**Daemon Process Manager – Architecture and Design doc.**

1. **What is a Daemon Process Manager?**

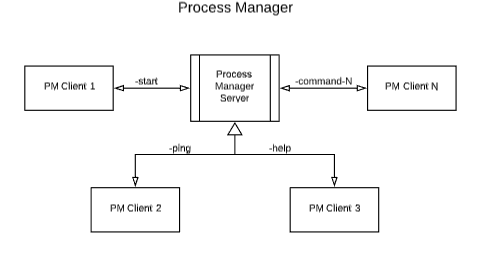
* **Daemon Process Manager** is a daemon/background process to manage other processes.

1. **Features**,

* **Configuration driven:** All the information are read from a configuration file, mostly XML file.
* **CLI architecture:** Command line interface architecture.
  + E.g - where **pm** is the process name,
    - **pm start –id0** : start process with the id “id0” in the configuration file.
    - **pm ping** : check the status of all the processes defined in the configuration file.
* **Process Monitoring and Action Handlers**: Enabled with process monitoring and action handler capabilities.

1. **Architecture,**

* Client-server architecture via TCP/IP.



* This architecture allows to manage the Process Manager from remote system and also possibly even through cross platform systems.
  + Manage local system processes (linux-linux).
  + Manage remote system processes (linux-linux , windows-linux).
* Since client and server communicate with the common interface, they are decoupled and could be interconnected as N-Client – N-Server.

1. **Design,**

* Process Manager requires the following different components to function as desired,
  + **CommandManager** : Where all the commands are defined and the CLI parsing is done.
  + **PDU**: Common low-level cross-platform compatible **Predefined Data Unit** package which carries the request/response data between the client and servers.
  + **SocketManager**: Common interface responsible for sending and receiving the requests.
  + **RequestsHandler:** Common interface responsible to handle command request and response.
  + **ConfigurationManager:** Server side configuration manager which handles the configuration file read/write operations.
  + **ProcessManager:** Process data manager and responsible to primary operations like process start, stop ,restart, etc.
  + **Logger:** Log manager.
  + **HelpManager:** Client side self-explanative CLI helper.

1. **Typical request lifecycle,**

**Prerequisites:**

* Server is running configured.
* Clients and Servers reachable to each other.
* Are running the same binary versions (Common Interface)

**Prerequisites:**

* User runs the client with the respected command line arguments.
* Client CLI validates the request command, if failed to parse the command displays the proper error messages.
* Upon successful command validation it checks for the server connectivity and interface version running, either in case server unreachable/version mismatch it displays the proper error message and exit.
* Upon successful command request to the server, Server executes the command routine defined and returns the result to the client.
* Client display the response and exit.

**Supported commands,**

In the following example **pmc** is the **Process Manager Client** executable name.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **Command** | **Variants** | **Usage** |
| 1 | **Start**: Start the process or set of processes. | 1. **ALL**: Start all the processes listed in configuration. 2. **BY\_ID:** Start a specific process listed in configuration by ID. | 1. **./pmc start** 2. **./pmc start id-0** |
| **2** | **Stop**: Restart the process or set of processes. | 1. **ALL**: Stop all the processes listed in configuration. 2. **BY\_ID:** Stop a specific process listed in configuration by ID. | 1. **./pmc stop** 2. **./pmc stop id-0** |
| 3 | **Restart**: Restart the process or set of processes. | 1. **ALL**: Restart all the processes listed in configuration. 2. **BY\_ID:** Restart a specific process listed in configuration by ID. | 1. **./pmc restart** 2. **./pmc restart id-0** |
| 4 | **Ping**: Ping the process or set of processes. | 1. **ALL**: Ping all the processes listed in configuration. 2. **BY\_ID:** Ping a specific process listed in configuration by ID. | 1. **./pmc ping** 2. **./pmc ping id-0** |
| 5 | **Config**: Display or perform an operation on the configuration data. | 1. **DISPLAY ALL**: Display all the processes listed in configuration. 2. **DISPLAY BY\_ID:** Display a specific process listed in configuration by ID. 3. **RELOAD:** Reload data from the configuration file. | 1. **./pmc config -display** 2. **./pmc config -display -id0** 3. **./pmc config -reload** |
| 6 | **Setting**: Display or perform an operation on the server setting data. | 1. **DISPLAY ALL**: Display all the server settings. 2. **DISPLAY BY\_ID:** Display a specific server setting by ID. 3. **RELOAD:** Reload server setting from the fi. | 1. **./pmc setting -display** 2. **./pmc setting display -id0** 3. **./pmc setting -reload** |
| 7 | **Help**: Help guide on the available commands. | Variants available for each of the supported commands.  When used without command, displays help for all commands. | 1. **./pmc help** 2. **./pmc help -start** 3. **./pmc help -stop** 4. **./pmc help -ping** 5. **./pmc help -help** |