Troubleshooting djbdns
Systematically Fix It Quickly

Presenter: Steve Litt

Presented at Greater Orlando Linux User Group

Wednesday, November 2, 2011

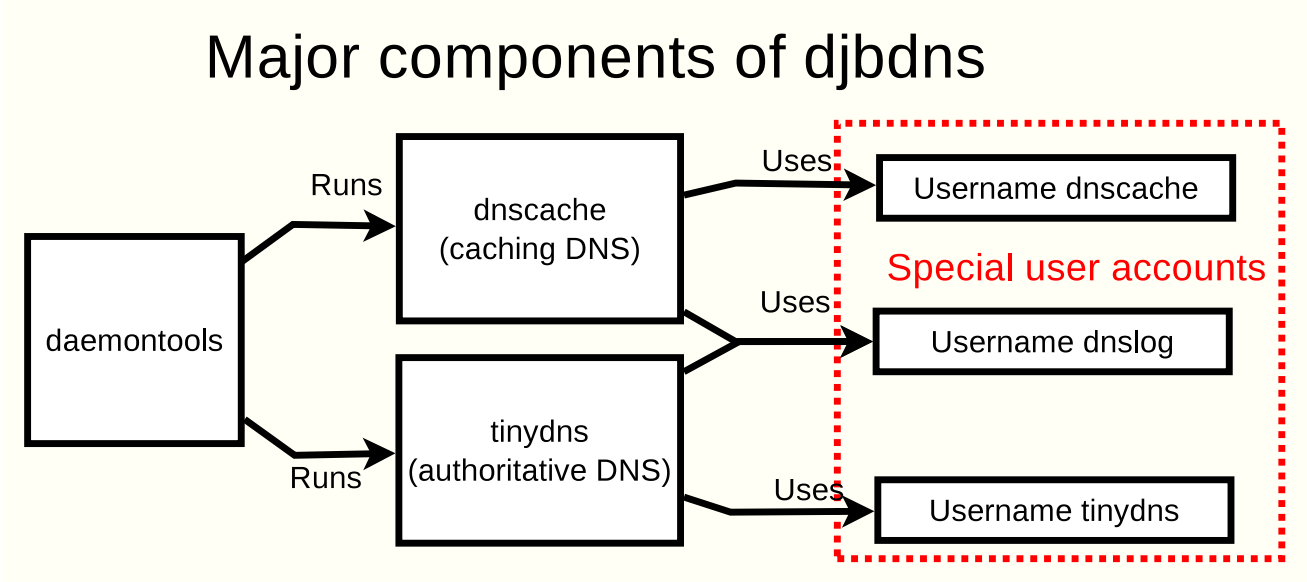
A Troubleshooters.Com® Publication.

See http://www.troubleshooters.com for more detailed information on troubleshooting and the *Universal Troubleshooting Process*.

What is djbdns?

- djbdns is a DNS server
- Translates IP addresses to domain names and vice versa
- Like Bind, but different
  - Separates resolver from authoritative server
  - More modular
  - In my opinion, more reliable
    - Only malfunctions when upgraded, installed or changed

At This Course's End You'll Be Able To:



- Name all elements of the Troubleshooter's Mindset
- Name all major components of a simple djbdns setup
- Narrow the root cause to a major component in a few minutes
- Run any major component by itself
- Understand the subcomponents of common major components
- Isolate to the subcomponent in a few minutes
- Solve the whole problem within an hour, 99% of the time

Troubleshooter's Mindset

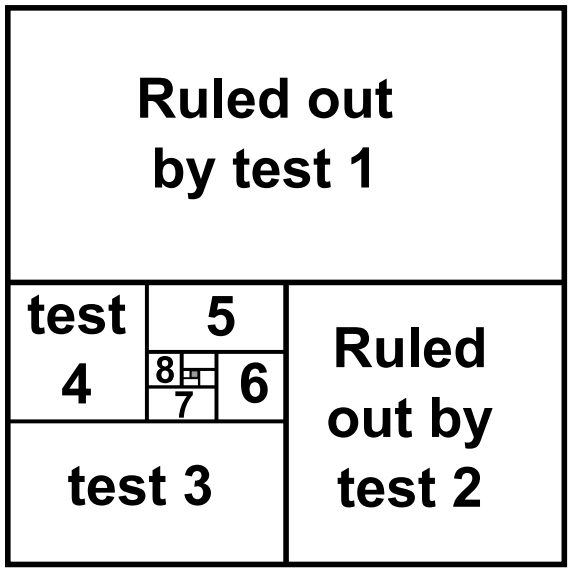
- Success depends on how you ***operate*** your brain
- It's not how much you know, it's how you organize it
  - Block diagram (sometimes called Mental Model)
- Look for the root cause
- Don't try to fix it, just try to narrow it down
- Always ask, "How can I narrow it down just one more time?"
- Stay rational...
- Choose diagnostic tests using the quadruple tradeoff...
- Use the 10 step universal troubleshooting process...

Troubleshooter's Mindset: Stay Rational

- Don't panic
- Don't get angry
- No emotion: Hunt the root cause like a cold, deadly predator.
- Feel free to ask for help.
- Be willing to give help.
- Don't fall victim to mental blocks.
- Don't be arrogant.

Quadruple Tradeoff

- Even divisions
- Ease
- Likelihood
- Safety



The Ten Step Universal Troubleshooting Process

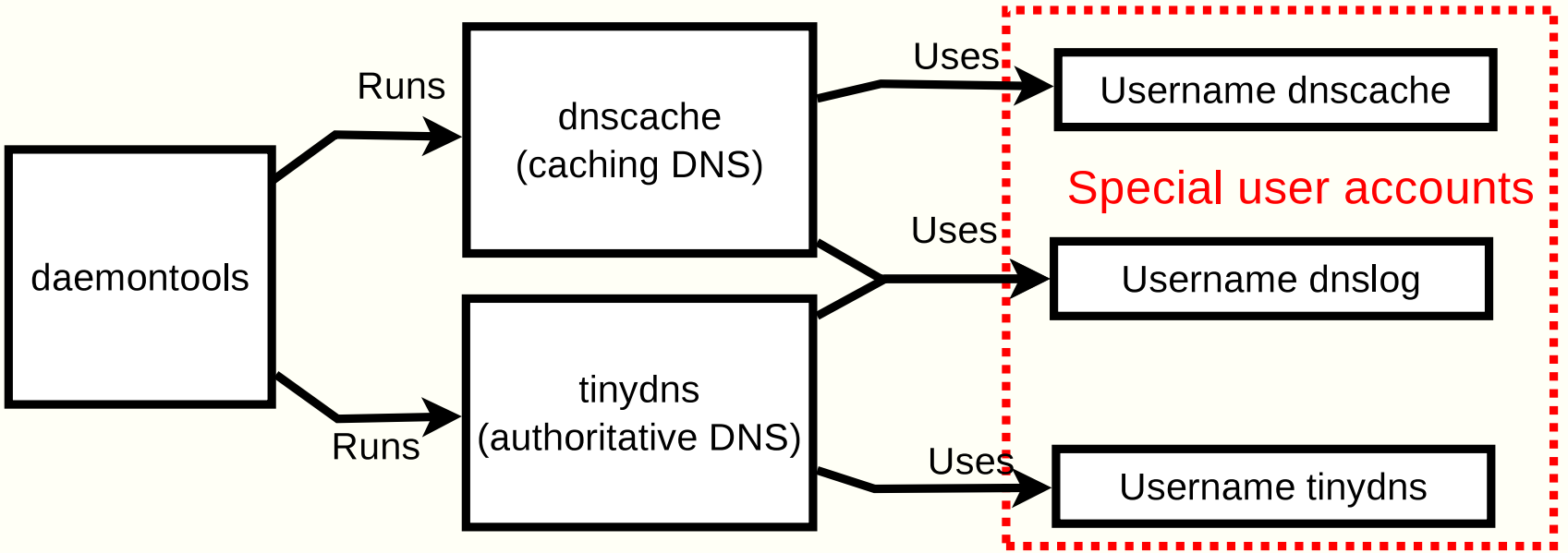
1. Prepare
2. Make a damage control plan
3. Get the symptom description
4. Reproduce the symptom
5. Corrective Maintenance
6. Narrow it down the root cause
7. Repair or replace the root cause
8. Test
9. Take Pride
10. Prevent future occurrence

Diagnostic Tests
Assumptions

- Local subnet is 192.168.100.0/24, local domain is domain.cxm
- djbdns installation is on mydesk.domain.cxm at 192.168.100.2
- You're working at mydesk.domain.cxm at 192.168.100.2
- Domain/subnet also has wincli.domain.cxm at 192.168.100.5
- DNS resolver (dnscache) is at 192.168.100.2
- Auth DNS (tinydns) for domain.cxm and subnet is at 127.0.0.1
- The directory scanned by svscan is /service
- The dnscache directory is at /var/djb/service/dnscache
- Enable with: ln -s /var/djb/service/dnscache /service
- The tinydns directory is at /var/djb/service/tinydns
- Google has a public DNS resolver at 8.8.8.8
- All commands should be done logged in as root. BE CAREFUL when working as root!!!

djbdns High Level Mental Model

Major components of djbdns



Diagnostic Tests:
Quick Preliminary High Level Diagnostic Tests

TEST	MEANING
id dnscache id tinydns id dnslog	Check for valid standard djbdns accounts.
ping 8.8.8.8	Ping Google public DNS to test Internet connectivity.
dig @8.8.8.8 -x 8.8.8.8	Verify that Google DNS at 8.8.8.8 reverse-resolves itself.
dig @192.168.100.2 -x 8.8.8.8	Test dnscache: Reverse-resolve Google's public DNS server at 8.8.8.8, using your dnscache at 192.168.100.2.
dig @192.168.100.2 google.com	Test dnscache: Resolve google.com using your dnscache.
dig @127.0.0.1 wincli.domain.cxm	Test tinydns: Query for wincli.domain.cxm on your authoritative DNS server at 127.0.0.1.
dig @127.0.0.1 -x 192.168.100.5	Test tinydns: Reverse-query for 192.168.100.5 on your authoritative DNS server at 127.0.0.1. NOTE: If the forward authoritative query works but not the reverse, or vice versa, it's probably a misconfigured tinydns (/service/tinydns/root/data) rather than a non-functional one.
dig @192.168.100.2 wincli.domain.cxm	Test dnscache to tinydns link
ps ax | grep svscan	On standard installations this command should produce a line of output containing "svscan /service". If the directory after the word svscan is anything other than /service, then you have a non-standard installation, and you need to carefully evaluate everything.
ls -ldF /service/*	Verify tinydns and dnscache are symlinked to /service.

dnscache Mental Model

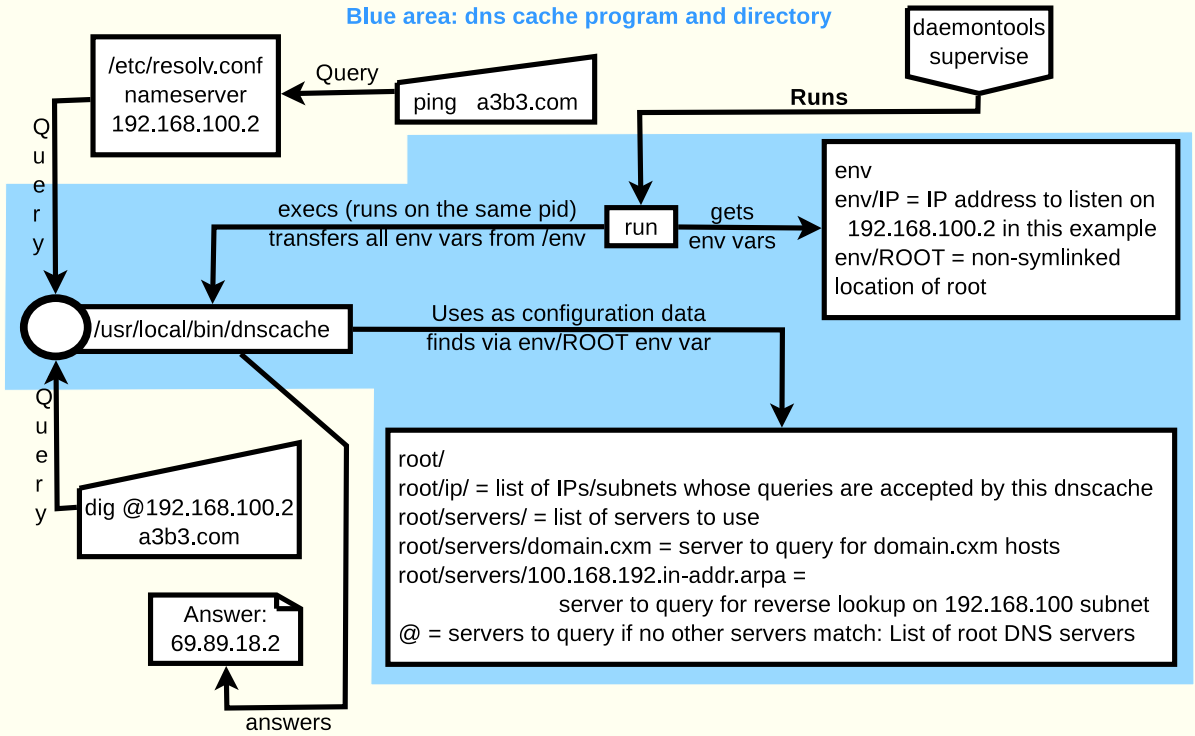
Directory view of dnscache

```
dnscache                    <-- The service
|-- run*                    <-- Script to run (runs /usr/local/bin/dnscache)
|-- seed                    <-- Seed for random numbers
|-- env/                    <-- Dir containing environment vars
|   |-- CACHESIZE           <-- Env vars, fname=key, contents=value
|   |-- DATALIMIT
|   |-- IP
|   |-- IPSEND
|   `-- ROOT                <-- Dir for special config
|-- root/
|   |-- ip/                 <-- Dir defining what IP addresses can access this dnscache
|   |   |-- 127.0.0.1
|   |   |-- 192.168.100
|   |   `-- 192.168.100.2
|   `-- servers/            <-- Dir allocating servers to domains and subnets
|       |-- @
|       |-- 100.168.192.in-addr.arpa
|       `-- domain.cxm
|-- supervise/              <-- Dir used by supervise module of daemontools to control this service
|   |-- control|
|   |-- lock
|   |-- ok|
|   `-- status
`-- log/                    <-- Dir dedicated to logging
```

Functional Layout of dnscache

Not including logging

Blue area: dns cache program and directory



Diagnostic Tests:
dnscache Alone

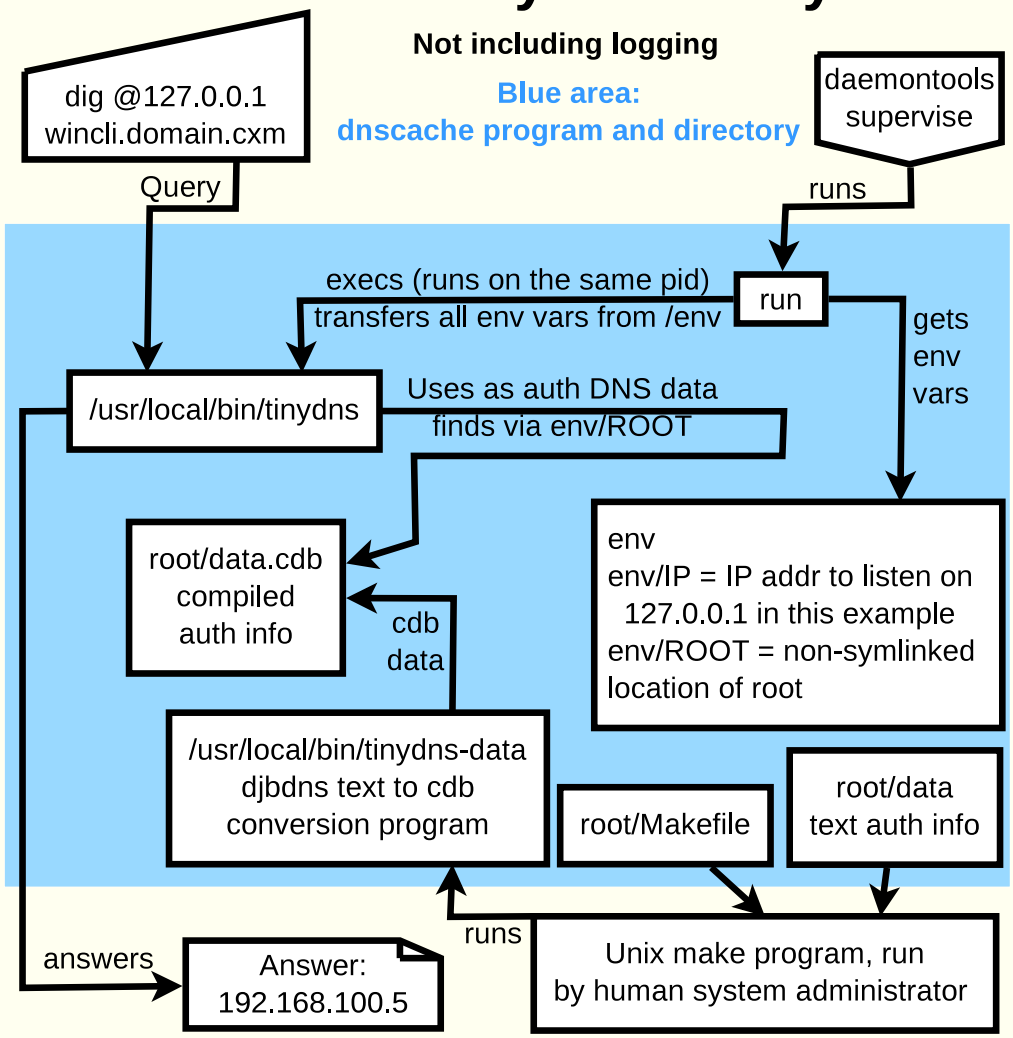
SITUATION	TEST	MEANING
If dnscache didn't work	ps ax | grep dnscache Look for "supervise dnscache" and "/usr/local/bin/dnscache"	If no /usr/local/bin/dnscache, your dnscache software isn't running. If also no "supervise dnscache", probable daemontools problem.
"supervise dnscache" there but not "/usr/local/bin/dnscache"	svc -u /var/djb/service/dnscache sleep 5 ps ax | grep dnscache dig @192.168.100.2 google.com	Turn on dnscache. It might have just been turned off. If it now works, probably all is well — retest.
Preceding test failed to fix problem, or if no "supervise dnscache" process	cd /var/djb/service/dnscache ./run ## Check for error messages ps ax | grep /usr/local/bin/dnscache dig @192.168.100.2 google.com	Directly run dnscache, no daemontools, no logging.
dnscache still doesn't work	ps ax | grep dnscache	Check for multiple instances of "supervise dnscache" or "/usr/local/bin/dnscache", fix if found, repeat previous step.
If you got dnscache working		If you had to do the ./run command, dnscache works OK, probable daemontools problem.

tinydns Mental Model

Directory view of tinydns

```
tinydns          <-- The service
|-- run*         <-- Script to run (runs /usr/local/bin/tinydns)
|-- env/         <-- Dir containing environment vars
|   |-- IP
|   `-- ROOT
|-- root/        <-- Dir for special config
|   |-- add-alias*
|   |-- add-childns*   <-- Programs for adding
|   |-- add-host*          authoritative DNS records to "data"
|   |-- add-mx*
|   |-- add-ns*
|   |-- data       <-- Text version of authoritative DNS records
|   |-- data.cdb   <-- Compiled version of auth DNS records
|   `-- Makefile   <-- So "make" converts from data to data.cdb
|-- supervise/   <-- Dir used by supervise module of
|   |-- control|     daemontools to control this service
|   |-- lock
|   |-- ok|
|   `-- status
`-- log/         <-- Dir dedicated to logging
```

Functional Layout of tinydns



Diagnostic Tests:
tinydns Alone

SITUATION	TEST	MEANING
If tinydns didn't work	ps ax | grep tinydns Look for "supervise tinydns" and "/usr/local/bin/tinydns"	If no /usr/local/bin/tinydns, your tinydns software isn't running. If also no "supervise tinydns", probable daemontools problem.
"supervise tinydns" there but not "/usr/local/bin/tinydns"	svc -u /var/djb/service/tinydns sleep 5 ps ax | grep tinydns dig @127.0.0.1 wincli.domain.cxm	Turn on tinydns. It might have just been turned off. If it now works, probably all is well — retest. NOTE: If dig gives question section but no answer, check root/data info.
Preceding test failed to fix problem, or if no "supervise tinydns" process	cd /var/djb/service/tinydns ./run ## Check for error messages ps ax | grep /usr/local/bin/tinydns dig @127.0.0.1 wincli.domain.cxm	Directly run tinydns, no daemontools, no logging. NOTE: If dig gives question section but no answer, check root/data info.
tinydns still doesn't work	ps ax | grep tinydns	Check for multiple instances of "supervise tinydns" or "/usr/local/bin/tinydns", fix if found, repeat previous step.
If you got tinydns working		If you had to do the ./run command, tinydns works OK, probable daemontools problem.

Diagnostic Tests:
tinydns root/data File Troubleshooting

- Make and test trivial root/data
  - cd /service/tinydns/root
    mv data data.org
    touch data
    ./add-ns domain.cxm 127.0.0.1
    ./add-ns 100.168.192.in-addr.arpa 127.0.0.1
    ./add-host mydesk.domain.cxm 192.168.100.2
    ./add-host wincli.domain.cxm 192.168.100.5
    make
    svc -t /service/tinydns; sleep 5
    dig @127.0.0.1 wincli.domain.cxm
    dig @127.0.0.1 -x 192.168.100.5
- If preceding worked, troubleshoot/test data.org
  - cd /service/tinydns/root
    mv data data.simple
    cp data.org data
    ## Troubleshoot as necessary
    make
    svc -t /service/tinydns; sleep 5
    dig @127.0.0.1 wincli.domain.cxm
    dig @127.0.0.1 -x 192.168.100.5

Diagnostic Tests:
Special Cases

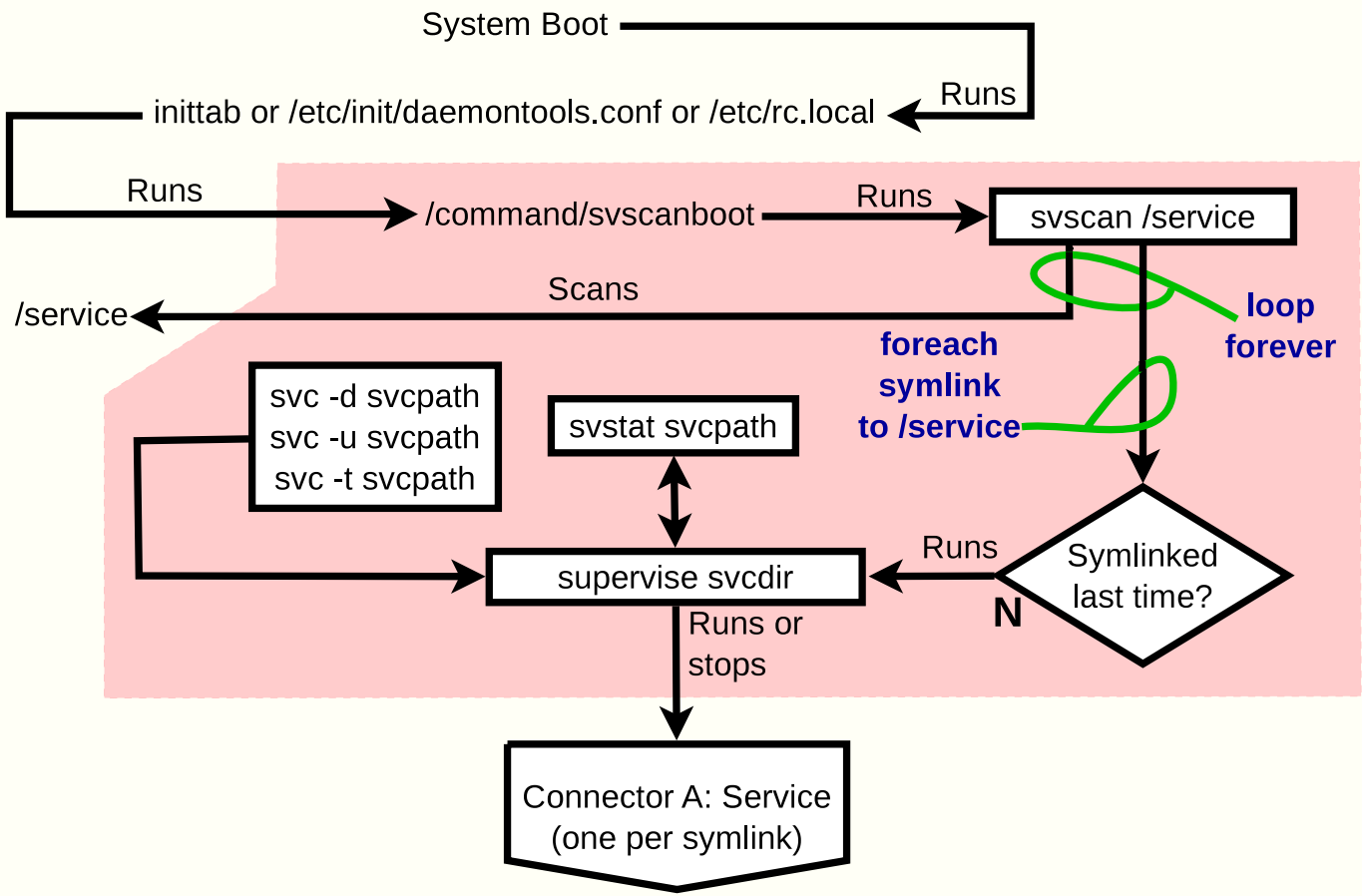
- dnscache answers queries from 192.168.100.2 but not 192.168.100.5
  - Add empty file /service/dnscache/root/ip/192.168.100
  - svc -t /service/dnscache

- dig @127.0.0.1 wincli.domain.cxm succeeds but dig @192.168.100.2 wincli.domain.cxm fails
  - Create following files in /service/dnscache/root/servers:
    - domain.cxm containing 127.0.0.1
    - 100.168.192.in-addr.arpa containing 127.0.0.1
  - svc -t /service/dnscache
  - This tells dnscache to query tinydns at 127.0.0.1 for domain.cxm

daemontools Mental Model

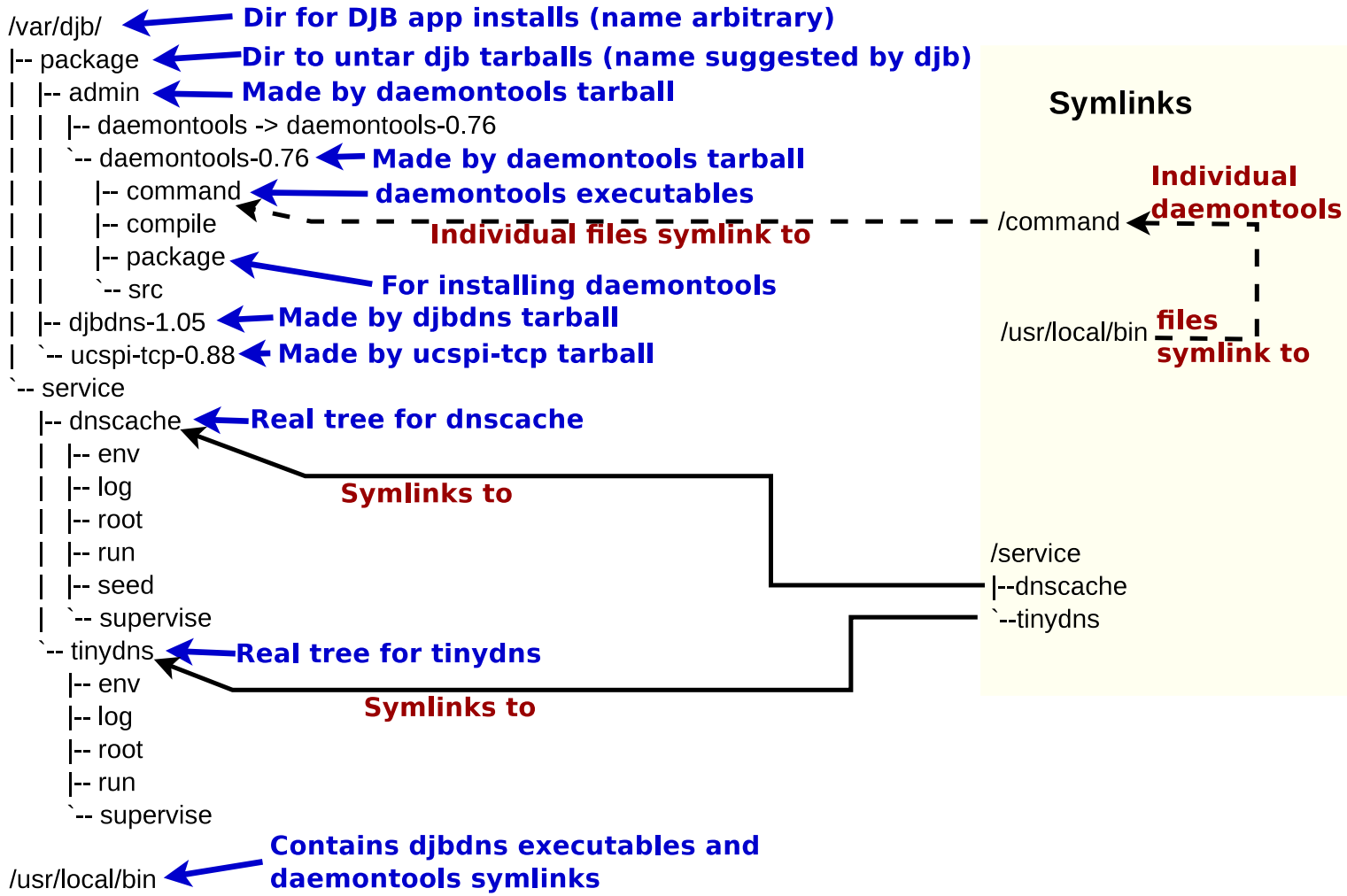
Components of daemontools

Pastel red area is daemontools system

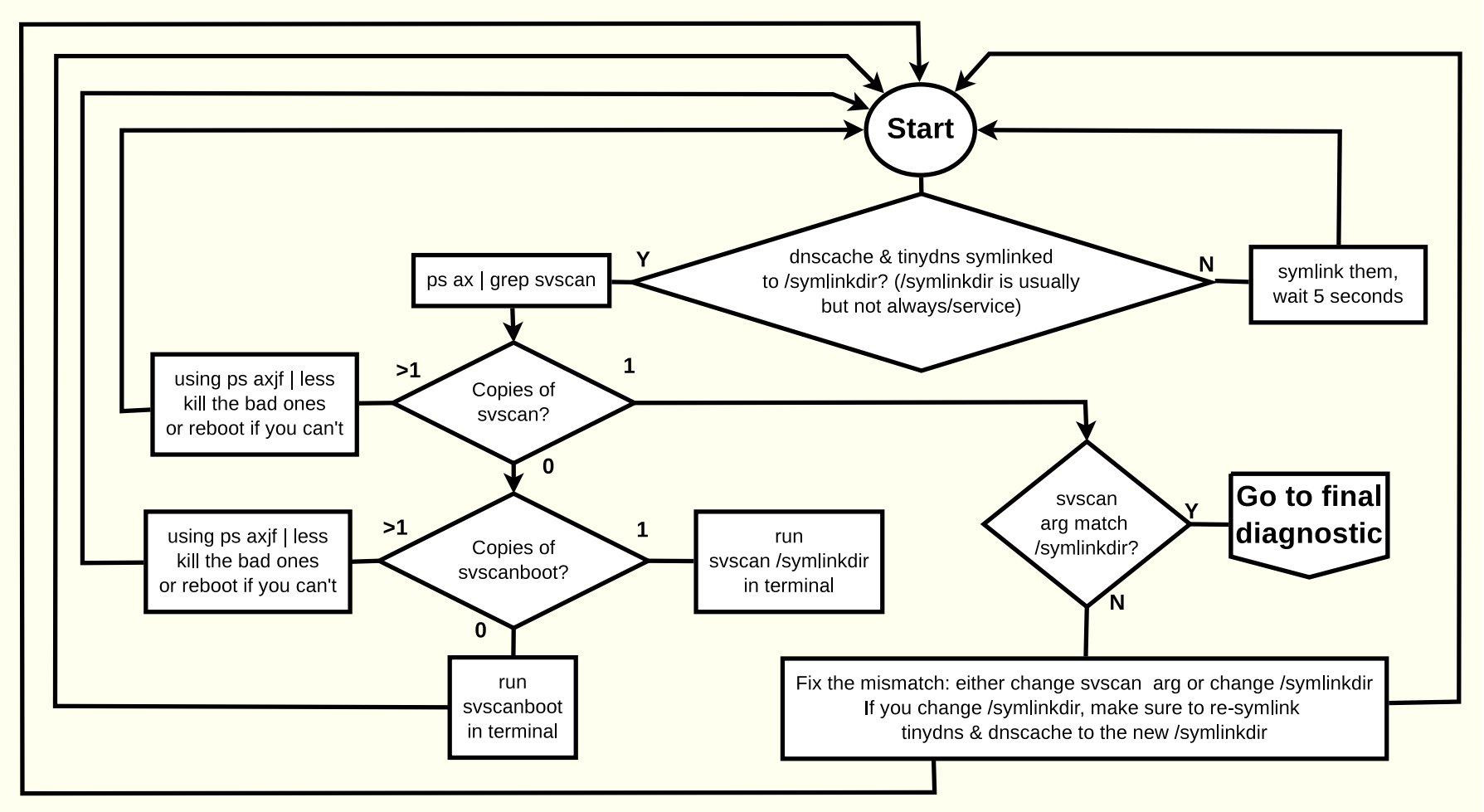


daemontools Mental Model
Directory Structure

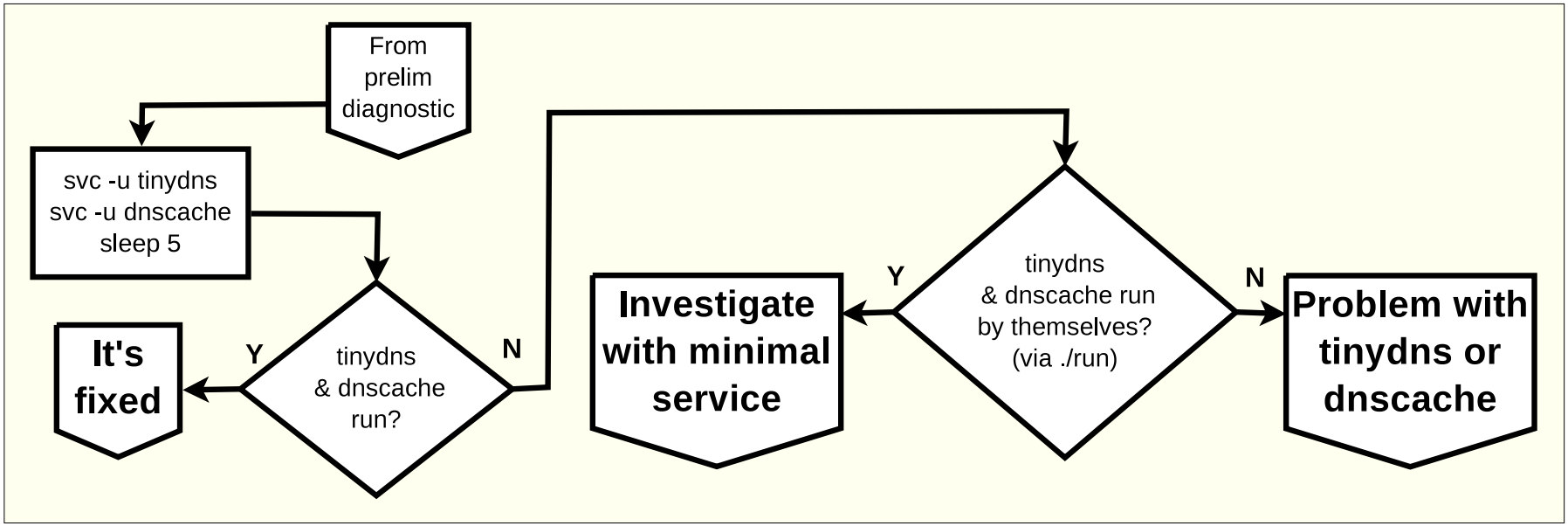
djb Directories and Symlinks



Diagnostic Tests:
Daemontools Preliminary Diagnostic Procedures



Diagnostic Tests:
Daemontools Final Diagnostic Procedures



Summary

- Success at any task depends on how you ***operate*** your brain
- Organize your system knowledge as block diagrams
- Don't try to fix it, just try to narrow it down
- Always ask, "How can I narrow it down one more time?"
- Narrow down with tools such as
  - `ps ax | grep whatever`
  - `ps axjf | less`
  - `dig @serverIP`
  - `./run`
  - `svc -d path` or `svc -u path` or `svc -t path`
- Remember the Quadruple Tradeoff and the 10 Step Universal Troubleshooting Process

Now You Can:

- Name all elements of the Troubleshooter's Mindset
- Name all major components of djbdns
- Narrow the root cause to a major component in a few minutes
- Run any major djbdns component by itself
- Understand the subcomponents of each major components
- Isolate to the subcomponent in a few minutes
- Solve the whole problem within an hour 99% of the time

Troubleshooters.Com

- Your source for books on:
  - Troubleshooting
  - Rapid Learning
  - Human Performance
- Your source for Troubleshooting courses and courseware
- Home of the *Universal Troubleshooting Process*

- http://www.troubleshooters.com

See http://www.troubleshooters.com for complete info on troubleshooting, including books and courseware.