W3C DOM3 Events
Teleconf - 4 Mar 2014

W3C Bug List:
https://www.w3.org/Bugs/Public/buglist.cgi?product=WebAppsWG&component=DOM3%20Events&resolution=---&list_id=21979

This document is closed for editing. This discussion continues in the 15 April 2014 teleconf document:
https://docs.google.com/document/d/1mRTqY-2la5keSHhJUtEvSG3brMe_W3KRlkLWFVbV9yM/edit

Previous discussion was recorded in the 18 Feb 2014 teleconf document:
https://docs.google.com/document/d/1F12YUgO4hda5MYQK0NozgN_ceDdtvMKg7RvqJzci3y0/edit

23167 (PENDING)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23167
Define key name for TV

- garykac: TV is one of many mobile keys that we need a section for. We'd like it to be generic.
- garykac: If we can get a list soon, we should add it. If we can't get a list together soon, then we'll need to create an addendum doc that summarizes the mobile keys.
- garykac: We should spin up the addendum spec now so this has a place to live (not in UIEvents)
- travis: will follow up on best practice for this...

23697 (TRAVIS)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23697
KeyboardEventInit in UI Events doesn't include keyCode/charCode/which but in D3E, it includes them

- garykac: Should we include charCode, keyCode and which in the constructor?
    - masayuki: If web developers need to initialize charCode, keyCode or which, they need to call initKeyEvent() for Firefox, initKeyboardEvent() for other browsers (?). So, I think if they cannot be initialized with constructor, it's inconvenient for some web developers.
    - garykac: agree. I think they should be in the spec, but we should include them in Appendix B.1, which is where we discuss these attributes.
    - masayuki: I also think that if web browsers don't support the attribute, they don't need to support them with constructor.
    - garykac: right, and putting the info in the "legacy" key event Appendix means that

they won't be required.

- garykac: I think this info should be in Appendix "B.1 Legacy KeyboardEvent supplemental interface"

23748 (TRAVIS)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23748
Shouldn't getModifierState() definition refer Modifiers Keys which are defined in .key value?

- garykac: Spec should be updated to refer to the modifier key table instead of listing them here.
- garykac: Recommend update spec as suggested

23750 (TRAVIS)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23750
ScrollLock key should be defined in 6.3.2 Modifier Keys

- garykac: I can make arguments for/against this change. Not sure if it's worth adding there.
- Travis: make sense to have scroll-lock as a modifier.

23751 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23751
"Select" key is not defined in the latest ED

- garykac: Adding this key won't cause any problems, and it is defined VK_SELECT
- garykac: The bug also mentions adding Help, Stop, Again, Props, Undo, Front, Copy, Open, Paste, Find, Cut. This sounds good to me. I'm not sure what section to put them in yet though.
- masayuki: Ah, I forgot to file a bug for defining .code values for them. I'll do it soon.
- garykac: bug with additional keys is 24739

23886 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23886
Needs to define key value when dead key is pressed twice

- garykac: Review example #34 (second example in section "6.4.3 Dead keys")
- garykac: What does DeadCircumflex + q generate on all platforms? Is it DeadCircumflex + 'q' or is it '^' + 'q'?

23906 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23906
[proposal] Add (Mouse|Keyboard)Event.accelKey

- garykac: This bug describes our proposed solution (from the meeting in Tokyo) to the problem mentioned in 22703.
- garykac: should be called the 'osKey'

23908 (GARY)

https://www.w3.org/Bugs/Public/show_bug.cgi?id=23908
What's the good key value of printable keys with Meta key (Command key) on Mac?

- garykac: Masayuki's suggestions here in the bug sound reasonable (and this is an OS idiosyncrasy).
- garykac: We'll need to take care with writing our description to make sure it isn't confusing.
- travis: agree

23909
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23909
Which is expected whether keydown event is fired before or after native IME handles it

- garykac: Not sure if this is out-of-scope or not (ie: implementation specific)
- garykac: Does "5.2.7.5 Key Events During Composition" need to be expanded?
- masayuki: If keydown event is fired before compositionstart, web developers cannot know if the key event causes composition. However, consuming keydown event can prevent following compositionstart.
- masayuki: Otherwise, i.e., keydown event is fired after IME handles the key event, it depends on the platform if keydown event is fired or not, the order of keydown and compositionstart and it's possible to cancel compositionstart by consuming keydown event.
    - garykac In section 5.2.7.4 Canceling Composition Events, we say that canceling the composition may not always succeed. So we already have potential platform differences.
- masayuki: If .key value is changed by IME (bug 23910), the keydown event's .key value may depend on platform or IME.
- masayuki: If we will take the suggestion of bug 23910 (KeyboardEvent.isComposing), first keydown event's value depends on this bug.

23910 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23910
What's the good .key value while IME is open?

- garykac: Requests example of |key| for IME events.
- garykac: Are the examples #35 and #36 in section 6.4.4 not sufficient?
    - masayuki: it's not enough for Kana input mode of Japanese IME and Korean IME. I'm not sure Chinese IME.
    - FYI: Some Japanese IME changes the keyboard layout to Kana-input layout actually, but the others does not change the keyboard layout. Insted of that, they map every key input to their own Kana input layout.
- garykac: What would be a good example to add? (I don't follow the 's' and 'と' example in the bug). Ah, I think I understand now: 'a' = 'ち', 'd' = 'し'
    - masayuki: yes, you understand correctly.
- garykac: OK. This sounds like a good example to include then. I'd rather just pass the value from the OS along than having to calculate something.

    - masayuki: that may cause different behavior between Roman input mode and Kana input mode. E.g., if a web app wants to handle ‘c’ while editor doesn’t have focus, it also hadles ‘そ’ for Kana input mode. Because e.g., Kotoeri changes the keyboard layout, but ATOK doesn’t do it. I.e., in this case, only Kotoeri user cannot access the ‘c’ key handler if the web app doesn’t handle ‘そ’.
    - garykac: Ugh. So it sounds like we don’t want to explicitly require a particular value there, but rather say that it is OS/IME specific.
    - masayuki: On Mac OS X, Firefox referes ASCII capable keyboard layout for computing .key value. On Windows, Firefox ignores Kana Lock state for doing that.
    - garykac: So FF always returns ‘そ’ for |key| on either platform (never ‘c’)?
    - masayuki: No, always return ‘c’. never returns ‘そ’.
    - garykac: I’m fine with that, but how do you know when it is safe to apply this transform? Couldn’t another keyboard layout require a different mapping? Or do you know that whenever you see ‘そ’ that it is that layout so you can safely apply the mapping?
    - garykac: Also would that mapping apply for mobile (where you might only see the kana)?
    - masayuki: Kana input mode is minority in Japan, therefore, even if web apps are made by Japanese people, they usually don’t test Kana input mode. Therefore, we found this problem actually and we changed our behavior for Kana input mode users :-(
    - garykac: ^_^ That makes sense. We should make sure the spec doesn’t preclude that behavior since it sounds useful. I’m wondering if it should be *required* in the spec - it seems very specific and I don’t want to bloat the spec.
    - masayuki: Note that compositionupdate will tells actual input character if composition actually occurs.
    - garykac: True. People should be looking at the composition events if they want the real char intended by the user.
    - masayuki: yes. on mobile, I’m not sure the platform limitation because I’ve never found IMEs which support Kana input mode with external USB keyboard on Android. I recommend that the Spec should append note for both implementers and authors about this issue.
    - garykac: sounds good to me.
    - garykac: I have to head out now, but please continue making comments and I’ll look over them tomorrow. thanks!

23911 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23911
[Proposal] Add KeyboardEvent.isComposing attribute

- garykac: Recommend moving to UI Events unless this is blocking something significant
- masayuki: This is very simple even if we take since it should true only between compositionstart and compositionend. And this is useful for Firefox’s internal

implementation when Firefox will fire key events even during composition as D3E defined. Therefore, I'd like this is fixed in D3E.
- Travis: Agree--do it!

23913 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=23913
beforeinput should be fired only when the DOM change is caused by direct input by keypress or composition
- garykac: Not sure if there is consensus or not in the bug.
- garykac: Ojan suggests that beforeinput/input shouldn't fire for execCommand
- masayuki: beforeinput and input are really different idea. input is useful for catching every change of the editor after actually modified. beforeinput is an alternative event of keypress and it's cancelable. Therefore, I believe that there is no reason why they are paired always. (except the similar name)
- masayuki: Gecko doesn't fire DOM event synchronously if "script blocker" is set internally. E.g., it's set while <input>.value is being set. I'm not sure the detail of the reason, though. However, the comments in the tree document about security. In such case, it's impossible to fire a "cancelable" beforeinput event.
- masayuki: beforeinput is defined for an alternative event of keypress, it should be fired only when changes are caused by keypress. If it's useful beforeinput is fired immediately before compositionend (and after compositionupdate?), it's okay because it doesn't have any troubles.
- masayuki: Anyway, I strongly hope that D3E spec documents beforeinput isn't always paired with input event. If it's not good name for this, beforeinput should be renamed, e.g., willinserttext? Then, we don't need to add .data attribute to input event. The attribute isn't useful at deleting character by Backspace or Delete. Additionally, if such name, the new event isn't needed to be fired at deleting character.
- garykac: beforeinput is intended to fire before any input event so that it can potentially be canceled. It is not supposed to be restricted to only key events. This was a stated goal because there are accessibility and mobile scenarios (screenreaders, voice input, …) that we want to go through this 'catch-all' mechanism.
- garykac: We also recognize that canceling beforeinput may not (in some cases) be able to cancel the input. The spec should be written to say that cancelling beforeinput SHOULD prevent input.

24044 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=24044
How do browsers decide combining character from non-combining character at computing .key value of dead key?
- masayuki: E.g., on Windows, we can get non-combining character with WM_DEADCHAR message. I.e., if KeyboardEvent.key value of dead key should be combining character, browser vendors need a conversion table from non-conbining character to combining character for all cases! I think that it's impossible...

- masayuki: If the .key value of deadkey is always DeadKey, this can be marked as INVALID.
- garykac: related to 23886?

24247 (TRAVIS)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=24247
"guidelines for selecting and defining key values" is a dead link
- Question: should we remove the text that says people can define their own key values? There will be new |key| values from different layouts, but we should probably not encourage new special keys in the spec.
- garykac: It's worthwhile having the code saying how to determine |key| values, but I agree that we shouldn't explicitly encourage vendors to add new values.
- Travis: We should get rid of this. We don't want folks to mint their own. If they want new ones, they can be added to the annex.

24738 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=24738
D3E spec should define if keypress is fired when preceding beforeinput is consumed by preventDefault()
- garykac: Purposely left unspecified because this is deprecated. If all current implementation agree on the order, then we can add that info here, otherwise I'd just leave it.

24739 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=24739
Define code values for the special keys on Sun keyboard
- garykac: Need to verify that the Japanese keys are all physical keys. If so, we need to add them. I'd like to get a picture of this keyboard layout if possible.

24740 (GARY)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=24740
Define code values for the special keys on Mac keyboard
- garykac: F13-19 already covered. We'll need to add Eisu key. I need to double check Kana vs. KanaMode keys.
- garykac: picture of keyboard: http://s3.amazonaws.com/dotclue.org/kybard.jpg

24741 (TRAVIS)
https://www.w3.org/Bugs/Public/show_bug.cgi?id=24741
CustomEvent inherits from Event
- garykac: Looks like the spec should be updated, but need to double-check
- Per DOM4: interface CustomEvent : Event {...

24797 (TRAVIS)

https://www.w3.org/Bugs/Public/show_bug.cgi?id=24797
Stop using AbstractView

- garykac: Clearly a bug, needs to be fixed in the spec, but need to verify the correct fix:
    - garykac: Should the 'view' attribute be removed entirely? Or should the type simply be changed to 'Window'? Need to confirm.
    - Travis: I think we just need to change the type. AbstractView predates HTML5 Window definition.

==========
CLOSED
==========

~~22073~~
~~https://www.w3.org/Bugs/Public/show_bug.cgi?id=22073~~
~~'Meta' key is the same as the 'OS' or 'Win' key~~

- ~~garykac: We discussed this in Tokyo and Masayuki has created bug #23906 to expose the “accel” key to make this platform neutral. I’m not sure it’s worth keeping this bug around since we can use 23906 to track.~~
- ~~garykac: Recommend close this bug after adding a link to 23906~~

~~23186~~
~~https://www.w3.org/Bugs/Public/show_bug.cgi?id=23186~~
~~EventListener argument of addEventListener and removeEventListener should not be nullable~~

- ~~This is RESOLVED with the text update mentioned in the bug.~~
- ~~garykac: CLOSED.~~

~~23943~~
~~https://www.w3.org/Bugs/Public/show_bug.cgi?id=23943~~
~~It's too difficult to search the code which access KeyboardEvent.key~~

- ~~RESOLVED - See http://www.w3.org/2008/webapps/wiki/21-Jan-2014 for discussion (pasted into the bug).~~
- ~~garykac: It’s too late to make some of the suggested changes (attribute names) and we can’t add a prefix because most of the key values are simple Unicode chars.~~
- ~~garykac: CLOSED~~

~~24729~~
~~https://www.w3.org/Bugs/Public/show_bug.cgi?id=24729~~
~~WeelEvent inherits from MouseEvent, but WheelEventInit (UIevents) doesn't from MouseEventInit~~

- ~~garykac: clear bug in spec. it should be updated. FIXED~~