

RulesProjPromotion

It looks like rules were, for a time, maintained in a separate branch. This page is related to that time, and not currently correct. It looks like automatic rule promotion currently happens when "make" is run in the masses/ directory.

Rules Project: Promotion of Rules

from the sandboxes to the core ruleset, that is.

(this page split from [RulesProjSandboxes](#), part of [RulesProjectPlan](#))

Getting rules from the sandbox, into the distribution:

- each user gets their own sandbox as discussed on [RulesProjSandboxes](#)
- checked-in rules in the sandboxes are mass-checked in the nightly mass-checks
- to migrate a rule from "sandbox" (dev) to "core" (production) ruleset uses C-T-R; ie. votes are not required in advance
- also C-T-R to migrate from "sandbox" to "extra" ruleset

Rules that get promoted from a "sandbox" to "core" should pass the following criteria:

- pass "--lint"!
- S/O ratio of 0.95 or greater (or 0.05 or less for nice rules)
- > 0.25% of target type hit (e.g. spam for non-nice rules)
- < 1.00% of non-target type hit (e.g. ham for non-nice rules)

These numbers are really just ball-park figures and should be fine-tuned as we go. (DuncanFindlay) We can automate those criteria pretty easily. We can also vote for rules that don't pass those criteria, but we think should be put into core for some reason.

2005-11-17: These are now in place. Promotable rules show up in dark text on the [RuleQaApp](#), non-promotable rules (at least not without a vote) show up in lighter text.

Further criteria:

- not too slow 🐢 TODO: need an automated way to measure that, probably derived from the timing plugin in bug 4517.
- TODO: criteria for overlap with existing rules?
- TODO: Scores. How are we going to score stuff that's pushed out with sa-update? (I can't see any documentation on it at the moment) Is there going to be logs kept and perceptron runs or simply votes for interim scores based on the mails they target.

"Promotion" is a manual process, where the rule developer copies the lines for that rule from one file to a file in the 'core' directory. The promotable rules will be highlighted in some way in the rule-QA app.

Moving files out of trunk into the new rules project

(this has taken place. This section is now of historical interest only.)

Code-tied rules stay with main tree in current rules directory, with the exception of 25_replace.cf which is really just another way to write body/header rules. Basically, the static stuff that is tied to code does not move to the rules project. Everything else moves.

In more detail – files that DO NOT move to rules project:

[JustinMason](#): note that SVN paths are listed as "ROOT/rules/trunk". This is the trunk; by having that, it allows branches of the rules project at e.g. "ROOT/rules/branches/vX.Y.Z", similarly to how the code SVN repo has trunk and branches. (As to what way exactly we'd branch, versions, etc. let's see how that develops in the future.)

```

25_accessdb.cf      (plugins in core code)
25_antivirus.cf
25_dcc.cf
25_domainkeys.cf
25_hashcash.cf
25_pyzor.cf
25_razor2.cf
25_spf.cf
25_textcat.cf
25_uribl.cf
60awl.cf
60_whitelist_subject.cf
20_dnsbl_tests.cf (eval tests in EvalTests.pm)
20_html_tests.cf (rawbody ones can move to ROOT/rules/trunk/core/)
20_net_tests.cf
23_bayes.cf
60_whitelist.cf
init.pre           (Misc non-cf files)
local.cf
name-triplets.txt
regression_tests.cf
triplets.txt
user_prefs.template
v310.pre

```

Files that DO get moved:

```

25_body_tests_es.cf -> ROOT/rules/trunk/core/es/
25_body_tests_pl.cf -> ROOT/rules/trunk/core/pl/
30_text_de.cf       -> ROOT/rules/trunk/core/de/
30_text_fr.cf       -> ROOT/rules/trunk/core/fr/
30_text_it.cf       -> ROOT/rules/trunk/core/it/
30_text_nl.cf       -> ROOT/rules/trunk/core/nl/
30_text_pl.cf       -> ROOT/rules/trunk/core/pl/
30_text_pt_br.cf    -> ROOT/rules/trunk/core/pt_br/

20_advance_fee.cf   -> ROOT/rules/trunk/core/
20_drugs.cf         -> ROOT/rules/trunk/core/
20_p**n.cf          -> ROOT/rules/trunk/core/    [wikicensorship!]

10_misc.cf          -> ROOT/rules/trunk/core/
20_anti_ratware.cf  -> ROOT/rules/trunk/core/
20_body_tests.cf    -> ROOT/rules/trunk/core/
20_compensate.cf    -> ROOT/rules/trunk/core/
20_fake_helo_tests.cf -> ROOT/rules/trunk/core/
20_head_tests.cf    -> ROOT/rules/trunk/core/
20_meta_tests.cf    -> ROOT/rules/trunk/core/
20_phrases.cf       -> ROOT/rules/trunk/core/
20_ratware.cf       -> ROOT/rules/trunk/core/
20_uri_tests.cf     -> ROOT/rules/trunk/core/
25_replace.cf       -> ROOT/rules/trunk/core/ [code dependent, but these will change a lot]
50_scores.cf        -> ROOT/rules/trunk/core/
60_whitelist_spf.cf -> ROOT/rules/trunk/core/

```

(update: this is now complete.)

Algorithm for compilation

The ROOT/rules/trunk svn path is now the rules source directory.

The ROOT/trunk/rules svn path – ie "rules" in the [SpamAssassin](#) source tree – is the rules build output directory.

Rules are compiled from source dir to output dir. All rules in "core" are always promoted (for backwards compatibility).

Rules will be autorenamed, if there's a collision between a new rule name and one that's already been output by the compiler.