

CyberLink for MacOSX

Programming Guide

Table of Contents

1	Introduction.....	1
2	Setup.....	2
2.1	System Requirement.....	2
2.2	Installer.....	2
3	Control Point.....	3
3.1	Class Overview.....	3
3.2	Initiating.....	4
3.3	Root Devices.....	4
3.4	Control.....	4
4	License.....	6

1 Introduction

UPnP™¹ architecture is based on open networking to enable discovery and control of networked devices and services, such as media servers and players at home.

UPnP™ architecture is based on many standard protocols, such as GENA, SSDP, SOAP, HTTPU and HTTP. Therefore you have to understand and implement these protocols to create your devices of UPnP™.

CyberLink for MacOSX is a development package for UPnP™ developers. The CyberLink controls these protocols automatically, and supports to create your control points quickly.

Please see the following site and documents to know about UPnP™ in more detail.

Document	URL
UPnP™ Forum	http://www.upnp.org/
Universal Plug and Play Device Architecture	http://www.upnp.org/download/UPnPDA10_20000613.htm
Universal Plug and Play Vendor's Implementation Guide	http://www.upnp.org/download/UPnP_Vendor_Implementation_Guide_Jan2001.htm

¹ UPnP™ is a certification mark of the UPnP™ Implementers Corporation.

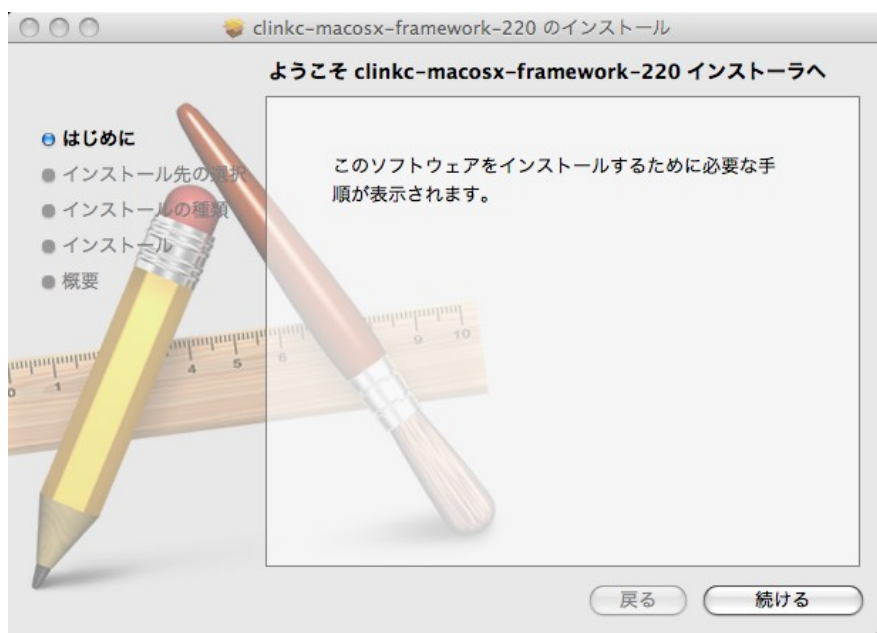
2 Setup

2.1 System Requirement

Currently, the framework is based on Objective-C 2.0. Thus, the current framework requires MacOSX v10.5, Leopard, later. In the future, I will support other lower MacOSX and iPhone v2.0.

2.2 Installer

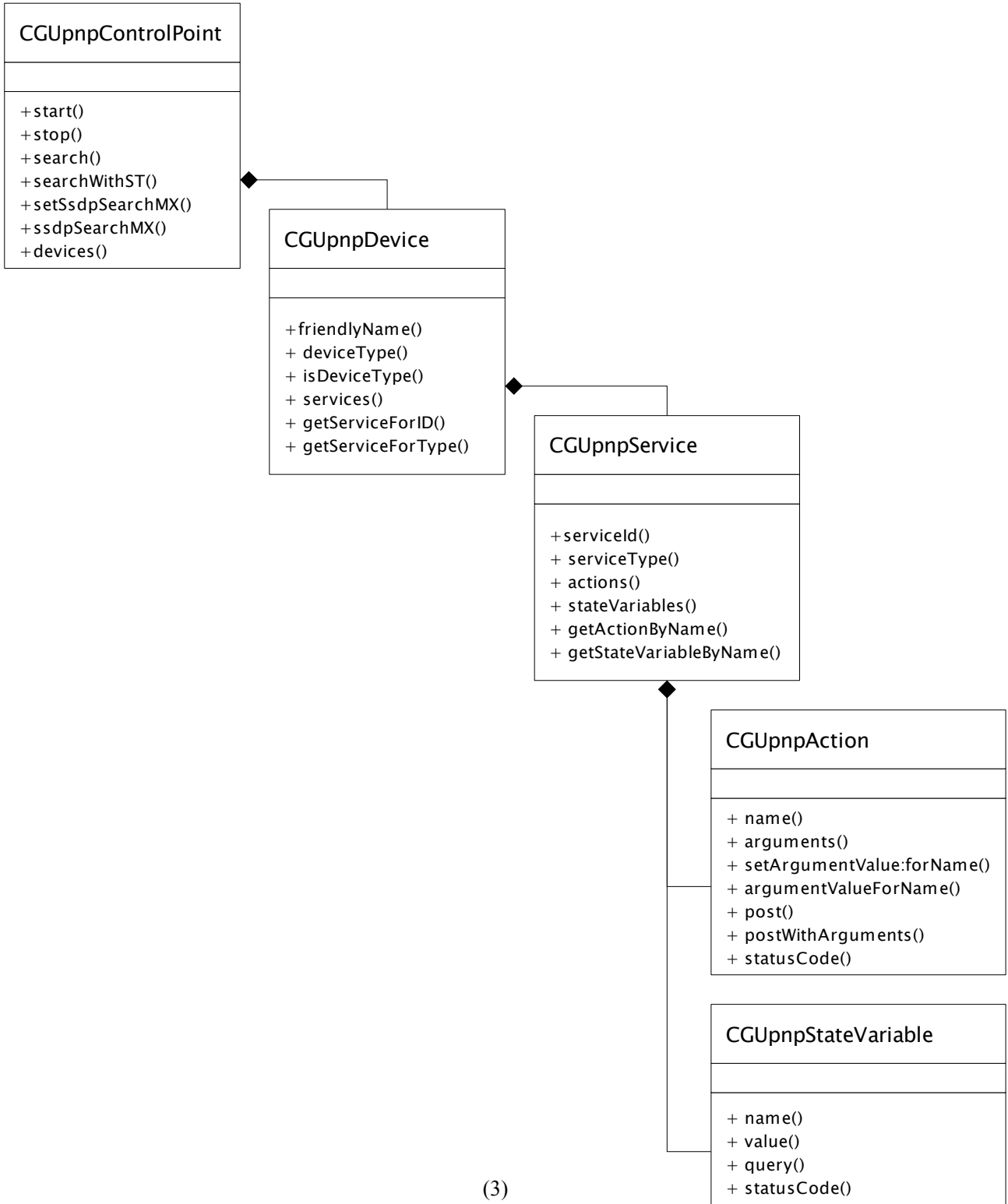
The framework is distributed as a install package as the following. Using the installer, the framework is installed into '/Library/Framework/CyberLink.framework' as default.



3 Control Point

3.1 Class Overview

The following static structure diagram is related classes of CyberLink to create your control point of UPnP™. The control point has some root devices in the UPnP™ network.



3.2 Initiating

To create a UPnP™ control point, create an instance of `CGUpnpControlPoint` class. The new instance is activated automatically using `CGUpnpControlPoint::start`. Use `CGUpnpControlPoint::search` or `searchWithST` to find the devices in the local network.

```
#import <CyberLink/UPnP.h>
... ..
CGUpnpControlPoint *ctrlPoint = [[CGUpnpControlPoint alloc] init];
... ..
[ctrlPoint search];
```

3.3 Root Devices

Use `CGUpnpControlPoint:devices` to get all root devices which the control point found. The method returns an `NSArray` object which has the devices as instances of `CGUpnpDevice`.

```
#import <CyberLink/UPnP.h>
... ..
CGUpnpControlPoint *ctrlPoint = [[CGUpnpControlPoint alloc] init];
... ..
[ctrlPoint search];
NSArray *devArray = [ctrlPoint devices];
for (CGUpnpDevice *dev in devArray)
    NSLog(@"%@", [dev friendlyName]);
```

3.4 Control

The control point can send action or query control messages to the discovered devices. To send the action control message, use `CGUpnpAction:setArgumentValue:forName` and `CGUpnpAction:post`. You should set the action values to all input arguments, and the output argument values is ignored if you set. The following sample posts an action control request that sets a new time, and outputs the response result.

```
CGUpnpDevice *clockDev = ...
CGUpnpService *timeService = [clockDev getServiceForType:@"urn:schemas-upnp-org:service:xxxx:1"];
CGUpnpAction *setTimeAct = [timeService getActionForName:@"SetTime"];

NSString *currTime = ....
[ setTimeAct setArgumentValue: currTime forName:@"NewTime"];
if ([setTimeAct post]) {
    NSArray *argArray = [ setTimeAct arguments];
    for (CGUpnpArgument *arg in argArray)
        NSLog(@"%@ = %@", [arg name], [arg value]);
}
```

```
}
```

Similarly, to send the query control message, use `CGUpnpStateVariable::query`. The following sample posts a query control request, and output the return value.

```
CGUpnpDevice *clockDev = ....  
CGUpnpService *timeService = [clockDev getServiceForType:@"urn:schemas-upnp-org:service:xxxxx:1"];  
CGUpnpStateVariable *timeStateVar = [timeService "time"];  
if ([timeStateVar query])  
    NSLog(@"%@ = %@", [timeStateVar name], [timeStateVar value]);
```

4 License

Copyright (c) 2008 Satoshi Konno

Collectively the Copyright Owners All rights reserved.

Subject to the below, redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the names of the Copyright Owners nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT OWNERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

While any of the respective Copyright Owners licenses and/or distributes this software and any files contained therein under its or third party copyrights, for clarification it is hereby stated that no patent license explicit or implied is granted by any of the Copyright Owners in connection to this license or distribution of this software.