

Firefox data preference changes

[Current state](#)

[Backing preferences](#)

[Firefox client changes](#)

[Open questions](#)

Current state

This is how the data settings look now in the preferences UI in Firefox:

Nightly Data Collection and Use

We strive to provide you with choices and collect only what we need to provide and improve Nightly for everyone. We always ask permission before receiving personal information.

[Privacy Notice](#)

- Allow Nightly to automatically send technical and interaction data to Mozilla [Learn more](#)
- Allow Firefox to install and run studies [View Firefox Studies](#)
- Allow Nightly to send crash reports to Mozilla [Learn more](#)

There are two notable changes here:

- As of Firefox 56, we dropped the option for "extended Telemetry" (bug no.?).
- The SHIELD study option is a new addition - they are actually injected by the SHIELD system add-on [at runtime](#).

Backing preferences

So, what about the [actual settings](#) behind that, that control our data collection in the background?

datareporting.healthreport.uploadEnabled

That preference should have seen no change. The "send technical and interaction data" setting should control this as before.

This is our main preference that controls whether Firefox can send data and follows the user choice.

toolkit.telemetry.enabled

This preference controls previously controlled our "extended" Telemetry collection. With the data preferences simplification, this effectively becomes a "can we send pre-release data?" preference.

We want this to behave in the following way:

- On release: it should always be off or false.
- On pre-release (beta, nightly): it should always be on.

This pref has a few consumers, so we can't easily remove it and have to keep it working correctly.

We can consider replacing or obsoleting it in the future.

Current handling of Telemetry pref defaults

In our build system, we enable `MOZ_TELEMETRY_ON_BY_DEFAULT` for all builds that are not on Release or Beta ([code](#), using `RELEASE_OR_BETA`). This gets picked up in [Preferences.cpp](#) and is overridden for the "beta" channel specifically.

A side-note:

As a consequence, the current Developer Edition does not have Telemetry on by default. They moved to building from beta, so `RELEASE_OR_BETA` is set, but the update channel is set to "aurora". That means that the override in `Preferences.cpp` probably does not work for them. A fix of this behavior could be based on `MOZ_DEV_EDITION` or similar.

Firefox client changes

The changes mean that we want to do the following changes for Firefox 58:

- Force-lock or override the `toolkit.telemetry.enabled` pref (from `Preferences.cpp`?).
- Change `nsITelemetry.canRecordExtended` to
 - always return `true` on pre-release, if `canRecordBase` is true,
 - always return `false` on release, if `canRecordBase` is true,
 - but allow overriding this for testing through the hidden pref below.
- Migrate tests to use a different, hidden pref, say `toolkit.telemetry.testing.prerelease`.
- Update our documentation with the new terminology, semantics and pref changes.
 - This affects the [client docs](#), the [new probe guide](#) and the [data docs](#).

Later we can follow up with more cleanup:

- Update `nsITelemetry` to match the new behavior & terminology.
 - Obsolete `canRecordBase` & `canRecordExtended`
 - Instead introduce `canRecord` & `canRecordPrerelease`, ...
- Introduce new, more clearly named preferences.

- Update probe registries to match the new behavior & terminology.
 - Affects *Histograms.json*, *Scalars.yaml*, *Events.yaml*.
 - They should use a boolean property *collectOnRelease* now (instead of the current *releaseChannelCollection:"opt-out"/"opt-in"*).
 - This affects both pipeline and client. We need to first prepare the pipeline, then migrate the client registries.

Open questions

- Do we still need the *toolkit.telemetry.enabled* pref to be toggled on/off, regardless of channel?
 - We use it for testing, but we can migrate that to using a new hidden pref.
 - Do consumers like Test Pilot or SHIELD need to toggle *toolkit.telemetry.enabled* when active, even on release?
 - Test Pilot apparently uses it read-only.
- Can we override or permanently lock prefs?
 - This is needed for non-disruptive updates, making *toolkit.telemetry.enabled* effectively an *"is this prerelease?"* pref.
 - Can we force-override or lock this pref from configs or through *Preferences.cpp*?
- How to deal with the TMO dashboards?
 - Currently they show the 1% opt-in population from the release channel.
 - Short-term we can just remove the release population.
 - Longer-term we can think about whether we can show the opt-out release data, but that may need some legal/policy/BD review.
- Did the Android preference UI change too?
 - If not, are we still good to apply the same mechanics there?
 - Any concerns from the Android team on having the pref force-locked?