

Software Engineering
FOR Network Engineers

SASHA ROMIJN | @MXSASH | SASHA@DASHCARE.NL

Sasha Romijn

- ▶ Over a decade of professional software engineering
- ▶ Mostly contracted Python & Django (former Django committer)
- ▶ Main author of Internet Routing Registry Daemon v4 for NTT
- ▶ Write the Docs core team
- ▶ she/her

Why do
network engineers
need to know about
software engineering?

Adopting good practices early is cheaper

- ▶ Reduce dependencies on quickly/poorly written code
- ▶ Reduce fear of change
- ▶ Increase the bus factor

1. Use version control

- ▶ **git, mercurial, (svn, cvs, rcs)**
- ▶ **You don't need network**
- ▶ **Commit your changes**
- ▶ **Maybe use GitHub or Gitlab**
- ▶ **<https://try.github.io/>**

```
# set the right author name/email
git config --global user.name "full name"
git config --global user.email "name@example.com"

git init          # make a git repo
git add .         # add all current files
git commit -a     # commit all changes (editor opens)
git push <url>   # if you want to push changes (not required)
```

```
# <edit your local files>
```

```
# add any new files  
git add new_file.py
```

```
# commit all changes  
git commit -am 'Added IPv4 support.'
```

```
# if you want to push changes (not required)  
git push
```

2. Write tests

- ▶ Save time, reduce errors, be confident with change
- ▶ Start simple: one (integration) test can suffice
- ▶ Test the weirder scenarios
- ▶ Tests can grow with your code
- ▶ Find a bug? Write a test

EXAMPLE: IPV6 ROUTE SIZE STATS

```
URL="https://ftp.ripe.net/ripe/dbase/split/ripe.db.route6.gz"
```

```
ROUTES=`curl -s $URL | gunzip`
```

```
echo "$ROUTES" \  
| egrep 'route6:' \  
| cut -f 2 -d / \  
| sort \  
| uniq -c \  
| sort -n
```

EXAMPLE INPUT ROUTE OBJECT

```
route6:      2001:db8::/33
origin:      AS64496
mnt-by:      EXAMPLE-MNT
source:      RIPE
```

EXPECTED OUTPUT

1 116

1 122

2 120

[.....]

1098 42

3982 29

7022 32

11299 48

NON-TESTABLE CODE

```
URL="https://ftp.ripe.net/ripe/dbase/split/ripe.db.route6.gz"  
ROUTES=`curl -s $URL | gunzip`
```

```
echo "$ROUTES" \  
| egrep 'route6:' \  
| cut -f 2 -d / \  
| sort \  
| uniq -c \  
| sort -n
```

```
# Testable version

if [[ $1 == "-" ]]
then
    ROUTES=`cat` # read all data from stdin for testing
else
    URL="https://ftp.ripe.net/ripe/dbase/split/ripe.db.route6.gz"
    ROUTES=`curl -s $URL | gunzip`
fi

echo "$ROUTES" \
| egrep 'route6:' \
| cut -f 2 -d / \
| sort \
| uniq -c \
| sort -n
```

TEST INPUT

```
route6:      2001:db8::/33
origin:      AS64496
mnt-by:      EXAMPLE-MNT
source:      RIPE
```

```
TEST_INPUT=""...<last slide>..."
```

```
EXPECTED_OUTPUT=" 1 33"
```

```
TEST_OUTPUT=`echo "$TEST_INPUT" | ./dbcount.sh -`
```

```
if [[ $TEST_OUTPUT == $EXPECTED_OUTPUT ]]
```

```
then
```

```
    echo "TEST PASSED"
```

```
    exit 0
```

```
else
```

```
    echo "TEST FAILED"
```

```
    echo -e "expected output:\n$EXPECTED_OUTPUT"
```

```
    echo -e "actual output:\n$TEST_OUTPUT"
```

```
    exit 1
```

```
fi
```

ACTUAL SCRIPT OUTPUT

```
1 116
1 122
1 descr: $Id: route6:2001:67c:1010::-47,v 1.1 2010-05-20 13:01:19 nico Exp $
1 descr: $Id: route6:2a01:3f0::-32,v 1.2 2010-05-20 10:35:04 nico Exp $
2 120
```

[.....]

```
1098 42
3982 29
7022 32
11299 48
```


BREAKING OBJECT

```
route6:      2001:67c:1010::/47
descr:      DNSNODE Anycast RIPE
descr:      $Id: route6:2001:67c:1010::-47,v 1.1 2010-05-20 13:01:19 nico Exp $
origin:     AS8674
mnt-lower:  NETNOD-MNT
mnt-routes: NETNOD-MNT
mnt-by:     NETNOD-MNT
created:    2010-05-20T13:02:23Z
last-modified: 2010-05-20T13:02:23Z
source:    RIPE
```

EXPANDED TEST INPUT

```
route6:      2001:db8::/33
origin:      AS64496
descr:       $Id: route6:2001:67c:1010::-47,v 1.1 2010-05-20 13:01:19 nico Exp $
mnt-by:      EXAMPLE-MNT
source:      RIPE
```

TEST RESULT

TEST FAILED

expected output:

1 33

actual output:

1 33

1 descr: : route6:2001:67c:1010::-47,v [.....]

3. Consider continuous integration

- ▶ If you have a test and GitHub, this is doable
- ▶ Can be added on later

ENABLING CI CAN BE SIMPLE

```
version: 2
jobs:
  build:
    working_directory: ~/repo
    docker:
      - image: circleci/python:3.6.1
    steps:
      - checkout
      - run: ./test.sh
```

4. Work on readability

- ▶ Comment the how, not so much what
- ▶ How obvious is this code in a year?
- ▶ `a` is a terrible variable name and you can do better

```
URL="https://ftp.ripe.net/ripe/dbase/split/ripe.db.route6.gz"
ROUTES=`curl -s $URL | gunzip`
```

```
echo "$ROUTES" \  
| egrep 'route6:' \  
| cut -f 2 -d / \  
| sort \  
| uniq -c \  
| sort -n
```

```
U="https://ftp.ripe.net/ripe/dbase/split/ripe.db.route6.gz"
```

```
R=`curl -s $U | gunzip`
```

```
echo "$R" \  
| egrep 'route6:' \  
| cut -f 2 -d / \  
| sort \  
| uniq -c \  
| sort -n
```


5. Refactor

- ▶ Restructuring without behaviour change
- ▶ Requires not to be scared of change
- ▶ Correcting poor choices made initially
- ▶ Learn from hindsight

6. Don't get attached

- ▶ Code is volatile
- ▶ Software engineering sometimes means working for a day and discarding everything

```
# XXX We don't use this function anymore  
# <300 line function from 4 years ago>
```

```
# I think this is broken so I've disabled it  
# <10 lines of gibberish with syntax errors>
```

7. Try not to over-engineer

- ▶ Focus on the requirements existing now
- ▶ Sometimes balance is difficult

8. Write documentation

- ▶ Start with a README
- ▶ What does this do?
- ▶ What problem does it solve?
- ▶ How does it interact?
- ▶ What are known limitations?

The `dbstats.sh` script generates stats on the number of route6 objects for each prefix in the RIPE database. Upon running the script, it will output lines with two numbers, first number being the number of objects, second being the prefix length.

This was developed originally by Sasha Romijn in 2019 to serve as input for the process.

The included `test.sh` is a small test script.

The output is based on the latest RIPE DB split dump, so the output may be based on 24 hour old data.

It does not support IPv4.

9. Know your limitations

- ▶ Without enough experience or mentoring, there are limits to what you can write well. Don't bite off too much.


```
diff --git a/irrd/mirroring/mirror_runners_export.py b/irrd/mirroring/mirror_runners_export.py
index 3d2d6ab..8d03b72 100644
--- a/irrd/mirroring/mirror_runners_export.py
+++ b/irrd/mirroring/mirror_runners_export.py
@@ -61,6 +61,7 @@ def _export(self, export_destination):
         object_bytes = remove_auth_hashes(obj['object_text']).encode('utf-8')
         fh.write(object_bytes + b'\n')

+
+    os.chmod(export_tmpfile.name, 0o644)
+    if filename_export.exists():
+        os.unlink(filename_export)
+    if filename_serial.exists():
```

```

diff --git a/docs/admins/configuration.rst b/docs/admins/configuration.rst
index c9d374e..fb95dab 100644
--- a/docs/admins/configuration.rst
+++ b/docs/admins/configuration.rst
@@ -293,7 +293,7 @@ Sources
gzipped. If there is no serial information available (i.e. the journal is
empty) no serial file is produced. If the database is entirely empty, an error
is logged and no files are exported. This directory needs to exist already,
- IRRd will not create it.
+ IRRd will not create it. File permissions are always set to ``644``.
|br| Default: not defined, no exports made.
|br| Change takes effect: after SIGHUP, at the next ``export_timer``.
* ``sources.{name}.export_timer``: the time between two full exports of all
diff --git a/irrd/mirroring/mirror_runners_export.py b/irrd/mirroring/mirror_runners_export.py
index 3d2d6ab..8cf2b6e 100644
--- a/irrd/mirroring/mirror_runners_export.py
+++ b/irrd/mirroring/mirror_runners_export.py
@@ -11,6 +11,8 @@
from irrd.storage.queries import RPSLDatabaseQuery, DatabaseStatusQuery
from irrd.utils.text import remove_auth_hashes

+EXPORT_PERMISSIONS = 0o644
+
logger = logging.getLogger(__name__)

@@ -61,6 +63,7 @@ def _export(self, export_destination):
    object_bytes = remove_auth_hashes(obj['object_text']).encode('utf-8')
    fh.write(object_bytes + b'\n')

+    os.chmod(export_tmpfile.name, EXPORT_PERMISSIONS)
+    if filename_export.exists():
+        os.unlink(filename_export)
+    if filename_serial.exists():
@@ -70,6 +73,7 @@ def _export(self, export_destination):
    if serial is not None:
        with open(filename_serial, 'w') as fh:
            fh.write(str(serial))
+    os.chmod(filename_serial, EXPORT_PERMISSIONS)

    self.database_handler.record_serial_exported(self.source, serial)
    logger.info(f'Export for {self.source} complete, stored in {filename_export} / {filename_serial}')
diff --git a/irrd/mirroring/tests/test_mirror_runners_export.py b/irrd/mirroring/tests/test_mirror_runners_export.py
index bca0b00..d8de10d 100644
--- a/irrd/mirroring/tests/test_mirror_runners_export.py
+++ b/irrd/mirroring/tests/test_mirror_runners_export.py
@@ -1,9 +1,10 @@
+import os
from itertools import cycle, repeat

import gzip
from unittest.mock import Mock

-from irrd.mirroring.mirror_runners_export import SourceExportRunner
+from irrd.mirroring.mirror_runners_export import SourceExportRunner, EXPORT_PERMISSIONS
from irrd.utils.test_utils import flatten_mock_calls

@@ -39,10 +40,14 @@ def test_export(self, tmpdir, config_override, monkeypatch, caplog):
    runner.run()
    runner.run()

-    with open(tmpdir + '/TEST.CURRENTSERIAL') as fh:
+    serial_filename = tmpdir + '/TEST.CURRENTSERIAL'
+    assert oct(os.lstat(serial_filename).st_mode)[-3:] == oct(EXPORT_PERMISSIONS)[-3:]
+    with open(serial_filename) as fh:
+        assert fh.read() == '424242'

-    with gzip.open(tmpdir + '/test.db.gz') as fh:
+    export_filename = tmpdir + '/test.db.gz'
+    assert oct(os.lstat(export_filename).st_mode)[-3:] == oct(EXPORT_PERMISSIONS)[-3:]
+    with gzip.open(export_filename) as fh:
+        assert fh.read().decode('utf-8') == 'object 1 🐛\nauth: CRYPT-PW DummyValue # Filtered for security\n\n' \
+        'object 2 🌈\n\n'

```



Thank you!

@MXSASH

SASHA@DASHCARE.NL