

```
<?xml version="1.0" encoding="US-ASCII"?>
```

```
<!-- NodeJS Pool Controller Plugin Copyright (C) 2020 Robert Strouse This program is free software;
you can redistribute it and/or modify it under the terms of the GNU General Public License as published
by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License for more details. You should have received a copy of
the GNU General Public License along with this program; if not, write to the Free Software Foundation,
Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. -->
```

```
<implementation>
```

```
<functions> function pcpInitialize(lul_device) -- Put the /etc/cmh-ludl and /etc/cmh-lu directories
into the Lua include path. if (package.path:find("/etc/cmh-ludl/?.lua;/etc/cmh-lu/?.lua", 1, true)
== nil) then package.path = package.path .. "/etc/cmh-ludl/?.lua;/etc/cmh-lu/?.lua" end --
Load the Lua implementation. package.loaded.L_PoolController = nil luup.log("Loading Pool
Controller") pcp_plugin = require("L_PoolController") -- Create our global callbacks
pcpCallbackHandler = pcp_plugin.pcpCallbackHandler -- Call initialization code. return
pcp_plugin.pcpInitialize(lul_device) end function locals() local vars = {} local ndx = 1 while
true do local ln, lv = debug.getlocal(2, ndx) if ln ~= nil then vars[ln] = lv else break end ndx =
ndx + 1 end return vars end function upvals() local vars = {} local ndx = 1 local func =
debug.getinfo(2, "f").func while true do local ln, lv = debug.getupvalue(func, ndx) if ln ~= nil
then vars[ln] = lv else break end ndx = ndx + 1 end return vars end </functions>
```

```
<startup>pcpInitialize</startup>
```

```
- <actionList>
```

```
- <action>
```

```
<serviceId>urn:rstrouse-com:serviceId:PoolController1</serviceId>
```

```
<name>SetEventData</name>
```

```
<job> luup.log("Setting Data target for: " .. lul_device .. " to " .. lul_settings.targetData)
pcp_plugin.jobSetData(lul_device, lul_settings, lul_job) return 4, nil </job>
```

```
</action>
```

```
- <action>
```

```
<serviceId>urn:upnp-org:serviceId:Dimming1</serviceId>
```

```
<name>SetLoadLevelTarget</name>
```

```
<job> luup.log("Setting Dimming target for: " .. luup.devices[lul_device].id .. " to " ..
lul_settings.newLoadlevelTarget) pcp_plugin.jobSetDimmingTarget(lul_device,
lul_settings, lul_job) return 4, nil </job>
```

```
</action>
```

```
- <action>
```

```
<serviceId>urn:upnp-org:serviceId:SwitchPower1</serviceId>
```

```
<name>SetTarget</name>
```

```
<job> luup.log("Setting Binary target for: " .. luup.devices[lul_device].id .. " to " ..
lul_settings.newTargetValue) pcp_plugin.jobSetBinaryTarget(lul_device,
lul_settings, lul_job) return 4, nil </job>
```

```
</action>
```

```
- <action>
```

```
<serviceId>urn:upnp-org:serviceId:HVAC_UserOperatingModel</serviceId>
```

```
<name>SetModeTarget</name>
```

```
<job> luup.log("Setting Body Heat target for: " .. luup.devices[lul_device].id .. " to " ..
lul_settings.NewModeTarget) pcp_plugin.jobSetBodyHeatModeTarget(lul_device,
lul_settings, lul_job) return 4, nil </job>
```

```

    </action>
- <action>
  <serviceId>urn:upnp-org:serviceId:TemperatureSetpoint1_Heat</serviceId>
  <name>SetCurrentSetpoint</name>
  <job> luup.log("Setting Body Setpoint target for: " .. luup.devices[lul_device].id .. " to
    " .. lul_settings.NewCurrentSetpoint) pcp_plugin.jobSetBodySetpointTarget
    (lul_device, lul_settings, lul_job) return 4, nil </job>
</action>
- <action>
  <serviceId>urn:rstrouse-com:serviceId:PoolChlorinator1</serviceId>
  <name>SetChlorSetpoint</name>
  <job> pcp_plugin.jobSetChlorSetpointTarget(lul_device, lul_settings, lul_job) return 4,
    nil </job>
</action>
- <action>
  <serviceId>urn:rstrouse-com:serviceId:PoolChlorinator1</serviceId>
  <name>SuperChlorinate</name>
  <job> pcp_plugin.jobSuperChlorinate(lul_device, lul_settings, lul_job) return 4, nil
    </job>
</action>
</actionList>
</implementation>

```