

About Juke It

Thank you for choosing Juke It. Juke It is an extremely easy program to create Flash jukeboxes from your audio files.

Juke It creates SWF files to be played with the Macromedia Flash Player. The latest Flash Player can be downloaded for free from Macromedia's web site at <http://www.macromedia.com>.

System Requirements

- Power PC or Intel based Macintosh or greater
- Macintosh OSX 10.2 or higher
- 64 MB of RAM
- 50 MB of available hard disk space
- QuickTime 4.0

The latest QuickTime can be downloaded for free from Apple's QuickTime website at <http://www.quicktime.com>

Getting Started

Below is a brief description and steps to start you to create your jukebox players with Juke It.

Add audio files to the playlist

To start using Juke It you need to add audio files that you wish to be played in the jukebox. You can add audio files by browsing for them by click the "**Add**" button in the Juke It window or dragging and dropping audio files onto the Juke It window. Click [here](#) for more information on Adding/Removing files.

Customize the Output Settings

Use the "**Output Settings**" window to customize various properties for the look and behaviour of the jukebox. You will need to specify the filename of the jukebox swf as well as the location to where you would like all of the files for the jukebox to be saved to.

Export the jukebox

When you're ready to export the jukebox, you will need to do an export. by choosing "**Export...**" from the "**File**" menu. Click [here](#) for more information on exporting your jukebox.

Add/Remove Audio files to the Playlist

Adding Audio files to the Playlist

Audio files may be added to the playlist by:

- Clicking on the "**Add**" button on the main window.
- Drag and drop a audio file(s) onto the main window.

Supported audio file types include AIFF, MP3, Wave (WAV), SWF. Note that added SWF files will not be resampled, and will just be copied directly over to be used in the jukebox.

Removing Audio files from the Playlist

Select the entry from the list you want to remove. Then click the "**Remove**" button. You can also choose "**Remove All**" to remove all files from the list.

Export Jukebox

When you're ready to deliver your jukebox to an audience, you must export the jukebox for playback.

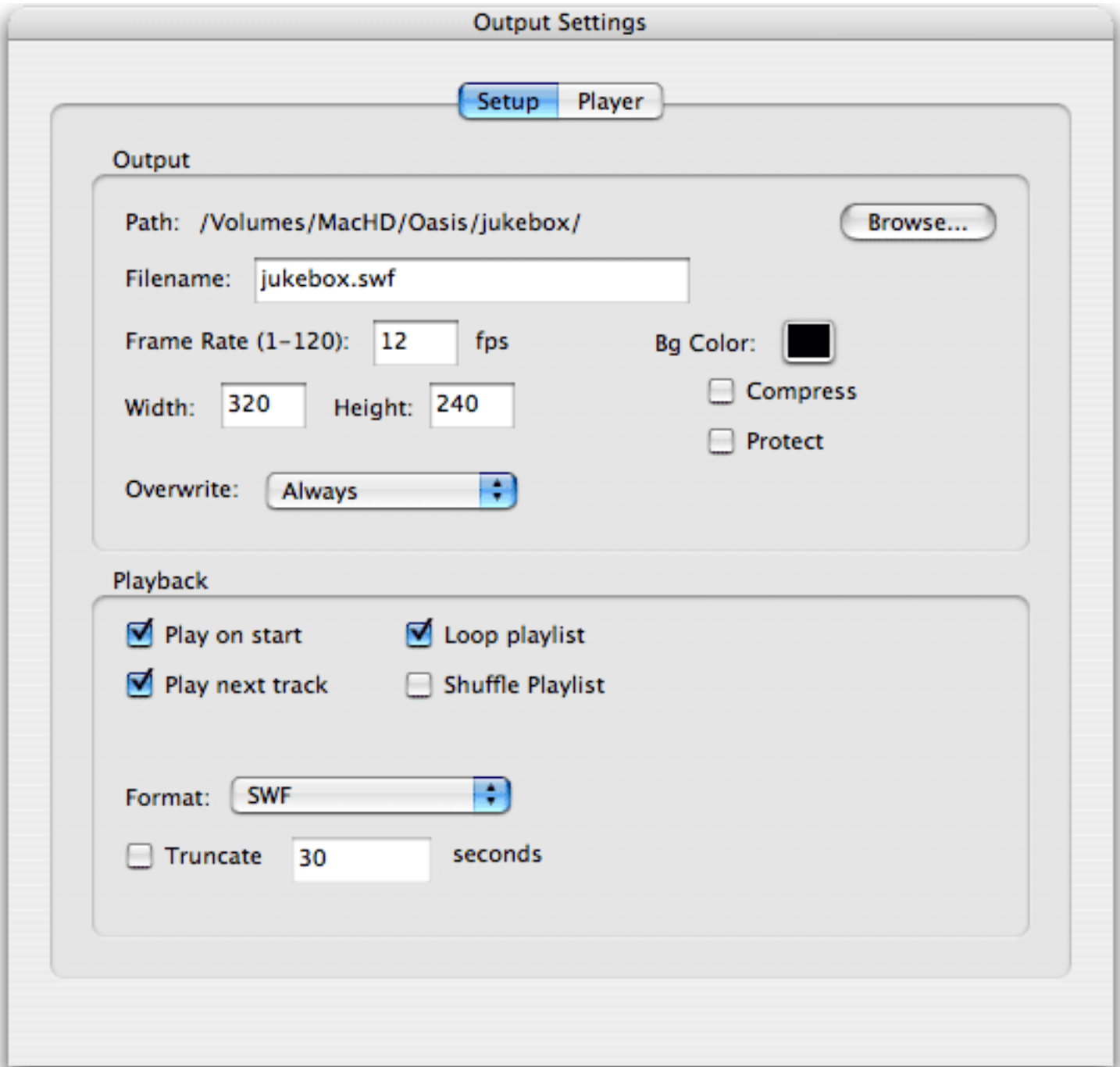
Export

Exporting the jukebox will produce a Flash Movie (SWF) file for the jukebox player along with any audio and images files needed. SWF movie files can be played in the Flash Player or any browser that has the Flash Player plugin installed. To export the slide show, select "**Export**" from the "**File**" menu or the toolbar.

All files will be saved to the specified folder in the [Setup](#) tab. The filename of the jukebox player, is the main file that would be referenced for display in your web page.

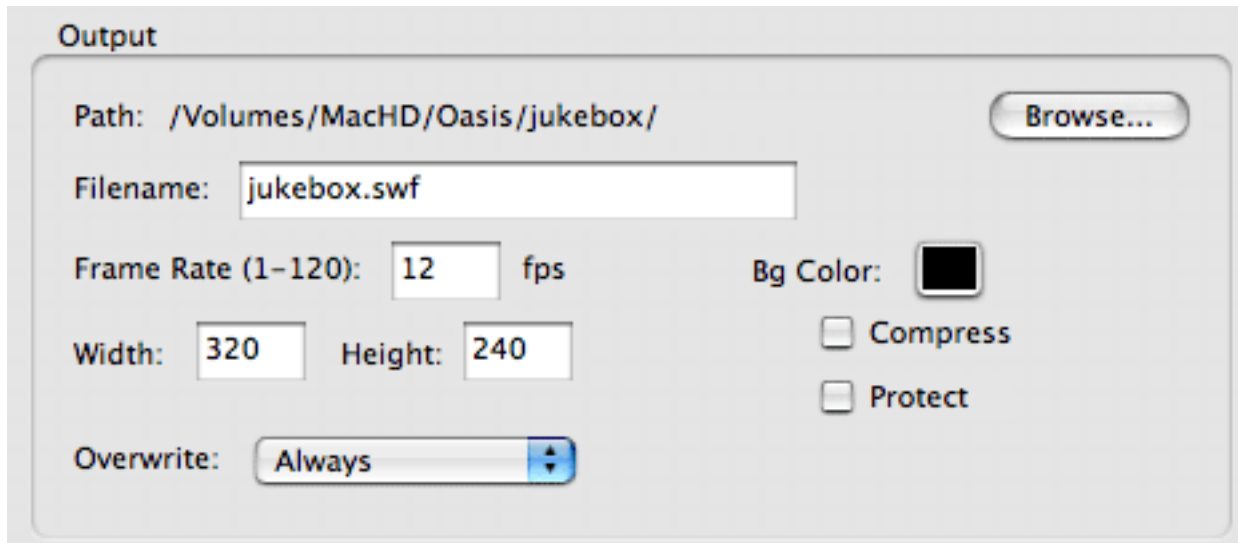
Setup

The **Setup** tab allows you to specify general information regarding the jukebox output and how the jukebox should behave.



Output

The **"Output"** options allows you to specify where the files for the jukebox will be saved to.



Path: Where all of the files for the jukebox will be outputted to. We recommend this be a specific folder for your jukebox project, as the program will output multiple files specifically for the jukebox including the main jukebox SWF, a playlist file and all of the audio and image files needed for the playlist.

Browse...: Click to choose where you would like the files to be outputted to.

Filename: The filename of the jukebox SWF player. This will be the main SWF file that would be displayed in your HTML web page

Frame Rate: The frame rate to use for the jukebox SWF file.

Width: The width of the jukebox SWF file.

Height: The height of the jukebox SWF file.

Bg Color: The background color of the jukebox SWF file.

Compress: If checked, the output will be compressed.

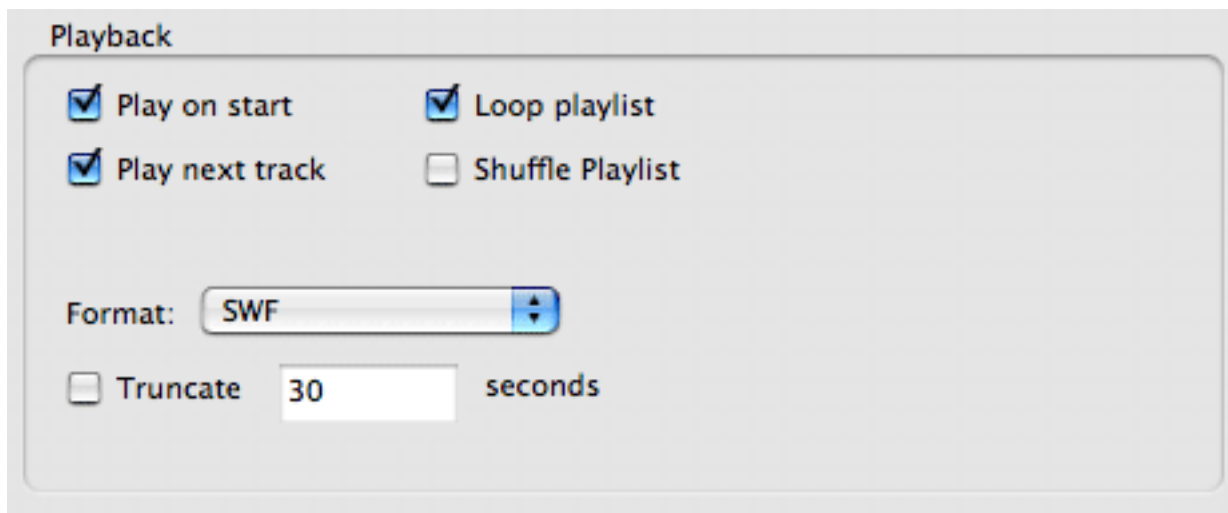
Protect: If checked, the output will not be able to be imported into the Flash authoring tool.

Overwrite: Choose what action should be taken, if an output file already exists.

- Always
- Never
- Ask

Playback

The "**Playback**" options allows you to specify how the playback behaviour of the jukebox.



Play on start: Automatically start playing the first track when the jukebox is loaded.

Play next track: Automatically play the next track in the playlist after the current track has finished playing.

Loop playlist: Continuously play all of the songs in the playlist.

Shuffle playlist: Randomly play the songs in the playlist.

Format: Choose what format to output the audio files as.

- SWF
- FLV

Truncate: If checked, you can specify a maximum number of seconds to be played from all tracks in the playlist.

Save/Load Projects

Saving Juke It Projects

You can save the settings for your current session to a Juke It Project file (.juka) by choosing "**Save Project**" or "**Save Project As...**" from the "**File**" menu or toolbar. To overwrite the current project file, choose "**Save Project**". To save the file in a different location or with a different name, choose "**Save Project As...**".

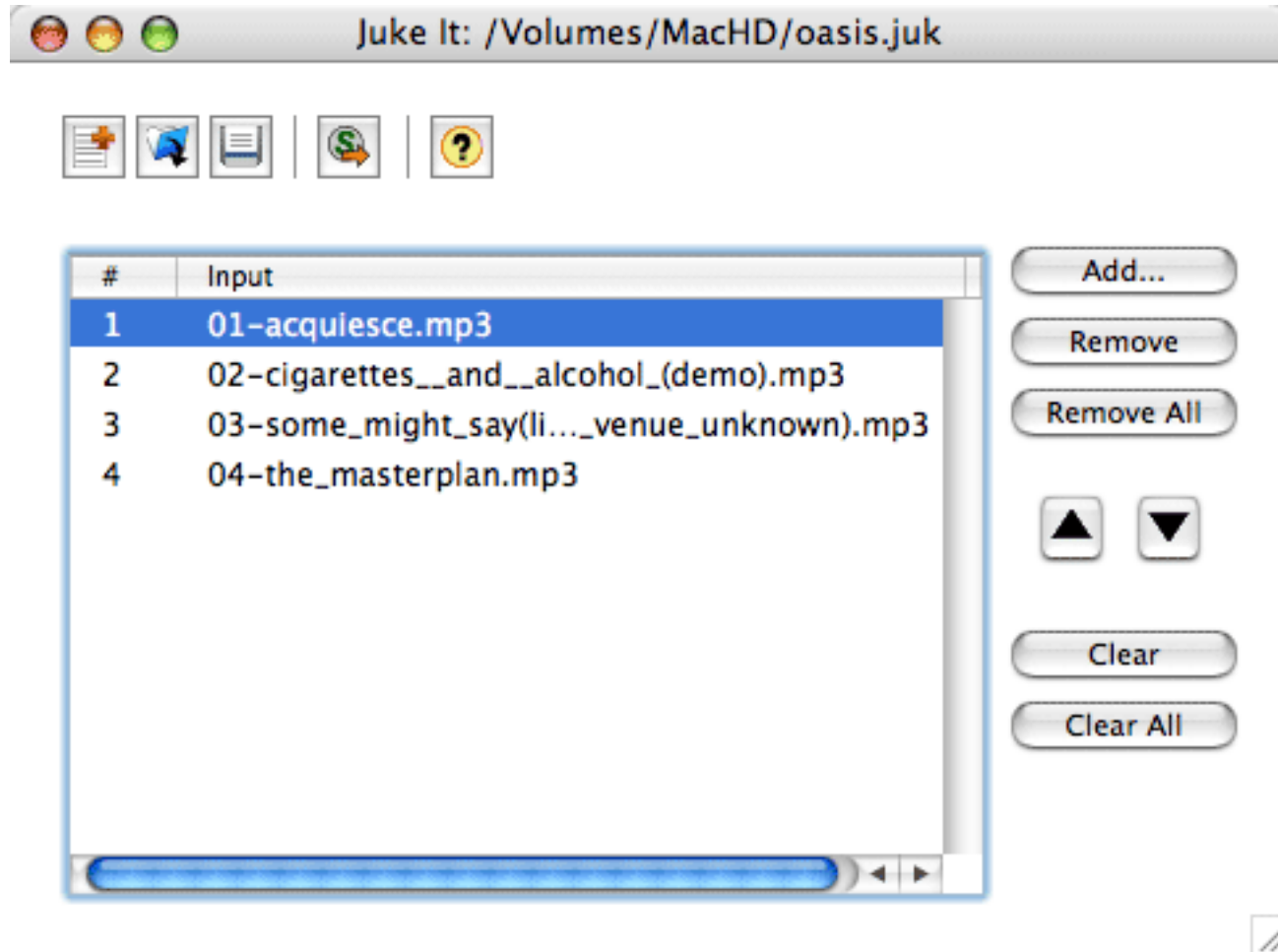
Please note that the project file does not save any video/images/sounds used in the project. If you move or delete any of the files used in the project, Juke It will not be able export the files properly.

Loading Juke It Projects

You can load and previously saved Juke It Project file (.juka) by choosing "**Open Project**" from the "**File**" menu or toolbar.

Juke It

The **Juke It** window is the main window and is where you add and remove the audio files you wish to be played in your jukebox player. The order in which the files are in this window, is the order they will be played in the jukebox player (unless the Shuffle option is chosen).



Add...: Click to choose a source audio file to be added to the playlist.

Remove: Click to remove the current audio file from the list.

Remove All: Click to remove all files from the list.

Up Arrow: Click to move the current selected file up in the playlist order.

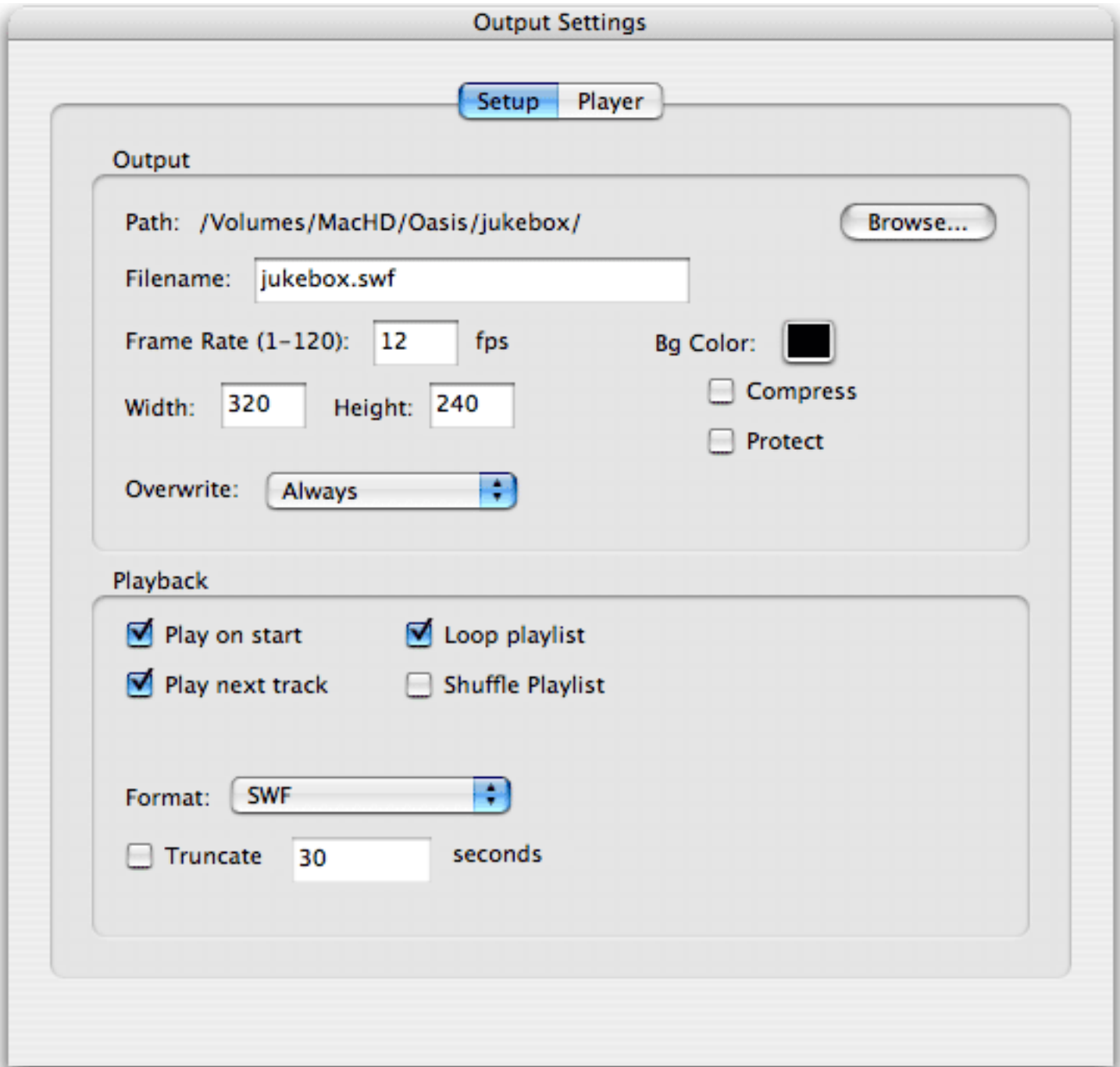
Down Arrow:: Click to move the current selected file down in the playlist order.

Clear: Click to delete the exported file if it exists.

Clear All: Click to delete all exported files if they exist.

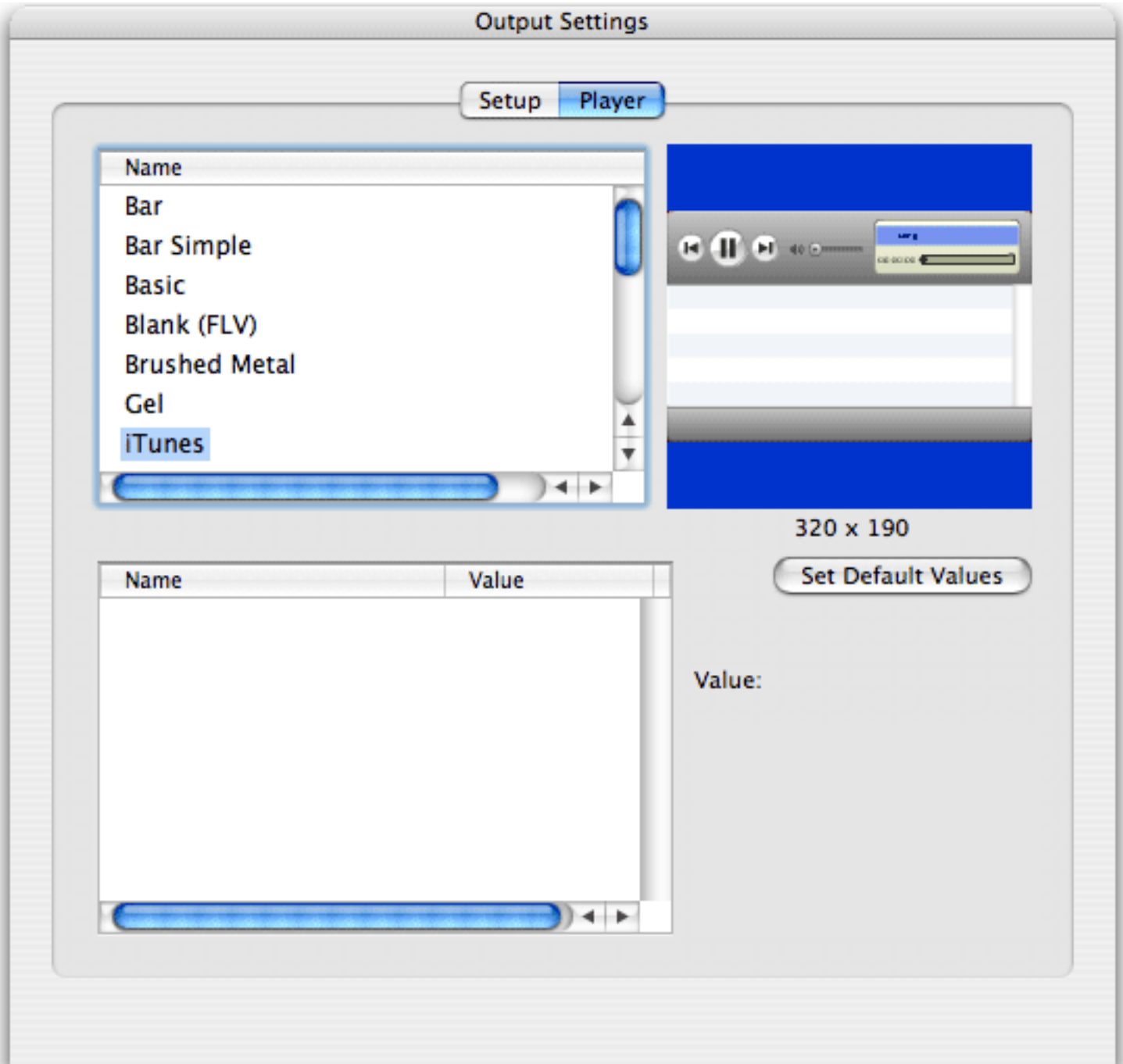
Output Settings

The **Output Settings** dialog allows you to customize the look and behaviour of the jukebox player. You will also specify where you would like the files for the jukebox to be outputted to.

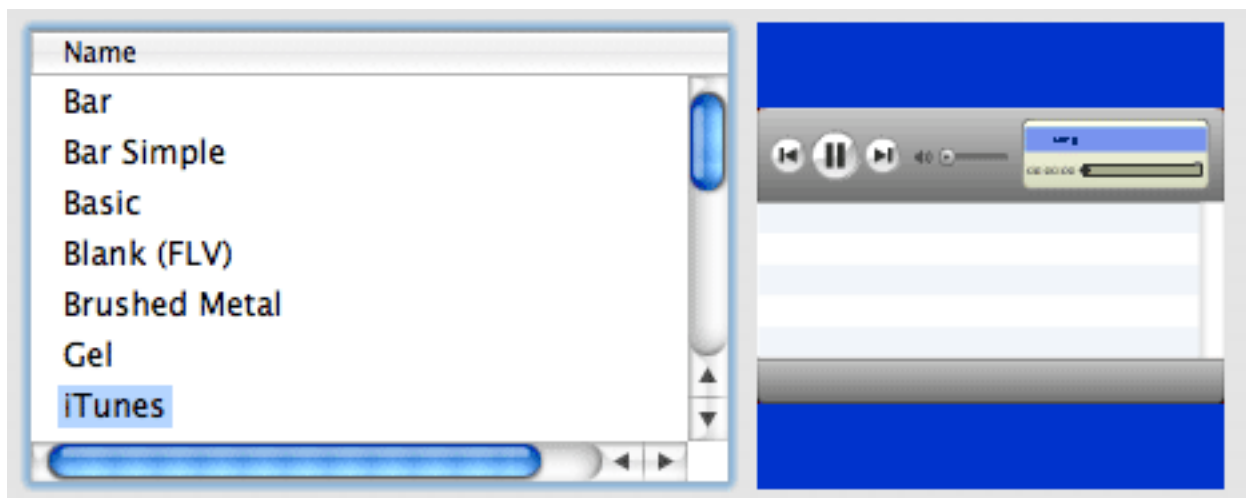


Players

The "**Player**" tab allows you to add a Player to the encoded video to add playback and audio controls.



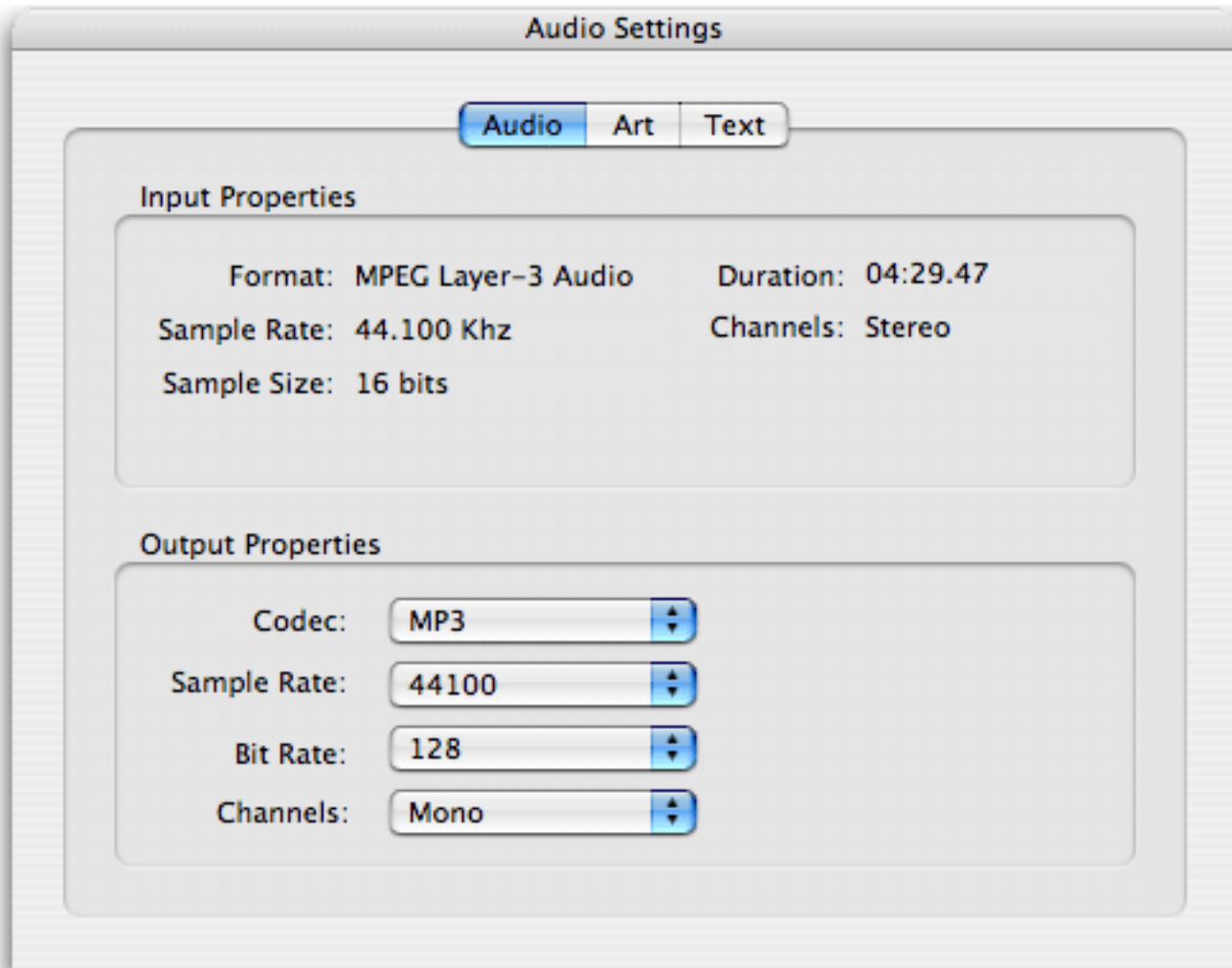
Player



Choose a Player template to use from the list. After selecting a template from the list, you will see a Preview of what the Player template looks like below.

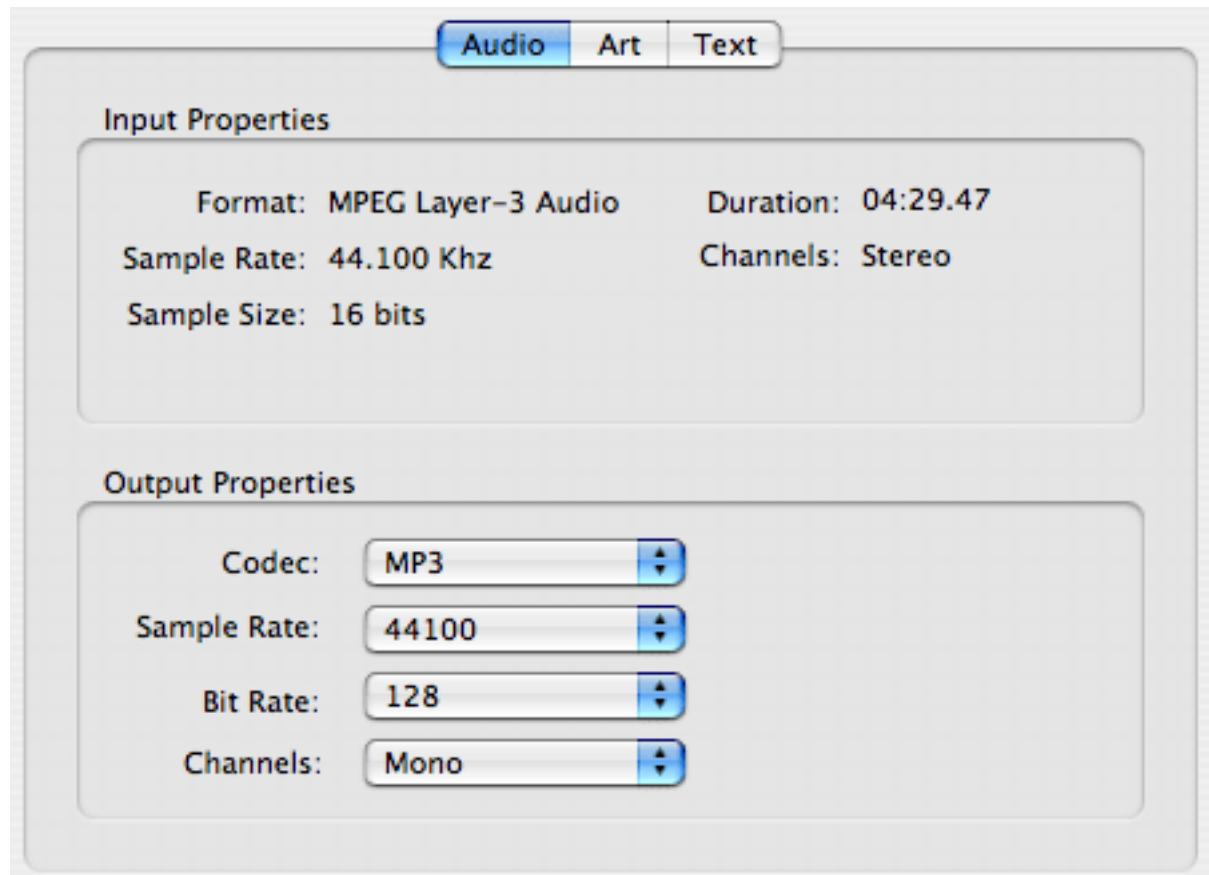
Audio Settings

The **Audio Settings** dialog allows you to specify the audio quality for the tracks. You can also specify an image and track descriptions to display in the jukebox player (In player templates that support these features).



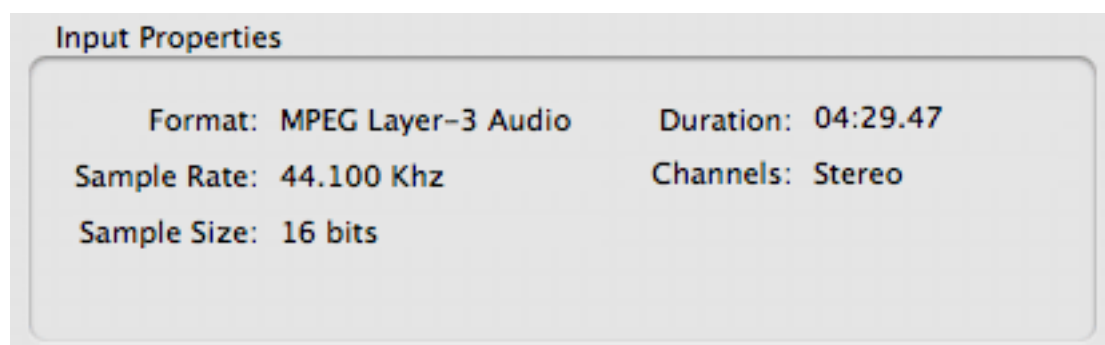
Audio

The **Audio** tab allows you to see the properties of the audio source and set the output quality settings.



Input Properties

The "**Input Properties**" displays various properties of the input audio source.



Format: The format of the audio source.

Sample Rate: The sample rate of the audio source

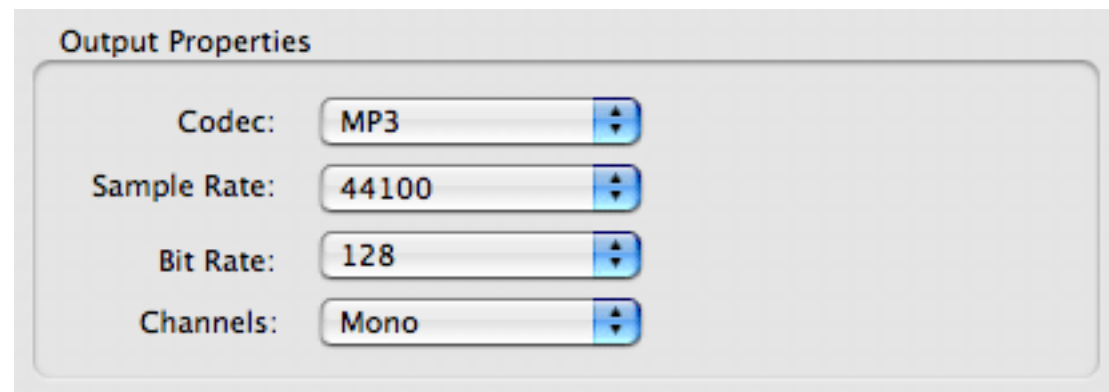
Sample Size: The sample size of the audio source.

Duration: The duration of the audio source.

Channels: The number of channels in the audio source.

Output Settings

The "**Output Settings**" allows you to choose various settings to control the quality of the encoded audio.



Codec: The codec to use when encoding the audio.

- MP3

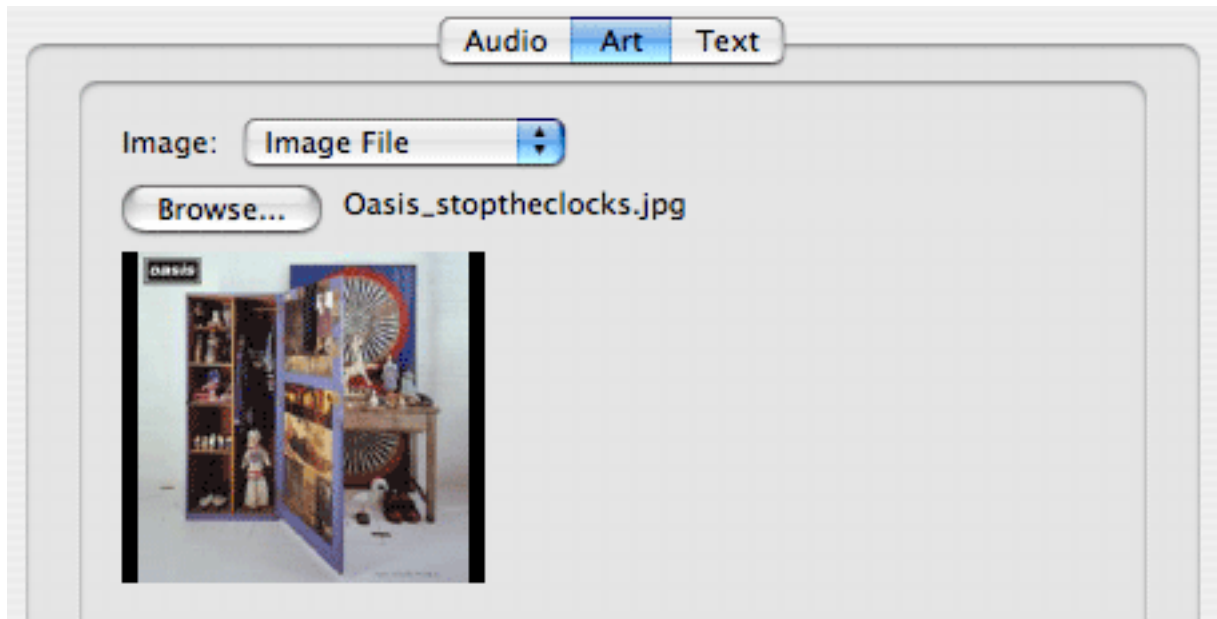
Sample Rate: The sample rate to use when encoding the audio. Higher sample rates will result in higher audio quality, but larger file sizes.

Bit Rate: The bit rate to use when encoding the audio. Higher bit rates will result in higher audio quality, but larger file sizes.

Channels: The number of channels to use when encoding the audio.

Art

The **Art** tab allows you to specify an image or SWF to display in the jukebox player when the chosen track is selected. Note that only some of the players support this feature.



Input

The "**Progress Display**" options allows you to specify a graphical display for the Preloader.

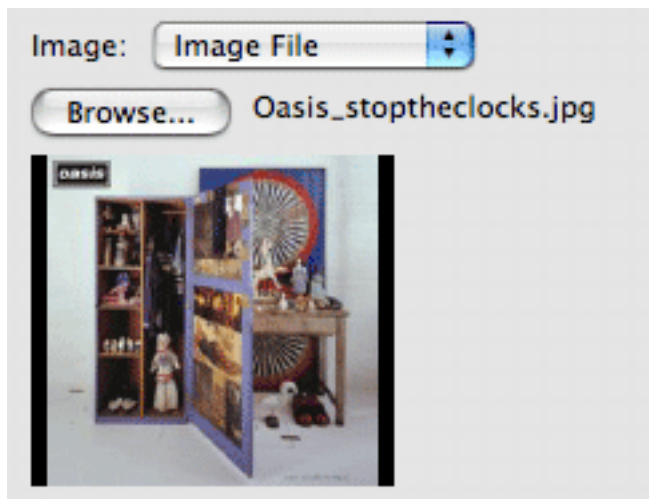
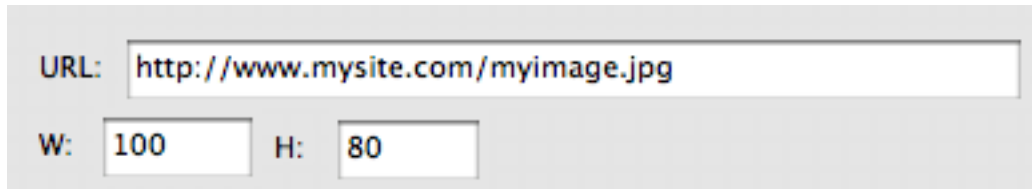


Image: Choose what to use as the source image or SWF to display.

- **Image File:** Select a local image or SWF to use. The chosen file will be converted or copied to the output location of the jukebox

- **URL/Path:** Enter a full or relative remote path to the image or swf you wish to be displayed. If an image is chosen, the format of the image needs to be a non-progressive jpeg file.

Browse...: Click to choose an image or SWF to display.



A screenshot of a web form with a light gray background. It contains three input fields: a text box for the URL containing 'http://www.mysite.com/myimage.jpg', and two smaller text boxes for width and height, containing '100' and '80' respectively.

| | | |
|------|-----------------------------------|-------|
| URL: | http://www.mysite.com/myimage.jpg | |
| W: | 100 | H: 80 |

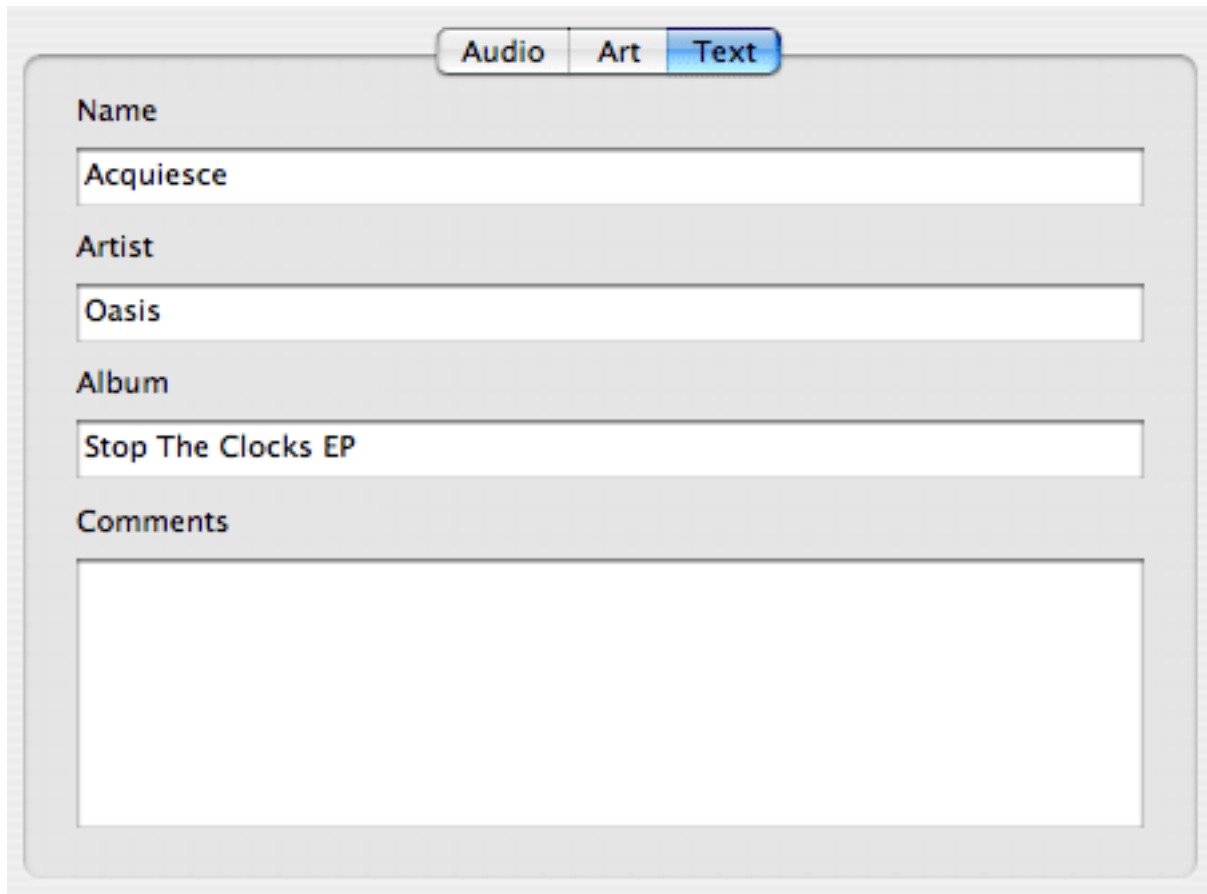
URL:: Enter the URL of the image or SWF to display.

W:: Enter the width in pixels of the remote image or SWF that is specified by the URL.

H:: Enter the height in pixels of the remote image or SWF that is specified by the URL.

Text

The **Text** tab allows you to specify text to display in the jukebox related to the selected track. Note that only certain players support this feature.



The image shows a screenshot of a software interface with three tabs: 'Audio', 'Art', and 'Text'. The 'Text' tab is selected and highlighted in blue. Below the tabs, there are four input fields with the following labels and values:

- Name:** Acquiesce
- Artist:** Oasis
- Album:** Stop The Clocks EP
- Comments:** (An empty text area)

Name: The name or title of the audio track.

Artist: The artist's name of the audio track.

Album: The album name of the audio track.

Comments: Any comments regarding the audio track.

Frequently Asked Questions

What is a SWF file?

The Macromedia Flash file format (SWF) (pronounced "swiff") delivers vector graphics and animation over the Internet to the Macromedia Flash Player.

What is a FLV file?

The Macromedia FLV file format lets you import a static video stream with encoded audio.

What platforms are supported for Macromedia Flash playback?

*Macintosh OS 9.x, OS 10.x
Windows 98, ME, 2000,
XP*

How many people have the Macromedia Flash Player installed?

The Macromedia Flash Player is the world's most pervasive software platform, used by over 1 million professionals and reaches more than 97% of Internet-enabled desktops worldwide as well as a wide range of devices.

What version of the Flash Player are the SWF file exported from Juke It compatible with?

Juke It creates SWF files that are compatible with the Flash Player version 7 and up. If you are publishing Projector files, the Flash Player is not needed since the player is already a part of the Projector file.

There isn't any audio when I encode my audio. What should I do?

If running Windows, install or re-install the latest QuickTime with the "Recommended" Installation option checked.

What is the maximum audio duration that I can encode?

The Flash player imposes a 16,000 maximum frame limit on SWF files. Anything beyond this will not be played. So the actual maximum duration will depend on the frame rate used for the SWF file (10fps = ~26 minutes)

Flash FLV files however do not have this 16,000 frame limit.

Can I add my own players to Juke It?

Yes, it is possible to create your own custom Players and add them for use in Juke It. You will need to use Macromedia/Adobe's Flash authoring tool and follow our [specification](#) for creating Players for Juke It.

Players Specification

You can create your own custom player templates to be used in Juke It by following the specifications found on this page. You can create your players with the Macromedia Flash authoring tool (At least version 7 is required). Once created, you can add the players to the Players folder where Juke It is installed, and the next time you launch Juke It, you should see the new players added to the list.

The following specification will describe the overall layout and what is required or optional when creating your players.

- Player Hierarchy
 - [Parameters/Variables](#) (required actionscript)
 - [videoPlayer](#) (required)
 - [video](#) (required)
 - [frame](#)
 - frame_n (where n is 1...number of elements in the frame)
 - [vidBtns](#)
 - rwBegBtn
 - rwBtn
 - playPauseBtn
 - playBtn
 - pauseBtn
 - stopBtn
 - ffBtn
 - ffEndBtn
 - prevBtn
 - nextBtn
 - [sndBtns](#)
 - muteBtn
 - volSlider
 - volBtn
 - volBar
 - [vidBar](#)
 - progressBtn
 - progressBar
 - loadBar
 - bgBar
 - [clock](#)
 - time_cur

- time_rem
 - time_tot
 - [playlist](#)
 - plframe_n (where n is 1...number of elements in the frame)
 - playlistLb
-

Parameters/Variables

The first frame of the main timeline is where you can specify how the elements in the player should position and scale relative to the size of the video within the Player.

Required:

format = "video2swf";

version = "1.0";

name = "The name of your player to be displayed in Juke It ";

Optional:

min_player_width = (the minimum width in pixels the player can be);

min_player_height = (the minimum height in pixels the player can be);

ITEM_hanchor = "none/left/center/right"; (where ITEM is the name of the element to set the anchor for; eg. vidBtns_hanchor="left");

ITEM_vanchor = "none/top/center/bottom"; (where ITEM is the name of the element to set the vanchor for; eg. vidBtns_vanchor="bottom");

ITEM_hstretch = "true/false/fixed"; (where item is the name of the element to set the stretch for; eg. frame_1_hstretch="true");

ITEM_vstretch = "true/false/fixed"; (where item is the name of the element to set the stretch for; eg. frame_1_vstretch="false");

vidbar_width: If using a vidBar in the Player, you will need to set this to the original width of the bar.

vidbar_scroll: If using a vidBar with a slider button, you will need to set this to the original amount it can scroll within the vidBar.

volSlider_scroll: If using a volSlider, you will need to set this to the original amount it can scroll within the volSlider.

videoPlayer

The first frame of the main timeline must have an instance of the movie clip that holds the entire player and its elements. The instance must be named "videoPlayer". The videoPlayer should be placed at (0,0) and the Document dimensions should match the size of the videoPlayer.

Required:

[video](#)

Optional:

[frame](#)

[vidBtns](#)

[sndBtns](#)

[vidBar](#)

[clock](#)

[playlist](#)

video

The video instance will define where the actual video will be placed in the player. The instance must be named "video" and should contain a rectangle shape defining the area for the video. The video and elements around the video will automatically be adjusted based on the actual dimensions of the Flash video.

frame

The frame can be used to create the background for the player. For each element in the frame, there will be an instance named "frame_n" where "n" is a number from 1...number of frame elements.

vidBtns

The vidBtns can contain the buttons to control the playback of the audio, such as rewinding, fast forwarding, pausing and playing.

Optional: The following must be Movie clip instances with the following names which contain the actual Button. For toggle buttons, the first button state must be in the first frame of the containing movie clip, and the second state must be in the second frame.

rwBegBtn: Jump to the beginning of the audio

rwBtn: Rewinds the audio

playPauseBtn: Toggle button to play and pause the audio

playBtn: Plays the audio
pauseBtn: Pauses the audio
stopBtn: Stops the audio
ffBtn: Fast forwards the audio
ffEndBtn: Jump to the end of the audio
prevBtn: Jump to the previous track in the playlist
nextBtn: Jump to the next track in the playlist

sndBtns

The sndBtns can contain the controls to control the audio such as muting and changing the volume.

Optional: The following must be Movie clip instances with the following names which contain the actual Button. For toggle buttons, the first button state must be in the first frame of the containing movie clip, and the second state must be in the second frame.

muteBtn: Toggle button to mute and unmute the audio
volSlider: Slider bar to set the volume. This must have an instance named **volBar** which is the background bar, and a **volBtn** which is the slider button. The first frame variables should have the following **volSlider_scroll** set to the number of pixels the volBtn can scroll.

vidBar

The vidBar can be used to add a scrubber and to show the current position in the video.

Optional: The following must be Movie clip instances with the following names which contain the actual Button. For toggle buttons, the first button state must be in the first frame of the containing movie clip, and the second state must be in the second frame.

The first frame variables should have a variable **vidbar_width** set to the original width of the vidBar. The first frame variables should have a variable **vidbar_scroll** set to the number of pixels the progressBtn can scroll.

progressBtn: Button to show the current position in the video. Can also be used to scrub through the video by dragging it along the video bar.
progressBar: A bar showing the current position in the video
loadBar: A bar showing the current amount of the video loaded/buffered
bgBar: The background bar

clock

The clock can display relevant times for the audio, such as the current time and the remaining time.

Optional:

time_cur: Displays the current time of the audio. This must be a dynamic text field with the variable name set to "time_cur"

time_rem: Displays the remaining time of the audio. This must be a dynamic text field with the variable name set to "time_rem"

time_tot: Displays the total time of the audio. This must be a dynamic text field with the variable name set to "time_tot"

playlist

The playlist displays all available tracks to play in the jukebox player.

Optional:

playlistLb: This is a Flash List Component instance. It must be called "playlistLb"

The frame can be used to create the background for the playlist. For each element in the frame, there will be an instance named "plframe_n" where "n" is a number from 1...number of frame elements.

Custom Variables

You can allow certain elements to be customizable from within Audio2SWF by adding variables. The following is an example for allowing the user to choose the color for the video buttons.

```
v2sVar1_name = "videoPlayer.vidBtnsColor";  
v2sVar1_type = "color";  
v2sVar1_value = "0xffffffff";  
v2sVar1_label = "Video buttons color";  
v2sVar1_notes = "Color of the video buttons playback buttons";
```

Each custom variables entry must begin with v2sVarN_ (where N is the variable number).

_name: Then name of the actual variable used in Flash

_type: Type of variable. Can be the following:

- **alphanum:** alpha numeric string (eg. "my value 123")
- **alpha:** alpha (eg. "my value")
- **int:** integer values (eg. 1, 2, 3)
- **float:** decimal values (eg. 3.14, 5.29)

- **bool:** true or false
- **color:** RGB color in hex format 0xRRGGBB

_value: The default value for the variable

_label: The name to be displayed in the list of variables in Audio2SWF

_notes: A description of the variable which will be displayed in Audio2SWF

Contact Us

E-mail:

support@verticalmoon.com

Web:

<http://www.verticalmoon.com/support>