

# Technical Disclosure Commons

---

Defensive Publications Series

---

June 13, 2017

## A Method to Manage Multiple Tabs in an Efficient Way

Darel Huang  
*Hewlett Packard Enterprise*

Andy Liang  
*Hewlett Packard Enterprise*

Nate Wu  
*Hewlett Packard Enterprise*

Follow this and additional works at: [http://www.tdcommons.org/dpubs\\_series](http://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Huang, Darel; Liang, Andy; and Wu, Nate, "A Method to Manage Multiple Tabs in an Efficient Way", Technical Disclosure Commons, (June 13, 2017)  
[http://www.tdcommons.org/dpubs\\_series/553](http://www.tdcommons.org/dpubs_series/553)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## A Method to Manage Multiple Tabs in an Efficient Way

### Abstract

There are lots of enterprise software which can help users to remote use the server/OS/machine and also help to monitor server/OS/machine status. Let's take an example with a Baseboard Management Controller (BMC) product. By using a BMC, a user can remote login to multiple servers and check out the system status and configuration by different webpage.

Our solution is to help users to monitor multiple servers in 1 single page by using the web browser tabs. Users can be alerted for some emergency events by the different color web tabs. Different events or activities information will be pulled in and show up on belonging web tabs. Users can also set up different buttons on the web tab for further operation actions. By using this solution, users can monitor the servers easily and execute further actions in a convenient/centralized way.

### Detail

This disclosure relates to help users to monitor multiple servers in 1 single page by using the web browser tabs.

A web browser tabs method is disclosed that to help users to operate in an efficient way. Users can be alerted for some emergency events by the different color web tabs. Different events or activities information will be pulled in and show up on belonging web tabs. Users can also set up different buttons on the web tab for further operation actions. By using this solution, users can monitor the servers easily and execute further actions in a convenient/centralized way.

There are lots of enterprise software which can help users remote the servers, Oses, machines, and monitor their status. Let's take an example with a BMC product. By using BMC, users can remote login to multiple servers, check out the system information and status, change system configurations, even perform firmware updates in a single webpage.

However, when users need to monitor or make configuration changes to multiple systems in BMC, they have to open multiple browsers or, more likely, multiple tabs in one browser. In this case, users will have to keep eyes on all the tabs at once, switching from tab to tab constantly. Furthermore, whenever there are critical warnings or actions to be performed timely on a specific system, there's a big chance that the user will neglect the messages since there are too many tabs to monitor at the same time.

So here we're proposing a solution to help users monitor status and perform actions for multiple tabs in a more convenient way.

There are inconvenient and common use cases in current model, which are the major problems we trying to resolve with our proposed solution.

- Users need to switch from tab to tab to check status or take actions for different systems.
- Users are very likely to overlook the notification if one of the many tabs has critical warning.
- It is inconvenient that current tab icons do not allow user customization.
- It is ideal if the emergence of the icons could be intelligently determined by use frequency or time-sensitive actions based on user data analysis and webpage activity detection.

The browser tabs we see today consisted of 2 elements: a favicon - an icon which symbolize the website; and a text field - a text area to describe current content of the website. Both the favicon and the text field are controlled by HTML code of the webpage. There are no functions for users to interact with.

In our proposal, we would like to make 'click-able icons' on the tab and their functions are mapping to the 'actual functions' provided by the webUI of the browser tab. For example, in iLO we have the system 'power on' button in its webUI; with current model, users will need to switch to the tab, go through webUI and find the 'power on' button, click the button, then return to current tab; however, with our solution, we will make correspondent 'click-able icon' for the button and put it on the browser tab, so the users can click on the icon directly and perform the action(power on) for that specific system.

The other function we'd like to introduce is the 'notify tab' which could help notify users of all kinds of messages from the webUI of the browser tab by changing the tab color. For example, in iLO we have the 'system health information' in its webUI; with current model, users will need to switch to the tab, go through webUI and find 'system health information', check status, then return to current tab; with our solution, we will setup a criteria table and change the tab color by event or status severity according to the criteria table, so the users can identify the system status or problem severity by the tab color and handle issues with priority.

In our design, both the 'click-able icon' and 'notify tab' are able to be customized. These functions should be implemented by web or browser plugin developers, and eventually, users should be able to customize the settings by themselves (to decide which click-able to be put on the tab or what kind of color represents critical event on the tab).

Last, 'Smart display', a more advanced feature of our design is to analyze and detect user preference and webpage activities. Based on the information collected, intelligently determine the emergence of the icons by use frequency or time-sensitive actions, and make the 'icon customization' auto and smart. For example, when we open a remote session from a BMC web interface and want to enter the BIOS menu, we have to monitor the remote session windows closely and press 'F1' once the system is ready to accept function key strokes, however, if we missed the timing to press the function keys, we'll have to reboot the system and start over; with 'Smart display', it will detect the current activities of webpage and determine which 'click-able icons' to be displayed on the tab(in this case, 'F1', 'F2', 'F9', 'F11', ..., etc.), so we can simply click the 'F1' on the tab once available without actually stay on the remote session window, which means you can switch to other windows and perform other tasks while waiting and you won't missed it! Despite the webpage activity detection, the 'Smart display' will analyze user preference (use frequency) about the 'click-able icons' and display the most frequent used ones for specific webpages to increase convenience.

### **Click-able icon**

New tab icons could be clicked by cursor and put on the browser tabs, the icon functions are mapping to the 'actual functions' provided by the webpage.

### **Notify tab**

New browser tabs that will notify users of all kinds of events and messages by changing their colors.

### **Customized 'Click-able icon' and 'Notify tab'**

Our design allows users to customize the new icons and the new tabs mentioned above.

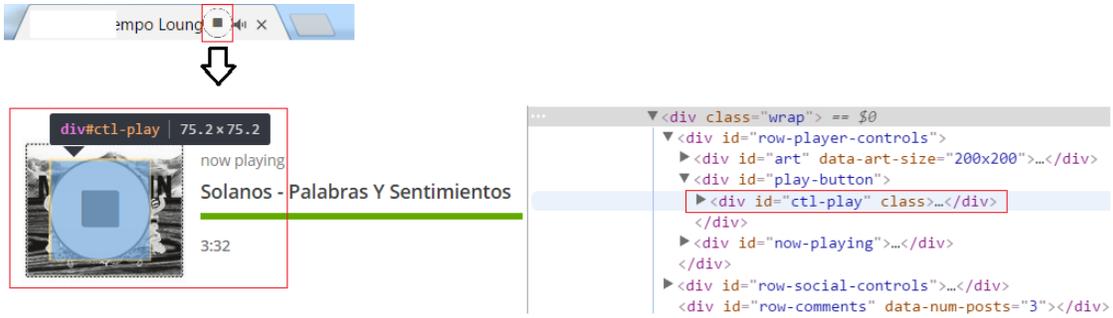
### **Smart display**

Display the right icons at the right time. Dynamically switch the displayed icons on demand.

There are three parts will need to be done for this implementation.

User-Interface: There are two possible UI providers from web-browser and web-browser plugin/extension. The UI software will listen the web events and possible user inputs to link to the user defined notifications and server defined clickable icon.

Step 1: link click-able icon on the tab to the action icon in the web browser.



Step 2: allow user define events with colors and icons.

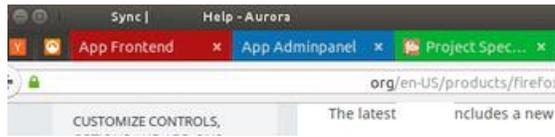


Fig. 1

Server side: The server monitors user selections to help UI software pop-up the correct icons on the tab and able to change the icons dynamically with different events.

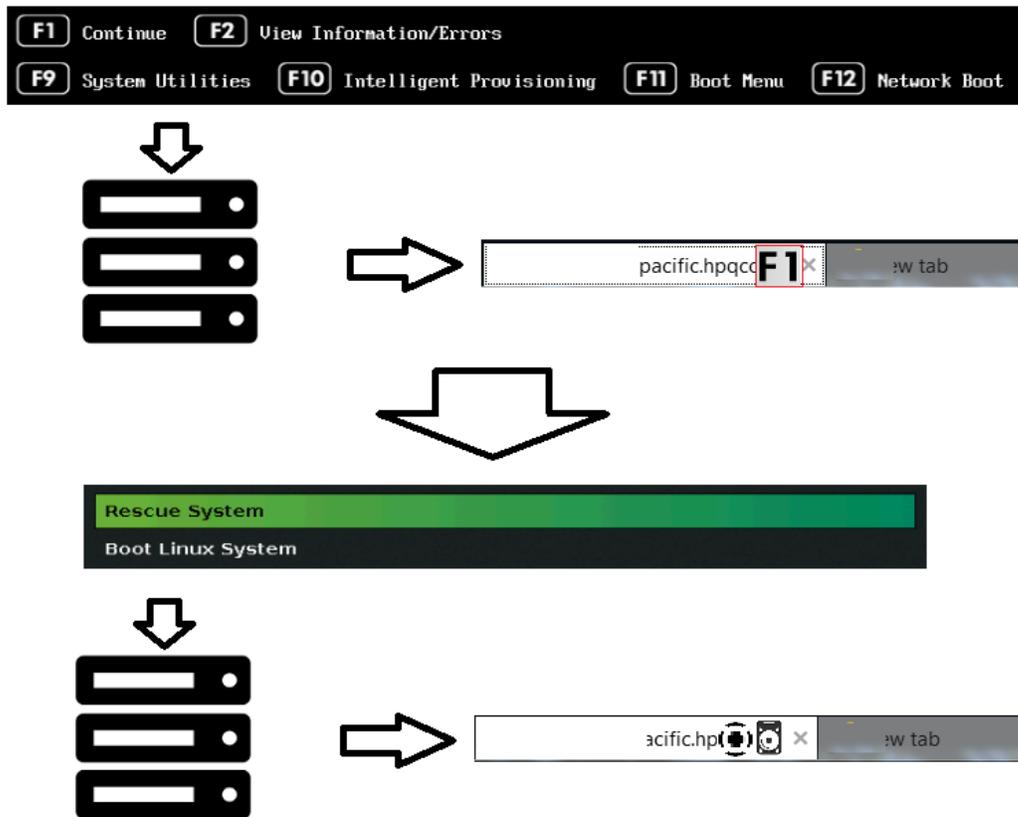


Fig. 2

Client side: User can define the notification methods to specific web-events with prefer colors and icons. After that, users directly clicks the icon on the tab to response the web events.

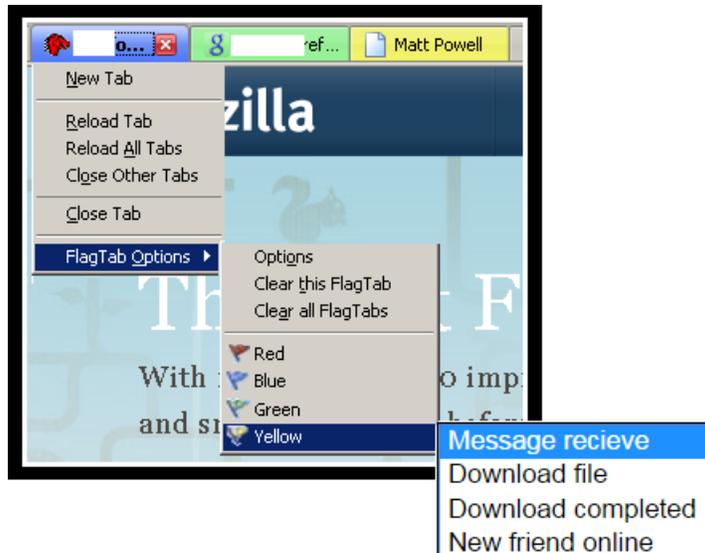


Fig. 3

### **Sample Use cases**

Users do the virus scan on multiple remote console:

A user opens a web browser to connect to multiple remote computers in different tabs and do the virus scan. The last tab is activate in the web browser.

The computer on the first tab found a virus and need the user do the reaction. The tab's color will become red and the two click-able icons will show on the tab. One is for skip and the other one is for delete.

The computer on the second tab is complete without any problems. One click-able icon will be shown on the tab for the user confirmation.

After the user delete the virus in the computer on first tab, the computer also complete the scan without any other problem. One click-able icon is shown on the tab for the user confirmation.

System admin try to boot up one system remotely and listen the music.

One tab is playing a music collection and there are three click-able icons for last music, next music, and stop music.

The other tab for boot up the system shows two click-able icons for boot into HD and into BIOS. After the user click the boot into HD icon, the icons become windows and Linux log for use to choose which OS that he/she wants to use in this time.

Disclosed by Darel Huang, Andy Liang and Nate Wu, Hewlett Packard Enterprise