

SMART HOME HACKS™

Tips & Tools for Automating Your House



O'REILLY®

Gordon Meyer

HACK
#91

Track Home Events with iCal

Gain insight into your home's behavior by displaying events in calendar format.

To make your home automation system smarter, you need to have a good understanding of how the system sees your typical, day-to-day activities in the home. That is, discernable patterns derived from your motion detectors, lamp usage, and changes to the home's thermostat can be used to automate these actions for you. For example, by examining your home automation system's logs, you might notice the air conditioning is turned up a few minutes after the garage door sensor is triggered in the late afternoon. This occurs, of course, when you return home from work. It's simple to automate this, but you might not notice it if you weren't looking at the log files.

On the other hand, slogging through log files is tedious work, and they're usually so detailed that finding useful tidbits requires a lot of luck. What would be more useful is a timeline-based graphical view of your home automation data. By combining a little bit of clever scripting with Apple's iCal (<http://www.apple.com/ical/>; free), that's exactly what this hack creates.

This hack works by adding three AppleScript functions via Indigo's support for [attachment scripts](#) [Hack #88]. The new functions enable you to log device events and status changes to iCal calendars. The hack automatically creates the calendars, groups events, and messages according to how you specify them, as shown in [Figure 7-12](#).

In this example, separate calendars are used to log changes in house mode, Indigo application messages, motion detectors, and various other devices as reflected in the Calendars list on the left side of the iCal window. The main window shows the consolidated view of all the calendars, which is where you can discern patterns and trends in your home. For example, notice that the Master Bedroom Ceiling Fan was on from 10:30 a.m. to 11:00 a.m., even though the house was in *Away Mode* at the time. This might be the result of a flaw in your logic for when the [house is unoccupied](#) [Hack #70], or it might indicate that the fan came on automatically due to rising temperatures.

It's a neat technique, but it's best to use it in moderation. Just as log files can get overwhelming, overloading your calendar with notes about every event would quickly grow into a mess. Additionally, as nifty as iCal is, it does tend to bog down with hundreds of events. To help you avoid these problems, you have to specifically call the functions provided by this hack's attachment scripts to send data to iCal. This enables you to choose to record only important events.



Figure 7-12. Home data in iCal

If you want to track the use of a device, such as the ceiling fan mentioned earlier, you define a trigger action that calls the `LogDevice()` function, as shown in Figure 7-13.

The parameter to `LogDevice()` should be the name of the unit (as shown in the action in Figure 7-13). The function uses the name to retrieve information from Indigo about the device and its state. Here's the script that implements the function:

```
using terms from application "Indigo"
  on LogDevice(DeviceName)
    tell application "Indigo"
      log "Log Device: " & DeviceName
      if (name of device DeviceName exists) and (description of device
DeviceName does not contain "future") then
        set eventdate to current date
        set UnitDisc to description of device DeviceName
        set UnitType to type of device DeviceName
        set unitstate to on state of device DeviceName
        set theNote to "" --Optionally put a note here based on
other variables
        set theLocation to UnitType & " " & UnitDisc

        LogIniCal(UnitType, DeviceName, theNote, theLocation,
eventdate, unitstate)
```



Figure 7-13. Logging a device action

```

        end if
    end tell
end LogDevice
end using terms from

```

The other function you can use to send data to iCal is `LogMessages()`. You use this function to create iCal events to track things that are not device actions. The house entering Away Mode is an example of this type of event. Here's the script:

```

on logmessages(MessageGroup, message, EventKind)
    set MinSummaryLength to 5
    set theLocation to "Message"
    set eventdate to current date
    if MessageGroup contains "Messages" then set EventKind to false

    set OldDelims to AppleScript's text item delimiters
    set AppleScript's text item delimiters to " "
    set theList to the text items of message
    set AppleScript's text item delimiters to OldDelims
    set x to count of items in theList
    if x is greater than MinSummaryLength then
        set i to 0
        repeat with n from 1 to x
            set i to i + 1
            if i is 1 then

```

```

        set LogTemp to item i of theList
    else
        set LogTemp to LogTemp & " " & item i of theList
    end if
    if i is MinSummaryLength then exit repeat
end repeat
set LogTemp to LogTemp & "..."
else
    set LogTemp to message
end if
LogIniCal(MessageGroup, LogTemp, message, theLocation, eventdate,
EventKind)
end logmessages
end using terms from

```

The `LogMessages()` function takes three parameters: `MessageGroup`, `Message`, and `EventKind`. `MessageGroup` specifies the calendar to which the event will be added. In [Figure 7-12](#), for example, you might set this parameter to `House Mode` to add an event to that calendar. The `Message` parameter is a text string that will become the title of the event when it's added to iCal, such as `Sleep Mode`. Finally, `EventKind` can be `true`, which indicates the event is beginning, or `false`, which means the event is ending. This determines the duration of the iCal event.

You call `LogMessages()` using the same method as `LogDevice()`—using an event trigger with an embedded AppleScript. To log when you're leaving home and the house is entering `Away Mode`, the action would be:

```
LogMessages("House Mode", "Away Mode", true)
```

When `Away Mode` is turned off, this call ends the iCal event:

```
LogMessages("House Mode", "Away Mode", false)
```

Finally, there's one more piece to discuss. Both `LogMessages()` and `LogDevice()` rely on another attachment script function, `LogIniCal()`. This function does all the heavy lifting of communicating with iCal, creating and ending events, adding new calendars, and so on:

```

on LogIniCal(theCalName, theSummary, theNote, theLocation, eventdate,
EventKind)
    tell application "iCal"
        set Callist to (title of every calendar)
        if theCalName is not in the Callist then create calendar with
name theCalName
        set TargetCal to (first calendar whose title is theCalName)

        if EventKind is true then
            --Make new started event
            make event at end of events of TargetCal with properties
{start date:eventdate, summary:theSummary, description:theNote, location:
theLocation, status:tentative}

```

```

else if EventKind is false then
    try
        set EventList to (every event whose summary is
theSummary) of TargetCal
        set TargetEvent to (last item of EventList whose status
is tentative)

        if status of TargetEvent is tentative then
            set (end date) of TargetEvent to eventdate
            set status of TargetEvent to confirmed
        end if
    on error
        make event at end of events of TargetCal with properties
        {start date:eventdate, summary:theSummary, description:(theNote & " (no
starting event found)"), location:theLocation, status:cancelled}
    end try
end if
end tell
end LogIniCal

```

You can combine all three of these scripts into one AppleScript. Save it in compiled format using Script Editor, and install it in Indigo's attachment script folder (*~/Documents/Indigo User Data/Scripts/Attachments*). Then, choose Reload Attachments from Indigo's script menu.



To download these scripts, visit Greg's web site (<http://homepage.mac.com/gregsmith/icalstats.html>).

Hacking the Hack

If you want to see what's happening in your home while you're away, set up iCal to automatically publish the calendars you're interested in over the Internet. You will be able to view them in a web browser or subscribe to the remote calendars from another computer's copy of iCal.

For more on using iCal with home automation, see "Remember Important Events" [Hack #25]. If you want to view data about your home automation system using charts and graphs, see "Chart Home Automation Data" [Hack #92].

—Greg Smith