I've been working on a roadmap for Thunderbird for some time. It's always going to be work in progress, but below is a listing of some of the things I think we should be focusing on. Open to comments and insights. Some parts are not exactly clear how/if they should be done. If you reply, please quote only the relevant parts.

The tentative target release(s) are in brackets. After 68, the next general release is going to be Thunderbird 78. After that, it's likely version 90 but we don't know yet what the ESR schedule is that far in the future, so I've marked them TB2021 and TB2022.

We're planning to hire some more people soon, which should help speed things up. Watch maildev and tb-planning for job postings.

-Magnus

Thunderbird 78 +++ Roadmap

This provides a listing of maintenance and feature work wanted for Thunderbird 78 release and beyond. How much of it we can do for 78 depends on how much staff resources we have available.

NOTE: this is a work in progress, so everything is subject to change.

1. Address book rewrite [TB78]

The address book needs to a big time refresh. We will design the address book such that it could work as a general utility, not only inside Thunderbird although Thunderbird is the primary consumer of it.

The new address book would live inside a tab, like calendar, tasks, and chat currently do. It will provide a sane internal API for accessing contacts, and add flexibility to what kind of data can be stored for a contact (more than two email, oh yes!), and contact tagging.

After address book rewriting is done, it should be easier to add support for some long requested features, such as the below.

- CardDAV support: bug 546932
- vCard v4: bug 29106

Tracked in bug 841598.

2. Fluent migration [TB78/TB2021]

For localization Firefox is moving (https://arewefluentyet.com/) to Fluent, and Thunderbird will follow. For the Thunderbird 78 timeframe, target migrating 30% of the localization.

Firefox has officially deprecated the old style localization system since October 2019.

This is tracked in bug 1492751.

3. Migration from XUL to HTML [TB78]

Following Firefox, Thunderbird will also move away from XUL documents and towards top-level (X)HTML windows. Thunderbird has over 200 top level documents, so this is a significant effort. The Firefox team is planning to have the migration partly scripted, which will help in pulling it through.

In parallel, the conversion from individual XUL elements to corresponding HTML ones will continue.

Top level XUL->XHTML conversion is tracked in bug 1572398.

4. Protocol rewrites in JavaScript [TB2021/TB2022]

Implement one or two protocols in JavaScript, and make it an option to use that implementation. Prime candidates are IMAP and SMTP.

We should consider dropping support for Movemail.

5. JMAP support [TB2021/TB2022]

Implementing JMAP (RFC 8620) support at an early stage could be useful for getting experience in (re-)implementing protocols, without worrying to much about existing limitations, and at the same time obtain support for JMAP.

Tracked in bug 1322991.

6. Backend

6.1. Kill RDF [TB68/TB78]

RDF removal has been in the works during the previous cycle, and it’s finally time to drop it. This makes folder-related code, and feeds related code easier to work with.

Tracked in bug 420506. DONE!

6.2. Kill Mork [TB78]

The odd Mork file/”database” format is very unapproachable, and hard to understand. We should pay away this technical debt to finally get rid of it, paving the way for more modern approaches, and enabling consistent, proper async database operations. Mork is used for

- Address book data: tracked in bug 382878
- Folder message indexing data (cache): tracked in bug 1572000

To enable data migration, Mork removal would happen once a major Mork-free release is done.

Tracked in bug 453975.

6.3. Global Index for Messages [TB2021]

Currently the index of messages is per folder (and based on Mork). This causes severe limitations to implementing a proper conversation view of messages. Especially for Gmail, there is also message duplication on the file system since their All Mail folder has all mails, but the other folders do not know about this… For our special message views (Unified Inbox etc.) and search folders, lack of cross folder database makes the implementations far more complicated than they should be.

This work is tracked in bug 1572000.

6.4. Progress with libmime removal [TB78/TB2021/TB2022]

Libmime is the cruftiest piece of code we have: it is hard to understand and its maintenance is error prone. The library is used all over the place, so to the extent it’s possible we should start converting code over one part at a time. A number of years back the MIME header parsing was converted over to using the JSMime JavaScript library.

Message sending is very entangled with libmime. For Thunderbird 78 we will finish the rewrite of sending in JavaScript. This work is tracked in bug 1211292.

For Thunderbird 78, we should also get started with using JavaScript for the body parsing of MIME. Add an option to use it so dogfooding can find problems. Once successful, turn the new parsing on as default for the next major release. After default for a release, drop libmime.

6.5. Investigate Microsoft TNEF support [TB78/TB2021]

Investigate if integrating TNEF support can be integrated cleanly enough. If it can, we could integrate it. If not, we should take the needed steps to make it possible.

Tracked in bug 77811.

6.6. Maildir [TB78]

Our maildir support is pretty good by now. Finish up the last things needed to have maildir (our one file per message storage) on by default for new accounts.

Tracked in bug bug 845952

7. Mail front-end-ish

7.1. Front-end alignments with Firefox [TB78]

Use the JavaScript actor API, like Firefox?. Should help with performance.

7.2. Response Design [TB78/TB2021]

Responsive design: make the UI use proper layout depending on the screen size. Proper support for widescreen monitors (mails/chat/calendar side-by-side would be nice).

7.3. Better offline and slow network support [TB2021]

We should provide better offline and slow-network support (use local data first; performance boost). Leverage existing browser level caching and offline support mechanisms to achieve this. To be able to use those, this means data access through HTTP or pre-fetched into an IndexedDB database.

Requires investigation of feasibility.

7.4. Folder pane usefulness improvements [TB78]

The folder pane should allow mixing multiple folder modes like Outlook. Thunderbird already support different folder modes, but it's currently one at a time. It would be much more useful to be able to show a section with (e.g.) Favorites on top, and All Folders beneath it. Allow the user to choose which modes to put where.

Tracked in bug 1574129

7.5. Moving chrome UI to content UI [TB78/TB2021]

Thunderbird top level UI is currently part of the chrome UI. With the target of becoming a super fat privileged web application eventually, this needs to be rethought: web applications live in content. It's also become clear that the platform support for many features one can use in a web page using web technologies is implemented in a way essentially requiring the "page" to be located in a browser element (I'm looking at you, form controls and validation), and accessed through http (localStorage etc.). We should investigate to what degree we can move our chrome UI basically into a tabbrowser. Making sure we're treated as an http accessed web page will enable many features without hacks: localStorage, IndexedDB, CSP etc. - even simple functionality such as opening new windows have caused breakage recently.

For add-ons wanting good access to modifying the UI, this would be a significant win, since WebExtensions are primarily designed for modifying content.

Requires investigation of feasibility.

7.6. Separation of chrome UI and content, redesigned content data access [TB78/TB2021]

Internally Thunderbird is using mailbox: and imap: etc. URI references to access the mail data. This area has proven to be very problematic especially during the last year. To eliminate much of these problems, we will separate out the data access providing that through IndexedDB database or over an HTTP internal API instead. This will make back-end/front-end separation much cleaner, and of course more webby.

Requires investigation of feasibility. But can fairly easily be done is small pieces.

8. PGP support [TB78]

Encryption is an increasingly important part of email communication. Thunderbird will include native support for reading and sending PGP encrypted email. We want to enable doing many of the things Enigmail does, but old baggage like writing inline PGP and support for arcane encryption mechanisms will not be carried over.

Tracked in bug 22687.

9. Calendar

9.1. Libical removal [TB78]

Putting some effort into making ical.js performant enough will let us drop our very old fork of the C++ libical and focus on making the JavaScript implementation great. For 78 ical.js would be on by default, but the old libical could be enabled if necessary. After Thunderbird 78, libical will be removed completely.

Tracked in bug 978570.

9.2. Performance improvements [TB78/TB2021]

The calendar is eating a lot of the Thunderbird resources. Move calendar to its own process (or similar, framescripts?), and figure out how it would communicate with the rest of the application then.

Requires investigation of feasibility.

9.3. Integrate Lightning to core [TB78]

Calendar is de facto an integrated part of the Thunderbird functionality. Lightning would be fully merged into the code base and no longer treated as an add-on.

Tracked in bug 1493008**.**

9.4. XPCOM removals [TB78/TB2021]

Remove use of all calendar XPCOM components and replace them with simple JavaScript classes if possible. Reduce or eliminate XPCOM dependencies.

Ongoing effort.

9.5. Self contained Calendar and Task tabs [TB2021]

Make the calendar and tasks tabs self contained.

Compose window redesign and improvements [TB78]

- Implement a more moderning addressing experience: bug 440377.
- oEmbed/Twitter Card/Open Graph support: bug 1572648
- Support markdown - and extensions to it allowing nice graphs etc.: bug 1280912

10. Chat

10.1. OTR support [TB68/TB78]

To facilitate distributed strong end-to-end encrypted instant messaging, Thunderbird is adding support for the OTR protocol.

Tracked in bug 954310.

10.2. Chat UI facelift

The chat UI needs to be modernized and less IRC-centric. IRC and similar are commonly full page, while many 1:1 communications are often in a smaller space or smaller window. A sidebar to have in any windows could be nice for such cases of communication.

10.3. Matrix support

The Matrix network should be properly supported , allowing a Thunderbird teams type setup.

Tracked in bug 1572636.

10.4. Support for Mozilla IRC replacement

Mozilla is going to stop using IRC for developer communication. The decision on what's going to replace hasn't yet been made. If it's an open system, Thunderbird should ensure we have support for it. Matrix is one of the conceivable alternatives that are open.

11. Smart workflow widgets [TB78/TB2021]

Thunderbird should support smart widgets for voting, time scheduling. These could be built around micro-formats and for teams using Thunderbird only everything would work without problems (all over email). If someone uses another client, some manual actions would still be required. Fallback would still be pure text, so adding more structural data could only be a winning situation.

One way this would work for events, is that we mark up the emails we send out with some structural data, say an hCalendar. When replying to a message, we check if the original mail had included an hCalendar, and can offer quick options to reply and including that for that hCalendar event you are an attendee. It can also link to the UUID of an existing event in your calendar and offer response options related to that, e.g. check if you'd have conflicts in your schedule.

For things like simple votes, be smart and let a [vote](maybe as Emoji) in the subject mean there's a vote going on. Upon request by the user, create a voting summary and clean it up into a nicely formatted output. Automate the things that take some time to copy-paste from many mails, but could easily be automated. Make the Thunderbird users look good.

12. Mail data utilization and integration [TB78/TB2021]

- detect structured data (JSON-LD), allow acting on that data (see schema.org, xref KItinerary) — e.g. let you add flight data into your calendar directly
- detect micro-formats [related to smart-widgets]
- detect the most common free texts (especially: times and dates), and allow inline viewing of calendar schedule for that time, or contact information for a know contact in your address book etc.
- improved support for drag-n-drop as well as copy-paste of many data types: mail data, events,

tasks, contacts

13. Data fetcher [TB???]

Support the use case where a mail arrives only to give you a notification and a link. Allow Thunderbird to go grab the relevant data from that link and put it back into a message for convenient reading and archiving.

This is enabling the global data stream to be inside Thunderbird without much knowledge about the external forums etc. The feature will require some technical knowledge at least at first, but recipes on how to get data for various sources could later be created and shared by power users.

14. Data access: import/export [TB78]

Add real support to import/export profiles, or single accounts. Look at the import/export tools add-on for inspiration.

15. IMAP notes [TB2021]

Storing text notes on the IMAP server would be useful, and the feature is low hanging fruit not necessarily very hard to implement.

Tracked in bug 713843.

16. Improved mailing list handling [TB78]

Better support for mailing list handling:

- Mailing list management UI
- filtering to folders, autodetect lists and offer to set up
- handle the case of 2 mailing lists being the recipient better

17. Filtering rewrite in JavaScript [TB78]

Rewrite into JavaScript. Move to async operations once that’s done.

Tracked in bug 1574136.

18. Extensions [TB78]

Only WebExtension type extensions will be supported. We will continue to add useful WX APIs to support this effort. Traditional “full” access to everything can only be done through a Thunderbird Web Extension Experiment. When a WX API is available, add-ons must use that instead. Experiments should be considered just that, and add-on authors are encouraged to work with us to enable WX APIs they need. Experiments can and will break - it’s fully up to add-on authors to keep up with these changes.

19. Troubleshooter [TB2021]

One of the main problems users have is that they can't send/receive emails. Add a trouble shooter that could help diagnose the problem.

20. Performance improvements [TB78 and beyond]

Set up performance measuring. Investigate performance bottlenecks and improve UI responsiveness and application startup time. Concrete goals to be set up once we have more information.

21. Improved system integration [TB78]

- system notifications (windows) - also improved notifications in general (XXX refs to bugs)
- irc:// link support. Bug 953953
- mid: links to be fully supported and Thunderbird should register itself to open mid: links - would be super helpful for cross referencing of mails, events, and tasks, even from other applications. Bug 264270.
- calendar: handle clicking .ics events attachments internally, and opening from file system: bug 357480.

22. Enterprise features [TB???]

- cert deployment?
- XMPP/Jingle support. Bug 1018060 [If we can get sponsorship?]

23. TOR uplift for Thunderbird [TB???]

Consider an option to connect through Tor. Some mail providers support connecting through their onion services.

For the general Tor uplift effort at Mozilla, see https://wiki.mozilla.org/Security/Tor_Uplift/Tracking

24. Releng

24.1. Single hg repo for Thunderbird [TB78]

Thunderbird will move to use a single repository for all the code to stop all the problems that have to do with the mozilla-central comm-central split. Instead Thunderbird code will live in a separate repository that is based on mozilla-central but has a comm branch and a comm/ subfolder.

24.2. Own l10n repository [TB78]

Thunderbird localization will start using its own localization repository. This will allow us to do string changes in point releases when needed. This is currently not possible due to technical constraints in how the localization system is set up.

24.3. Minimize releng related differences compared to Firefox [TB78]

To make it easier to follow Firefox release engineering changes, we try to further align with the Firefox release setup. The single repository will help in that we don’t need to check out more than one repo. Ideally only the repo url and some config variables would have to be different, except for product specific build needs.

25. Web Site

Make the web site more marketing focused with stories on how to best utilize various features in Thunderbird and highlight workflows.

```
_______________________________________________
Maildev mailing list
Maildev@lists.thunderbird.net
http://lists.thunderbird.net/mailman/listinfo/maildev_lists.thunderbird.net
```