AutoQA origins

how it all started
AutoQA origins

There is a lot of stuff that Fedora QA team needs to test to ensure best user experience:

- install media
- anaconda installer
- basic desktop environment
- software (application/library) updates
- RPM package validity
- YUM repository validity
AutoQA origins

Unfortunately we would need to grow much more hands and heads to be able to tackle all of that. And still...

So let's be lazy – let's have a computer do that! We just need to write the proper software, that's all.

And then we can just sit and watch!

(Have you heard that the main motivator for most software projects is programmers' laziness? Now you know.)
AutoQA origins

A software called AutoQA.

Meaning automated QA taking care of everything needed. Meaning artificially-intelligent state-of-the-art Skynet-like system solving all our needs (morning coffee included).

Well, we’re not there yet.

But we’re closing.
Current state of AutoQA
Current state of AutoQA

- heavy development (surprise surprise!)

- packages available for Fedora and EPEL (in a third-party repository)

- documentation available, not always up-to-date

- results just sent to mailing list and (optionally) to package maintainers for now, database with API being prepared
Current state of AutoQA

- working tests that check:
  - anaconda installer
  - software updates
  - RPM package validity
  - YUM repository validity
Current state of AutoQA

July 2010 Archives by thread

- Messages sorted by: [subject] [author] [date]
- More info on this list...

Starting: Thu Jul  1 13:03:16 UTC 2010
Ending: Sat Jul 31 11:07:22 UTC 2010
Messages: 3437

- rats_sanitY FAIL rawhide-i386 autoqa at fedoraproject.org
- conflicts: 69 packages with file conflicts in rawhide-i386 autoqa at fedoraproject.org
- reproclosure: 47 packages with unresolved deps in rawhide-i386 autoqa at fedoraproject.org
- conflicts: 55 packages with file conflicts in f12-updates-i386 autoqa at fedoraproject.org
- reproclosure: 28 packages with unresolved deps in f12-updates-i386 autoqa at fedoraproject.org
- conflicts: 44 packages with file conflicts in f13-updates-testing-i386 autoqa at fedoraproject.org
- reproclosure: 6 packages with unresolved deps in f13-updates-testing-i386 autoqa at fedoraproject.org
- conflicts: 38 packages with file conflicts in f13-updates-i386 autoqa at fedoraproject.org
- reproclosure: 8 packages with unresolved deps in f13-updates-i386 autoqa at fedoraproject.org
- conflicts: 57 packages with file conflicts in f12-updates-testing-i386 autoqa at fedoraproject.org
- reproclosure: 29 packages with unresolved deps in f12-updates-testing-i386 autoqa at fedoraproject.org
- rpmlint: 0 errors, 10 warnings for package jokosher-1.0-0.12.20100703b7zr.fc13 autoqa at fedoraproject.org
- rpmguard: 0 warnings for package jokosher-1.0-0.12.20100703b7zr.fc13 autoqa at fedoraproject.org
AutoQA tests
AutoQA tests

Rpmlint - checks RPM package for common mistakes and packaging guidelines conformance using rpmlint tool

Stored logs available at <http://test1250.test.redhat.com/results/35576-autotest/brutus.test.redhat.com/>

jokosher.src: W: spelling-error Summary(en_US) multi -> mulch, mufti
jokosher.src: W: spelling-error %description -l en_US powerful -> powerful, Powell, powerless
jokosher.noarch: W: spelling-error Summary(en_US) multi -> mulch, mufti
jokosher.noarch: W: spelling-error %description -l en_US powerful -> powerful, Powell, powerless
jokosher.noarch: W: python-bytecode-without-source /usr/lib/python2.6/site-packages/Jokosher/Profiler.pyc
jokosher.noarch: W: python-bytecode-without-source /usr/lib/python2.6/site-packages/Jokosher/Profiler.pyo
jokosher.noarch: W: no-manual-page-for-binary jokosher

2 packages and 0 specfiles checked; 0 errors, 10 warnings.
AutoQA tests

Rpmguard - compares older package build available in some stable repository with the new build and prints interesting differences


N: Comparing wine-desktop-1.1.38-1.fc12 and wine-desktop-1.2.0-0.6.rc6.fc12 (archs: noarch) ...
W: requirement-added wine-core(x86-32) = 1.2.0-0.6.rc6.fc12
W: requirement-removed wine-core = 1.1.38-1.fc12
W: provision-added mimehandler(application/x-ms-shortcut)
N: 3 warnings
N: ----
N: Comparing wine-openal-1.1.38-1.fc12 and wine-openal-1.2.0-0.6.rc6.fc12 (archs: x86_64, i686) ...
N: ----
N: Comparing wine-common-1.1.38-1.fc12 and wine-common-1.2.0-0.6.rc6.fc12 (archs: noarch) ...
W: executable-removed /usr/bin/wine
W: executable-removed /usr/bin/wineprefixcreate
N: 2 warnings
N: ----
N: Comparing wine-ldap-1.1.38-1.fc12 and wine-ldap-1.2.0-0.6.rc6.fc12 (archs: x86_64, i686) ...

AutoQA tests

**Initscripts** - performs LSB-compliance checks on packages containing initscripts

- only some are supported, but we are constantly adding support for more
AutoQA tests

**Upgradepath** - checks whether it is ok to push some package build to a selected repository and not break the ability to do “yum upgrade” afterwards

```
freeciv-3.0-1.fc12 into dist-f12-updates
================================
[ OK ] dist-f10
  Latest package: freeciv-2.1.6-1.fc10
[ OK ] dist-f10-updates
  Latest package: freeciv-2.1.9-1.fc10
[ OK ] dist-f11
  Latest package: freeciv-2.1.9-1.fc11
[ OK ] dist-f11-updates
  Latest package: freeciv-2.2.0-2.fc11
[ OK ] dist-f12
  Latest package: freeciv-2.1.9-2.fc12
[ OK ] dist-f12-updates
  Latest package: freeciv-2.2.2-1.fc12
[FAIL] dist-f13
  Latest package: freeciv-2.2.0-1.fc13
  Failed condition: Requested package <= Latest package
[FAIL] dist-f13-updates
  Latest package: freeciv-2.2.2-1.fc13
  Failed condition: Requested package <= Latest package
```
AutoQA tests

Repoclosure - checks for broken package dependencies in YUM repositories

---

Repoclosure: 102 pac...

Added target repo from http://download.fedoraproject.org/pub/fedora/linux/development/rawhide/x86_64/os
Reading in repository metadata - please wait....

Checking Dependencies
Repos looked at: 1
target
Num Packages in Repos: 21877
package: 1:gnome-bluetooth-2.90.0-5.fc15.x86_64 from target
  unresolved deps:
    libgnome-control-center.so.1()(64bit)
package: 1:gnome-games-extra-2.31.91.1-1.fc15.x86_64 from target
  unresolved deps:
    libclutter-gtk-0.10.so.0()(64bit)
package: PragMARC-20060427-6.fc13.1686 from target
  unresolved deps:
    libgnarl-4.4.so
    libgnat-4.4.so
package: PragMARC-20060427-6.fc13.x86_64 from target
  unresolved deps:
    libgnarl-4.4.so()(64bit)
AutoQA tests

Conflicts - checks for package and file conflicts in YUM repositories

== File conflicts, listed by conflicting packages ==
2:postfix-2.7.1-1.fc14.i686
exim-4.71-4.fc14.i686
ssmtp-2.61-15.fc14.i686
/usr/share/man/man1/mailq.1.gz

1:syncевolution-devel-1.0.1-4.fc14.i686
libgbus-devel-0.2-4.fc12.i686
/usr/include/gdbus.h

dconf-devel-0.5-2.fc14.i686
vala-0.9.4-1.fc14.i686
/usr/share/vala/vapi/dconf.vapi

perl-pip-1.16-2.fc14.noarch
python-pip-0.7.2-5.fc14.noarch
/usr/bin/pip

colordiff-1.0.9-3.fc12.noarch
rubygem-term-ansicolor-1.0.5-1.fc13.noarch
/usr/bin/cdiff
AutoQA tests

**Rats_sanity** - checks whether Rawhide compose is available and complete

![Example output of a test](https://fedorahosted.org/pipermail/autoqa-results/2010-August/030898.html)

- **CHECK:** <checksum> has valid data: OK
- **CHECK:** has valid data: OK
- **CHECK:** fetching RELAX NG schema for comps
- **CHECK:** comps syntax and grammar: OK
- **CHECK:** critical path dependency test:
  - expanding @core
  - 48 packages added
  - expanding @critical-path-base
AutoQA tests

Rats_install - performs an automated anaconda installation of a Rawhide compose

== kernel boot test ==
opening serial console
serial console at /dev/pts/0
polling for boot messages
kernel version 2.6.35-0.55.rc6.git0.fc14.x86_64 booted, initrd started OK
TEST RESULT: kernel_boot: OK

== anaconda loading stage2 test ==
polling for stage2 startup message
anaconda 14.12 stage2 has started
TEST RESULT: load_stage2: OK

== anaconda disk probe test ==
Starting minimon log monitor on port 60991
anaconda now in step setuptime
anaconda now in step autopartitionexecute
anaconda now in step storagedone
anaconda now in step enablefilesystems
storage setup completed OK
TEST RESULT: diskprobe: OK

== anaconda package installation test ==
Waiting up to 20 minutes for package installation to complete
anaconda now in step bootloadersetup
AutoQA tests

- and some more tests being written...
  - **Package sanity** - tests whether a new package build can be installed, upgraded, downgraded, removed, etc
  - **Depcheck** - complex test whether new package build will break something in a YUM repository or not
  - **Anaconda** - testing anaconda installation using standard DVD media
AutoQA usage
Envisioned usage...

- execute tests upon main Fedora test events
  - package updates, repo updates, new Fedora milestones/releases
  - warning relevant people or even blocking actions when something goes wrong
Envisioned usage...

- provide service for third-parties
  - Koji showing you test results after each new package build
  - Bodhi showing you test results for every proposed package update

<table>
<thead>
<tr>
<th>Package Acceptance Dashboard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>autofs-5.0.5-23.fc14</strong></td>
</tr>
<tr>
<td>Package sanity</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>Depcheck</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>File conflicts</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>Upgrade path</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>jkeating</td>
</tr>
<tr>
<td>rpmlint regressions</td>
</tr>
<tr>
<td>autoqa</td>
</tr>
<tr>
<td>kparal</td>
</tr>
</tbody>
</table>
Envisioned usage...

- schedule custom tests as required
  - “Hey, what about testing feature xyz for all Fedora packages, right now?”
- accept new tests covering different software aspects from contributors
  - thus making AutoQA better and better and better...
AutoQA benefits

a.k.a. “What's in it for you?”
AutoQA benefits

- **Package maintainers**: a whole range of tests executed on any new build or update proposal; fast notice if something’s wrong or suspicious.
- **Release engineers**: much harder to break repo, tests specifically designed to prevent that.
- **Developers**: some tests designed to inform about general software flaws (LSB compliance, etc).
- **Users**: more reliable updates (do I have to say anything more?)
Hacking AutoQA
Hacking AutoQA

A little bit of architecture:
https://fedoraproject.org/wiki/AutoQA_architecture
Hacking AutoQA

- different watchers watch for different events and call the autoqa harness
- hook definitions provide arguments to tests
- tests are scheduled with autotest-server
- autotest-server controls autotest-clients
- autotest-clients execute tests and return results
Hacking AutoQA

AutoQA tests consist of:

- **test script** itself - written in any language, pick your favourite
- **autoqa test object** - main entry point for the test, written in Python, running test script, collecting results
- **control** files containing some metadata - job scheduling, etc
class rpmlint(AutoQATest):
    version = 1 # increment if setup() changes

    def setup(self):
        util.system('yum -y install rpmlint')

@ExceptionCatcher("self.run_once_failed")
def run_once(self, envr, name, koji_tag, **kwargs):
    koji = autoqa.koji_utils.SimpleKojiClientSession()
    pkgurls = koji.nvr_to_urls(envr)
    rpmdir = os.path.join(self.tmpdir, 'rpms')
    os.makedirs(rpmdir)
    print "Saving RPMs to %s" % rpmdir
    for p in pkgurls:
        # fetch package to rpmdir
        print "Grabbing %s" % p
        localfile = os.path.join(rpmdir, os.path.basename(p))
        autoqa.util.grabber.urlgrab(p, localfile)
        cmd = 'rpmlint %s' % rpmdir
        result = 'PASS'
        try:
            self.outputs = util.system_output(cmd+" 2>&1", retain_output=True)
        except error.CmdError, e:
            result = 'FAIL'
            self.result = "FAILED"
            self.outputs = e.result_obj.stdout
Hacking AutoQA

AutoQA tests executed on localhost:

- autoqa post-koji-build
  --kojitag dist-f13-updates --local
  --test rpmlint icecream-0.9.6-2.fc13

- autoqa post-koji-build
  --kojitag dist-f13-updates --local
  --test rpmguard yum-3.2.28-3.fc13

- autoqa post-bodhi-update
  --title freeciv-3.0-1.fc13
  --target-tag dist-f13-updates
  --local --test upgradepath freeciv-3.0-1.fc13

LIVE DEMO
The Future
AutoQA future

Exciting things awaiting us:

- **ResultsDB**
  - all results stored here, well-defined API for querying from anywhere and anyone, acting as a service

- **Fedora Message Bus**
  - watchers periodically checking for changes are not good enough, we should use asynchronous message passing between all Fedora services (and also for public consumption)
AutoQA future

Exciting things awaiting us:

- Bodhi/Koji integration
  - let's show green/red fields of AutoQA results right beside a new package build status in Koji or a proposed new package update status in Bodhi

- moving generic Red Hat tests to Fedora
  - initscripts test is the first example, we want to continue in this approach
Help us
Helping AutoQA

Join us and help, if:

- you want to have something specific tested - just provide the script, we’ll help you with AutoQA integration
- you want to work with us on AutoQA core architecture
- you’re really bored over the long drizzly autumn nights
Helping AutoQA

Useful links:

- https://fedoraproject.org/wiki/AutoQA
- https://fedorahosted.org/autoqa
- https://fedorahosted.org/mailman/listinfo/autoqa-devel
- https://fedoraproject.org/wiki/QA:Package_Update_Acceptance_Test_Plan
Questions?
(answers not guaranteed)

Contact: kparal@redhat.com
Attributions

Impress template:
- courtesy of Emily Dirsh (fedoraproject wiki)

Cliparts:
- worktopussy: courtesy of Marcel 'ciscai' Trindade (flickr.com)
- walking horse outline: courtesy of 'warszawianka' (openclipart.org)