

MotM - Supervisor

Ed Silber

Github: <http://bit.ly/1gsVJtr>

IE Python User Group

March 18, 2014

Introduction

- Supervisord is a process control system written in Python that can be used to manage scripts, applications and/or frameworks as Unix daemons
 - Child programs you write that run in supervisord are referred to as sub-processes
 - It requires Python 2.4 – 2.7, but not Python 3
 - Supervisord has been tested with Linux, Solaris, FreeBSD and Mac OS X, but will not run under any version of Windows
-

Supervisord Components

- supervisord

- The server piece that is responsible for starting and managing sub-processes configured in it's configuration file, typically located in `/etc/supervisord.conf`

- supervisorctl

- A client command-line tool used to connect and control supervisord and its sub-processes
-

Supervisord Components

- Web server
 - A simple web UI with similar functionality to supervisorctl; it must be configured in supervisord.conf
- XML-RPC Interface
 - Provided in the web server, this API can be used to monitor and control supervisor and its sub-processes programmatically

Installing

- The simplest way to install is via pip:
pip install supervisor
- Next, generate a default configuration file (as root):
echo_supervisord_conf > /etc/supervisord.conf
- Scripts are available to automatically run supervisord on startup

Sub-processes

- Don't daemonize programs running in supervisord—they should run in the foreground
 - Sub-processes should not detach themselves from the terminal where they are started
 - Environmental variables can be set in supervisord.conf which makes them accessible to sub-processes
-

Logging

- Supervisord will log messages on it's own health and sub-process state changes
 - Each program managed by supervisord can have a stderr and stdout log file, or both combined in one file
 - Both supervisord and programs can be configured with log file rotation, maximum sizes and the number of backups to keep
 - Logging is configured in `supervisord.conf`
-

Additional Features

- Run a sub-process from start to finish continuously while supervisord is running
 - Custom event listeners can be written to respond to changes in supervisord states
 - The XML-RPC API can be extended for deeper programmatic control of supervisord
-

Demo: Log Monitoring

- Installing and configuring supervisord, including using an init.d script
- Monitoring a log file (/var/log/messages) and replicating 1–10 lines at a time into another file
- Keeping track of where we are, i.e., state management
- Installing the demo:

1. `pip install supervisor`

2. `cp supervisord_demo/config/supervisord.conf /etc/supervisord.conf`

3. `cp supervisord_demo/config/supervisord /etc/init.d/supervisord`

Resources

- All information in this presentation provided by the excellent documentation at:
<http://supervisord.org>
- Startup scripts are discussed and referenced from:
<http://supervisord.org/running.html#running-supervisord-automatically-on-startup>