



**HAL**  
open science

## Software tools for object-based audio production using the Audio Definition Model

Geier Matthias, Thibaut Carpentier, Markus Noisternig, Olivier Warusfel

► **To cite this version:**

Geier Matthias, Thibaut Carpentier, Markus Noisternig, Olivier Warusfel. Software tools for object-based audio production using the Audio Definition Model. International Congress on Spatial Audio, Sep 2017, Graz, Austria. hal-01574183

**HAL Id: hal-01574183**

**<https://hal.science/hal-01574183>**

Submitted on 4 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Software tools for object-based audio production using the Audio Definition Model (ITU-R Recommendation BS.2076)

Matthias Geier, Thibaut Carpentier, Markus Noisternig, Olivier Warusfel  
UMR 9912 STMS IRCAM-CNRS-UPMC

## Abstract

We present a publicly available set of tools for the integration of the Audio Definition Model (ADM) in production workflows. ADM is an open metadata model for the description of channel-, scene-, and object-based media within a Broadcast Wave Format (BWF) container. The software tools were developed within the European research project ORPHEUS (<https://orpheus-audio.eu/>) that aims at developing new end-to-end object-based media chains for broadcast. These tools allow for recording uncompressed multichannel audio files with embedded XML metadata according to the ADM specifications, and monitoring and rendering existing BWF-ADM files over headphones or arbitrary loudspeaker setups. The bundle also includes a plugin (available in all major plugin formats) for interoperability with digital audio workstations, which provides support for time-varying metadata via automation tracks.

## Introduction

The tools presented here were originally developed for experimenting with incorporating reverb into an object-based production workflow, within the scope of the ORPHEUS project [1]. However, they can also be used in a more general context to create, load and play back audio files containing ADM metadata. In its current state, only a subset of the ADM specification [2] is supported, but many common use cases are already covered. The current limitations, as well as the ADM format itself, are described below.

## The ORPHEUS Project

The European research project ORPHEUS [1] is dedicated to improving the production, distribution and reproduction of audio content. It is developing, implementing and validating a new end-to-end object-based media chain for audio content.

Object-based media is a novel approach for creating and deploying interactive, personalised, scalable and immersive content, by representing it as a set of individual assets together with metadata describing their relationships and associations. This allows media objects to be assembled in ground-breaking ways to create new user experiences.

ORPHEUS started in December 2015 and has a duration of 30 months. It receives funding from the European Commission under the Horizon 2020 programme.

## The Audio Definition Model

Recommendation ITU-R BS.2076 [2] – the Audio Definition Model (ADM) – is a metadata model for describing the technical format and content of audio files and streams. It primarily uses XML as its format language, and has been designed for incorporation into RIFF-based audio files like BWF [3] and BW64 [4] files. XML data is stored in a chunk of type `<axml>`, while the association of audio tracks with the identifiers used in the XML description is laid out in a chunk of type `<chna>`.

Each audio channel has one of the pre-defined types “Objects”, “DirectSpeakers”, “Binaural”, “HOA” or “Matrix”. Tracks of type “Objects” have an associated position, gain and several other parameters. All these parameters can potentially change over time. Linear interpolation between adjacent positions is available. The type “DirectSpeakers” can be used to include channel-based beds. Every channel represents one loudspeaker feed and it has the position of the loudspeaker associated with it. Pairs of channels of type “Binaural” simply hold the left and right channel of a binaural mix. The “HOA” type can be used to include different kinds of Ambisonics mixes. Channels of type “Matrix” contain gain and phase shift values that can be used to implement, for example, mid-side stereophony.

A single ADM file can contain multiple “audio programmes”, which are different subsets of all available

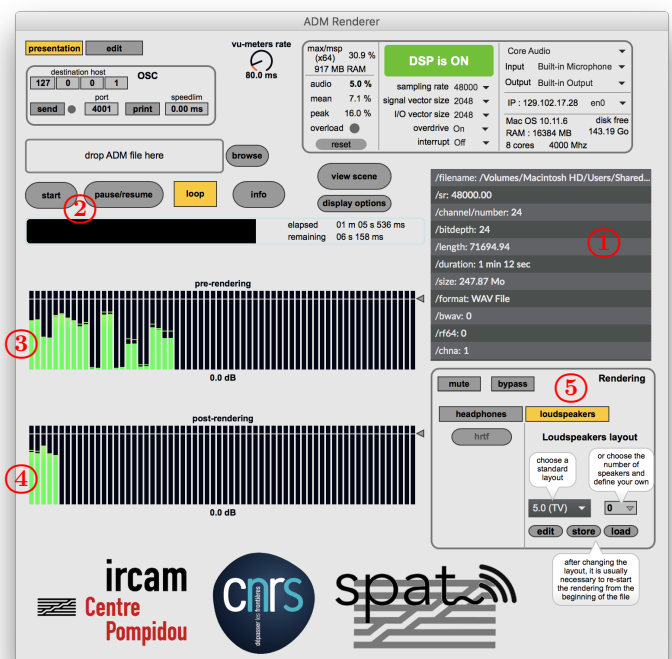
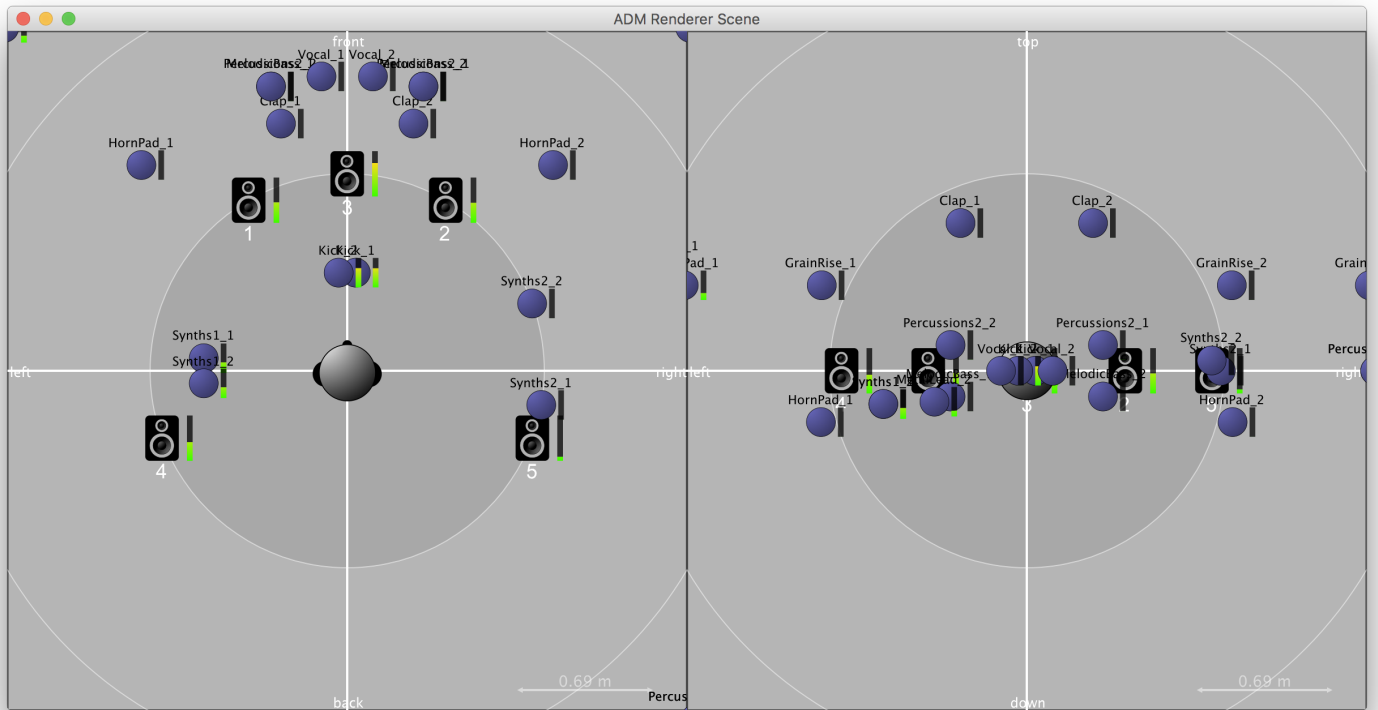


Figure 1: Main user interface of the ADM Player/Renderer



**Figure 2:** Scene display of the ADM Renderer. A very similar screen is used in the ADM Recorder. Left pane: view from above; right pane: view from behind. The displayed ADM file is `machine_aer_bwf.wav` from <http://www.bbc.co.uk/rd/publications/saqas>.

audio tracks and metadata, to be chosen by the consumer at rendering time. This can for example be used to provide parts of the material in multiple languages, while language-independent content like background music can be re-used in multiple “audio programmes”.

## The ADM Tools

The ADM tools are stand-alone programs, freely available for macOS and Windows. Currently, there are three tools available: a player with built-in renderer, a recorder which is using the same renderer for monitoring and a tool called “ADM ExtractXML” that can be used to get to the XML data stored inside a BWF/ADM file. The three tools are described in more detail in the following sections.

The ADM Player and the ADM Recorder can communicate with other programs via Open Sound Control (OSC) [5]. Note that OSC communication is not part of the ADM specification. The ADM Tools use an ad-hoc OSC syntax which facilitates the inter-app communication and can be easily used with other OSC-aware software/devices. The “Tosca” plugin, described further below, can serve as an ancillary tool to connect a Digital Audio Workstation (DAW) to the ADM Recorder via OSC.

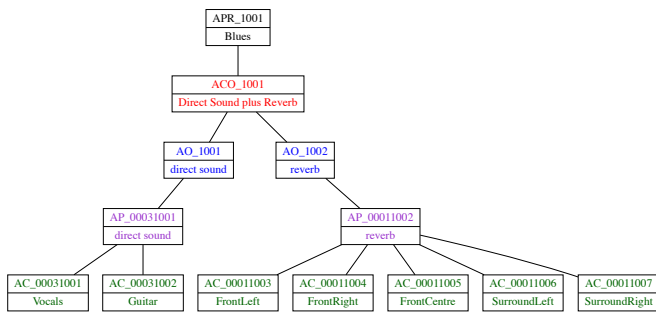
The ADM Tools can be downloaded from <https://www.ircam.fr/project/detail/orpheus-1/> and the Tosca plugin for AU, VST, VST3 and AAX is available at <http://forumnet.ircam.fr/product/spat-en/tosca-en/>.

## ADM Player + Renderer

The ADM Player can load WAV/BWF/BW64 files and read ADM metadata and channel assignments from their respective chunks in the file. Upon loading a file, it performs a number of “sanity checks” on the ADM metadata. This can be very helpful to check whether a given ADM file is compliant. The ADM Player can play back the audio content of the file and produce synchronized control messages based on the ADM metadata obtained from the file. Using this audio data and the metadata, the built-in renderer can calculate appropriate signals for real-time reproduction using headphones or loudspeaker setups.

The main user interface is shown in figure 1. Summary information about the content of the currently open file (number of objects, number of tracks, duration, etc.) is displayed in the list on the right ①. Controls for playback (start/stop/resume/loop/seek) are located on the left ②. Below them, there are level meters for each audio track in the file ③ and below these there are level meters for the rendered output channels ④. On the lower right, the rendering settings can be chosen ⑤.

Both loudspeaker-based and headphone-based reproduction is possible. For rendering on loudspeakers, VBAP [6] is used. The loudspeaker setup can be chosen from a predefined set of common loudspeaker setups or by manually specifying custom loudspeaker positions. Up to 64 loudspeakers are supported. For binaural rendering, Head-Related Transfer Functions (HRTFs) can be selected from a large set of available HRTF datasets. Further datasets can be loaded using the SOFA/AES69 format [7].



**Figure 3:** Diagram of the XML metadata structure of an ADM file, generated with the tool *ADM ExtractXML*. The IDs shown in the boxes are used internally to create connections between ADM elements.

In the top left corner of the user interface, OSC messages can be enabled which can for example be used to control an external renderer (while switching the built-in renderer to “bypass” mode) or further process the metadata generated by the ADM Player. The OSC messages can be sent to other applications running either on the same computer or on another computer on the network.

The “view scene” button opens a window showing the current position of all audio “objects” in the ADM file, as shown in figure 2. The scene window also allows interactively moving objects with the mouse.

In its current state, the ADM Player/Renderer only supports a subset of the ADM specifications. Most notably, it doesn’t support multiple “audio programmes”, nested ADM elements and non-PCM audio data.

## ADM ExtractXML

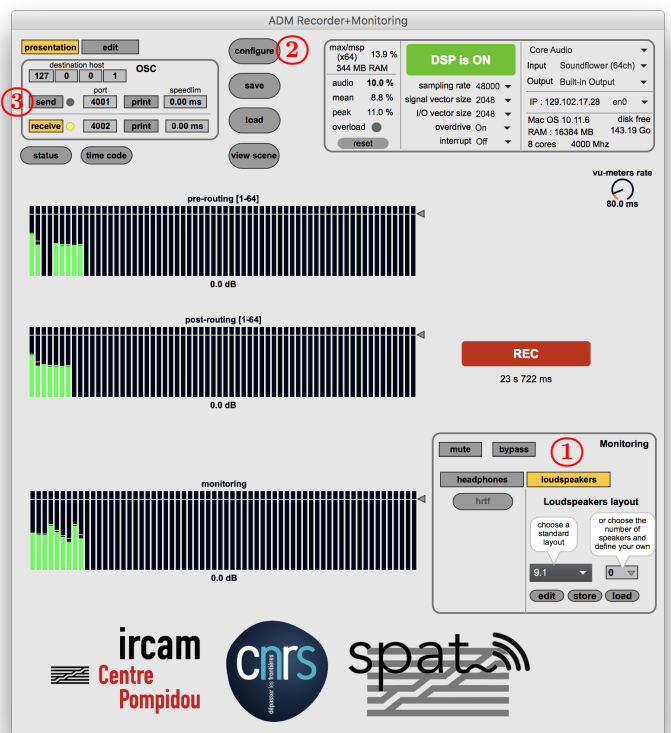
Given a sound file with ADM metadata, this tool can be used to create a separate XML file containing only the ADM metadata (i.e. the contents of the `<axml>` chunk). The XML file will be stored in the same directory as the original ADM file. The contents of the `<chna>` chunk are also stored in a separate file. Like the ADM Player, this tool runs several “sanity checks” on the metadata to make sure it complies with the ADM specification [2].

At the same time as extracting the `<axml>` chunk, the tool also creates structural representations of the XML elements in the DOT graph description language [8]. If the *Graphviz*<sup>1</sup> software is installed, PNG images with those graphs are automatically created. See figure 3 for an example graph.

## ADM Recorder + Monitoring

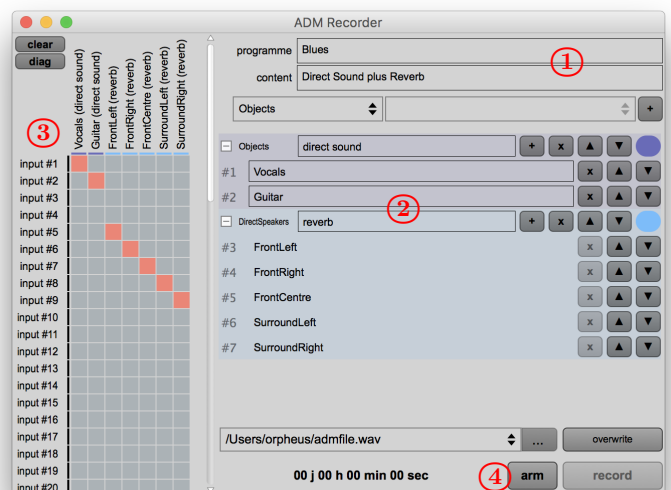
The ADM Recorder can record up to 64 audio channels into a BWF/BW64 file. For each of these channels, it can also record automation data which is stored as ADM metadata in the same file. This automation data can be received via OSC messages.

The ADM Recorder has a built-in renderer that can be used for monitoring.



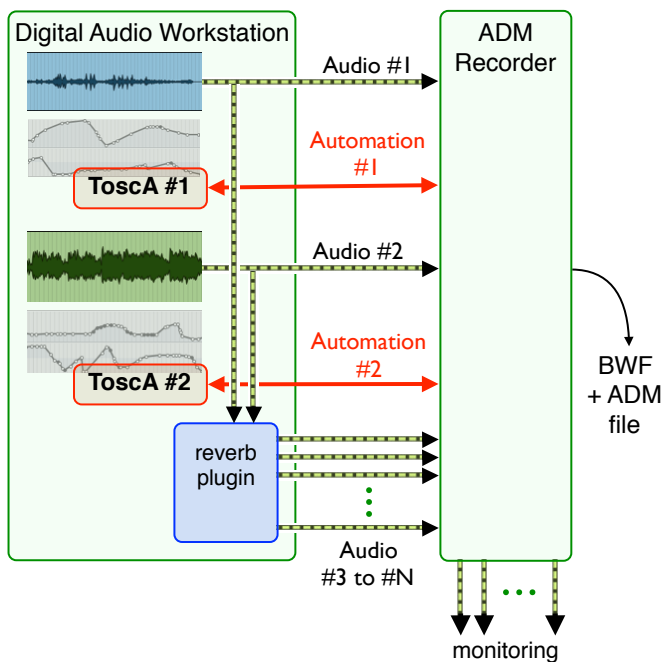
**Figure 4:** Main window of the ADM Recorder

Figure 4 shows the main window of the ADM Recorder. The monitoring settings ① in the bottom right of the window are the same as for the ADM Player/Renderer described above. The “configure” button ② can be used to open a window for editing ADM metadata as shown in figure 5. On the top of this window, the programme and content name can be specified ①. Below that, a list of “audio packs” ② can be set up. Each pack can contain one or more “channels”. For the “DirectSpeakers” type, one of a set of pre-defined channel-based setups (e.g. stereo, 5.1, 22.2) can be selected which populates the channels appropriately. A “Binaural” pack automatically contains two channels, one for the left and one for the right ear. Each channel has a unique channel number that can be used together with the routing matrix on the left side ③



**Figure 5:** Metadata editor and routing matrix. The resulting ADM structure for this example can be seen in figure 3.

<sup>1</sup><http://www.graphviz.org/>



**Figure 6:** Using the TosCA plugin to connect a Digital Audio Workstation (DAW) to the ADM Recorder

to assign input audio channels to the desired ADM tracks.

Once the packs and channels are set up as desired, the “arm” button ④ can be used to enable recording. This will open a scene display window similar to figure 2 which allows configuration of the initial positions of all audio objects. The scene window also allows moving objects with the mouse during recording, which will record the object trajectories in the resulting ADM file. However, in practice it will be more convenient to connect the ADM Recorder to a Digital Audio Workstation (see the section about the “TosCA” plugin below) which can record automation data for multiple objects (possibly one after another). The DAW can then play back all automation tracks at once, which in turn can be recorded by the ADM Recorder. A track with an LTC time code signal can be used to synchronize start and stop times between the DAW and the ADM Recorder. Instead of using a DAW, the automation data can of course be produced by any software, as long as the generated OSC messages are compatible with the ADM Recorder.

In the top left of the main window (figure 4), host and port settings ③ can be configured for receiving and sending OSC messages that can transport ADM metadata between applications and between computers. Messages which are received via this interface are stored as XML metadata in the recorded ADM file. This interface can also be used to establish a connection with the TosCA plugin.

The ADM Recorder only supports a subset of the ADM specifications. It doesn’t support multiple “audio programmes” nor audio channels with different lengths. All audio content will have the same duration.

## DAW Automation with TosCA

TosCA [9] is a plugin for Digital Audio Workstations (DAWs) that allows the recording of automation tracks for arbitrary parameters that can be sent and received as OSC messages via a network interface. Primarily developed for the remote control of spatial audio processors, this tool also turns out to be very handy for “linking” the ADM tools with a DAW. It can be used together with the ADM Recorder to record movements and changing gain values as automation tracks in a DAW.

An instance of the TosCA plugin has to be inserted into every track that is supposed to contain automation data, see figure 6. Each TosCA instance can then be configured with a channel ID that is used to associate tracks in the DAW with channels defined in the ADM Recorder. The ADM Recorder will in turn store the recorded automation data in the ADM file.

The audio outputs of the DAW tracks can be provided to the ADM Recorder either via hardware connections or via virtual sound drivers. Refer to [9] for additional details.

## Future Work

As mentioned above, not all features of the ADM specification are implemented yet. Further tools are planned that will allow to select audio tracks and edit metadata of ADM files as well as to merge the contents of multiple files into one. This way, ADM files could be created that hold multiple “audio programmes”, for example to enable selection between multiple language versions.

Preliminary support for HOA is provided, however this may evolve in future versions as future versions of the ADM specification are expected to contain significant changes in this area.

## Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 687645.



## References

- [1] Andreas Silzle, Robert Sazdov, Michael Weitnauer, Olivier Warusfel, Werner Bleisteiner, Peter Vanselow, Jean-Yves Aubie, Benjamin Duval, Niels Bogaards, Chris Baume, and Uwe Herzog. The EU project ORPHEUS: Object-based broadcasting – for next generation audio experiences. In *Tonmeistertagung*, 2016.
- [2] ITU-R. Recommendation BS.2076: Audio Definition Model, 2015. <https://www.itu.int/rec/R-REC-BS.2076/>.
- [3] EBU. Tech 3285: BWF - A format for audio data files in broadcasting, 2011. <https://tech.ebu.ch/publications/tech3285/>.

- [4] ITU-R. Recommendation BS.2088: Long-form file format for the international exchange of audio programme materials with metadata, 2015. <https://www.itu.int/rec/R-REC-BS.2088/>.
- [5] Matthew Wright. The Open Sound Control 1.0 specification, 2002. [http://opensoundcontrol.org/spec-1\\_0/](http://opensoundcontrol.org/spec-1_0/).
- [6] Ville Pulkki. Virtual sound source positioning using Vector Base Amplitude Panning. *Journal of the Audio Engineering Society*, 45(6):456–466, 1997.
- [7] Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yôiti Suzuki, Kanji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, and Markus Noisternig. Spatially Oriented Format for Acoustics: A data exchange format representing Head-Related Transfer Functions. In *134<sup>th</sup> AES Convention*, 2013.
- [8] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30(11):1203–1233, September 2000.
- [9] Thibaut Carpentier. Tosca: An OSC communication plugin for object-oriented spatialization authoring. In *41<sup>st</sup> International Computer Music Conference*, 2015. <https://hal.archives-ouvertes.fr/hal-01247588v1/>.