DREAM DSP DESIGNER

Software Guide - Version 1.3

Table of Contents

Overview

DREAM DSP Designer software is an intuitive graphical development tool which can be used to configure the DSPs on SAM5000 ICs for different audio processing tasks, based on audio processes and effects provided by the “libFX5000” library, see below. This document will guide you to build your project from the design to the parameter control.

Dream offers a fast library with ready to use audio processes and functions for SAM5000 series ICs, the “libFX5000” library. This library includes modules like “AudioTrack” (high quality stereo pitch shifter and other processes), “FBCancel” (Audio feedback suppressor), “LiveMic” (Reverb/Echo and other effects), “LiveMic-DualEcho” (stereo dual Echo and other processes), “MixPA” (many different kind of audio processes), “MixPA-XT” (like “MixPA”, but additionally including the RMS level detector), “MixSPDIFOUT” (same as “MixPA” module, but additionally including the SPDIF-OUT process), “MultiFX” (different kind of effects processes), “SpringReverb” (including a simulation of vintage reverberation effect used in guitar amps) and “VReverb” modules are available. A complete description of the library is given in the DreamDSPDesignerHelp HTML document. It can be accessed on the “Bus editor” or “DSP Editor” page by a right click on a DSP or process symbol respectively.

As shown in figure 1, the first step is to design your project using DREAM DSP Designer (see chapters 1-3 in this documentation). When your DSP design is ready you can generate the firmware source files to run your DSP design on a target board, using SamVS-C (see chapter 4). Finally, the process parameters can be controlled from DSP Designer or from your hardware.

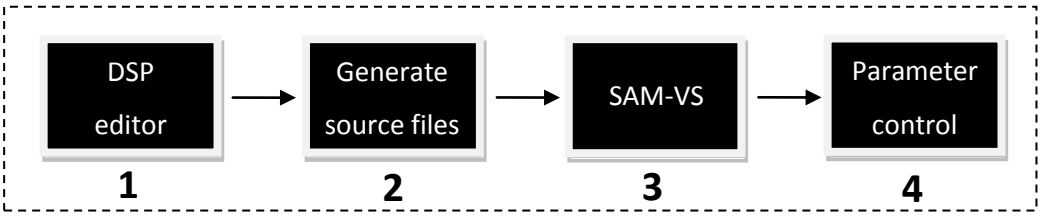


Figure 1 DSP designer workflow

The DREAM DSP Designer software can run on Windows computers (Win10, 8, 7).

In the DreamDSPDesigner software .zip file from download you’ll find 4 files, please copy all into same directory on your local hard drive.

You’ll find also some other documents at the download place (a description of MIDI messages used by Dream DSP Designer for parameter control, changes notes etc.).

Please regularly check the download places for Dream DSP Designer and SamVS-C software updates: www.dream.fr/downloads.html

Here is also the link to an online tutorial: www.youtube.com/watch?v=FGBkQAuQmsI

(!) Note: This video is based on an older version (V1.020beta), especially the step to build a SamVS project for running on a target board is much simplified now (see chapter 4 for this).

1. Project settings

As shown in Figure 2, the first screen “Project Settings” is used to set up your project. From there, the target chip, sampling rate, memory size, MIDI in/out for target board connection and project directories can be set.

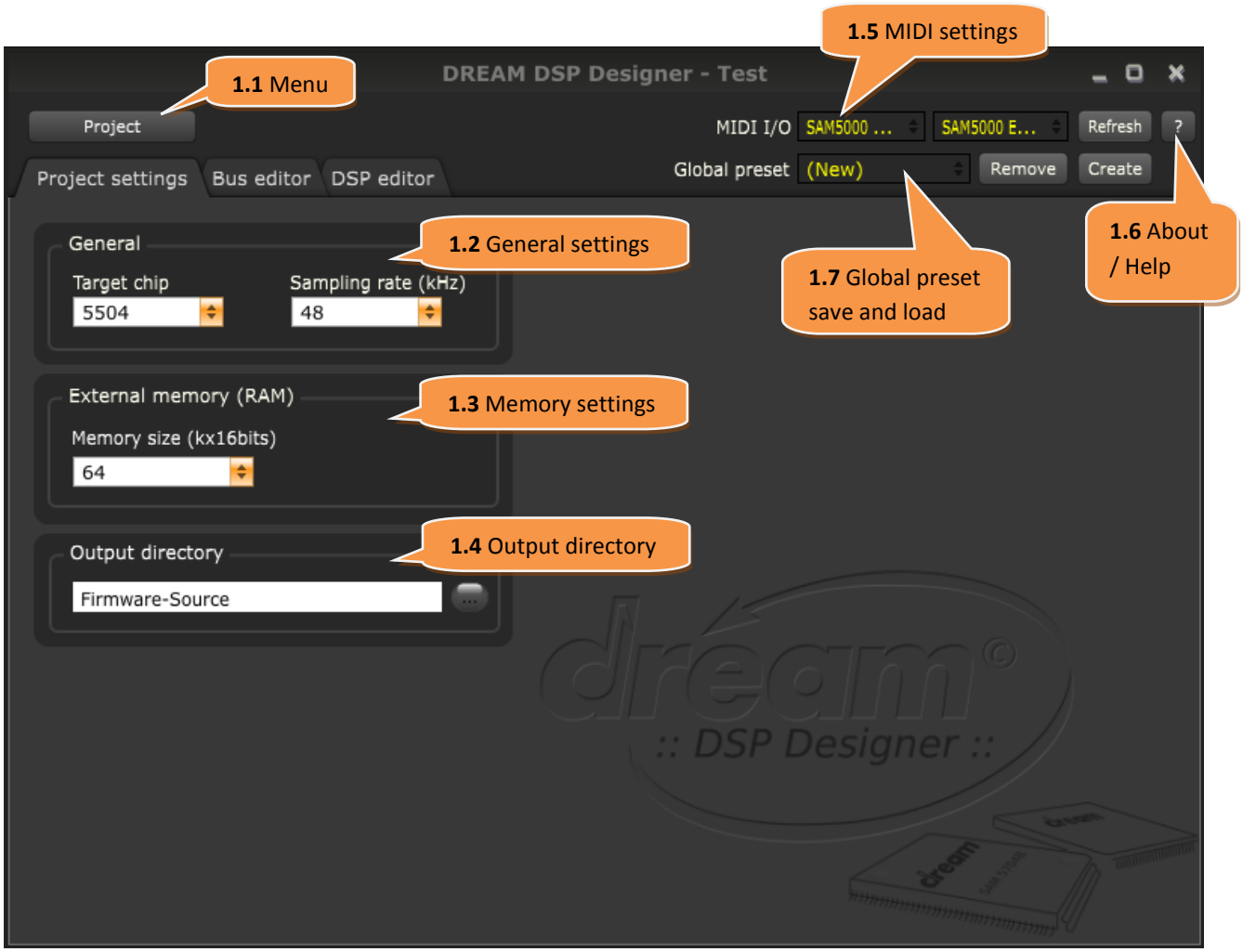


Figure 2 Project settings screen

1.1 Menu

The menu items are given in Figure 3.

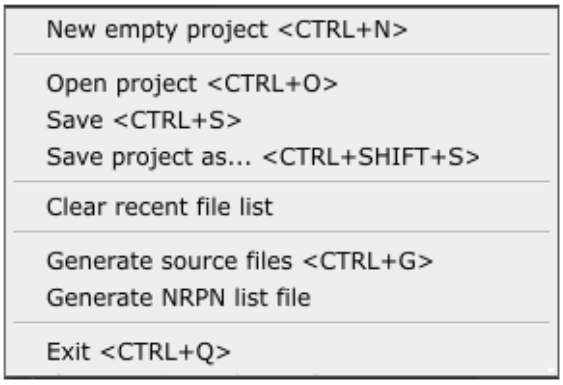


Figure 3 Project menu

- **New empty project**: you can create a new project
- **Open project**: you can open an existing project from an existing .DDP format file
- **Save**: you can save your project to an existing .DDP file. The top screen bar will include the project name.
- **Save project as...**: you can save your project to a .DDP file in a selected directory
- **Open recent file list:** you can open recently opened projects from the list
- **Clear recent file list**: to clear the recently opened files
- **Generate source files**: generates the source files from your design
- **Generate NRPN list file**: generates the MIDI NRPN command list in a separate .csv. It also contains the MIDI parameters and MIDI->physical / physical-> MIDI equations
- **Exit**: to close the program. If you have changed your design or settings a dialog box will prompt you to save your project.

1.2 General

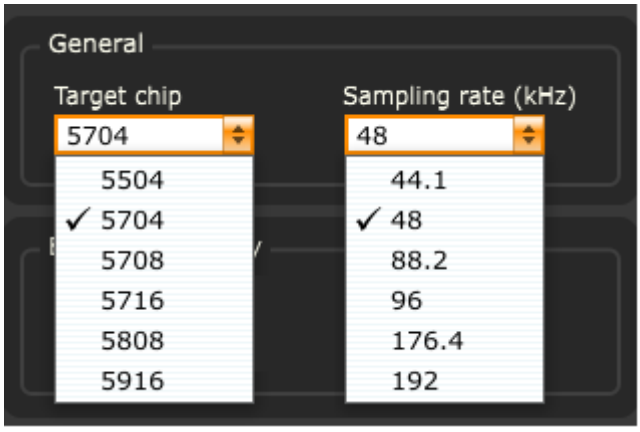


Figure 4 General settings: target chip and sampling rate selections

- **Target chip**: you can select your chip depending on how many DSP you want to use. The target chip selection is also used to prepare the SamVS project when generating the source files.

- **Sampling rate**: you can select your sampling rate from 44.1 kHz to 192 kHz, used for filter, delay time calculation and DSP cycle resource verification.

Please be aware that for Sampling rates >48KHz only MixPA, MixPaXT and MixSPDIFOUT modules can be used.

1.3 Memory settings

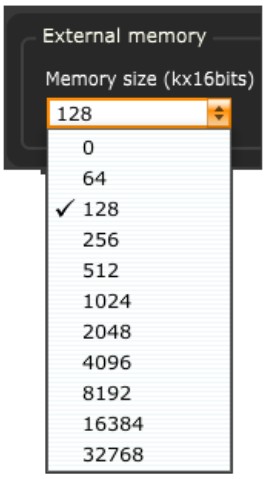


Figure 5 Memory settings

Memory size: here you can select the external memory size you want from 0 to 32768 k*16bits. This selection refers to static or dynamic processes requiring external memory space.

1.4 Output directory

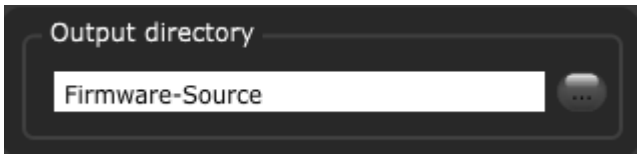


Figure 6 Output directory

Here choose your output directory where the generated source files should be stored. Press the button to open the file selector dialog.

1.5 MIDI settings

Figure 7 MIDI settings

- **MIDI I/O**: select your MIDI input and output port to the connected board (typically “SAM5000 Eval Board”)
- **Refresh**: use this button to refresh the MIDI I/O list, reconnect the MIDI I/O to the connected board, and send all current controls value to the board. This function is very useful after running an updated firmware on the connected board (see also chapter 4.2)

1.6 About / Help

Figure 8 About window

The [?] button displays the DSP Designer software version. It also provides a link to the software guide and the download page for DREAM development software.

1.7 Global Presets

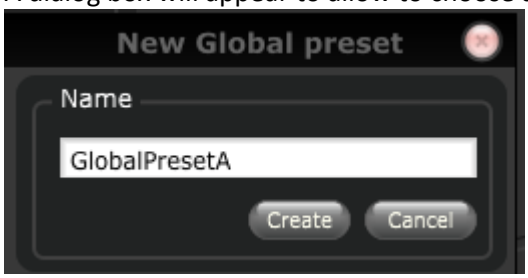
DREAM DSP Designer V1.3 and up allows saving and recall of global presets. Such global presets include all process parameters of all DSPs. Several global presets can be stored for A/B compare, or for use in the application firmware to load different overall settings.

Upon saving a Global preset an XML file (*projectname*_global.presets) is generated which includes all the process panel settings of the whole project. From this XML file the GlobalPreset.c/.h files are generated when using “Generate source files” command (see chapter 4.5).

The XML file *projectname*_global.presets stores the process control panel parameters in an optimized (memory space saving) way: for the same kind of panel (for example all Biquad types, MixN types, GainN types ...) it builds a dictionary of settings.

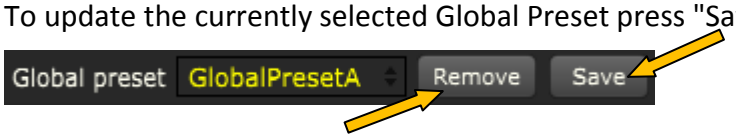
To save a Global Preset select (New) from the Global preset list-box, then press "Create" button:

A dialog box will appear to allow to choose a name for the Global preset:



You can select a Global preset by clicking on the Global preset list-box:

To update the currently selected Global Preset press "Save" button:



Use "Remove" button to remove the selected Global preset.

(!) Note: Global presets should only be generated on a final DSP design. In case you need to do modifications on the DSP design (add or remove processes etc.) you must store the project under another project name.

The Dream DSP Designer "Refresh" button is using another database inside the .DDP file itself for reloading all process parameters, it works independent for the Global Presets management.

The Global Presets management is also not linked to the process presets database (*projectname*.presets) exported in presets.c (see chapter 3.6 and 4.4), both can be used independently.

2. Bus editor

In the “bus editor” tab, the connections between the DSP, the audio input/output and internal buses can be defined. Figure 9 shows the possible configurations for a 4 DSP design. Here a 5704 chip is used with 2 loaded DSP with Multi-FX modules and 2 empty DSP. When creating a new project, a DSP has no input/output pins until a process is dropped into it in the DSP editor tab. A DSP can be added or deleted using the mouse right click. Finally the scroll bars can be used to adjust the window, especially when the density of components increases.

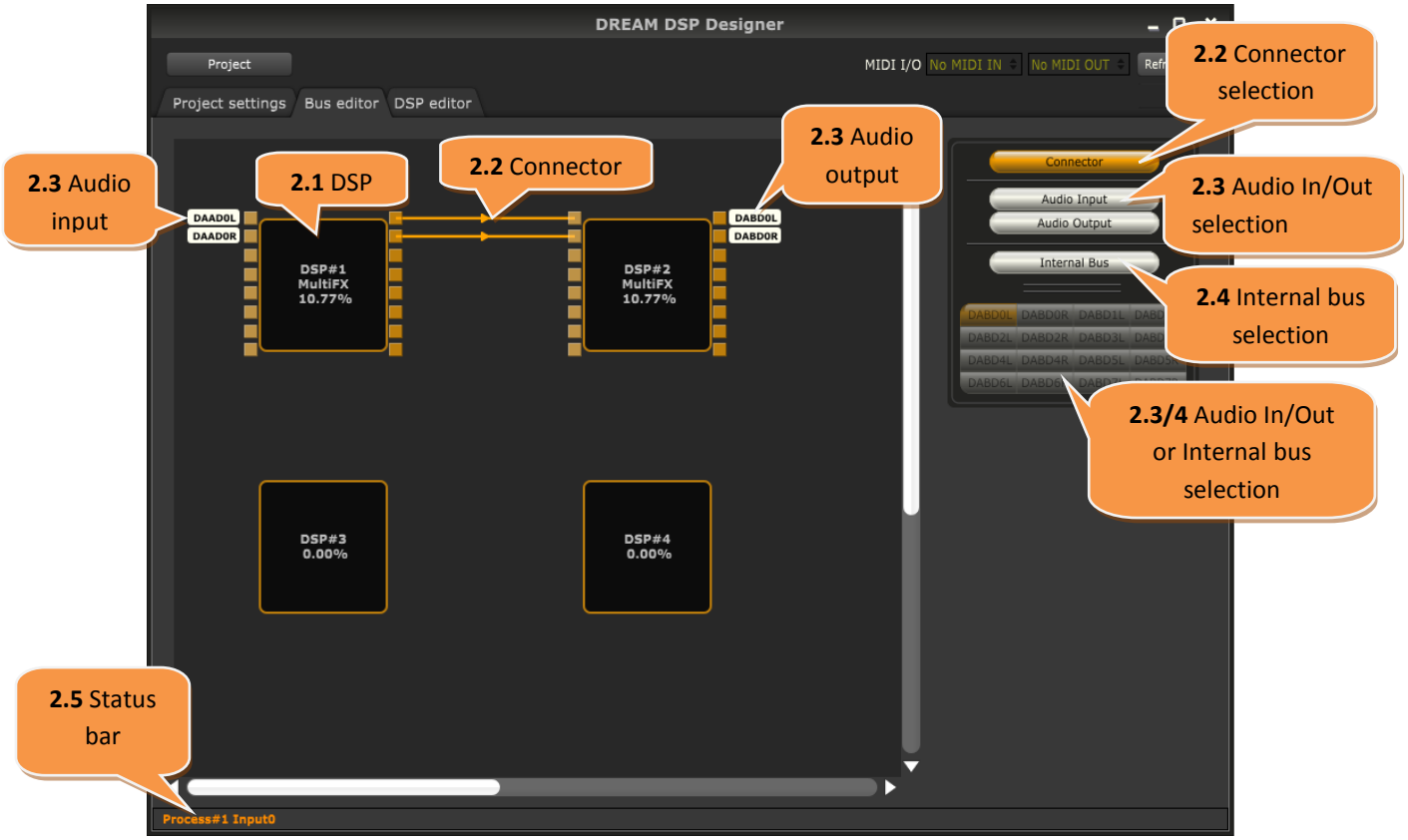
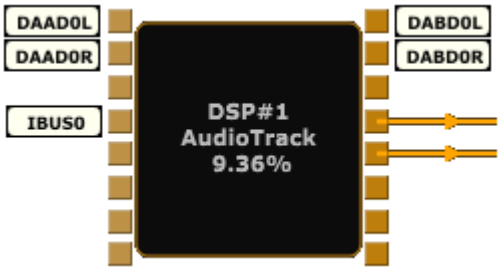


Figure 9 Bus editor tab

2.1 DSP edition



The DSP is represented by its inputs, outputs, bus labels, total used cycles and current used module. The DSP number can be set from 4 to 16 depending on the target chip selected on the first tab. For instance, the 5708 chip will contain 8 DSP maximum.

Figure 10 DSP with labels and connections

2.2 Connector

In the right menu, the connector button must be activated (orange color) to draw a connection between DSP input and output. You cannot connect an input to an input, or an output to an output. The arrow indicates the destination. To delete a connector, right click on it and select “Delete Connector”

2.3 Audio Input/Output

Figure 11 Audio input/output label selection

To connect a DSP input/output to one of the chip physical input/output, click the Audio Input/Output button to display the chip input/output list, select the desired input/output by clicking the corresponding button, then assign it to a DSP input/output by clicking the DSP pin it should be connected to. You can only connect an “Audio Input” to a DSP input pin and “Audio Output” to a DSP output pin. Finally, to disconnect a pin, right click on it and select “Delete BusPin”.

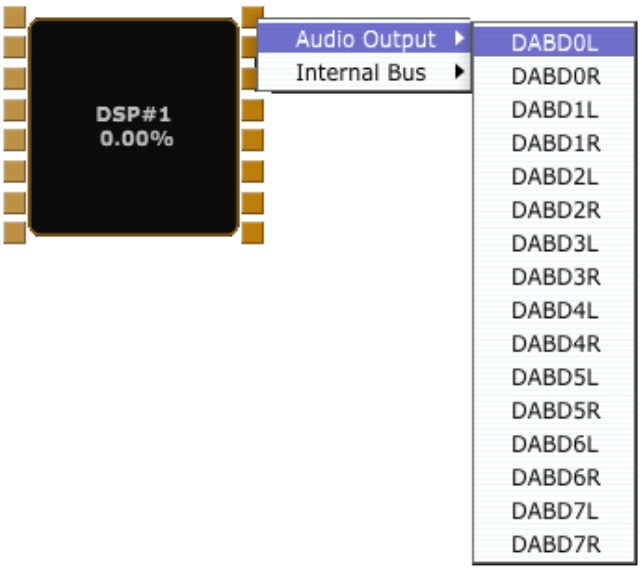


Figure 12 Audio output labels

Another way to connect an audio input/output to a DSP is to right click on the considered DSP pin and choose the desired input/output as depicted in Figure 12.

2.4 Internal Bus

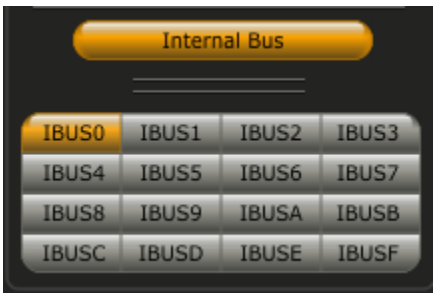


Figure 13 Internal bus label

To connect an internal "remix" bus to one of the chip physical input/output, click the Internal bus button to display the list, select the desired input/output by clicking the corresponding button, then assign it to a DSP input/output by clicking the DSP pin it must be connected to. In SAM5000 series, there are 16 remix busses available. All DSP outputs going into the same remix bus will be accumulated. Finally, to delete a bus, right click on it and select "Delete BusPin".

2.5 Status bar

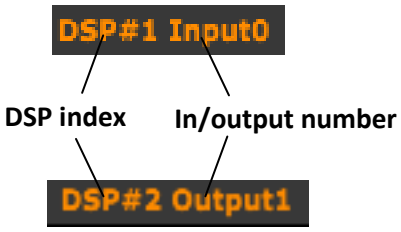


Figure 14 DSP/In and out index

The input/output names are shown in the status bar when moving the mouse pointer on top of them. The index depends on the input/output DSP index.

3. DSP editor

The “DSP editor” page can be accessed by clicking the corresponding tab or by double clicking on the desired DSP in the “Bus editor” tab. It allows selecting the module, configuring and connecting the associated static and dynamic processes. The DSP can be selected by using the tabs or by double clicking on the DSP in the Bus editor tab. Similarly to the bus editor tab, the process and in/output numbers can be shown in the status bar when moving the mouse over it. Figure 15 shows an example with a crossover filter and the following described sections.

Figure 15 DSP editor tab

3.1 Modules and processes

The first step is to select a module from the module list, then to drag and drop the desired processes. The component contains the process ID, the cycle number and the process index. It can have N input/output selected by an increment/decrement button and a RAM type selector.

As depicted in Figure 16, there are plenty of “dynamic” and “static” processes depending on the selected module. “Dynamic” processes can be used with several instances on a DSP edit page, whereas “static” processes can only be used once per module. Static processes can be distinguished by a “(s)” symbol at the beginning of their name in the process list.

Hint: When no static process is needed in your design, it is recommended to use the MixPA (or MixPaXT) module.

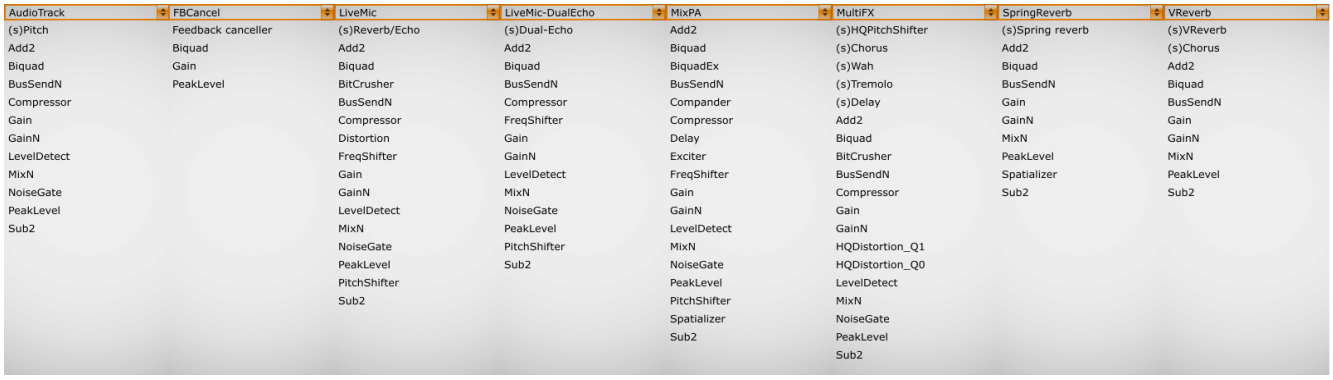


Figure 16 Available modules and associated static and dynamic processes

The MixPaXT and MixSPDIFOUT modules are not shown in the list above, their process list is basically same as for the MixPA module (see the Help document for all modules process details).

In this document only basic processes are described. The full description of each process is available by a right click on the process components.

Operators: Add2 and Sub2

Figure 17 Add2 and Sub2 processes

The add2 and subb2 processes are respectively used to make additions and substractions with 2 inputs and 1 output. Only these processes do not contain controls.

Biquad / BiquadEx: you can change your Biquad filter band number

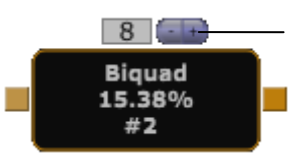


Figure 18 Biquad process

The Biquad process allows building filters from a selection of 40 filter types. The number of bands is controlled by the inc/dec buttons on top of the process symbol, with a maximum of 31 bands. The BiquadEx process offers 2 more types (2nd order Hi/Lo shelf filter).

(!) Note: Crossover filter types with a slope of 18dB or more consume several bands. For instance, for a "Butterworth Lowpass 36dB/Oct *3-band*", the 2 next filter band parameters must not be used.

The filter parameters: frequency, gain can be adjusted using the rotary sliders and the red circles as well on the graphic. The quality factor Q can be set by the horizontal slider. Furthermore, a slider value can be more precisely changed by pressing the CTRL keyboard control. The next figure shows the Biquad process control window.

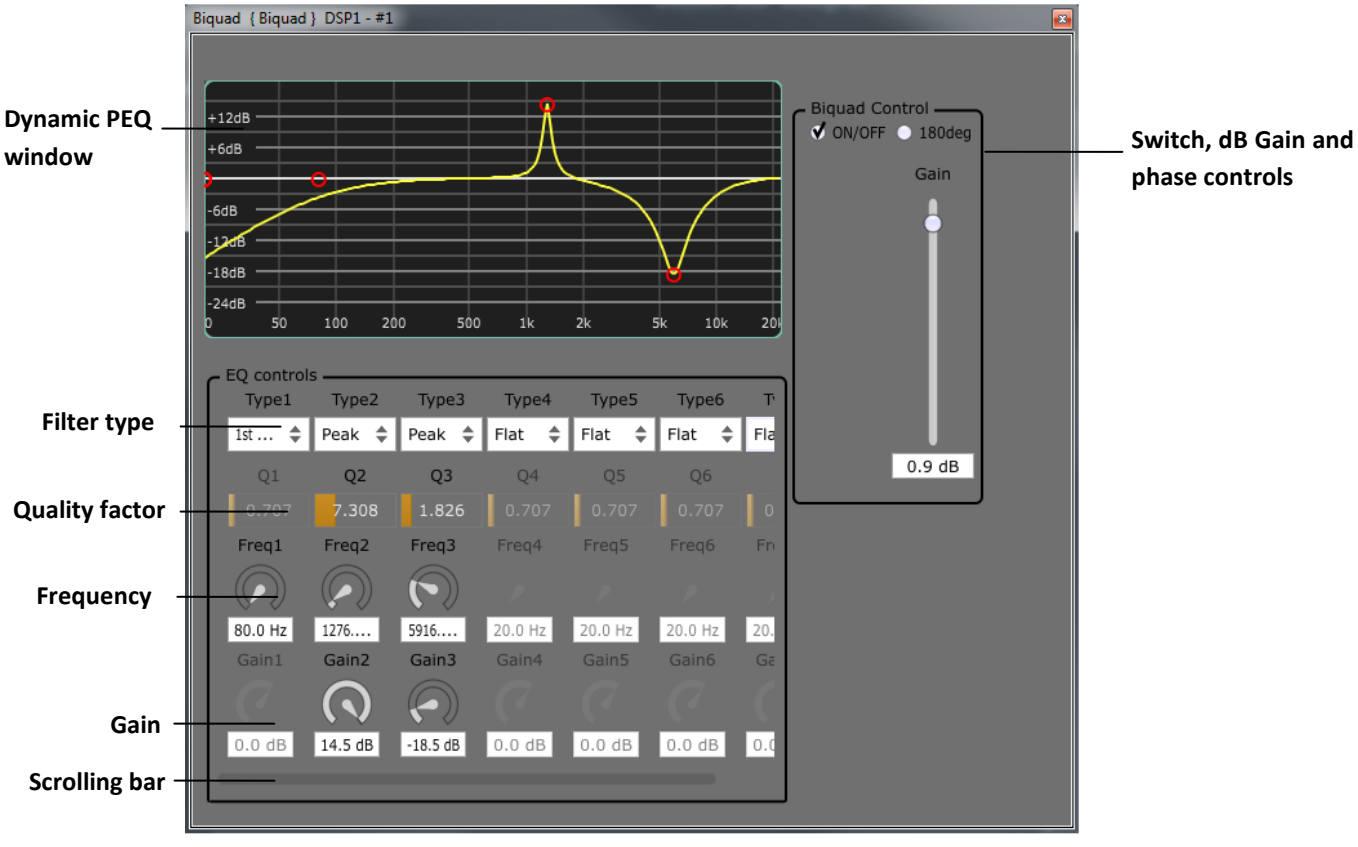


Figure 19 Biquad process parameters control

Compressor

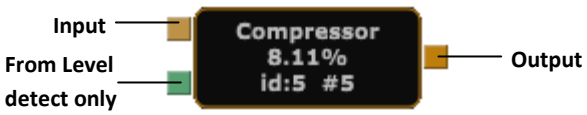


Figure 20 Compressor process

The compressor process allows to modify the threshold, and ratio from sliders and red dots on the graphic. The boost and phase can also be controlled. Additionally, a VU meter allows to see the gain reduction level in dB. This process has a green input to connect a level detector. Please check section

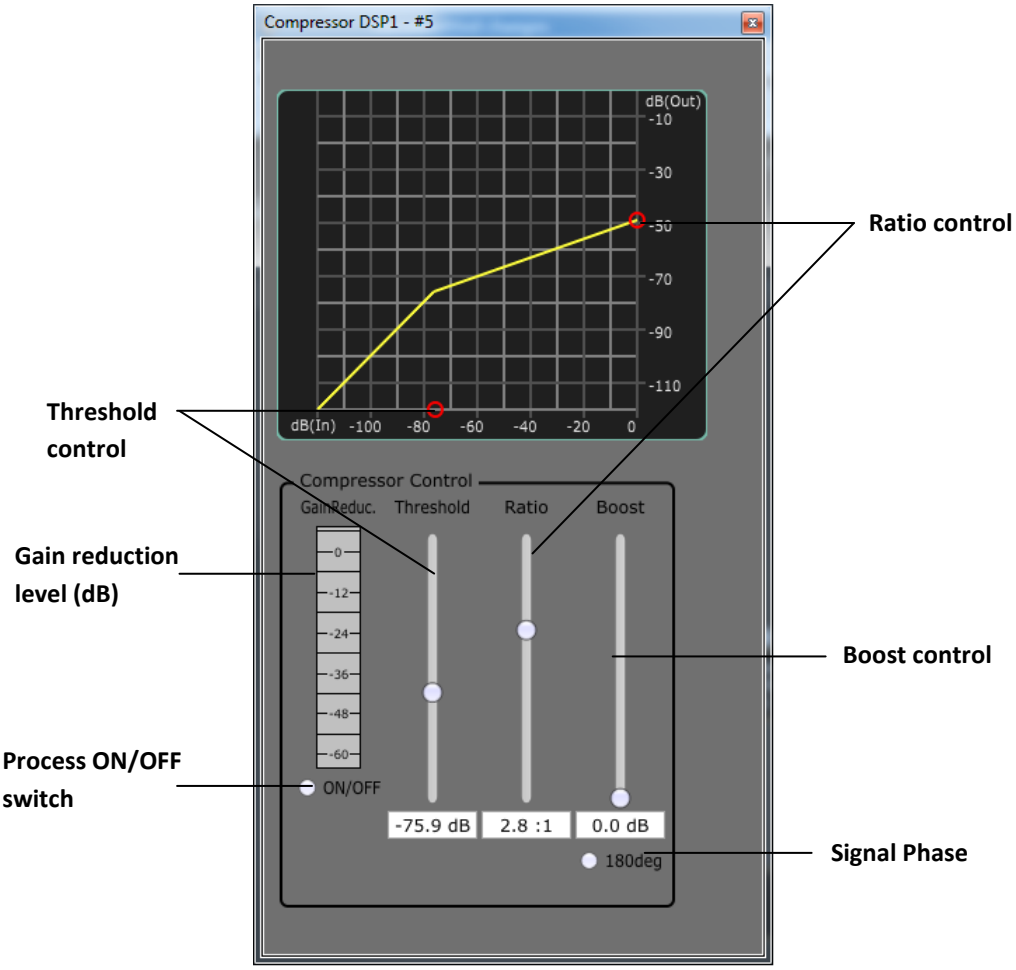


Figure 21 Compressor process parameter control

Multiple input/output processes: BusSendN, GainN and MixN

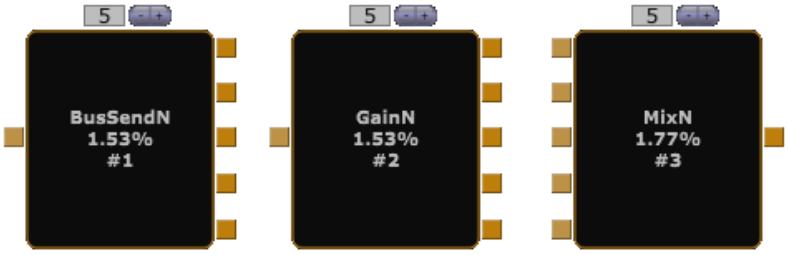


Figure 23 BuSendN, GainN and MixN processes

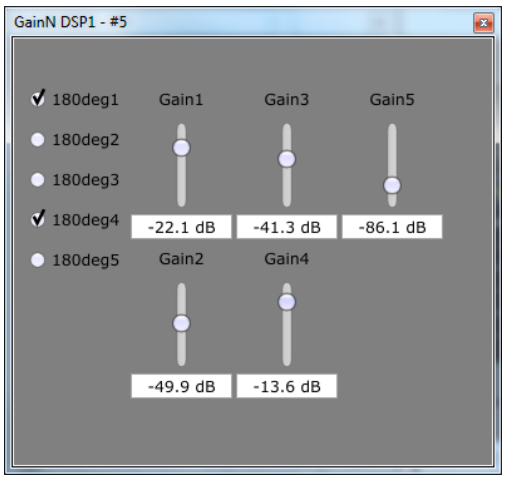


Figure 22 GainN parameter controls

The Multiple input/output processes (BusSendN, GainN and MixN) have the same gain (dB) and phase (0° and 180°) controls. BusSendN is accumulating the outputs of all processes to the same outputs, the GainN process is overwriting the audio samples on these outputs.

(!) Note: All the connected input/output indexes must be consecutive. Connectors and input/output label cannot be used on the same process.

Hint: for summing a connector with an audio input, you can use the "Add2" process instead of MixN.

Delay

This is a channel delay process. You can change the delay length in ms and type of Internal/External RAM:

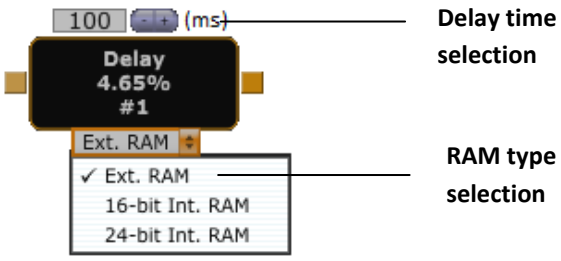


Figure 24 The Delay process

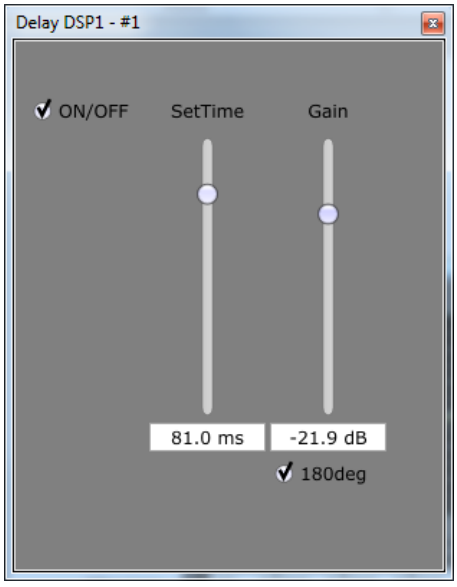


Figure 25 Delay parameter controls

3.2 DSP input/output

Figure 26 Input/Output labels selection

To connect a process input/output to as DSP input/output, click on the Input/Output button to display the process input/output list, select the desired input/output by clicking the corresponding button, then assign it to a process input/output. You can only connect a DSP input to a process input pin, and a DSP output to a process output pin. To delete it, right click on it and select “Delete BusPin”.

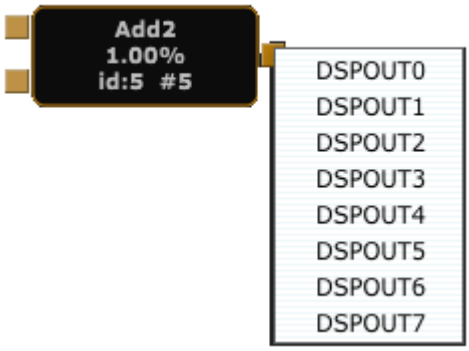


Figure 27 Output labels selection

Another way to connect input/output is to use the menu by doing a right click on the desired process input/output pin as depicted in Figure 27. When a process does not have input/output (for instance SpringReverb), please configure the first DSP input/output on the bus editor page as shown in the HTML help.

3.3 Connector

Similarly to the DSP connections, the connector button must be activated to draw a connection between DSP input and output. The arrow indicates the destination. As shown in Figure 28, there are orange connectors for audio connections (attached to orange pins) and green connectors for virtual connections (attached to level detect, compressor and wah green pins). To delete a connector, right click on it.

Figure 28 Process connectors

3.4 Used DSP cycles / RAM

As shown in Figure 29, the DSP cycles / RAM progress bars are updated on the fly when adding and setting up a process to monitor the available resources left. Internal 16/24 bit main RAM is not shown.

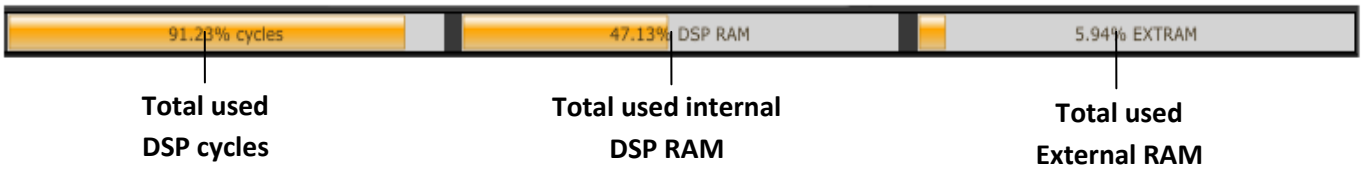


Figure 29 DSP cycles and RAM progress bars

3.5 Process options

As shown in Figure 30a, a process contour is becoming yellow when it is individually selected. As shown in Figure 30b, the right click menu on a process allows to Remove or Copy, as well as renaming it. You can use the CTRL key for multi-process selection for moving, copying or deleting them all together. After saving a project, process control settings can be saved and loaded from this menu, which is further described in the next section. Finally, as shown in Figure 30c, the right click menu besides a process allows copy or delete a selection of one or several processes with its connections. "Select all" allow to select all processes on the current DSP page, including all connectors. This menu also allows to Undo/Redo previous steps.

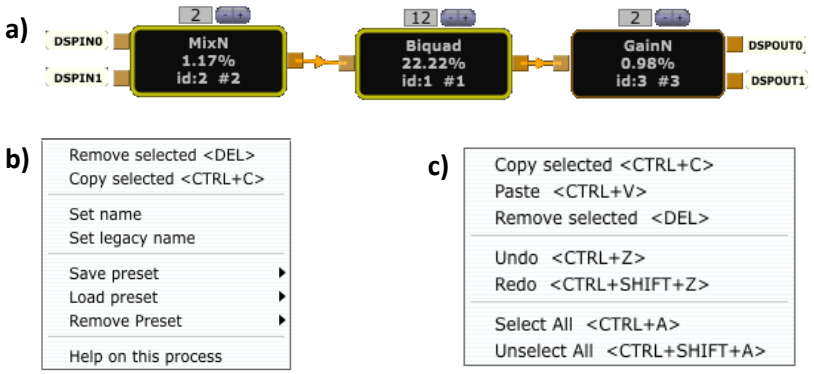
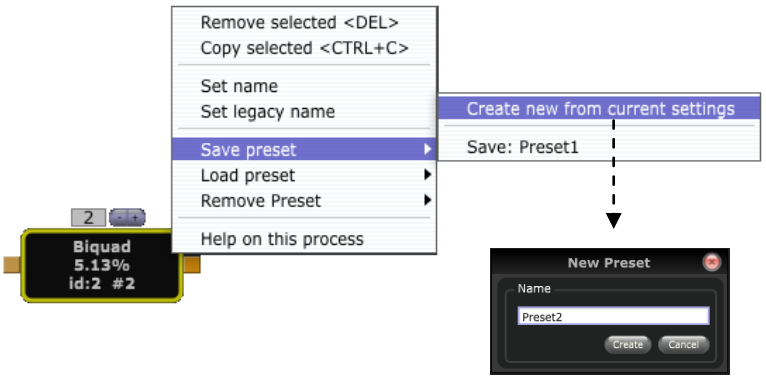


Figure 30 Process options: a) Process selection b) right click menu and c) menu besides a selected process

3.6 Preset management

This chapter describes the Preset management available for storing different settings per process. For "Global presets" see chapter 1.7.



It is possible to save process control settings once a project is saved. For this, click on "Create new from current settings". It will open a new dialog box where the desired preset name can be entered. After pressing the "Create" button, the preset will be saved into a .presets file located next to the project file with the same name as the project.

Figure 31 Preset saving procedure

Afterwards, the preset will be available in the "Load preset" item as shown in Figure 32a. When a preset is loaded and the setting changed, there is still the option to resave the preset again. Moreover, Figure 32b shows how to remove a preset from the list. There is also the possibility to delete the preset list. Moreover a preset can be loaded from a DSP to another DSP as long as the process is the same. A preset file can also be taken from a project to another as long as the .ddp and .presets files have the same name. Please refer to the section 4.4 for further details regarding preset loading in the firmware source code.

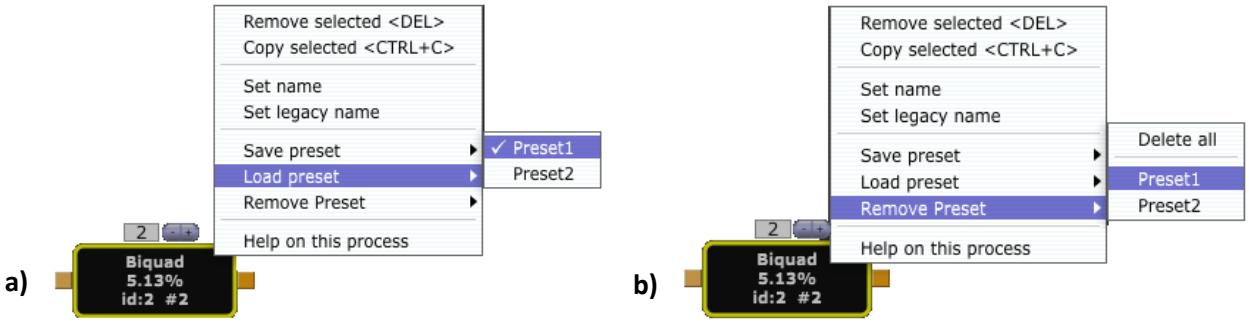


Figure 32 Process right click menu items. a) Loading and b) removing presets

3.7 HTML Help

Each (dynamic or static) process control is described in the DreamDSPDesignerHelp HTML document. It can be accessed from process contextual menu (right click). Figure 33 shows an example with the Biquad process.

(!) Note: in this help file only the Modules of libfx5000 library under control of DSP Designer are described. Please see the SamVS-C FX5000 Library.chm documentation for the complete list of module and associated functions, such as the initialization files.

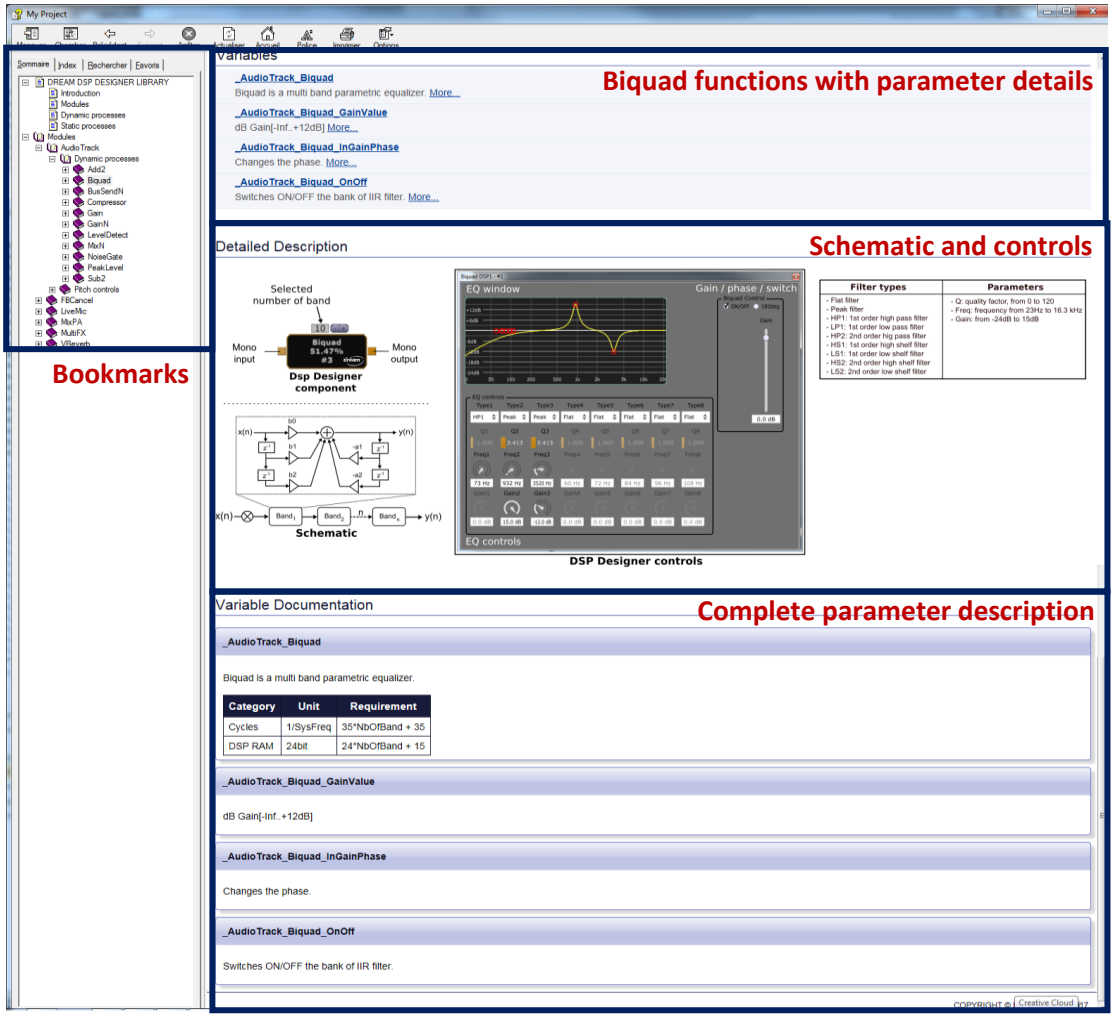


Figure 33 DreamDSPDesignerHelp HTML document

4. Source file generation and firmware running

4.1 Source file generation

The C files can be generated from DREAM DSP Designer in "Project" menu -> "Generate source file". An output LogFile window is then created as shown in Figure 34. Design errors are described here. There must be no error to load a firmware into the board. From version 1.105, the SamVS-C .dcp project file is also generated and a USB descriptor is added in main.c for easy start.

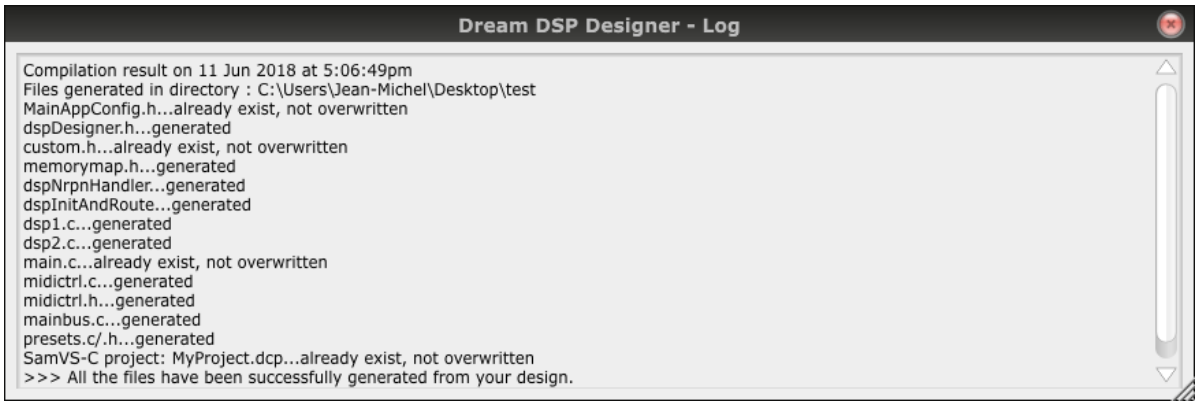


Figure 34 Output log file window

4.2 Running the firmware using SamVS-C

Here are the steps to follow to run a firmware based on generated source file on a target board, before controlling the parameters from DREAM DSP Designer:

1) Open the .dcp SamVS-C project file generated by DREAM DSP Designer
2) In "Project>Settings>General" select your chip (pre-selected from DREAM DSP Designer) and the board you are using
3) Build, load and run the firmware (e.g. by pressing [F7] and [F5] respectively) in SamVS-C
4) In case of warning message "!W CODERAM overflow..." select in "Project>Settings>Chip-Features" the next bigger CODE size under "Code/Cache RAM" selector box. Please see SamVS User Manual and ProgRef5000 for details on memory mapping.
5) When firmware is launched, make sure the "SAM5000 Eval Board" is selected in the MIDI I/O selectors in DREAM DSP Designer
6) Go to the "DSP editor" tab and open the process parameter control windows you want to use

Figure 35 shows a screenshot of SamVS-C, here using 2 DSPs in the design:

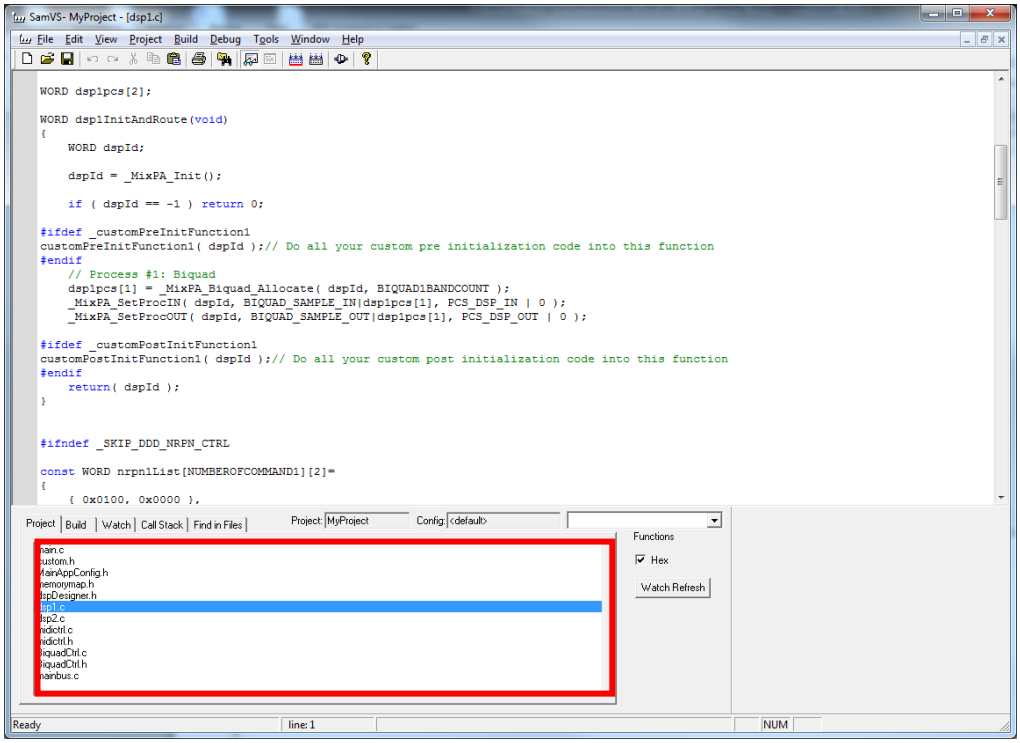


Figure 35 List of project files required in your SamVS-C project

These files will be generated only if they do not exists already at the output folder. They can be edited by the user, and will not be overwritten by DREAM DSP Designer:

- <**projectname>.dcp**: SamVS project file with pre-defined settings (Target IC etc.)
- **main.c**: firmware main entry point, used typically for initializations and base level loop
- **MainAppConfig.h**: allows to define the start addresses of internal 16/24bit MAIN RAM and external RAM
- **custom.h**: custom code (see section 4.3)

Following files should not be modified by user (under Dream DSP Designer control):

- **memorymap.h**: memory definitions for processes requiring internal/external RAM buffers
- **dspDesigner.h**: DSP related definitions
- **mainbus.c**: all DSP start-up and routings
- **dsp1.c…dsp16.c**: specific DSP module initialization and control functions
- **midictrl.c/.h**: MIDI control functions
- **BiquadCtrl.c/.h**: Biquad filter control functions
- **presets.c/.h**: presets handling (see section 4.4)
- **GlobalPresets.c/.h**: global presets handling (see section 4.5)

4.3 Adding custom code

It is possible to add custom code to the DSP generated files (dsp1.c, dsp2.c, etc.) by use of the custom.h file. In this file C #define lines and prototype functions are predefined for each DSP. They are commented out by default so that the custom code(s) are inactive by default.

There are 4 areas where custom code can be executed in the dsp#.c file:

1) Before process allocation using the `_customPreInitFunction`, allowing additional preparation
after module initialization
2) After process allocation using the `_customPostInitFunction` for parameter settings/preset
preloading
3) Before processing NRPN function call using the `_customPreNrpnFunction` for parameter range adjustments
4) After processing NRPN function call using the `_customPostNrpnFunction` to add customized
nrpn number

Please follow these steps to insert the user code within one of these areas:

1) The corresponding C '#define' line in custom.h must be uncommented to execute code in the desired area as shown in Figure 36.
2) Implement the "area" C function in separate C file as shown in …. Add this file also to SamVS-C Project by using "Add file…" or drag&drop to the project file list.

Uncommented C '#define' line

```
// This file can be modified by user and will not be overwritten by Dream DSP Designer.
#ifndef __CUSTOM_H__
#define __CUSTOM_H__

//DSP#1 custom functions, uncomment the '#define' line where you want to add custom code,
// and implement the function below the '#define' in a separate C file

//#define _customPreInitFunction1
WORD customPreInitFunction1( WORD dspId );
//#define _customPostInitFunction1
WORD customPostInitFunction1( WORD dspId );
//#define _customPreNrpnFunction1
WORD customPreNrpnFunction1( WORD dspId, WORD nrpn, WORD val8bit, WORD value, DWORD dvalue );
#define _customPostNrpnFunction1
WORD customPostNrpnFunction1( WORD dspId, WORD nrpn, WORD val8bit, WORD value, DWORD dvalue );


#endif //__CUSTOM_H__
```

Figure 36 Uncommented C '#define' line in custom.h

4.4 Preset handling

From version 1.107(beta), the generated files named presets.c and presets.h contain the presets created as shown in section 3.6. In presets.c, the `dspDesigner_LoadPreset()` function is defined and can be called from the user application C code as shown in Figure 37 to load a preset from single code line. The "presets.h" header file should be included in C source file for this. The `dspDesigner_LoadPreset()` must be called earliest after DSP initialization is done by `dspDesigner_InitAndRoute()` and `dspDesigner_InitNrpnFunction()` functions.

In the following example, a 3-Bands Biquad process is in DSP#1 (channel=0), process ID #2, is loaded with the first preset.

```
// load 3-Bands Biquad preset
    dspDesigner_LoadPreset( 0, 2, (const WORD*)&biquad3_controls,
                            (const WORD*)&biquad3_presets, 0 );
```

Figure 37 Example of Biquad preset loading

Figure 38 shows another example of loading a SpringReverb preset, with SpringReverb process located in DSP#2 (channel=1), process ID #1, second preset.

```
// load Spring-Reverb preset
    dspDesigner_LoadPreset( 1, 1, (const WORD*)&s_spring_reverb_controls,
                            (const WORD*)&s_spring_reverb_presets, 1 );
```

Figure 38 Example of SpringReverb preset loading

4.5 Global Preset handling

Additionally to the process presets described above the Dream DSP Designer V1.3 and up allows also the saving and recall of global presets (see chapter 1.7). If Global presets have been stored in the Dream DSP Designer project the GlobalPreset.c/.h files are generated when using “Generate source files” command.

The GlobalPreset.c source file provides the “dspDesigner_LoadGlobalPreset()” function for loading all parameters The generated “main.c” source file is prepared for loading the first Global preset (globalpreset[0]) at startup if _USE_GLOBAL_PRESET define is set 1.

```
#define _USE_GLOBAL_PRESET 1 // 1=include global preset handling, 0=not using global preset
```

In order to load other global presets you just need to write e.g.

```
dspDesigner_LoadGlobalPreset( globalpreset[1] );   // load second global preset
```

Appendix: List of Figures

Appendix 2: List of Shortcuts

Shortcut	Function
CTRL+N	New empty project
CTRL+O	Open project
CTRL+S	Save project
CTRL+SHIFT+S	Save project as…
CTRL+G	Generate source files
CTRL+W	Close output log window
CTRL+Q	Exit the software
CTRL+C	Copy selection
CTRL+V	Paste selection
DEL	Delete selection
CTRL+Z	Undo
CTRL+SHIFT+Z or CTRL+Y	Redo
CTRL+A	Select all
CTRL+SHIFT+A	Unselect all
CTRL + slider	Fine parameter tuning and multi process selection

www.dream.fr

DREAM S.A.S.
Z.I. 21140 Semur-en-Auxois FRANCE
info@dream.fr