That pref was thought to be for extensions which wanted to completely replace our own Session Restore functionality. While this has worked somehow for Tab Mix Plus, we've had several issues with people ending up both Session Restore and Tab Mix Plus disabled (see bug 435055 and its duplicates).

Furthermore, there are several code points which will also break when Session Restore has been disabled (such as the list of recently closed tabs). Instead of adding try-catch-blocks wherever we use Session Restore, I'd much rather encourage extensions authors to override both nsSessionStartup and nsSessionStore to provide the same API with their own functionality (or implementing a dummy-API and making sure for themselves that they've correctly replaced all known consumers). This would also make the lives of those other extension authors simpler who so far can't be too sure that the Session Store component actually works (through whatever implementation). Note that privacy concerned users will still be able to disable writing to sessionstore.js through the browser.sessionstore.resume\_from\_crash pref. Created an attachment (id=332726) [details]. Remove the pref (note: bug 448725 should be wont fixed if this is fixed). Created an attachment (id=333820) [details]. Remove (buggy) note from API comments. (From update of attachment 332726 [details]) , a problem with this patch is that the session data is still stored in memory while the app is running, and by removing this pref, there's no way to disable that. some users do not want the recently-closed-tabs menu, and others don't want any session tracks stored in memory at all. &gt; some users do not want the recently-closed-tabs menu, That's what browser.sessionstore.max\_tabs\_undo is for: setting it to 0 effectively disables the feature. &gt; and others don't want any session tracks stored in memory at all. Then again, we don't save any data that wouldn't be in memory, anyway, or do we? I'd rather introduce a different pref or different means to cater the privacy sensitive users than have this half-baked cut-it-all pref which AFAICT so far has produced more issues than it's solved. &gt; Then again, we don't save any data that wouldn't be in memory, anyway, or do we? Fair point. I'm not sure. &gt; I'd rather introduce a different pref or different means to cater the privacy sensitive users than have this half-baked cut-it-all pref which AFAICT so far has produced more issues than it's solved. Yes, agreed the pref is not ideal for this purpose. So max\_tabs\_undo=0 + resume\_from\_crash=false is fine for now, until proper \"private browsing\" is supported. &gt;+ // XXXzeniko should't we just disable this item as we disable &gt;+ // the tab browser-multiple items above - for consistency? &gt;+ this.mUndoCloseTabMenuItem.hidden = &gt;+ Cc[\"@mozilla.org/browser/sessionstore;1\"].

&gt;+ getService(Ci.nsISessionStore).&gt;+ getClosedTabCount(window) == 0;+1,r=me, &gt; +1,That's bug 350731. Care to convince mconnor that he's wrong? ;-) Pushed as 17120:e712e96d7861 and 17121:adb1ef78dd21. One men: This patch significantly affects Tab Mix Plus: Instead of disabling SessionStore, you'll now have to replace it by shipping a component which implements the same API. You should be able to keep that component minimal, though, and just call your own code whenever the API is used. This will make the lives of people using the SessionStore API (such as Session Manager) somewhat simpler, as they can just use the API without having to worry too much about the implementation behind it. If you want to offer the option of switching between your and our implementation, see e.g. the ignore-history component of the Torbutton extension for how to overwrite a component with the original still being available internally (so that you can either pass API calls forward or handle them yourself). Sounds like this has impact for extension authors and so should be documented on MDC no problem. current Tabmix dev-build already not disable SessionStore. I currently have only one problem , how to disable the restore after restart. can you add a pref for this, or some other way to do it? &gt; I currently have only one problem , how to disable the restore after restart. You've got several options for that:\* Set the prefs browser.sessionstore.resume\_from\_crash and browser.sessionstore.resume\_session\_once both to false as early as possible and make sure that browser.startup.page isn't 3. \* Delete the file sessionstore.js as early as possible (e.g. when the profile-after-change notification is dispatched).\* For Firefox 3.1: Respond to the sessionstore-state-read notification by setting the subject's data member to an empty string (cf. bug 448741 comment#6).Or is there a use case I'm missing?