DJ PRO

Music is love

Documentation

Date: 15.03.2024

Version: 2024.1.2

© 2015-2024 **cross**tales LLC

https://www.crosstales.com

Table of Contents

1. Overview....4
2. Features....5
2.1. DJ and turntables....5
2.2. Filter & order....5
2.3. Documentation & control....6
2.4. Compatibility....6
2.5. Integrations....6
3. Demonstration....7
3.1. Play....7
3.2. Mixer....7
3.3. Piano....8
3.4. 3DAudio....8
3.5. Loudspeaker....9
3.6. PlaySingleFile....9
3.7. RecordList....10
4. Setup....10
4.1. Schema....11
4.2. Turntable....12
4.3. Providers....13
4.3.1. RecordProviderPath....13
4.3.2. RecordProviderPlaylist....13
4.3.3. Other providers....13
4.4. Sets....14
4.4.1. RecordSet....14
4.4.2. DJSet....14
4.5. SimplePlayer....15
4.6. Other components....15
4.6.1. Loudspeaker....15
4.6.2. RecordSaver....15
4.6.3. CrossFader....15
4.6.4. RecordFader....16
4.6.5. Looper....16
4.6.6. SurviveSceneSwitch....16
4.6.7. Eventer....16
4.6.8. BPMAnalyzer....16
5. API....17
5.1. Player....17
5.1.1. Play....17
5.2. Callbacks....18
5.2.1. Playback start and end....18
5.2.2. Buffering start, end and progress....18
5.2.3. Audio start, end and time....19
5.2.4. Errors....19
5.2.5. Filter change....19
5.2.6. Records change....19
5.2.7. Record change....20
5.2.8. Example....20
5.3. Complete API....20
6. Third-party support (PlayMaker etc.)....21
7. Verify installation....21
8. Upgrade to new version....21

9. Important notes....22
9.1. Android....22
9.2. UWP (WSA)....22
10. OnRadio....23
10.1. Step 1....23
10.2. Step 2....23
10.3. Step 3....23
11. Problems, improvements etc....24
12. Release notes....24
13. Credits....24
14. Contact and further information....25
15. Our other assets....26

Thank you for buying our asset "DJ PRO"!

If you have any questions about this asset, send us an email at dj@crosstales.com.
Please don't forget to rate it or write a little review – it would be very much appreciated.

1. Overview

If you ever wanted to listen to your own music within your game, you can do so now.
DJ is, simply put, the missing link between Unity and your music-library.

As the name promises, it's a solution to build the music player of your dreams – just like a real DJ. It adds and plays audio files from anywhere inside your application, it's very configurable and you can add as many records, sets and turntables as you wish.

DJ is a must for everyone who loves music!

2. Features

2.1. DJ and turntables

- Support for those **audio formats** on **any platform**:
  - MP3
  - OGG
  - XM
  - IT
  - MOD
  - S3M
  - WAV
  - AIFF
- **Export** audio from a **turntable** or the whole session
- Read the **lyrics** of the **current record**
- Open **Spotify** with the **current record**
- Tune into **multiple turntables** at the **same time** (and blend between records)
- Reads **tags** like **ID3**
- **Reads** and **saves** M3U, PLS and XSPF files
- **Crossfade** between **turntables** and **records**
- **Loop** between **start** and **end** position
- **Events** for certain positions
- Analyze the **BPM** of a record[1]
- **Cache records** to play them with no delay (e.g. a piano)
- **Watchdog** for file system changes
- **Performance**: Very low impact on performance!
- **No limits**: Does survive changing scenes! The music is not interrupted even during a load operation if necessary.

2.2. Filter & order

Filter and **order** (ascending/descending) the **records** by:

- Name
- Title
- Artist
- Album
- Genre
- Rating
- Year
- Duration
- Path
- Size
- Format

1 Not supported for MP3-audio in standalone builds and the Unity Editor

2.3. Documentation & control

- **Test** all audio **records** inside the **editor**
- Powerful **API** for **maximum control**
- Detailed **demo scenes**
- Comprehensive **documentation** and **support**
- Full **source code** (including libraries)

2.4. Compatibility

- Supports **all build platforms**
- Works with **Windows**, **Mac** and **Linux editors**
- Compatible with **Unity 2019.4 – 2023**
- Supports **AR** and **VR**
- **C# delegates** and **Unity events**
- **OnRadio** integration for song art
- Works great with **Radio**
- Works with **Online Check**
- Works with **File Browser**

2.5. Integrations

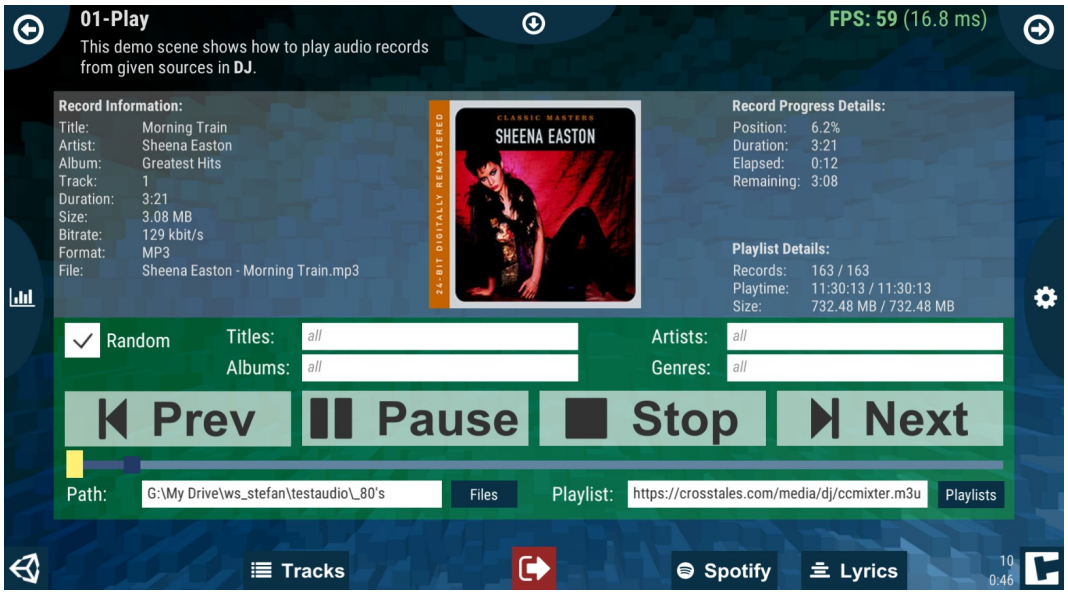
- Audio Visualizer
- Complete Sound Suite
- PlayMaker
- Visualizer Studio
- Apollo Visualizer Kit
- Rhythm Visualizator
- Volumetric Audio

3. Demonstration

The asset comes with many demo scenes to show the main usage.

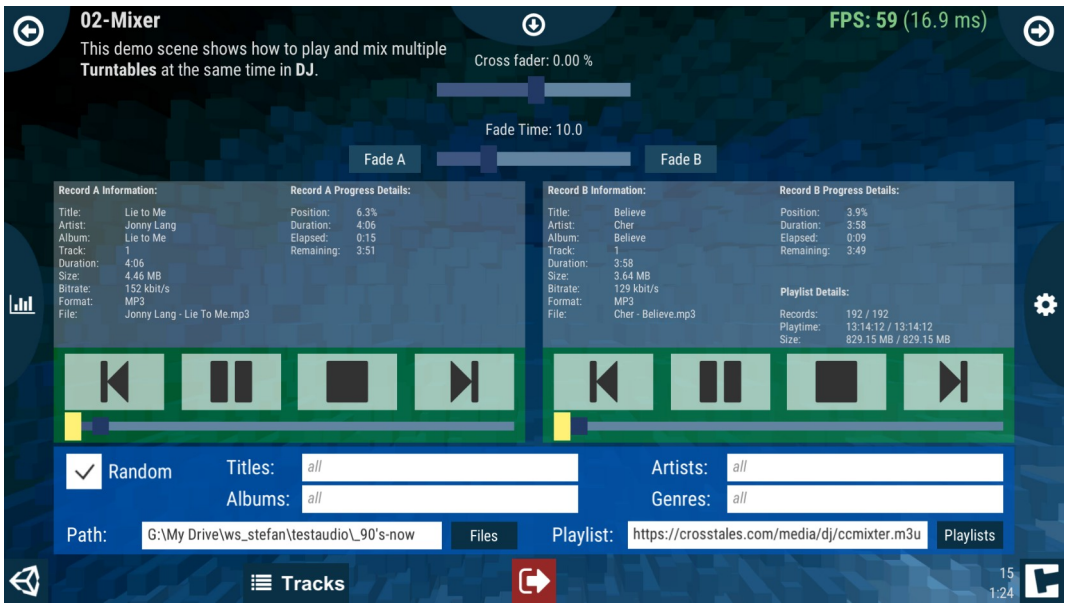
3.1. Play

This demo scene shows how to play audio records from given sources.



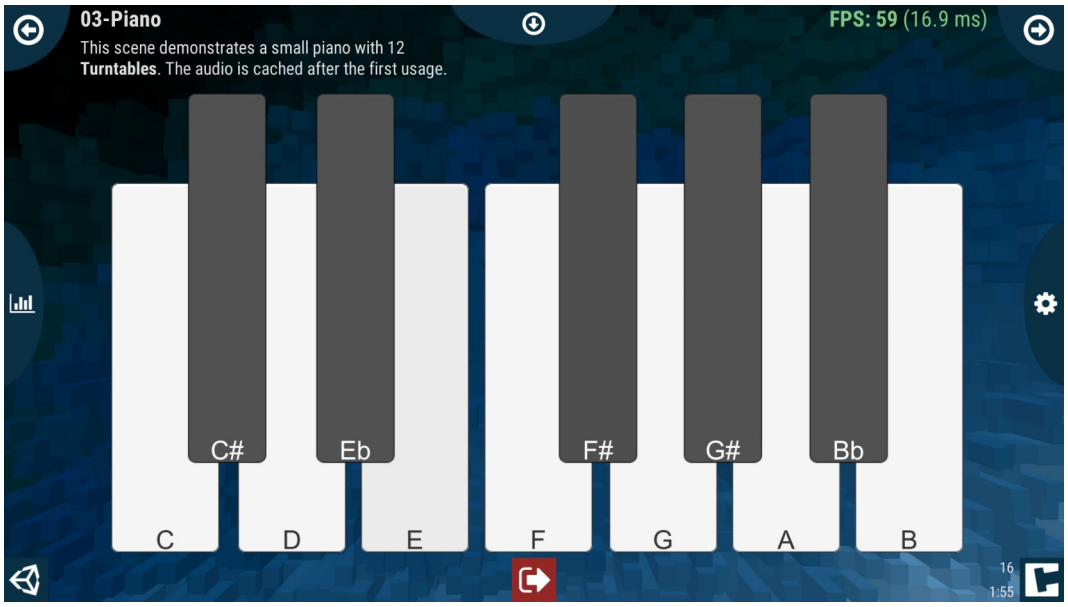
3.2. Mixer

This demo scene shows how to play and mix multiple **Turntables** at the same time.



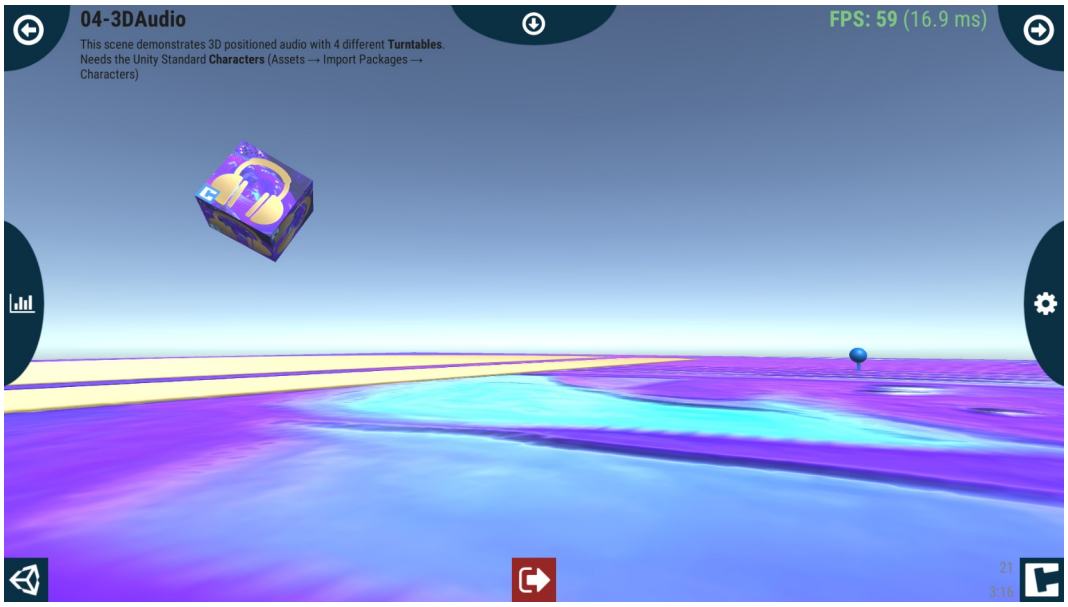
3.3. Piano

This scene demonstrates a small piano with 12 **Turntables**. The audio is cached after the first usage.



3.4. 3DAudio

This scene demonstrates 3D positioned audio with 4 different **Turntables**.

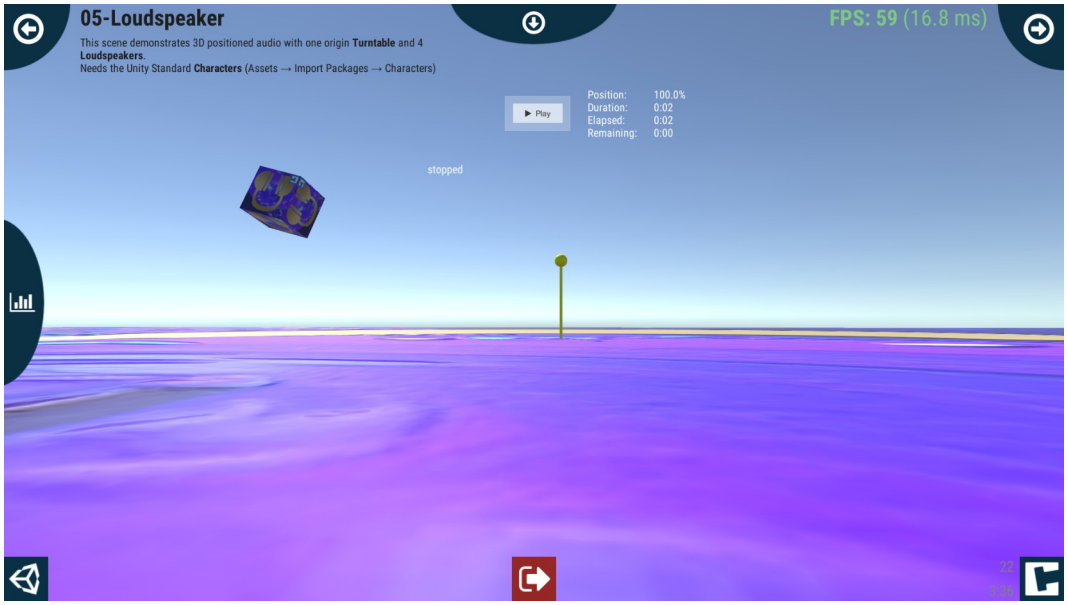


Needs the Unity Standard Assets-package.

3.5. Loudspeaker

This scene demonstrates 3D positioned audio with one origin **Turntable** and 4 **Loudspeakers**.

Needs the Unity Standard Assets-package.



3.6. PlaySingleFile

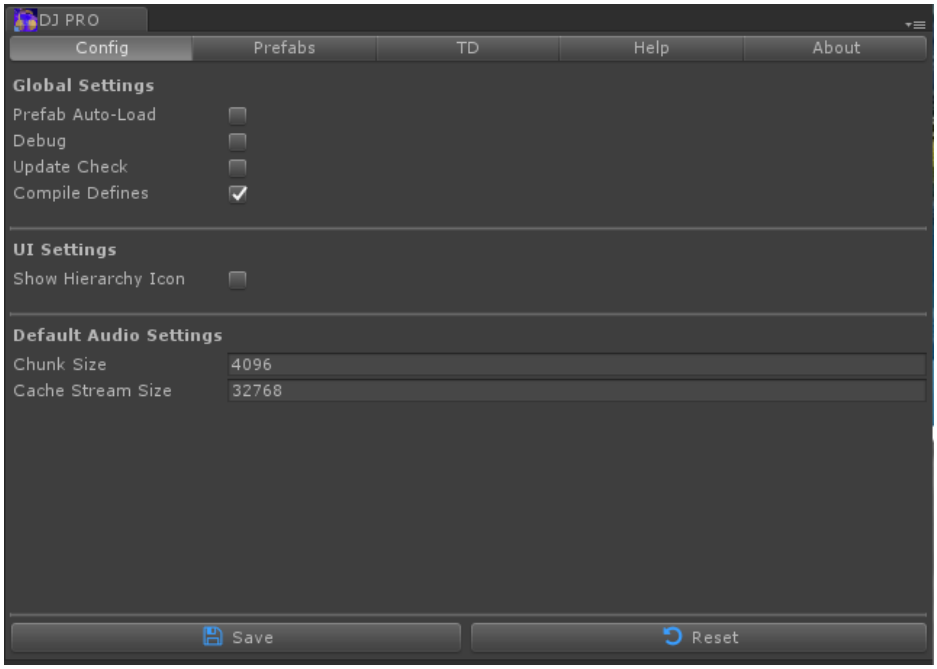
This demo scene shows to play a single audio record with a Turntable.

3.7. RecordList

This demo scene shows how to play records with a Turntable.

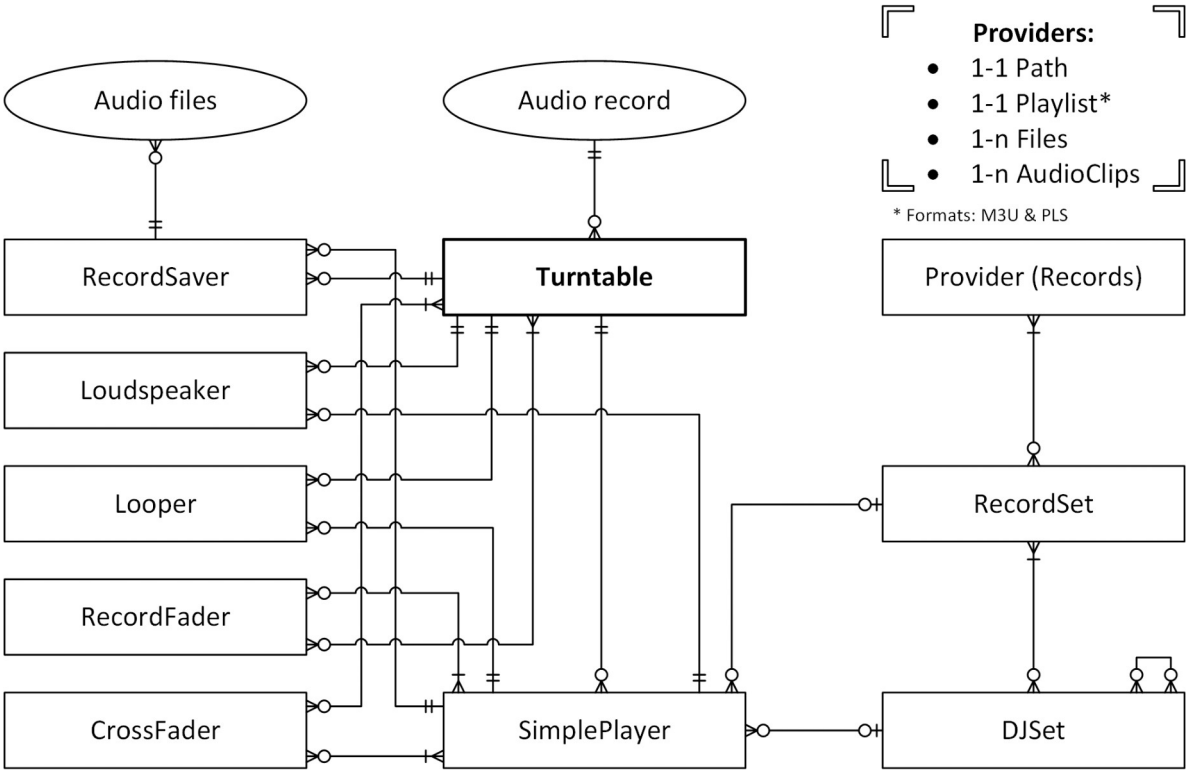
4. Setup

"DJ PRO" has global settings under "Edit\Preferences..." and under "Tools\DJ PRO\ Configuration...":



4.1. Schema

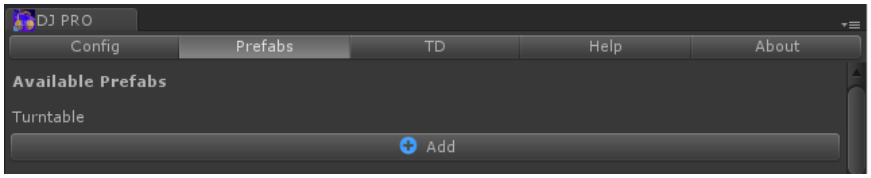
The following graphic explains the relationships between all relevant components:

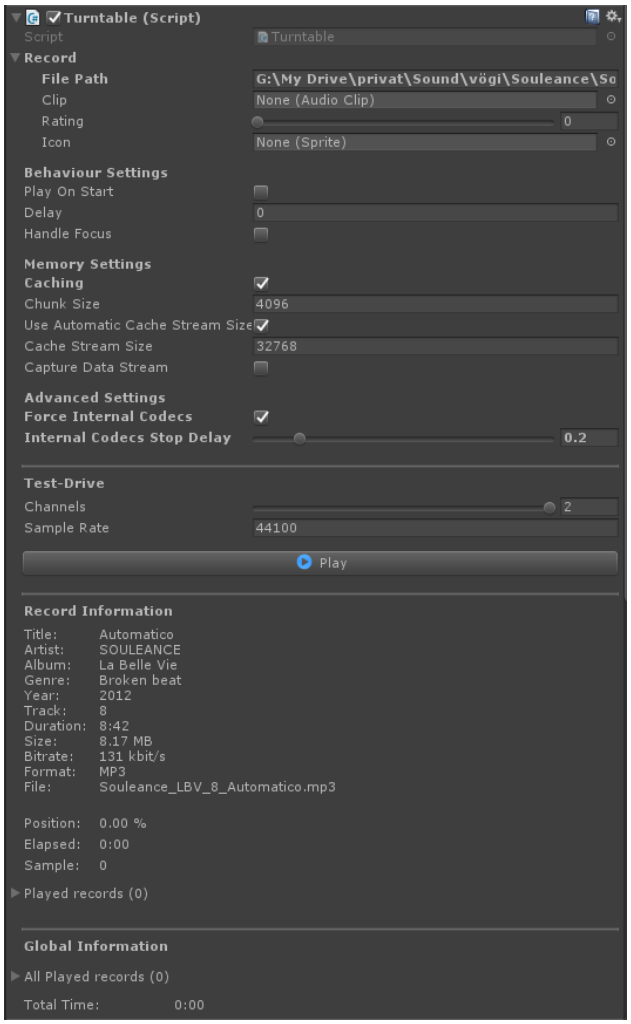


4.2. Turntable

There are four ways to use a single turntable:

1. Add the prefab **Turntable** from Assets/Plugins/crosstales/DJ/Resources/Prefabs to the scene
2. Or go to *Tools => DJ PRO=> Prefabs => **Turntable***
3. Right-click in the *hierarchy-window => DJ PRO => **Turntable***
4. Add it from the Prefabs-tab:





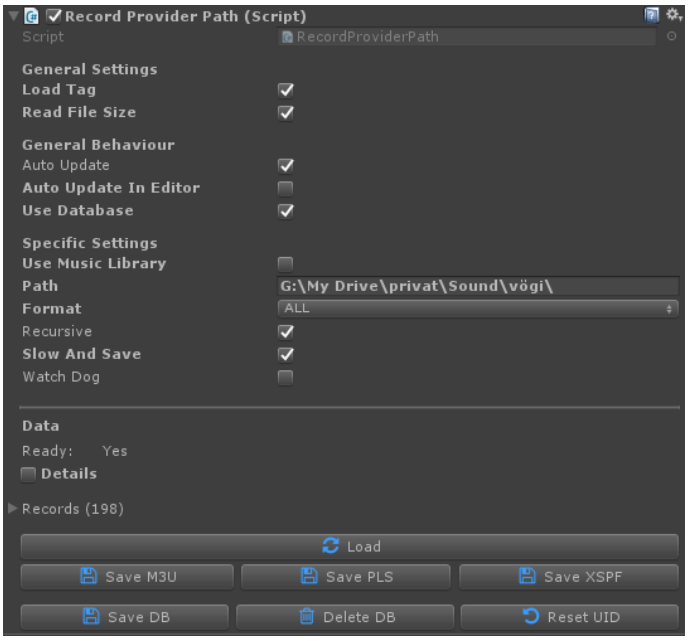
Just enter a file in "File Path" and you're good to go.

4.3. Providers

There are two different types of providers:

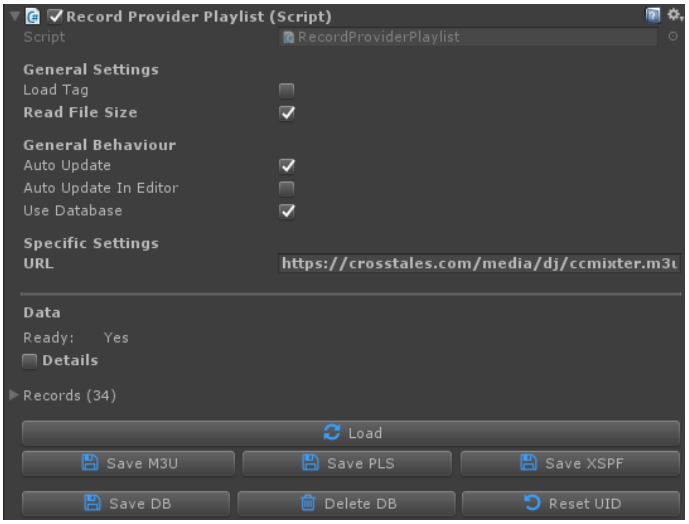
4.3.1. RecordProviderPath

This provider scans a path for audio files.



4.3.2. RecordProviderPlaylist

This provider reads a playlist file (M3U/PLS) with all audio files.



4.3.3. Other providers

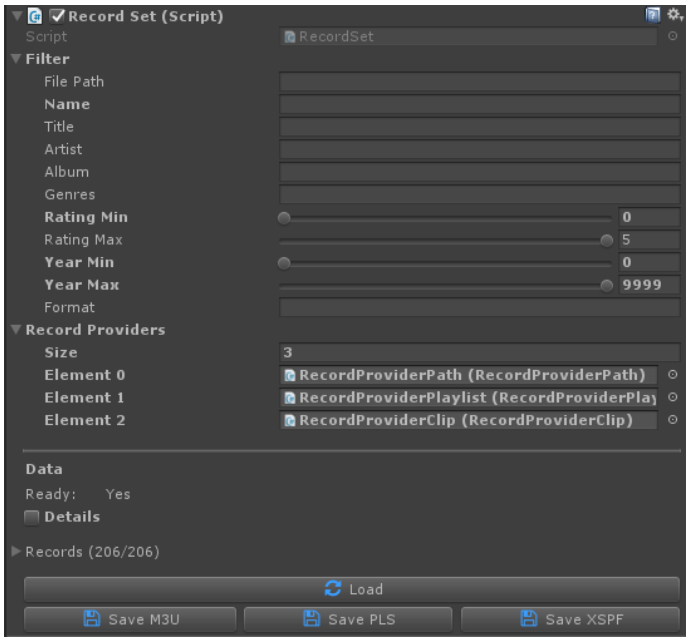
RecordProviderClip and RecordProviderFile manage a list of records.

4.4. Sets

There are two different types of sets:

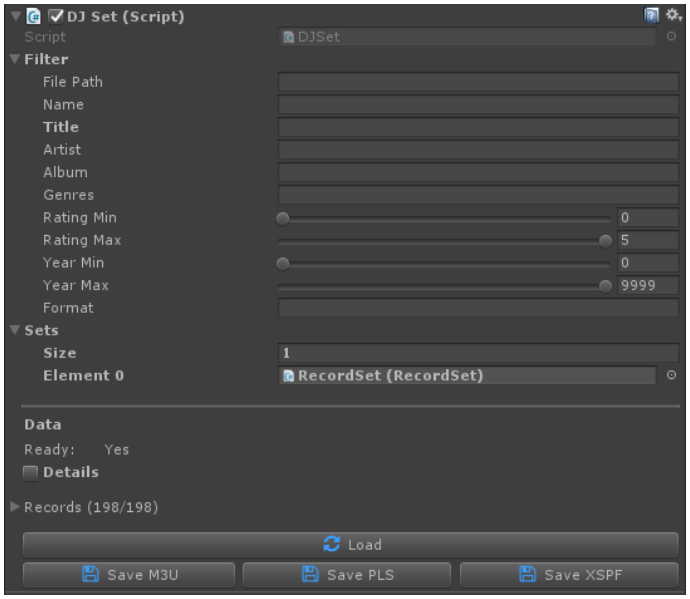
4.4.1. RecordSet

This set consists of 1-n providers.



4.4.2. DJSet

This set consists of 1-n sets.



4.5. SimplePlayer

The "SimplePlayer" combines a "Turntable" and a "Set" and offers functions like "play next record" etc.

4.6. Other components

The other components can be added in the same way as "Turntable"

4.6.1. Loudspeaker

This is useful to use the same player on multiple locations in the game.

4.6.2. RecordSaver

Allows to save songs from a player as WAV files.

4.6.3. CrossFader

Allows to fade between two players.

4.6.4. RecordFader

Allows to fade between records via two players.

4.6.5. Looper

Allows to set loop points in a record.

4.6.6. SurviveSceneSwitch

Allows any Unity gameobject to survive a scene switch. This is especially useful to keep the music playing while loading a new scene.

4.6.7. Eventer

Allows to set event points in a record.

4.6.8. BPMAnalyzer

Analyze the BPM of an audio record.

This is **not supported** for **MP3**-audio in **standalone builds** and in the **Unity editor**.

5. API

The asset contains various classes and methods. The most important ones are explained here.

Make sure to **include** the **name space** in the relevant source files:

```
using Crosstales.DJ;
```

5.1. Player

These are the important methods for the different players (**Turntable** and **SimplePlayer**).

5.1.1. Play

TBD – add all classes and methods

5.2. Callbacks

There are various callbacks available. Subscribe them in the "Start"-method and unsubscribe in "OnDestroy".

5.2.1. Playback start and end

```
PlaybackStart(Record record);
PlaybackStart OnPlaybackStart;
```

Triggered whenever the playback starts.

```
PlaybackEnd(Record record);
PlaybackEnd OnPlaybackEnd;
```

Triggered whenever the playback ends.

5.2.2. Buffering start, end and progress

```
BufferingStart(Record record);
BufferingStart OnBufferingStart;
```

Triggered whenever the buffering starts.

```
BufferingEnd(Record record);
BufferingEnd OnBufferingEnd;
```

Triggered whenever the buffering ends.

```
BufferingProgressUpdate(Record record, float progress);
BufferingProgressUpdate OnBufferingProgressUpdate;
```

Triggered whenever the buffering progress changes.

5.2.3. Audio start, end and time

```
AudioStart(Record record);
AudioStart OnAudioStart;
```

Triggered whenever the audio starts.

```
AudioEnd(Record record);
AudioEnd OnAudioEnd;
```

Triggered whenever the audio ends.

```
AudioPlayTimeUpdate(Record record, float playtime);
AudioPlayTimeUpdate OnAudioPlayTimeUpdate;
```

Triggered whenever the audio playtime changes.

5.2.4. Errors

```
ErrorInfo(Record record, string info);
ErrorInfo OnErrorInfo;
```

Triggered whenever an error occurs.

5.2.5. Filter change

Callback for **Sets** and **SimplePlayer**.

```
FilterChange();
FilterChange OnFilterChange;
```

Triggered whenever a filter changes.

5.2.6. Records change

Callback for **Sets** and **SimplePlayer**.

```
RecordsChange();
RecordsChange OnRecordsChange;
```

Triggered whenever a the audio records change.

5.2.7. Record change

Callback for **SimplePlayer**.

```
RecordChange(Record record);
RecordChange OnRecordChange;
```

Triggered whenever another record is played.

5.2.8. Example

```
void Start() {
 // Subscribe event listeners
 Turntable.OnPlaybackStart += playBackStart;
 Turntable.OnPlaybackEnd += playBackEnd;
 Turntable.OnAudioPlayTimeUpdate += audioPlayTime;
}

void OnDestroy() {
 // Unsubscribe event listeners
 Turntable.OnPlaybackStart -= playBackStart;
 Turntable.OnPlaybackEnd -= playBackEnd;
 Turntable.OnAudioPlayTimeUpdate -= audioPlayTime;
}

private void playBackStart(Record record) {
 Debug.Log("Playback started");
}

private void playBackEnd(Record record) {
 Debug.Log("Playback ended");
}

private void audioPlayTime(Record record, float playtime) {
 Debug.Log("Playtime: " + Helper.FormatSecondsToHourMinSec(playtime));
}
```

5.3. Complete API

Please read the DJ-api.pdf for more details.

6. Third-party support (PlayMaker etc.)

"DJ PRO" supports various assets from other publishers. Please import the desired packages from "Assets/Plugins/crosstales/DJ/3rd party".

7. Verify installation

Check if DJ is installed:

```
#if CT_DJ
      Debug.Log("DJ installed: " + Util.Constants.ASSET_VERSION);
#else
      Debug.LogWarning("DJ NOT installed!");
#endif
```

8. Upgrade to new version

Follow this steps to upgrade the version of "DJ PRO":

1. Update "DJ PRO" to the latest version from the "Unity AssetStore"
2. Inside the project in Unity, go to menu "File" => "New Scene"
3. Delete the "Assets/Plugins/crosstales/DJ" folder from the Project-view
4. Import the latest version from the "Unity AssetStore"

9. Important notes

9.1. Android

Android 10 and higher needs additional permissions to read the file system.

Add this attribute to "AndroidManifest.xml":

```
<manifest ... >
  <application android:requestLegacyExternalStorage="true" ... >
    ...
  </application>
</manifest>
```

Or even better - install "Runtime File Browser" to get full access to the filesystem:

https://assetstore.unity.com/packages/slug/113006?aid=1011lNGT

9.2. UWP (WSA)

UWP is a sandboxed system, this means, it's not allowed to browse the file system or read any files from your local machine without user consent.

This means, the user has to select a file or path with a NATIVE file selector and it will work.

Therefore, we recommend to use our File Browser PRO.

10. OnRadio

OnRad.io has a massive radio station search engine that plugs into DJ PRO providing a wide array of audio to enrich any game play.

There's an infinite way to play music by artist, song title, genres or stations that provide players great audio satisfaction.

OnRad.io is an annual service that maintains a list of 100'000 radio stations that are constantly changing.

As an OnRad.io customer you can rely on our meticulously maintained station list and a unique search engine that allows for searching by song or artist as the songs are played on radio.

All customers receive a unique partner token then will allow for queries that can look up song or station data and get streaming URLs so those stations can be played inside any Unity app using DJ PRO.

There are 3 price tiers for the OnRad.io for DJ PRO based on the number of DAILY API queries desired. To enable OnRad.io service follow these steps:

10.1. Step 1

Purchase the desired OnRad.io tier from https://dar.fm/upgrade.php#radiopro

Pro Bronze:	$99.95/year - 10'000 daily API queries (special DJ PRO price $79.95)
Pro Plus:	$139.95/year - 50'000 daily API queries
Pro Platinum:	$299.95/year - 250'000 daily API queries

10.2. Step 2

Email sales@dar.fm with your receipt number requesting a partner token for DJ PRO.

10.3. Step 3

Insert the partner token received in Step 2 into DJ PRO to enable full access to OnRad.io.

11. Problems, improvements etc.

If you encounter any problems with this asset, just send us an email with a problem description and the invoice number and we will try to solve it.

We will try and make a version for all platforms as well, please bear with us.

12. Release notes

See "VERSIONS.txt" under "Assets/Plugins/crosstales/DJ/Documentation" or online:

https://crosstales.com/media/data/assets/DJ/VERSIONS.txt

13. Credits

"DJ PRO" uses modified versions of:

- NAudio 1.7.2
- NLayer 1.1.0
- NVorbis 0.8.4
- TagLib-sharp 2.1.0.0

The search is provided by Spotify.

The lyrics are provided by AZLyrics.

The audio files for the demos are from CC Mixter.

The song art API is provided by OnRadio.

The icons are based on Font Awesome.

We aren't affiliated to those companies.

Improvements for the M3U-parser:
Robert Bragg

14. Contact and further information

crosstales LLC
Schanzeneggstrasse 1
CH-8002 Zürich

Homepage:	https://www.crosstales.com/en/portfolio/DJ/
Email:	dj@crosstales.com
AssetStore:	https://assetstore.unity.com/lists/crosstales-42213
Forum:	https://forum.unity.com/threads/coming-soon-dj-play-your-music.502627/
Documentation:	https://www.crosstales.com/media/data/assets/DJ/DJ-doc.pdf
API:	https://www.crosstales.com/media/data/assets/DJ/api/
Windows-Demo:	https://drive.google.com/file/d/1-4SPrq-1mN0tGq5chss3WM3I-L9r2tvV/view?usp=sharing
Mac-Demo:	https://drive.google.com/file/d/1-5Et_ClrTHXJgm_iedSt0NBepgvN7qHx/view?usp=sharing
Linux-Demo:	https://drive.google.com/file/d/1-7uarhyrEE0aon3vTCmJ-d0Nt9avgg7C/view?usp=sharing
Android-Demo:	https://drive.google.com/file/d/1-AuAMPrxHHrAMUNknHuahxwYXoi5oaQI/view?usp=sharing

15. Our other assets

3D Skybox	Those beautiful packages contain professional 8k, HDR, stereoscopic 360° real-world skyboxes for your projects.
Bad Word Filter	The "Bad Word Filter" (aka profanity or obscenity filter) is exactly what the title suggests: a tool to filter swearwords and other "bad sentences".
File Browser	File Browser is a wrapper for native file dialogs on Windows, macOS, Linux and UWP (WSA).
Online Check	You need a reliable solution to check for **Internet availability**? Here it is!
Radio	Radio allows implementing free music from Internet radio stations into your project..
RT-Voice	RT-Voice uses the computer's (already implemented) TTS (text-to-speech) voices to turn the written lines into speech and dialogue at run-time! Therefore, all text in your game/app can be spoken out loud to the player.
True Random	True Random can generate "true random" numbers for you and your application. The randomness comes from atmospheric noise, which for many purposes is better than the pseudo-random number algorithms typically used in computer programs.
Turbo Backup	Turbo Backup is the fastest and safest way to backup your Unity project. It only stores the difference between the last backup, this makes it incredible fast.
Turbo Builder	Turbo Builder creates builds for multiple platforms in one click. It works together with Turbo Switch to offer an incredible fast build pipeline.

Turbo Switch	Turbo Switch is a Unity editor extension to reduce the time for assets to import during platform switches. We measured speed improvements up to 100x faster than the built-in switch in Unity.