

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

*SDK Usage tips – Make your  
device preset capable*

*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 talk a bit on how to make your device preset capable.

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 – Make your device preset capable / v1.0

## First: The Main Device ParameterContext:

Fig 1



Fig 1 shows there are two basic modules that are already parameter capable. On the left you can see them, called “Empty Insert fx” and “Empty Effect”.

You can see by selecting one of them and right-clicking in the Module Attributes window, there is an option called “Delete ParameterContext”. This means, that internally you can add parameters to your device. You can use these basic startups perfectly to make your own preset capable effects.

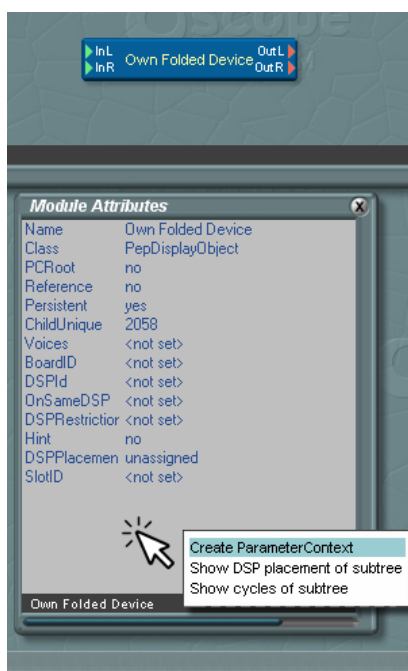
Now...if you would built your device from scratch and would “fold” your circuits into your main device, the parametercontext is not enabled by default. See fig 2. In this case you should create it.

If you ever have problems messing up the parameters, the indicator to this is that when you create a parameter, it should say where. If this name is not your name device, you should enable this flag. In Fig 2, after creating the parametercontext on “Own Folded Device” the

create parameter message would say “Create parameter in Own Folded Device”. You will see this in a later stadium of the procedure.

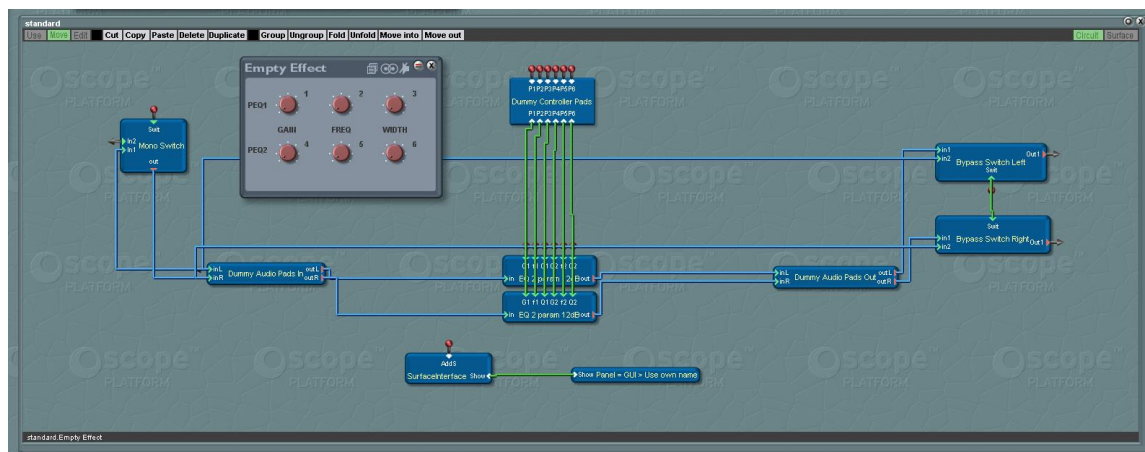
This function is also handy for creating parameter(and thus preset capability) for certain sub functions of a created device. Through cross drag and drop between the main device and sub circuits you can create advanced preset recalling(partially recalling for example inserts).

Fig 2



## Second: Creating Parameters to your device:

Fig 3

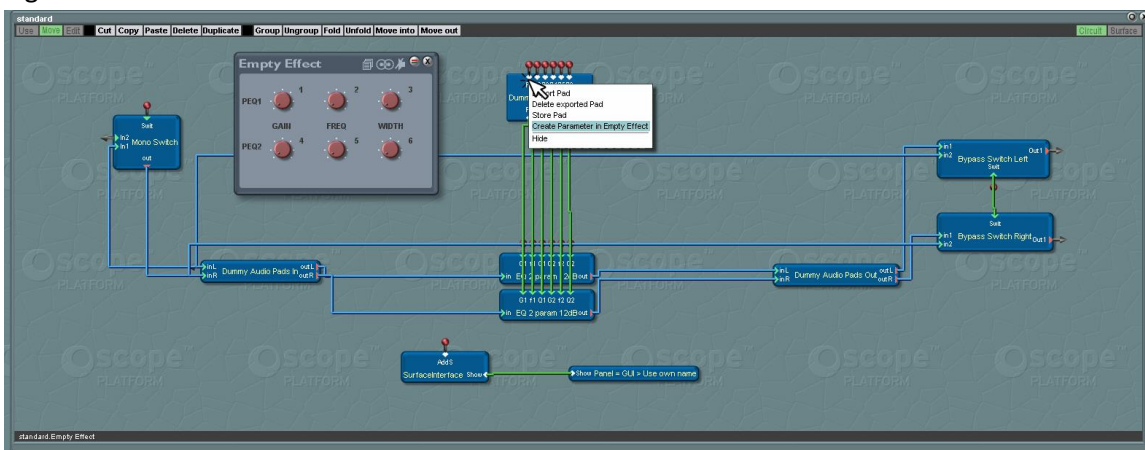


Now, as an example(Fig 3) I have created a 2 band stereo EQ, and used the Empty Effect basic module (see the names at the bottom left (standard.Empty Effect) as a starting point. Some of the internals I renamed to make the circuit easier to understand. The two EQ modules handling the left and right are interconnected, and the left channel is connected to a so called “Dummy controller Pads” module I have create for good overview, and keeping things centralized. **This is one of the best habits to have:** you connect your GUI knobs and buttons to these, and you make parameters(the stuff recalled with presets) from these. This ensures you can easily change internal circuits, without losing the knob assignments to the GUI, and keeping your already assigned parameters. The last will also insure that upon changing the circuit you can keep the device backwards compatible, since de parameter ID’s (uniqueID in Fig 5) are still there, since they were made from the dummy connector pads. In another reader I will talk more specifically on pads, vars, strings etc and dummy pads.

The dummy connectors with on top the red antenna’s is where the knobs 1 – 6 are connected. On the dummy I called them P1 to P6. (the red antenna’s indicate there is an connection from another level to this pad).

**Step 1:** To make a parameter which can be recalled through presets, you rightclick on a pad from the dummy connector and select: Create Parameter in Empty Effect (Fig 4)

Fig 4



**Step 2:** Open your Parameter window from the Scope menu. (Fig 5) Here you can manage your parameters. Before I made the new parameters, I made sure all was cleared here, to make it as easy and manageable as possible. So if you use the Empty Effect module as a basic, you will see extra parameters here. If you use the correctly named and created dummy pads, the naming here is kept easy to understand. By default a controller dummy pad will be called just “D” here. More on this in a separate reader.

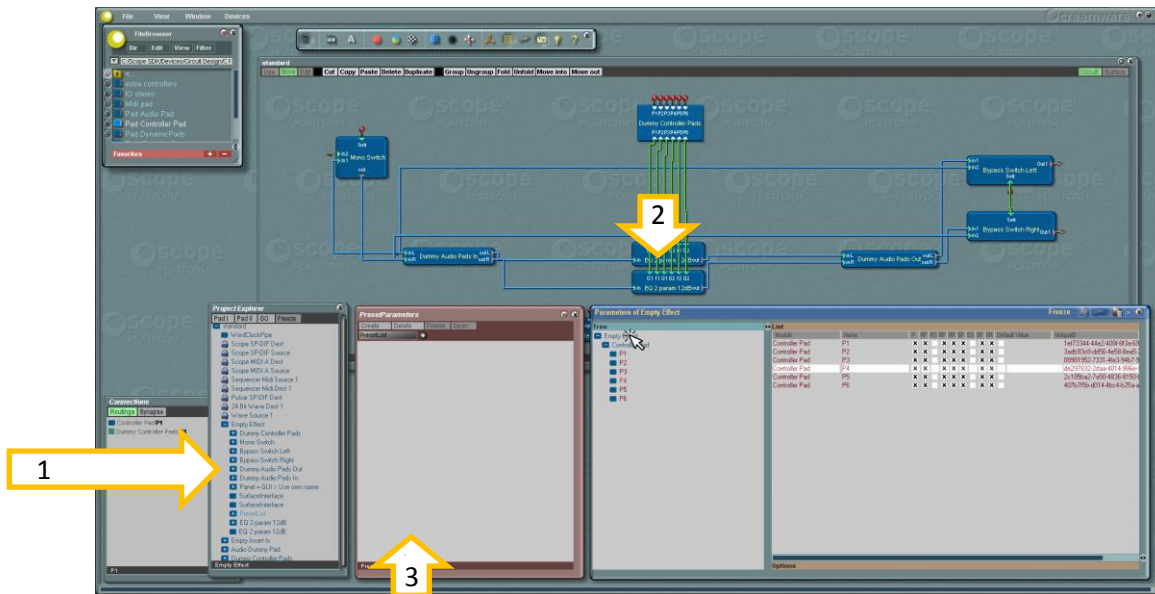
**Step 3:** In the left column create a default tree. (Fig 5) In Fig 6 you can see the tree folder created through this.

Fig 5



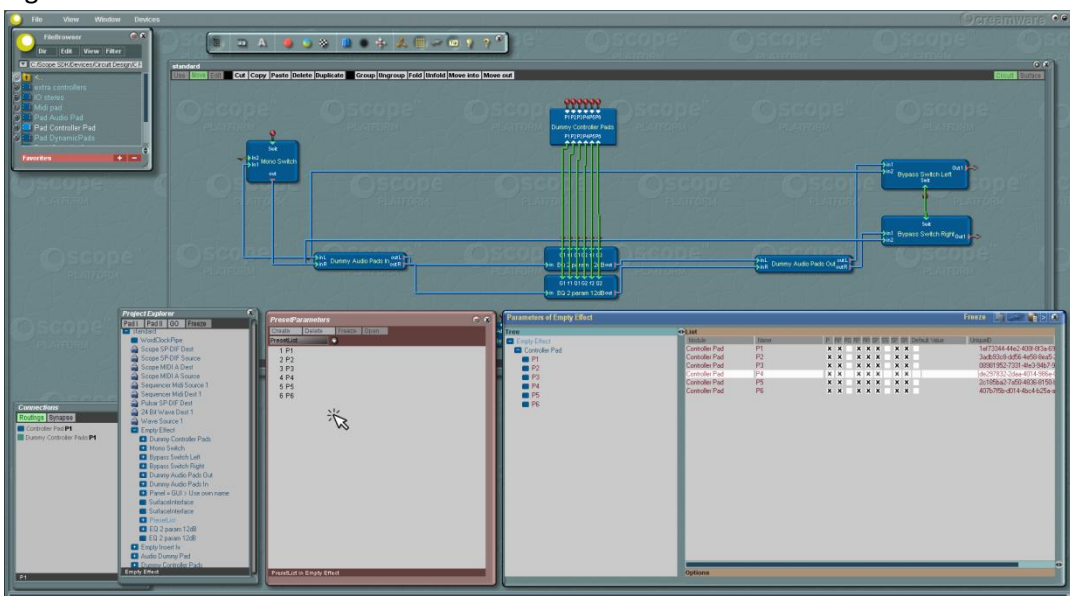
**Step 4:** Open the PresetParameter window, here it is shown red(Fig 6). Select your main device (Empty Effect in this example) so both the Parameter Window and the PresetParameter window show the module they should. If you created your device from scratch in a folded way, it could be you should create a presetlist first. You can see in the ProjectExplorer that there already is a PresetList and a corresponding SurfaceInterface(1). If this is the case you don't need to create a presetlist since it already exists. But in case of errors, deleting and starting from scratch can save a lot of time.

Fig 6



**Step 5:** Freeze both windows without clicking on any other module (you can see the parameter window shows “Empty Effect” in the upper windowbar(2), and the presetlist window also shows “Empty Effect”, but in the bottom bar(3). Drag the folder from the tree(see mousecursor Fig 6) and drop it on the preset window.(Fig 7)

Fig 7



**Step 6:** Now resave your device as a backup, as you should have also done at the beginning of this procedure. A good habit I have is: I make a circuit, and save it as “1\_give it a name.mdl”. After I have worked on it I will save the next development stage as “2\_give it a name.mdl”. The day after that... “3\_give it a name.mdl”. If anything goes wrong, I can always go back a step. There is NO UNDO in scope, nor will there ever be. Take great care in saving, especially when you protect it, and save it over your latest version.

Back to the project, you can either reload your own device, to make sure the newly assigned stuff is loaded correctly by Scope. Or....copy it, and paste your copy back into the project, delete the other one. Does the same ;-)

## Now you have a preset capable device.

On your built-from-scratch-folded-device, connect your preset button to the “AddShow” pad of Surface Interface. Open your familiar presetlist editor, make a bank, and start making presets. The example mdl is the device used to make this reader. (Fig 8-9)

Fig 8-9

