Article

Murray Oldfield · Feb 20, 2017 3m read

 Open Exchange

Yape - Yet another pButtons extractor (and automatically create charts)

Note (October 2022): yape has been deprecated and replaced by YASPE, there is no more development on yape.

Note (June 2019): A lot has changed, for the latest details go here

Note (Sept 2018): There have been big changes since this post first appeared, I suggest using the Docker Container version, the project and details for running as a container are still in the same place published on GitHub so you can download, run - and modify if you need to.

Working with customers on performance reviews, capacity planning and trouble-shooting I am often unpacking and reviewing Caché and Operating System metrics from pButtons. Rather than wade through html files cutting and pasting sections to be graphed in excel I published a post a while ago with a utility to unpack pButtons metrics written using unix shell, perl and awk scripting. While this is a *useful time-saver* its not the full story... I also use scripts to automatically chart the metrics for quick review and inclusion in reports. However these charting scripts are not easily maintained and become especially messy when there is site specific configuration needed, for example list of disks for iostat or windows perfmon, so I never publicly published the graphing utilities. But I am happy to say there is a much simpler solution now.

Serendipity occurred when I was at a customer site looking at system performance with Fabian and he showed me what he was doing using useful Python modules for charting. This is a much more flexible and maintainable solution than the scripts I was using. And the ease of integrating Python modules for file management and charting, including interactive html you can share, means the output can be much more useful. Taking Fabians posts as a base I wrote **Yape** to quickly and simply extract then chart customer pButtons files in several formats. The project is published on GitHub so you can download, run and modify if you need to.

Overview

At the moment this process has *two* steps.

Step 1. extractpButtons.py

Extract interesting sections from pButtons and write to .csv files for opening with excel or processing with charting with graphpButtons.py.

Step 2. graphpButtons.py

Chart files created at step 1. Currently output can be line or dot charts as .png or as interactive .html with options to pan, zoom, print, etc.

Readme.md on GitHub has details on how to set up and run the two python scripts and will be the most up to date reference.

Other Notes

As an example: Using the options for adding prefixes to the output and input directories you can easily walk a directory with a set (e.g. a weeks) pButtons html files and output to a separate directory for each pButtons file.

```
for i in `ls *.html`; do ./extract_pButtons.py $i -p ${i}_; done

for i in `ls *.html`; do ./graph_pButtons.py ./${i}_metrics -p ${i}_; done
```

In the short term while I continue with the series on Caché capacity planning and performance I will use the charts created with these utilities.

I have tested on OSX, but not on Windows. You should be able to install and run Python on windows, please leave feedback on your experience with Windows. I suspect you will have to make changes to the file path slashes for example.

> Note: Until a few weeks ago I had never written anything in Python, so if you are a Python expert there may be some things in the code that are not best practice. However I use the scripts nearly every day so I will continue to make improvements. I expect my Python skills will improve — but please feel free to ' educate' me if you some see something that should be corrected!

If you find the scripts useful let me know and check back regularly to get new features and updates.

#Best Practices #Monitoring #Performance #System Administration #Tips & Tricks #Caché #InterSystems IRIS #InterSystems IRIS for Health
Check the related application on InterSystems Open Exchange