**Documentation on Process Monitor**

1. **General Information:**

* This project is built and run successfully on Windows 7 with Visual Studio 2010.
* The source is under \\Windows\\Source\\ProcessMonitor.
* Open \\Windows\\Source\\ProcesMonitor\\ProcessMonitor.sln file with Visual Studio 2010 and start rebuilding or modifying.

1. **User guide**

The executable file is under [\\Windows\\Demo](file:///\\Windows\\Demo), just hit it to start the program which immediately lists all processes on the machine and scans them by Metascan-online technology asynchronously. The list of process is updated every 5 second by default. The scanning for processes and their modules is updated right after results are sent back from Metascan online. Because the communication between the monitor and metascan – online is asynchronous, the scanning result for a process or a module could be sooner than the others.

The main interface is simple as Figure 1:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Figure 1: Main user interface |  |

Figure 1 shows the main user interface of ProcessMonitor. The table under Processes and Modules lists all processes detected on the machine. In this table, the first column is the process identifier (PID) of each process on the machine. The second, third and fourth are process name, process executable path and total thread open by the process, corresponding. Two final columns are the scanning results from metascan online for the processes and their modules. The result in the last column will be updated asynchronously. The line under the table tells statistical information about the number of processes and modules listed, as well as the number of processes/modules scanned up to that time.

Clicking any line on first table will activate the second one to show all modules of the selected process and their corresponding scanning results.

1. **Strong points**

The program is designed to report scanning result asynchronously with 5 threads: one for listing processes and their modules, one for checking hash code with metascan online when a new process is detected, one for sending binary file content of the process in case metascan online didn’t hash it, one the checking scanning percentage and the other for reporting results to the user interface. The system doesn’t wait for the scanning result from metascan online in synchronous fashon, so the overall time running should be lower.

Hash code of repeated modules is cached on local machine, so the system doesn’t need to make a checking with metascan online for them.

The scanning result is cached on local machine, which means, in the next process listing, the system will check the processes and their modules with the local cache and only request metascan online when a completely new process or/and modules are detected.

The project uses built-in or self-designed technologies to deal with making request, receiving response and parsing json string from metascan online.

1. **Limitation:**

* Functionalities on Options Dialog as in Figure 2 need be edited more.
* The engine to send and receive data asynchronously to and from Metascan online needs a permanent api key to work without interruption for a long time.
* Update frequency is another functionality that needs to be fulfilled.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Figure 2: Options Dialog |  |