# Understanding Sleep Proxy Service

December 2009, updated periodically

Mac OS X 10.6 SnowLeopard, launched August 2009, included a new feature, Sleep Proxy Service, also known as Wake on Demand.

This page describes what you can expect Sleep Proxy Service to do, and, if it doesn't seem to be working correctly, how you can investigate what's going on.

## What to Expect

Sleeping machines can be woken automatically when needed to provide network services. This lets you put machines to sleep to save power, without losing accessibility to the network services those machines provide. Examples of network services include:

- Macs with USB-connected printers, providing Printer Sharing.
- Macs with iTunes media, being served to Apple TVs or other iTunes machines.
- Back to My Mac machines at home that are sleeping, and then woken on demand when you connect to them (from inside or outside the home).
- Command-line login using ssh.
- Any other Bonjour-advertised service.

## Hardware You Will Need to Run the Tests Described Here

1. A Mac, providing network services, that will go to sleep.
2. A Sleep Proxy, that will wake the sleeping machine on demand.
3. A client, that will access the sleeping machine (1), causing the Sleep Proxy (2) to wake it.

## Testing Sleep Proxy Service

### 1. The Mac That Will Go to Sleep

The Mac that will go to sleep needs to be offering some Bonjour-advertised network services (e.g. services enabled in Sharing Prefs, or apps like iTunes running with sharing enabled).

It needs to have "Wake for network access" enabled in the Energy Saver Prefs.

In the case of a laptop, due to power management rules enforced by the current system software, it needs both (a) to be connected to AC power and (b) to have a display available (either lid open, or an external display connected).

Sleep Proxy wakeup is supported on all Macs connected via Ethernet.

On Macs connected via Wi-Fi, Sleep Proxy wakeup is supposed to be supported when associated with compatible Wi-Fi base station. Current Apple AirPort base stations (running up-to-date firmware) are all supposed to support Wake on Wireless LAN (WoWLAN), but there have been reports of it not working reliably, and other Wi-Fi base stations may not support WoWLAN at all. When troubleshooting Sleep Proxy you should test initially using a server Mac connected via Ethernet, to eliminate Wi-Fi compatibility issues. If that works reliably, you can then move on to testing a server Mac connected via Wi-Fi.

### 2. The Sleep Proxy

In normal operation, the Sleep Proxy will be a low-power device that is always on, like an AirPort base station or Apple TV. No configuration is required. These devices automatically offer Sleep Proxy service on the network. It's okay to have more than one Sleep Proxy on the network — the Mac going to sleep will pick the best one automatically.

In the instructions below I'll show you how to use a Mac as your Sleep Proxy, instead of an AirPort base station or Apple TV. Being able to easily view the system log on your Mac gives you better visibility into what's going on and what the Sleep Proxy is doing.

### 3. The Client

Anything. Tiger, Leopard, SnowLeopard, Lion, Windows, Linux, iPhone, iPad, Android, or anything else that has TCP/IP, it doesn't matter. Connect to your sleeping server from any client device and the server will magically wake to serve you, regardless of whether you connect by IP address, by hostname, or via Bonjour browsing. The server will wake regardless of whether the client is right next to it on the same link, or 30 Internet hops away on the other side of the planet.

### Reachability

Other Wake-on-LAN solutions require the client be be on the same LAN segment (same broadcast domain) as the sleeping server that will be woken. Many Wake-on-LAN solutions *claim* to work remotely, but really they only work as long as they're lucky and ARP cache entries don't time out. You may put a server to sleep and then immediately test it by waking it from a remote client, and it may wake successfully, and it may be tempting to conclude from that superficial short-term test that you have a reliable long-term solution, that you can depend on days or weeks from now when you're away from home and your need your computer to wake up for you. In reality, the short-term test only works if you're lucky and the router's ARP cache entry for the sleeping server has not yet timed out. If you put the machine to sleep and then wait an hour or two before you try to wake it remotely, you may find it's not so dependable after all.

The Sleep Proxy Service solves this problem by being your proxy on the same LAN segment as the sleeping server, so it can wake the sleeping server directly, without going through the router's ARP cache. You may not be able to reach the sleeping server directly to wake it, but the Sleep Proxy Service can, on your behalf. The Sleep Proxy Service notices your attempt to access the server, and wakes it for you, reliably, every time, without failures caused by timed-out ARP cache entries. If your server is currently reachable across the public Internet when it's awake, then you can now reach it the same way even when it's asleep. There are a variety of ways to have a server be reachable across the public Internet, all of which work with Sleep Proxy Service. Examples include:

- Giving the server a genuine public global IPv4 or IPv6 address, unhindered by firewalls or NATs that will block traffic.
- Using a wide-area service directory like Wide-Area Bonjour [DynDNS page] [AFP548 page 1] [AFP548 page 2], along with a home NAT gateway that supports either NAT Port Mapping Protocol or the newer Port Control Protocol.

- Using Apple’s Back to My Mac service.
- Manually creating a static NAT port mapping.
- Manually configuring a default NAT host (also sometimes called a “DMZ host”).

The remainder of this article assumes that if you want to have a server be reachable across the public Internet you already know how to do that; the focus here is on how to have that server device sleep to save power, and wake automatically when needed.

## Technical Details:

What Will Wake a Sleeping Machine?

To reduce unnecessary wakeups, the Sleep Proxy will wake a sleeping machine **only** when the proxy sees an *incoming IP UDP* packet or *incoming IP TCP connection request* for a Bonjour-advertised service on the sleeping machine. In particular this means that “pinging” a sleeping machine WILL NOT wake it, since “ping” is not one of the Bonjour-advertised services on the sleeping machine.

There is one exception to the rule above, for the benefit of command-line users: For TCP port 22 (ssh) the Sleep Proxy will *also* wake a sleeping machine for plain data packets, not just incoming IP TCP connection requests (TCP SYN). This means that you can log in to a machine via ssh, put it to sleep with “pmset sleepnow” and then the next time you type a keystroke in that window, the sleeping machine will magically wake for you. When you’ve finished what you wanted to do, you can let it idle sleep by itself, or for maximum power savings type “pmset sleepnow” to put it straight back to sleep immediately.

## Running the Tests: Using a Mac as your Sleep Proxy

### Part One: Setting up the Sleep Proxy Mac

1. In Energy Saver Prefs, set your test sleep proxy Mac to never sleep

2. Edit your test sleep proxy Mac’s `/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist` file to add “-OfferSleepProxyService” and the priority (10) like this:

```
<key>ProgramArguments</key>
<array>
    <string>/usr/sbin/mDNSResponder</string>
    <string>-launchd</string>
    <string>-OfferSleepProxyService</string>
    <string>10</string>
</array>
```

**Important: Perform this edit carefully, since if the `/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist` file is malformed or corrupted, it will prevent the machine from booting.**

**Also, if you edited the file using emacs, you MUST remove the emacs backup file (the file with a tilde at the end of the name, `/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist~`) or your Mac will not start up.**

**Important: Do not skip the previous check! If you do not remove the emacs backup file your Mac will not start up! If this happens you’ll have to boot your Mac from an external disk or bootable DVD (or put your Mac into Target Disk Mode and connect it to another Mac to access the filesystem) to remove the offending file from your boot volume.**

3. Restart your test sleep proxy Mac.

4. After it has restarted, enable Sleep Proxy logging, by typing this in a Terminal window:

```
sudo killall -USR1 mDNSResponder
```

This turns on mDNSResponder debug logging. You only need to do this once to turn on mDNSResponder debug logging until the next reboot. You can reduce the volume of mDNSResponder log messages to show just Sleep Proxy messages with this command:

```
sudo syslog -c mDNSResponder en
```

Now watch `/var/log/system.log`, like this:

```
tail -F /var/log/system.log
...
Jan 13 20:18:28 santana-mini mDNSResponder[18]: SIGUSR1: Logging Enabled
Jan 13 20:18:32 santana-mini sudo[253]: cheshire : ... ; COMMAND=/usr/bin/syslog -c mDNSResponder en
```

Leave that window open while you run the tests below.

Now that your Sleep Proxy Mac is up and running, we’ll move on to the sleeping server...

### Part Two: Setting up the Mac That Will Go to Sleep

5. Set up server Mac you’re going to put to sleep.

Some newer Macs have network interface hardware that includes its own built-in Sleep Proxy. This is great, but you’re not going to see how Sleep Proxy works if the sleeping Mac is using its own built-in Sleep Proxy instead of the Network Sleep Proxy you just set up. Mac OS X 10.7 Lion includes the ability to tell mDNSResponder to ignore the built-in Sleep Proxy and use a Network Sleep Proxy instead. To tell mDNSResponder to ignore the built-in Sleep Proxy and use the Network Sleep Proxy you just set up, edit your server Mac’s `/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist` file to add “-UseInternalSleepProxy 0” as shown below.

(If your server Mac you’re going to put to sleep is not running Mac OS X 10.7 Lion you can skip this step and continue with the rest of these tests, but see Further Debugging Information below to see how to determine whether your Mac has a built-in Sleep Proxy or whether it used your network Sleep Proxy. If your Mac has a built-in Sleep Proxy and it’s not working for you, your only option is to upgrade to Mac OS X 10.7 Lion so you can tell mDNSResponder not to use the built-in Sleep Proxy. Sorry.)

```
<key>ProgramArguments</key>
<array>
```

```
        <string>/usr/sbin/mDNSResponder</string>
        <string>-launchd</string>
        <string>-UseInternalSleepProxy</string>
        <string>0</string>
    </array>
```

**Important: Perform this edit carefully, since if the `/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist` file is malformed or corrupted, it will prevent the machine from booting.**

**Also, if you edited the file using emacs, you MUST remove the emacs backup file (the file with a tilde at the end of the name, `/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist~`) or your Mac will not start up.**

**Important: Do not skip the previous check! If you do not remove the emacs backup file your Mac will not start up! If this happens you'll have to boot your Mac from an external disk or bootable DVD (or put your Mac into Target Disk Mode and connect it to another Mac to access the filesystem) to remove the offending file from your boot volume.**

6. Restart the server Mac you're going to put to sleep.

7. After it has restarted, open a Terminal window and verify that it sees your new sleep proxy with priority "10":

```
dns-sd -B _sleep-proxy._udp local
Browsing for _sleep-proxy._udp.local
Timestamp     A/R Flags if Domain      Service Type            Instance Name
20:16:11.935  Add     2  4 local.      _sleep-proxy._udp.      30-34-10-70 AirPort Base Station
20:16:44.833  Add     2  4 local.      _sleep-proxy._udp.      10-33-73-74 Mac mini
^C
```

8. Verify that mDNSResponder on your machine that will sleep has detected your new Sleep Proxy:

In a Terminal window on the machine that will sleep, type:

```
sudo killall -INFO mDNSResponder; tail -1000f /var/log/system.log
```

The section "Network Interfaces" in the syslog output lists the network interfaces on your machine that Bonjour knows about. For any interface which (a) Bonjour believes to be capable of Wake On LAN and (b) Bonjour has discovered a Bonjour Sleep Proxy, you should see a black sun ("☀") immediately preceding the address at the end of the line, like this:

```
------ Network Interfaces ------
v4 en0(4) 00:17:F2:CD:7E:0A 00:00:00:00:00:00 v4 0.0.0.0 ⊙ ⇆ ☀ 192.168.1.50
```

For any interface which (a) Bonjour believes to be capable of Wake On LAN but (b) no Sleep Proxy has been discovered yet, you should see a white sun ("☼"). This means that Bonjour is willing to use a Sleep Proxy on this interface, but hasn't found one yet.

```
------ Network Interfaces ------
v4 en0(4) 00:17:F2:CD:7E:0A 00:00:00:00:00:00 v4 0.0.0.0 ⊙ ⇆ ☼ 192.168.1.50
```

9. Select "Sleep" from the Apple menu to put your server Mac to sleep.

10. On your test sleep proxy Mac you should see messages saying a client has registered its Bonjour records, like this:

```
Jan 13 20:25:04 santana-mini mDNSResponder[18]:
    SPS Registered 1 2 18 My\032Sleeping\032Mac._ssh._tcp.local. SRV 0 0 22 my-sleeping-mac.local.
...
...
...
```

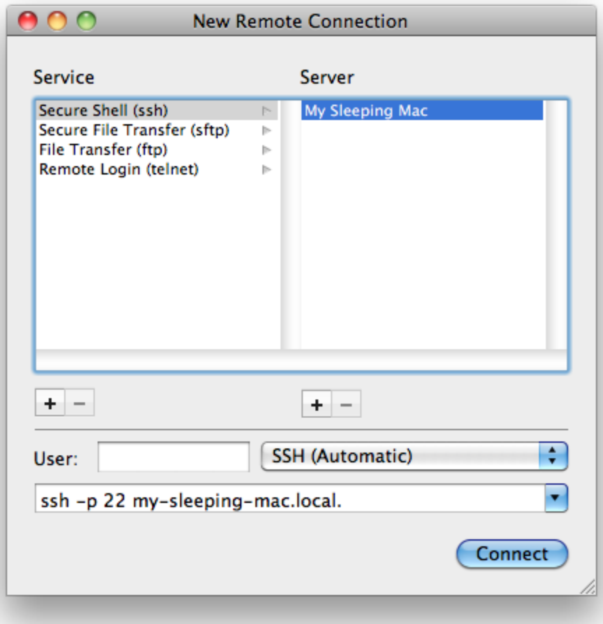
Now that your server Mac is sleeping, it's time to access it from a client and see it wake...

**Part Three: Waking the Sleeping Mac**

11. Access your sleeping server Mac over the network.

For example, if your sleeping server Mac has ssh enabled, on a client machine you can:

1. Run Terminal
2. Press Cmd-Shift-K ("New Remote Connection" on the "Shell" menu)
3. In the "Service" list, click on "Secure Shell (ssh)"
4. In the "Server" list, click on your sleeping server's name
5. The command field at the bottom of the window will auto-populate with the appropriate command
6. Click "Connect"



SSH (Automatic)

ssh -p 22 my-sleeping-mac.local.

(Or you can open a Terminal window and type: `ssh my-sleeping-mac.local`)

12. Your sleeping server Mac should wake, virtually instantaneously. Depending on the model of Mac, you may even hear its power relays click on at the exact moment you click the “Connect” button.

13. On your test sleep proxy Mac you should see messages saying why it woke the sleeping server, like this:

```
Jan 13 20:27:11 santana-mini mDNSResponder[18]: Received 74-byte TCP from 10.0.1.2:52512 to 10.0.1.201:22 SYN
Jan 13 20:27:11 santana-mini mDNSResponder[18]: Waking host at en0 10.0.1.201
    H-MAC 00:14:51:66:CB:22 I-MAC 00:14:51:66:CB:22 for 18 My\032Sleeping\032Mac._ssh._tcp.local. SRV 0 0 22 my-sleeping-mac.local.
```

The key line is the “Waking host” line. In this case the Sleep Proxy is waking the sleeping server because the Sleep Proxy saw a TCP SYN to port 22.

If your sleeping server Mac keeps waking when you don’t want it to, check the `system.log` messages on your test sleep proxy Mac and look for the “Waking” line. That should tell you what service is being accessed that is causing it to wake. If you don’t see any, then the sleep proxy didn’t wake your server. On the sleeping server Mac that woke up you can run “pmset -g log” which may give a hint as to why it thought it woke up.

### Further Debugging Information

For more information about what’s happening on the server Mac that’s going to sleep, you can turn on detailed Sleep Proxy logging just like you did for your Sleep Proxy Mac:

```
sudo killall -USR1 mDNSResponder
sudo syslog -c mDNSResponder en
tail -F /var/log/system.log
```

Now, after the server Mac has woken, you can look at its `system.log` messages to see what it did.

In particular, if you see “`BeginSleepProcessing: ... using local proxy`” then that means you need to go back and do step (5) to tell mDNSResponder not to use the local NIC proxy.

If you see “`BeginSleepProcessing: ... Found Sleep Proxy Server ...`” then that means your server Mac is using a Sleep Proxy it found on the network, and the message tells you *which* network Sleep Proxy it is using.

### Using Bonjour Sleep Proxy with Microsoft Windows

With Bonjour Sleep Proxy you can use any Windows machine (or indeed any device which supports TCP/IP) as a client to wake sleeping machines, with no additional software. Bonjour Sleep Proxy supports all TCP/IP clients.

Bonjour Sleep Proxy can also wake sleeping Windows server machines on demand.

1. Download the Bonjour for Windows SDK from Apple’s Open Source page
2. Open the `C:\Program Files\Bonjour SDK\Installer` directory and run Bonjour for Windows installer
3. Open the `C:\Program Files\Bonjour SDK\Bin` directory, double-click the Bonjour Control Panel, and turn on the “Enable Wake on Demand” checkbox.

When it sleeps, your Windows machine will now register its Bonjour-advertised services with your network Sleep Proxy.

If your Windows machine doesn’t have any Bonjour-advertised services, you can simulate that by enabling some network service, and then using the `dns-sd` command-line tool in a Command Prompt window to create a Bonjour registration on its behalf. For example, if you enabled ssh login on your Windows machine, you’d advertise that like this:

```
dns-sd -R "My Windows Machine" _ssh._tcp "" 22
```

Leave the `dns-sd` command running while doing your testing — you can minimize the Command Prompt window if you want to get it out of the way.

More information about the `dns-sd` command is available in Chapter 6 of the O'Reilly book Zero Configuration Networking: The Definitive Guide.

---

*http://www.dns-sd.org*