

User Interface - Bug #6513

Invalid key sequence response in FILL-IN widget inside Web client

06/13/2022 06:05 PM - Eugenie Lyzenko

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Sergey Ivanovskiy	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 06/13/2022 06:17 PM - Eugenie Lyzenko

The bug can be reproduced in any standalone testcase containing FILL-IN widget. But only within Web client.

To recreate start the Web client and rapidly press SHIFT + f + w keys. The expected feedback is FW but FILL-IN shows WF in web client.

The key sequence coming to JS is:

```
keydown (SHIFT)
keydown (F)
keydown (W)
keypress (W)
  emits W in fill-in
keyup (F)
  emits F in fill-in
keyup (W)
keyup (SHIFT)
```

And we have WF in FILL-IN instead of expected FW.

If we press simple f+w the keys coming to JS is:

```
keydown (f)
keypress (f)
  emits f in fill-in
keydown (w)
keypress (w)
  emits w in fill-in
keyup (f)
keyup (w)
```

In this case the text is fw and this is expected.

What I think is we have to emit the key sending from JS to Java within keydown instead of keypress or keyup. The onkeydown callback always keep the proper sequence in web browsers.

Any suggestions why we do not emit the keys on keydown?

#2 - 06/14/2022 05:44 AM - Sergey Ivanovskiy

- File testKey.html added

I was trying to reproduce this sequence of key events independently and failed to reproduce it. In Firefox if the SHIFT + F was pressed first followed by W and all these keys were pressed rapidly with SHIFT was hold on.

```
keydown keyCode=16      which=16      charCode=0      AltGraph=false
  shiftKey=false ctrlKey=false altKey=false metaKey=false
  key=Shift keyValue=83 char=undefined code=ShiftLeft location=1 repeat=false
keydown keyCode=70 (F)  which=70 (F)   charCode=0      AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=F keyValue=70 char=undefined code=KeyF location=0 repeat=false
keypress keyCode=70 (F)  which=70 (F)   charCode=70 (F) AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=F keyValue=70 char=undefined code=KeyF location=0 repeat=false
Fkeydown  keyCode=87 (W)  which=87 (W)   charCode=0      AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=W keyValue=87 char=undefined code=KeyW location=0 repeat=false
keypress keyCode=87 (W)  which=87 (W)   charCode=87 (W) AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=W keyValue=87 char=undefined code=KeyW location=0 repeat=false
Wkeyup   keyCode=70 (F)  which=70 (F)   charCode=0      AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=F keyValue=70 char=undefined code=KeyF location=0 repeat=false
keyup    keyCode=87 (W)  which=87 (W)   charCode=0      AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=W keyValue=87 char=undefined code=KeyW location=0 repeat=false
keyup    keyCode=16      which=16      charCode=0      AltGraph=false
  shiftKey=true  ctrlKey=false altKey=false metaKey=false
  key=Shift keyValue=83 char=undefined code=ShiftLeft location=1 repeat=false
```

#3 - 06/14/2022 05:47 AM - Sergey Ivanovskiy

All these keys were pressed rapidly with SHIFT was hold on.

#4 - 06/14/2022 05:56 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Any suggestions why we do not emit the keys on keydown?

We emit keys on keypress if it occurs and on keyup if it doesn't happen.

#5 - 06/14/2022 08:04 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

I was trying to reproduce this sequence of key events independently and failed to reproduce it. In Firefox if the SHIFT + F was pressed first followed by W and all these keys were pressed rapidly with SHIFT was hold on.

Try Chrome.

#6 - 06/14/2022 08:07 AM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

All these keys were pressed rapidly with SHIFT was hold on.

You need to press SHIFT+f+w very fast, to have f down when pressing w.

#7 - 06/14/2022 02:50 PM - Eugenie Lyzenko

- % Done changed from 0 to 100
- Status changed from New to WIP

The 3821c updated for review to revision 13975.

This is the fix for original issue. The idea is to restore the same key handling sequence for alphabetic characters with SHIFT key pressed to the same route as used when SHIFT key is not holding down.

If no objections can be put into testing.

#8 - 06/14/2022 02:50 PM - Eugenie Lyzenko

- Status changed from WIP to Review

#9 - 06/14/2022 03:31 PM - Greg Shah

- Start date deleted (06/13/2022)

Sergey: Please review.

#10 - 06/15/2022 08:49 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Sergey Ivanovskiy wrote:

All these keys were pressed rapidly with SHIFT was hold on.

You need to press SHIFT+f+w very fast, to have f down when pressing w.

I cannot reproduce this issue with Google Chrome too. I think that real events observed by jsript keydown, keypress and keyup always follow the way how keys are pressed.

In Chrome

```
keydown  keyCode=70 (F)  which=70 (F)  charCode=0      AltGraph=false
         shiftKey=false ctrlKey=false altKey=false metaKey=false
         key=f keyValue=102 char=undefined code=KeyF location=0 repeat=false
keypress keyCode=102 (f)  which=102 (f)  charCode=102 (f)  AltGraph=false
         shiftKey=false ctrlKey=false altKey=false metaKey=false
         key=f keyValue=102 char=undefined code=KeyF location=0 repeat=false
textInput data=f
fkeydown  keyCode=16      which=16      charCode=0      AltGraph=false
         shiftKey=true  ctrlKey=false altKey=false metaKey=false
         key=Shift keyValue=83 char=undefined code=ShiftLeft location=1 repeat=false
keydown  keyCode=87 (W)  which=87 (W)  charCode=0      AltGraph=false
         shiftKey=true  ctrlKey=false altKey=false metaKey=false
         key=W keyValue=87 char=undefined code=KeyW location=0 repeat=false
keypress keyCode=87 (W)  which=87 (W)  charCode=87 (W)  AltGraph=false
         shiftKey=true  ctrlKey=false altKey=false metaKey=false
         key=W keyValue=87 char=undefined code=KeyW location=0 repeat=false
textInput data=W
wkeyup   keyCode=70 (F)  which=70 (F)  charCode=0      AltGraph=false
         shiftKey=true  ctrlKey=false altKey=false metaKey=false
         key=F keyValue=70 char=undefined code=KeyF location=0 repeat=false
keyup    keyCode=87 (W)  which=87 (W)  charCode=0      AltGraph=false
         shiftKey=true  ctrlKey=false altKey=false metaKey=false
         key=W keyValue=87 char=undefined code=KeyW location=0 repeat=false
keyup    keyCode=16      which=16      charCode=0      AltGraph=false
         shiftKey=false ctrlKey=false altKey=false metaKey=false
         key=Shift keyValue=83 char=undefined code=ShiftLeft location=1 repeat=false
```

#11 - 06/15/2022 08:57 AM - Greg Shah

Sergey: Have you reviewed Eugenie's change?

#12 - 06/15/2022 09:03 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

The 3821c updated for review to revision 13975.

This is the fix for original issue. The idea is to restore the same key handling sequence for alphabetic characters with SHIFT key pressed to the same route as used when SHIFT key is not holding down.

If no objections can be put into testing.

The keyboard test don't show that these changes are incorrect but from these changes it follows that the last event in my case keyF that was merged with SHIFT key is merged with keyW and finally it becomes W in the queue. It is incorrect and very confusing. We need to understand the root cause of this issue. The fast key pressing using another keyboard can produce different results according to my experience with Dutch(Netherlands) OS settings.

#13 - 06/15/2022 06:46 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie Lyzenko wrote:

The 3821c updated for review to revision 13975.

This is the fix for original issue. The idea is to restore the same key handling sequence for alphabetic characters with SHIFT key pressed to the same route as used when SHIFT key is not holding down.

If no objections can be put into testing.

The keyboard test don't show that these changes are incorrect but from these changes it follows that the last event in my case keyF that was merged with SHIFT key is merged with keyW and finally it becomes W in the queue. It is incorrect and very confusing. We need to understand the root cause of this issue.

I have reviewed the original code. And the scenario you described above is happening **before** my recent fix. And this is the where the root cause is. Take a look at the code in p2j.keyboard.js (with active issue before the fix):

```
function onkeydown(evt, allowDefaultAction)
{
...
    var last = queue.peek();
    if (last === undefined)
    {
...
    }
    else
```

```

    {
      if (isModifier(last.keyCode) || last.key == "AltGraph")
      {
        mergeWithLastEvent(last, evt);
        var permitted = allowDefaultAction || permittedKeyStrokes(last);
        if (!permitted)
        {
          p2j.consumeEvent(evt);
        }
      }

      return permitted;
    }
  }
  ...

```

When the sequence SHIFT + f + w is pressed rapidly, the SHIFT is pushed to queue, then f is merged with top key in queue ("Shift") so the queue has shifted f key at this time. Then w is coming and it merged again with top entry of queue so the queue has only single shifted w. So when the onkeypress is coming the JS finds only shifted w. So the shifted f become dropped, and have no onkeypress. That's why the shifted f key event emitted only when onkeyup is coming. And when we emit key event on key up - it is too late to have correct key sequence. This is the root cause.

My fix is based on idea to disable key merging for SHIFT key combination with further alphabetical keys. I think there should be no Shift+(a-z) shortcuts or key accelerators so we do not loose any functionality. All shortcuts usually stated with Ctrl key. The only exception I know is Shift+Ins but in this case Ins is not alphabetic key. So I do not expect the regression with the fix. If you know shortcut I missed - please let me know.

#14 - 06/17/2022 11:04 AM - Eugenie Lyzenko

I have observed the possible alternatives for the fix and found we can not avoid the logic currently used in fix.

I do not see any regression with the current version. Please let me know if you have (with recreation scenario) and I'll take a look. Otherwise we can consider the issue has been fixed.

#15 - 06/17/2022 03:54 PM - Eugenie Lyzenko

The other possible fix version for p2j.keyboard.js is:

```

function onkeydown(evt, allowDefaultAction)
{
  ...
  var last = queue.peek();
  -   if (last === undefined)
  +   if (last === undefined || (last.shiftKey && isAlphabeticCharacter(evt)))
  {
    last = {
      key      : key,
      code     : code,

```

```

        keyCode : keyCode,
        shiftKey : evt.shiftKey,
        altKey : evt.altKey,
        ctrlKey : evt.ctrlKey,
        metaKey : evt.metaKey,
        charCode : evt.charCode,
        ctrlCount : (evt.ctrlKey || keyCode === keys.CTRL) ? 1 : 0,
        location : evt.location
    };
}
else
{
    if (isModifier(last.keyCode) || last.key == "AltGraph")
    {
        mergeWithLastEvent(last, evt);
        var permitted = allowDefaultAction || permittedKeyStrokes(last);
        if (!permitted)
        {
            p2j.consumeEvent(evt);
        }

        return permitted;
    }
    else if (last.keyCode === keyCode || keyCode === keys.IME_PROCESSKEY)
    {
        //console.debug(last.ctrlCount);
        var repeatCode = last.keyCode;
        // auto-repeats that have no typed keys.
        last = {
            key : key,
            code : code,
            keyCode : repeatCode,
            shiftKey : evt.shiftKey,
            altKey : evt.altKey,
            ctrlKey : evt.ctrlKey,
            metaKey : evt.metaKey,
            charCode : evt.charCode,
            ctrlCount : last.ctrlCount,
            mergedKeys : last.mergedKeys,
            mergedCodes : last.mergedCodes,
            mergedKeyCodes : last.mergedKeyCodes,
            location : last.location
        };

        if (repeatCode !== keys.IME_PROCESSKEY)
        {
            queue.push(last);
            if (keyCode === keys.IME_PROCESSKEY)
            {
                var keyUpEvt = p2j.createKeyEvent("keyup", false, true, null, evt.ctrlKey,
                    evt.altKey, evt.shiftKey, evt.metaKey, repeatCode, 0, key);

                // check if the keyboard event is created successfully
                if (keyUpEvt)
                {
                    onkeyup(keyUpEvt);
                }

                p2j.consumeEvent(evt);
                return;
            }
            else
            {
                onkeyup(evt);
            }
        }

        var permitted = allowDefaultAction || permittedKeyStrokes(last);
        if (!permitted)
        {
            p2j.consumeEvent(evt);
        }
        return permitted;
    }
    else

```

```

    {
      last = {
        key      : key,
        code     : code,
        keyCode  : keyCode,
        shiftKey : evt.shiftKey,
        altKey   : evt.altKey,
        ctrlKey  : evt.ctrlKey,
        metaKey  : evt.metaKey,
        charCode : evt.charCode,
        ctrlCount : (evt.ctrlKey || keyCode === keys.CTRL) ? 1 : 0,
        location : last.location
      };
    }
  }
  queue.push(last);
...

```

This is more logically clean but actually the same as the current fix. So I doubt if I need to commit this.

#16 - 06/17/2022 04:14 PM - Sergey Ivanovskiy

Yes, I think you should not this new fix because last event was allocated if there is no such event in the events queue.

#17 - 06/17/2022 04:19 PM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Sergey Ivanovskiy wrote:

Eugenie Lyzenko wrote:

The 3821c updated for review to revision 13975.

This is the fix for original issue. The idea is to restore the same key handling sequence for alphabetic characters with SHIFT key pressed to the same route as used when SHIFT key is not holding down.

If no objections can be put into testing.

The keyboard test don't show that these changes are incorrect but from these changes it follows that the last event in my case keyF that was merged with SHIFT key is merged with keyW and finally it becomes W in the queue. It is incorrect and very confusing. We need to understand the root cause of this issue.

I have reviewed the original code. And the scenario you described above is happening **before** my recent fix. And this is the where the root cause

is. Take a look at the code in p2j.keyboard.js (with active issue before the fix):
[...]

When the sequence SHIFT + f + w is pressed rapidly, the SHIFT is pushed to queue, then f is merged with top key in queue ("Shift") so the queue has shifted f key at this time. Then w is coming and it merged again with top entry of queue

It is not correct. It should not be merged. For the original version of the code only modifier keys are merged with next pressed keys.

so the queue has only single shifted w. So when the onkeypress is coming the JS finds only shifted w. So the shifted f become dropped, and have no onkeypress. That's why the shifted f key event emitted only when onkeyup is coming. And when we emit key event on key up - it is too late to have correct key sequence. This is the root cause.

#18 - 06/17/2022 04:35 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie Lyzenko wrote:

Sergey Ivanovskiy wrote:

Eugenie Lyzenko wrote:

The 3821c updated for review to revision 13975.

This is the fix for original issue. The idea is to restore the same key handling sequence for alphabetic characters with SHIFT key pressed to the same route as used when SHIFT key is not holding down.

If no objections can be put into testing.

The keyboard test don't show that these changes are incorrect but from these changes it follows that the last event in my case keyF that was merged with SHIFT key is merged with keyW and finally it becomes W in the queue. It is incorrect and very confusing. We need to understand the root cause of this issue.

I have reviewed the original code. And the scenario you described above is happening **before** my recent fix. And this is the where the root cause is. Take a look at the code in p2j.keyboard.js (with active issue before the fix):
[...]

When the sequence SHIFT + f + w is pressed rapidly, the SHIFT is pushed to queue, then f is merged with top key in queue ("Shift") so the queue has shifted f key at this time. Then w is coming and it merged again with top entry of queue

It is not correct. It should not be merged. For the original version of the code only modifier keys are merged with next pressed keys.

It is not correct but it happen before rev 13975. The fix restores the correct key sequence, this is the idea of the change.

#19 - 06/20/2022 04:53 AM - Sergey Ivanovskiy

I don't understand now why this condition is not enough. It should return true only if a pressed key is SHIFT, ALT, CONTROL or AltGr

```
function isModifier(key)
{
    return key === keys.SHIFT ||
           key === keys.CTRL  ||
           key === keys.ALT   ||
           key === keys.LEFT_WINDOW || key === keys.RIGHT_WINDOW;
}
```

What is incorrect in this condition?

```
.....
    if ((isModifier(last.keyCode) || last.key == "AltGraph"))
    {
        mergeWithLastEvent(last, evt);
        var permitted = allowDefaultAction || permittedKeyStrokes(last) ||
                        (last.shiftKey && isAlphabeticCharacter(last));
        if (!permitted)
        {
            p2j.consumeEvent(evt);
        }
    }

    return permitted;
}
.....
```

It looks that I missed something.

#21 - 07/12/2022 10:32 AM - Greg Shah

- Assignee changed from Eugenie Lyzenko to Sergey Ivanovskiy

#22 - 07/12/2022 02:24 PM - Sergey Ivanovskiy

I cannot reproduce this issue with the current version rev. 14057 (3821c). Eugenie, could approve/validate that this issue is not reproduced with the current version on your system?

#23 - 07/12/2022 04:47 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

I cannot reproduce this issue with the current version rev. 14057 (3821c). Eugenie, could approve/validate that this issue is not reproduced with the current version on your system?

Yes, on my system (14058) the original issue is fixed.

#24 - 07/13/2022 06:36 AM - Sergey Ivanovskiy

Should we change the status of this task?

#25 - 07/13/2022 06:49 AM - Greg Shah

Can we close it?

#26 - 07/13/2022 08:37 AM - Sergey Ivanovskiy

Yes.

#27 - 07/13/2022 08:53 AM - Greg Shah

- Status changed from Review to Closed

Files

testKey.html	6.46 KB	06/14/2022	Sergey Ivanovskiy
--------------	---------	------------	-------------------