Send Options (UI sketch)   Full proposal P-T1 by Thomas Düllmann 2015-10-10

Use the following settings to specify in which text format messages composed in HTML format should normally be sent.

Default Message Delivery Format

When composing HTML messages, start with the following delivery format: [Restore Default]

- *Auto-Detect*
- Plain Text Only (ask before removing HTML) {to avoid silent, unexpected dataloss, we can only offer this option if we enforce asking at least for all cases of (!convertible::plain && !convertible::altering); we could use relevant send options below to derive user's wish wrt degree of acceptable lossy conversion}
- Rich Text (HTML) Only
- Plain and Rich (HTML) Text

Note: You can specify a different default message delivery format for each identity in the account settings.

Settings For Delivery Format Auto-Detect

Treat recipients whose text format preference is unknown as recipients who prefer to receive messages formatted as: {prefers-unknown} [Restore Default]

- Plain text
- *HTML*
- Both plain text and HTML

Note: Use the address book to specify preferred text formats for recipients.

For messages composed in HTML format, if all recipients prefer to receive messages formatted as plain text: {allPlain} [Restore Default]

- Ask me what to do {even for convertible::plain unless auto-downgrade is enabled below}
- Try converting to plain text, otherwise ask {ask if (!convertible::plain && !convertible::altering)}
- ~~Convert the message to plain text (removes all HTML!)~~ {this silent/uncontrolled dataloss case must be removed! It does not make sense to compose with consequential, potentially extensive HTML formatting and then default(!) to silently dumping all formatting, inline images etc. when sending. Compose with plaintext editor instead, or don't use formatting.}
- Send the message in HTML anyway
- *Send the message in both plain text and HTML*

Automatically use plain text delivery format for simple messages [Restore Default]

[X*] When there's virtually no HTML formatting, send message as plain text: {convertible::plain}

- only when all recipients prefer plain text
  {allPlain}
- *even when some recipients prefer HTML, or plain text and HTML*
  {allPlain; mixed: somePlain & someHTML; some-unknown-both}
- even when all recipients prefer HTML, or plain text and HTML
  {allPlain, mixed: somePlain & someHTML, some-unknown-both; all-unknown-both; allHTML}

HTML Domains

Recipients of the following domains prefer to receive messages formatted as HTML, unless specified otherwise in the recipients' contact properties.

Plain Text Domains

Recipients of the following domains prefer to receive messages formatted as plain text, unless specified otherwise in the recipient's contact properties.

[Note: For the actual UI, Dropdown boxes will be used instead of bullets, so the real options layout will be much cleaner, more intuitive and condensed than this functional sketch.]

Implementation Details (Send Options, Proposal P-T1)

➔ **New global pref for default per-message delivery format** (allow users to bypass auto-detect); ideally augmented by optional **per-identity-pref** (different accounts/identities→different needs; allow using global pref)

➔ **Eliminate massive silent dataloss behaviour from Delivery-Format: Plain (from menu) and Send-Options: Plain and recipients: allPlain; ask as required, at least for convertible > altering.**

➔ **New global pref for disambiguating prefers-unknown:**
prefers-unknown == **(a) Plain | (b) *HTML* | (c) Both**
→ **see auto-detect delivery formats results table below**

➔ **New global pref to enable/disable auto-downgrade before auto-detection;**
user can freely choose recipients types for whom that pref will be applied:
auto-downgrade== true | false   for one of the following recipient type sets:

- allPlain
- allPlain | mixed (may include *some* unknown-prefers-both, but not *all*Both)
- allPlain | allBoth | allHTML

if (auto-downgrade-pref==true && convertible::plain)
→ send plaintext for applicable recipient types
(auto-detect results per table below will be bypassed)

➔ **Various modifications of options layout and labels**

Result Table for Delivery Format Auto-Detect (Proposal P-T1)

Note: This table is only for cases of
auto-downgrade-pref==false ||
(auto-downgrade-pref==true && !convertible::plain)
i.e. auto-downgrade disabled OR
(auto-downgrade enabled AND HTML formatting > convertible::plain)
(when pref==true && no formatting, table not used for applicable recipient combinations; instead, plaintext will be sent, bypassing Auto-Detect)

Recipients' preferences	Proposed	Current
1. all plaintext	→ SendOptions [only for allPlain] (default_html_action pref): - Ask - *Try converting to plaintext, otherwise ask* if (!convertible::plain && !convertible::altering) - HTML - Both	→ plaintext (silent, hardcoded/forced *full* dataloss downgrading)
2. all html	→ HTML	→ HTML
3. all unknown a) ==allPlain: b) ==allHTML c) ==allBoth	 → use SendOptions (case 1) → *HTML (case 2)* → Both	→ SendOptions [for mixed/all unknown] - Ask (only when !convertible==plain) - plaintext (silent *full* dataloss downgrading) - HTML - *Both*
4. unknown & plaintext a) ==allPlain b) ==HTML & plain c) ==Both & plain	 → use SendOptions (case 1) → *Both* → Both	→ SendOptions [only for mixed] - … (see above) - *Both*
5. unknown & html a) ==plain & HTML b) ==allHTML c) ==Both & html	 → Both → *HTML (case 2)* → Both	→ SendOptions [only for mixed] - … (see above) - *Both*

Benefits of Full Send Options Proposal P-T1 by Thomas Düllmann 2015-10-10

1. Allow user to set a **global preference for the default send format of *every* message** (message-centric vs. recipient-centric). Iow, allow user to bypass all the age-old cruft which nobody really uses these days, to define the preferred delivery format for each recipient and go into guessing games how it will be sent by auto-detect. Instead, have a stable default setting in the menu which can be changed on a per-message basis as before. (But for the peace of mind of some developers, we will still have a default option to auto-downgrade to plaintext when there's virtually no formatting in an HTML composition). Ideally, because different accounts/identities have different needs, we should make this a **per-identity preference** (where each identity defaults to **"use global preference" option** in the list of possible default delivery formats).
2. **Prevent silent massive dataloss of styling, inline images etc.** for cases where all recipients prefer plain **(allPlain)**, but HTML formatting > convertible::Altering. (composing full HTML and then dumping upon sending does not make sense as a default behaviour, so at least we need to ask the user for that very rare case).
3. **Make silent, datalossy HTML-to-Plaintext behaviour optional** (iow: allow user to switch off forced auto-downgrade before auto-detect entirely) and allow user to fine-tune the types of recipients for which silent datalossy behaviour should be accepted (type selector for auto-downgrade).
4. **Disambiguate "prefers-unknown" cases** – let user decide how to interpret unknown (prefers-plain | prefers-html); allow explicit "unknown==prefers-both" which is also current default, but now explicit and reliable.
5. **Disentangle send options (allPlain vs. mixed) and rectify their description, visual hierarchy and effects.**
   - Send options now only apply when all recipients prefer plaintext (but user can freely define "unknown==plaintext" to make send options apply to mixed cases of plain & unknown).
   - Mixed cases involving "unknown" are now more configurable but also come with some sane, fixed defaults. Altogether, much more transparent and predictable. If one recipient prefers-plain, and the other prefers-html, the logical thing to do is to send multipart/alternative (both plaintext+html), right? Otherwise "prefers-html" would be pointless.
   - Improve/correct captions and visual hierarchy to be more transparent how the settings interact. E.g., most options including Html and plaintext domains are only effective with delivery-format: auto-detect. So we must have headlines and visual arrangement reflecting that. Also, clarify the meaning of "prefers-plain" to mean just that ("prefers-plain", not "message will definitely be sent as plain-only not matter what").

   Problem of current cross-type send options: Per current wrong label, send options should effect the same behaviour for all cases of "allPlain", "allUnknown", and "mixed", which is nonsense. In reality, current send options are only effective for "all unknown" or "mixed" (plain & html; plain & unknown; html & unknown). But then, some choices have never made sense for that conglomerate of different recipient types.

   - "Ask" is a bad default because it bothers the user all the time for "unknown".
   - "Plaintext" is usually a bad default for mixed cases, especially when involving silent, potentially massive dataloss (styles, inline images etc.). We must ask before non-trivial dataloss (says Magnus, too), that's why I changed massive dataloss option "silently send plaintext no matter what" into "try trivial downconvert, otherwise ask".
   - "HTML" is unusable/bad default for cases like "allPlain"; bad for "mixed" with plain.
   - "Both" is also not the best default for a number of cases like "allPlain" or "all unknown==html", as we are increasing the message size by sending multipart/alternative without need. Which are many cases because most recipients in AB are unknown, and unknown in 2015 mostly means "can receive HTML without any problem". So allowing user to just send HTML is much better.
6. **All of these powerful options nicely tucked away** in send options dialogue, and implemented with an **intuitive layout of just 4 simple options with dropdown boxes.**