Digital Audio Networking Processor

CobraNet™

DSP Conductor™ User's Guide

http://www.cirrus.com

OCT 2005
DS694UM1

Contacting Cirrus Logic Support

For all product questions and inquiries contact a Cirrus Logic Sales Representative.
To find one nearest you go to **www.cirrus.com**

Table of Contents

List of Figures

1. Introduction

This guide describes DSP Conductor for CobraNet version 1.1.0.

DSP Conductor is a graphical programming environment for configuring the Cirrus Logic CS4961xx network processor.

The base version of DSP Conductor allows you to download your configuration to the Cirrus Logic EV-2 CobraNet evaluation module or the CobraNet CO2 module from Attero Tech.

If you are a product manufacturer and wish to use DSP Conductor to develop firmware for you own product based on the CS4961xx Network Processor, you must send an e-mail to *dspconductor@cirrus.com* to obtain a registration code that enables DSP Conductor to download configurations to your product as well as create firmware images that can be used for mass production. The *Tools ⇨ Register Software* menu item will automatically create this e-mail which will contain the required Registration ID number that will uniquely identify your computer.

If you have not already read AN275, *"Getting Started with DSP Conductor"*, please read that document first. AN275 contains important information about setting up your environment that is not repeated in this document.

This guide assumes you have already:

1. Setup your development network with a PC, some CobraNet devices and an Ethernet switch.
2. Registered and installed DSP Conductor and CobraNet Discovery software.
3. Setup the audio connections to your CobraNet devices.
4. Assigned IP addresses to you CobraNet devices.
5. Updated your CobraNet devices to the latest firmware version.
6. Downloaded and controlled the example project file.

These steps are all explained in AN275, *"Getting Started with DSP Conductor"*.

2. Building a Configuration

This section describes the concepts and features of DSP Conductor in the general sequence you will need to understand them in order to design and listen to a DSP configuration. The *Reference* section at the back of this document covers details that are not covered in this section.

2.1 DSP Conductor Main Window

Below is the DSP Conductor *Main* Window. The following text describes the main window's features.

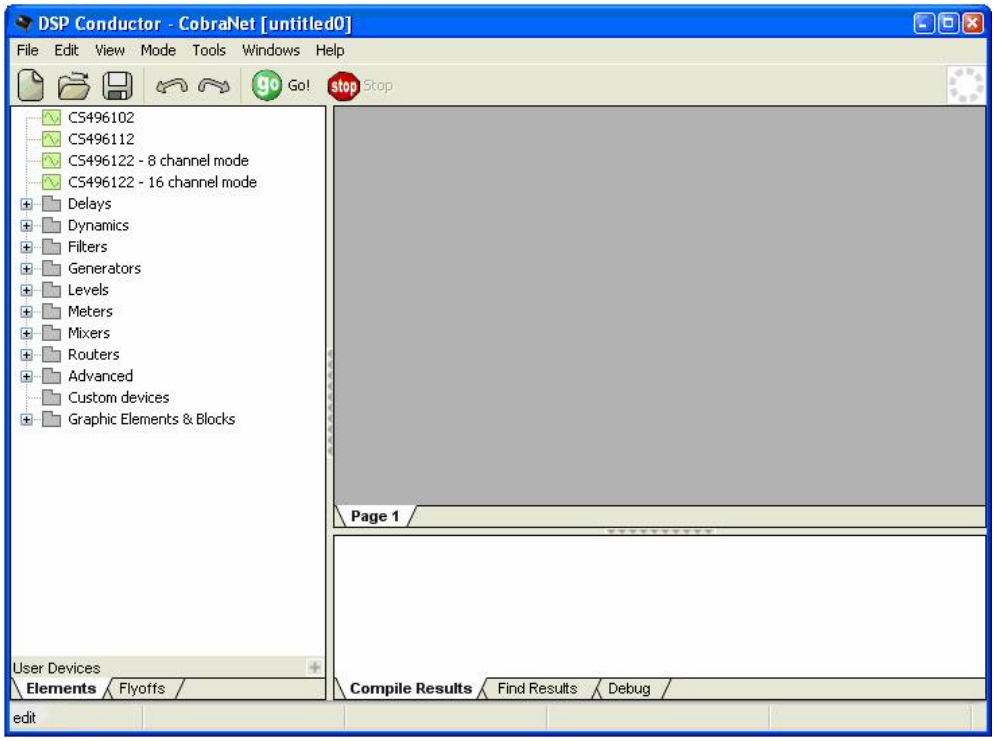

Figure 1. DSP Conductor Main Window

The very top of the window is the ***Title Bar***, which displays the title of the application and the name of the project that is currently loaded.

Below the Title Bar is the ***Menu Bar***. See the *Menu Reference* section for a description of each item in the menu.

Below the Menu Bar is the ***Tool Bar***. The tool bar contains buttons that are shortcuts for some of the most frequently used menu items. See the *Tool Bar Reference* section for a description of each button.

At the far right edge of the Tool Bar is the ***Activity Indicator*** which animates to indicate when DSP Conductor is actively communicating with your CobraNet device.

Below the Tool Bar and to the left side of the window is the ***Utility Frame*** which contains tabs for the *Elements* and *Flyoffs* windows.

The ***Elements Window*** is the palette of digital signal processing (DSP) primitives (elements) that can be used in your configuration. See *Adding Elements* section.

The ***Flyoffs Window*** contains flyoffs that are used to send signals between the pages of a multi-page configuration as as well as connect your processing to the inputs and outputs of your hardware. See *Pages and Flyoffs* section.

Below the Tool Bar and to the right side of the window is the ***Workspace*** that contains the pages of your configuration. The workspace can contain multiple pages, initially there is one page titled *Page 1*.

Below the *Workspace* is the ***Output Frame*** which contains tabs for the *Compile Results*, *Find Results*, and *Debug* functions.

At the very bottom of the window is the ***Status bar***.

2.2 Adding Elements

Elements are the digital signal processing primitives that make up your DSP configuration. The *Elements* window is a tree view that categorizes the elements by function. Click on ⊞ and ⊟ in the tree view to expose or hide the elements in each category. Each available element is described in the *Elements Reference*. Please be aware that the term *device* is sometimes used in the software as a synonym for an element.

To insert an element into your configuration, simply drag the element with the left mouse button from the *Elements* window to a *Workspace* page and release the button at the location where you want the element to be placed. If you change your mind in the middle of the drag, press <Esc> to cancel it.

You can also drag a category from the *Elements* window to the *Workspace* page. When you release the mouse button you will see a popup menu containing the contents of that *Category*. When you select an element from the menu it will be placed on the page. If you press <Esc> or click outside of the menu, the menu will disappear and nothing will be placed.

The *Elements Reference* section contains a description of every Element available in the Elements window.

2.3 Anatomy of an Element

Below is a typical element - a 4 x 2 mixer.

Figure 2. Typical Element - 4 x 2 Mixer

Along the left edge are the *Input Ports* where audio signals enter the element.

Along the right edge are the *Output Ports* where audio signals leave the element.

The yellow dot at the top is the *Enable Port*, which is not used in this release of the software.

An element's *Control Panel* can be opened by double-clicking on the element. However, the controls can not be adjusted at this point. Please read on.

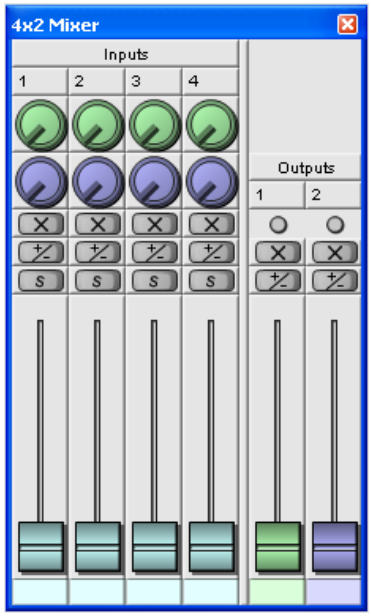

Figure 3. Control Panel - 4 x 2 Mixer

The *Device Properties* dialog for an element is displayed by clicking on the element to select it and either choosing *Tools* ⇨ *Device Properties…* or pressing <Alt> + <Enter>. The *Device Properties* dialog is where you configure the element, in this case by entering the number of inputs and number of outputs.

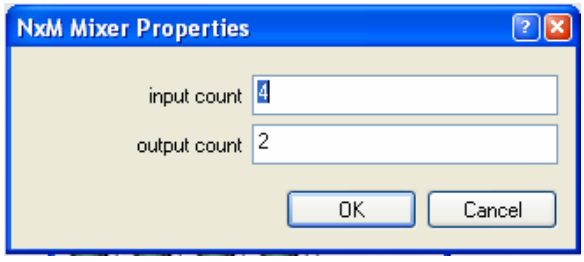

Figure 4. *Device Properties* Dialog - 4 x 2 Mixer

2.4 Editing Basics

Most of the tasks that you perform in DSP Conductor to create and edit your configuration require you to be in *Edit* mode. Use *Mode* ⇨ *Edit* to select *Edit* mode, use the shortcut <Ctrl> + <E>, or click the icon on the tool bar. When you are in *Edit* mode the cursor is the standard arrow.

- To move an element around on its page, simply click on the element and drag it where you want it to be.
- To delete an element, click on it and press <Delete>.
- To change the text displayed on an element, click on it and type the desired text.
- To duplicate an element, press and hold the <Ctrl> key, click on the element, drag the element with the left mouse button, and release the mouse button before releasing the <Ctrl> key. This will leave a copy of the element at the location where you released the mouse button.

All of the editing operations that you can perform in DSP Conductor on a single element may also be performed on all selected elements simultaneously. Selected elements are displayed with a red border.

- To select a single element, click on the element.
- To select multiple elements, click on the elements while holding down the <Shift> key. This will select the element you clicked on as well as all elements between it and the already selected element(s).
- Holding the <Ctrl> key while clicking on an element will toggle its selection.
- You can also select multiple elements by dragging a box that intersects or encloses them.
- Holding the <Shift> key while dragging a box will select those elements while preserving the state of any previously selected elements.
- Holding the <Ctrl> key while dragging a box will toggle the selected state of all of the elements that are inside or intersected by the box.
- Type <Ctrl> + <A> to select all of the elements on the page.
- To clear the selection, click on the *Workspace* somewhere between the elements.

DSP Conductor has its own clipboard format which is only understood by DSP Conductor.

- Selecting *Edit* ⇨ *Copy* or pressing <Ctrl> + <C> copies the selected elements to the clipboard.
- Selecting *Edit* ⇨ *Paste* or pressing <Ctrl> + <V> will change the cursor to a down-pointing arrow with a box behind it which allows you to choose the location at which to paste the elements.
- Selecting *Edit* ⇨ *Cut* or pressing <Ctrl> + <X> is identical to a copy command followed by a delete command.

- To move the selected elements, click on one of them and drag to move them all. You will see the outlines of the selected elements move until you release the mouse button. If you change your mind in the middle of dragging, press <Esc> to cancel.

- To delete all of the selected elements, press <Delete>.

- To tidy up the arrangement of the selected elements, use the *Align* and *Pack* items in the *Edit* menu. Selecting *Edit ⇨ Pack ⇨ Left* or pressing <Ctrl> + <L> is particularly useful for making a column of elements line up.

- To duplicate the selected elements, press and hold the <Ctrl> key while dragging the elements and release the mouse button at the location where you want the duplicates to be placed.

Almost everything you can do in DSP Conductor can be undone by selecting *Edit ⇨ Undo* or pressing <Ctrl> + <Z>. After you undo something, you can redo it selecting *Edit ⇨ Redo* or pressing <Ctrl> + <Shift> + <Z>. If you undo an editing operation, and then perform another operation, the redo command is no longer available. The number of steps of editing that can be undone is configured in the *Tools ⇨ User Preferences...* dialog.

2.5 Selecting the Chip

The three members of the CS4961xx Network Processor family each have different I/O capabilities.

The ***CS496102*** supports 2 channels of digital audio at the serial input port and 2 channels of digital audio at the serial output port. This is ideal for connection to a stereo CODEC. This part also supports 8 channels of audio received from the CobraNet network and 8 channels of audio transmitted to the CobraNet network.

The ***CS496112*** supports 8 channels of digital audio at the serial input port and 8 channels of digital audio at the serial output port. This part also supports 8 channels of audio received from the CobraNet network and 8 channels of audio transmitted to the CobraNet network.

The ***CS496122*** supports two modes of operation, depending on how the converters are connected to the processor:

- In 8-channel mode it supports, like the CS496112, 8 channels of digital audio at the serial input port, 8 channels of digital audio at the serial output port, 8 channels of audio received from the CobraNet network, and 8 channels of audio transmitted to the CobraNet network. This is the mode in which the EV-2 development system must operate.

- In 16-channel mode it supports 16 channels of digital audio at the serial input port, 16 channels of digital audio at the serial output port, 16 channels of audio received from the CobraNet network and 16 channels of audio transmitted to the CobraNet network. 16-channel mode is only supported at the 48 kHz sampling rate. Even if you know you only intend to run your configuration at 48 kHz, you may still choose to run in 8-channel mode because it consumes less system resources.

You can run a project designed for the CS496102 on a CS496112 or a CS496122. You can run a project designed on the CS496112 on the CS496122. You can also run a project designed for the CS496122 in 8-channel mode on the CS496112.

The *Elements* window contains special *Processor* elements representing the CS496102, CS496112, and one for each mode of the CS496122. To specify which version and mode of the processor you intend to target with your DSP Conductor project, simply drag the appropriate *Processor* element onto a *Workspace* page. You can change the type later by editing the properties of the element. You can only put one *Processor* element in your project or the compile process will fail.

The CS4961xx *Processor* elements are special because when one is inserted, the *Flyoffs* window is populated with the correct *System* flyoffs representing the points where the digital audio data enters and leaves the processor you have selected.

Flyoffs, which are covered in greater detail in the section titled *Pages and Flyoffs*, are objects that allow you to wire audio signals between pages of your configuration. *System* flyoffs are special because they conceptually allow you to wire between the hardware and a point in your configuration. You cannot change the name of a *System* flyoff.

The audio signals from the serial input ports are represented by flyoffs named *Audio.Input.1*, *Audio.Input.2*, etc. The audio signals sent to the serial output ports are represented by flyoffs named *Audio.Output.1*, *Audio.Output.2*, etc. The audio signals received from the CobraNet network are represented by flyoffs named *CobraNet.Input.1*, *CobraNet.Input.2*, etc. The audio signals sent to the CobraNet network are represented by flyoffs named *CobraNet.Ouput.1*, *CobraNet.Ouput.2*, etc.

Flyoffs are automatically arranged in the *Flyoffs* window in categories that are generated by breaking up the flyoffs' names at the locations of the periods. Refer to the following figure that shows the partially expanded contents of the *Flyoffs* window after inserting the *CS4996102* element into the configuration and dragging the *Audio.Input* category into the configuration.

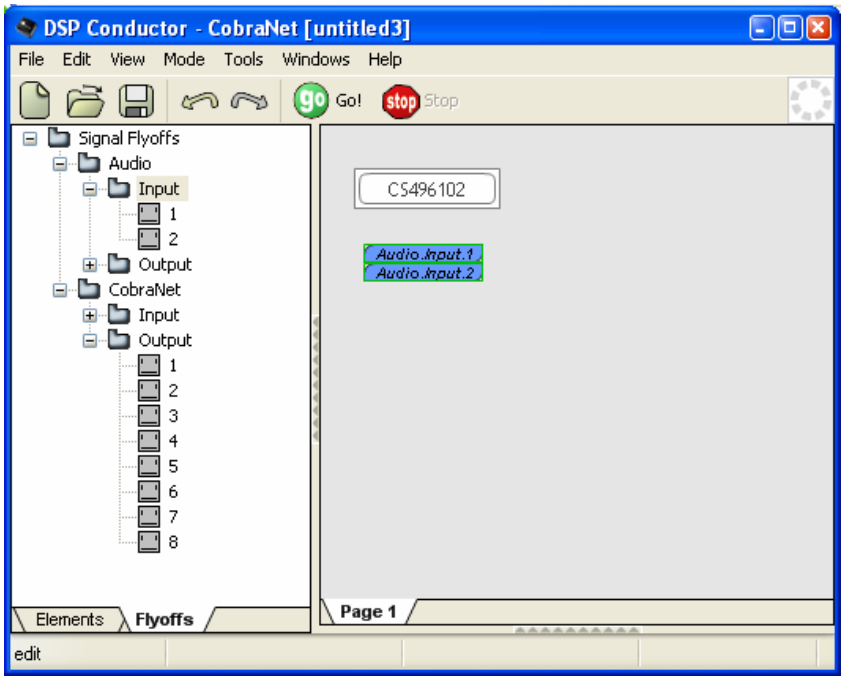

Figure 5. The *Flyoffs* Window

You can drag an individual flyoff from the *Flyoffs* window to a *Workspace* page, and you can also drag a category of flyoffs from the *Flyoffs* window, which will place all of the flyoffs in that category onto the page. The flyoffs have a green outline until they are properly wired to elements.

If you are using the EV-2 CobraNet Development System, you must select the *CS496122* element in 8-channel mode. The *CobraNet EV-2 Development System Manual* describes how the EV-2 software can be used to configure the serial ports, but the default configuration is that stereo analog inputs are represented by flyoffs named *Audio.Input.1* and *Audio.Input.2*, and the AES input is represented by *Audio.Input.3* and *Audio.Input.4*. Likewise, the stereo analog outputs are represented by *Audio.Output.1* and *Audio.Output.2* and the AES transmitter is represented by *Audio.Output.3* and *Audio.Output.4*.

The CobraNet CO2 module from Attero Tech is based on the CS496102 processor. It supports stereo analog inputs represented by *Audio.Input.1* and *Audio.Input.2* and stereo analog output represented by *Audio.Output.1* and *Audio.Output.2*.

The control panel of the *CS4961xx* element contains controls for monitoring the status of the CobraNet interface for activity, faults. For convenience, it allows you to assign bundle numbers to the CobraNet transmitters. However, the bundle numbers are not stored as part of the DSP configuration.

All of the processors in the CS4961xx family support 8 CobraNet bundle receivers and 4 bundle transmitters. You cannot make use of all of these directly without changing the sub channel mappings of the CobraNet core. See the *CobraNet Hardware User's Manual* for information about this topic. By default, all of the processors map the first CobraNet bundle receiver to *CobraNet.Input.1* thru *CobraNet.Input.8*, and *CobraNet.Output.1* thru *CobraNet.Output.8* are mapped to the first CobraNet bundle transmitter. Additionally, the CS496122 in 16-channel mode maps the second CobraNet bundle receiver to *CobraNet.Input.9* thru *CobraNet.Output.16* and *CobraNet.Output.9* thru *CobraNet.Output.16* are mapped to the second CobraNet bundle transmitter.

2.6 Wiring Basics

You must add wires to your configuration to connect the elements and define the signal flow. To do so, you must be in *Wire* mode. To enter *Wire* mode, select *Mode* ⇨ *Wire* or press <Ctrl> + <W>. The cursor changes to a pair of pliers ( ) when you are in *Wire* mode.

The simplest way to add a wire is to click on the output port (along the right edge) of an element and drag to the input port (along the left edge) of another element. Audio wires can fan out meaning you can connect an output port to multiple input ports but you cannot connect multiple output ports to an input port. (If you need to do this, add a mixer.)

If you want to wire all of the output ports of an element to the input ports of another element, click in the center of the first element and drag to the right and when you are over the second element, release the mouse button. If you want to do this in the reverse, drag to the left. The direction you move the mouse immediately after clicking the mouse determines whether you begin dragging from the inputs ports (drag left), the output ports (drag right), or the enable port (drag up).

You can wire from an ad-hoc selection of ports, as long as they are all the same type, by selecting them using <Ctrl> + Click and then dragging from one of them.

If you are dragging multiple wires, either from an element or from an ad-hoc selection of ports and you want to connect the wires to a column of devices, drag the wires over a port of the top device in the column and release the mouse button.

If you are dragging wires from multiple input ports and you want them all to connect to a single output port, hold the <Shift> key as you drag over the output port (the cursor will show a "G", indicating you are ganging wires) and release the mouse button.

If you click in the middle of a wire and drag, you will create a *Waypoint* that allows you to route the wire in an aesthetically pleasing manner. Once you have placed the waypoint, you will need to go back to *Edit* mode (<Ctrl> + <E>) to move it again.

If you select a waypoint in *Wire* mode and press <Delete>, the waypoint will be removed and the wire will travel more directly between the ports it connects.

If you select a waypoint in *Edit* mode and press Delete, the waypoint and the wires connected to it will be removed.

If you wire the elements of your configuration such that you have created a feedback loop, the compile will fail.

2.7 Save

It is always a good idea to save your work often. Choose *File* ⇨ *Save* to save you configuration. If you have not saved the project previously, you will be presented with the *Save As…* dialog allowing you to enter the name and location to store the project. DSP Conductor projects are saved with the file extension *.CPA* which stands for Conductor Project Archive.

2.8 Go

When you are ready to listen to your configuration, click the *Go* button on the Tool Bar or select *File* ⇨ *Go* or press <F9>. This invokes a series of three distinct steps: *Check and Compile*, *Download*, and *Connect*. (These steps are also available individually in the *Tools* ⇨ *Debug* menu, however there is little reason you should need them individually.)

2.8.1 Check and Compile

The *Check and Compile* step analyzes your configuration and generates the DSP code necessary to implement it. If the compile fails for any reason you will get an error message and the *Compile Results* window will display more information about why the compile failed. If the compile operation succeeds, the *Compile Results* window will display any warnings, if present, and a table of DSP resources consumed by the configuration.

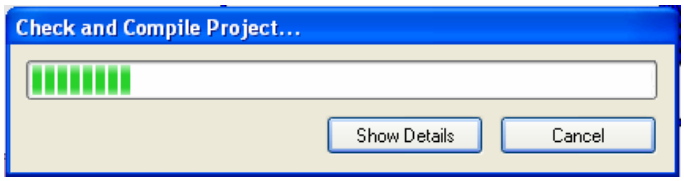

Figure 6. *Check and Compile* Dialog

2.8.2 Download

During the *Download* step you are first presented with the *Download* dialog. Here you choose whether you actually want to download your configuration to your CobraNet device or simply emulate the configuration.

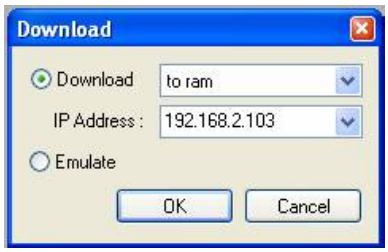

Figure 7. *Download* Dialog

If you choose *Emulate*, you are allowed to adjust the controls without actually connecting to a CobraNet device. This does not actually simulate the DSP processing but it does allow you to see the ranges of the various controls and it gives you an opportunity to enter some initial settings before you actually pass audio.

If you choose *Download*, you must select whether to download to RAM or to Flash memory. Configurations downloaded to RAM will be lost when the CobraNet device is reset or power-cycled. Configurations downloaded to Flash memory will be retained when the device is reset, but this takes a little longer. While you are iterating on your configuration it is recommended to simply download to RAM.

You must also enter the IP address of your CobraNet device. It is recommended that you have the CobraNet Discovery application running at this point so you can see (and assign if necessary) the IP address of your CobraNet device. See AN275, *"Getting Started with DSP Conductor"* for a description of this process. The IP address field is stuffed with the most-recently used IP address, and the dropdown list contains all of the addresses that have been used.

When you click *OK*, DSP Conductor will verify that the device is responding and confirm that it is either a Cirrus Logic EV-2 or a CobraNet CO2 from Attero Tech. If not, it will check that you have installed a Manufacturer Registration Key. If not, you will have to choose another IP address.

See the *Introduction* for instructions on how to obtain a Manufacturer Registration Key.

Assuming the device is responding and you are authorized to download to it, the configuration is transferred over the network to the CobraNet Device using TFTP (Trivial File Transfer Protocol). If this succeeds, you proceed to the *Connect* step.

If you are using an EV-2 and you wish to download your configuration to the Flash memory so it will run automatically on power-up or reset, you must enable *Variable Persistence* in the CobraNet Core. See the CobraNet datasheet for more information.

2.8.3 Connect

Connect simply means DSP Conductor establishes SNMP (Simple Network Management Protocol) communication with the CobraNet device running your configuration. Any changes you make to the settings (See *"How to Gesture"*) are immediately sent over the network and you can hear the effect in real time. DSP Conductor is also continuously polling for *meter* and *clip* information and making that information visible in each element's control panel. While you are connected, the activity indicator at the far right side of the tool bar is animated.

If there is a problem causing the SNMP communications with the device to fail, an error message is reported and you will no longer be connected. A few of the many possible causes of communications errors are: A network cable is bad or has been unplugged, the CobraNet device has been reset or power cycled, or a configuration has been downloaded to the CobraNet device from another copy of DSP Conductor.

2.9 How to Gesture

To adjust the control settings, you must be in *Gesture* mode. In *Gesture* mode the cursor becomes a hand icon. You select *Gesture* mode from the *Mode* ⇨ *Gesture* menu item or the shortcut <Ctrl> + <G>. Gesturing will only have an effect if you are *Connected*, which means the activity indicator at the right edge of the tool bar is animated.

To gesture a knobs or fader (slider) item, simply click on the control and move the mouse up or right to increase the value, or down or left to decrease the value. When gesturing a knob, do not attempt to 'twist' the control by moving the mouse in a circle around it - use the method described above.

If you hold the <Shift> key down while gesturing a knob or fader, you will enter 'high resolution' mode where the control responds at one tenth of the speed it does normally.

To gesture a button control, simply click on it.

Some controls, such as the *Type* control in the *High Pass* and *Low Pass* filters allow you to choose from a list of legal values. When you click on the control, a menu pops up containing the legal values of the control. Simply click on the one of you choice.

Most controls also accept typed-in values. If you type in a value and press <Enter>, the value will be assigned to the selected control. You must be careful to specify the suffix character in some circumstances. For example, a time constant control may have a range of 10ms to 10s. Since the units of this control are seconds, when you type in *20*, it is interpreted as *20 seconds* which is clipped to the allowed range and the result is 10 seconds. If you desire to set the control to *20ms* you must type in *20m* or *20ms*.

All of the rules for multiple selection such as <Shift> + Click and <Ctrl> + Click as well as dragging a selection box and <Shift> or <Control> + dragging a selection box that are described in *Editing Basics* apply to *Gesture* mode as well. Any gesture or typing operation that you perform while multiple controls are selected will apply to all of the selected controls.

The *Undo* and *Redo* commands do not apply to adjusting control values.

2.10 Pages and Flyoffs

Your DSP Conductor configuration is not limited to a single *Workspace* page. You can spread your configuration across multiple pages as a method of organizing it into functional sections, or simply because you don't like having to deal with the scroll bars inherent with placing a large configuration on one page.

To add a page, right-click in the space to the right of the tabs at the bottom of the *Workspace* window and choose *Add Page*. This will add a new page to the workspace with an automatically generated title (*Page 4*, *Page 5*, etc.)

You can rename a page by double-clicking on the page's tab and typing a new name.

You can rearrange the order of the pages by clicking on a page tab and dragging it left or right to its new location.

You can delete a page by right-clicking on the page's tab and selecting *Delete* from the popup menu.

So how do you wire between elements on different pages? The answer is with *Flyoffs*. A flyoff is an item on the page that gives a signal a name that can be referenced on a different page. Two flyoffs on different pages with the same name are wired together.

To create a flyoff, begin dragging a wire (or multiple wires) and press the right mouse button while holding down the left button. This will drop a flyoff for each wire you were dragging. Initially these flyoffs have no names, so they cannot yet be used to connect anything.

To give a flyoff a name, you must be in *Edit* or *Wire* mode. Select the flyoff and begin typing the name or right-click it. This will bring up the field for entering the flyoff's name. Below the entry field is a list of the other flyoffs that you have already created that match what you have typed so far. This makes it easier to connect a flyoff to an existing flyoff without having to retype the entire name, simply click on it in the list.

If you give two flyoffs the same name and they are each connected to output ports of elements, the flyoffs and the wires connected to them will be displayed with a red outline. This indicates that you have created an illegal connection and you must delete or rename one of them or the compile will fail.

Flyoffs that are not assigned a name or that are assigned a name that has no partner are displayed with a green outline. This indicates it is part of an incomplete connection. Once you assign another flyoff to the same name the green border will disappear. If you compile while any of your wires or flyoffs are green, you will get a warning about an unconnected input or output.

The names of all of the flyoffs that you have created are displayed in the *Flyoffs* window. You can drag a name from this window to your *Workspace* page and where you release the mouse button, a new flyoff with this name is created.

It is a good idea to name related flyoffs using the convention of fields separated by periods. For example *Banana.1*, *Banana.2*, *Banana.3*, etc. This causes the flyoffs to be arranged in the *Flyoffs* window in a category called, in this case, *Banana* containing *1*, *2*, *3*, etc. If you drag the category from the *Flyoff* window to your *Workspace* page, you will deposit the entire contents of the category at once. This is very handy when working with bus-like groups of signals.

2.11 Snapshots

A *Snapshot* is a file that saves the settings of all of the controls in your configuration. This may be useful, for example, if you want to perform an A/B comparison between two different sets of settings.

Snapshots are managed through the *Tools* ⇨ *Snapshots...* menu item. This brings up the *Snapshot Manager* dialog. This item is only available if you are connected, which means you have pressed *Go*, everything succeeded, and the activity indicator is animated.

Figure 8. *Snapshots Manager* Dialog

Pressing *New* will allow you to enter a name for the snapshot and all of the control settings will be saved under this name. You can also use this to save new settings to an existing snapshot.

If you select a snapshot name from the list and press *Load*, all of the control settings will be loaded from this snapshot and applied to your CobraNet device and the *Snapshots Manager* dialog will go away.

If you select a name from the list and press *Delete*, the selected snapshot will be deleted.

To close the dialog box, click the red *X* in the upper right corner.

Snapshot operations may not be undone with *Edit ⇨ Undo* menu item.

2.12 Deliverables

If you are using DSP Conductor to develop a product, you may want to write some software to adjust some settings in your configuration, and you may like to be able to mass produce the configuration you have created. DSP Conductor will generate deliverables that help you achieve these goals.

Settings may be adjusted through the host port for products that contain a microcontroller. Settings may be adjusted through SNMP if you are writing a program to control your product over an IP network. In order to adjust the settings, you need to know the details about the layout of the coefficients that you must change. This is a complicated topic that is discussed in detail in AN279, *"Controlling and Monitoring DSP Conductor Configurations"*. DSP Conductor will generate the C header file containing the required information. This header file is the first deliverable. DSP Conductor can also generate an XML file that contains a superset of the information contained in the header file. You can write a program to parse this file directly, or you can transform this file into a C header file that is more suitable to your needs

In order to mass produce your configuration, you need an image of the Flash memory that can be burned into your product during manufacture. This Flash image is the second deliverable, but it is only available if you have installed a Manufacturer Registration Key.

The *Tools ⇨ Generate Deliverables* sub-menu contains items for generating these deliverables. *Tools ⇨ Generate Deliverables ⇨ As XML...* brings up a browser to select where to save the XML file. By default, the file will have the same name as your configuration with the *.xml* file extension. *Tools ⇨ Generate Deliverables ⇨ As Header...* brings up a browser to select where to save the C header file. By default, the file name will be the same as your configuration with the *.h* file extension.

Tools ⇨ Generate Deliverables ⇨ Patch Image... is only available if you have a Manufacturer Registration Key installed. This item will allow you to browse for the base firmware image to patch with the DSP configuration you have designed with DSP Conductor. It will then allow you browse for the location to save the patched firmware image.

When you choose any of the items in the *Generate Deliverables* submenu, the settings and configuration that are described in the generated deliverables is your configuration as of the last compile operation. If you have done any editing or have adjusted any settings, you should press *GO* one more time before you generate deliverables.

3. Reference

3.1 Tool Bar Reference

The buttons on the toolbar are simply shortcuts to commonly used menu items.

Table 1: Tool Bar Commands

Icon	Menu Command	Shortcut
	*File ⇨ New*	<Ctrl> + <N>
	*File ⇨ Open*	<Ctrl> + <O>
	*File ⇨ Save*	<Ctrl> + <S>
	*Edit ⇨ Undo*	<Ctrl> + <Z>
	*Edit ⇨ Redo*	<Ctrl> + <Shift> + <Z>
	*File ⇨ Go*	<F9>
	*File ⇨ Stop*	no shortcut

3.2 Menu Reference

3.2.1 File Menu

New — Creates a new configuration. This will launch a new copy of DSP conductor containing new configuration with a single, blank page in the workspace.
Shortcut: <Ctrl> + <N>

Open — Presents the Windows file browser to select a previously saved project to open. If the current project is new and has not been modified in any way, the selected configuration will open in the current copy of DSP Conductor; otherwise the configuration will be opened in a new copy of DSP Conductor. You cannot have the same configuration open in two copies of DSP Conductor at the same time.
Shortcut: <Ctrl> + <O>

Save — Save the current project to its current name. If the current project has not yet been given a name, the *Save As...* dialog will appear. Files are saved with the

extension *.cpa* which stands for "Conductor Project Archive".
Shortcut: <Ctrl> + <S>

Save As… Presents the *Save As...* dialog to enter a new name for the project.
Shortcut: <Ctrl> + <Alt> + <S>

Properties… Presents the *Project Properties* dialog which allows you to specify the properties of the current configuration. These properties are used during the compile process, so changing the value here will take effect the next time you press *Go*.

Figure 9. *Project Properties* Dialog

This version of DSP Conductor has no configurable project properties of general interest. This will change in future releases.

Advanced... presents the *Advanced Properties* dialog. Most of the items here should not be tampered with.

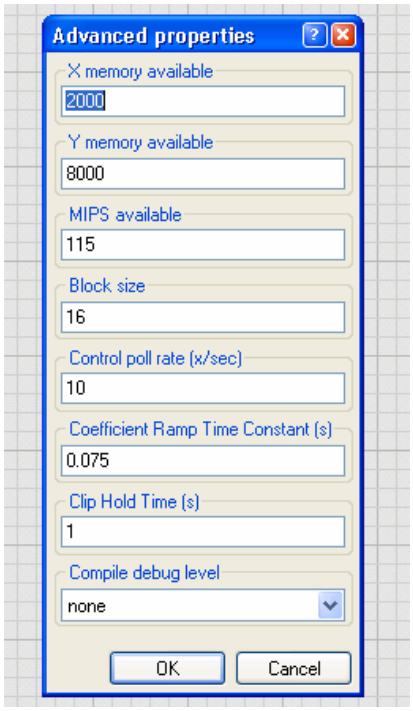

Figure 10. *Advanced Properties* Dialog

X memory available indicates how many words of memory are available in the X memory space of the processor. Should be *2000* – do not change!

Y memory available indicates how many words of memory are available in the Y memory space of the processor. Should be *8000* – do not change!

MIPS available indicates how many millions of instructions are available per second on the processor. Should be *115*, do not change!

Block size specifies the length of the vector in which audio samples are processed. The longer the block size, the more efficient the processing, but you can save a very small amount of memory and latency if the block size is smaller. The range is 4 to 16 samples.

Control poll rate (x/sec) indicates how fast DSP conductor requests new SNMP values from the device while connected. This effects how much network traffic is generated by the program, but it does not actually effect the compiled configuration.

Coefficient Ramp Time Constant(s) specifies the time constant used for ramping coefficients in the DSP. If the value is too large the processor will appear to respond sluggishly to controls, if it is too small you will hear audio artifacts when you adjust controls. The recommended value is 0.075 seconds.

Clip Hold Time(s) specifies how long the clip light stays on after an element detects internal clipping.

Compile debug level enables output of internal details of the compile process to the Compile Results window. This is probably only useful if you are talking to a technical support person.

Recent Files This is a list of the four most-recently opened configurations. Choosing an item from the list is identical to choosing *Open* and selecting that file.

Go Compile the configuration, choose the target, download the project to the target it and control it.
Shortcut: <F9>

Stop Cease control communications with the processor.

Exit Close the application. If you have made any changes to your configuration, you will be asked if you want to save.

3.2.2 Edit Menu

Undo Undo the last editing operation. The text of the menu item will change to indicate what operation will be undone. The number of levels of undo is specified in the *Tools ⇨ User Preferences…* dialog. By default you may undo the last 100 editing operations.
Shortcut: <Ctrl> + <Z>

Redo Redo the last editing operation that was undone. When you perform any editing operation (except *Undo*), *Redo* becomes unavailable.
Shortcut: <Ctrl> + <Shift> + <Z>

Cut Remove the selected objects from the configuration and place them in the clipboard. This menu item is only available in *Edit* mode when one or more objects are selected.
Shortcut: <Ctrl> + <X>

Copy In *Edit* mode, place a copy of the selected objects in the clipboard. This uses a private clipboard format that is only understood by DSP Conductor.
In *Gesture* mode, copy the values of the selected controls to the clipboard. This uses a text format that is understood by most programs. When more that one control is selected, the values are separated by Tab characters.
Shortcut: <Ctrl> + <C>

Paste In *Edit* mode, prepare to insert the objects previously cut or copied to the clipboard. The cursor changes to a down-pointing arrow with a box indicating you need to select the location to insert the objects. If you press the <Esc> key, the operation is cancelled. When you click the mouse, the objects will be inserted.
In *Gesture* mode, paste the values from the clipboard into the selected controls.
Shortcut: <Ctrl> + <V>

Duplicate Bring up the *Duplicate* dialog that allows you to easily make multiple copies of the selected element(s) and arrange them in a column or row, in a single operation.
Shortcut: <Ctrl> + <D>

Delete In *Edit* mode, delete the selected objects.

This item is not available in *Gesture* Mode.

In *Wire* mode, delete the selected wire. If no wire is selected but a wiring port is selected, delete all of the wires connected to the port. If no port is selected but a device is selected, delete all of the wires connected to the device.
Shortcut: <Delete>

Select All Select all of the objects in the current window.
Shortcut: <Ctrl> + <A>

Find Bring up the *Find* dialog that allows you to easily find elements, controls, and flyoffs in your design. The found items are listed in the *Find Results* tab, where you can click on them to hilight them in the workspace page.
Shortcut: <Ctrl> + <F>

Clear Find Hilight Remove the highlighting applied to the workspace page as a result of clicking in the *Find Results* window.

Align Aligns selected objects.

Left aligns the left edges of the selected objects with the left edge of the leftmost selected object.

Top aligns the top edges of the selected objects with the top edge of the topmost selected object.

Right aligns the right edges of the selected objects with the right edge of the rightmost selected object.

Bottom aligns the bottom edges of the selected objects with the bottom edge of the bottommost selected object.

Pack Aligns selected objects closely together with no space between them.

Left aligns the left edges of the selected objects with the left edge of the topmost selected object and adjust the vertical positions of the subsequent objects so they are packed in a column with no space between them.
Shortcut: <Ctrl>+<L>

Top aligns the top edges of the selected objects with the top edge of the leftmost selected object and adjust the horizontal positions of the subsequent objects so they are packed in a row with no space in between them.

Right aligns the right edges of the selected objects with the right edge of the topmost selected object and adjust the vertical positions of the subsequent objects so they are packed in a column with no space between them.

Bottom aligns the bottom edges of the selected objects with the bottom edge of the leftmost selected object and adjust the horizontal positions of the subsequent objects so they are packed in a row with no space in between them.

Arrange Adjusts the order in which a selected objects are arranged on the Workspace page.

Bring to Front adjusts the *Z* order of the selected object so it is in front of all other objects on the page.

Send to Back adjusts the Z order of the selected object so it is behind all other objects on the page.

Copy Special Copies specific information to the clipboard.

Picture (Enhanced Metafile) copies a picture of the current page to the Windows clipboard in Enhanced Metafile format (*.emf*). This is a vector format that scales better that the bitmap format obtained when you press <Print Screen>.

Label (Text) copies the label(s) of the selected object(s) to the clipboard in text format (*.txt*). Each label will be on its own line..

3.2.3 View Menu

Zoom Affects the zoom factor in effect for the current window. Zooming, like panning, is cosmetic only. The zoom setting is not saved in the configuration and does not interact with *Undo/Redo*.

Satellite zooms all the way out so the elements on the page are very small, as though viewed from space.

Out zooms out by a factor of 2, making everything on the page half as big.
Shortcut: <Ctrl> + <Page Up>

Normal resets the zoom factor for the current page to the nominal value.
Shortcut: <Ctrl> + <Home>

In zooms in by a factor of 2, making everything on the page twice as big.
Shortcut: <Ctrl> + <Page Down>

Microscope zooms all the way in, making the elements on the page huge, as though viewed through a microscope.

Fit all adjusts the zoom factor so all of the objects on the current page can fit within the current extents of the window.

Show/Hide Output Frame Toggle the visibility state of the Output Frame. The Output Frame is the tabbed window at the lower right containing the *Compile Results* window and the *Debug* output window. The same result can be achieved by clicking on the tiny row of triangles that reside in the splitter between the *Design* frame and the *Output* frame.
Shortcut: <F10>

Show/Hide Output Frame Toggle the visibility state of the *Utility* frame. The *Utility* frame is the tabbed window on the left containing the *Elements* window and the *Flyoffs* window. The same result can be achieved by clicking on the tiny row of triangles that reside in the splitter between the *Utility* frame and the *Design* frame.
Shortcut: <F11>

Show/Hide Status Bar Toggle the visibility state of the status bar.
Shortcut: <F12>

3.2.4 Mode Menu

Edit Change to *Edit* mode. Edit mode allows you to move objects, edit their properties and delete them. Certain operations, such as inserting a new element, will switch to *Edit* mode automatically. See *Editing Basics*.
Shortcut: <Ctrl> + <E>

Wire Change to *Wire* mode. Wire mode is used to add, delete and arrange wires. See *Wiring Basics*.
Shortcut: <Ctrl> + <W>

Gesture Change to *Gesture* mode. *Gesture* mode is used to adjust control settings. After you press *GO*, if everything succeeds, *Gesture* mode is selected automatically. If you are not connected, *Gesture* mode has no effect. See *How to Gesture*.
Shortcut: <Ctrl> + <G>

Paint Change to *Paint* mode. *Paint* mode is used to change the color of objects on the screen. The color palette will appear allowing you to choose the color. Clicking on an object paints it the chosen color. Typing <Ctrl> + <C> will choose the color of the selected object.

To change the background color of a window, you must use the *Page Properties* dialog accessible by right-clicking on the *Pages* tab and choosing *Properties…*

3.2.5 Tools Menu

Group Bind together the selected objects into a group so their relative positions are always maintained. This item is only available if two or more objects are selected.
Shortcut: <Ctrl> + <I>

Ungroup Unbind the objects in a group so they may be moved individually. This item is only available if a group is selected.
Shortcut: <Ctrl> + <U>

ExpressionLabel... Bring up the ExpressionLabel dialog box. This dialog allows you to assign a series of labels to a selection of Elements or a series of names to a selection of Flyoffs.

The series of names is generated by an *ExpressionLabeling* formula. A formula is comprised of 2 basic tokens: literal (normal) characters and commands. A command is a processing instruction surrounded by curly braces which generates a string based on the current iteration index.

There are 6 types of *ExpressionLabel* commands:

Table 2: *ExpressionLabel* Commands

{X+=Y}	auto increment by Y, starting with X
{X+=Y/Z}	auto increment by Y, starting with X, every Z items
{X-=Y}	auto decrement by Y, starting with X
{X-=Y/Z}	auto decrement by Y, starting with X, every Z items
{label1,label2,label3}	cycle between label1, label2, and label3
{*X}	replace with wildcard series X

For example, use the following ExpressionLabel to label a set of flyoffs: *Flyoff.1*, *Flyoff.2*, *Flyoff.3*, etc.:

Flyoff.{1+=1}

To label a series of flyoffs: *Flyoff.gain.1*, *Flyoff.mute.1*, *Flyoff.gain.2*, *Flyoff.mute.2*, the following would be used:

Flyoff.{mute,gain}.{1+=1/2}

If the selected items already have text associated with them, the *Expresssion-Label* engine does its best to reverse engineer a formula which represents the progression. In the case that the engine cannot determine a pattern, it represents the series with a wildcard series. For example, take a set of items with text *Bob.1.input*, *Joe.2.input*, and *Frank.3.input*. The engine will detect the numbering sequence as well as the *.input* suffix but it will not be able to detect a pattern from *Bob*, *Joe,* and *Frank* so it will use a wildcard series. The generated *ExpressionLabel* formula will look like:

{*0}{1+=1}.input

The **0* will be replaced with the series defined by first (0-based) wildcard. If additional wildcards were detected their numbers would be represented by their index. To change the previous example to *Input.1.Bob*, *Input.2.Joe*, and *Input.3.Frank* you would change the automatically generated *ExpressionLabel* formula to:

Input.{1+=1}{*0};

Replace Text... Brings up the *Replace Text* dialog. This dialog allows you to replace the text occurring on the surface of the selected elements. In this dialog you enter the text to find and the new text to replace it with. This menu item is only available if at least one element is selected.
Shortcut: <Ctrl> + <E>

Define User Device... Brings up the *Define User Device* dialog. This menu item is typically used when you have created and wired a schematic within a hierarchical block, and you want to save this block to the *Elements* window so it can be easily reused in future configurations. This menu item is not available if you do

not have an element selected. The *Define User Device* dialog allows you to enter the name for your creation, or select an existing name if you wish to overwrite the previous version. Your element will show up in the *User Devices* panel at the bottom of the *Elements* window.
Shortcut: <Ctrl> + <E>

Properties Depending on the type of the object or objects selected, bring up the *Device Properties* dialog, the *Control Properties* dialog or the *Graphic Properties* dialog.
Shortcut: <Alt> + <Enter>

User Preferences... Bring up the *User Preferences* dialog. These preferences are shared by all copies of DSP conductor, although some changes may not take effect until the program is restarted.

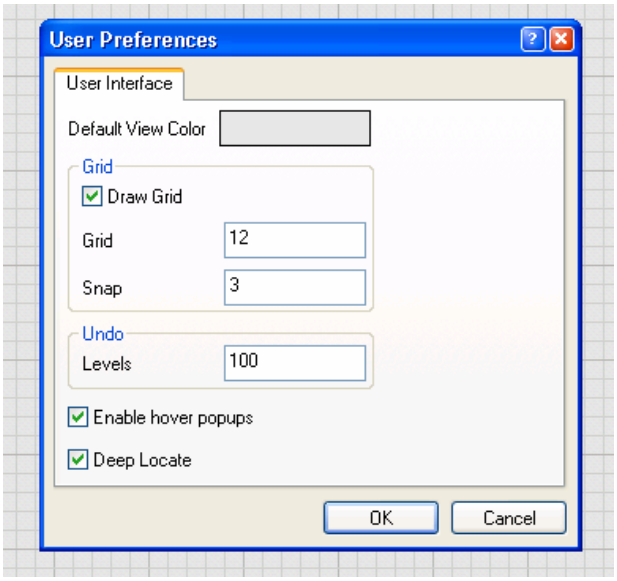

Figure 11. *User Preferences...* Dialog

Default View Color selects the color of all subsequently created pages. Click on the color box to bring up the color chooser dialog box.

Draw Grid displays a cool grid that looks like e-paper.

Grid (Size) specifies the size of the grid, if it is turned on. You should make the grid a multiple of the snap value.

Snap specifies the unit to which the position of elements on the screen is quantized. This will not change the position of any elements until the next time you move them.

Undo Levels specifies the count of operations that can be undone with *Edit* ⇨ *Undo*. The *Undo* history is stored in memory, so you may want to reduce this number if your computer does not have a lot of memory.

Enable hover popups enables hover text to popup near the cursor when you hold the mouse still over a control or a wiring port.

Deep Locate has no effect in this version of the software.

Snapshots... See the *Snapshots* section for a description of this menu item.

Generate Deliverables See the *Deliverables* section for a description of the itmes in this sub-menu.

Register Software... Brings up the *DSP Conductor Registration* dialog. This dialog allows you to easily compose an e-mail to Cirrus Logic requesting a Manufacturer Registration Key. It also allows you to enter that key after it has been sent back to you by a representative of Cirrus Logic.

Debug Pay no attention to the items in the debug menu. None of these items are necessary for normal use of DSP Conductor, so they are not documented here.

3.2.6 Windows Menu

Next Window Switches focus to the next running copy of DSP Conductor.
Shortcut: <Ctrl> + <Alt> + <N>

Previous Window Switches focus to the previous running copy of DSP Conductor.
Shortcut: <Ctrl> + <Shift> + <N>

Next Page Switches to the next page in the workspace. If you have a hierarchical block open that contains multiple pages, this command will change to the next page in that block.

Previous Page Switches to the previous page in the workspace. If you have a hierarchical block open that contains multiple pages, this command will change to the previous page in that block.

List of open projects Presents a list that contains the names of the projects open in all copies of DSP Conductor that are currently running and allows you to quickly switch between them.

3.2.7 Help Menu

About Displays the version of DSP Conductor. Click anywhere to make the window disappear.

3.3 Elements Reference

The following text describes each available element. The section immediately below describes the format in which the element descriptions will be presented. Each subsequent section describes the actual elements.

3.3.1 Element Name (Example)

Purpose

A description of the purpose this element serves.

Properties

This section contains a list of each of the properties of the element and a brief description of each. If the element has no properties, this section will not exist. (Properties are edited by selecting the element and typing <Alt> + <Enter>.)

Controls

This section contains a list of each control in the element and a brief description of each. If the element has no controls, this section will not exist. Controls exist in the element's control panel, which is exposed by double-clicking. You must be connected and be in *Gesture* mode (<Ctrl> + <G>) in order to adjust controls.

Inputs

This section contains a list of the input ports on the element. If the element has no inputs, or has only a single input about which nothing interesting can be said, this section will not exist.

Outputs

This section contains a list of the output ports on the element. If the element has no outputs, or has only a single output about which nothing interesting can be said, this section will not exist.

Notes

This optional section provides a place for more detailed discussion of the element, if necessary.

3.3.2 CS4961xx

Purpose

This element identifies the model number of the processor you have chosen to target for your configuration. It also allows you to monitor the CobraNet status and allows you to assign CobraNet bundle numbers.

Properties

Chip selects the processor model. The choices are CS496102, CS496112, CS496122 in 8-channel mode and CS496122 in 16-channel mode.

Controls

Conductor indicates this device is the conductor of the CobraNet Network.

Fault indicates a fault has occurred.

Tx Error flashes to indicate a transmit error.

Rx Error flashes to indicate a receive error.

Error Count indicates a count of the number of errors since the CobraNet device has been reset.

Output Bundle Active (1 or 2) indicates that the respective bundle number is being transmitted.

Output Bundle Number (1 or 2) specifies the bundle number to transmit on.

Input Bundle Active (1 or 2) indicates that the respective bundle number is being received.

Input Bundle Number (1 or 2) specifies the bundle number to receive.

Inputs

The CS4961xx element is special because it does not have any input ports, but instead it generates *System* flyoffs that allow you to wire to the audio sources that are the inputs to your configuration.

Audio.Input.1 through Audio.Input.16 – Depending on the model selected, you will have 2, 8, or 16 Audio Inputs.

CobraNet.Input.1 through CobraNet.Input.16 – Depending on the model selected, you will have 8 or 16 CobraNet Inputs.

Outputs

The CS4961xx element is special because it does not have any output ports, but instead it generates *System* flyoffs that allow you to identify the outputs from your configuration.

Audio.Output.1 through Audio.Output.16 – Depending on the model selected, you will have 2, 8, or 16 Audio Outputs.

CobraNet.Input.1 through CobraNet.Input.16 – Depending on the model selected, you will have 8 or 16 CobraNet Outputs.

Notes

Please see *Selecting the Chip* for a discussion of this element and capabilities of the different models in the CS4961xx product line.

3.3.3 Delay

Purpose

The *Delay* element implements a multi-tap audio time delay. The number of output taps and the maximum length of the delay are configured by properties. The *Elements* window contains a number of combinations of delay time and tap count as well as a *Custom…* version. These are all the same Element with different pre-configurations of the properties for you convenience.

Properties

Delay unit declares the units used for specifying the length of the audio delay buffer, as well as the units used by the delay time controls. You may choose between millisecond, samples, meter, and feet. Time is converted to samples using the sample rate specified in *Project Properties* dialog (*File ⇨ Properties...*). Distances are converted to time assuming a speed of sound of 340 meters/second.

Max delay (unit) specifies the length of the audio delay buffer. The DSP has a limited amount of memory available for time delay (approximately 140ms @ 48 kHz if you do nothing else), so you should use the smallest amount necessary for you application. If you run out of delay memory, the compile will fail.

Tap count specifies the number of outputs from the delay buffer. It is much more memory efficient to use a single multi-tap delay rather than multiple single-tap delays when possible.

Controls

Master Bypass bypasses the delay buffer so all delay taps have no time delay introduced.

Master Mute mutes all outputs from the delay buffer.

Master Phase Invert inverts the phase at all outputs from the delay buffer.

Master Gain applies a gain to all outputs of the delay buffer.

Master Delay Time proportionally scales the delay time of each output from the delay buffer and has a range of 0% to 100% and it .

Tap Clip (one per tap) indicates that clipping has occurred at the respective output.

Tap Bypass (one per tap) bypasses the respective delay output so it has no time delay introduced. This is the same as setting the Delay Time to 0

Tap Mute (one per tap) mutes the respective delay output.

Tap Phase Invert (one per tap) inverts the phase at the respective delay output.

Tap Solo (one per tap) solos the respective delay output by muting all of the other delay outputs.

Tap Gain (one per tap) applies a gain at the respective delay output.

Tap Delay time (one per tap) controls the time delay introduced in the signal at the respective output. This value is scaled by the Master Delay Time control value to compute the actual time delay.

Inputs

input is the input to the audio time delay.

Outputs

output_1 through output_<Tap count> are the output taps from the audio time delay.

3.3.4 Compressor/Limiter

Purpose

This *Dynamics* device operates as either a compressor or a limiter. It attenuates the incoming audio signal based on how much the detected signal level exceeds the threshold level. It has a soft knee feature to smooth the transition between un-attenuated and attenuated.

Properties

Detector type - RMS or Peak. If the *RMS* detector is selected, the element behaves as a compressor. If a *Peak* detector is selected, the element behaves as a limiter.

Channel count is the number of program channels that are processed by the element. All program signals have exactly the same attenuation applied.

Sidechain count is the number of signals that are fed to the level detector used for the gain-reduction calculation. If this is zero, then all of the program channels are sent to the detector. If it is not zero, then only the sidechain signals are sent to the detector.

Controls

Bypass bypasses the element so it applies no attenuation to the program channels.

Threshold sets the threshold signal level above which attenuation will be applied to the program signals.

Ratio is the reciprocal of the slope of the response curve in the active region. If the input signal exceeds the threshold by *n* dB, the output signal will exceed the threshold by *n*/*Ratio*. A ratio of *1* causes the element to have no effect; a ratio of *100* has a severe effect, hardly allowing the output to exceed the threshold.

Signal indicates the level of the signal relative to the threshold. *Below* means the detected signal is less than (Threshold - Soft knee). *Above* means the detected signal is above (Threshold + Soft knee). *Knee* means the detected signal is within the range of the soft knee, on either side of the threshold.

Attack sets the time constant that determines how quickly the compressor/limiter can increase attenuation in response to detected signals over the threshold.

Release sets the time constant that determines how quickly the compressor/limiter will decrease attenuation when the detected signal falls below the threshold.

Gain reduction meter is the current gain reduction being applied to the program signals.

Soft Knee controls how far on either side of the threshold the soft knee reaches. The soft knee allows for a smooth transition between the inactive region and the active region.

Inputs

Input_1 through input_<Channel count> are the program inputs.

Input_<Channel count+1> through input_<Channel count+Sidechain count> are the optional sidechain inputs.

Outputs

output_1 through output_<Channel count> are the program outputs.

Notes

You may want to use a multi-channel compressor/limiter rather than multiple single-channel compressors/limiters for stereo and other-format signals because it is more efficient and, since it applies the same attenuation to all channels, it will not introduce inadvertent panning effects when one channel is attenuated differently from another.

You may want to filter the side chain signal to make it more or less sensitive to certain regions of the audio spectrum.

This element is not a true hard limiter, even when used with a peak detector, because of the finite attack time used for the gain reduction calculation. If it is critical that the output signal not exceed a threshold, you need to put a small delay in the program path (not in the sidechain path) to allow the detector and gain reduction logic to "look ahead" at the signal before it is processed.

3.3.5 Parametric Equalizer

Purpose

This element implements a single-band parametric equalizer useful for boosting or cutting certain frequencies of the audio spectrum.

Controls

Clip indicates that clipping has occurred inside the equalizer. Either reduce the Gain control or reduce the level of the signal entering the equalizer.

Bandwidth controls the width, in octaves, of the equalizer's response about the center frequency. The range of this control is 0.1 to 3.

Frequency controls the center frequency of the equalizer's response.

Gain controls the gain of the equalizer at the center frequency.

3.3.6 All-pass Filter

Purpose

This element implements a telescoped-topology all-pass filter which has a very long impulse response that is spectrally white and can be used as a reverberation element.

Properties

Max delay (ms) specifies the length of the delay buffer.

Controls

Delay controls the time delay in the feedback loop of the filter. The impulse response is many times longer than this, depending on the Gain.

Gain controls the gain in the feedback loop. As the Gain approaches 0 dB, the impulse response grows very long. As the gain approaches -100 dB, this element has no effect.

Clip indicates clipping has occurred in the filter.

3.3.7 Low-pass/High-pass Filter

Purpose

These filters are complements of each other and contain exactly the same properties and controls, so they are explained together.

The low-pass filter element passes frequencies below the corner frequency (the pass band) and attenuates frequencies above the corner frequency (the stop band).

The High-pass filter element passes frequencies above the corner frequency (the pass band) and attenuates frequencies below the corner frequency (the stop band).

Properties

Maximum slope specifies the maximum slope of the frequency response curve in the stop band. DSP cycles are consumed in proportion to the maximum slope, so choose the shallowest response that meets your needs.

Choose from 6 to 48 dB/Octave in 6 dB steps. The odd-order selections (6,18,30,42) consume as many MIPS as the next-higher, even-order (12,24,36,48) selection.

Controls

Bypass bypasses the filter, allowing all frequencies to pass.

Mute mutes the filter.

Invert inverts the polarity at the output of the filter.

Clip indicates that clipping has occurred within the filter.

Bessel Normalization setting only matters if *Type* is *Bessel*. Select the normalization rule from:

- *Phase Match*
- *Time Delay*
- *-3*
- *-6*
- *None*

Type selects the response characteristic of the filter from:

- *Variable Q*
- *Butterworth*
- *Linkwitz-riley*
- *Bessel*

Slope controls the slope of the filter in the stop band. No slope greater than the Maximum slope property is allowed.

Q - This control is only in effect if the Type control is set to variable *Q*. A high *Q* has a peaking effect at the corner frequency.

Frequency adjusts the filter's corner frequency.

Gain controls the gain of the filter in the pass band.

3.3.8 Pink Noise Generator

Purpose

This element generates pink noise, which contains equal energy per octave.

Controls

Mute mutes the output of the generator.

Level controls the RMS level of the output of the generator.

Notes

Multiple instances of this Element will produce apparently uncorrelated output. The generator is based on a pseudo-random noise (PRN) white noise generator passed through a pinking filter. The output of the pseudo-random generator is a sequence that repeats every 2^32 samples, which is almost 25 hours at a 48 kHz sample rate. Each copy of this element is initialized with a different starting point in the sequence.

3.3.9 Sine Wave Generator

Purpose

This element generates a pure sinusoidal tone.

Controls

Mute mutes the output of the generator.

Frequency controls the frequency of the generated tone.

Level adjusts the RMS level of the generated tone. The peak level is 3 dB higher.

3.3.10 White Noise Generator

Purpose

This element generates uniform pseudorandom white noise.

Controls

Mute mutes the output of the generator.

Level adjusts the RMS level of the generated noise. The peak level is 4.77 dB higher.

Notes

Multiple instances of this element will produce apparently uncorrelated output. The generator is a pseudo-random noise (PRN) white noise generator with a sequence length of a little longer than a day. Each copy of this element is initialized with a different starting point in the sequence.

3.3.11 Gain

Purpose

This element applies a gain to the signal. This is the most simple element.

Controls

Mute mutes the signal.

Clip indicates clipping has occurred in the element. Turn down the *Gain* control or reduce the level of the input signal.

Gain adjusts the gain to apply to the signal.

3.3.12 Level Meter

Purpose

This element provides readout of the peak and RMS level of a signal or multiple signals, with the ability to hold the maximum of each level (peak hold).

Properties

Input count specifies the number of channels of metering.

Deep mode allows metering very-low-level signals by expanding the range of the meter readout. The normal range is +35 dB to -35 dB. The range in deep mode is +35 dB to -100dB.

Controls

Meter Element displays the *RMS* value, the *maximum RMS* value, the *Peak* value and the *maximum Peak* value of the input signal. The RMS value is the top of the lower multi-colored region. The peak value is the upper multi-colored bar. The maximum RMS is the lower red horizontal bar and the maximum Peak is the upper red horizontal bar.

RMS Response Time sets the time constant of the RMS detector. Smaller values make the meter more responsive, larger values make it more accurate for steady state signals. When using DSP Conductor, there is no reason this should be faster than the poll rate, which is 100ms.

Peak Decay Time sets the time constant of the Peak detector.

Infinite Hold indicates that the maximum values should be held until this control is turned off. This is useful for latching the maximum level reached during a program, for example, or for capturing glitches during a long term test.

Hold Time sets the length of time that the maximum values are held before being reset. This is only applicable if *Infinite Hold* is disabled.

3.3.13 Signal Presence Detector

Purpose

This element is simply an indicator of whether there is signal present at its input.

Controls

Presence indicates the input signal is, or has been, above the threshold level.

Infinite Hold - If *Infinite Hold* is on, when the input signal exceeds the threshold, the *Presence* indicator will turn on and stay on until *Infinite Hold* is disabled.

Hold Time specifies how long the Presence indicator will stay on after the input signal drops below the threshold. This is only applicable if *Infinite Hold* is disabled.

Threshold sets the peak level above which the *Presence* indicator will become active.

3.3.14 Mixer

Purpose

This element provides a matrix mixer with a unique gain between each input and each output.

Properties

Input count determines the number of inputs to the mixer.

Output count determines the number of outputs from the mixer.

Controls

Crosspoint gains (Input count x Output count) defines the gain matrix which is arranged with 1 column per input and 1 row per output.

Input Mute <1 through Input count> mutes the input to the mixer so that particular signal does not appear at any output.

Input Inver <1 through Input count>t inverts the polarity of the signal at the input to the mixer.

Input Solo <1 through Input count> causes the input to the mixer to be soloed by muting all inputs that are not soloed. This means the solo is heard across all outputs; this is unlike the solo feature of many mixing boards where the solo only applies to specific outputs.

Input Gain <1 through Input count> applies a gain to the signal at the input to the mixer.

Output Clip <1 through Output count> indicates that clipping has occurred at that output.

Output Mute <1 through Output count> mutes the output from the mixer.

Output Invert <1 through Output count> inverts the polarity of the signal at the output of the mixer.

Output Gain <1 through Output count> applies a gain at the output of the mixer.

Notes

There are a number of different sizes of mixers available in the elements window as well as a custom version (*Custom...*). These are all the same element with different pre-configurations of the *Input count* and *Output count* properties. Any one can be turned into any other at a later time by editing the properties of the element.

3.3.15 Router

Purpose

The Router allows you to route any input signal to any output. Only one input may be assigned to an output at a time, meaning the router does not mix. One input may be assigned to any number of outputs at a time, meaning the router can fan out.

Properties

Input count specifies the number of inputs to the router.

Output count specifies the number of outputs from the router.

Controls

Output mute <1 through Output count> mutes the output from the router.

Input Select matrix - The matrix of select buttons is arranged as a column per output and a row per input. The buttons in each column are mutually exclusive; when you press one, the previously pressed button in that column becomes un-pressed.

Notes

This is a click-free router that ramps down the gain of the current input selection before ramping up the gain of the new selection.

The buttons within each column of the input select matrix are actually all copies of the same input-select control that have their properties configured to make them appear to act like radio buttons.

3.3.16 Multiplier

Purpose

Provides a method to multiply one or more signals by a control signal.

Properties

Channel count shows the number of program channels passing through the multiplier.

Inputs

input_1 through input_<Channel count> shows program inputs to the multiplier.

input_gain sets the gain for the control signal that is multiplied with each of the program signals. This is always the last input port.

Outputs

Output_1 through output_<Channel count> sets the program output from the multiplier.

Notes

This device is actually used internally within the *multi-channel dynamics* devices to apply the computed attenuation to the program signals.

3.3.17 Quadrature Generator

Purpose

Generates a pure tone with both Sine and Cosine outputs.

Controls

Mute mutes the outputs from the generator.

Frequency controls the frequency of the generated test tone.

Level adjusts the RMS level of the output of the generator. The peak level is 3 dB higher.

Outputs

output_sin is the sine output from the generator. It is identified with a small "S" on the face of the element.

output_cosine is the cosine output from the generator. It is identified with a small "C" on the face of element.

3.3.18 LMS Adaptive Filter

Purpose

This element implements a *Least Mean Square* adaptive filter. See a text on adaptive signal processing if you are unsure of what an adaptive filter is or how to use one.

Properties

Tap count specifies the number of taps in the FIR filter.

Weight readout determines if the control panel for the device contains readouts of the value of each coefficient of the FIR filter. You will want to turn this off if you use long filters.

Controls

Reset clears all of the weights and starts a new adaptation.

Freeze halts the adaptation process and freezes the weights. The element will act like a simple FIR filter using the adapted weights.

Step Size controls how fast the adaptation occurs. A higher value will adapt faster at the expense of noise. A lower value will adapt slower with less noise.

Weight readouts will appear only if the *Weight readout* property is checked. There is one control per weight in the FIR filter. The readout is the linear value of each weight.

Inputs

Input is the program input to the FIR filter.

Input Reference is the reference input to the filter. The LMS algorithm will adjust the weights of the FIR filter to make the output of the filter match the reference input a closely as possible.

Outputs

Output is the program input after it has been passed through the FIR filter.

Output Error is the difference between the filter output and the *Reference* input.

Custom Devices

This is the location where user-developed elements will appear in the *Elements* window. Writing your own elements requires authoring XML files that describe the element and

writing the processing code in assembly language. This requires an advanced development kit that is not part of this release.

3.3.19 Bump Panels and Labels

Purpose

Bump panels and Labels are simple graphic embellishments that can be inserted into your project for documentation or aesthetic reasons. They have no effect on the DSP resources your configuration will consume.

Properties

If you select one of these items and choose *Tools* ⇨ *Graphic Properties...* or type <Alt> + <Enter> you will see the *Graphic Properties* dialog that allows you to configure it. This dialog contains two tabs; the *Block* tab and the *Text* tab. The *Block* tab allows configuration of the shape, size, style, and color of the graphic. The *Text* tab allows customization of the text displayed by the graphic.

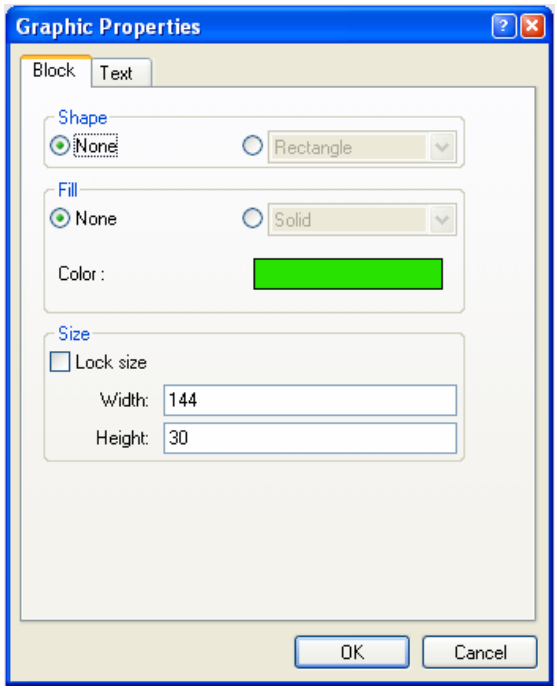

Figure 12. *Graphic Properties...* Dialog, *Block* Tab

The ***Shape*** field specifies the shape of the graphic. If *None* is selected then the graphic will be text only and the *Fill* section does not apply. The choices for *Shape* are *Rectangle*, *Rounded Rectangle*, *Ellipse*, *3D up* (a bump panel), or *3D down* (a dip panel).

The ***Fill*** indicates if the body of the graphic is an opaque fill or is transparent. If *None* is selected, the graphic is transparent. If fill is set to *Solid*, the body of the graphic is filled with the color shown in the *Color* rectangle. If fill is set to *Gradient*, the body of the graphic is filled with a smooth light-to-dark gradient of the color shown in the *Color* rectangle. Select a new color by clicking on the *Color* rectangle to display the color chooser.

Lock size means the graphic will be the size specified by the *Width* and *Height* fields. If *Lock size* is not checked, the graphic can be changed to the size specified in the *Width* and *Height* fields, but it can be adjusted in *Edit* mode if the cursor is clicked in one of the 8 resize hotspots at the center of each edge and in the corners.

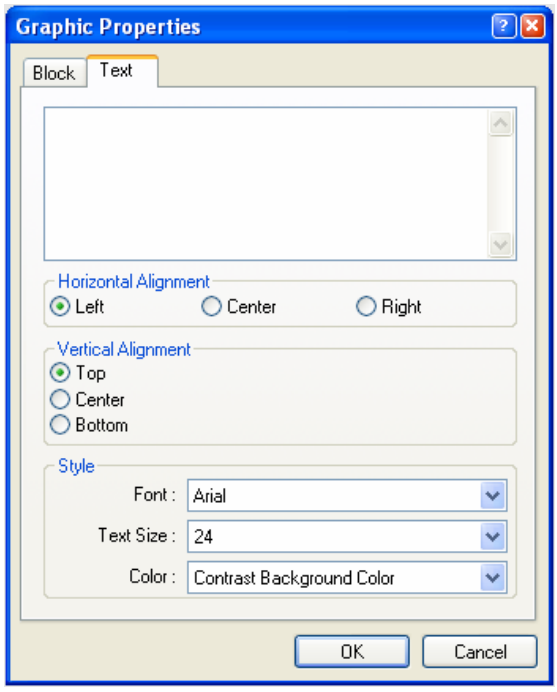

Figure 13. *Graphic Properties...* Dialog, *Text* Tab

The ***Text*** tab contains a field for editing the text that is displayed by the object. Text can also be changed in *Edit* mode by simply clicking on the graphic and typing the new text, but this field allows easier editing.

The ***Horizontal Alignment*** radio buttons specify how the text is aligned horizontally within the bounds of the graphic.

The ***Vertical Alignment*** radio buttons specify how the text is aligned vertically within the bounds of the graphic.

The ***Style*** section allows you to format the text displayed by the object.

The *Font* dropdown allows you to choose the font from among the installed fonts on your computer.

The *Text Size* dropdown allows you to choose the point size of the text.

The *Color* dropdown specifies the rule used to choose the color of the text. *Painted Color* means the text will be the color shown in the color rectangle on the *Block* tab. *Contrast Painted Color* means the text will be white if the color shown on the *Block* tab is dark and black if the color shown on the *Block* tab is light. *Background Color* makes the text the same color as the background of the *Workspace* page. *Contrast Background Color* makes the text white if the background of the *Workspace* page is dark and black if the background of the page is light.

3.3.20 Hierarchical/Control Block

Purpose

The *Hierarchical/Control Block* element allows you to use hierarchy as a method of organizing and managing the complexity of your project. It does not consume any DSP resources; it is simply a container for other Elements.

Properties

Input count sets the number of inputs to the *Hierarchical* block. Each input port will have a corresponding object along the left edge of the *Schematic* page that is similar to a flyoff that makes the signal wired to the input port available on the *Schematic* page.

Output count set the number of outputs from the *Hierarchical* block. Each output port will have a corresponding object along the right edge of the *Schematic* page that is similar to a flyoff that make outputs from the *Schematic* page available at the output port.

Create Control Page creates a page called *Controls* that is a peer to the *Schematic* page.

Create Schematic Page creates the *Schematic* page, which you must have if *Input count* or *Output count* is not zero.

Controls

The *Hierarchical* block contains no controls, but if you check the *Create Control Page* property, you can copy controls from any element on the *Schematic* page and paste them here to provide a simplified interface to your schematic.

Notes

Some common examples of when this might be useful include: an input processing block that contains a gate, an equalizer, and a compressor, or an output processing block that contains a delay, some high- and low-pass filters, and a limiter.

You can make your Hierarchical block available for use in other configurations by choosing *Tools ⇨ Define User Device...*