Adding a new Telemetry ping type

Telemetry has the ability to report data for systems outside of the main Browser Telemetry that powers telemetry.mozilla.org. Some examples of these “external” documents are heartbeat and testpilot.

General Considerations

- What type of data will go into the pings?
- What population will be reporting data? The opt-in “extended” Telemetry population (primarily pre-release) or the opt-out “base” population (most clients on all channels)? Or a different, custom-defined population?
- It is strongly recommended to define a schema for the payload for both client- and server-side validation purposes.

Client code

Pings are queued for submission using the “TelemetryController.submitExternalPing” function in the browser. These documents are encapsulated in the “payload” field of the common Telemetry ping format.

Considerations for adding a new document type include:

- At what frequency will the pings be submitted? Daily? Triggered by some user action?
    - The ping frequency has effects on how long it takes for collected data to arrive on the server for analysis. More frequent generally means less latency, though there are limitations on how frequently data should be submitted (see below).
- What will go into the pings? Note that pings larger than 1MB will not be sent.
- Should they include the Telemetry environment?

Any new code that submits a new ping needs data collection review from a Data Peer (:bsmedberg).

Ping-sending limitations

The client throttles ping submissions in order to avoid impacting the user experience while sending data. The main limitations to keep in mind are that the rate of ping-sending is limited to 10 per minute (as of June 1, 2016), storage for pending pings is limited to 15MB, and the maximum size of a single ping is 1MB.

If you exceed the throttling rate for extended periods, the client will start discarding collected data once the storage limit is reached. The oldest pings are discarded first when this happens. This can cause "gaps" in a client's data record.

Documentation

You should document the format and behavior of the new ping:

- What contents does the ping have?
- What structure and format does it have?
- What is it used for?
- When is it sent and under what constraints?

This can happen in-tree (rendered on ReadTheDocs) or in an alternative location (like the Wiki or the corresponding GitHub repo).
You should also list your ping in the table of known pings.

Testing

There are various ways to confirm that your code is working as expected.

All submitted pings are archived locally in Firefox. You can use `about:telemetry` to inspect the submitted pings.
If you need to inspect details about what's going on with Telemetry, you can set the pref `toolkit.telemetry.log.level` to `0` for detailed output in the browser

To inspect the outgoing pings in a local setup, you can use the gzipServer. This allows running a local server and validating the outgoing submissions.

Finally, you should monitor your incoming pings in the data pipeline (e.g. through self-serve analysis). This confirms that your pings are actually received and routed correctly through the pipeline. It also allows to look up data per client and inspect whether pings are missing etc.

Server code

Pings submitted via the above client code are available for analysis using the Spark infrastructure at analysis.telemetry.mozilla.org. Once a particular record is sent by the client, it is typically available for analysis within one hour.

By default, new pings will be bucketed with all non-standard pings into an "OTHER" category, which takes a little more effort to fetch them for analysis jobs. To avoid that, file a bug in the *Metrics: Pipeline* component in bugzilla requesting that a new Telemetry document type be added. This will make it easy to filter for only the new document type (splitting it out of the "OTHER" category).

Server-side considerations include:

- Estimate of daily data volume: *# of users * ping frequency * ping size*
- Will any special payload decoding be necessary?
- Strongly recommended to define a schema for the payload for validation purposes. This will likely soon become required.
- Will any derived datasets be needed (for accessing the data via sql.t.m.o, for example)?
- What sort of reporting / analysis will need to be done on this data?