

LANdini

an OSC networking app for LANs
version 0.22 (2014-03-18), by Jascha Narveson
partially based on code from LNX Studio by Neil Cosgrove

What is this?

This is a small program that can act as a routing center for OSC messages between computers on a LAN, doing the background book-keeping necessary for smoothly functioning and reliable OSC communication, especially on wireless routers. LANdini also maintains a shared network time and an active user list, both of which can be accessed by external applications via OSC.

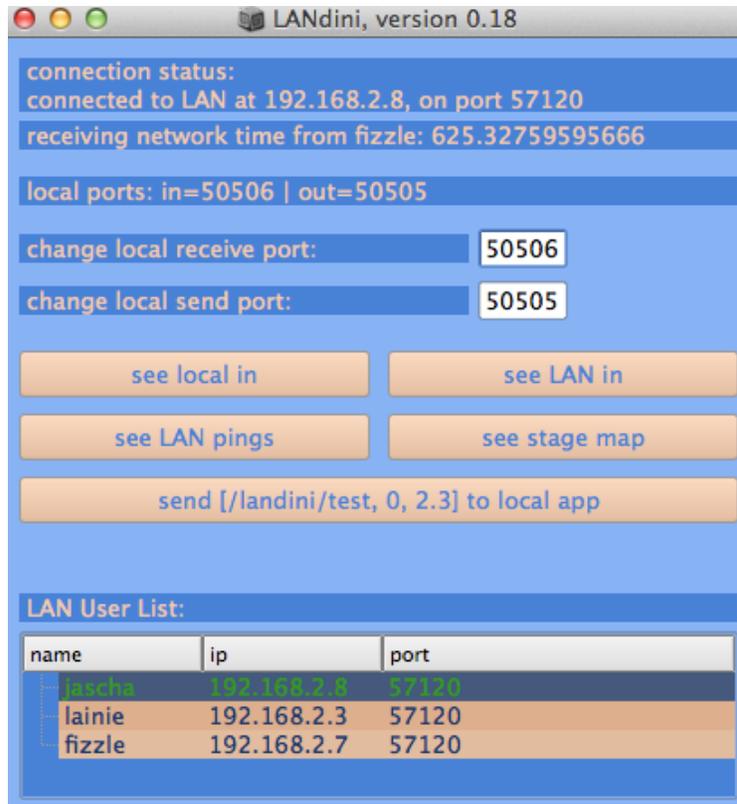
Running LANdini

LANdini currently runs on Mac OSX 10.6+. A cross-platform version is an eventual goal, but doesn't exist yet (sorry!).

Assuming you have a mac running OSX 10.6 or higher, do the following:

- 1 Make the following settings true on all the computers you want on the network:

"System Preferences..." -> Security -> Firewall: OFF
"System Preferences..." -> Sharing -> File Sharing: ON
- 2 Quit any networking apps currently running on all the computers you want to network with.
- 3 Start LANdini on all the computers you want to network with. You'll see a window like this:



The LANdini window shows you:

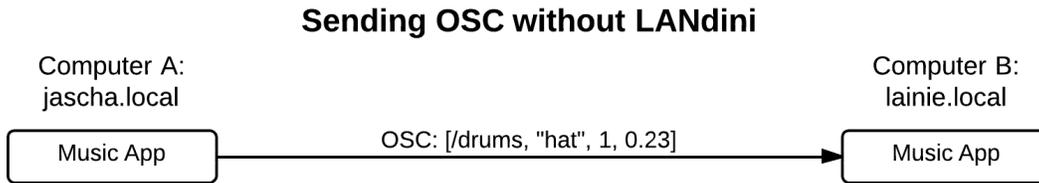
- your LAN IP address and port
- network time, including who's acting as the current time server (the time is the same on all the computers on the network).
- some options for changing send/receive ports (not recommended!)
- buttons for viewing OSC traffic and accessing the "stage map" feature (still in an early stage of development)
- a test-button for sending a dummy OSC message to your local app
- a list of active users on the bottom, including network name, IP, and OSC communication port. Your own machine shows up in green.

4

Start the software that you'll be using LANdini with. Your patch works with LANdini by sending to its local receive port (default: 50506) and listening to LANdini messages from its local send port (default: 50505). Once you've set up communication to and from LANdini, you can start interacting with it through the LANdini API.

Using the LANdini API

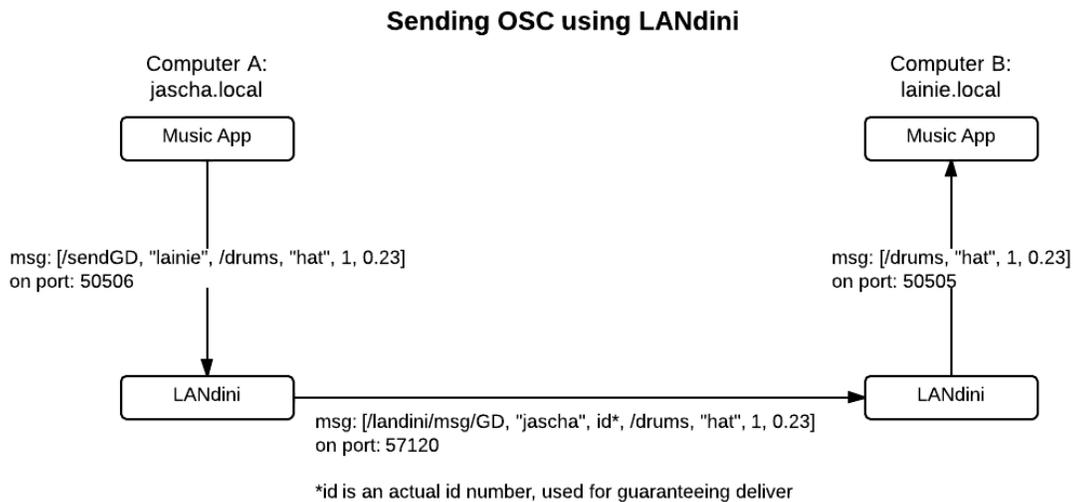
Sending an OSC message like `[/drums, "hat", 1, 0.23]` from one user on a network to another (let's say it's Jascha sending to Lainie), would normally look like this:



To use LANdini, you need to preface the OSC message with two pieces of info:

- The sending protocol, which can be one of three things:
 - `"/send"` (normal sending)
 - `"/send/GD"` ("guaranteed delivery" sending)
 - `"/send/OGD"` ("ordered guaranteed delivery" sending)
- the destination, which can be:
 - `"all"` (everyone on the network including yourself)
 - `"allButMe"` (everyone on the network but yourself)
 - `"username"` (a particular network user by name or ip)

Sending the same message as above using the guaranteed delivery protocol would look like this:



In this case, for Jascha to send a message to Lainie, he actually has to send it to the copy of LANdini on his own computer, which sends it to the copy of LANdini on Lainie’s computer, which then delivers it in its original form to her music app.

Notice that the outgoing message just has two extra pieces of info (the sending protocol and the destination), while the received message on the other end is unchanged. This makes retrofitting patches fairly easy.

These three sending protocols are central features of LANdini, but other features such as network time and the “stage map” are also included. These are accessed by sending your local copy of LANdini OSC messages of the appropriate form. The API is summarized in the chart on the next page.

IMPORTANT NOTE!

Sending to LANdini = **port 50506** over 127.0.0.1 (ie: localhost)
Receive from LANdini = **port 50505** over 127.0.0.1 (ie: localhost)

OSC Message Format	Description
<pre>["/send", (destination*), args...]</pre> <p>* destination = "all", "allButMe", or a user name</p>	<p>Send using normal OSC – it’ll either get there or it won’t, and this should be okay with you.</p>
<pre>["/send/GD", (destination*), args...]</pre> <p>* destination = "all", "allButMe", or a user name</p>	<p>Send using Guaranteed Delivery. OSC messages are guaranteed to get to their destination, though they may be slightly delayed if re-sends are necessary.</p>
<pre>["/send/OGD", (destination*), args...]</pre> <p>* destination = "all", "allButMe", or a user name</p>	<p>Sending using Ordered Guaranteed Delivery. OSC messages are guaranteed to arrive and be executed <i>in the order they were sent</i>. Execution of these messages is potentially slower than the other two methods, though is still generally fairly zippy.</p>
<pre>["/networkTime"]</pre> <p>is answered by:</p> <pre>["/landini/networkTime", time in secs]</pre>	<p>LANdini always maintains a group network time. The time server is determined alphabetically, with the next in line taking over if the current server quits or crashes. Time is given as a float value in seconds.</p>

<pre>["/numUsers"] is answered by: ["/landini/numusers", # of users]</pre>	<p>This will result in an answer from LANDini as an OSC message with the message type <code>"/landini/numusers"</code> followed by an integer representing the number of users on the LAN.</p>
<pre>["/userNames"] is answered by: ["/landini/usernames", names...]</pre>	<p>This will result in an answer from LANDini as an OSC message with the message type <code>"/landini/username"</code> followed by a series of strings representing the names of the users on the LAN.</p>
<pre>["/stageMap"] is answered by: ["/landini/stagemap", name1, x1, y1, ...]</pre>	<p>This will result in an answer from LANDini as an OSC message with the message type <code>"/landini/stagemap"</code> followed by the name, x position, and y position of each member of the LAN. Info can be collected and sorted to effect OSC send-outs that make use of the spatial information contained in the stagemap.</p> <p>For convenience, LANDini delivers the name/x/y arguments in order from "left" to "right" (ie: from low-to-high values of x).</p>

Other Features:

OSC traffic monitors

LANdini's GUI has some buttons that allow you to monitor traffic coming in on local and LAN ports. For convenience, LANdini offers two buttons for incoming LAN traffic – one that weeds out the large volume of redundant user-pings, and one that lets you see the high volume of redundant user-pings.

Each window has a filter fields in it – if it isn't blank, it will only display messages that contain the text entered in to this field. Very useful for avoiding headaches when looking for something specific!

Stage Map

The "see stage map" button will open up a window with simple representations of all the users on the LAN. You can only move the square that represents you – the other squares update with new position information as other users move themselves in to place.

Right now, LANdini can be queried to supply a simple list of users with x/y coordinates. Future versions may have more convenience methods to yield the list in different configurations, but in-house sorting in your own patch is possible using this method.

What's next?

Future versions of LANdini will feature:

- outgoing OSC traffic monitor windows
- better internal logic for setting the network time server
- ongoing improvements to the send/receive algorithms

Changes:

- 0.22** • Stage Map feature fixed
- 0.21** • bug fixed in the GUI code
- 0.20** • check for missing GD messages upon receipt of GD messages, in addition to during status-pings – this should speed up delivery time for GD messages in some cases
- the "I am here" broadcast messages are now only replied to if the sending user isn't in the receiving user's list already. Also, the reply

now simply contains the receiving user's info, as opposed to an array of the whole list. This should cut down on background network traffic.

- Discrepancies in version numbers now throw up a warning window on start-up
- 0.19**
- fixed a problem with laptop-LANDini communicating with iOS LANDini
 - "Stage Map" is still broken – will fix soon!
- 0.18**
- added tests to avoid redundant time-server changing and messaging
 - added display of the current time server
 - fixed a small bug in the list set-up process
- 0.17**
- added the user's own info to the list of active users
- 0.16**
- changed the "all" destination to mean everyone including the sender, and added "allButMe" as an alternative.
 - Added the test-button