# Roomie Remote
# Device Development Kit

**Version 2.1.0**

2/1/2014

# Introduction

There are four sample method files in this package representing the fundamental methods Roomie uses to communicate with devices. Each is described below. Commands are contained in RoomieCodes.plist files for all communication methods. Multiple command sets of any type can be combined into one RoomieCodes file.

The “Sample Remote” folder demonstrates Mapped Remote Design editing.

This document is intended to cover the following use cases:

1. Adding/Editing Infrared Command Sets
2. Adding/Editing IP Command Sets
3. Adding/Editing Serial Command Sets
4. Adding/Editing Mapped Remote Designs

**These files should be edited using Xcode,** freely available on the Mac App Store**.** If you don’t have access to Xcode, any high quality XML text editor will do the job such as BBEdit, it just requires more work to make sure you maintain the XML formatting whereas Xcode removes that need and provides an easier to use GUI. Do not use low-quality text editors such as Notepad on Windows as they’re certain to corrupt the XML formatting. *For Mapped Remote Editing, Xcode should be considered a hard requirement. If you’re just editing some codes, you can get by with a very good text editor.*

**Infrared learning is built into Roomie – there is no need to use this DDK for basic learning of infrared codes to use with your configuration.** This means using the techniques described here in general are not necessary for infrared devices. However, there are some cases where this may still be the best method such as a manufacturer’s device not in the main library that has made available an official infrared code document as calculated codes may in some cases be more accurate or provide better response than learned codes. We would also encourage you to submit such documents to us for inclusion in the main library.

# Quick Start: Add an Infrared Device

1. First, use your iTach adapter to learn each code from the target device per the IR learning FAQ. This step may be skipped if you already have the codes in a document form from another source. If you learn the codes from within Roomie, the rest of the steps below are not needed. The remaining steps are for use with manual addition of codes to Roomie from an external source such as a document or iLearn, an application provided by Global Caché for lower level code learning.

2. Open the file 'RoomieCodes-Template.plist' in the 'Sample Infrared' folder of this DDK. Match the first command name you learned to the *Special Command Names* section of this document and insert it in the Template file where it says 'INSERT NAME OF COMMAND HERE' in line 8. If you're using Xcode, you wont see line numbers but you will see the text. For instance, if you learned a 'PLAY' command, you would edit line 8 at the first key/string combination in the Template to look something like:

```
<key>PLAY</key><string>38000,%d,1,96,32,16,[shortened]2492</string>
```

3. Optionally, edit the code itself if necessary by changing the "1" after the frequency to "%d" as shown in the SAMPLE GC CODE at line 13. This enables repeat functionality for the code. If you are using Pronto codes instead of Global Caché codes, your code line may look more like this (another sample of this is shown in line 16 of the Template):

```
<key>POWER TOGGLE</key><string>0000 006E 0000 [shortened] 0EB5</string>
```

4. Continue in the same way for each code you have learned. One line per code. Just put the command name in all caps between the "key" tags and the code between the "string" tags. Both Global Caché style codes and Pronto style codes are accepted and are interchangeable.

5. Edit the type, brand, and category values in the Template. The type should be one of the types listed in the *Available Device Types* section of this document. Whatever matches your device most closely is fine. Enter the numeric value such as 16 for a Receiver. Finally, edit the brand and category to be whatever describes your device best such as 'Panasonic' and 'BDT-220'.

6. Now that you have completed entering the codes and set the type, brand, and category (do not change the 'gc' value listed as the method – that is the value all infrared devices use), it is time to clean up the file. Delete all of the lines marked 'DELETE THESE LINES' and all the other sample code lines.

7. Save the file with the name ‘RoomieCodes.plist’. Backup your Roomie configuration to Dropbox using Roomie’s Settings panel. Then, insert the new RoomieCodes.plist file into the resulting directory and Restore your configuration to Roomie.

8. Using Roomie, you can now confirm the device has been added by looking at the display in Settings labelled “Custom Devices” which will show you a count of the number of custom devices and if you tap into that show you the device itself.

9. Moving to the appropriate room in Roomie and tapping Edit, you will now be able to tap Add Device, select your iTach adapter or L5, and then the device type you entered above, the brand above, and the category above to add the custom device to Roomie.

10. If you need to add another custom device to the same RoomieCodes file, you would perform the same steps above except duplicate the entire section from the first ‘dict’ tag to the last ‘dict’ tag. Edit the new section to match your second custom device and leave the first ‘dict’ section unchanged. Note that the sample in the ‘Sample Serial’ folder of this DDK also shows exactly how to format multiple devices in one RoomieCodes file.

Once you’ve completed adding your custom device and verified that it is operational, feel free to submit it to us via email at support@roomieremote.com for inclusion in the main library.

---

# Custom Devices

The following sections go into a greater level of detail than the above walkthrough. They also describe other control methods such as IP, HTTP, and Serial. The following are the four primary types of control methods provided by Roomie:

## 1. Infrared

The format of infrared commands follows the Global Caché Infrared format available from their website. The "%d" wildcard is replaced when sending the command with a number of repetitions. This format is also used by the built-in learner functionality in the iTach series. See the IR learning FAQ for information on learning IR codes.

Roomie also supports direct Pronto codes as shown in the sample infrared file. Global Caché and Pronto IR formats can be used interchangeably in the RoomieCodes file and will be converted by Roomie automatically as needed.

A special case not depicted in the sample is that of toggle codes – that is a special type of infrared code that changes every other time it is sent. It was a technique used by mostly older devices to try to prevent unintended commands. If you are entering a command for a toggle code infrared device, separate the second code from the first in the same command entry with a semicolon like this:

```
36000,%d,1,96,32,16,[...],32,2492;36000,%d,1,96,32,16,[...],32,2492
```

Note that true toggle IR codes are extremely rare. The likelihood that your device uses toggle codes unless it is a Microsoft/Marantz/Philips device we already have codes for is almost zero.

Conversion utilities are available on the Internet to convert between other popular IR formats and the Global Caché format. There is no need to convert if your codes are in the common Pronto format and begin with 0000 or 5000.

## 2. IP – HTTP

The first line of an HTTP method command is always the base URL. If a separator blank line is included, everything above the separator line will be treated as a header. Otherwise, and in the case of no separator line, everything is treated as body. The examples shown in the ‘Sample HTTP’ folder are for an HTTP POST device with 2 special header lines and an XML body. To use GET instead of POST, simply remove the “POST” before the base URL.

```
[POST] URL
[Optional Header Lines]
[Separator Line if Header Lines Exist]
[Body Text if Applicable]
```

The only required segment is the URL.

## 3. IP – TCP

This example shows a simple TCP device and also demonstrates several “internal” commands used by Roomie. The “.STATUS” command is used by Roomie to provide regular refreshes with various device types such as receivers. Any command preceded by a “.” and ending with ‘FORMAT’ or ‘STATUS’ is hidden from users and is intended only for use by Roomie internally for two-way feedback. Commands preceded by “.” but not ending in those two keywords are offered to the user as parameterized commands that allow the user to configure up to 3 variable parameters.

## 4. Serial

The first example here is for one of the very rare binary devices. There is no command entry difference between an IP TCP device and a Serial device. However, in this case, the device is a binary device and thus the commands themselves are entered in hex notation rather than as text.

The second example demonstrates more advanced forms of commands. The .VOLUME SET command is used in any IP or Serial command set to indicate that the device can be explicitly set for volume with the parameter placed at the ## marker. Note this command also demonstrates the way to write binary commands in ASCII notation using the common ‘\x0D’ style notation representing the hexadecimal value of ASCII 13. Binary commands or bytes can be written in either the \xXX style or pure binary regardless of whether the method for the command set is binary or something else. Editing pure binary in a command set is exceptionally difficult as it is encoded, so we recommend either always using \xXX notation with string parameters or using Xcode which makes it possible to edit raw binary data without the encoding. The .VOLUME SET command must be paired with a .VOLUME FORMAT command. Defined Volume Formats are detailed later in this document. If a particular format is not defined, it will not be able to be interpreted. Please contact support@roomieremote.com if you would like a format added.

Note that there is really no difference from Roomie’s perspective between IP TCP and Serial. The method types can be used interchangeably. For example, a “lineio” device at TCP port 23

is identical to a Serial device using “lineio”. The only way in which they are different is their device configuration which is not a part of the command set and thus not reflected in a RoomieCodes file.

# Methods

The prior samples provide an overview into the methods used by Roomie to communicate with devices. The following is a complete list of available methods and their meaning. A command set must include type, cat, brand, method, and an array of codes.

| | |
|---|---|
| **gc** | Global Caché Infrared Method (also supports Pronto Infrared Codes) |
| **binary-ack** | Binary TCP or Serial Method requiring a response before new data |
| **binary** | Binary TCP or Serial Method that does not require any response |
| **lineio** | TCP or Serial Method using text commands with CR line endings |
| **lineio-crlf** | TCP or Serial Method using text commands with CRLF line endings |
| **http** | HTTP Method including GET/POST, body, and header functionality |
| **udp** | UDP Method for sending raw UDP packets |
| **rtsp** | Live Video via RTSP over UDP |
| **rs** | Roomie's Internal Wi-Fi Synchronization Protocol |
| **onkyo** | Convenience to wrap simple Onkyo commands for Onkyo's encodings |
| **samsungdtv** | Samsung Packet Encoding |

Method names must be entered using the exact syntax above.

## Device Types

The type field must be filled in with an appropriate numeric device type. The following are available device types:

| | | |
|---|---|---|
| 0 | A/V System | |
| 1 | Auxiliary | |
| 2 | Blu-rayPlayer | |
| ~~3~~ | ~~Cable~~ | *Deprecated, use 18* |
| 4 | CD | |
| 5 | DTV Converter | |
| 6 | DVD | |
| 7 | DVD/VCR Combo | |
| 8 | Game System | |
| 9 | Home Theater System | |
| 10 | iPod Dock | |
| 11 | LaserDisc | |
| 12 | Lighting | |
| 13 | MediaPlayer | |
| 14 | Multizone System | |
| 15 | Projector | |
| 16 | Receiver/Pre-Amp | |
| ~~17~~ | ~~Satellite~~ | *Deprecated, use 18* |
| 18 | Set Top Box | |
| 19 | Soundbar | |
| 20 | Subwoofer | |
| 21 | Switcher | |
| 22 | Tuner | |
| 23 | TV | |
| 24 | TV/DVD Combo | |
| 25 | TV/DVD/VCR Combo | |
| 26 | TV/VCR Combo | |
| 27 | VCR | |
| 28 | Video Processor | |
| 29 | Climate Control | |
| 30 | Video Camera | |

# Reserved Command Names

Roomie treats various command names as special. Button types and other functions are automatically generated based on Roomie analyzing the various commands available from a given device. **Therefore, you must take special care not to pick common command names at will**, but rather to use standard command names from the following lists whenever possible. Of course, if a command you are learning is not in the set below, you may call it whatever you like, but in general it's a good idea to make sure to use the names below if the command you're entering performs that function. Capitalization matters.

The following is a list of the most common standard command names. Another way to look for command names is to watch the command bar at the top of the screen while using Roomie when using existing devices.

**Top Bar Key Commands**
POWER ON, POWER OFF, POWER TOGGLE
VOLUME UP, VOLUME DOWN, MUTE TOGGLE, MUTE ON, MUTE OFF
CHANNEL UP, CHANNEL DOWN

**Directional Pad Key Commands**
CURSOR UP, CURSOR LEFT, CURSOR RIGHT, CURSOR DOWN, CURSOR ENTER

**Play Pad Key Commands**
PLAY, PLAY PAUSE TOGGLE, STOP, RECORD, PAUSE, NEXT, PREVIOUS, FORWARD, REVERSE, SLOW FORWARD, SLOW REVERSE, REPLAY, SKIP

**Number Pad Key Commands**
DIGIT [0-9], DIGIT ENTER, DIGIT SEPARATOR, DIGIT 100, PREVIOUS CHANNEL

**Other Commands**
FUNCTION BLUE, FUNCTION GREEN, FUNCTION YELLOW, FUNCTION RED

Any command beginning with "." and ending with the word "STATUS" is treated as a potential feedback. Custom feedback is not supported at this time so all feedbacks must use a known format as defined in a later section.

**Injection Commands**
.**PLAY SET** Sets the desired position from the progress device. Default format is seconds.
**.CHANNEL SET** Sets the desired channel from Roomie's guide. No format.
**.KEY DOWN** Sends an individual keystroke such as "A". No format.

**.KEY UP** If needed by the device, Roomie can also tell it the key is now up.
**.KEY DELETE** Precisely specify how to delete a character from the device.
**.KEY STRING** Sends a complete string of characters updated with each change.
**.KEY STRING END** Override the previous command to send complete strings only when the user has completed the field.

Injection commands support parameters. Parameterized commands must begin with "." in the command name. Parameter 1 is demarcated by ##, parameter 2 by $$, and parameter 3 by ^^. Multiple occurrences of ## will all be replaced by parameter 1 and the same rule applies for each parameter.

**Format Commands:**
.VOLUME FORMAT

Any command ending with the word "FORMAT" prefixed with "." is treated as a format specifier for the data type indicated. For instance, .VOLUME FORMAT specifies a particular volume format used for the .VOLUME SET command.

Before crafting your own command names for commands not mentioned above, we recommend looking through similar devices in Roomie for appropriate command names. Roomie keys off many different command names for many different types of devices for remote formatting purposes, feedback, and other functionality.

# Data Formats

Any device with the ability to provide feedback should be assigned a format specifier. These are the currently supported formats, beginning with volume formats:

| | |
|---|---|
| 1 | Pure DB (Exactly as shown on receiver. “12.0”, “-35.5”) |
| 2 | Pure DB No Decimal (Per receiver without decimal. “120”, “-355”) |
| 3 | Pure DB Top Only (Per above without decimal. “12”, “-35”) |
| 4 | Yamaha Pre-2008 Binary Format |
| 5 | Onkyo THX Volume Format (-81-19 no decimal) |
| 6 | Anthem (-90-10 no decimal) |
| 7 | B&W Panorama (ASCII 0 through ASCII R, 35 values) |
| 8 | Sony Binary Volume Format |
| 9 | Marantz Old Volume Format (Pre-Denon merge) |
| 50 | Denon (Modern Denon/Marantz volume format) |
| 100 | Pioneer |
| 101 | Pioneer Old (Custom Pioneer volume formats <= 2009) |
| 200 | Direct Value Hex (0-100 in hexadecimal. “05”, “64”) |
| 201 | NAD (-99 to 19 no decimal) |
| 300 | Percentage (0-100 in decimal. “5”, “100”) |
| 301 | iTunes Custom Volume Format |
| 302 | VLC Custom Volume Format |
| 400 | 0-10 (Epson) |
| 401 | 0-20 (Epson) |
| 402 | 0-30 (Epson) |
| 403 | 00-60 (Sharp TV, leading zero 0-60) |
| 404 | 00-63 (Panasonic TV, leading zero 0-63) |
| 405 | 0-99 (McIntosh) |
| | |
| 500 | SmartLinc Device |
| 501 | ISY Device |
| 502 | SmartLinc Scene |
| 503 | ISY Scene |
| 504 | Lutron Integration ID |
| 505 | Lutron Button |
| 506 | ISY Program ID |
| 507 | Somfy Device |
| 508 | Global Caché Port |
| 509 | Philips Hue Light |
| 510 | Philips Hue Saturation |
| 511 | Philips Hue Hue |
| 512 | Vera Light |
| 513 | Vera Scene |
| 514 | Indigo Device |
| 800 | 00-99 (Generic Zero Prefix non-volume value) |
| 1000 | Channel |
| 2000 | Application Name (Roomie Agent) |
| 2001 | Script Name (Roomie Agent) |

## Dynamic Command Parameters

Roomie supports many special keywords inside the content of a command. Roomie will automatically replace any of the following commands with the indicated value:

| | |
|---|---|
| $COUNT$ | Unique incrementing number for each command |
| $COUNT9 | Unique incrementing number for each command (mod 10) |
| $DEVICENAME$ | System name for the current device, cleaned of quotes |
| $DEVICEID$ | Device specific UUID |
| $MAC$ | Local WiFi MAC (entered in Settings), colon delimiters |
| $RMAC$ | Remote device MAC (from “persistent” device key) |
| $DMAC$ | Local WiFi MAC, dash delimiters |
| $LMAC$ | Local WiFi MAC, no delimiters |
| $TAG$ | 16 character Unique ID |
| $SONOSID$ | Sonos native serial number format |
| $SERIAL$ | Serial number (from “serial” device key) |
| $HUEID$ | Philips Hue authenticated username |
| $SONYCOOKIE$ | Sony authenticated cookie |
| $COOKIE$ | Normal Cookie |
| $LGTOGGLE$ | Toggle between \x01 and \x02 on each command |
| $AUTH1$ | Insert user’s first authentication parameter for the device |
| $AUTH2$ | Insert user’s second authentication parameter for the device |

## Loading Commands into Roomie

Follow these steps to load a new command set into Roomie:

1. Finish your modifications to the commands.
2. Ensure the command set contains definitions for type, cat, brand, method, and an array of codes as shown in the samples.
3. Make sure the file containing the command set is named “RoomieCodes.plist” like the samples.
4. Choose whether to load your commands via Roomie Agent or Dropbox. For simplicity, first use the “Backup to Dropbox” or “Backup Configuration” features from Dropbox or Roomie Agent respectively depending on which method you have selected.
5. You should now see your “Roomie.plist” file in the Dropbox Roomie directory or Roomie Agent saved configuration directory.
6. Add the RoomieCodes.plist file to the same directory.
7. Use “Restore from Dropbox” or “Restore Configuration” from Roomie Agent as appropriate.

Your command set should now be loaded into Roomie. Note that loading RoomieCodes.plist into Roomie removes any prior changes you have made using this method, so every time you do this the slate is clean (any commands from a prior load will no longer exist). You may also add multiple devices to RoomieCodes.plist. Loading RoomieCodes.plist with no command sets removes all custom commands.

# Editing Mapped Remotes

The Sample Remote directory contains 3 sample files required for adding a Mapped Remote to Roomie:

1. remote-samsungtv2013.png
2. remote-samsungtv2013@2x.png
3. remote-samsungtv2013.xib

Name your files whatever you would like as long as the structure of this triple is maintained. The first PNG should be in non-Retina resolution with a maximum width of 320 and of arbitrary height. The second should be the same image in Retina resolution with a maximum width of 640 and a height double that of the first image. Using larger sizes is allowed and Roomie will expand the remote field to accomodate. However, anything wider than 320 will not fit on an iPhone size screen. The XIB file of matching filename is where the mapping takes place. This file must be edited with Xcode.

The *Sample Remote.xcodeproj* file is provided as a simple way to open the image (.png) and map (.xib) together for editing in Xcode, the most mature user interface layout tool available. The xcodeproj file has no relevance to Roomie itself and it should not be included in your images folder or Roomie configuration. Only the 2 required or 3 with Retina resolution files should be placed into your Roomie images folder when finished.

Layout your Mapped Remote exactly as shown in the sample. You may add new buttons, resize them, delete them, etc. It's generally safest to copy existing buttons if you want to create new buttons and then edit them. For simplicity, setting the button name to the command name will become a part of the Mapped Remote. For instance if you set the title of a button to "DIGIT 0", that command will be applied to whichever device the Mapped Remote is matched against later. If you do not set a button name, the button will be blank and available for customization after import to Roomie.

Once all of your buttons are in the proper positions and all button titles are set to their appropriate command names, save the files and then move the 1-2 .png files and .xib file into either your Roomie images folder on Dropbox or the images folder in the configuration folder of Roomie Agent. Restore your configuration from either of those to add the Mapped Remote. The mapped image will be available as a Background Image when editing Remote Designs from the Custom tab. If you select any background image with an associated .xib mapping, the Remote Design will automatically switch over to the mapped style and allow editing of all of the buttons you've created.