **Deploy**
  - Select: **Setup / Verify** or **Edit Remote Config** (same result)

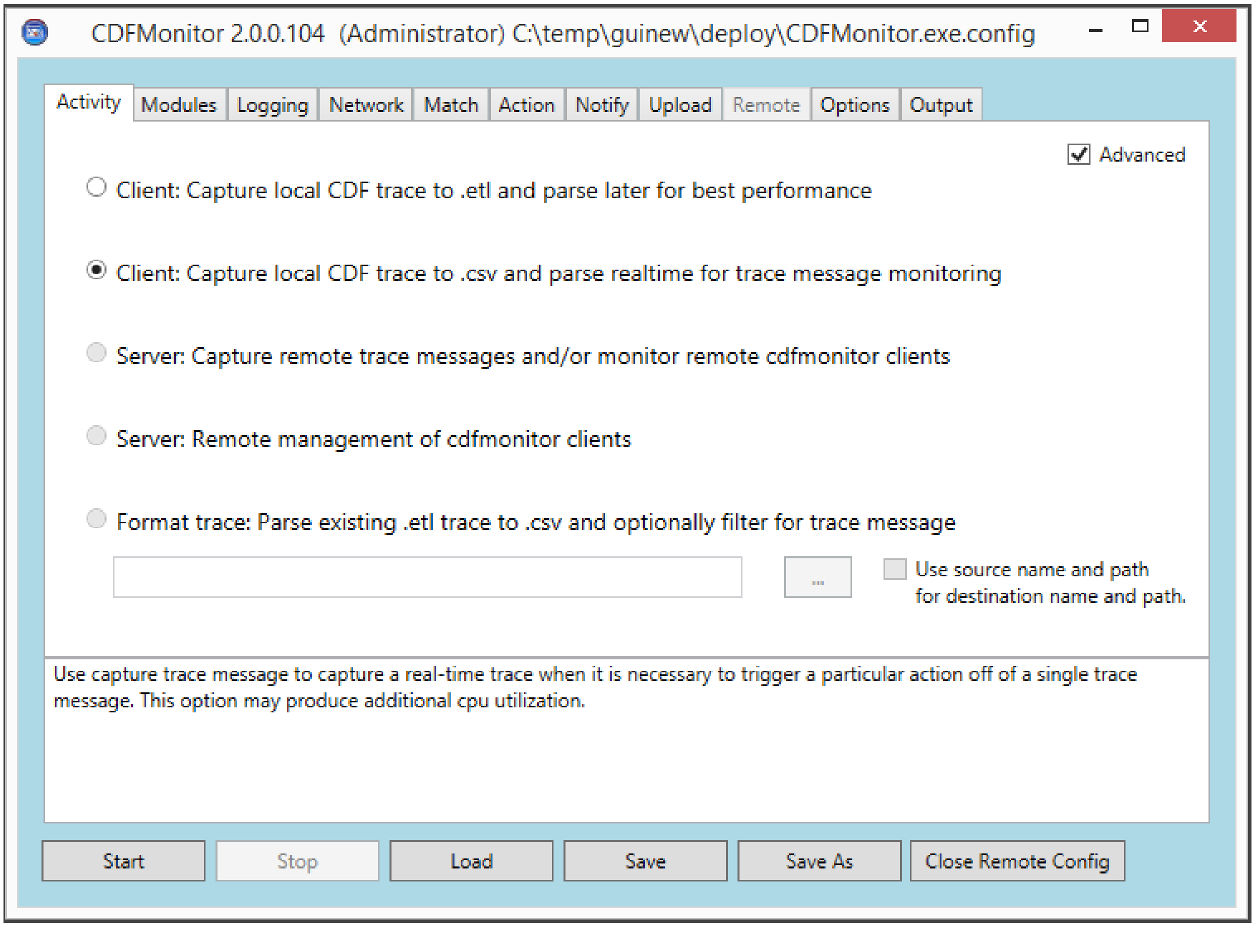


Deploy does not appear to be configured or configured correctly. Select Yes to setup / repair.

```
Process
ProcessRemoteMachines:exit:count:1
RemoteMachines items:1
Specify 'Deploy Source Path' location.
Fail:VerifySetting:Exit:DeployPath:False
```

  - Select: **Yes** to create a new subdirectory named ‘deploy’ in the working directory, copy a clean configuration to the new directory, and open the new remote configuration file for editing.
  - **NOTE:** When editing a remote configuration file, the CDFMonitor window form will change colors as shown below and the title bar will have (REMOTE CONFIG) in it.
  - Certain options will be preset and others will be disabled or read only when compared to the prior CDFMonitor view for local configuration. **Runas** will be set to **Service** for example. Other options are grayed as they do not apply for the activity being performed.
  - Make the changes for the remote configuration, **Save**, and **Close Remote Config. This is not editing a configuration file on a remote machine.**

- **Remote: Activity Tab**
  - Select: the appropriate activity for the remote config.



☑ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

● Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

○ Server: Capture remote trace messages and/or monitor remote cdfmonitor clients

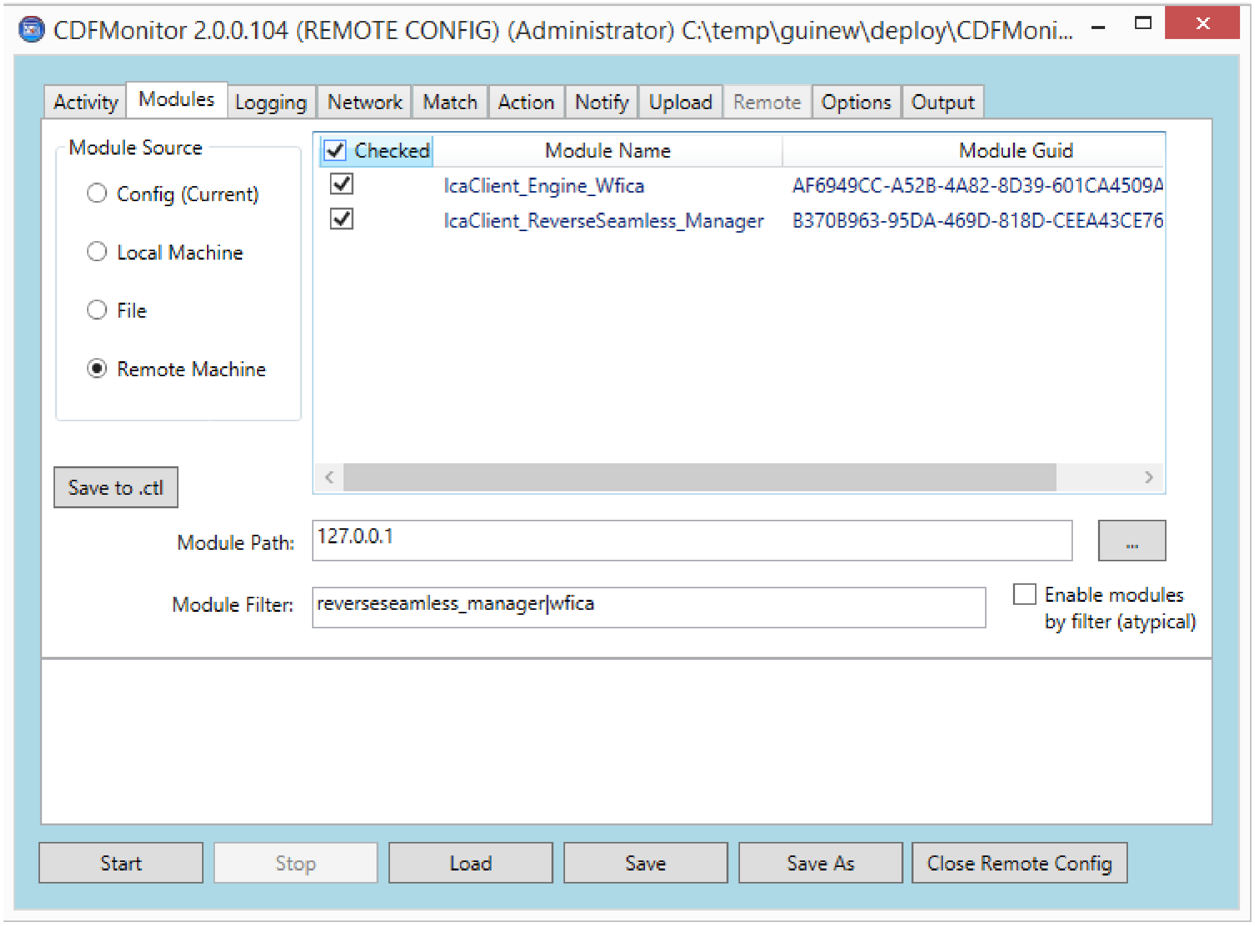
○ Server: Remote management of cdfmonitor clients

○ Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

□ Use source name and path for destination name and path.

Use capture trace message to capture a real-time trace when it is necessary to trigger a particular action off of a single trace message. This option may produce additional cpu utilization.

- **Remote: Modules Tab**
  - Select: the appropriate source and modules.
  - If **Remote Machines** is populated with a machine, on the modules tab, **Remote Machine** will be selected. The machine from the local config will be added to **Module Path** in the remote config as shown below and the modules list will be enumerated from that remote machines registry.
  - **Module Source** can be from any of the following sources and when saved will be saved into the remote config file for deploy.
    - **Modules Source** can be from any of the following sources and when saved will be saved into the remote config file for deploy.
    - **Config (Current)** will display what is currently embedded in the remote config file.
    - **Local Machine** will display modules from the local machine and not the remote machine.
    - **File** will display modules from file configured in 'Module Path'
    - **Remote Machine** will display modules from the remote machine specified in 'Module Path'



CDFMonitor 2.0.0.104 (REMOTE CONFIG) (Administrator) C:\temp\guinew\deploy\CDFMoni...

Module Source

○ Config (Current)

○ Local Machine

○ File

● Remote Machine

☑ Checked	Module Name	Module Guid
☑	IcaClient_Engine_Wfica	AF6949CC-A52B-4A82-8D39-601CA4509A
☑	IcaClient_ReverseSeamless_Manager	B370B963-95DA-469D-818D-CEEA43CE76

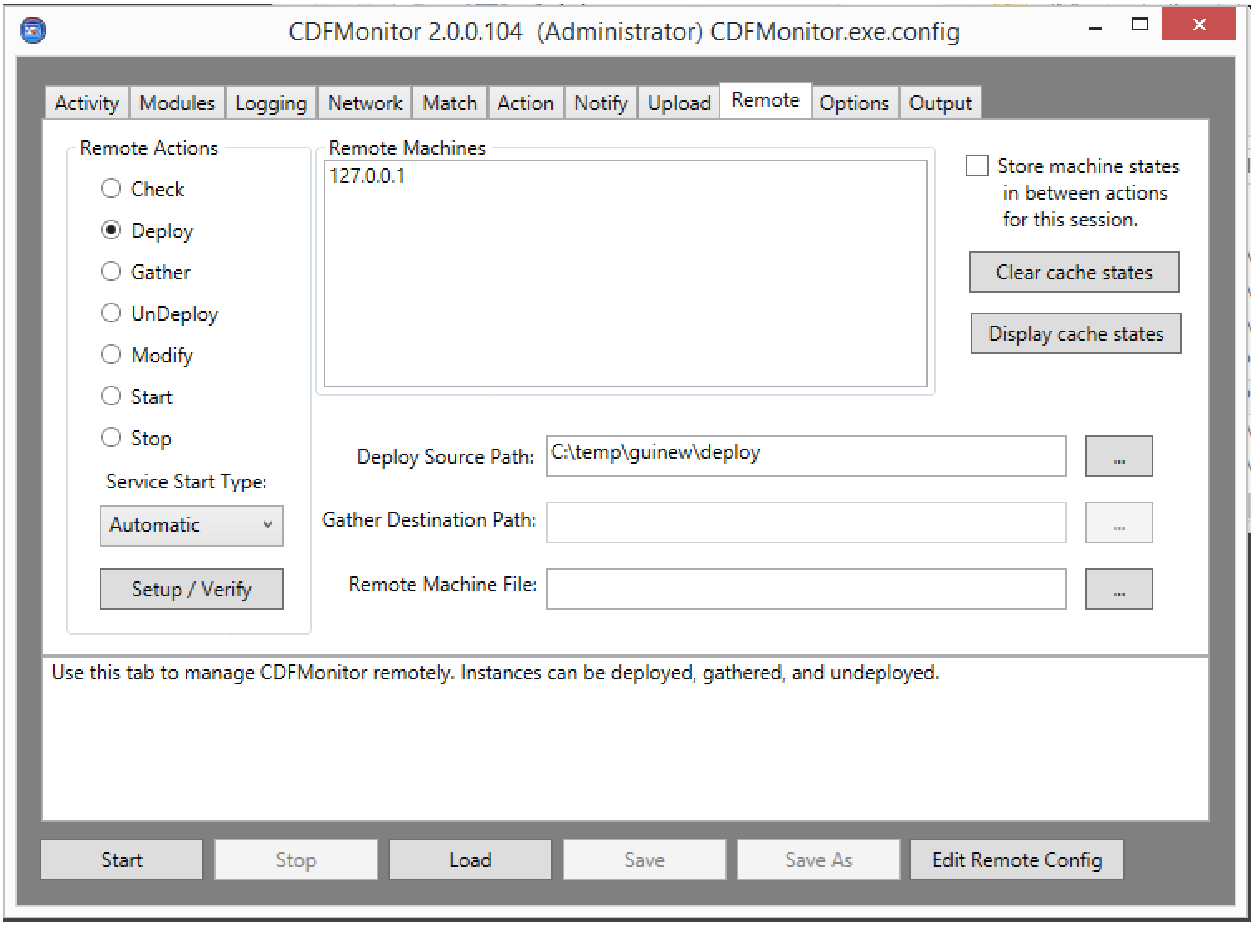
Module Path: 127.0.0.1

Module Filter: reverseseamless_manager|wfica

□ Enable modules by filter (atypical)

- **Remote: Logging Tab**
  - Use defaults for most use cases. See Logging for additional information.
- **Remote: Network Tab**
  - Optional: Advanced option that is useful for monitoring remote CDFMonitor deployments for trace messages or for remote management. Setting the IP address or FQDN of a CDFMonitor server instance for **Udp Ping Server**, in the remote config enables a UDP ‘ping’ packet with status info to the server. See Network for additional information.
- **Remote: Match Tab**
  - Optional: Advanced option when activity is set to **Client: Capture local CDF trace to .csv**. See Match for additional information.
- **Remote: Action Tab**
  - Optional: Advanced option when activity is set to **Client: Capture local CDF trace to .csv**. See Action for additional information.
- **Remote: Notify Tab**
  - Optional: Advanced option when activity is set to **Client: Capture local CDF trace to .csv**. See Notify for additional information.
- **Remote: Upload Tab**
  - Optional: Advanced option when activity is set to **Client: Capture local CDF trace to .csv**. See Upload for additional information.

- **Remote: Options Tab**
  - Optional: Advanced options for starting and stopping of traces based on Windows Event Log Event IDs. See Options for additional information.
- Select: After all configurations have been set, **Save** and **Close Remote Config**.
  - After closing, the local config of current CDFMonitor instance remote management activity will again be visible as shown below. The **Deploy Source Path** will now be populated with a deploy directory that contains the remote config file that was saved.
- See Walkthrough: Deploying a remote configuration for deploying the remote configuration.



CDFMonitor 2.0.0.104 (Administrator) CDFMonitor.exe.config

Remote Actions

○ Check
● Deploy
○ Gather
○ UnDeploy
○ Modify
○ Start
○ Stop

Service Start Type: Automatic

Remote Machines: 127.0.0.1

□ Store machine states in between actions for this session.

Deploy Source Path: C:\temp\guinew\deploy

Gather Destination Path:

Remote Machine File:

Use this tab to manage CDFMonitor remotely. Instances can be deployed, gathered, and undeployed.

Walkthrough: Deploying a remote configuration

The steps below demonstrate how to deploy a remote configuration to machines specified in **Remote Machines** on the Remote tab.

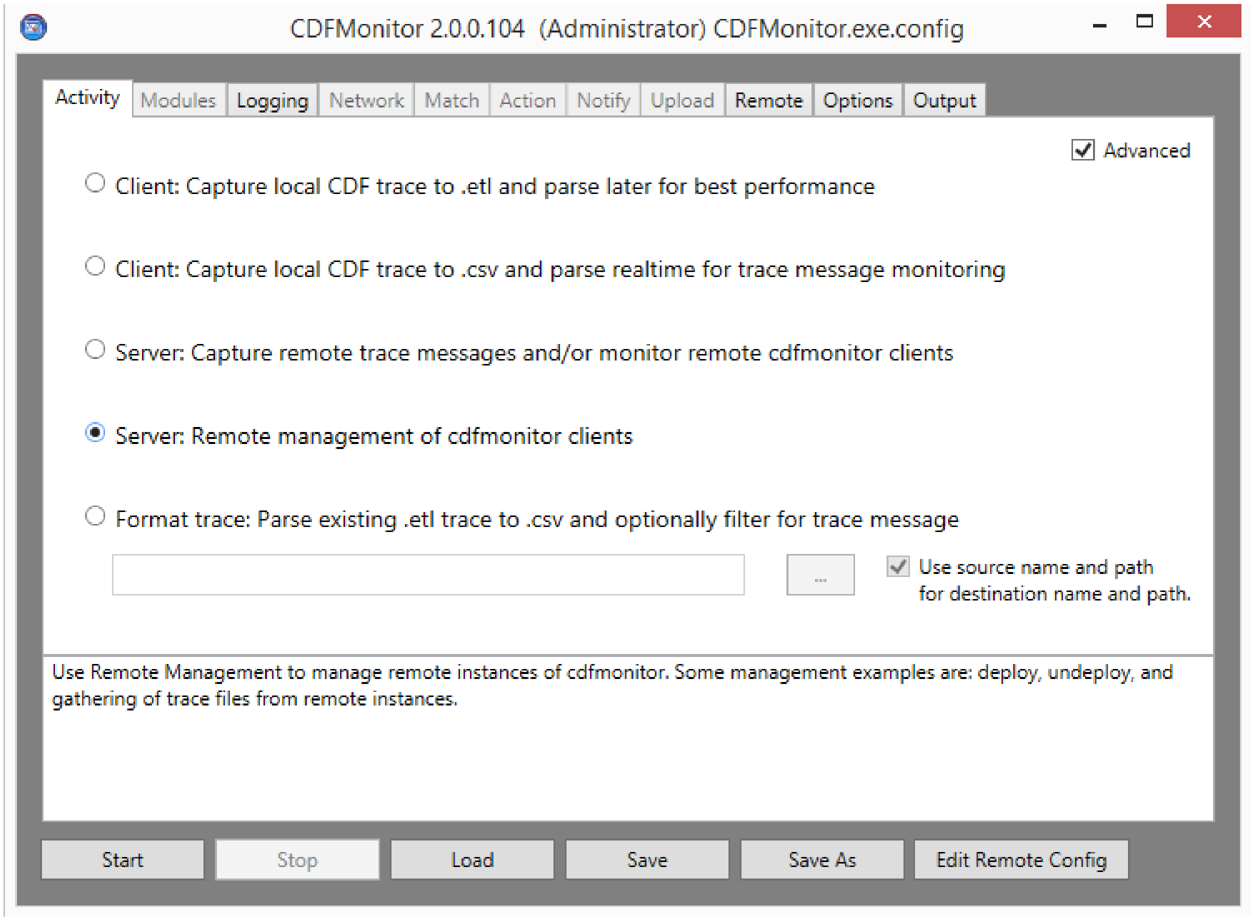
NOTE: Deploying CDFMonitor remotely requires WMI and UNC (SMB) access to the ‘admin$’ share of the remote machine.

Deploy will:

- Connect to remote machines.
- Stop CDFMonitor service if started.
- Copy cdfmonitor.exe and cdfmonitor.exe.config to ‘admin$\cdfmonitor’ from directory specified in **Deploy Source Path**.
- Start CDFMonitor service if configured.

NOTE: Before deploying a remote instance of CDFMonitor, a remote configuration has to be created or specified if one already exists. See Walkthrough: Creating a remote configuration.

- Open CDFMonitor
- **Activity Tab**
  - Check: **Advanced**
  - Select: **Server: Remote management of cdfmonitor clients**



☑ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

○ Server: Capture remote trace messages and/or monitor remote cdfmonitor clients

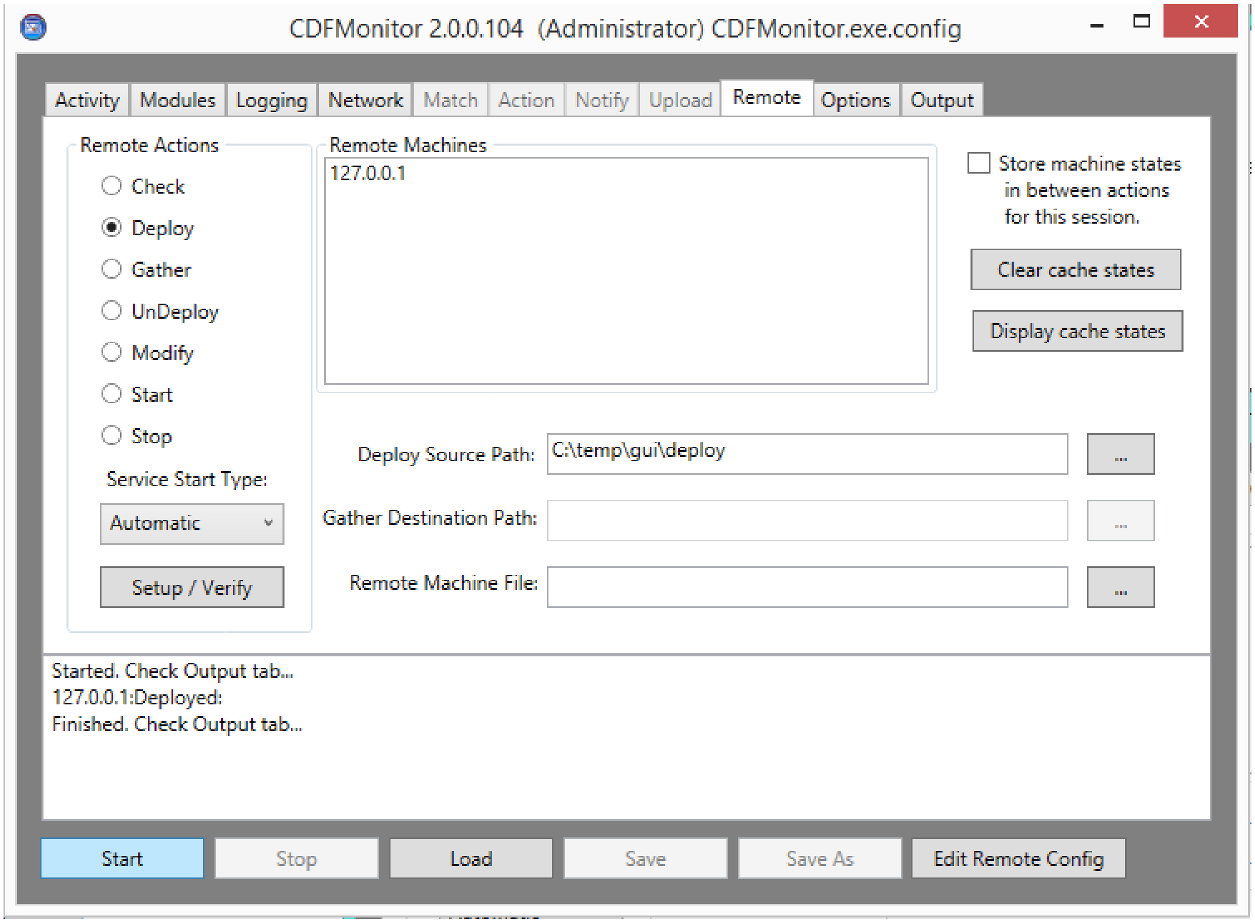
● Server: Remote management of cdfmonitor clients

○ Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

☑ Use source name and path for destination name and path.

Use Remote Management to manage remote instances of cdfmonitor. Some management examples are: deploy, undeploy, and gathering of trace files from remote instances.

- **Logging Tab**
  - Defaults are sufficient for remote management requirements.
- **Options Tab**
  - Use defaults for most use cases.
  - **NOTE:** if Alternate credentials are needed for connecting to remote machines WMI, selecting **Use Credentials** on Options tab will enable the use of alternate credentials for remote connectivity.
  - See Options Section for additional information.
- **Remote Tab**
  - Enter machine(s) IP address, FQDN, or name in the **Remote Machines** list.
  - Select: **Deploy**
  - Select: **Start**
  - Verify each machines’ status is ‘Deployed’ in status window or Output tab.



Remote Machines: 127.0.0.1

Deploy Source Path: C:\temp\gui\deploy

```
Started. Check Output tab...
127.0.0.1:Deployed:
Finished. Check Output tab...
```

Walkthrough: Gathering data from a remote configuration

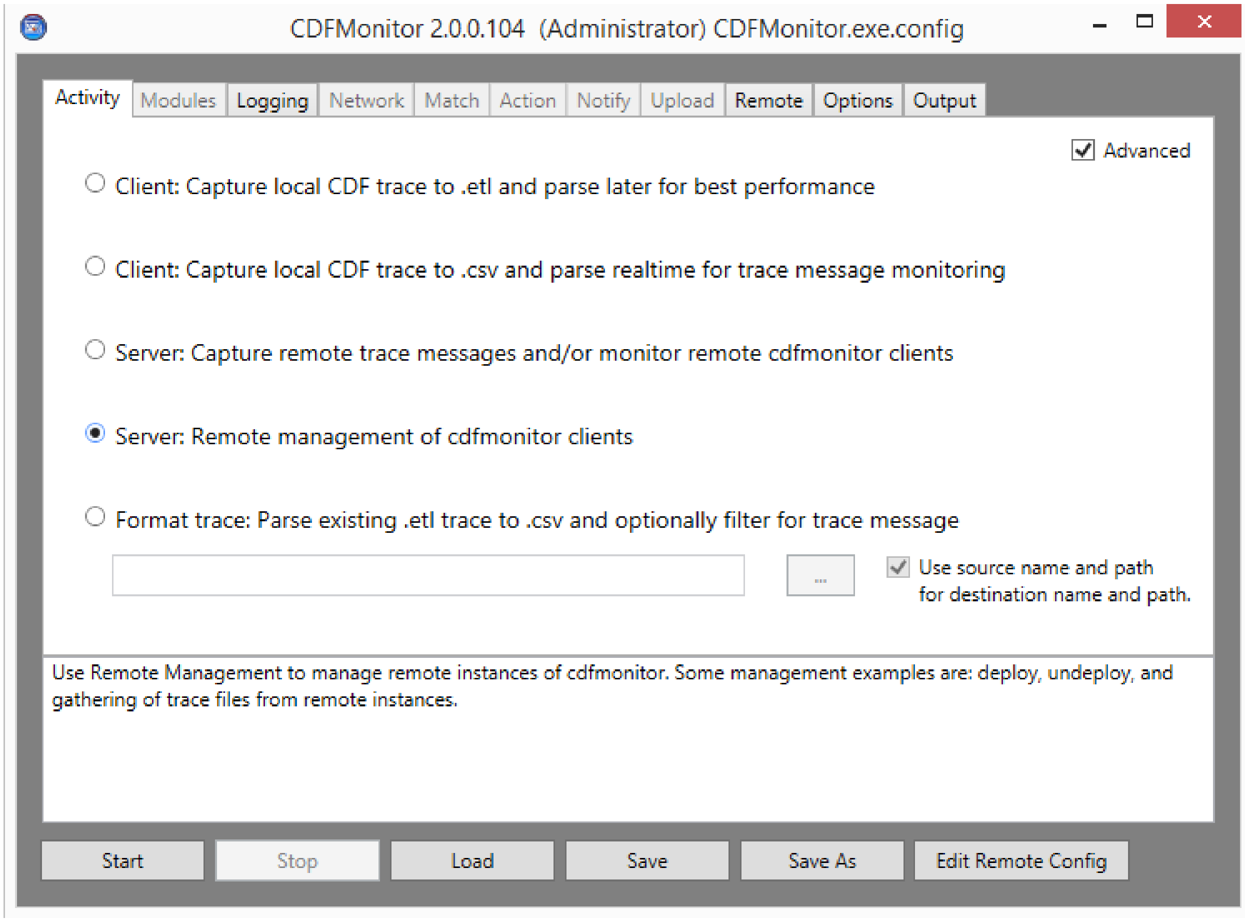
The steps below demonstrate how to gather data from machines specified in **Remote Machines** on the Remote tab.

NOTE: Gathering CDFMonitor data remotely requires WMI and UNC (SMB) access to the 'admin$' share of the remote machine.

Gather will:

- Connect to remote machines.
- Stop CDFMonitor service if started.
- Zip files specified in remote config file which is typically the trace and log files.
- Copy zip from 'admin$\cdfmonitor' to a machine named subdirectory specified in **Gather Destination Path**
- Restart CDFMonitor service if it was previously started.

- Open CDFMonitor
- **Activity Tab**
    - Check: **Advanced**
    - Select: **Server: Remote management of cdfmonitor clients**



☑ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

○ Server: Capture remote trace messages and/or monitor remote cdfmonitor clients

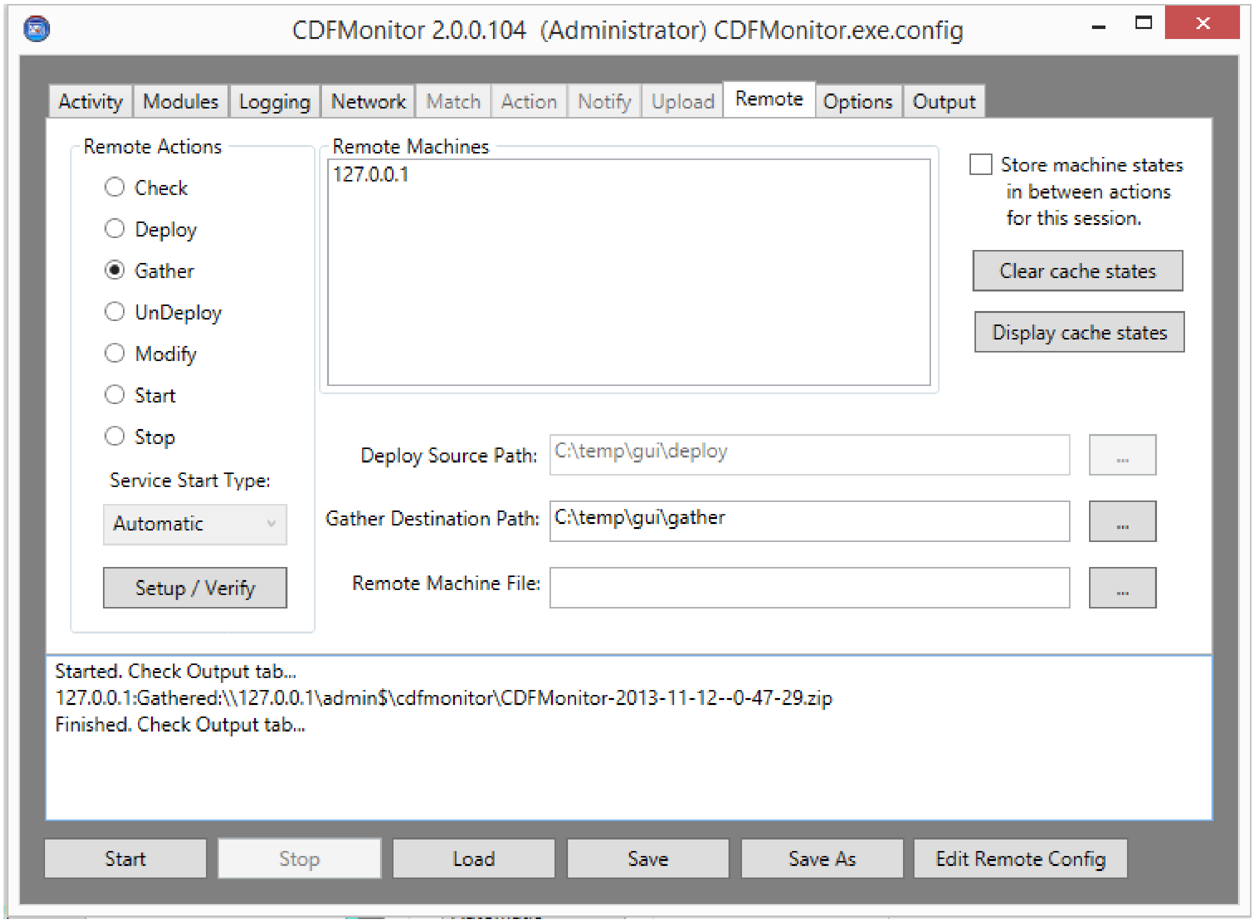
● Server: Remote management of cdfmonitor clients

○ Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

☑ Use source name and path for destination name and path.

Use Remote Management to manage remote instances of cdfmonitor. Some management examples are: deploy, undeploy, and gathering of trace files from remote instances.

- **Logging Tab**
    - Defaults are sufficient for remote management requirements.
- **Options Tab**
    - Use defaults for most use cases.
    - **NOTE:** if Alternate credentials are needed for connecting to remote machines WMI, selecting **Use Credentials** on Options tab will enable the use of alternate credentials for remote connectivity.
    - See Options Section for additional information.
- **Remote Tab**
    - Enter machine(s) IP address, FQDN, or name in the **Remote Machines** list.
    - Select: **Gather**
    - (OPTIONAL) Enter: a destination directory for **Gather Destination Path**. If one is not provided CDFMonitor will prompt to create one when starting activity or verifying setup.
    - Select: **Start**
    - Verify each machines' status is 'Gathered' in status window or Output tab.



Remote Actions

○ Check
○ Deploy
● Gather
○ UnDeploy
○ Modify
○ Start
○ Stop

Service Start Type: Automatic

Remote Machines: 127.0.0.1

□ Store machine states in between actions for this session.

Deploy Source Path: C:\temp\gui\deploy

Gather Destination Path: C:\temp\gui\gather

Remote Machine File:

```
Started. Check Output tab...
127.0.0.1:Gathered:\\127.0.0.1\admin$\cdfmonitor\CDFMonitor-2013-11-12--0-47-29.zip
Finished. Check Output tab...
```

Walkthrough: Undeploying a remote configuration

The steps below demonstrate how to undeploy CDFMonitor from machines specified in **Remote Machines** on the Remote tab.

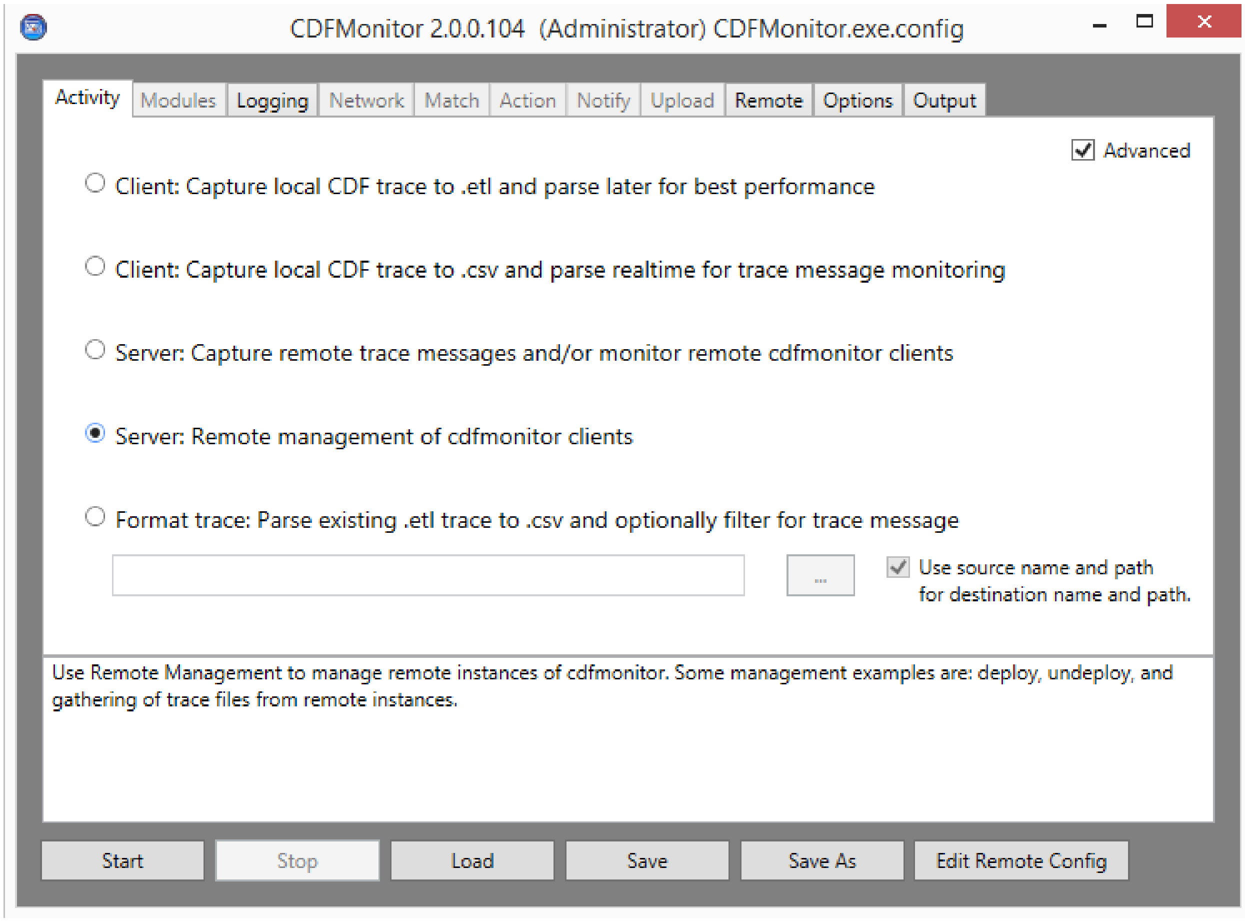
NOTE: Undeploying CDFMonitor remotely requires WMI and UNC (SMB) access to the 'admin$' share of the remote machine.

Undeploy will:

- Connect to remote machines.
- Stop CDFMonitor service if started.
- Uninstall CDFMonitor service.
- Delete all contents in 'admin$\cdfmonitor' directory and the directory itself.

NOTE: Undeploy will **NOT** gather or save any trace files. See Walkthrough: Gathering data from a remote configuration on how to gather data before Undeploying.

- Open CDFMonitor
- **Activity Tab**
    - Check: **Advanced**
    - Select: **Server: Remote management of cdfmonitor clients**



CDFMonitor 2.0.0.104 (Administrator) CDFMonitor.exe.config

☑ Advanced

○ Client: Capture local CDF trace to .etl and parse later for best performance

○ Client: Capture local CDF trace to .csv and parse realtime for trace message monitoring

○ Server: Capture remote trace messages and/or monitor remote cdfmonitor clients

● Server: Remote management of cdfmonitor clients

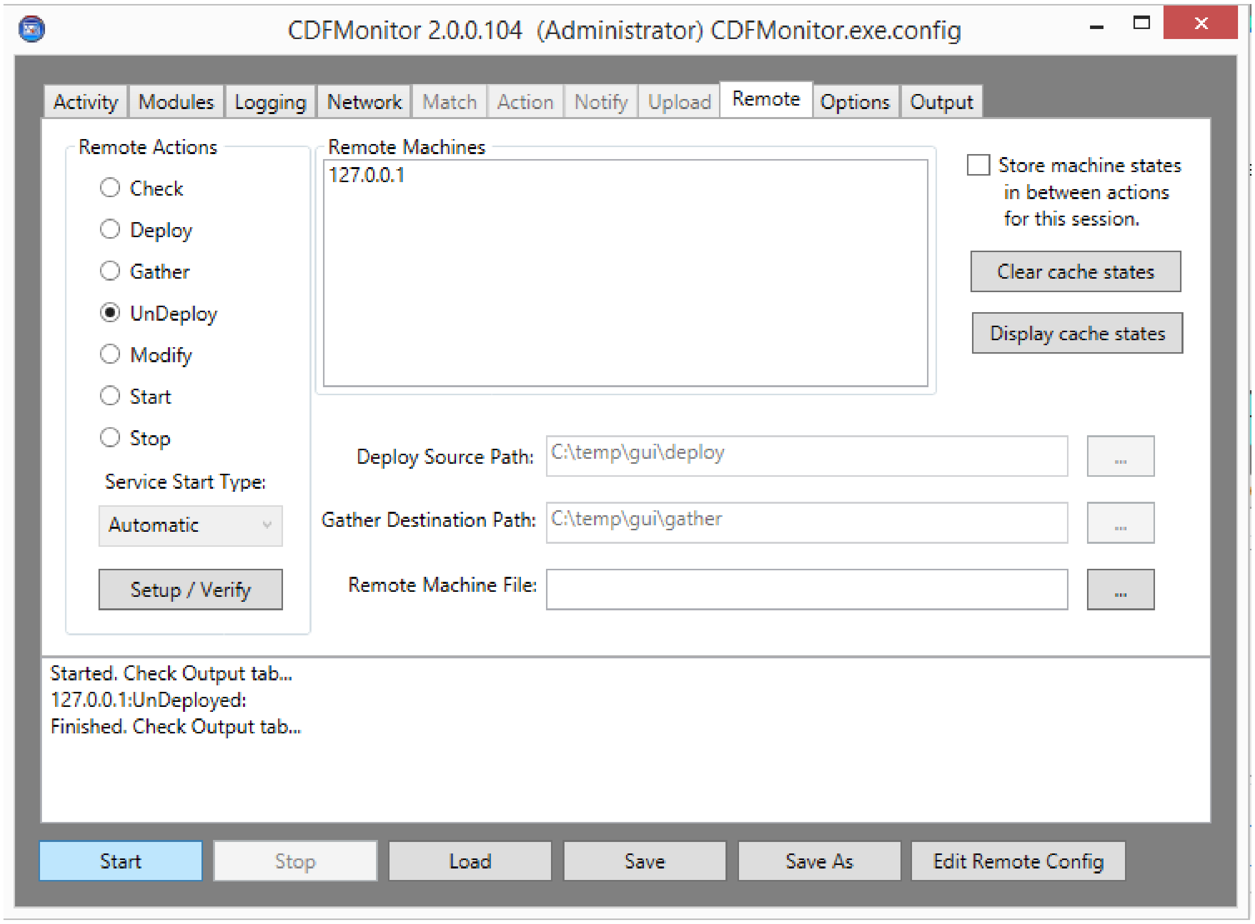
○ Format trace: Parse existing .etl trace to .csv and optionally filter for trace message

☑ Use source name and path for destination name and path.

Use Remote Management to manage remote instances of cdfmonitor. Some management examples are: deploy, undeploy, and gathering of trace files from remote instances.

- **Logging Tab**

  - Defaults are sufficient for remote management requirements.
- **Options Tab**
  - Use defaults for most use cases.
  - **NOTE:** if Alternate credentials are needed for connecting to remote machines WMI, selecting **Use Credentials** on Options tab will enable the use of alternate credentials for remote connectivity.
  - See Options Section for additional information.
- **Remote Tab**
  - Enter machine(s) IP address, FQDN, or name in the **Remote Machines** list.
  - Select: **Undeploy**
  - Select: **Start**
  - Verify each machines' status is 'Undeployed' in status window or Output tab.



Remote Actions:
○ Check
○ Deploy
○ Gather
● UnDeploy
○ Modify
○ Start
○ Stop

Service Start Type: Automatic

Remote Machines: 127.0.0.1

□ Store machine states in between actions for this session.

Deploy Source Path: C:\temp\gui\deploy

Gather Destination Path: C:\temp\gui\gather

Remote Machine File:

```
Started. Check Output tab...
127.0.0.1:UnDeployed:
Finished. Check Output tab...
```

851 West Cypress Creek Road Fort Lauderdale, FL 33309 954-267-3000 http://www.citrix.com