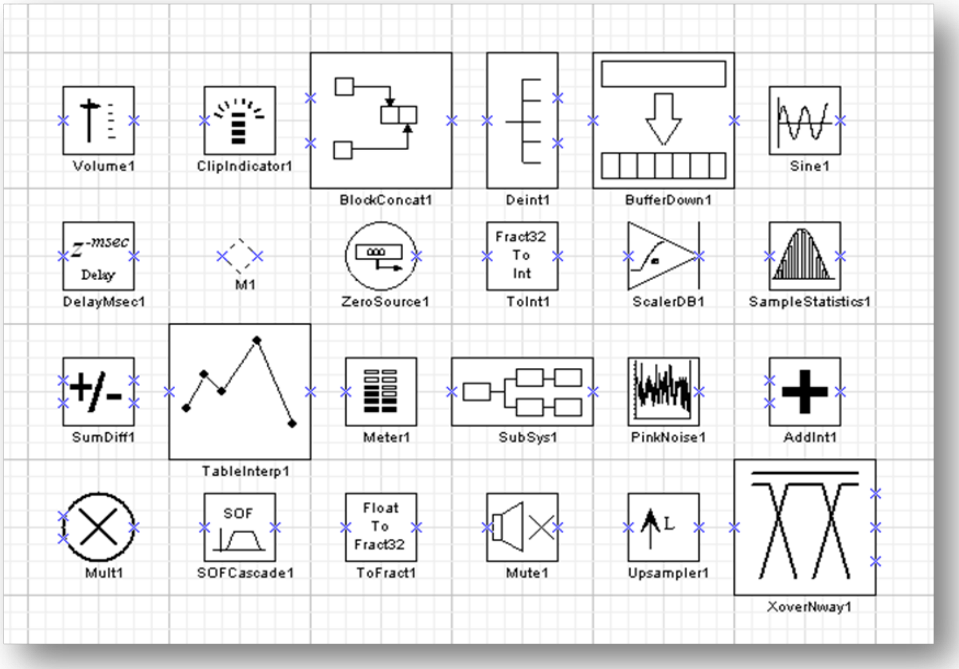
DSP CONCEPTS

Audio Weaver
Linux Integration Guide



Feb. 2017

Copyright Information

© 2006-2017 DSP Concepts, LLC., ALL RIGHTS RESERVED. This document may not be reproduced in any form without prior, express written consent from DSP Concepts, LLC.

Printed in the USA.

Disclaimer

DSP Concepts, LLC reserves the right to change this product without prior notice. Information furnished by DSP Concepts is believed to be accurate and reliable. However, no responsibility is assumed by DSP Concepts for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of DSP Concepts, Inc.

Change Log

Version	Date	Description	Author
01	2017.Feb.06	Original document + TOC + formatting changes	AN

Table of Contents

1. Introduction

The Audio Weaver command-line server (`AWE_command_line`) is an application that runs on Linux targets and listens to TCP/IP commands from the Audio Weaver Designer Windows application. DSP Concepts releases different versions of this application for various processor architectures such as ARM Cortex-A and Intel x86. `AWE_command_line` should theoretically run on most Linux distributions. As a baseline, the distributions used for testing at DSP Concepts are:

- Raspbian Jessie Lite (for Raspberry Pi platforms): https://www.raspberrypi.org/downloads/raspbian/
- eewiki Debian (for other platforms): https://www.eewiki.net/display/linuxonarm/Home

For those using a different distribution, there are a few requirements and optional configurations to be aware of:

- `AWE_command_line` requires ALSA libraries to be installed. The ALSA drivers on many audio SoC devices are not well supported. For this reason, it is recommended that an USB audio card (such as https://www.amazon.com/dp/B00NMXY2MO) be used for initial testing.

2. Running Audio Weaver Server on the Linux Target

A basic invocation of the AWE_command_line server looks like this:

```
sudo ./AWE_command_line
```

`sudo` is required to spawn some real-time threads.

The startup screen looks something like this:

```
debian@rpi2 AudioWeaver $ sudo ./AWE_command_line
Audio Weaver Server
Copyright (c) DSP Concepts 2006-2016


Target Information
Name: Linux
Version: 1.0.0.4
Processor type: CortexA7
Profile clock rate: 10 MHz
Sample rate: 44100 Hz
Basic block size: 32 samples
Supports object ID addressing: Yes
Supports SetID: Yes
Supports set/get merge: Yes
Communication buffer size: 264 words
Is floating point: Yes
Is FLASH supported: Yes
>
```

Typing `exit` in the above prompt will cause the application to exit.

`AWE_command_line` will detect audio devices on startup, and combine their channels (if possible). Further audio device configuration can be done by modifying the `AWE_CmdLine.ini` before restarting `AWE_command_line`.

Here's an example of a portion of `AWE_CmdLine.ini`:

```
[MergeInputDevices]
Device0=1`USB Audio Device(0)
Device1=1`USB Audio Device(1)
[MergeOutputDevices]
Device0=1`USB Audio Device(0)
Device1=1`USB Audio Device(1)
```

To disable `USB Audio Device(1)`, simply replace the `1`` with `0``:

```
[MergeInputDevices]
Device0=1`USB Audio Device(0)
Device1=0`USB Audio Device(1)
[MergeOutputDevices]
Device0=1`USB Audio Device(0)
Device1=0`USB Audio Device(1)
```

Note that the input section and the output section each need one or more enabled audio devices.

3. Connecting to Audio Weaver Designer

3.1 Audio Weaver Designer Version 4.x

The Audio Weaver Designer Windows application can be downloaded from here: http://www.dspconcepts.com/products/audio-weaver

(1) Set the Connection Type to **Remote Server**:
File->Global Preferences->[Server Connection->Connection Type->**Remote Server**]

(2) Set the Remote Connection String to **12001,15001,<ipaddr>**:
File->Global Preferences->[Server Connection->Connection Type->**12001,15001,<ipaddr>**]

where <ipaddr> is the IP address of the the remote Linux platform running the AWE_command_line_application. 12001 is currently not used. 15001 is the default socket port used to talk to AWE_command_line -- the defaults can be changed through a switch to AWE_command_line.

(3) Set the Source to **Line Input**:
File->File Properties>[PC Audio>Source->**Line Input**]
File input is not supported in the current Linux release

(4) Click the Reconnect to Server icon

(5) Click the Build and Run icon

3.2 Audio Weaver Designer Version 5.x

1. Set the Connection Type to **Remote Server**:
   File->Global Preferences->[Server Connection->Connection Type->**Remote Server**]

2. Set the Remote Connection String to **12001,15001,<ipaddr>**:
   File->Global Preferences->[Server Connection->Connection Type->**12001,15001,<ipaddr>**]
   where <ipaddr> is the IP address of the the remote Linux platform running the AWE_command_line_application. 12001 is currently not used. 15001 is the default socket port used to talk to AWE_command_line -- the defaults can be changed through a switch to AWE_command_line.

3. Set the Source to **Line Input**:
   Layout->Layout Properties>[PC Audio>Source->**Line Input**]
   File input is not supported in the current Linux release

4. Click the Reconnect to Server icon

5. Click the Build and Run icon

4. Profiling an Audio Weaver Design

For profiling, standard utilities like `htop` and `pidutils` can be used. AWE_command_line spawns a number of threads. The most compute-intensive is the `AudioThread` (scheduled as a real-time thread).

5. Running a Script From the Command Line

When a design is debugged, the connection to the Audio Weaver Designer is no longer necessary. An Audio Weaver script can be invoked directly from the command line:

```
sudo ./AWE_command_line -script:awe-scripts/AutoAmpExample.aws -nocmd
```

6. Troubleshooting

The `AWE_command_line` application writes logging information to `/tmp/awelog.txt`. This can be very useful for debugging runtime issues.