

## User Interface - Bug #3962

### To handle an application shortcut if there is a browser's default action assigned to the same shortcut

03/15/2019 07:17 AM - Sergey Ivanovskiy

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

#### History

##### #1 - 03/15/2019 07:22 AM - Sergey Ivanovskiy

CTRL+F4 shortcut closes the current web tab in the browser. Thus there is an issue if the same shortcut is used by the converted application. For Firefox the pinned tabs can help to handle this particular case if CTRL+F4 or CTRL+W keys are pressed.

<https://support.mozilla.org/en-US/kb/pinned-tabs-keep-favorite-websites-open>

##### #2 - 03/15/2019 07:57 AM - Greg Shah

This is the list of Chrome shortcuts.

<https://support.google.com/chrome/answer/157179?hl=en>

I don't know which of these is treated in this "cannot be seen by javascript" manner.

##### #3 - 03/15/2019 09:35 AM - Sergey Ivanovskiy

The web client is tried to disable all default actions except the COPY/CUT/PASTE but it doesn't work for some shortcuts.

In p2j.keyboard.js we define this private method permittedKeyStrokes of this GUIKeyboardReader class to check if a current keyboard event must be consumed or not

```
/**
 * Tests if the keys strokes can invoke the permitted default actions: cut (shift + del,
 * ctrl + x), copy (ctrl + ins, ctrl + c) and paste (shift + ins, ctrl + v). For Safari,
 * pressing COMMAND + C, COMMAND + X and COMMAND + V can invoke the system clipboard copy,
 * cut and paste actions.
 *
 * @param {Event} evt
 *       The merged key stroke event that represents two pressed keys: a modifier and
 *       a standard key.
 *
 * @return True if the keys strokes are from this set: shift + del, ctrl + x, ctrl+ins,
 *         ctrl + c, shift + ins and ctrl + v, otherwise false.
 */
function permittedKeyStrokes(evt)
{
    if (p2j.isSafari)
    {
        if (evt.metaKey &&
            (evt.keyCode === keys.X || evt.keyCode === keys.C || evt.keyCode === keys.V))
        {
            return true;
        }
    }
}
```

```
    }
  }
  else
  {
    if (evt.shiftKey && (evt.keyCode === keys.DELETE || evt.keyCode === keys.INSERT))
    {
      return true;
    }
    else if ((evt.ctrlKey && (evt.keyCode === keys.X || evt.keyCode === keys.C ||
      evt.keyCode === keys.V || evt.keyCode === keys.INSERT))
    {
      return true;
    }
  }
}

return false;
}
```

It seems that there is no another common way. For an example, for IE we use this code in p2j.js module

```
if ("onhelp" in window)
{
  // suppress the browser help window in IE
  window.onhelp = function ()
  {
    return false;
  }
}
```

to suppress F1 default action.

**#4 - 03/26/2019 04:54 AM - Sergey Ivanovskiy**

Another idea is to add a virtual keyboard in order to handle all these cases.

**#5 - 03/26/2019 07:07 AM - Greg Shah**

I like the virtual keyboard idea. It is a universal "safety net". We can have a button that brings it up.

We may also want to provide a way to easily "alias" key sequences. For example, we can allow the customer to configure an alias so that some key sequence (e.g. SHIFT-F7) in the web client will generate a target key sequence (e.g. CTRL-F4) in FWD. The idea is that this is faster than bringing up the virtual keyboard. If the user can re-train themselves, it may be a workable solution.