Advanced Gtk+ Sequencer

Developer's Book

Copyright (C) Joël Krähemann. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Contents

List of Tables

Dedication

This book is dedicated to my friend.

Foreword

I began to code with C in spring 2002 and hadn't much programming skills, yet. You may ask me why the C programming language? Well, my friend who was already a convient free software user and hacker recomended me it. He told me that C is a standard on Unix like operating systems so it would be a good choice.

After started with language basics and several discussions with my friend about pointers he advised me of Gtk+. While I was doing my first steps in GUI programming with C, I was sure to extensively use it and became a persuaded free software user and programmer.

A year later I really understood the object orientated matter of GObject and how to write objects and widgets myself. C wasn't like Java where you just couldn't implement no classes just everything was a class or at least a method.

First output with AGS happend via Open Sound System device drivers but the entire application lacked of a thread safe concept. But for now you may write tasks.

Be part of the fun.

Chapter 1

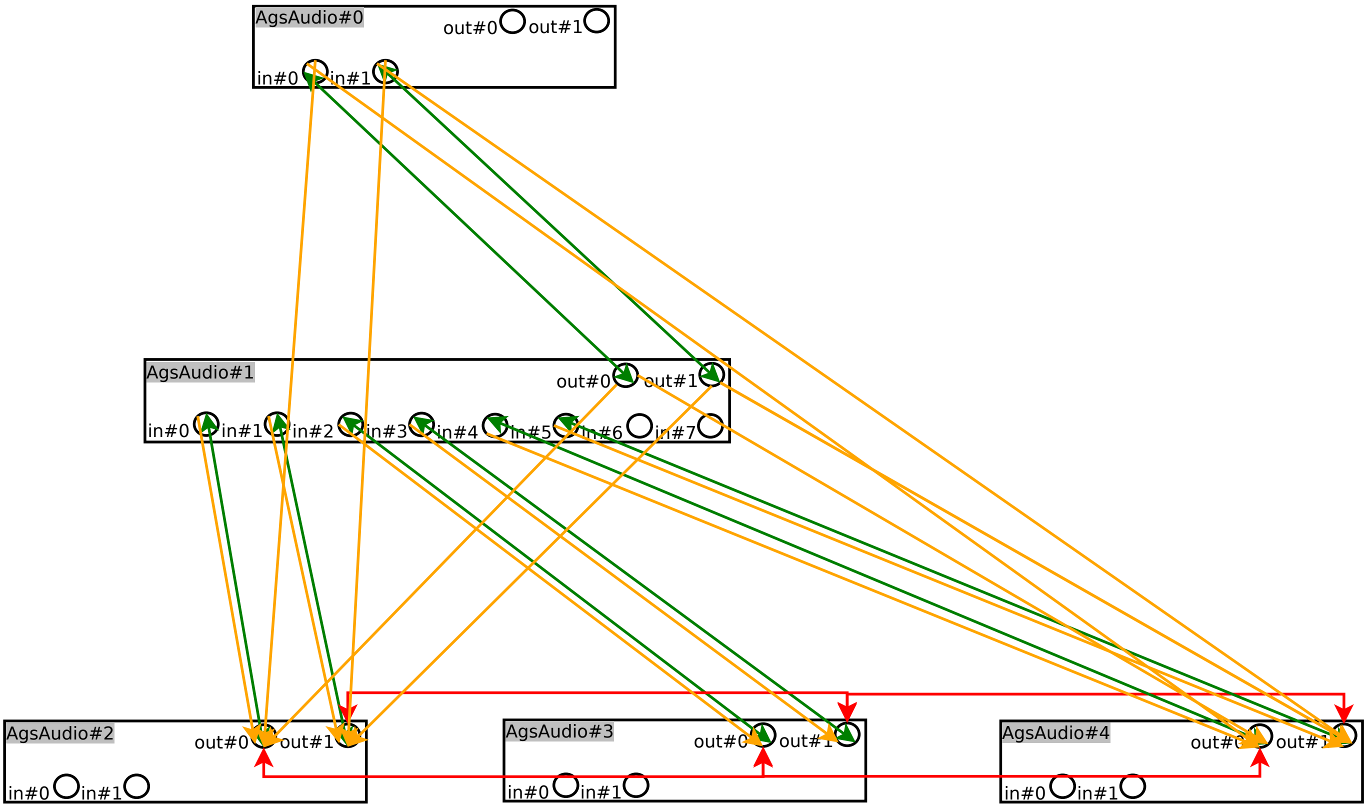
Linking

In this section you get some knowledge about ags internals. Here you get an overview of the audio layer. All code related to it is located in subdirectory <ags/audio>. Linking AgsChannel is a quiet complex thing but If you wish to do so you can just call ags_channel_link() and this will the especially covered here.

1.1 Overview

AgsAudio, AgsChannel and AgsRecycling are involved in linking. When talking about linking we should view AgsChannel objects as networked and therefore exists an additional nested network of AgsRecycling objects.

The AgsAudio object gives clarification about how AgsChannel has to be accessed either synchronously or asynchronously. Further it tells us whether AgsOutput or AgsInput has a new audio stream which causes in conjunction a dedicated AgsRecycling associated with the appropriate AgsChannel.



- green:

object	flags
Audio#0	AGS_AUDIO_SYNC | AGS_AUDIO_OUTPUT_HAS_RECYCLING
Audio#1	AGS_AUDIO_ASYNC
Audio#2	AGS_AUDIO_ASYNC | AGS_AUDIO_OUTPUT_HAS_RECYCLING
Audio#3	AGS_AUDIO_ASYNC | AGS_AUDIO_OUTPUT_HAS_RECYCLING
Audio#4	AGS_AUDIO_ASYNC | AGS_AUDIO_OUTPUT_HAS_RECYCLING

Table 1.1: AGS network layer table

  - Bidirectional linked AgsChannel to an other AgsChannel.
  - Generally you link an AgsOutput to an AgsInput.

- red:
  - Bidirectional linked AgsRecycling to an other AgsRecycling on the same level.
  - They are linked across AgsAudio objects.
  - Same level means the linked AgsRecycling are all child nodes of a parent AgsRecycling.

- yellow:
  - Unidirectional linked AgsRecycling to an AgsChannel.
  - First AgsRecycling of an AgsOutput and last AgsRecycling of an (other) AgsOutput are linked to an AgsChannel.

1.2 Limitations

- You may not create any kind of loops.
- You may not set AGS_AUDIO_INPUT_HAS_RECYCLING without setting AGS_AUDIO_OUTPUT_HAS_RECYCLING flag.

1.3 Hands-On

There may be two ways how you can link AgsChannel objects.

Example 1.1 Prerequisites

```
AgsTaskThread *task_thread;
AgsDevout *devout;
AgsAudio *master_audio, *slave_audio;
AgsLinkChannel *linkChannel;
GError *error;

/* some pseudo code */
devout = AGS_WINDOW(gtk_widget_get_toblevel(widget))->devout;
task_thread = AGS_AUDIO_LOOP(AGS_MAIN(devout->ags_main)->main_loop)->task_thread;

/* create AgsAudio objects */
master_audio = (AgsAudio *) g_object_new(AGS_TYPE_AUDIO,
           "devout", devout,
           NULL);
```

```
slave_audio = (AgsAudio *) g_object_new(AGS_TYPE_AUDIO,
          "devout", devout,
          NULL);

/* assign AgsAudioSignal objects to master_audio and slave_audio */
ags_audio_set_flags(master_audio,
        AGS_AUDIO_OUTPUT_HAS_RECYCLING);
ags_audio_set_flags(slave_audio,
        (AGS_AUDIO_ASYNC | AGS_AUDIO_OUTPUT_HAS_RECYCLING | AGS_AUDIO_INPUT_HAS_RECYCLING)) ↩
            ;

/* create AgsChannel objects within master_audio and slave_audio */
ags_audio_set_audio_channels(master_audio, 2);
ags_audio_set_pads(master_audio, AGS_TYPE_OUTPUT, 1);
ags_audio_set_pads(master_audio, AGS_TYPE_INPUT, 1);

ags_audio_set_audio_channels(slave_audio, 2);
ags_audio_set_pads(slave_audio, AGS_TYPE_OUTPUT, 1);
ags_audio_set_pads(slave_audio, AGS_TYPE_INPUT, 8);
```

Assumed you know really what you do, you may be interested in following code.

Example 1.2 Thread-Unsafe way

```
/* link master_audio's input with slave_audio's output */
ags_channel_set_link(ags_channel_nth(master_audio->input, 0),
         ags_channel_nth(slave_audio->output, 0),
         &error);


ags_channel_set_link(ags_channel_nth(master_audio->input, 1),
         ags_channel_nth(slave_audio->output, 1),
         &error);
```

But generally you wish to create an AgsTask object and let it to link the AgsChannel for you.

Example 1.3 Multithread-Safe way

```
/* creating AgsLink task and add it to AgsDevout */
link_channel = ags_link_channel_new(ags_channel_nth(master_audio->input, 0),
            ags_channel_nth(slave_audio->output, 0));
ags_task_thread_append_task(task_thread, link_channel);

link_channel = ags_link_channel_new(ags_channel_nth(master_audio->input, 1),
            ags_channel_nth(slave_audio->output, 1));
ags_task_thread_append_task(task_thread, link_channel);
```

Chapter 2

Effects

You may directly inherit by <ags/audio/ags_recall.h> to do some wicked stuff. But generally you should inherit by these subclasses of AgsRecall:

- <ags/audio/ags_recall_audio.h>
- <ags/audio/ags_recall_audio_run.h>
- <ags/audio/ags_recall_channel.h>
- <ags/audio/ags_recall_channel_run.h>
- <ags/audio/ags_recall_recycling.h>
- <ags/audio/ags_recall_audio_signal.h>

You probably wish to have different context for fields of an effect, that's what these objects take on. But before we cover them in detail, we take a look at the lifecycle an effect must accomplish.

2.1 Play/recall context

Don't mix this context up with static/runtime context we talked before. The AgsRecall may have two faces or may be just one for play context.

The play context will be called in case the higher level of AgsRecycling will output to a device e.g. the soundcard and no further processing will be done.

The recall context means that the AgsRecall will pass one or more cycles of copying or sequencing. This design is intended to make ags as modular and reusable over different use cases as possible. Practically it should be possible to chain up several sequencers.

2.2 Hands-On instantiating an effect

After you got an overview of the basic lifecycle of an effect it's time to create an effect. In this guide we will cover instatiating an effect by using the echo effect. In the following chapter we'll take a look inside the echo effect.

2.2.1 AgsRecallContainer

AgsRecallContainer isn't a recall itself but you can use it to retrieve a different context.

Example 2.1 Creating AgsRecallContainer

```
AgsMachine *machine;
AgsDevout *devout;
AgsAudio *audio;
AgsChannel *channel;
AgsRecallContainer *echo_container;

/* some pseudo code */
machine = (AgsMachine *) gtk_widget_get_ancestor(widget,
             AGS_TYPE_MACHINE);

/* retrieve some essencial objects */
audio = machine->audio;
devout = audio->devout;

/* create the container */
recall_container = (AgsRecallContainer *) g_object_new(AGS_TYPE_RECALL_CONTAINER,
                   NULL);
```

2.2.2 AgsRecallAudio context

This is a context you want to use for fields applicable to the entire AgsAudio object.

Example 2.2 Creating AgsEchoAudio

```
AgsEchoAudio *echo_audio;

echo_audio = (AgsEchoAudio *) g_object_new(AGS_TYPE_ECHO_AUDIO,
             "audio", audio,
             "container", echo_container,
             NULL);

AGS_RECALL(echo_audio)->flags = AGS_RECALL_TEMPLATE;
```

2.2.3 AgsRecallChannel context

This context you can use for fields applicable to the AgsChannel you want to modify.

Example 2.3 Creating AgsEchoChannel

```
AgsEchoChannel *echo_channel;

echo_channel = (AgsEchoChannel *) g_object_new(AGS_TYPE_ECHO_CHANNEL,
                 "channel", channel,
                 "container", echo_container,
                 "recall_audio", echo_audio,
                 "delay", (devout->frequency * (60 / devout->bpm) / 4),
                 "repeat", 3,
                 "fade", -0.25,
                 "dry", 0.5,
                 NULL);

AGS_RECALL(echo_channel)->flags = AGS_RECALL_TEMPLATE;
```

2.2.4 AgsRecallAudioRun context

The AgsRecallAudioRun class will be duplicated for a parental running AgsChannel. There may be several AgsChannel objects as parental owning a run.

Example 2.4 Creating AgsEchoAudioRun

```
echo_audio_run = (AgsEchoAudioRun *) g_object_new(AGS_TYPE_ECHO_AUDIO_RUN,
              "audio", audio,
              "container", echo_container,
              "recall_audio", echo_audio,
              NULL);

AGS_RECALL(echo_audio_run)->flags = AGS_RECALL_TEMPLATE;
```

2.2.5 AgsRecallChannelRun context

The AgsRecallChannelRun behaves like an AgsRecallAudioRun but is designated to an AgsChannel object.

Example 2.5 Creating AgsEchoChannelRun

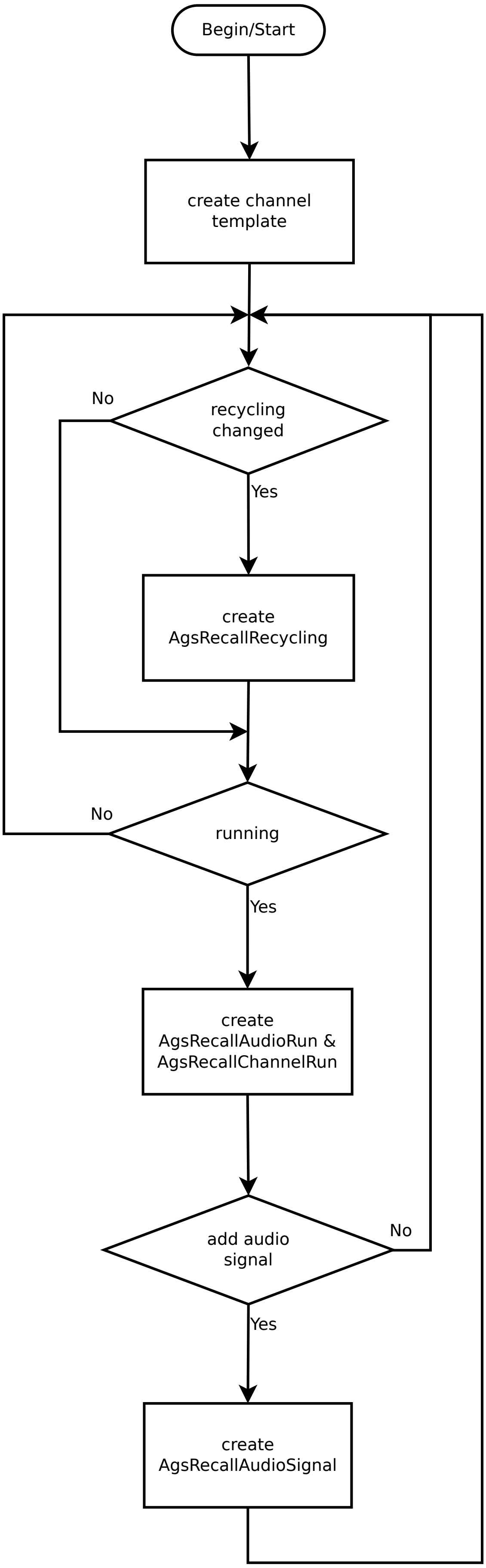
```
AgsEchoChannelRun *echo_channel_run;

echo_channel_run = (AgsEchoChannelRun *) g_object_new(AGS_TYPE_ECHO_CHANNEL_RUN,
                  "channel", channel
                  "container", echo_container,
                  "recall_audio", echo_audio,
                  "recall_channel", echo_channel,
                  "recall_audio_run", echo_audio_run,
                  NULL);

AGS_RECALL(echo_channel_run)->flags = AGS_RECALL_TEMPLATE;
```

2.3 The basic lifecycle of an effect

In this section I'll introduce the keyword run which can be understood as a playing instance. But I rather talk about run because it's not guaranted that the recall outputs directly to a device.


Begin/Start
create channel template
recycling changed
No
Yes
create AgsRecallRecycling
running
No
Yes
create AgsRecallAudioRun & AgsRecallChannelRun
add audio signal
No
Yes
create AgsRecallAudioSignal

The implemented effect as a subclass of AgsRecall resides as template on the appropriate AgsAudio or AgsChannel.

When recycling changes on input, new AgsRecallRecycling will be added. This class function may be of relevancy:

- channel_class->recycling_changed

As a new run occures the AgsRecallAudioRun and AgsRecallChannelRun will be duplicated, dependencies resolved, state initialized and enter the play loop hierarchy. These class functions will be called on the recall:

- channel_class->duplicate
  - This function will be called on the template object to instantiate the the object which will pass further processing.

Further processing:

- recall_class->resolve_dependencies
  - The recall may want to depend on a other recall (eg. a counter) and may ignore following calls while rather do processing on an event of the dependency.
- recall_class->run_init_pre, recall_class->run_init_inter & recall_class->run_init_post
  - Will be called only once for the run reffering to dedicated AgsGroupId.
- recall_class->run_pre, recall_class->run_inter & recall_class->run_post
  - Will be called for each cycle of a run reffering to AgsGroupId.
  - There may be more than one AgsGroupId for a template i.e. there can exist more than one run at the very same time.

As soon as an add_audio_signal event will be emitted on an AgsRecycling, the AgsRecallAudioSignal subclass will be instantiated which performs audio stream manipulation. These class functions will be called on the recall:

- recall_class->run_init_pre, recall_class->run_init_inter & recall_class->run_init_post
- recall_class->run_pre, recall_class->run_inter & recall_class->run_post

When you're done with processing call:

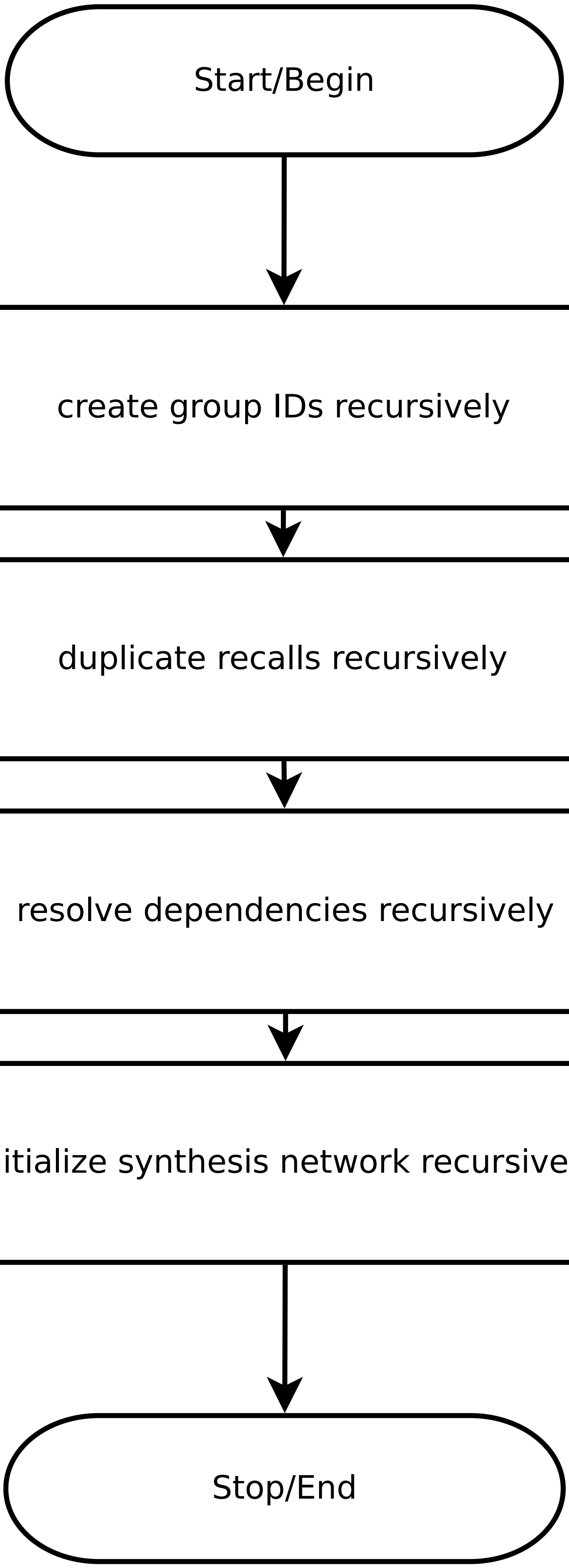
- recall_class->done

2.4 A closer look at effects

First we look at recall initialization and afterwards at processing audio data with run etapes.

2.4.1 Recursive initialization

Initilization recusivly is done by calling ags_channel_recursive_play_init().


Start/Begin
create group IDs recursively
duplicate recalls recursively
resolve dependencies recursively
initialize synthesis network recursively
Stop/End

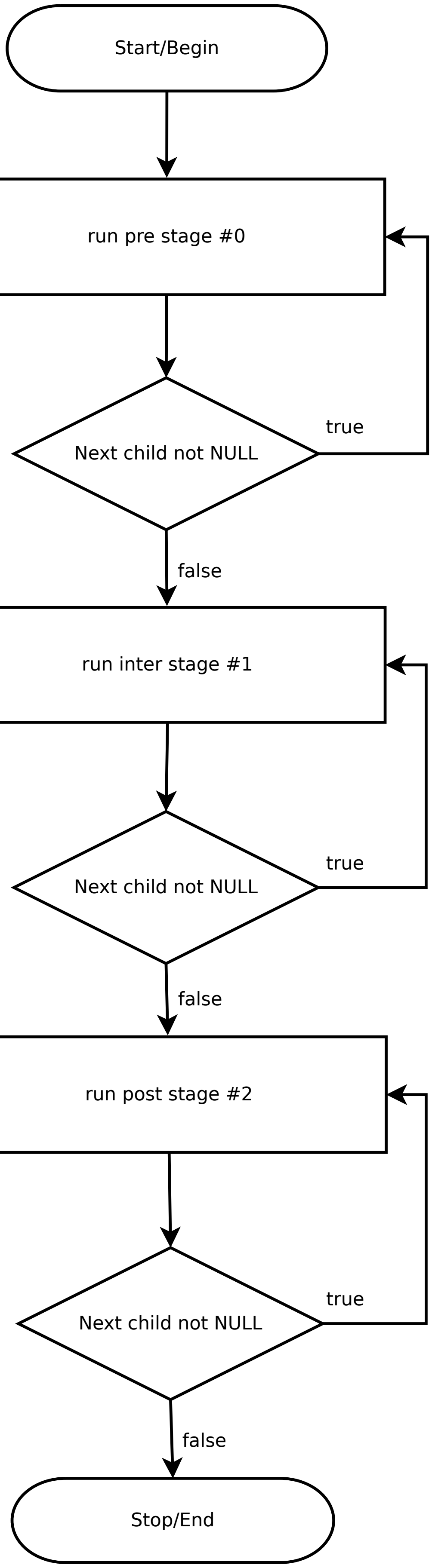
The initialization occurs in one part.

The following procedures needs to be passed:

- Allocating group id and recall countainer.
  - Acts as a unique identifier for runtime.
- Duplicating recalls and pass default properties.
  - Set up functional runtime objects.
- Resolve dependencies and inject.
  - Dynamic connect of objects.
- Initialized tree recursivly as entire initialization stage.
  - The created context is now ready to be processed ...

2.4.2 The different run stages

As mentioned before audio processing will be done within an AgsRecallAudioSignal subclass.


Start/Begin
run pre stage #0
Next child not NULL
true
false
run inter stage #1
Next child not NULL
true
false
run post stage #2
Next child not NULL
true
false
Stop/End

This object is running the stages as illustrated and you may have noticed it's recursive. The run phase is divided in 3 stages with dedicated pass within tree.

run_pre() is the very first etape. Its purpose is mainly of allocating or preparing buffers.

run_inter() is the second phase and acts as effect processor.

run_post() is the third phase and is usually used for doing clean-up.

These recalls implementing those functions generally inherit by AGS_TYPE_RECALL_AUDIO_RUN, AGS_TYPE_RECALL_CHAN or AGS_TYPE_RECALL_AUDIO_SIGNAL.

Run time values are written in real time using atomic operations. This is done by AgsPort and inherit by AGS_TYPE_RECALL_AUDIO or AGS_TYPE_RECALL_CHANNEL.

AgsRecallContainer matches related recalls. This means you pack recalls of same XML type eg. ags-echo to be grouped, litteraly:

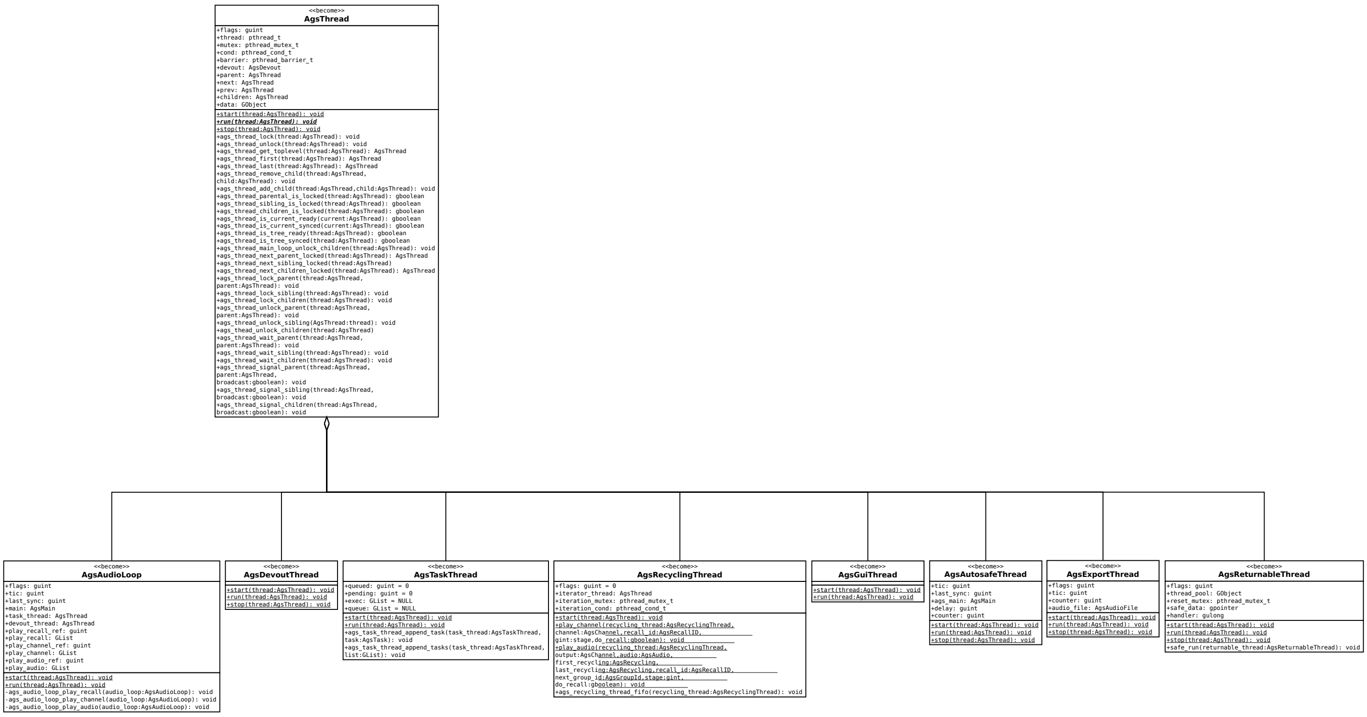
- ags-echo-channel
- ags-echo-channel-run
- ags-echo-recycling
- ags-echo-audio-signal

Chapter 3

Multi-Threaded Tree

Ags is a able to run in a super threaded fashion. AgsThread object is therefore the most basic thread which will let run your very own thread within the tree. For locking the tree you must wait until every mutex is acquired.

3.1 Well-Known Threads


<<become>>
AgsThread
+flags: guint
+thread: pthread_t
+mutex: pthread_mutex_t
+cond: pthread_cond_t
+barrier: pthread_barrier_t
+devout: AgsDevout
+parent: AgsThread
+next: AgsThread
+prev: AgsThread
+children: AgsThread
+data: GObject
+start(thread:AgsThread): void
+run(thread:AgsThread): void
+stop(thread:AgsThread): void
+ags_thread_lock(thread:AgsThread): void
+ags_thread_unlock(thread:AgsThread): void
+ags_thread_get_toplevel(thread:AgsThread): AgsThread
+ags_thread_first(thread:AgsThread): AgsThread
+ags_thread_last(thread:AgsThread): AgsThread
+ags_thread_remove_child(thread:AgsThread,
child:AgsThread): void
+ags_thread_add_child(thread:AgsThread,child:AgsThread): void
+ags_thread_parental_is_locked(thread:AgsThread): gboolean
+ags_thread_sibling_is_locked(thread:AgsThread): gboolean
+ags_thread_children_is_locked(thread:AgsThread): gboolean
+ags_thread_is_current_ready(current:AgsThread): gboolean
+ags_thread_is_current_synced(current:AgsThread): gboolean
+ags_thread_is_tree_ready(thread:AgsThread): gboolean
+ags_thread_is_tree_synced(thread:AgsThread): gboolean
+ags_thread_main_loop_unlock_children(thread:AgsThread): void
+ags_thread_next_parent_locked(thread:AgsThread): AgsThread
+ags_thread_next_sibling_locked(thread:AgsThread)
+ags_thread_next_children_locked(thread:AgsThread): AgsThread
+ags_thread_lock_parent(thread:AgsThread,
parent:AgsThread): void
+ags_thread_lock_sibling(thread:AgsThread): void
+ags_thread_lock_children(thread:AgsThread): void
+ags_thread_unlock_parent(thread:AgsThread,
parent:AgsThread): void
+ags_thread_unlock_sibling(AgsThread:thread): void
+ags_thead_unlock_children(thread:AgsThread)
+ags_thread_wait_parent(thread:AgsThread,
parent:AgsThread): void
+ags_thread_wait_sibling(thread:AgsThread): void
+ags_thread_wait_children(thread:AgsThread): void
+ags_thread_signal_parent(thread:AgsThread,
parent:AgsThread,
broadcast:gboolean): void
+ags_thread_signal_sibling(thread:AgsThread,
broadcast:gboolean): void
+ags_thread_signal_children(thread:AgsThread,
broadcast:gboolean): void
<<become>>
AgsAudioLoop
+flags: guint
+tic: guint
+last_sync: guint
+main: AgsMain
+task_thread: AgsThread
+devout_thread: AgsThread
+play_recall_ref: guint
+play_recall: GList
+play_channel_ref: guint
+play_channel: GList
+play_audio_ref: guint
+play_audio: GList
+start(thread:AgsThread): void
+run(thread:AgsThread): void
-ags_audio_loop_play_recall(audio_loop:AgsAudioLoop): void
-ags_audio_loop_play_channel(audio_loop:AgsAudioLoop): void
-ags_audio_loop_play_audio(audio_loop:AgsAudioLoop): void
<<become>>
AgsDevoutThread
+start(thread:AgsThread): void
+run(thread:AgsThread): void
+stop(thread:AgsThread): void
<<become>>
AgsTaskThread
+queued: guint = 0
+pending: guint = 0
+exec: GList = NULL
+queue: GList = NULL
+start(thread:AgsThread): void
+run(thread:AgsThread): void
+ags_task_thread_append_task(task_thread:AgsTaskThread,
task:AgsTask): void
+ags_task_thread_append_tasks(task_thread:AgsTaskThread,
list:GList): void
<<become>>
AgsRecyclingThread
+flags: guint = 0
+iterator_thread: AgsThread
+iteration_mutex: pthread_mutex_t
+iteration_cond: pthread_cond_t
+start(thread:AgsThread): void
+play_channel(recycling_thread:AgsRecyclingThread,
channel:AgsChannel,recall_id:AgsRecallID,
gint:stage,do_recall:gboolean): void
+play_audio(recycling_thread:AgsRecyclingThread,
output:AgsChannel,audio:AgsAudio,
first_recycling:AgsRecycling,
last_recycling:AgsRecycling,recall_id:AgsRecallID,
next_group_id:AgsGroupId,stage:gint,
do_recall:gboolean): void
+ags_recycling_thread_fifo(recycling_thread:AgsRecyclingThread): void
<<become>>
AgsGuiThread
+start(thread:AgsThread): void
+run(thread:AgsThread): void
<<become>>
AgsAutosafeThread
+tic: guint
+last_sync: guint
+ags_main: AgsMain
+delay: guint
+counter: guint
+start(thread:AgsThread): void
+run(thread:AgsThread): void
+stop(thread:AgsThread): void
<<become>>
AgsExportThread
+flags: guint
+tic: guint
+counter: guint
+audio_file: AgsAudioFile
+start(thread:AgsThread): void
+run(thread:AgsThread): void
+stop(thread:AgsThread): void
<<become>>
AgsReturnableThread
+flags: guint
+thread_pool: GObject
+reset_mutex: pthread_mutex_t
+safe_data: gpointer
+handler: gulong
+start(thread:AgsThread): void
+run(thread:AgsThread): void
+stop(thread:AgsThread): void
+safe_run(returnable_thread:AgsReturnableThread): void

3.2 Syncing With AgsAudioLoop

Every child thread of AgsAudioLoop will be synced as long you don't write your very own start class function of AgsThread. The thread is running as long as AGS_THREAD_RUNNING is set as flag. Put your own thread routine by overwriting class function run of AgsThreadClass.

3.3 Interfacing Audio-Layer with GUI-Layer and vice versa

At topmost there’s AgsAudioLoop whereby AgsTaskThread guarantees no concurrent memory access. To run tasks instantiate subclass of AgsTask and add to queue by calling:

- ags_task_thread_append_task()
- ags_task_thread_append_tasks()

3.4 Thread-Pool

The thread pool can be used to pull threads of AGS_TYPE_RETURNABLE_THREAD. It should return as soon as possible because of potential hang-up. It is recommended to stop thread after very first run.

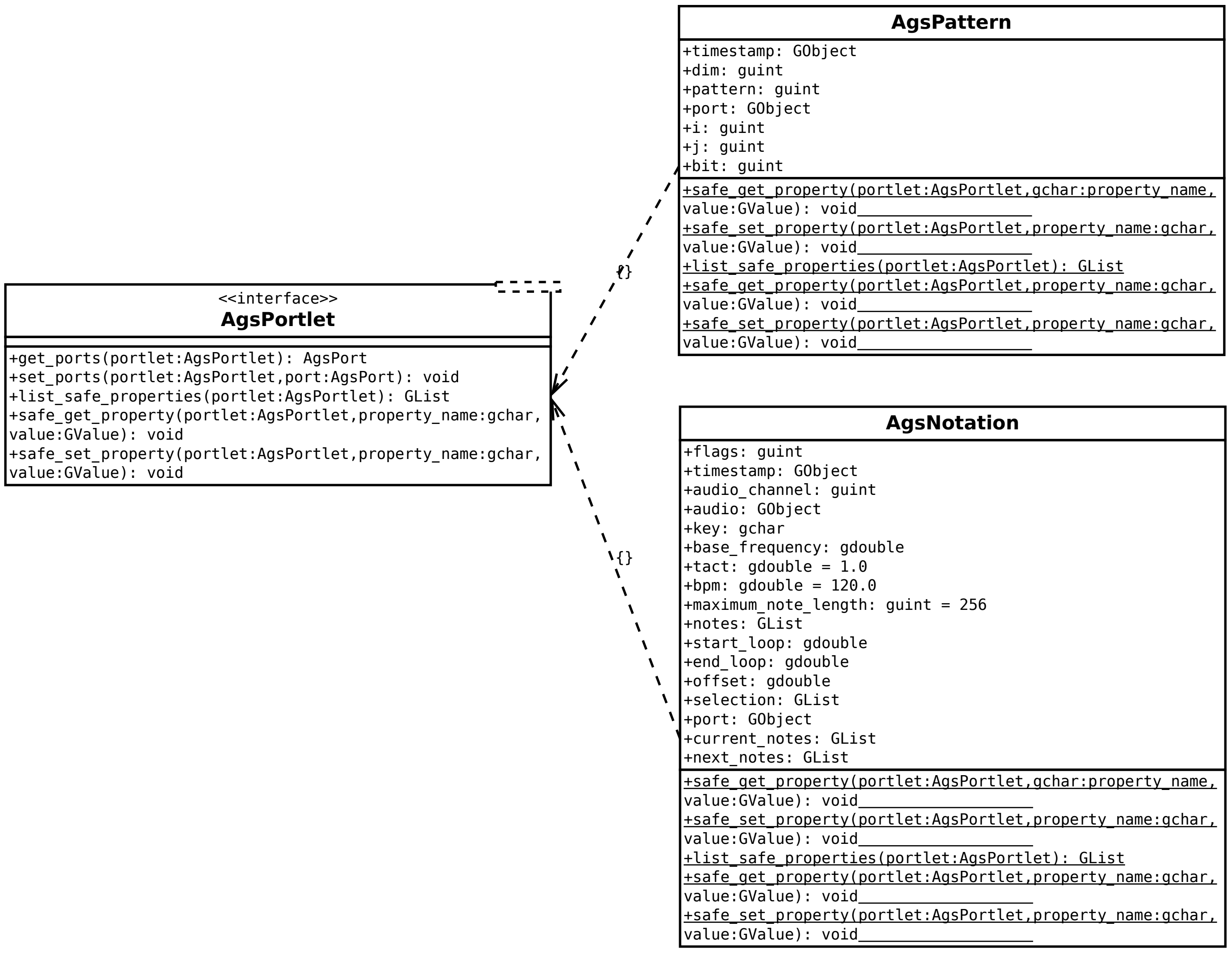
Chapter 4

Port to safe read/write

The AgsPort object may contain primitive types as well objects. The read/write operations are generally done by atomic operations. There exists interfacing AgsTaskThread functions but not used, for now. Representing objects should implement AgsPortletInterface.

4.1 Portlet interface safe read/write properties

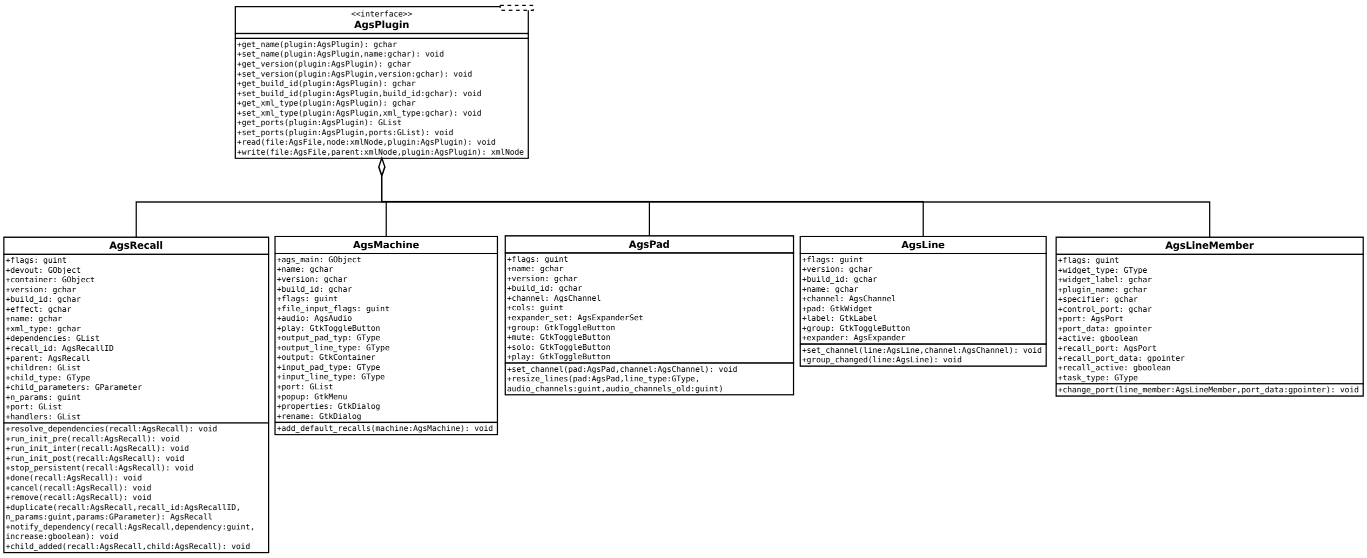
The AgsPortletInterface is used where a GObject is represented by the assigned port. The implemententing class is responsible for thread safe get/set properties. Most common way is using mutices.


AgsPattern
+timestamp: GObject
+dim: guint
+pattern: guint
+port: GObject
+i: guint
+j: guint
+bit: guint
+safe_get_property(portlet:AgsPortlet,gchar:property_name,
value:GValue): void
+safe_set_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
+list_safe_properties(portlet:AgsPortlet): GList
+safe_get_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
+safe_set_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
{}
<<interface>>
AgsPortlet
+get_ports(portlet:AgsPortlet): AgsPort
+set_ports(portlet:AgsPortlet,port:AgsPort): void
+list_safe_properties(portlet:AgsPortlet): GList
+safe_get_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
+safe_set_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
{}
AgsNotation
+flags: guint
+timestamp: GObject
+audio_channel: guint
+audio: GObject
+key: gchar
+base_frequency: gdouble
+tact: gdouble = 1.0
+bpm: gdouble = 120.0
+maximum_note_length: guint = 256
+notes: GList
+start_loop: gdouble
+end_loop: gdouble
+offset: gdouble
+selection: GList
+port: GObject
+current_notes: GList
+next_notes: GList
+safe_get_property(portlet:AgsPortlet,gchar:property_name,
value:GValue): void
+safe_set_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
+list_safe_properties(portlet:AgsPortlet): GList
+safe_get_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void
+safe_set_property(portlet:AgsPortlet,property_name:gchar,
value:GValue): void

Chapter 5

Plugin interface to do abstraction

The AgsPlugin interface defines some elementary functions like get_name() or set_name() and therefor get_version(), set_version(), get_build_id() and set_build_id(). Further it contains functions used to persist it using XML. This would be get_xml_type(), set_xml_type(), read() and write(). And get_ports() and set_ports() to allow thread safe communication between the different layers of Advanced Gtk+ Sequencer but to be said those functions accomplish for persisting and restoring.



<<interface>>
AgsPlugin
+get_name(plugin:AgsPlugin): gchar
+set_name(plugin:AgsPlugin,name:gchar): void
+get_version(plugin:AgsPlugin): gchar
+set_version(plugin:AgsPlugin,version:gchar): void
+get_build_id(plugin:AgsPlugin): gchar
+set_build_id(plugin:AgsPlugin,build_id:gchar): void
+get_xml_type(plugin:AgsPlugin): gchar
+set_xml_type(plugin:AgsPlugin,xml_type:gchar): void
+get_ports(plugin:AgsPlugin): GList
+set_ports(plugin:AgsPlugin,ports:GList): void
+read(file:AgsFile,node:xmlNode,plugin:AgsPlugin): void
+write(file:AgsFile,parent:xmlNode,plugin:AgsPlugin): xmlNode
AgsRecall
+flags: guint
+devout: GObject
+container: GObject
+version: gchar
+build_id: gchar
+effect: gchar
+name: gchar
+xml_type: gchar
+dependencies: GList
+recall_id: AgsRecallID
+parent: AgsRecall
+children: GList
+child_type: GType
+child_parameters: GParameter
+n_params: guint
+port: GList
+handlers: GList
+resolve_dependencies(recall:AgsRecall): void
+run_init_pre(recall:AgsRecall): void
+run_init_inter(recall:AgsRecall): void
+run_init_post(recall:AgsRecall): void
+stop_persistent(recall:AgsRecall): void
+done(recall:AgsRecall): void
+cancel(recall:AgsRecall): void
+remove(recall:AgsRecall): void
+duplicate(recall:AgsRecall,recall_id:AgsRecallID,
n_params:guint,params:GParameter): AgsRecall
+notify_dependency(recall:AgsRecall,dependency:guint,
increase:gboolean): void
+child_added(recall:AgsRecall,child:AgsRecall): void
AgsMachine
+ags_main: GObject
+name: gchar
+version: gchar
+build_id: gchar
+flags: guint
+file_input_flags: guint
+audio: AgsAudio
+play: GtkToggleButton
+output_pad_typ: GType
+output_line_type: GType
+output: GtkContainer
+input_pad_type: GType
+input_line_type: GType
+port: GList
+popup: GtkMenu
+properties: GtkDialog
+rename: GtkDialog
+add_default_recalls(machine:AgsMachine): void
AgsPad
+flags: guint
+name: gchar
+version: gchar
+build_id: gchar
+channel: AgsChannel
+cols: guint
+expander_set: AgsExpanderSet
+group: GtkToggleButton
+mute: GtkToggleButton
+solo: GtkToggleButton
+play: GtkToggleButton
+set_channel(pad:AgsPad,channel:AgsChannel): void
+resize_lines(pad:AgsPad,line_type:GType,
audio_channels:guint,audio_channels_old:guint)
AgsLine
+flags: guint
+version: gchar
+build_id: gchar
+name: gchar
+channel: AgsChannel
+pad: GtkWidget
+label: GtkLabel
+group: GtkToggleButton
+expander: AgsExpander
+set_channel(line:AgsLine,channel:AgsChannel): void
+group_changed(line:AgsLine): void
AgsLineMember
+flags: guint
+widget_type: GType
+widget_label: gchar
+plugin_name: gchar
+specifier: gchar
+control_port: gchar
+port: AgsPort
+port_data: gpointer
+active: gboolean
+recall_port: AgsPort
+recall_port_data: gpointer
+recall_active: gboolean
+task_type: GType
+change_port(line_member:AgsLineMember,port_data:gpointer): void

5.1 Hands-on

5.1.1 get_type()

First you need to tell object type system to use the AgsPluginInterface this is normally done in the classes get_type() function.

Example 5.1 ags_echo_channel_get_type()

```
GType
ags_echo_channel_get_type()
{
  static GType ags_type_echo_channel = 0;

  if(!ags_type_echo_channel){
    static const GTypeInfo ags_echo_channel_info = {
      sizeof (AgsEchoChannelClass),
      NULL, /* base_init */
      NULL, /* base_finalize */
      (GClassInitFunc) ags_echo_channel_class_init,
      NULL, /* class_finalize */
      NULL, /* class_channel */
      sizeof (AgsEchoChannel),
      0,    /* n_preallocs */
      (GInstanceInitFunc) ags_echo_channel_init,
    };

    static const GInterfaceInfo ags_plugin_interface_info = {
      (GInterfaceInitFunc) ags_echo_channel_plugin_interface_init,
      NULL, /* interface_finalize */
      NULL, /* interface_data */
    };

    ags_type_echo_channel = g_type_register_static(AGS_TYPE_RECALL_CHANNEL,
               "AgsEchoChannel\0",
               &ags_echo_channel_info,
               0);

    g_type_add_interface_static(ags_type_echo_channel,
                                AGS_TYPE_PLUGIN,
                                &ags_plugin_interface_info);
  }

  return(ags_type_echo_channel);
}
```

5.1.2 plugin_interface_init()

Implement its set_ports() function.

Example 5.2 ags_echo_channel_plugin_interface_init()

```
void
ags_echo_channel_plugin_interface_init(AgsPluginInterface *plugin)
{
  ags_echo_channel_parent_plugin_interface = g_type_interface_peek_parent(plugin);

  plugin->set_ports = ags_echo_channel_set_ports;
}
```

5.1.3 set_ports()

Finally load the ports restored of XML file.

Example 5.3 ags_echo_channel_set_ports()

```
ags_echo_channel_set_ports(AgsPlugin *plugin, GList *port)
{
  while(port != NULL){
    if(!strncmp(AGS_PORT(port->data)->specifier,
                "./delay[0]\0",
                10)){
      g_object_set(G_OBJECT(plugin),
                   "delay\0", AGS_PORT(port->data),
                   NULL);
    }else if(!strncmp(AGS_PORT(port->data)->specifier,
                      "./repeat[0]\0",
                      11)){
      g_object_set(G_OBJECT(plugin),
                   "repeat\0", AGS_PORT(port->data),
                   NULL);
    }else if(!strncmp(AGS_PORT(port->data)->specifier,
                      "./fade[0]\0",
                      9)){
      g_object_set(G_OBJECT(plugin),
                   "fade\0", AGS_PORT(port->data),
                   NULL);
    }else if(!strncmp(AGS_PORT(port->data)->specifier,
                      "./dry[0]\0",
                      8)){
      g_object_set(G_OBJECT(plugin),
                   "dry\0", AGS_PORT(port->data),
                   NULL);
    }

    port = port->next;
  }
}
```

Appendix A

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See Copyleft.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright © 2013 Joël Krähemann

Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in
the section entitled “GNU Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with… Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts
being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix B

Related projects

Here is a brief list of Advanced Gtk+ Sequencer's direct dependencies.

- gmp: Gnu MP Bignum, http://gmplib.org
- libasound2: ALSA, http://alsa-project.org
- ladspa-sdk: LADSPA, http://www.ladspa.org
- libsndfile1: Sndfile-1, http://www.mega-nerd.com/libsndfile
- libinstpatch1: Instpatch-1, http://www.swamiproject.org
- libxml2: Gnome XML-2, http://www.gnome.org
- libuuid: Universaly Unique Identifier, http://www.sourceforge.net/p/libuuid
- libglib-2.x: GLib-2, http://www.gtk.org
- libgobject-2.x: GObject-2, http://www.gtk.org
- libgdk-2.x: Gdk-2, http://www.gtk.org
- libcairo-2.x: Cairo-2, http://www.cairographics.org
- libgtk-2.x: Gtk-2, http://www.gtk.org
- libxmlrpc-c: XMLRPC-C, http://xmlrpc-c.sourceforge.net