Chapter 1. Troubleshooting System Errors

If you have Evergreen installed and are encountering systematic errors, here is the steps to find the cause and solution to most problems. These instructions assume standard locations and file names for Evergreen installations, and may also include commands for specific Linux distributions.

Procedure 1.1. Systematic Evergreen Restart to isolate Errors

1. Stop Apache:

   ```
   /etc/init.d/apache2 stop
   ```

   or

   ```
   apache2ctl stop
   ```

2. Stop OpenSRF:

   ```
   osrf_ctl.sh -l -a stop_all
   ```

   You should get either output resembling this:

   ```
   Stopping OpenSRF C process 12515...

   Stopping OpenSRF C process 12520...

   Stopping OpenSRF C process 12526...

   Stopping OpenSRF Perl process 12471...

   Stopping OpenSRF Router process 12466...
   ```

   Or, if services have already been stopped, output maay look like this:

   ```
   OpenSRF C not running

   OpenSRF Perl not running

   OpenSRF Router not running
   ```

   Occasionally osrf_ctl.sh fails to kill OpenSRF processes, so we should check to make sure that none are still running with the command:

   ```
   ps -aef | grep OpenSRF
   ```

   You should manually kill any OpenSRF processes.

   If you were unable to stop OpenSRF with the above methods, you could also try this command:

   ```
   rm –R /openils/var/run/*.pid
   ```

   This will remove the temporary OpenSRF process files from the run directory which may have been left over from a previous system boot cycle.

3. Restart Ejabberd and Memcached with the following commands:

   ```
   sudo /etc/init.d/ejabberd restart

   sudo /etc/init.d/memcached restart
   ```

4. Start the OpenSRF router and check for errors

   ```
   /openils/bin/osrf_ctl.sh -l -a start_router
   ```

   If the router started correctly, output will be:

   ```
   Starting OpenSRF Router
   ```

   If router does not start correcltly, you should check the router error log files for error information.

   Evergreen 1.6 uses two routers, a public one and a private one, with two different logfiles:

   ```
   /openils/var/log/private.router.log

   /openils/var/log/public.router.log
   ```

   A quick way to find error information in the logs is with the grep command.

   ```
   grep ERR /openils/var/log/*router.log
   ```

   As a final sanity check, look for router processes using the process status command:

   ```
   ps -aef | grep Router
   ```

5. Start the OpenSRF perl services and check for errors

   ```
   /openils/bin/osrf_ctl.sh -l -a start_perl
   ```

   You should see the following output:

   ```
   Starting OpenSRF Perl

   * starting all services for norcrossfx.norcross.esi

   * starting servivce pid=7484 opensrf.settings

   * starting servivce pid=7493 open-ils.cat

   * starting servivce pid=7495 open-ils.supercat

   * starting servivce pid=7497 open-ils.search

   * starting servivce pid=7499 open-ils.circ

   * starting servivce pid=7501 open-ils.actor

   * starting servivce pid=7502 open-ils.storage

   * starting servivce pid=7509 open-ils.penalty

   * starting servivce pid=7512 open-ils.collections

   * starting servivce pid=7514 open-ils.ingest
   ```

```
* starting servivce pid=7517 open-ils.permacrud
```

```
* starting servivce pid=7522 open-ils.fielder
```

```
* starting servivce pid=7527 open-ils.vandelay
```

```
* starting servivce pid=7516 open-ils.reporter
```

If the perl services do not start correclty or you receive errors, search for errors in the following log files:

```
/openils/var/log/router.log
```

```
/openils/var/log/osrfsys.log
```

At this point you can use the **grep** command to find errors in any of the Evrgreen log files:

```
grep ERR /openils/var/log/*.log
```

As a final sanity check, look for OpenSRF processes:

```
ps -aef | grep -i opensrf
```

6. Start the OpenSRF C services and check for errors:

   ```
   /openils/bin/osrf_ctl.sh -l -a start_c
   ```

   And output should be:

   Starting OpenSRF C (host=localhost)

   If the *c* service does not start, check for errors by grepping the log files for errors:

   ```
   grep ERR /openils/var/log/*.log
   ```

   Check for OpenSRF processes:

   ```
   ps -aef | grep -i opensrf
   ```

7. Smoke test with autogen.sh

   The Autogen tool will take some dynamic information from the database and generate static Javascript files for use by the OPAC and staff client. It is also able to refresh the proximity map between libraries for the purpose of efficiently routing hold requests.

   As user opensrf, you invoke Autogen with the command:

   ```
   /openils/bin/autogen.sh -c /openils/conf/opensrf_core.xml -u
   ```

   If Autogen completes successfully, the output will be:

   ```
   Updating fieldmapper
   ```

   ```
   Updating web_fieldmapper
   ```

   ```
   Updating OrgTree
   ```

   ```
   removing OrgTree from the cache...
   ```

```
Updating OrgTree HTML

Updating locales selection HTML

Updating Search Groups

Refreshing proximity of org units

Successfully updated the organization proximity

Done
```

If Autogen does not complete its task and you recieve errors, use **grep** to find errors in the log files:

```
grep ERR /openils/var/log/*.log
```

8. Connect to Evergreen using the srfsh command-line OpenSRF client

```
/openils/bin/srfsh
```

Note

In order for you to connect using **srfsh**, you will need to have set up the .srfsh.xml configuration file in your home directory as as described in the installation chapter.

You will then see the *srfsh* prompt:

```
srfsh#
```

At the srfsh prompt, enter this command:

```
login admin open-ils
```

You should the request verification:

```
Received Data: "6f63ff5542da1fead4431c6c280efc75"

------------------------------------

Request Completed Successfully

Request Time in seconds: 0.018414

------------------------------------


Received Data: {

"ilsevent":0,

"textcode":"SUCCESS",

"desc":" ",

"pid":7793,

"stacktrace":"oils_auth.c:312",
"payload":{

"authtoken":"28804ebf99508496e2a4d2593aaa930e",

 "authtime":420.000000

}

}


------------------------------------

Request Completed Successfully

 Request Time in seconds: 0.552430

------------------------------------

 Login Session: 28804.  Session timeout: 420.000

srfsh#
```

If you encounter errors or if you are unable to connect, you should consult the `srfsh.log` file. The location of this file is configured in your `.srfsh.xml` configuration file and is `/openils/var/log/srfsh.log` by default.

Pressing **Ctrl**+**D** or entering “exit” will terminate srfsh.

9. Start Apache and check for errors:

```
/etc/init.d/apache2 start
```

or

```
apache2ctl start
```

You should see output:

```
* Starting web server apache2

...done.
```

the Apache OpenSRF modules write to the `/openils/var/log/gateway.log`

However, you should check all of the log files for errors:

```
grep ERR /openils/var/log/*.log
```

Another place to check for erros is the Apache error logs generally located in in the `/var/log/Apapche2` directory

If you encounter errors with Apache, a common source of potential problems are the Evergreen site configuration files `/etc/apache2/eg_vhost.conf` and `/etc/apache2/sites-available/eg.conf`

10. Try to authenticate with the staff client

11. Try to authenticate with the OPAC