



ProppFrexx Payout

Version 4.0

Installation and User Guide

© 2016 RADIO42, Bernd Niedergesäß. All rights reserved.

ProppFrexx Payout, ProppFrexx ONAIR and RADIO42 are registered trademarks in Germany.

All other company, brand, and product names may be trademarks of their respective holders.

This material is the confidential property of RADIO42 and its subsidiaries or licensors and may be used, reproduced, stored or transmitted only in accordance with a valid RADIO42 license or sublicense agreement.

RADIO42
Bernd Niedergesäß
Gryphiusstrasse 9
22299 Hamburg
Germany

Revision Date: Juni 2016

Table of Contents

Concept and Idea	3
The Idea.....	3
The Concept	4
INSTALLING ProppFrexx Payout	5
Hardware and software requirements.....	5
Supported Operating Systems.....	5
Supported Audio Interfaces	5
Hardware requirements.....	5
3rd Party software requirements.....	5
Installing ProppFrexx Payout	6
Registering ProppFrexx Payout.....	6
STARTING ProppFrexx Payout	7
Command-Line Options.....	7
WORKING with ProppFrexx Payout	9
The Main Screen	9
The Settings and Options Dialog.....	11
The Output Device Configuration Dialog.....	13
CONFIGURATION of ProppFrexx Payout	16
TilesConfig.xml	16
The Tiles Designer.....	20
Updating the configuration	21

Table of Figures

Figure 1: ProppFrexx Payout Main Screen.....	3
Figure 2: Direct Payout.....	4
Figure 3: In-Direct Payout.....	4
Figure 4: ProppFrexx Payout Hardware-ID.....	6
Figure 5: Elements of the Main Screen	9
Figure 5: Settings and Options dialog.....	11
Figure 5: Settings and Options dialog.....	13
Figure 5: Tile Designer dialog.....	20

Concept and Idea

ProppFrexx Payout is designed as a modern Windows Tile application, which is fully touch enabled and can be used as an application to serve either direct or indirect payout of audio (music) content in a kiosk like fashion, e.g. for hotels, restaurants, bars, shops, theaters etc.

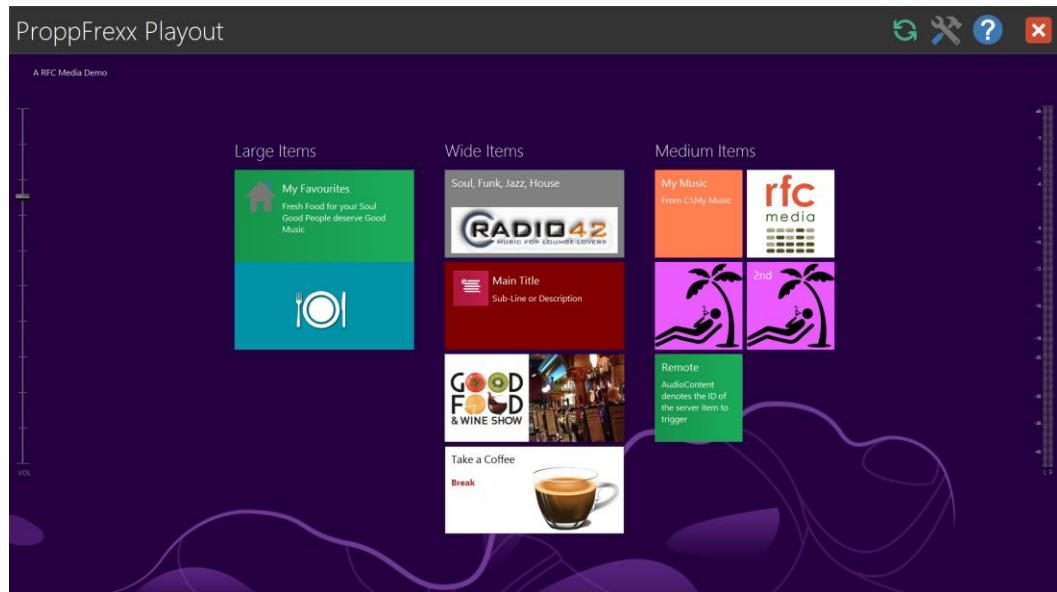


Figure 1: ProppFrexx Payout Main Screen

The look and feel is fully customizable to individually adjust the entire design to the related company (or brand) using this application. The application is so simple to use, that it can easily be operated by really everyone (from a shop assistant, bartender or dishwasher to a top level manager). It can be run on any modern Windows 8 or 10 device, starting with a simple table to a larger desktop server. It can be run as a standalone application with direct payout capabilities connected to an in-house music unit or as a pure remote unit, which triggers the payout on another (server) instance, which might be connected to a complex multi-output music system; or anything in between (mixed hybrid mode).

Each tile represents a music item which starts playing by a simple click or touch. A click on the same tile stops the payout. A click on another tile will play the other music item. This can for example be used in restaurants, hotels or shops to play different moods of music, e.g. at different times of the day. A music item of a tile can either be a:

- **Streaming URL** (playing a fully branded 24/7 internet stream)
- **Pre-prepared Playlist** (playing a customized music sequence)
- **Folder Content** (randomly playing hand-selected local content)
- **Remote ID** (playing a tile item on a connected remote server)

As a result, ProppFrexx Payout is the perfect payout tool to integrate into any environment.

The Idea

Many high-class hotels, SPAs, top restaurants or shops really take care about everything to serve their guests with the best possible experience. This includes the design of the interior (color, material and haptic), the lightning concept, the look of the employees as well as the selection of items presented in their guest toilets. However, a complete and overall perfect sensation can only be achieved by including music into this concept. And unfortunately this important element is often overlooked. Many companies do play the same music every evening with only a few variations and with no surprise to the customers.

ProppFrexx Payout serves as a tool and platform to fill this gap. Together with e.g. RFC Media (the leading creator of custom produced, content-rich, digital music stations for brands, events and ideas) it offers a customized and hand-crafted sound design to make a customer stay even more enjoyable.

The Concept

Playout what you need and how you like it, where you need it. Here are some setup examples:

- 1) ProppFrexx Payout is directly connected to a music system via a build-in soundcard:



Figure 2: Direct Payout

- 2) ProppFrexx Payout is also connected remotely to a ProppFrexx Payout Server which serves a multi-output music system (one instance can control many 'rooms').

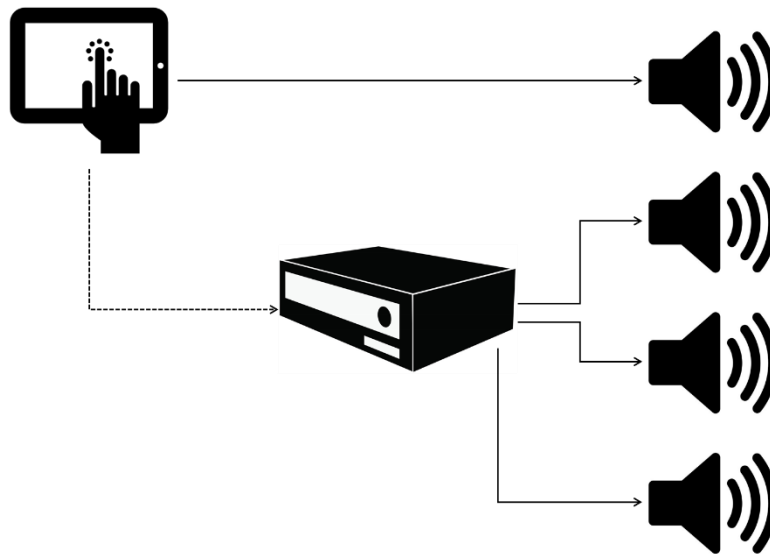


Figure 3: In-Direct Payout

As you might guess, you can connect each tile not only to the same, but to an individual output. Thus, multi-channel soundcards or even multiple soundcards can be used to playout multiple items in parallel. All might be controlled remotely by a single ProppFrexx Payout instance. Even complex networked or cross-linked systems might be set up.

As you might individually design the tiles on each ProppFrexx Payout instance you can arrange the sound design just as you need it.

The integration of the different music item types (streaming URLs, playing pre-prepared playlists or random folder content) lets you simply adjust the sound design to the mood and scene of your choice.

INSTALLING ProppFrexx Payout

Hardware and software requirements

Supported Operating Systems

- Microsoft Windows® 7 64-bit or 32-bit
- Microsoft Windows® Server 2008 family 64-bit or 32-bit
- Microsoft Windows® 8/8.1 64-bit or 32-bit
- Microsoft Windows® Server 2012 family 64-bit or 32-bit
- Microsoft Windows® 10 64-bit or 32-bit
- Microsoft Windows® Server 2015 family 64-bit or 32-bit

Supported Audio Interfaces

Any single or multi-channel soundcard with a Core Audio (WASAPI) driver supported by the underlying OS.

For best quality we recommend to use a professional audio soundcard which supports full 32-bit or 24-bit processing. Note that you can use any number of soundcards in your system.

Hardware requirements

ProppFrexx Payout requires the following hardware:

Category	Recommended
CPU	Dual-Core 1.8 GHz or higher
RAM	4 GB
Disk space	At least 10 GB
Monitor:	1024x768, 32-bit or higher



The hardware requirements might depend on the type of use and the final deployment scenario. However, the system is able to fully scale with your environment.

3rd Party software requirements

ProppFrexx Payout requires the Microsoft **.NET Full Framework in version 4.6 or above**. The latest .NET Framework will normally be installed with your Windows Operating System. However, you can obtain the .NET Framework from the Microsoft website: <http://msdn.microsoft.com/netframework/default.aspx>

Furthermore, the **Microsoft Windows Media Audio Codec** resp. the **Microsoft Windows Media Foundation** is required if you intend to playback WMA, AAC or MP3 files. The Windows Media Audio Codec will normally be installed with your Windows Operating System, however, you might also install the latest Windows Media Player: <http://www.microsoft.com/windows/windowsmedia>

Installing ProppFrexx Payout

After you have successfully downloaded the file *ProppFrexx_Payout.zip*, please unzip this file to a new and empty directory of your choice (do not use any write-protected system folder like the Program Files folder), which will result in the following sub-folders being created:

- /AddOns : contains any BASS add-ons to support playback of additional audio formats (codecs).
- /Content : default folder where the design and configuration files should be placed into. This folder contains a sample configuration by default.

The root installation folder will contain the executable file as well as all needed libraries (.dll files).


The main configuration file is called *TilesConfig.xml* contained in the */Content* sub-folder. This file contains all references and definitions of all tiles incl. the references to the audio content.



Tip: Make sure to use relative references (paths) within all your configurations, in order to be independent from the final deployment (e.g. when you later want to copy your configuration to different target devices).

Registering ProppFrexx Payout

ProppFrexx Payout is licensed not sold. This means you need a valid license key file in order to run the software permanently. Without a valid license you run ProppFrexx Payout in demo mode for 21 days. After this testing period you need a valid license.

In order to obtain a valid license, please send your ProppFrexx Payout *Hardware-ID* to proppfrexx@radio42.com. You'll find your ProppFrexx Payout *Hardware-ID* in the *Settings and Options* dialog (click on the  icon in the top bar and enter the *Master Password*, which defaults to "ProppFrexx" without the quotes):

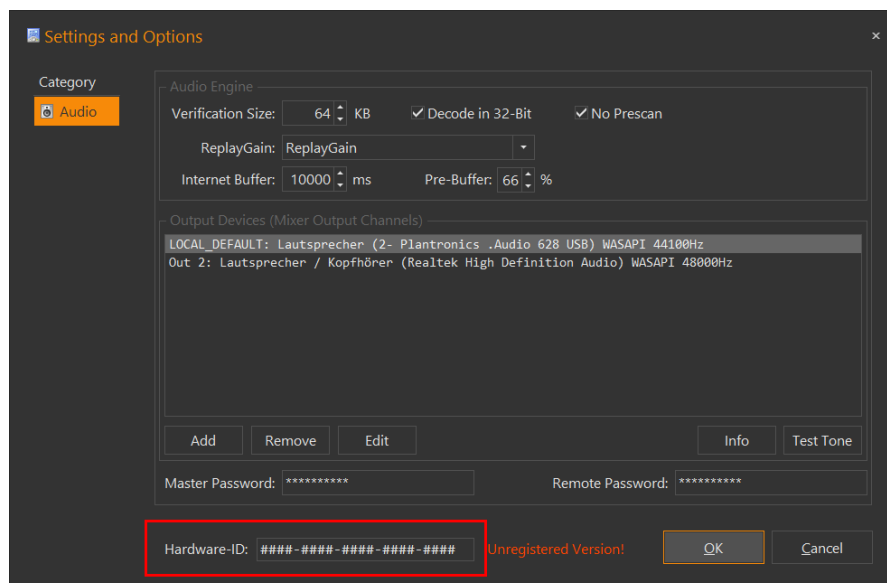


Figure 4: ProppFrexx Payout Hardware-ID

After you have received back a license file, save this license file into your installation folder, which is the same folder containing the *ProppFrexx_Payout.exe*.

STARTING ProppFrexx Payout

You start ProppFrexx Payout by executing the file *ProppFrexx_Payout.exe* – e.g. via a double-click or by creating a related link on the desktop.

On a final deployment device, you might automatically start ProppFrexx Payout when starting your device.

Command-Line Options

Certain functionality can be activated by starting ProppFrexx Payout with a dedicated command-line option.

`-content=<folder>`

If specified ProppFrexx Payout will be enforced to use the given content folder instead of the default */Content* sub-folder underneath the executable file.

`-screen=<#>`

If specified ProppFrexx Payout will be shown on the given screen number (1=first or primary screen, default). This option might be useful in 2 scenarios: a) when you run ProppFrexx Payout in a multi-monitor environment and/or b) when you want to start ProppFrexx Payout multiple times on the same device in parallel.

`-allowmultiple`

If specified ProppFrexx Payout will be allowed to start multiple time on the same device in parallel.

`-client=<remote-server-address>`

If specified ProppFrexx Payout will be started in client or hybrid mode. The remote server address must be specified in the format: `tcpex://host:port`. The remote server will be another ProppFrexx Payout instance started with the `-server` command-line option.

`-server=<port>`

If specified ProppFrexx Payout will be started in server mode. This enables clients to connect to this instance and trigger playback remotely. In order to trigger playback remotely, the client's tile item will use the value *REMOTE_SERVER* in its *OutputTo* configuration entry and use the tile item *ID* of this remote server in its *AudioContent* configuration entry.

`-proxy=<user:pass@server:port>`

If specified ProppFrexx Payout will use this proxy for any internet connection.

`-agent=<userAgent>`

If specified ProppFrexx Payout will use this user agent value for any internet stream.

-design

If specified ProppFrexx Payout will be started in *Designer* mode. This enabled various options and additional buttons in order to directly edit the current configuration online. Note, together with the *-config* command-line option you might also maintain multiple configurations in parallel.

Examples:

Start ProppFrexx Payout as a server on the 2nd screen:

```
ProppFrexx_Payout.exe -server=9142 -screen=2
```

Start ProppFrexx Payout as a client (connecting to a server running at 192.168.10.15):

```
ProppFrexx_Payout.exe -client=tcpx://192.168.10.15:9142
```

Start ProppFrexx Payout using an alternative content directory

```
ProppFrexx_Payout.exe -config="C:\My Playouts\Config Main"
```


WORKING with ProppFrexx Playout

The Main Screen

The main screen contains the following visual elements:

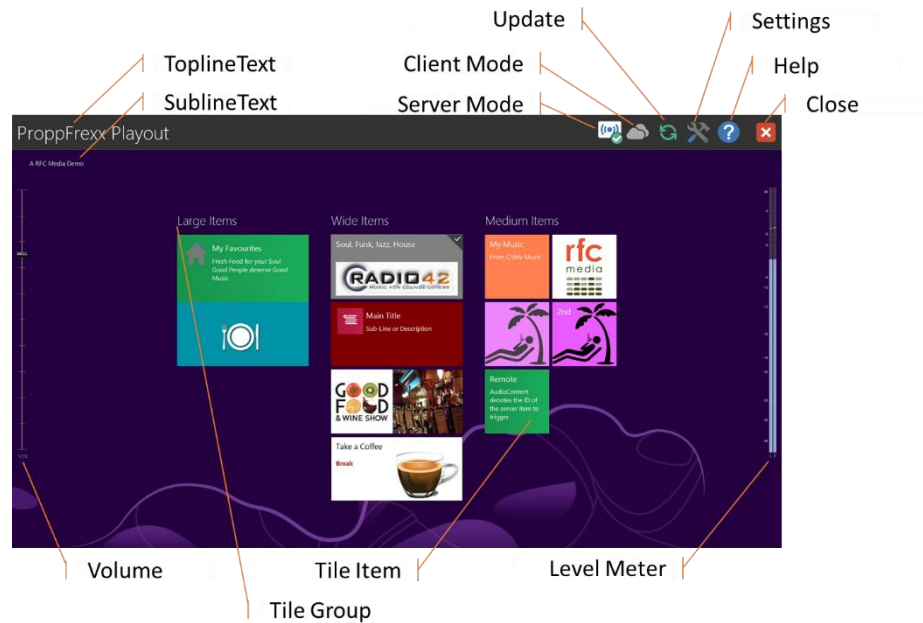


Figure 5: Elements of the Main Screen

ToplineText

Defines the large text to display in the top window bar. The color and background cannot be changed.

SublineText

Defines the small text to display in the main tile. The background color and image of the tile area can be customized.

Server Mode

This button is only displayed in server mode (see the `-server` command-line option). A click can temporarily activate or deactivate the server mode. A green check mark will indicate a running server. A red exclamation mark will indicate a server error and a grayed symbol will indicate a temporarily deactivated server.


Client Mode

This button is only displayed in client mode (see the `-client` command-line option). A click can trigger a (re)connect to the server. A green check mark will indicate a connected client. A grayed symbol will indicate, that the client is not connected to then server.


Update

This button triggers an update of the configuration. You can optionally provide in the tile configuration a special *UpdateCheckURL* which will point to a certain *update.xml* file hosted


on any web-server. When clicking on this button the *update.xml* file will be evaluated. In case the web-server sided version number is higher than the currently used local version number, a configurable content zip file will be downloaded and new content (tiles as well as audio content) will be downloaded and installed automatically. You can hide the update button via the *AllowUpdate* tile configuration entry.

Settings 

This button will open the general settings and options dialog. You need to enter the *Master Password* beforehand to open it. The default *Master Password* is "ProppFrexx" (without the quotes).

Help 

This button will show a small end user help text.

Close 

This button will close and exit the application.

Volume

This slider controls the output volume of the *LOCAL_DEFAULT* output and defaults to 75%. It has no effect on any other outputs.

Level Meter

This indicates the current sound level of the *LOCAL_DEFAULT* output in a graphical way in order to quickly see, that something is playing locally.

Tile Group

Logically groups multiple tile items together and can have an optionally group text displayed on top. You might use groups to order to music item (tiles) e.g. by room, genre or mood.


Tile Item

Represents one playable music item which starts playing by a simple click or touch. A check mark in to upper right corner will indicate, that the tile is currently playing. When using multiple-outputs in parallel, there can only be one playing item per output. A music item of a tile can either be a:

- **Streaming URL** (playing a fully branded 24/7 internet stream)
- **Pre-prepared Playlist** (playing a customized music sequence)
- **Folder Content** (randomly playing hand-selected local content)
- **Remote ID** (playing a tile item on a connected remote server)

The entire layout of the individual tiles and groups as well as the design of the application is fully customizable in one central configuration file: *TilesConfig.xml*, which is located in content folder.

The Settings and Options Dialog

You invoke the *Settings and Options* dialog by clicking on the  icon in the top bar and entering the *Master Password* (which defaults to "ProppFrexx" without the quotes):

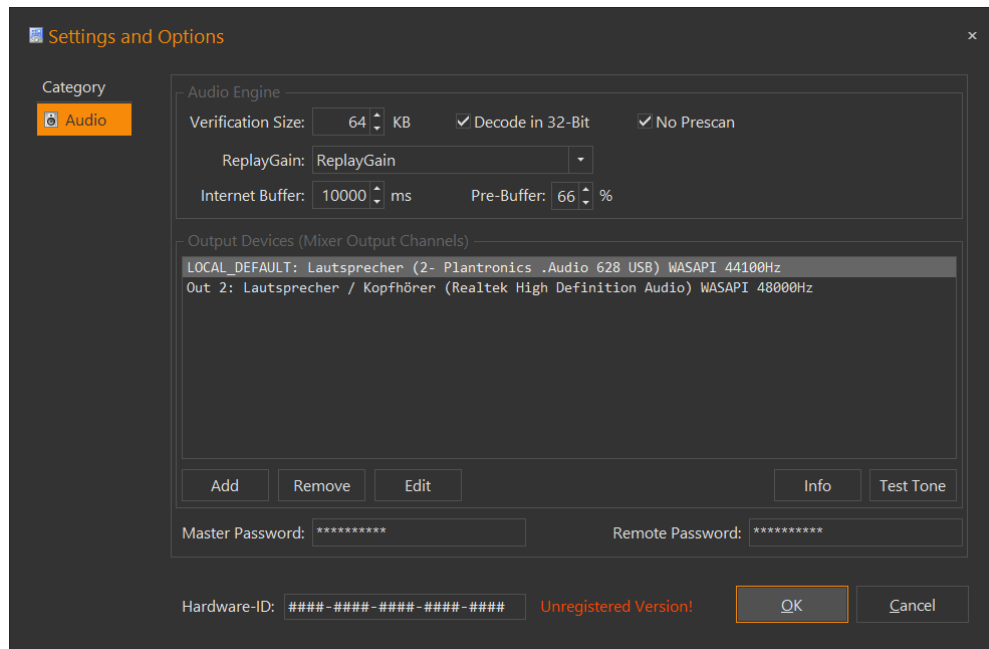


Figure 6: Settings and Options dialog

This dialog allows you configure certain options:

Verification Size

Defines the amount of data in kilo bytes to check in order to verify/detect the file format of any audio file. The verification length excludes any tags that may be at the start of the file.

Decode in 32-Bit

If checked all audio tracks will be decoding in 32-Bit floating-point resolution, else 16-Bit integer resolution will be used. The main advantage of floating-point channels, aside from the increased resolution/quality, is that they are not clipped until output. So even if the output device is not capable of outputting the channel in its full quality, the quality is still improved.

No Prescan

If checked, no prescan of MP3/2/1 files is performed. To ensure exact determination of a track duration and to ensure sample accurate seeking it is recommended to leave this option unchecked. However pre-scanning of audio files require an initial full file read, which might take some time on slower devices. If you experience issues (e.g. long playback start times) when using slow storage devices, you might try to activate this option.

Relay Gain Mode

Defines the replay gain mode to use for playback. Replay gain calculation might allow you to normalize the perceived audio volume to an equal level for all tracks.

Off: No replay gain will be applied.

Normalization: The peak audio level will be normalized to be at 0 dB.

ReplayGain: Standard replay gain will be applied (for tracks not having a replay gain value set one will automatically be calculated).

Normalization + ReplayGain: First normalization will take place, followed by the standard replay gain calculation.

Dynamic ReplayGain: Like the standard replay gain calculation, but the effective replay gain value will be adjusted according to the maximum replay gain value calculated so far.

Internet Buffer

The streaming download buffer length in milliseconds to be used when playing content from the internet. Increasing the buffer length decreases the chance of the internet stream stalling, but also increases the time taken to create the stream as more data has to be pre-buffered. The internet buffer length should be larger than the length of the playback buffer, otherwise the stream is likely to stall soon after starting playback. The default is 10 seconds.

Internet Pre-Buffer

Amount to pre-buffer when opening internet streams. This setting determines what percentage of the internet buffer length should be filled when opening internet streams. The default is 66%.

Mixer Output Channels

Each element in this list represents one mixer output channel, which is a logical playback device. You might define any number of output channels and assign it to any of your physical soundcard devices.

In the tile configuration you will reference a mixer channel output name in order to define on which device a certain tile item should be played out.

Add

Adds a new output channel to the mixer.

Remove

Removes the selected output channel from the mixer.

Edit

Let's you define the detail settings of the selected output mixer channel.

Info

Shows detailed information about the currently selected output device.

Test Tone

Generates a 1kHz test tone for the currently selected device as long as this button is pressed.

Master Password

The master password is used to protect this configuration dialog (resp. which needs to be entered prior to the *Designer* mode).

Authorization Password

This password is used to verify any remote control client and server. Thus, client and server must use the same password in order to be able to establish a connection.

The Output Device Configuration Dialog

You invoke the *Output Device Configuration* dialog by clicking on the *Edit* button in the list of the *Output Devices* in the *Settings and Options* dialog (or by double-clicking on a resp. entry):

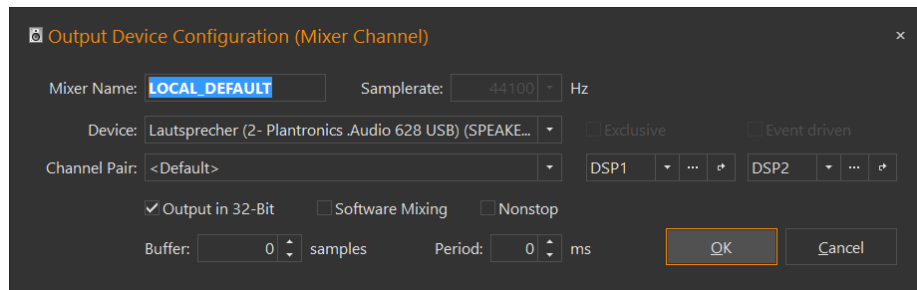


Figure 7: Settings and Options dialog

This dialog allows you configure a certain output:

Mixer Name

Defines the name of the output channel and is used to identify the output in the tile configuration.

Samplerate

The sample rate of the mixer output channel in Hz.

If a played track using this output does not match this frequency it will automatically be resampled.

Device

The physical soundcard driver resp. device to be used with this output mixer channel.

Channel Pair

If your soundcard device supports multiple speakers/channels you might assign a certain speaker pair to this mixer channel. Select which channel pair should be used with this mixer channel.

Exclusive

Use the device in exclusive or shared mode? Exclusive: The device can only be used exclusively by this application - which allows greater precision and lets you choose the output sample rate.

Shared: The device can be shared by multiple applications. Enforces to use the default sample rate and format, see the windows sound control panel for more information.

Event driven

Use the event driven system? By default, WASAPI pushes data to device, i.e. it's not event-driven, as the event-driven system isn't always supported/working with all drivers.

Enable this option to turn on the event driven system, in which case the WASAPI driver requests for data whenever needed. Some devices/soundcards work better in this mode - so you can only try which mode works better in your case.

Output in 32-Bit

Use 32-Bit floating-point sample data for this mixer output channel? The main advantage of floating-point channels, aside from the increased resolution/quality, is that they are not clipped until output. If your soundcard support 24- or 32-Bit output this option will result in highest sound quality. If unchecked 16-Bit audio resolution is used.

Note: Internal DSP/FX processing will anyhow use full 32-Bit floating-point precision. So this setting only effects the data send to the output device.

Software Mixing

By default, hardware mixing is used (whenever available), enable this option to enforce software mixing for this mixer channel. If you discover any trouble with your soundcard (e.g. clicks or hops) try to enable this option.

Nonstop

When setting the output to nonstop, an output signal will always be generated (at least silence) even if no input (source) is present or paused for this mixer channel. This setting might be useful if you want to use this mixer channel for internet broadcast streaming which requires a constant output stream.

The disadvantage of a nonstop output is, that there will be a delay (by the buffer size) until a source signal will actually be heard.

When disabling this option (non-nonstop) the mixer channel is stalled (not producing any output) when no source is connected or when any source is paused. If a source resumes or is connected, the output will start instantly even if it was stalled.

However, to support synchronized outputs it is recommended to enable the NONSTOP option for all physical outputs.

Buffer

The buffer length directly defines the latency of the audio signal. A smaller buffer decreases the latency but increases the chance that the playback might break.

WASAPI: The buffer length in milliseconds. 0 = use default length (see device's windows sound control panel).

Period

The update period is the amount of time between updates of the playback buffers of channels. Shorter update periods allow smaller buffers to be set, but as the rate of updates increases, so the overhead of setting up the updates becomes a greater part of the CPU usage. Specify 0 to use the default update period.

DSP 1 / DSP 2

Allows you to assign up to two VST (v2.4) DSP plug-ins per output channel.

Click on the '...' button to select or remove a VST plug-in. Click on the last button to invoke the DSP internal editor dialog.

Note: Make sure to only use 32-bit VST plug-ins for the 32-bit version of this software and only use 64-bit VST plug-ins for the 64-bit version of this software.

CONFIGURATION of ProppFrexx Payout

The configuration of the look and feel (design) as well as the functionality and grouping of the tiles is done in one central configuration file called *TilesConfig.xml*. This file must reside in the used content folder, which is by default the */Content* sub-directory. You might edit the *TilesConfig.xml* file either manually with any Xml Editor of your choice; or you might start ProppFrexx Payout with the *-design* command-line option to invoke the *Designer* mode.

TilesConfig.xml

The *TilesConfig.xml* file contains the following elements:

`<Tiles> </Tiles>` section

Root xml element defining the tiles configuration. All other elements must be specified underneath.

`<AllowDrag>` element (Boolean, 0 or 1)

Is dragging of tile items within groups allowed? In Design Mode you can always re-arrange the Items and Groups.

Parent: `<Tiles>`

`<AllowDragBetweenGroups>` element (Boolean, 0 or 1)

Is dragging of tile items between groups allowed. In Design mode you can always re-arrange the items and groups.

Parent: `<Tiles>`

`<AllowUpdate>` element (Boolean, 0 or 1)

Should the update button be visible and available?

Parent: `<Tiles>`

`<UpdateCheckURL>` element (string)

The Url or local path to the update.xml file containing the latest version and effective content zip location.

Parent: `<Tiles>`

`<CurrentVersion>` element (integer)

The version number of this design configuration (a higher number denotes a new version).

Parent: `<Tiles>`

`<BackgroundImage>` element (string)

The relative or absolute path to a background image to display on the main window.

Parent: `<Tiles>`

`<BackgroundColor>` element (string)

The background color of the main window in Html notation (starting with a # character).

Parent: <Tiles>

<ToplineText> element (string)

The big top window bar text to display.

Parent: <Tiles>

<SublineText> element (string)

The smaller main tile window text to display.

Parent: <Tiles>

<GroupList> </GroupList> section

Xml element defining the list of available groups. All other group sections must be specified underneath.

Parent: <Tiles>

<Group> </Group> section

Xml element defining one particular tile group. All other group elements must be specified underneath.

Parent: <GroupList>

<GroupText> element (string)

The tile group top text to display.

Parent: <Group>

<ItemList> </ItemList> section

Xml element defining the list of available items within one group. All other item sections must be specified underneath.

Parent: <Group>

<Item> </Item> section

Xml element defining one particular tile item. All other item elements must be specified underneath.

Parent: <ItemList>

<ID> element (integer)

Number to uniquely identify the item. Caution: Each tile item MUST have a unique ID!

Parent: <Item>

<AudioContent> element (string)

Reference to the content to be played (can be a StreamUrl, a FolderPath, a PlaylistFile or a remote ID). Use relative paths for folders and playlists whenever possible.

Parent: <Item>

<OutputTo> element (string)

Name of Output Mixer to use (or REMOTE_SERVER).

Parent: <Item>

<ItemType> element (string)

Defines the size and behavior of the tile. Must be one of these values: *LargeStatic*, *WideStatic*, *WideAnimated*, *MediumStatic* or *MediumAnimated*.

Parent: <Item>

<BackgroundColor> element (string)

The background color of the tile item's first page in Html notation (starting with a # character).

Parent: <Item>

<MainText> element (string)

The main title to display on the first page of the tile item.

Parent: <Item>

<SubTitle> element (string)

The subline text to display underneath the main title on the first page of the tile item.

Parent: <Item>

<IconImage> element (string)

The relative or absolute path to an 64x64 image to display on the first page of the tile item in the upper left corner.

Parent: <Item>

<BackgroundImage> element (string)

The relative or absolute path to a background image to display on the first page of the tile item (e.g. Medium=160x160, Wide=328x160, Large=328x328).

Parent: <Item>

<SecondPageTitle> element (string)

The title to display on the second page of the tile item.

Parent: <Item>

<SecondPageImage> element (string)

The relative or absolute path to an inner image to display on the second page of the tile item (e.g. Medium=160x160, Wide=328x160, Large=328x328).

Parent: `<Item>`

`<SecondPageBackgroundImage>` element (string)

The relative or absolute path to a background image to display on the second page of the tile item (e.g. Medium=160x160, Wide=328x160, Large=328x328).

Parent: `<Item>`

The order of the groups resp. the order of the items within a group represent the order in which the appear on screen.

How to configure a Remote Tile Item?

A remote tile can only be used, if ProppFrexx Payout is started with the *-client* command-line option (meaning the client is connected to a running remote server instance of ProppFrexx Payout). In the tile item configuration make sure to use the following values for the following *<item>* elements:

When using the special *REMOTE_SERVER* keyword in the tile item's *<OutputTo>* configuration element, the *<AudioContent>* element is evaluated differently. In it specify the *ID* value of the tile item of the connected ProppFrexx Payout remote server instance.

Example:

```
<ID>2</ID>
<OutputTo>REMOTE_SERVER</OutputTo>
<AudioContent>15</OutputTo>
```

ID 2 is the local ID of the client tile item (must be unique in the client configuration); when this tile item is played, it triggers the payout of the remote tile with the ID 15 on the connected remote instance of ProppFrexx Payout. As such, the payout is started on the remote instance and not on this client (the level meter of the client will not reflect the payout as well as the client volume control has no effect on a remote payout; but the client's check mark should indicate a successful remote triggering).

The Tiles Designer

Instead of manually editing the *TilesConfig.xml* file you might also start ProppFrexx Payout with the *-design* command-line option (note, that you must enter the *Master Password* to confirm this at start up). This invokes the *Designer* mode, which adds the following buttons to the main window:



: Click here to invoke the *Tile Designer* dialog



: Click here to save the current layout and design to the current *TilesConfig.xml* file.



: Click here to export the current configuration to a new and empty folder and to create local sub-folders for all relevant content as needed to ensure a local payout.

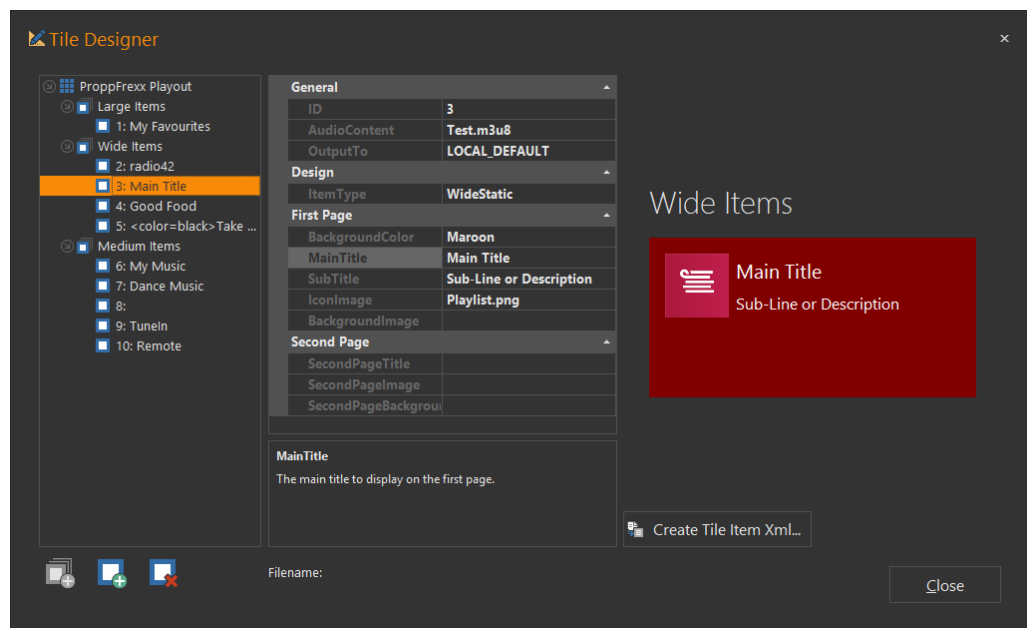


Figure 8: Tile Designer dialog

The *Tile Designer* dialog lets you configure all options as explained above in a simple way.

On the left side you'll see a tree representing your groups and tile items in a hierarchical way. Just click on one entry to edit its content in the middle property grid. A selected tile item will be previewed on the right-side as it would appear in your main tile control.


Three buttons let you add new groups or items resp. remove an element from the tree. Please note, that you cannot rearrange groups or items in this dialog. Please use the dragging feature of the main tile in order to arrange the items and groups exactly as you need them.

The *Create Tile Item Xml...* button will create the xml for exactly one tile item and place that into the clipboard, in case you want to use that to manually edit your *TilesConfig.xml* file.



Caution: The selected paths to e.g. the *AudioContent* or *Images* is NOT automatically normalized to a relative path (relative to the *TilesConfig.xml* file). However, it is highly recommended to only use relative paths (whenever possible) in order to ensure, that a configuration is working on all deployed dev ices; e.g. make sure all images and audio files are accessible on the target device.

Updating the configuration

The system allows you to configure an automatic update option. If configured, the given `<UpdateCheckURL>` (see above) is evaluated automatically at each startup of the application resp. if the `<AllowUpdate>` option is set to 1, if a user clicks on the  update button.

The `<UpdateCheckURL>` denotes the location of a `update.xml` file e.g. on your web-server (a sample file is provide in the installation folder). The `update.xml` file looks e.g. like this:

```
<?xml version="1.0" encoding="utf-8"?>
<ProppFrexxPayout>
  <version>2</version>
  <url>http://www.proppfrexx.radio42.com/payout/content.zip</url>
</ProppFrexxPayout>
```

The `<version>` element defines the version number to compare to the `<version>` element of the current `TilesConfig.xml` file. In case the version of the provided `update.xml` is greater than the version of the current `TilesConfig.xml` file, the update takes place.

In this case the provided `content.zip` file as denoted by the `<url>` element in the `update.xml` is downloaded and then unzipped to the current content directory – overwriting any existing files and sub-folders!

As such, the provided content zip file should be a complete zip of your new configuration, incl. all needed audio files, the new `TilesConfig.xml` and any other needed files (e.g. images).

The content zip should not include the `/Content` folder itself, but start right underneath: