RuleLifeCycle

Rule Life Cycle

The life-cycle of a rule goes like this.

- A rule starts off in a developer's sandbox as an experimental rule, one that he doesn't want to publish just yet.
- Alternatively, it may be a non-experimental, but still in-sandbox, testing rule. These need to be marked by the developer with "tflags publish".
- The developer may decide to switch the rule back and forth between those two states.
- Non-experimental rules' promotability is measured (see SaUpdateBackend).
- If it's good enough, it's published to the "active set".
- A good rule may be manually copied from the sandbox to the "rules" directory.
- Eventually, it stops being good enough, through the normal attrition process for antispam rules, and it stops meeting the promotion criteria.

List Of Rule States

Rules in sandbox:

- **experimental** – don't promote me. "T_" prefix in the rulesrc source file, "tflags nopublish", or the absence of a "tflags publish", implies this. These rules are compiled, by the "build/mkrules" compiler at "make" time, to "rules/70_sandbox.cf".
- **s_poor** – promotable, listed with "tflags publish", but not meeting promotion criteria. Compiled to "rules/70_sandbox.cf". "T_" is prefixed to the rule name.
- **s_good** – promotable, listed with "tflags publish", and meeting criteria. Rules in this state are copied into the "active set". Compiled to "rules /72_active.cf".

Rules in the engine tarball:

- **core** – no promotion criteria are needed; this is part of the core ruleset. Often tied closely to the distributed `Mail::SpamAssassin` perl modules. These are not compiled at all by "build/mkrules", and are always distributed. (new as of bug 5123.)

Deleted rules:

- **gone** – rule has been deleted. If a rule scores badly in core for "an extended period of time", it goes here. (Right now, this has to be done manually.)

(History: mailing list message, bug 5123)

State Transitions

The permitted transitions for those rule states, therefore, are as follows:

- **experimental** <---> **s_poor**
- **experimental** <---> **s_good**
- (hand copy) **s_good** <---> **core**
- (hand copy) **core** <---> **gone**

List Of Build States

Some rules are only used from certain build states. Here are the list of states that SpamAssassin goes through, or that rules are packaged as, during various parts of its build process.

- **builddir**: "./spamassassin", or similar, run from inside build dir
- **make_test**: "make test"
- **mass_check**: MassCheck run from inside "masses" dir
- **bbtest**: the testing buildbot
- **bbmass**: the bbmass buildbot
- **nightly**: the NightlyMassCheck
- **make_install**: what's installed via "make install"
- **tarball**: what's put in distributed tarballs (by make dist, make disttest)
- **sa_update**: what's delivered via sa-update

Build States vs. Rule States Matrix

And here's the table listing what rules are usable, where. 👍 indicates that a rule in that state is indeed usable from the listed build state.

	experimental	s_poor	s_good	core
builddir	👍	👍	👍	👍
make_test	👍	👍	👍	👍
mass_check	👍	👍	👍	👍

bbtest	👍	👍	👍	👍
bbmass	👍	👍	👍	👍
nightly	👍	👍	👍	👍
make_install			👍	👍
tarball			👍	👍
sa_update			👍	👍