Intel® Platform Power Estimation Tool (IPPET) Quick Start Guide v1.23

IPPET is a prototype power monitoring utility that uses Intel-specific energy MSRs to break down power consumption per process and displays them in real-time on a web browser. IPPET uses the EnumProcesses API to get a snapshot of all running processes and then spread the cpu and GPU power over non-idle processes running during the interval. IPPET can display the power data using a built-in web server by connecting their browser to localhost. For security reasons, remote connections to the localhost server are not allowed at this time.

System Requirements

* Hardware must be Sandybridge or later chip. (Tool requires energy MSRs)
* Port 8080 open for localhost connection (this should be the case by default)
* Windows 7 and 8
* If you are using Microsoft Internet Explorer, you need version >= 9.
  + IE compatibility mode doesn’t work (since it reverts IE back to ver8)

Installing IPPET

1. Run the IPPET\_v1.\*\_setup.exe file.
   1. The default install dir is c:\Program Files (x86)\Intel\Intel(R) Platform Power Estimation Tool\
   2. The installer will prompt you to download the WinRing0 driver files if it can’t find the files on your system.
2. The WinRing0 driver files should have been installed during the install process but here is the manual process:
   1. You need to download the winring0 driver from realTemp.
      1. <http://www.techpowerup.com/downloads/SysInfo/Real_Temp>
         1. click on 'Real Temp 3.70',
         2. click on 'Download now'
         3. select the server to download from
         4. click on ‘open’ or ‘save’ file.
         5. unzip RealTemp\_370.zip somewhere
            1. the IPPET installer will ask for the location of the winring0.sys file during installation. The installer will copy the 4 WinRing0 files to the IPPET dir.
         6. copy the RealTemp WinRing0\* files to the the IPPET dir.
            1. If you've done everything right then the files:

WinRing0.dll

WinRing0.sys

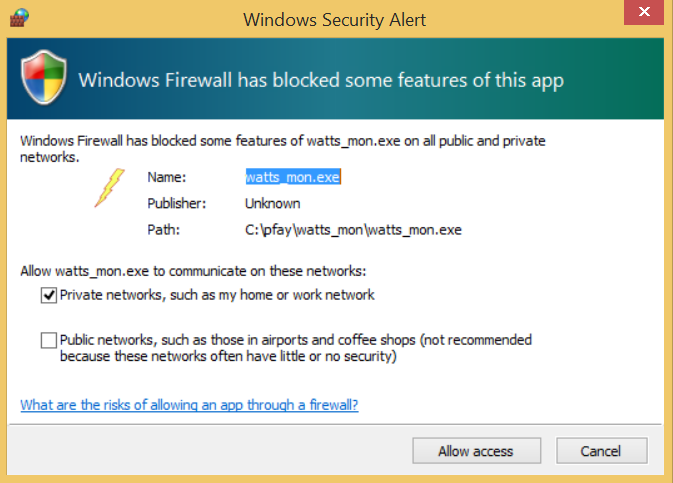
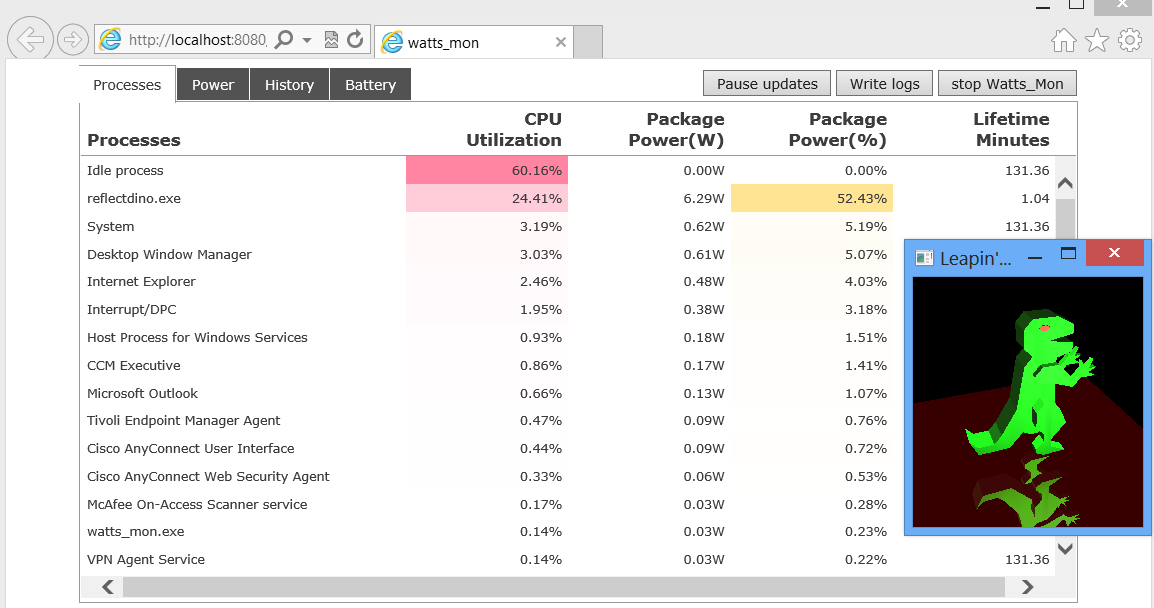
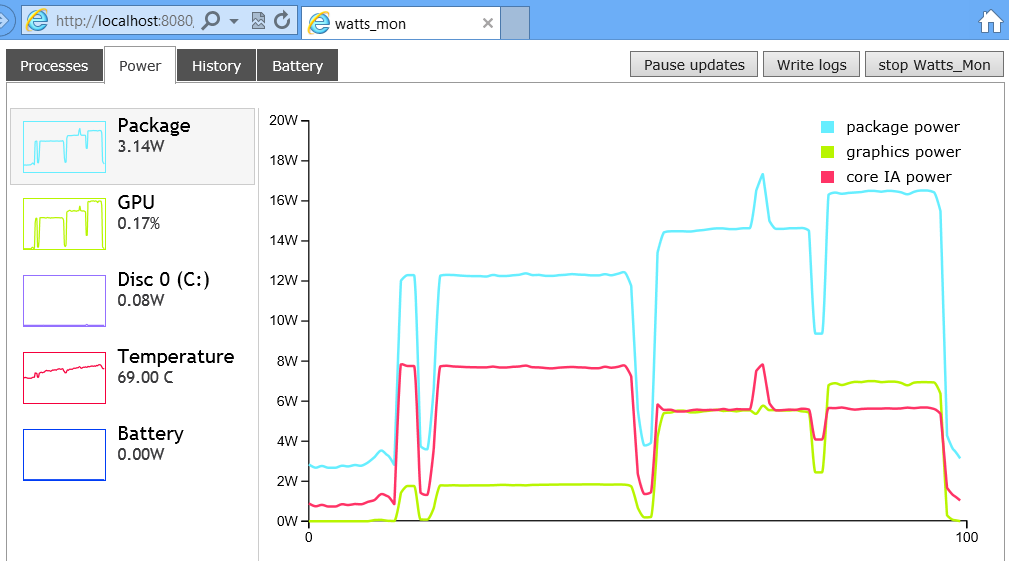
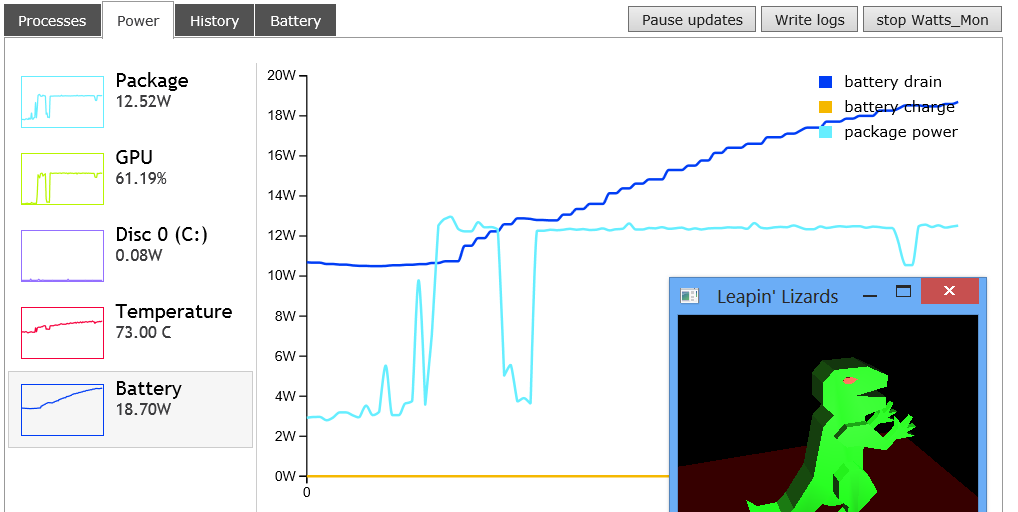
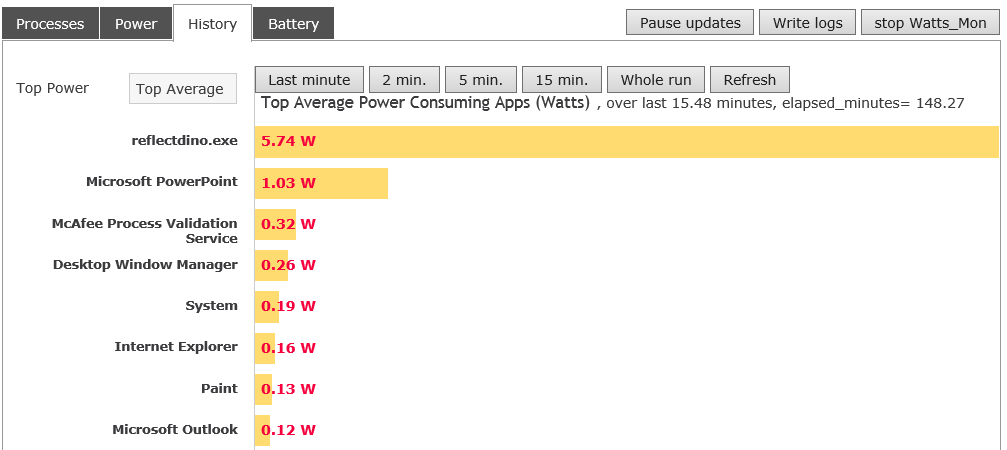
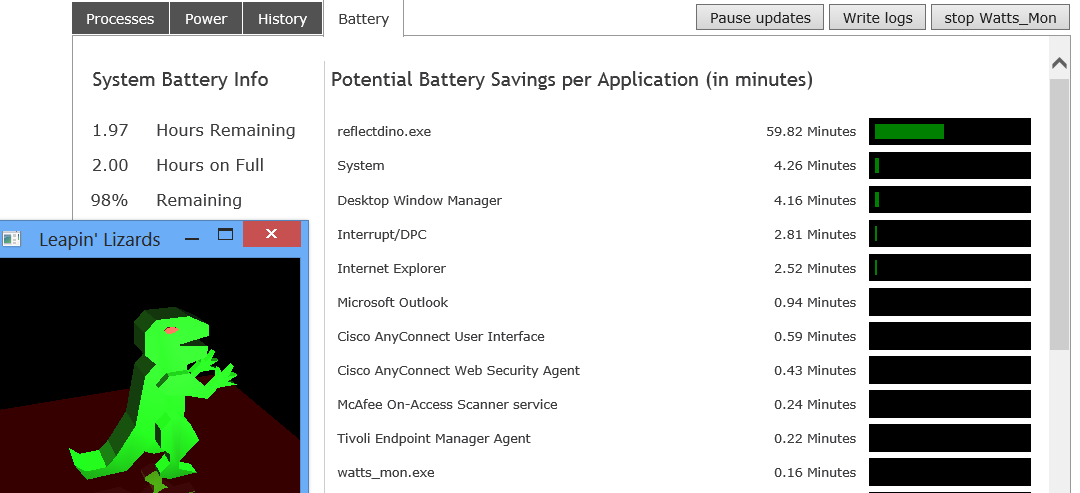
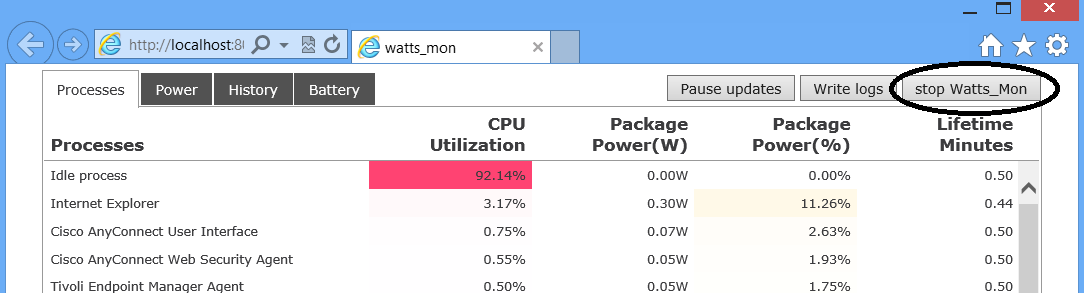
WinRing0x64.dll

WinRing0x64.sys

* + - * 1. Will be in the same dir as ippet.exe. If you haven’t installed
    1. The above 4 WinRing0\* files have to be in the same dir as ippet.exe

1. You should be ready to go now.

Running IPPET

1. Click on the IPPET icon on your desktop. This assumes you’ve run the IPPET\_\*\_setup.exe and gotten the WinRing0 driver files from RealTemp.
   1. You will get a request to ‘run as administrator’.
   2. IPPET should start in the same dir as where the IPPET.exe is installed. The log file will be put in the IPPET dir once you stop IPPET.exe
   3. You may see be a popup like below (but I don’t think you’ll get the popup). By default, IPPET doesn’t allow connections between your system and other systems. You can uncheck both ‘Private networks’ and ‘Public Networks’. Then hit ‘Cancel’. Note that the image below references watts\_mon, the old name for IPPET.
   4. 
2. The power data collection will start now
3. A browser window should start and the IPPET web page should display.
   1. The browser window make take a few seconds to load
   2. You may get a message from Windows Firewall asking if it should block IPPET.exe. If you get this popup, please see step 3c above.
   3. You can turn off the ‘auto load the web page’ with command line option ‘-auto\_load n’
4. If you need to load the web page manually, open the browser of your choice and type [http://localhost:8080/IPPET](http://localhost:8080/watts_up)
   1. There can only be 1 instance of the backend server running at a time but you can have multiple browser windows displaying simultaneously (not sure why you would want to but you can).
5. The IPPET web page ‘Processes’ tab displays CPU utilization per process along with power breakdown (This will be soon changed to app name per feedback in EBL WG meeting).
   1. Note that the ‘Package Power(W)’ includes both CPU and GPU power.
   2. 
6. ‘Power’ tab shows component power for CPU, GPU, Disk (assumes SSD), Temperature and Battery
   1. 
   2. In the above case, the graph indicates starting a graphics intensive app (reflectdino.exe). The reflectdino window starts out small and uses little of the GPU. Then each new plateau in graphics power indicates where the reflectdino window was increased in size and used more of the GPU.
   3. Here is a screen shot of the Power->Battery tab. The battery usage data supplied by the OS appears to have a 1-5 minute moving average built in so as you increase power usage, the battery usage eventually reaches a steady state (assuming that the load on the battery is also a steady state load). The screen shot below shows that ‘package power’ exceeding the ‘battery drain’ which is clearly not possible. The ‘battery drain’ show a moving average drain rate rather than the actual instantaneous drain rate
   4. 
7. The ‘History’ tab shows current summary of the power consumption across various processes. ‘Top Power’ indicated total consumption for the current history interval in WHr. ‘Top Average’ indicates average power consumption for current collection in W. You can change the history interval by clicking on the Interval buttons. For instance, when you click on the ‘Last minute’ button, you are setting the start of the history interval to the current time (NOW) minus 1 minute. Clicking on ‘2 min.’ sets the start of the history interval to ‘Now – 2’ minutes, etc. You have to hit the ‘Refresh’ button to get the updated display… The history is somewhat expensive to compute so it is only re-computed on demand.
   1. The screen shot below shows that reflectdino.exe uses (when it runs) on average 5.74 Watts of power.
   2. The ‘Last minute’, ‘2 minutes’, etc buttons let you reset the starting point for the history calculation. For instance, say I kill reflectdino.exe. It will still be the highest power using app for the entire run but it is no longer running. So if I kill reflectdino then wait a minute, then hit the ‘last minute’ button (and the Refresh button), then reflectdino.exe will no longer appear in the history (since reflectdino will have moved out of the window of history interval).
   3. 
8. The Battery tab shows how many minutes you could ‘get back’ for your battery lifetime if you terminate a process.
   1. The screen shot below shows, for instance, that you could ‘get back’ 59.82 minutes of battery time if you closed reflectdino.exe.
   2. 
9. Once you close the command prompt, a compressed Excel file will be created in the current dir which stores raw data as needed for processing.
10. To stop IPPET, click on the ‘stop IPPET’ field at the top right of the GUI. See the circled field below.
    1. The IPPET backend server process will end but the web page remains.
11. 
12. You can also close the backend by hitting control-C on the back-end window.

IPPET features

For usage options, run the command: **IPPET.exe -h**

1. Let’s you turn on/off **logging** and change the name of the log file or specifying file location
   1. You can enable/disable zipping the files after
   2. You can use a circular buffer type behavior
2. Enable **printing to screen** or **starting the web server**, or both
3. Option to **include** **all processes** instead of the default non-zero cputime processes
4. Enable/disable GPU, disk, device power state, battery, data collection
5. Display processes using the process binary name or the process description (if the binary has a description).
6. Specify the interval for reporting the power usage.
7. Set how long to run.
8. Pass a command to IPPET. IPPET will run the command, write the logs and terminate.
   1. When running in this mode, you might want to add the option ‘-e n’ to disable printing output and disable the web update.
9. More options in the –h output.

You can also set default options in the ippet\options.txt file. See the help output for more info.

IPPET known issues and feature requests

List of known issues. See the –h output and versions.txt for more info.

1. The process lifetime can be wrong in the \*sys\_info.xls file. Windows 8 doesn’t reset ‘uptime’ when you boot the computer. If you do a restart, then uptime resets properly but doing a shutdown and then boot doesn’t reset the uptime. You can see this (on win 8) in Taskmanager’s performance tab->click on CPU icon->see Up time field. Also, most processes have a valid CreationTime but some processes have CreationTime=0 so IPPET has a hard time saying how long the process has been running. IPPET would like to use ‘up time’ as the creation time for the processes with CreationTime=0 but the invalid uptime field makes this impossible. The uptime field doesn’t give sensible results on Windows 8 unless you’ve rebooted. IPPET avoids this issue on Windows 8 by using the time that IPPET was started as the CreationTime for all processes (if the process’s CreationTime is 0 or is before IPPET’s StartTime).
2. I’m not sure how to display time for Metro Apps. All the cpu time for Metro apps gets charged to some other process.
3. If processes start and end between Process snapshots then IPPET won’t “see” the process. This issue could be avoided by using ETW to get process create/destroy events and will perhaps be implemented in future releases.
4. The Atom processor with cpuid model 0x4d (processor formerly codenamed Baytrail) CPU power formula is not correct at the moment in v1.23.

Feature requests.

1. Display more detail for Interrupts. I’m not sure how to do this yet.
2. Be able to change the interval for the history charts. This has been implemented as of v1.16.