Project to Extended WebGUI, for Unix/Linux Servers.

The objective of this project is not to be another tool for monitoring or administration, but to replace the basic need of direct access to the servers, working like a layer to keep interlaced user and server without a login and after that a change to another user with necessary privileges. That application can be executed with privileges enough to run all the instructions, tools and commands easily and faster.

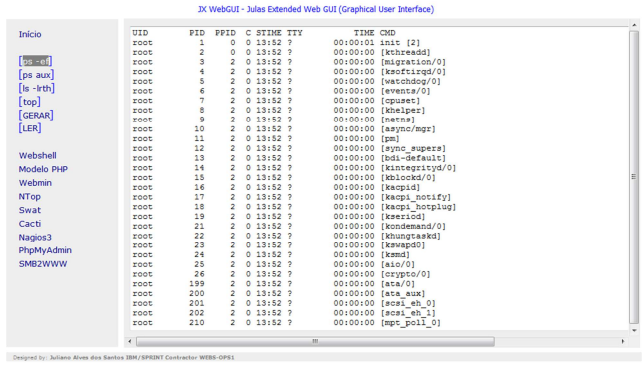
We must think about that layer like a Documentention as our WIKI, but a lot smarter, because we have not only the information about our Tasks, but the opportunity and condition to run it showing the System STDOUT to check if everything is working fine.

In specific cases when the usual command do not work, you can use the GET area to input an incremental command with parameters, or better, you can change to a WEBSHELL where you have a TRUE Console Command Line to work in the server you need with privileges granted already.

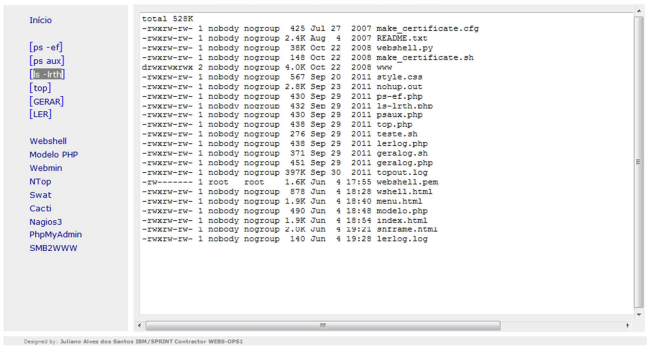
Below i've attached some screenshots of the funtional application, after each image i'll explain the steps and porpoise.

This is the initial page, here we have the menu area and the main page where we see the message, and when executed any command in the menu we will see the STDOUT of the OS.

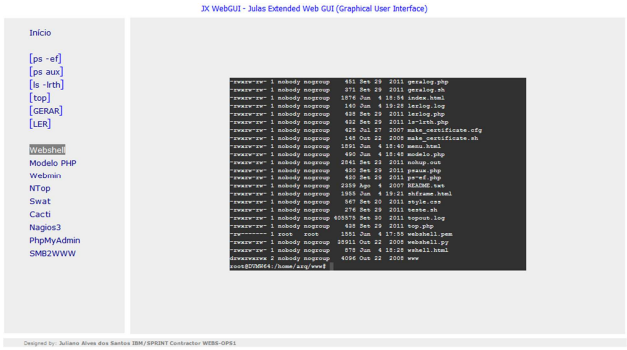
In that initial page i'm planning to keep the documentation page, only with information texts to teach all about each application, usage, monitors and alerts.



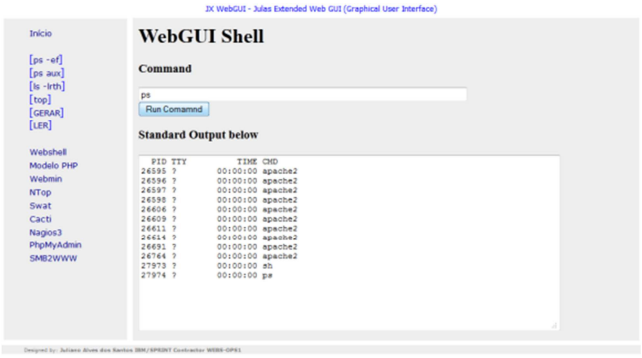
Here I ran the command ps –ef using the menu, I received all the STDOUT in the main frame. A big advantage over the console is the STDOUT cache, because we have a limited number of lines in the Shell, and here we can navigate in all STDOUT without any concern to lose info because the limited cache.



That another example I ran the command ls –lrth and I can see all the STDOUT organized in columns exactly the same way like in the Shell. Take a look in the file owned by ROOT. Is a certificate to grant privileges to run another tool of this project, we will see that tool just below.



Here I'm logged in a shell totally useful as a true Command Line, and the connection is granted due a certificate based in the user owned with necessary privileges. In this case I'm using Linux Debian 6, and the user is ROOT. It's possible to use with another user without any big issue.



Here is the example to run a customized or incremental command with different parameter as configured in the menu using a GET BOX to this.

See that the STDOUT is showing all info in a Text Box, this is a way to input both, command and OUTPUT in a SQL DataBase to keep history or documented actions for each day, week or month.

Obviously this is a very basic sample for this project, but for now, is a concept to grant and inspire a dimension of possibilities, I'm basing this concept in my own experience with support for Servers in clients outside IBM, where I use this to simplify and centralize all my actions in one place.

Some tools listed in the GUI, are existent tools for Linux as Webmin, Cacti, Nagios, Swat and NTop. I,m using just to illustrate a full integration of tools in only one place. I hope with this can clarify the intent to use this. Because we spend a lot time alternating between users, servers, searching passwords, monitoring and checking system health. With that tool we could make better use of resources by using less time monitoring and more in production and improvements in performance.

We still have the master factor, the economy, because with a tool like that, the job turns to be very simple, and the client can contract apprentices, with low costs because the tool make the main job, and the other and more able resources can be the source of maintenance like a next level, in cases beyond of the apprentices knowledge and to feed the Application commands and tasks.

Juliano Alves dos Santos
IBM Contractor for Sprint – Webs-Ops1