Bug 1269300

Need a way to invoke Print Preview from WebExtension add-on

Edit Bug

Stop Following

RESOLVED FIXED in Firefox 56 (NeedInfo from wbamberg)

Get help with this page

Status (bug RESOLVED as FIXED for Firefox 56 awaiting an answer on a request for information)

Product: Toolkit
Component: WebExtensions: General
Importance: P5 normal
Status: RESOLVED FIXED

Reported: a year ago
Modified: 4 minutes ago

People (Reporter: dw-dev, Assigned: dw-dev, Mentored, NeedInfo)

Assignee: dw-dev
Mentors: Shane Caraveo (:mixedpuppy)

Reporter: dw-dev
Triage Owner: Andy McKay [:andym]
NeedInfo From: wbamberg 22 minutes ago
CC: 12 people Add

Tracking ({dev-doc-needed})

Version: 45 Branch
Target: mozilla56
Keywords: dev-doc-needed
Points: ---

Duplicates: ~~1303733~~, ~~1355421~~

Firefox Tracking Flags (firefox56 fixed)

Details (Whiteboard: [design-decision-approved]triaged)

Whiteboard: [design-decision-approved]triaged
Votes: 0 votes Vote

See Also: ~~1355421~~

MozReview Requests ()

Submitter	Diff	Changes	Open Issues	Last Updated
dw-dev+495827	Bug 1269300 Patch 5;r?mixedpuppy	+337	0	15 days ago

Attachments (3 attachments, 4 obsolete attachments)

Print Edit WE 19.0b3 with calls to print(), printPreview() and saveAsPDF() Details
2 months ago dw-dev
74.73 KB, application/x-xpinstall

Bug 1269300 Patch 5; mixedpuppy : **review+** Details | Review
20 days ago dw-dev
59 bytes, text/x-review-board-request

saveaspdf.zip Details
14 hours ago Will Bamberg [:wbamberg]
2.30 KB, application/zip

Show Obsolete Attachments

Attach File

Add Comment ↓ Tags View

dw-dev (Assignee) Reply -
Description • a year ago

```
User Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Build ID: 20160407164938

Steps to reproduce:

I am developing a WebExtensions version of my 'Print Edit' add-on, but I cannot find any way of invoking
Firefox's normal Print Preview from the add-on.

A couple of years ago, I developed a version of 'Print Edit' for Chrome, which works very well.  So it is a
shame that I cannot do the same for Firefox.

The problem is that window.print() works differently in Firefox and Chrome/Opera.  In Firefox, window.print()
opens the Print dialog.  In Chrome/Opera, window.print() opens the Print Preview window.
```

Expected results:

A simple solution would be to provide a window.printPreview() function.

Alternatively, a window.doCommand() function would allow any standard Firefox command to be invoked, for example, window.doCommand("cmd_printPreview").

YF (Yang) ▾ -

Updated • a year ago

Component: Untriaged → WebExtensions
Product: Firefox → Toolkit

Andy McKay [:andym] ▾ -

Updated • a year ago

Whiteboard: [design-decision-needed]triaged

dw-dev ▾ (Assignee) Reply -

Comment 1 • a year ago

It has just occurred to me that an even simpler solution would be to change window.print() so that it opens the Print Preview window when called from a WebExtension content script.

This would provide full compatibility with Chrome/Opera, but would not affect existing XUL/XPCOM add-ons.

dw-dev ▾ (Assignee) Reply -

Comment 2 • a year ago

Comments 62 & 63 in ~~Bug 1189846~~ also discuss this issue.

~~https://bugzilla.mozilla.org/show_bug.cgi?id=1189846#c62~~

~~https://bugzilla.mozilla.org/show_bug.cgi?id=1189846#c63~~

Andy McKay [:andym] ▾ Reply -

Comment 3 • a year ago

Changing window.print() within an extension to do something different from window.print() within a web page is not a good idea in my opinion (comment 1), developers should mostly expect Firefox to be compatible with well, itself.

Adding window.printPreview() is one option (or a property to the method), but that would require landing code in Firefox and that seems a harder option since it would have to be supported for a very long time and the standard and approvals needed is much higher.

Adding in some WebExtensions API to the browser.tabs or browser.window, or maybe even browser.print (a new API) would be doable and probably the easiest approach to reach your target.

I don't think this is a priority for others right now and you know a lot about this subject. Would the next step be proposing the WebExtensions API that makes sense to you?

dw-dev ▾ (Assignee) Reply -

Comment 4 • a year ago

To be honest, I don't fully understand how the WebExtensions APIs are being implemented in Firefox.

The Webextensions version of my Print Edit add-on needs one new API, that is the ability to invoke the existing "cmd_printPreview" command, which just makes the following function call:

```
   PrintUtils.printPreview(PrintPreviewListener);
```

where "PrintPreviewListener" is defined in browser/base/content/browser.js.

I would envisage the API to be browser.print.printPreview() without any paramaters. This API would enter print preview mode for the currently active tab.

I would favour adding to browser.windows over browser.tabs, since print preview effectively takes over the whole

window when invoked. However, I think a new browser.print might be better if there are going to be other print API calls.

How would I go about creating a new WebExtensions API that just needs the ability to invoke the existing "cmd_printPreview" command ?

Andy McKay [:andym] ▾ Reply -

Comment 5 • a year ago

Cool. There's a couple of extra wrinkles that come with a WebExtension, but in the end you'll just be calling printUtils and providing a few extra things like creating the schema.

If you haven't built Firefox before, you can learn how to get set up to to work on it here:

https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Build_Instructions/Simple_Firefox_build

Some details on hacking on WebExtensions is here:

https://wiki.mozilla.org/WebExtensions/Hacking

Probably the best way to go is to look at some other APIs, one example is:

https://dxr.mozilla.org/mozilla-central/source/browser/components/extensions/ext-history.js
https://dxr.mozilla.org/mozilla-central/source/browser/components/extensions/schemas/history.json

I've added Kris as a mentor, he can help you out with any more questions.

Mentor: kmaglione+bmo@mozilla.com

Tim Nguyen :ntim ▾ -

Updated • a year ago

Status: UNCONFIRMED → NEW
Ever confirmed: true

Andy McKay [:andym] ▾ -

Updated • a year ago

Whiteboard: [design-decision-needed]triaged → [design-decision-approved]triaged

dw-dev ▾ (Assignee) Reply -

Comment 6 • a year ago

Kris,

I have looked at the documents and source code referenced by Andy.

As far as I can see, the changes I need to make to implement a printPreview() API are fairly trivial, and the effort involved will be far outweighed by setting up the Firefox build environment!

I have a few questions:

1. As an example, to add a new interface to chrome.tabs, what else would I need to modify besides ext-tabs.js and tabs.json?

2. I understand I would have to add the new bespoke interface to browser.tabs rather than to chrome.tabs. How would I do that?

3. We may want to add the new interface to a new set of interfaces, for example, to browser.print rather than to browser.tabs. Who makes that decision? Would I just have to add two new files, ext-print.js and print.json?

4. I still have a feeling that it would be nice to have a generic interface that allows any Firefox standard command to be executed. For example, in my case, browser.runtime.executeCommand("cmd_printPreview"), which would just call window.doCommand("cmd_printPreview"). Is this idea worth pursuing?

Flags: needinfo?(kmaglione+bmo@mozilla.com)

Andy McKay [:andym] ▾ Reply -

Comment 7 • a year ago

> 4. I still have a feeling that it would be nice to have a generic interface
> that allows any Firefox standard command to be executed. For example, in my
> case, browser.runtime.executeCommand("cmd_printPreview"), which would just
> call window.doCommand("cmd_printPreview"). Is this idea worth pursuing?

I would caution against that, however it depends what you mean by "standard command". We are making sure that any API we write is suitable for extensions, conforms to security and privacy standards, works well with e10s and is async. Not everything in Firefox meets that.

dw-dev (Assignee)
Comment 8 • a year ago

Reply -

Andy,

Okay, I will focus on implementing just the printPreview() interface.

Are you able to answer my questions 1, 2 & 3 in Comment 6 ?

Flags: needinfo?(amckay@mozilla.com)

Kris Maglione [:kmag]
Comment 9 • a year ago

Reply -

(In reply to dw-dev from comment #6)
> As far as I can see, the changes I need to make to implement a
> printPreview() API are fairly trivial, and the effort involved will be far
> outweighed by setting up the Firefox build environment!

It's actually pretty easy to set up a build environment these days:
https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Build_Instructions/Artifact_builds

> 1. As an example, to add a new interface to chrome.tabs, what else would I
> need to modify besides ext-tabs.js and tabs.json?

Nothing else.

> 2. I understand I would have to add the new bespoke interface to
> browser.tabs rather than to chrome.tabs. How would I do that?

We don't have a way to do that yet. We can figure it out later.

> 3. We may want to add the new interface to a new set of interfaces, for
> example, to browser.print rather than to browser.tabs. Who makes that
> decision? Would I just have to add two new files, ext-print.js and
> print.json?

It depends. We'd have to discuss the merits of the different approaches.

> 4. I still have a feeling that it would be nice to have a generic interface
> that allows any Firefox standard command to be executed. For example, in my
> case, browser.runtime.executeCommand("cmd_printPreview"), which would just
> call window.doCommand("cmd_printPreview"). Is this idea worth pursuing?

No, this isn't a stable interface. Those commands are mostly for internal use.
They come and go and change regularly, and most of them have serious security
implications.

Flags: ~~needinfo?(kmaglione+bmo@mozilla.com)~~

dw-dev (Assignee)
Comment 10 • a year ago

Reply -

(In reply to Kris Maglione [:kmag] from comment #9)

> > 3. We may want to add the new interface to a new set of interfaces, for
> > example, to browser.print rather than to browser.tabs. Who makes that
> > decision? Would I just have to add two new files, ext-print.js and
> > print.json?
>
> It depends. We'd have to discuss the merits of the different approaches.

The options for the new interface seem to be chrome.tabs.printPreview(), chrome.windows.printPreview() or chrome.print.printPreview().

In chrome.tabs, every interface either takes a tab parameter or returns one or more tabs. In chrome.windows,

every interface either takes a window parameter or returns one or more windows. But PrintUtils.printPreview() always previews the selected tab (and creates a preview tab and optionally a simplify page tab). So I don't think it really fits with the other chrome.tab or chrome.windows interfaces.

Chrome has a chrome.printProvider set of interfaces, but I don't think printPreview() really fits.

If we go for a new chrome.print set of interfaces, I could envisage chrome.print.print() and chrome.print.pageSetup(), as well as chrome.print.printPreview. Another alternative would be a new chrome.custom set of interfaces, which could be used for specific custom interfaces required by add-ons, such as chrome.custom.printPreview().

What do you think?

Flags: needinfo?(kmaglione+bmo@mozilla.com)

Andy McKay [:andym] ▾ -

Updated • a year ago

Flags: ~~needinfo?(amckay@mozilla.com)~~

Kris Maglione [:kmag] ▾ -

Updated • 10 months ago

Duplicate of this bug: ~~1303733~~

r.chrzanowski ▾ Reply -

Comment 12 • 10 months ago

If we are talking about print API in WebExtensions I would like it to be possible to call both printing and print priview with ability to pass printSettings object (allowing to chose printer name, page size, margins, header/footer etc.). Just like it was possible to do with current extensions.

Andy McKay [:andym] ▾ -

Updated • 9 months ago

Component: WebExtensions: Untriaged → WebExtensions: General
Summary: WebExtensions: Need a way to invoke Print Preview from WebExtension add-on → Need a way to invoke Print Preview from WebExtension add-on

Andy McKay [:andym] ▾ -

Updated • 9 months ago

Priority: -- → P5

dw-dev ▾ (Assignee) Reply -

Comment 13 • 8 months ago

Over the past couple of months, I have been working on creating WebExtensions versions of my other Firefox add-ons (Navigate Up, Search Site, Tile Tabs & Zoom Page), so I have not been able to progress this bug.

Returning to look at this bug again, I think the right way forwards is to add a new 'browser.tabs.printPreview(callback)' interface.

To do this, in tabs.json, we would add the following code:

```
    {
      "name": "printPreview",
      "type": "function",
      "description": "Shows print preview for active tab.",
      "async": "callback",
      "parameters": [
        {
          "type": "function",
          "name": "callback",
          "parameters": [
            {
              "type": "boolean",
              "name": "status",
              "description": "Whether print preview was successful."
            }
          ]
        }
      ]
```

```
    },


And, in ext-tabs.js, we would add the following code:

    printPreview(callback) {
      let tab = TabManager.activeTab;
      let mm = tab.linkedBrowser.messageManager;

      let onEntered = (message) => {
        mm.removeMessageListener("Printing:PrintPreview:Entered", onEntered);
        callback(message.data.failed);
      }

      mm.addMessageListener("Printing:Preview:Entered", onEntered);

      PrintUtils.printPreview(PrintPreviewListener);
    },


Is it possible for someone familiar with the Firefox build and review process to apply these changes?
```

dw-dev (Assignee) Reply -
Comment 14 • 8 months ago

```
Please could you give me some feedback on the proposed code changes in Comment 13.

Would these code changes be sufficient and acceptable if I made them ?
```

Tomislav Jovanovic :zombie Reply -
Comment 15 • 8 months ago

```
(In reply to dw-dev from comment #14)
 > Would these code changes be sufficient and acceptable if I made them ?

This bug is flagged as [design-decision-approved], so we want to do this.  You are asking if your code is a good
fix for the bug, or in other words, you are asking for code review.  If you produce your code in the form of a
patch and flag for review?, you should get a faster response.
```

kernp25 Reply -
Comment 16 • 4 months ago

```
Can i work on this bug?
```

dw-dev (Assignee) Reply -
Comment 17 • 4 months ago

```
Thanks for your interest.  I would welcome your help or advice.

Two API functions are required to make Print Edit WE really useful and equivalent to my original Print Edit add-
on:

  - chrome.tabs.printPreview(callback)
  - chrome.tabs.saveAsPDF(pageSettings,callback)

I have already done a prototype implementation of both these functions in my local version of mozilla-central.
And they both work really well with a slightly modified version of Print Edit WE.

However, there are a couple of areas where I would welcome help or advice:

1. I have not been able to get the asynchronous callback functions working, probably because of my unfamiliarity
with promises. Having callbacks is not essential, but would it would be nice to pass back a success/failure
status.

2. The next step is to submit the changes for review and release, hopefully in time for Firefox 56, if not
Firefox 55.
However, I have no experience of using the Mozilla tools to uploading changes to Firefox for review and release.

I can upload the two changed files (tabs.json & ext-tabs.js) if you are interested.
```

kernp25 Reply -
Comment 18 • 4 months ago

(In reply to dw-dev from comment #17)
> Thanks for your interest. I would welcome your help or advice.
>
> Two API functions are required to make Print Edit WE really useful and
> equivalent to my original Print Edit add-on:
>
>   - chrome.tabs.printPreview(callback)
>   - chrome.tabs.saveAsPDF(pageSettings,callback)

We also need an api for calling window.print()

Here's why:

1) Setting the pref "dom.disable_window_print" disables the window.print function/method
2) Calling window.print a third time will display a dialog "Prevent this page from creating additional dialogs" if the user clicks ok (by mistake) then window.print is disabled. ([Exception... "Component is not available" nsresult: "0x80040111 (NS_ERROR_NOT_AVAILABLE)" location: "JS frame :: debugger eval code :: <TOP_LEVEL> :: line 1" data: no])

https://dxr.mozilla.org/mozilla-central/source/dom/base/nsGlobalWindow.cpp#7916

r.chrzanowski ▾ Reply -
Comment 19 • 4 months ago

Personnaly I'd love to see API that would allow to query for available printers and change print settings (printer, page size, margins, even silent printing -- or just pass them once with a single printing job).
I can see that is something that that could be abused (silently sending hundreds of pages to network printer or something) but that would allow for ex. batch printing (for ex. invoices) from web applications.

dw-dev ▾ (Assignee) Reply -
Comment 20 • 4 months ago

(In reply to kernp25 from comment #18)

> We also need an api for calling window.print()
>
> Here's why:
>
> 1) Setting the pref "dom.disable_window_print" disables the window.print
> function/method
> 2) Calling window.print a third time will display a dialog "Prevent this
> page from creating additional dialogs" if the user clicks ok (by mistake)
> then window.print is disabled. ([Exception... "Component is not available"
> nsresult: "0x80040111 (NS_ERROR_NOT_AVAILABLE)" location: "JS frame ::
> debugger eval code :: <TOP_LEVEL> :: line 1" data: no])

I understand points (1) & (2) and that in these cases window.print() is disabled.

This would only be an issue for Print Edit WE if the user prints the same page multiple times, using the Print item on the right-click menu of the Print Edit WE toolbar button. I assume you are thinking of wider applications. Is it a good idea to bypass the "dom.disable_window_print" pref?

dw-dev ▾ (Assignee) Reply -
Comment 21 • 4 months ago

(In reply to r.chrzanowski from comment #19)

> Personnaly I'd love to see API that would allow to query for available
> printers and change print settings (printer, page size, margins, even silent
> printing -- or just pass them once with a single printing job).

For the moment, I would like to focus on the API support needed for Print Edit WE, which was the original driver for this interface request. Down the track, I think it would make sense to have a pageSetup(pageSettings) function, and possibly a printer enumeration function.

kernp25 ▾ Reply -
Comment 22 • 3 months ago

(In reply to dw-dev from comment #20)

> (In reply to kernp25 from comment #18)
>
> > We also need an api for calling window.print()
> >
> > Here's why:
> >
> > 1) Setting the pref "dom.disable_window_print" disables the window.print
> > function/method
> > 2) Calling window.print a third time will display a dialog "Prevent this
> > page from creating additional dialogs" if the user clicks ok (by mistake)
> > then window.print is disabled. ([Exception... "Component is not available"
> > nsresult: "0x80040111 (NS_ERROR_NOT_AVAILABLE)" location: "JS frame ::
> > debugger eval code :: <TOP_LEVEL> :: line 1" data: no])
>
> I understand points (1) & (2) and that in these cases window.print() is
> disabled.
>
> Is it a good idea to bypass the "dom.disable_window_print"
> pref?

window.print() should always be available to webextensions.

Comment hidden (mozreview-request) +

Andy McKay [:andym] ▾ Reply -

Comment 24 • 3 months ago

Thanks for the patch dw-dev. I don't actually review patches for WebExtensions, but at a quick glance, it has zero tests and all the code inside WebExtensions has unit tests to ensure that it has tests and won't regress when something changes inside Firefox.

If you look in browser/components/extensions/test, you'll find the tests for tabs. We aim for 100% coverage on our tests if possible.

I don't think it would hurt to have Kris give it a quick look over though to make sure you are on the right path.

Andy McKay [:andym] ▾ -

Updated • 3 months ago

Attachment #8860457 - Flags: ~~review?(amckay@mozilla.com)~~

dw-dev ▾ (Assignee) Reply -

Comment 25 • 3 months ago

Kris, Andy,

As you can see, I have submitted a patch to add print(), printPreview() and saveAsPDF(pageSettings) functions to the chrome.tabs WebExtensions API's. I have tested these functions using my local build of Firefox and modified version of my Print Edit WE add-on.

The key design considerations were:

1. Whether chrome.tabs.print() and chrome.tabs.printPreview() should have a pageSettings parameter. This parameter is not necessary for Print Edit WE and I cannot see the need in other scenarios. Both print and print preview will use the current (default) printer page settings. The user will be interacting with the print preview window and can use the Page Setup dialog to change the page settings.

2. Whether chrome.tabs.print() is really necessary. Arguably this function is not necessary since window.print() could be used instead, but see Comment 18 & Comment 22 above.

3. Whether these functions should have callback function parameters. These are not really necessary for knowing when a print or print preview has completed, because the 'afterprint' event can be used. They might be useful for passing back a status (error) indicator, especially in the case of saveAsPDF() if the user cancels the 'Save As' dialog. To be honest, I tried adding callback functions, but could not get them to work correctly.

4. In the longer term, I would expect there to be a chrome.tabs.pageSetup(pageSettings) function added, which would set the page settings for the current (default) printer.

I welcome your thoughts on these issues.

dw-dev ▾ (Assignee) Reply -

Comment 26 • 3 months ago

(In reply to Andy McKay [:andym] from comment #24)

> Thanks for the patch dw-dev. I don't actually review patches for
> WebExtensions, but at a quick glance, it has zero tests and all the code
> inside WebExtensions has unit tests to ensure that it has tests and won't
> regress when something changes inside Firefox.
>
> If you look in browser/components/extensions/test, you'll find the tests for
> tabs. We aim for 100% coverage on our tests if possible.
>
> I don't think it would hurt to have Kris give it a quick look over though to
> make sure you are on the right path.

Andy, I was aware of the automated tests in browser/components/extensions/test, but I haven't done anything about tests for these new interfaces, for two reasons:

1. I need some feedback and agreement on the design of these new interfaces, before spending time on developing automated tests.

2. It is not obvious to me what tests could be automated for these new interfaces, other than checking for 'beforeprint' events. Ideally, for chrome.tabs.print() and chrome.tabs.printPreview() there would be a "dummy" printer with default print settings in the test environment, but I am not aware if that is the case. Also, chrome.tabs.saveAsPDF() requires user interaction with the "Save As" dialog, so I'm not sure how this could be tested.

I think it would be good to get input from Bob Owen who has done a lot of work on the core print, print preview and print as PDF code, regarding the possibilities for testing. I know there is an intention to improve the testing of the core print and print preview functionality.

Flags: needinfo?(bobowencode@gmail.com)

Bob Owen (:bobowen) ▾ Reply -
Comment 27 • 3 months ago

(In reply to dw-dev from comment #26)
> (In reply to Andy McKay [:andym] from comment #24)

> I think it would be good to get input from Bob Owen who has done a lot of
> work on the core print, print preview and print as PDF code, regarding the
> possibilities for testing. I know there is an intention to improve the
> testing of the core print and print preview functionality.

Unfortunately, I've still not had chance to work on any printing tests.
Someone else has got quite a long way with testing using PDF.js (bug 1299848), but I don't think anything has landed.

I don't know enough about web extensions to know how much testing could be done, to make sure the correct things are being called without actually printing.
I would have thought that print preview could be tested though.
I don't know the print preview side of things all that well, I think we have some tests that mimick and test the layout, but I'm not sure what we have that fully invokes print preview.

Flags: ~~needinfo?(bobowencode@gmail.com)~~

Andy McKay [:andym] ▾ -
Updated • 3 months ago

Attachment #8860457 - Flags: review?(kmaglione+bmo@mozilla.com) → review?(mixedpuppy@gmail.com)

Andy McKay [:andym] ▾ -
Updated • 3 months ago

Mentor: kmaglione+bmo@mozilla.com → mixedpuppy@gmail.com

Shane Caraveo (:mixedpuppy) ▾ Reply -
Comment 28 • 3 months ago
mozreview-review

Comment on ~~attachment 8860457 [details]~~
~~Bug 1269300~~ Patch 2

https://reviewboard.mozilla.org/r/132450/#review139396

```
The patch seems pretty straight forward, I just need to digest all the bug comments and think about tests and
the api namespace.

::: browser/components/extensions/ext-tabs.js:763
(Diff revision 1)
 > +      },
 > +
 > +      printPreview() {
 > +        let activeTab = getTabOrActive(null);
 > +        let {PrintUtils} = activeTab.ownerGlobal;
 > +        let {PrintPreviewListener} = activeTab.ownerGlobal;

let {
  PrintUtils,
  PrintPreviewListener
} = activeTab.ownerGlobal

::: browser/components/extensions/ext-tabs.js:772
(Diff revision 1)
 > +
 > +      saveAsPDF(pageSettings) {
 > +        let activeTab = getTabOrActive(null);
 > +        let picker =
Components.classes["@mozilla.org/filepicker;1"].createInstance(Components.interfaces.nsIFilePicker);
 > +
 > +        picker.init(activeTab.ownerGlobal,"Save As",Components.interfaces.nsIFilePicker.modeSave);

Be sure to run ./mach eslint on the files, it will catch a bunch of code style nits.

::: browser/components/extensions/ext-tabs.js:802
(Diff revision 1)
 > +            printSettings.showPrintProgress = false;
 > +
 > +            printSettings.printFrameType = Components.interfaces.nsIPrintSettings.kFramesAsIs;
 > +            printSettings.outputFormat = Components.interfaces.nsIPrintSettings.kOutputFormatPDF;
 > +
 > +            if (pageSettings.orientation !== null) printSettings.orientation = pageSettings.orientation;

For those attributes that are named the same, do something like:

for opt in ["orientation", ...]
 if pageSettings.has(opt) printSettings[opt] = pageSettings[opt]

::: browser/components/extensions/schemas/tabs.json:1036
(Diff revision 1)
 > +        "type": "function",
 > +        "description": "Saves page in active tab as a PDF file.",
 > +        "parameters": [
 > +          {
 > +            "type": "object",
 > +            "name": "pageSettings",

Lets make this a type, see "Tab" in the types section of this file.
```

Attachment #8860457 - Flags: ~~review?(mixedpuppy@gmail.com)~~

dw-dev ▾ (Assignee) Reply -

Comment 29 • 3 months ago

```
Shane, thanks for getting involved and reviewing my first ever attempt at a Firefox patch.

----------------------------------

With regards to your review points:

 > I just need to digest all the bug comments and think about tests and the api namespace.

I chose the 'tabs' namespace because it seemed the best fit of the existing namespaces. Chrome already has a
'printerProvider' namespace, but I'm not sure the new API's fit into that namespace. If we go for a new
namespace, then I would favour 'print' or 'pagePrint' (similar to 'pageCapture'). If necessary, we could also
rename saveAsPDF() to printToPDF() to make it more obvious that this is a print-based function.
```

```
> let {
>   PrintUtils,
>   PrintPreviewListener
> } = activeTab.ownerGlobal
```

Yes.

```
> Be sure to run ./mach eslint on the files, it will catch a bunch of code style nits.
```

Good idea.

```
> For those attributes that are named the same, do something like:
>
> for opt in ["orientation", ...]
>  if pageSettings.has(opt) printSettings[opt] = pageSettings[opt]
```

I originally had the code as you suggest, but decided against it, because it would allow an add-on to change any of the settings in printSettings. I think an add-on should only be allowed to change those settings that are configurable by the user in the Page Setup dialog. So these are the settings I have defined in pageSettings. I could change the code to use a loop with a defined list of properties, if you would prefer.

```
> Lets make this a type, see "Tab" in the types section of this file.
```

Good idea.

Back in Comment 25, I mentioned that I had tried adding callback functions, but could not get them to work correctly.

I have now realized what I was doing wrong and have implemented a callback function for saveAsPDF(). The callback function is called when the 'Save As' dialog closes and returns a status code (0 = saved, 1 = cancelled, 2 = replaced, 3 = cannot create, 4 = cannot replace). I have tested this with a modified version of my Print Edit WE add-on and it works really well.

I have also done some more thinking about the feasibility of callback functions for print() and printPreview(), based on my knowledge of how things work in printUtils.js and browser-content.js.

I don't think there is any easy way of knowing when a print operation has completed. A 'Printing:Error' message is sent if printing is unsuccessful, but no message is sent if printing is successful. If we really want to have a callback, we would need to call webBrowserPrint.print() in the content process with a print progress listener. Without a callback function, add-ons can listen for the 'afterprint' event, although I am not sure exactly when this event is fired.

Similarly, I don't think there is any easy way of knowing when entry into print preview has completed. It would be possible to listen for the 'Printing:Preview:Entered' message, except that I can't see any way of knowing the identity of the print preview browser. As an alternative, add-ons could listen for the 'beforeprint' event, although I am not sure exactly when this event is fired.

My feeling is that callback functions for print() and printPreview() are not essential at the moment and can be left for a future enhancement.

What do you think?

dw-dev ▾ (Assignee) Reply -

Comment 30 • 3 months ago

I have now found a way to implement a callback function for printPreview(). The callback function is called when print preview mode has been entered and returns a status code (false = failed, true = suceeded).

With regards to testing these new APIs:

print() - Based on Bob Owen's comments and the print code in printUtils.js, I am struggling to see what can be tested programatically.

printPreview() - We could test that the print preview toolbar has been created and that the print preview browser has been created.

saveAsPDF() - We could test that the file has been created with non-zero size, but only if it is possible to programmatically respond to the 'Save As' dialog.

I am wondering whether we really need to make the pageSettings object into a type, since I have noticed that this has not been done for the object parameters to create(), query() or update() ?

Shane Caraveo (:mixedpuppy) ▾ -

Updated • 2 months ago

See Also: → ~~Bug 1355421~~

Shane Caraveo (:mixedpuppy) ▾ -

Updated • 2 months ago

Duplicate of this bug: ~~1355421~~

Dimitar Angelov ▾ Reply -

Comment 32 • 2 months ago

I would like to assign to discussion with some additions to extending print API available to web extensions.
I'm developer of extension called jsPrintSetup (https://github.com/edabg/jsprintsetup/wiki) which expose global object jsPrintSetup available to client side javascript. The jsPrintSetup object give more flexibility about printing to web developers.
The main features are:
- Get/Set print settings - margins, orientation, scaling, header and footer
- Get/Set global print settings and for selected printer
- Working with installed printers - select to which printer to print
- Save print settings to user preferences (as defaults to printer or global for all printers)
- Print with current setttings without need from saving to user preferences as required from 'window.print()'
- Print desired window or frame
- Unattended printing without print dialog (silent printing)
- Enhanced Paper data handling (with some limitations for different platforms)

All of these features are provided via nsIPrintSettingsService, nsIPrintSettings, nsIPrinterEnumerator and nsIWebBrowserPrint interfaces
https://github.com/edabg/jsprintsetup/blob/master/src/components/jsPrintSetup.js

It will be fine if Chrome printerProvider namespace will be extended with API to manage all print settings provided via
- nsIPrintSettings - manage all print settings provided via interface
- nsIPrintSettingsService (init settings from preferences/save to preferences)
- nsIPrinterEnumerator - get list of printers
- nsIWebBrowserPrint - calling print to desired window/frame with provided settings to desired printer

dw-dev ▾ (Assignee) Reply -

Comment 33 • 2 months ago

I need a little bit of help configuring mochitest.

This what I have done:

 - updated the code for the new print(), printPreview() and saveAsPDF() API's and rebuilt Firefox.

 - designed a mochitest test file 'browser_ext_tabs_printPreview.js' based on the existing test files for the other browser.tabs API's, and have placed this file in 'browser\components\extensions\test\browser'.

 - updated 'browser-common.ini' in 'browser\components\extensions\test\mochitest' to include 'browser_ext_tabs_printPreview.js'.

 - updated 'test_ext_all_apis.html' in 'browser\components\extensions\test\mochitest' to include 'tabs.printPreview'.

 - run: mach build

 - run: mach mochitest browser\components\extensions\test\mochitest\test_ext_all_apis.html.

The problem is that the test results always show the tests as 'Passed', even if I purposely introduce some errors into the 'browser_ext_tabs_printPreview.js' test file (or into the existing test files for the other browser.tabs API's).

So, I am not convinced that my new test file is getting run, or that I am invoking the test suite correctly.

What am I doing wrong?

Flags: needinfo?(mixedpuppy@gmail.com)

Shane Caraveo (:mixedpuppy) ▾ Reply -

Comment 34 • 2 months ago

test_ext_all_apis.html is only testing if the APIs listed are available in the namespace for an extension. e.g. if you changed the entry you added to tabs.printPreviewZZZZ and ran it, you should get a failure.

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~

Comment hidden (mozreview-request) +

Shane Caraveo (:mixedpuppy) ▾ -

Updated • 2 months ago

Attachment #8860457 - Flags: review?(amckay@mozilla.com) → review?(mixedpuppy@gmail.com)

dw-dev ▾ (Assignee) Reply -

Comment 36 • 2 months ago

Shane - apologies that this review was not directed to you.

With regards to the latest patch:

- The source code of tabs.json and ext-tabs.js has been updated in line with the comments in Comment 28, except for one comment - see Note 1.

- Callback functions (returning a status value) have been added to tabs.printPreview() and tabs.saveAsPDF().

- An automated test has been added for tabs.printPreview(), but not for tabs.print() and tabs.saveAsPDF() - see Note 2.

- The source code has been run through mach eslint and corrections made.

- All three functions have been thoroughly tested using a modified version of Print Edit WE (19.0b2) - see attachment.

I think this patch is ready for your approval.

Note 1

I tried copying the pageSettings to the printSettings using the following code:

```
    for (let setting of ["orientation", "scaling", "shrinkToFit", ... "marginBottom"]) {

      if (pageSettings.hasOwnProperty(setting)) {

        printSettings[setting] = pageSettings[setting];

      }
    }
```

However, this code and all of the other looping variants I tried (indexed arrays, etc) failed with the error message:

```
    NS_ERROR_XPC_CANT_MODIFY_PROP_ON_WN: Cannot modify properties of a WrappedNative
```

Further testing showed that:

 - printSettings["orientation"] = pageSettings["orientation"] always works

 - printSettings[propName] = pageSettings[propName] often fails with the above error message.

Therefore, I have reverted to my original long-hand copying of the settings.

Note 2

My view is that automated testing of tabs.print() and tabs.saveAsPDF() would be difficult:

print() - Bear in mind that this is essentially just a one line function. Based on Bob Owen's comments and the code in printUtils.js, it is difficult to see what automatic tests could be performed. Is it possible to detect the 'Print' dialog opening? Is it possible to close the 'Print' dialog programmatically?

saveAsPDF() - To do an automatic test, we would need to respond to the 'Save As' dialog, and then check that the PDF file has been created with the correct name and non-zero size. Is it possible to detect the 'Save As' dialog opening? Is it possible to respond to the 'Save As' dialog programmatically? Where would the saved file be created?

dw-dev (Assignee) Reply -
Comment 37 • 2 months ago

Created ~~attachment 8875888 [details]~~
Print Edit WE 19.0b2 with calls to print(), printPreview() and saveAsPDF()

This is the modified version of Print Edit WE (19.0b2) used to test the three new API functions: tabs.print(), tabs.printPreview() and tabs.saveAsPDF().

dw-dev (Assignee) -
Updated • 2 months ago

Attachment #8875888 - Attachment description: Print Edit WE 19.0b2 wih calls to print(), printPreview() and saveAsPDF() → Print Edit WE 19.0b2 with calls to print(), printPreview() and saveAsPDF()

Shane Caraveo (:mixedpuppy) Reply -
Comment 38 • 2 months ago
mozreview-review

Comment on ~~attachment 8860457 [details]~~
~~Bug 1269300~~ Patch 2

https://reviewboard.mozilla.org/r/132450/#review151552

Looking good, just a few things to work through.

::: browser/components/extensions/ext-tabs.js:775
(Diff revision 2)

```
> +
> +            let mm = ppBrowser.messageManager;
> +
> +            let onEntered = (message) => {
> +              mm.removeMessageListener("Printing:Preview:Entered", onEntered);
> +              return resolve(!message.data.failed);
```

I think this should use ExtensionError.

```
if (message.data.failed) {
  throw new ExtensionError("print preview failed");
}
resolve();
```

And remove the boolean callback.

Alternative: After seeing multiple status types on saveAsPDF, maybe return a string for the status here to be consistent. The status would be either "failed" or "success"

::: browser/components/extensions/ext-tabs.js:803
(Diff revision 2)

```
> +                let fstream = Components.classes["@mozilla.org/network/file-output-
stream;1"].createInstance(Components.interfaces.nsIFileOutputStream);
> +                fstream.init(picker.file, 0x2A, 0x1B6, 0);  // write|create|truncate, file permissions rw-
rw-rw- = 0666 = 0x1B6
> +                fstream.close();  // unlock file
> +              } catch (e) {
> +                let promptService = Components.classes["@mozilla.org/embedcomp/prompt-
service;1"].getService(Components.interfaces.nsIPromptService);
> +                promptService.alert(null, "Warning", retval == 0 ? "Cannot create file." : "Cannot replace
file because it is locked.");
```

Strings in UI will need to be localizable.

::: browser/components/extensions/ext-tabs.js:804
(Diff revision 2)

```
> +                fstream.init(picker.file, 0x2A, 0x1B6, 0);  // write|create|truncate, file permissions rw-
rw-rw- = 0666 = 0x1B6
> +                fstream.close();  // unlock file
> +              } catch (e) {
> +                let promptService = Components.classes["@mozilla.org/embedcomp/prompt-
service;1"].getService(Components.interfaces.nsIPromptService);
> +                promptService.alert(null, "Warning", retval == 0 ? "Cannot create file." : "Cannot replace
file because it is locked.");
> +                return resolve(retval == 0 ? 3 : 4);
```

```
return strings for status

::: browser/components/extensions/ext-tabs.js:807
(Diff revision 2)
 > +                  let promptService = Components.classes["@mozilla.org/embedcomp/prompt-
 service;1"].getService(Components.interfaces.nsIPromptService);
 > +                  promptService.alert(null, "Warning", retval == 0 ? "Cannot create file." : "Cannot replace
 file because it is locked.");
 > +                  return resolve(retval == 0 ? 3 : 4);
 > +                }
 > +
 > +                let psService = Components.classes["@mozilla.org/gfx/printsettings-
 service;1"].getService(Components.interfaces.nsIPrintSettingsService);

Shorten to Cc["X"].getService(Ci.X) (and elsewhere in patch)

::: browser/components/extensions/ext-tabs.js:819
(Diff revision 2)
 > +                printSettings.showPrintProgress = false;
 > +
 > +                printSettings.printFrameType = Components.interfaces.nsIPrintSettings.kFramesAsIs;
 > +                printSettings.outputFormat = Components.interfaces.nsIPrintSettings.kOutputFormatPDF;
 > +
 > +                if (pageSettings.orientation !== null) printSettings.orientation = pageSettings.orientation;

if () {
..stuff
}

You should run ./mach eslint --setup then ./mach eslint path/to/file, it will show you any nits for style.

::: browser/components/extensions/schemas/tabs.json:140
(Diff revision 2)
 >        },
 >        {
 > +        "id": "PageSettings",
 > +        "type": "object",
 > +        "description": "The page settings including: orientation, scale, background, margins, headers,
 footers.",
 > +        "properties": {

Please format the json.  Remove "Str" and change "BG" to "Background".

::: browser/components/extensions/schemas/tabs.json:1078
(Diff revision 2)
 > +        "async": "callback",
 > +        "parameters": [
 > +          {
 > +            "$ref": "PageSettings",
 > +            "name": "pageSettings",
 > +            "description": "The page settings including: orientation, scale, background, margins, headers,
 footers. Note: Page Setup dialog cannot configure settings for printing to PDF file."

We don't need to include examples of what page settings are.  Just say "The page settings used to print the
PDF".  The note should go on the MDN documentation, but could probably use some clarity.

::: browser/components/extensions/schemas/tabs.json:1084
(Diff revision 2)
 > +          },
 > +          {
 > +            "type": "function",
 > +            "name": "callback",
 > +            "optional": true,
 > +            "description": "Called after save as dialog closed.",

Is it after the dialog is closed, or once the PDF is completely saved?

::: browser/components/extensions/schemas/tabs.json:1089
(Diff revision 2)
```

```
> +            "description": "Called after save as dialog closed.",
> +            "parameters": [
> +              {
> +                "type": "integer",
> +                "name": "status",
> +                "description": "Save status: 0 = saved, 1 = cancelled, 2 = replaced, 3 = cannot create, 4 =
cannot replace."
```

Change status to string and return the strings rather than numbers.

::: browser/components/extensions/test/browser/browser_ext_tabs_printPreview.js:42
(Diff revision 2)

```
> +  isnot(ppToolbar, null, "print preview toolbar created");
> +
> +  is(ppTab, gBrowser.selectedTab, "print preview tab selected");
> +  is(ppTab.linkedBrowser.currentURI.spec, "about:printpreview", "print preview browser url correct");
> +
> +  PrintUtils.exitPrintPreview();
```

Let's be sure we're waiting for this to complete.

```
await BrowserTestUtils.waitForCondition(() =>  !window.gInPrintPreviewMode);
```

Attachment #8860457 - Flags: review?(mixedpuppy@gmail.com) → review-

Shane Caraveo (:mixedpuppy) ▾ Reply -

Comment 39 • 2 months ago

mozreview-review

Comment on ~~attachment 8860457 [details]~~
~~Bug 1269300~~ Patch 2

https://reviewboard.mozilla.org/r/132450/#review151584

::: browser/components/extensions/ext-tabs.js:803
(Diff revision 2)

```
> +                  let fstream = Components.classes["@mozilla.org/network/file-output-
stream;1"].createInstance(Components.interfaces.nsIFileOutputStream);
> +                  fstream.init(picker.file, 0x2A, 0x1B6, 0);  // write|create|truncate, file permissions rw-
rw-rw- = 0666 = 0x1B6
> +                  fstream.close();  // unlock file
> +                } catch (e) {
> +                  let promptService = Components.classes["@mozilla.org/embedcomp/prompt-
service;1"].getService(Components.interfaces.nsIPromptService);
> +                  promptService.alert(null, "Warning", retval == 0 ? "Cannot create file." : "Cannot replace
file because it is locked.");
```

Probably should just skip the promptservice and return an error. I'm also still a bit torn between the use of a string returned for an error vs. ExtensionError, thinking about it more.

dw-dev ▾ (Assignee) Reply -

Comment 40 • 2 months ago

Thanks for the feedback.

With regards to the points you've raised:

printPreview():

```
> I think this should use ExtensionError.
>
> if (message.data.failed) {
>   throw new ExtensionError("print preview failed");
> }
> resolve();
>
> And remove the boolean callback.
```

OK. I think we should take the same approach with saveAsPDF() - see last point below.

Please confirm that the error message can be retrieved in the callback function using browser.runtime.lastError.

> Strings in UI will need to be localizable.

Assuming we use ExtensionError for saveAsPDF(), then the only UI string will be "Save As" in the file picker dialog.

Do the error strings passed into ExtensionError() count as UI strings?

Which .dtd file should this string(s) be declared in?

> Shorten to Cc["X"].getService(Ci.X) (and elsewhere in patch)

OK.

> if () {
> ..stuff
> }
>
> You should run ./mach eslint --setup then ./mach eslint path/to/file, it will show you any nits for style.

I have already done this for ext-tabs.js and have just done it again. eslint does not flag any errors or warnings.

I can put curly braces on these "if" statements, but it will make the code a lot longer.

Single-line conditional assignments might be a cleaner solution, for example:

```
  printSettings.orientation = (pageSettings.orientation !== null) ? pageSettings.orientation : ;
```

> Please format the json. Remove "Str" and change "BG" to "Background".

I just copied the format from the "Tab" declaration as you had suggested.

What do you mean by: remove "Str"?

I have used "BG" for consistency with the setting in the standard printSettings object.

> We don't need to include examples of what page settings are. Just say "The page settings used to print the PDF".

OK.

> Is it after the dialog is closed, or once the PDF is completely saved?

At present, it is after the dialog is closed.

I could add a printProgressListener(), which is what I do in my original Print Edit add-on, but it is quite a lot of code. If we do this, there would be an additional error state ("save failed") to pass back in ExtensionError.

> Change status to string and return the strings rather than numbers.

Status will no longer be returned as parameter. Will call ExtensionError instead.

> Let's be sure we're waiting for this to complete.
>
> await BrowserTestUtils.waitForCondition(() => !window.gInPrintPreviewMode);

OK.

saveAsPDF():

>
> Probably should just skip the promptservice and return an error. I'm also still a bit torn between the use of a string
> returned for an error vs. ExtensionError, thinking about it more.

For consistency with printPreview(), I think we should just skip the promptservice and call ExtensionError with the strings "cannot save file" or "cannot replace file" in the error cases.

If you can provide clarifications on the above points, I should be able to update the patch in a couple of days.

Flags: needinfo?(kmaglione+bmo@mozilla.com) → needinfo?(mixedpuppy@gmail.com)

Shane Caraveo (:mixedpuppy) ▾

Comment 41 • 2 months ago

Reply -

(In reply to dw-dev from comment #40)
> Thanks for the feedback.
>
> With regards to the points you've raised:
>
> printPreview():
>
> > I think this should use ExtensionError.
> >
> > if (message.data.failed) {
> >   throw new ExtensionError("print preview failed");
> > }
> > resolve();
> >
> > And remove the boolean callback.
>
> OK. I think we should take the same approach with saveAsPDF() - see last
> point below.
>
> Please confirm that the error message can be retrieved in the callback
> function using browser.runtime.lastError.

Yeah. I was going to add...ext-management uses promptService and illustrates l10n if you needed an example. I'm on the fence about which is better.

> > Strings in UI will need to be localizable.
>
> Assuming we use ExtensionError for saveAsPDF(), then the only UI string will
> be "Save As" in the file picker dialog.
>
> Do the error strings passed into ExtensionError() count as UI strings?

No.

> Which .dtd file should this string(s) be declared in?

See the localization in ext-management.

> I have already done this for ext-tabs.js and have just done it again.
> eslint does not flag any errors or warnings.

If eslint passes here I'm fine.

> > Please format the json. Remove "Str" and change "BG" to "Background".
>
> I just copied the format from the "Tab" declaration as you had suggested.

This one I want formatted, it's harder to read as one line.

> What do you mean by: remove "Str"?

e.g. footerStrLeft, change to footerLeft

> I have used "BG" for consistency with the setting in the standard
> printSettings object.

We don't need or particularly want consistency with internal interfaces. The point here is to make the webextension interface more readable.

> > Is it after the dialog is closed, or once the PDF is completely saved?
>
> At present, it is after the dialog is closed.
>
> I could add a printProgressListener(), which is what I do in my original
> Print Edit add-on, but it is quite a lot of code. If we do this, there would
> be an additional error state ("save failed") to pass back in ExtensionError.

That could be nice, but not necessary at this point. I just want to be sure the behavior is documented.

> > Probably should just skip the promptservice and return an error. I'm also still a bit torn between the use of a string
> > returned for an error vs. ExtensionError, thinking about it more.
>
> For consistency with printPreview(), I think we should just skip the
> promptservice and call ExtensionError with the strings "cannot save file" or
> "cannot replace file" in the error cases.

preferably just simple strings.

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~

Comment hidden (mozreview-request) +

Comment hidden (mozreview-request) +

dw-dev ▾ (Assignee) Reply -
Comment 44 • 2 months ago

A quick commentary on Patch 3;

1. All of the suggested changes have been addressed in Patch 3.

2. My testing indicates that ExtensionError does not update the value of browser.runtime.lastError and does not provide any feedback to the calling add-on.

3. printPreview() - Now uses ExtensionError. This feels right, because printPreview() should always work, and so any failure is a browser error. I am slightly concerned that there is no feedback to the calling add-on, but in practice this is probably not a big issue.

4. saveAsPDF() - I no longer think that using ExtensionError is appropriate, because it does not provide any feedback to the calling add-on. My Print Edit WE add-on definitely needs to know the result of the save operation, and so I have decided to retain the callback 'status' parameter. The five status values are now: 'Saved', 'Replaced', 'Not saved', 'Not replaced' and 'Cancelled'. I could collapse this to just three status values: 'Saved', 'Failed' and 'Cancelled' - but I think this would be less useful.

5. The test file has been updated to check that 'window.gInPrintPreviewMode' is true when the printPreview() callback is called.

6. eslint has been run on all changed files. The test file passes all tests.

7. A build with Patch 3 works correctly with Print Edit WE 19.0b3 (see attachment).

Flags: needinfo?(mixedpuppy@gmail.com)

dw-dev ▾ (Assignee) Reply -
Comment 45 • 2 months ago

Created attachment 8876521 [details]
Print Edit WE 19.0b3 with calls to print(), printPreview() and saveAsPDF()

dw-dev ▾ (Assignee) -
Updated • 2 months ago

Attachment #8875888 - Attachment is obsolete: true

Shane Caraveo (:mixedpuppy) ▾ Reply -

Comment 46 • a month ago

```
The error is normalized depending on how you use the api.  Something like this:

chrome.api.foo(args, () => {
  runtime.lastError is set
})

browser.api.foo(args).then(() => { success }, error => {  });
```

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~

Shane Caraveo (:mixedpuppy) ▾ Reply -

Comment 47 • a month ago

mozreview-review

```
Comment on ~~attachment 8876520 [details]~~
~~Bug 1269300~~ Patch 3;

https://reviewboard.mozilla.org/r/147848/#review154072

I think this is looking fine now, last change request is to bring it a little more inline with how we're writing
tests these days.

::: browser/components/extensions/test/browser/browser_ext_tabs_printPreview.js:22
(Diff revision 1)
 > +        browser.tabs.printPreview(() => {
 > +          browser.test.assertTrue(true,"print preview callback function called");
 > +          browser.test.notifyPass("tabs.printPreview");
 > +        });
 > +      });
 > +    },

background: async function() {
  let tabs = await browser.tabs.query({....})
  await browser.tabs.printPreview()
  browser.test(...)
  browser.test.notifyPass(...)
});
```

Attachment #8876520 - Flags: review?(mixedpuppy@gmail.com) → review+

dw-dev ▾ (Assignee) Reply -

Comment 48 • a month ago

```
(In reply to Shane Caraveo (:mixedpuppy) from comment #46)

 > The error is normalized depending on how you use the api.  Something like
 > this:
 >
 > chrome.api.foo(args, () => {
 >   runtime.lastError is set
 > })
 >
 > browser.api.foo(args).then(() => { success }, error => {  });

I have just re-tested ExtensionError by modifying saveAsPDF() in ext-tabs.js as follows:

    try {
      let fstream = Cc["@mozilla.org/network/file-output-stream;1"].createInstance(Ci.nsIFileOutputStream);
      fstream.init(picker.file, 0x2A, 0x1B6, 0);  // write|create|truncate, file permissions rw-rw-rw- = 0666 =
0x1B6
      fstream.close();  // unlock file
    } catch (e) {
      throw new ExtensionError("Save as PDF failed");
      resolve(retval == 0 ? "Not saved" : "Not replaced");
      return;
    }

and then calling saveAsPDF() from Print Edit WE as follows:

    chrome.tabs.saveAsPDF(pageSettings,
    function(status)
    {
```

```
        console.log(status);
        if (chrome.runtime.lastError) console.log(chrome.runtime.lastError.message);
    });
```

For the saved or replaced cases, the console shows "Saved" or "Replaced".

For the not saved or not replaced cases, the console shows "Error: Save as PDF failed" which is logged by ExtensionError, but does NOT show "Save as PDF failed" which should be logged by the call in Print Edit WE if chrome.runtime.lastError != null.

dw-dev ▾ (Assignee)

Comment 49 • a month ago

Reply -

(In reply to Shane Caraveo (:mixedpuppy) from comment #47)

> Comment on ~~attachment 8876520 [details]~~
> ~~Bug 1269300~~ Patch 3;
>
> https://reviewboard.mozilla.org/r/147848/#review154072
>
> I think this is looking fine now, last change request is to bring it a
> little more inline with how we're writing tests these days.
>
> :::
> browser/components/extensions/test/browser/browser_ext_tabs_printPreview.js:
> 22
> (Diff revision 1)
> > +   browser.tabs.printPreview(() => {
> > +   browser.test.assertTrue(true,"print preview callback function called");
> > +   browser.test.notifyPass("tabs.printPreview");
> > +   });
> > +   });
> > +   },
>
> background: async function() {
>   let tabs = await browser.tabs.query({....})
>   await browser.tabs.printPreview()
>   browser.test(...)
>   browser.test.notifyPass(...)
> });

My understanding that the review has been granted. That's great!

Does this change need to be made before uploading this patch ?

Presumably browser.test(...) should still be browser.test.assertTrue(...) ?

How do I upload this patch so that it ends up in Nightly 56.0a1 ?

Flags: needinfo?(mixedpuppy@gmail.com)

Shane Caraveo (:mixedpuppy) ▾

Comment 50 • a month ago

Reply -

(In reply to dw-dev from comment #49)

> (In reply to Shane Caraveo (:mixedpuppy) from comment #47)
> > Comment on ~~attachment 8876520 [details]~~
> > ~~Bug 1269300~~ Patch 3;
> >
> > https://reviewboard.mozilla.org/r/147848/#review154072
> >
> > I think this is looking fine now, last change request is to bring it a
> > little more inline with how we're writing tests these days.
> >
> > :::
> > browser/components/extensions/test/browser/browser_ext_tabs_printPreview.js:
> > 22
> > (Diff revision 1)
> > > +   browser.tabs.printPreview(() => {
> > > +   browser.test.assertTrue(true,"print preview callback function called");
> > > +   browser.test.notifyPass("tabs.printPreview");
> > > +   });
> > > +   });
> > > +   },
> >
> > background: async function() {
> >   let tabs = await browser.tabs.query({....})
> >   await browser.tabs.printPreview()
> >   browser.test(...)
> >   browser.test.notifyPass(...)
> > });
>
> My understanding that the review has been granted. That's great!
>
> Does this change need to be made before uploading this patch ?

Yes, per my comment. I think we should also understand the error issue first. I was about to look at that more.

> Presumably browser.test(...) should still be browser.test.assertTrue(...) ?

yes, I was just overly illustrating where you can use await.

So lets async'ify the test, verify the error stuff, once that's done you can add checkin-needed to the keywords field.

Also for sake of clarity, remove the two older patches from reviewboard. You should be able to do that from the review summary page in one of the menus.

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~

Shane Caraveo (:mixedpuppy) ▾ Reply -

Comment 51 • a month ago

oh, lets run try before as well.

dw-dev ▾ (Assignee) Reply -

Comment 52 • a month ago

With regards to the runtime.lastError issue, I know the chrome.tabs and chrome.windows API's very well from the add-on calling side, and I have encountered a lot of the errors returned in runtime.lastError.

Looking at the source in ext-tabs.js and ext-windows.js, it appears that in all cases the errors are notified in runtime.lastError by rejecting a promise.

For example, in windows.create():

```
    return Promise.reject({message: "`tabId` may not be used in conjunction with `url`"});
```

dw-dev ▾ (Assignee) -

Updated • a month ago

Attachment #8860457 - Attachment is patch: true
Attachment #8860457 - Attachment mime type: text/x-review-board-request → text/plain

dw-dev ▾ (Assignee) -

Updated • a month ago

Attachment #8860457 - Attachment is obsolete: true

Attachment #8860457 - Attachment is patch: false
Attachment #8860457 - Flags: ~~review?(kmaglione+bmo@mozilla.com)~~

dw-dev (Assignee)
Updated • a month ago

Attachment #8876519 - Attachment is obsolete: true

dw-dev (Assignee)
Comment 53 • a month ago

Reply

(In reply to Shane Caraveo (:mixedpuppy) from comment #50 and comment #51)

> So lets async'ify the test,

OK.

> verify the error stuff,

I found this statement by Andy McKay in the Mozilla Add-ons Blog:

(https://blog.mozilla.org/addons/2016/03/11/webextensions-in-firefox-47)

"All asynchronous methods which accept a callback function will now return a Promise if no callback is passed. These promises resolve at the same time a callback would normally be called, and reject with the value of lastError in cases where that would otherwise be set."

In any case, I think that the callback 'status' parameter is a much better better way of returning the outcome of tabs.saveAsPDF.

No change proposed.

> once that's done you can add checkin-needed to the keywords field.

Before doing that, I assume I will have to submit Patch 4 with the async'ify test changes?

Can I just submit an incremental Patch 4 (make and test changes, hg commit, hg push review) or do I have to back out Patch 3 first?

Where do I add the 'checkin-needed' flag (which keyword field)?

> Also for sake of clarity, remove the two older patches from reviewboard.

I have marked Patch 1 and Patch 2 as obsolete.

> oh, lets run try before as well.

Do I do that or do you do that?

I have read the documentation about the try server, but I am slightly concerned that I could mess up my repository.

Flags: needinfo?(mixedpuppy@gmail.com)

Shane Caraveo (:mixedpuppy)
Comment 54 • a month ago

Reply

Sorry for the delayed response.

(In reply to dw-dev from comment #53)
> (In reply to Shane Caraveo (:mixedpuppy) from comment #50 and comment #51)
> > once that's done you can add checkin-needed to the keywords field.
>
> Before doing that, I assume I will have to submit Patch 4 with the async'ify
> test changes?

yeah, and i'll take another peak at it, I'll add the keyword when I r+

> Can I just submit an incremental Patch 4 (make and test changes, hg commit,
> hg push review) or do I have to back out Patch 3 first?

preferably just one patch.

> > oh, lets run try before as well.
>
> Do I do that or do you do that?

You should be able to do that from reviewboard, if not let me know I'll kick it off.

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~

Comment hidden (mozreview-request) +

dw-dev (Assignee) -
Updated • 20 days ago

Attachment #8876520 - Attachment is obsolete: true

dw-dev (Assignee) Reply -
Comment 56 • 20 days ago

A quick commentary on Patch 4:

1. The test file has been changed as you suggested in Comment 47.

2. eslint has been run on all changed files.

3. The test file passes all tests.

4. A build with Patch 4 works correctly with Print Edit WE 19.0b3 (see attachment).

5. I was going to run 'mach try ...', and I had a look at the TryChooser Syntax Builder, but I couldn't figure out what options to use.

6. Please could you submit this patch to the Try Server? And if everything goes well, then add the 'checkin-needed' flag.

Thanks.

Flags: needinfo?(mixedpuppy@gmail.com)


Shane Caraveo (:mixedpuppy) Reply -
Comment 57 • 20 days ago

Try submitted, will revisit this week.


Shane Caraveo (:mixedpuppy) Reply -
Comment 58 • 17 days ago

Hrm. tests didn't run. There is an eslint failure that will need to be fixed (run ./mach eslint --setup, then ./mach eslint path/to/stuff). New try started.

Comment hidden (mozreview-request) +

dw-dev (Assignee) Reply -
Comment 60 • 16 days ago

Apologies about Patch 4. Somewhere along the line I must have lost the Mozilla-specific eslint setup, since I didn't see the errors before submitting Patch 4. All of the eslint errors are corrected in Patch 5.

Regarding Patch 5:

1. eslint has been run on all files and the errors (in two files) have been corrected.

2. The printPreview test file passes all tests.

3. A build with Patch 5 works correctly with Print Edit WE 19.0b3 (see attachment).

4. Please could you submit this patch to the Try Server? And if everything goes well, then add the 'checkin-needed' flag.

Thanks.

Shane Caraveo (:mixedpuppy) ▾ Reply -
Comment 61 • 16 days ago
mozreview-review
Comment on attachment 8884626 [details]
~~Bug 1269300~~ Patch 5;

https://reviewboard.mozilla.org/r/132448/#review162148

Shane Caraveo (:mixedpuppy) ▾ -
Updated • 16 days ago

Attachment #8884626 - Flags: ~~review?(mixedpuppy@gmail.com)~~

Shane Caraveo (:mixedpuppy) ▾ -
Updated • 16 days ago

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~
Attachment #8884626 - Flags: review+

Shane Caraveo (:mixedpuppy) ▾ -
Updated • 16 days ago

Keywords: checkin-needed, dev-doc-needed

dw-dev ▾ (Assignee) Reply -
Comment 62 • 16 days ago

Shane, Who should take action on the 'dev-doc-needed' flag? Is that me or someone else? I am happy to contribute or write the first draft of the documentation.

Shane Caraveo (:mixedpuppy) ▾ Reply -
Comment 63 • 16 days ago

(In reply to dw-dev from comment #62)
> Shane, Who should take action on the 'dev-doc-needed' flag? Is that me or
> someone else? I am happy to contribute or write the first draft of the
> documentation.

You're certainly welcome to write up docs, I think you should be able to put it right into MDN also. Lets get some input on that to make sure.

Flags: needinfo?(wbamberg@mozilla.com)

Ryan VanderMeulen [:RyanVM] ▾ Reply -
Comment 64 • 16 days ago

Autoland can't push this until the patch meets the MozReview review requirements.
http://mozilla-version-control-tools.readthedocs.io/en/latest/mozreview/autoland.html#landing-commits

Assignee: nobody@mozilla.org → dw-dev@gmx.com
Keywords: ~~checkin-needed~~

dw-dev ▾ (Assignee) Reply -
Comment 65 • 15 days ago

Shane, I don't understand which of the MozReview review requirements have not been met. What do we need to do?

Flags: needinfo?(mixedpuppy@gmail.com)

Shane Caraveo (:mixedpuppy) ▾ Reply -
Comment 66 • 15 days ago
mozreview-review

Comment on attachment 8884626 [details]
~~Bug 1269300~~ Patch 5;

https://reviewboard.mozilla.org/r/132450/#review162536

Pulsebot ▾ Reply -
Comment 67 • 15 days ago

Pushed by mixedpuppy@gmail.com:
https://hg.mozilla.org/integration/autoland/rev/19626ad14682
Patch 5;r=mixedpuppy

Ryan VanderMeulen [:RyanVM] ▾ Reply -
Comment 68 • 15 days ago

In the future, please try to use commit messages which summarize what the patch is doing.

http://mozilla-version-control-tools.readthedocs.io/en/latest/mozreview/commits.html#write-detailed-commit-messages

Shane Caraveo (:mixedpuppy) ▾ Reply -
Comment 69 • 15 days ago

Yeah, I just realized that as I clicked land...nooooo

Flags: ~~needinfo?(mixedpuppy@gmail.com)~~

Wes Kocher (:KWierso) ▾ Reply -
Comment 70 • 15 days ago
bugherder

https://hg.mozilla.org/mozilla-central/rev/19626ad14682

Status: NEW → RESOLVED
Last Resolved: 15 days ago
status-firefox56: --- → fixed
Resolution: --- → FIXED
Target Milestone: --- → mozilla56

Caitlin Neiman (http://pronoun.is/she) ▾ Reply -
Comment 71 • 5 days ago

Thanks so much for writing this API, dw-dev! This has been added to our recognition wiki here:
https://wiki.mozilla.org/Add-ons/Contribute/Recognition#July_2017

If you're interested in setting up a profile on mozillians.org, I would be happy to vouch for your contributions.

Thanks again and welcome onboard!

Will Bamberg [:wbamberg] ▾ Reply -
Comment 72 • 4 days ago

(In reply to dw-dev from comment #62)
> Shane, Who should take action on the 'dev-doc-needed' flag? Is that me or
> someone else? I am happy to contribute or write the first draft of the
> documentation.

dw-dev, sorry to be slow replying. I'm happy to write some docs for this, and I'll ask you to take a look when I've done so.

Flags: ~~needinfo?(wbamberg@mozilla.com)~~

Will Bamberg [:wbamberg] ▾ Reply -
Comment 73 • 14 hours ago

Created attachment 8891595 [details]
saveaspdf.zip

The saveAsPDF API isn't working for me. I've attached a sample add-on: it just adds a browser action, and has a background script like this:

```
function handleClick() {
  browser.tabs.saveAsPDF({});
}

browser.browserAction.onClicked.addListener(handleClick);
```

In Firefox 56.0a1, when I click the browser action (say in http://example.com/), I get the "Save As" dialog. I select a name and location and choose "Save". A file does appear with the right name and location, but it's empty.

If I access the same feature using the browser's built-in UI: Ctrl-P/select "Save As PDF..." then the PDF is saved as expected.

Do I need to pass some different settings?

dw-dev (Assignee) Reply -

Comment 74 • 22 minutes ago

I have tried installing your saveaspdf.zip add-on in Nightly 56.0a1 on Windows 10 - and it works fine.

In my Print Edit WE add-on beta version I use:

```
    var pageSettings = {};

    pageSettings.orientation = 0;
    pageSettings.scaling = 1.0;
    pageSettings.shrinkToFit = false;
    pageSettings.showBackgroundColors = true;
    pageSettings.showBackgroundImages = true;

    browser.tabs.saveAsPDF(pageSettings,function(status) { });
```

It occurs to me that this issue may be operating system specific.

Which operating system are you using?

Flags: needinfo?(wbamberg@mozilla.com)

kernp25 Reply -

Comment 75 • 4 minutes ago

I have tested the add-on "saveaspdf.zip" and it was working fine for me.

Add Comment | Preview

Needinfo requested from *wbamberg@mozilla.com*.

Comments Subject to Etiquette and Contributor Guidelines

Need more information from other

Save Changes

Top ↑ | Format Bug ▲ | New/Clone Bug ▲