**Pages**

background.html: This page has almost no html code in it. It basically just calls the background.js script which in turn opens the mainpanel.html

mainpanel.html: This is its own window that the user interacts on the system with. It contains basically all the scripts/background and scripts/common scripts. It additionally contains some of the addon and lib files.

**Background**

background.js: this script is always running. It is the page that opens the mainpanel.html

main\_panel.js: \_\_\_\_\_\_\_\_\_

panel.js: \_\_\_\_\_\_\_\_\_

port.js: \_\_\_\_\_\_\_\_\_

script\_server.js: \_\_\_\_\_\_\_\_\_

simple\_record.js: \_\_\_\_\_\_\_\_\_

trigger.js: \_\_\_\_\_\_\_\_\_

url.js: this script compares the similarity between two urls

**Common**

common.js: this script has three JSON objects- (1) RecordState, (2) ReplayState, and (3) Ack that is shared between the content scripts and background backpage.

logging.js: \_\_\_\_\_\_\_\_\_

params:js: \_\_\_\_\_\_\_\_\_

scope.js: this script is one line that creates an empty object. I’m not sure if this is even used anywhere.

**Content**

compensation.js: This script contains a list of delta functions that compares the differences between two snapshot nodes. Used for the filtering process and for the replay process. Also checks for the existence of both of the nodes.

content\_script.js: \_\_\_\_\_\_\_\_\_

dom.js: This script has various functions that converts a node to an xpath expression. And from xpath expressions to a node.

injected.js: \_\_\_\_\_\_\_\_\_

port.js: This adds an additional layer/wraps around the existing Chrome api for ports and long-lived messages

simple\_record.js: \_\_\_\_\_\_\_\_\_

snapshot.js: Saves the properties of a node, nodes, and also has a function that makes a tree of their relationships.

target.js: \_\_\_\_\_\_\_\_\_