

# xyz-supervisor配置教程

文档更新日期：2022-02-12

## 一、基本介绍

Supervisor是一款采用Python编写、运行在Linux/Unix系统上的进程管理工具。该工具提供了一系列管理进程的功能。例如，启动进程、停止进程、查看进程日志、重启进程、分组管理进程、进程开机自启动、RPC接口等。但是由于Supervisor不能控制进程的启动顺序，也不能知道进程初始化完成的时间，因此需要Central Hub加以辅佐。并且，**Central hub作为一个特殊的节点运行在supervisor中。**

## 二、如何安装

公司的系统环境中默认已经安装了supervisor，因此实际安装时只需要再下载一个xyz-supervisor安装包。

### 1. 使用xyz-apt安装

1. 更新索引源：`sudo xyz_apt update`
2. 安装软件包：`sudo xyz_apt install xyz-supervisor`
3. 登录 [ <http://127.0.0.1:9001> ] 或者在终端输入 `central_hub_gui`，如页面正常打开，则说明安装成功。

### 2. 使用xyz-csc安装

1. 打开csc
2. 进入【软件安装/卸载】模块
3. 搜索xyz-supervisor，点击安装，安装最新版本的xyz-supervisor

## 三、如何配置

### 1. 配置节点

打开 `/home/xyz/xyz_app/central_hub/node/nodes.ini` 文件，在文件中进行节点配置。一个 `[program:xxx]` 块就是一个节点。或者，你也可以在终端输入 `central_hub_gui` 打开节点配置界面，进行节点配置。

### 2. 配置central hub

central hub的相关参数基本不需要修改，唯一需要修改的就是节点启动顺序。

### 3. 配置节点启动顺序

打开 `/home/xyz/xyz_app/central_hub/nodes.ini` 文件，找到 `boot_sequence` 字段。添加节点顺序。（每次修改节点后要记得确认节点顺序是否发生变动）

## 四、如何启用

### 1. 重新加载节点配置

当我们更新并保存了一些节点配置后，节点管理服务中的节点配置不会主动更新，这时就需要我们重新加载一次节点配置。这种加载方式只会关闭有改动的节点。其他未改动的节点的启停不会受影响。

```
supervisorctl update
```

### 2. 重启supervisor服务

（注意：该操作会重启supervisor服务，之前已打开的节点将全部被关闭。）执行指令如下：

```
sudo systemctl restart xyz_autostart.service
```

或者

```
supervisorctl reload
```

## 五、节点范例

### 配置整站需要的软件节点（以拆码垛项目为例）

1. 打开目录：`/home/xyz/xyz_app/central_hub`。所有和central hub相关的配置文件都在该目录下。
2. 选择相应项目类型的节点配置文件。预设节点配置的文件路径为 `/home/xyz/xyz_app/central_hub/node/dpt/nodes.ini`。
3. 将预设节点配置复制到 `/home/xyz/xyz_app/central_hub/node` 目录下。
4. 配置相关节点并保存。
5. 更新节点配置。可使用命令 `supervisorctl update`

注：

- 项目类型用简写的字母表示。如 `rebin`指播种站项目，`pp`指拣选站项目，`dpt`指拆码垛项目，`ind`指工业项目。
- 如果需要重启节点服务则可以使用 `supervisorctl reload` 或者 `sudo systemctl restart xyz_autostart.service`

## 六、节点参数说明

节点配置过程中涉及很多参数设置。我们以一个节点示例来举例：

```
[program:4-robot_node]
environment =
command = /opt/ros/noetic/bin/rosrun xyz_robot xyz_robot_node
process_name = %(program_name)s
priority = 40
autostart = false
autorestart = false
startsecs = 2
startretries = 1
user = xyz
stdout_logfile = /home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
stdout_logfile_maxbytes = 10MB
stdout_logfile_backups = 2
stdout_capture_maxbytes = 10MB
stdout_events_enabled = false
stderr_logfile = /home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
stderr_logfile_maxbytes = 10MB
stderr_logfile_backups = 2
stderr_capture_maxbytes = 10MB
stderr_events_enabled = false
_nodeid = robot_node
_nodename = Robot Driver 2.0
_monitor = true
_popup = false
_dependency =
_description =
```

先介绍几个比较重要的节点参数：

- **environment**：设置节点的单独环境变量
- **command**：启动这个节点的指令。该指令最好使用绝对路径，因为supervisor在运行时其环境变量不是xyz用户下的环境变量。而是单独在supervisord.conf配置文件中设置了环境变量
- **autostart**: 是否开机启动（随supervisor服务启动后启动）
- **autorestart**: 是否自动重启
- **startsecs**: 该节点启动成功的判定时间，比如设置2，那么就是指该节点需要启动运行2秒不退出后，会认为这个节点启动成功
- **\_nodeid**：节点编号，供central hub使用。节点编号作为唯一标示在central hub中用于识别节点。目前robot\_node节点的节点ID固定为 robot\_node，其他节点的节点ID非固定。
- **\_nodename**: 节点名称，供central hub使用。该名称主要是为了方便用户看，也可设置为中文
- **\_monitor**: 节点启动成功的信号监听。数值可设置true或false，如果为true，就开启对该节点的监听。即当启动该节点后，需要该节点向10001端口的socket服务推送一个启动成功的信号。**需要注意推送的启动成功信息中node\_id这个字段的字段名是 node\_name**，推送信息如下：

```
{
  "code": 0,
  "msg": ""
  "node_name": "node_id",
  "timestamp": 1644569370365
}
```

以下参数一般不需要额外改动，使用默认即可。另外，`stderr_logfile_maxbytes`等参数此处不作额外说明，主要用于设置日志的大小和备份数量等。

- **process\_name**：在supervisor中显示的进程名称，`%(program_name)s`是指使用`[program:4-robot_node]`中的`4-robot_node`
- **priority**：启动优先级，数值越大优先级越低。即使设置了也没有太大作用
- **user**：进程用户名，默认使用xyz
- **\_dependency**：节点依赖关系，经过各软件迭代，节点依赖的功能已经很少会被使用，目前没有使用需求。
- **\_description**：节点作用描述。

## 七、常见问题

### 1. 什么是节点监听？

节点启动有两种模式，一种是普通模式，另一种是节点监听模式。它们通过不同的标准来判断节点是否启动成功。普通模式下，根据设置的启动时间来判断节点是否成功启动。即，如果程序成功运行N秒并且中途没有退出，那么认为该节点启动成功。但是这种模式不能很好的运用在socket连接通信中。比如，socket服务启动后，虽然成功启动了，但是这个时候并没有与客户端建立连接，因此需要一个节点中的socket服务在与客户端建立连接后回调信息，告知central\_hub程序服务端已经与客户端建立连接。因此，就有了节点监听模式。

比较典型的就是robot\_node节点，为了确保robot\_node可以真正与机器人建立连接，因此我们需要启用节点的监听功能。robot\_node节点启动后，会尝试等待与机器人建立通信，这个过程节点一直处于RUNNING状态。当与机器人通信建立后，便会发送一个请求到central\_hub的节点服务中，告知节点服务，robot\_node节点启动成功。但是需要注意的是，robot\_node启动回告功能也需要设置。配置文件的路径在`/home/xyz/xyz_app/robot_config/robot_config.yml`。

由此延伸导致可能造成的误解，如：

- 为什么节点界面中robot\_node的状态是RUNNING，但是在studio、HMI中的机器人节点依然没有启动？

### 2. 为什么有的节点配置参数前面有下划线，比如\_nodeid？

目前带下划线的配置参数有\_nodeid, \_nodename, \_monitor, \_popup, \_dependency, \_description 6个。带上了下划线是因为这些参数本身并不是supervisor的参数，而是在central\_hub程序中需要用到。加上下划线方便区分这些自定义添加的参数字段。supervisor控制节点启停，提供基本的节点管理功能，central\_hub程序相当于在这个基础上又进行的一次功能拓展。

### 3. 为什么新建一个节点并且到central hub添加到启动顺序中后，HMI中依然没有显示？

可能原因如下：

- 没有更新节点配置
- 更新了节点配置，但是并没有重启central hub，导致central hub的节点列表没有更新

- central hub未启动
- HMI因为接口、网络等问题导致数据请求失败

#### 4. 【重启节点服务】后central\_hub\_gui界面里节点列表空了，9001端口的网址也打不开了？

可能是因为配置的节点参数出错了，导致将supervisor服务关闭后，再次加载节点参数时出错，致使节点不能正常开启。