

## *The dN $\alpha$ -Files*

*SDK Usage tips – Pads, vars, and  
dummy-pad usage*

*SDK Development for  
the Sonic Core Scope Platform*

## dNa – Digital&Analog

*DSP Plugins for Sonic Core Scope platform*

**DIGITAL with a PASSION for ANALOG**



## **Preface:**

Thank you for reading upon my Sonic Core Scope SDK development tips and tricks. I made the screenshots and examples in SDK 4 but that should not be a problem, since the latest v7 is built from this version as I understood from Sonic Core.

I had planned for a long time now to give some tips and tricks and good habits for the development on Scope. In this reader I will list talk on tricks regarding pads, vars and dummy-pad usage.

With dNa, I have come a long way in developing plugins for the Scope Platform; main goal was always trying to create essential and intuitive plugins, which combine the best of my analog and digital experiences into the dNa products. I learned the SDK by a lot of trial and error. This reader will hopefully contribute to maintaining the high standard that is called SCOPE.

One thing is for sure, I am very grateful for the users supporting dNa and Sonic Core in keeping Scope alive.

A special thanks goes out to the PlanetZ, Hitfoundry, OSS and FB friends and supporters with whom I've always had very nice personal contact with. Sorry i won't name you all, since I couldn't forgive myself if I forgot one of you, you know who you are! But one person in particular I have to name: Holger for making and keeping it all possible after all these years, and for being a good friend.

Sincerely,

Ray de Jager  
dNa – Digital&Analog

**Reader:** SDK Usage tips – Pads, vars, and dummy-pad usage / v1.0

## Pads:

In either PadLists, you can see pads. Pads are most of the time connections to vars. Pads you can create, export, delete, rename. The connection lines and antenna's you see on your window are made to pads.

Select you device, and for example with an effect you see an InL, InR, OutL and OutR pad. Ins on the left and outs on the right.

Select a pad in the PadList:

- Pressing U, D, L or R changes it representation in your circuit view.
- Pressing "I" or "O" you can change its color
- Pressing "E" will export a pad, make a new connection to the upper level of your device.
- Pressing "H" hides the pad
- Through right clicking on a pad in the padlist you can do all kinds of stuff.

## Vars:

In either PadLists you can see besides the pads, also the vars they are connected to. Vars is short for variables. These exist as several types. Most you can create, delete and rename.

Interesting for us is for example creating an integer, float, string or array var.

## Dummy Pads:

It is a good, maybe the best habit to have when using the SDK. Use dummy pads. A dummy pad is a pad which you use purely as connections between layers and modules. I use it for all kinds of stuff. You can easily save stuff in there with presets, also strings of text, or arrays; collections of values.

I use two types, the sync, or better called the audio pads, and the controller pads.

You can find both in your circuits. "Audio Pad" and "Controller Pad". Drag them to your circuit view. Both have just one connection. We're going to make some dummies.

## Audio Dummy Pad

- Make a copy of the "Audio Pad", so you have two modules of "Audio Pad", and fold them both into one module.
- Enter the module, there are two "Audio Pad" modules, select one, and name the connection pad "In" or "InL" in your PadList. Name the other one on the other module "Out" or "OutL"
- Connect them to each other, and "export" the connection.
- Make two more "Audio Pads" and repeat procedure, but name them "InR" and "OutR" (Fig 2)

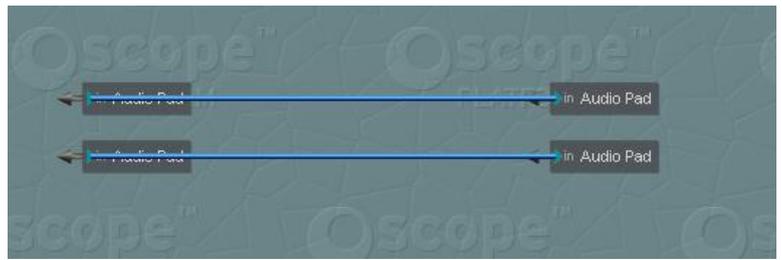
Now you should have a stereo module which just sends through audio. You can use this between parts of a circuit. Give it a logical and short name. I named mine "Audio Dummy Pad" (Fig 1)

Be carefull when you use other names for the pads, and you export them to you main device for use as just In and out connectors (as in the basic example modules "Empty Effect" and "Insert Effect fx").

Fig 1



Fig 2



For a stereo effect which can be used in an insert slot of another device, it is essential the main IO connectors are named, and take attention to Upper/ Lowercase: InL, InR, OutL, OutR.

Of course you can make all sorts of connectors, and since they use NO DSP at all, it is just a graphical representation, you can use them as much as you like, and name them as you like. I carry along for example an LFO signal through a device, or between stages in a circuit, connections to another level...you can easily cut and paste this dummy to different layers in your device, like say it was a portable patchbay.

### Controller Dummy Pad:

Select and take a closer look to the default Controller Pad module! Create a new pad, by right clicking in the PadList.

- Create for example an Integer Pad. Scope will create an integer pad and corresponding var in your module. (Fig 3)
- Give it a good name, a name of a value or knob or function it will represent. "Gain" "Level" "Treshold" "Bypass EQ" anything. In my example I have named them P1 to P6 (
- Copy the Controller Pad which you have just edited, and fold them both into one module. This will be your dummy pad module.
- Enter your dummy module, and connect them two controller pads, and export them both. Do this for any number of pads or functions your device will have(Fig 5). It will cost more time, but you will see in time why this is a good practice. I find it handy if I have a row of pads on top, and on bottom(Fig 4). I connect GUI parts, knobs and buttons to the upper ones, and the underlying circuit to the bottom row.

Fig 3

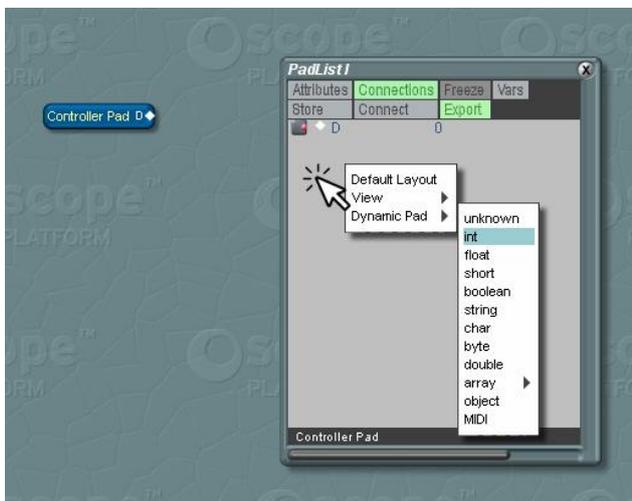
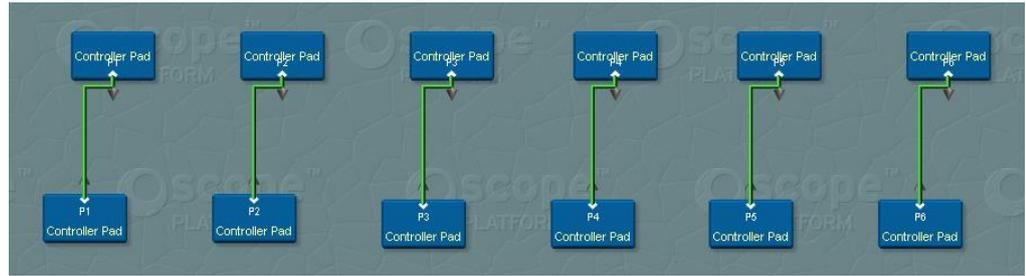


Fig 4



Fig 5



From this central connector you can also easily make parameters for your preset-capable device. Because you have named the dummies all the way to the vars, these names will show in the parameter window. This will make life a lot easier eventually. This you can also easily cut and paste to any level in your device. So eventually it is very easy keeping things centralized and thus manageable.

### **Making extra dummies for future use:**

I have found it a good practice to also include dummies, with values already stored, and store those already along with presets. I have dummies for value "0", "1" and the max value of "2147483647". This way I can expand possibilities and functions, make them switchable, and with these controllers disable these functions for older presets. This maintains backwards compatibility.

### **Making strings, array's or floats:**

In Fig 3 you can see you can make all sorts of vars and pads, and along with my other readers, you can use the dummy pads to store all kinds of data in presets: strings for names, arrays for a collection of values etc etc.