

Audacious - Feature #521

Filewriter Plugin patch to allow output to be piped to stdout/CON

March 16, 2015 19:53 - Jim Turner

Status:	Rejected	Start date:	March 16, 2015
Priority:	Minor	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	3.6.1		
Affects version:	3.6		

Description

Now that Audacious is more like a standard UNIX program accepting streaming input from stdin, it seems only fitting to also be able to pipe output as well! I've implemented a patch to the FileWriter plugin to do just that. The patch only requires modifying filewriter.cc! NOTE: NOT tested on Windows, as I have no way to build Audacious for Windows. Here is the diff (applied against v3.6), along with the full modified source file (attached);

```
124c124,126
< static GtkWidget *prependnumber_toggle;
---
> static gboolean use_stdout; /* JWT */
>
> static GtkWidget *prependnumber_toggle = nullptr;
126a129,130
> static GtkWidget * saveplace1 = nullptr;
> static GtkWidget * saveplace2 = nullptr;
157a162
> "use_stdout", "FALSE", /* JWT */
169a175
> use_stdout = aud_get_bool ("filewriter", "use_stdout"); /* JWT */
191c197
< if (! VFSFile::test_file (filename, VFS_EXISTS))
---
> if (use_stdout || ! VFSFile::test_file (filename, VFS_EXISTS))
223c229
< if (filenamefromtags)
---
> if (use_stdout) /* JWT */
225,228c231,237
< String title = in_tuple.get_str (Tuple::FormattedTitle);
< StringBuffer buf = str_encode_percent (title);
< str_replace_char (buf, '/', '-');
< filename = String (buf);
---
> #ifdef _WINDOWS
> filename = String ("file://CON");
> #endif
> #ifndef _WINDOWS
> filename = String ("file:///dev/stdout");
> #endif
>
232,236c241
< const char * original = strrchr (in_filename, '/');
< g_return_val_if_fail (original != nullptr, 0);
< filename = String (original + 1);
<
< if (! use_suffix)
---
> if (filenamefromtags)
238,240c243,259
< const char * temp;
< if ((temp = strrchr (filename, '.')))
```

```

<         filename = String (str_copy (filename, temp - filename));
---
>         String title = in_tuple.get_str (Tuple::FormattedTitle);
>         StringBuffer buf = str_encode_percent (title);
>         str_replace_char (buf, '/', '-');
>         filename = String (buf);
>     }
>     else
>     {
>         const char * original = strrchr (in_filename, '/');
>         g_return_val_if_fail (original != nullptr, 0);
>         filename = String (original + 1);
>
>         if (! use_suffix)
>         {
>             const char * temp;
>             if ((temp = strrchr (filename, '.')))
>                 filename = String (str_copy (filename, temp - filename));
>         }
242d260
<     }
244,249c262,267
<     if (prependnumber)
<     {
<         int number = in_tuple.get_int (Tuple::Track);
<         if (number >= 0)
<             filename = String (str_printf ("%d%%20s", number, (const char *) filename));
<     }
---
>     if (prependnumber)
>     {
>         int number = in_tuple.get_int (Tuple::Track);
>         if (number >= 0)
>             filename = String (str_printf ("%d%%20s", number, (const char *) filename));
>     }
251,261c269,274
<     if (save_original)
<     {
<         const char * temp = strrchr (in_filename, '/');
<         g_return_val_if_fail (temp != nullptr, 0);
<         directory = String (str_copy (in_filename, temp + 1 - in_filename));
<     }
<     else
<     {
<         g_return_val_if_fail (file_path[0], 0);
<         if (file_path[strlen (file_path) - 1] == '/')
<             directory = String (file_path);
---
>     if (save_original)
>     {
>         const char * temp = strrchr (in_filename, '/');
>         g_return_val_if_fail (temp != nullptr, 0);
>         directory = String (str_copy (in_filename, temp + 1 - in_filename));
>     }
263,264c276,282
<     directory = String (str_concat ({file_path, "/"}));
<     }
---
>     {
>         g_return_val_if_fail (file_path[0], 0);
>         if (file_path[strlen (file_path) - 1] == '/')
>             directory = String (file_path);
>         else
>             directory = String (str_concat ({file_path, "/"}));
>     }
266,267c284,286
<     filename = String (str_printf ("%s%s.%s", (const char *) directory,

```

```

<      (const char *) filename, fileext_str[fileext]));
---
>      filename = String (str_printf ("%s%s.%s", (const char *) directory,
>      (const char *) filename, fileext_str[fileext]));
>  }
321a341
>      aud_set_bool ("filewriter", "use_stdout", use_stdout);    /* JWT */
364c384,387
<      gtk_toggle_button_set_active(GTK_TOGGLE_BUTTON(use_suffix_toggle), FALSE);
---
>      gtk_widget_set_sensitive(saveplace1, TRUE);    /* JWT:NEXT 3 ADDED TO GREY OUT FILE OPTIO
NS WHEN USING stdout */
>      gtk_widget_set_sensitive(saveplace2, TRUE);
>      gtk_widget_set_sensitive(prependnumber_toggle, TRUE);
>      gtk_widget_set_sensitive(path_hbox, TRUE);
366a390
>      use_stdout = FALSE;
374a399,402
>      gtk_widget_set_sensitive(saveplace1, TRUE);    /* JWT:NEXT 4 ADDED TO GREY OUT FILE OPTIO
NS WHEN USING stdout */
>      gtk_widget_set_sensitive(saveplace2, TRUE);
>      gtk_widget_set_sensitive(prependnumber_toggle, TRUE);
>      gtk_widget_set_sensitive(path_hbox, TRUE);
375a404,419
>      use_stdout = FALSE;
>  }
> }
>
> static void filenamefromstdout_cb(GtkWidget *button, void * data)    /* JWT:ADDED TO GREY OUT FILE
OPTIONS WHEN USING stdout */
> {
>     if (gtk_toggle_button_get_active(GTK_TOGGLE_BUTTON(button)))
>     {
>         gtk_widget_set_sensitive(path_hbox, FALSE);
>         gtk_widget_set_sensitive(saveplace1, FALSE);
>         gtk_widget_set_sensitive(saveplace2, FALSE);
>         gtk_widget_set_sensitive(use_suffix_toggle, FALSE);
>         gtk_widget_set_sensitive(prependnumber_toggle, FALSE);
>         use_suffix = FALSE;
>         filenamefromtags = TRUE;
>         use_stdout = TRUE;
412c456
<         GtkWidget * saveplace1 = gtk_radio_button_new_with_label (nullptr,
---
>         saveplace1 = gtk_radio_button_new_with_label (nullptr,
416c460
<         GtkWidget * saveplace2 = gtk_radio_button_new_with_label_from_widget
---
>         saveplace2 = gtk_radio_button_new_with_label_from_widget
448c492
<         (nullptr, _("Original file tag"));
---
>         (nullptr, _("file tag"));
453c497
<         ((GtkRadioButton *) filenamefrom_toggle1, _("Original file name"));
---
>         ((GtkRadioButton *) filenamefrom_toggle1, _("file name"));
456c500,511
<         if (!filenamefromtags)
---
>         GtkWidget * filenamefrom_toggle3 =    /* JWT */
>         gtk_radio_button_new_with_label_from_widget
>         ((GtkRadioButton *) filenamefrom_toggle1, _("stdout"));
>         gtk_box_pack_start ((GtkBox *) filenamefrom_hbox, filenamefrom_toggle3, FALSE, FALSE, 0)
;
>
>         if (use_stdout)    /* JWT */

```

```

>     {
>         gtk_widget_set_sensitive (saveplace1, FALSE);
>         gtk_widget_set_sensitive (saveplace2, FALSE);
>         gtk_toggle_button_set_active ((GtkToggleButton *) filenamefrom_toggle3, TRUE);
>     }
>     else if (!filenamefromtags)
463c518
<         if (filenamefromtags)
---
>         if (filenamefromtags || use_stdout)
471a527,532
>         if (use_stdout) /* JWT */
>         {
>             gtk_widget_set_sensitive(path_hbox, FALSE);
>             gtk_widget_set_sensitive(prependnumber_toggle, FALSE);
>         }
>
477,478c538,539
<         g_signal_connect (filenamefrom_toggle2, "toggled",
<             (GCallback) filenamefromfilename_cb, nullptr);
---
>         g_signal_connect (filenamefrom_toggle2, "toggled", (GCallback) filenamefromfilename_cb,
nullptr);
>         g_signal_connect (filenamefrom_toggle3, "toggled", (GCallback) filenamefromstdout_cb, nu
llptr); /* JWT */

```

Thanks,

Jim Turner

History

#1 - March 17, 2015 14:59 - John Lindgren

- Status changed from New to Rejected

Not possible; all of the output formats (except Vorbis) assume that the output is seekable.

#2 - March 17, 2015 16:26 - Jim Turner

Interesting, b/c I used your latest GIT (as well as on my modified v.3.6) and tested ALL FOUR formats (wav, ogg, mp3, and fla) and they all SEEM to work! I tested both starting playback at the beginning of song file and even switching from ALSA to filewriter "midstream" and back, and I got a playable file each time (with artist & title, if it had them), of the selected type via

```
$>audacious http://anystreamstation.pls >mycapturefile.wav &
```

Am I missing something? Did you actually TRY the patch before rejecting?!

Regards,

Jim

Jim Turner wrote:

Am I missing something? Did you actually TRY the patch before rejecting?!

First, sorry for the very belated reply. It's likely that the files produced would be playable but not finalized correctly. For example, after encoding an MP3 file, we [try to] go back near the beginning of the file and write the LAME/Xing tag, which is needed in VBR files for correct length calculation and seeking. That would fail with stdout, and you would be left with a blank tag at the beginning of the file.

With a WAV file it's similar; we write the header initially indicating a zero file length and zero data length, then go back and update them later. I imagine some players (perhaps including Audacious--I haven't tested this) will play the files even with the zeroes, but that doesn't mean it's good to be writing them that way.

Not sure exactly what would happen with FLAC since the seeking is done at a lower level that we don't have control over in Audacious. Vorbis is the only one of which I can safely say it would work correctly, since the module contains no fseek() calls.

Files

filewriter.cc	18 KB	March 16, 2015	Jim Turner
---------------	-------	----------------	------------