

# **TURGEN**

## **Beginner's Guide**

---

Michael Kalouš of the BAKTRA Software

# Introduction

---

## What you can do with TURGEN?

- Create your own tapes with software for Atari 8-bit computers using a PC/Mac and an ordinary tape recorder. A similar method was used by reputable software publishers in the golden times of Atari.
- Transfer data from PC/Mac to Atari 8-bit computers using a data recorder and a cassette adapter
- Process tape images (.CAS files)

## How creation of tapes work?

In a nutshell, creating tapes with TURGEN is a four-step process.

1. Get software in a format that is transferable to tape (binary load files - .XEX, .COM, .OBJ, tape images - .CAS)
2. Select one of the supported systems for storing data on tapes
3. Generate signal (the signal typically sounds like screeching) and store it to a WAVE file or send it directly to the sound card
4. Record the signal to tape using an ordinary tape recorder

## What Atari software is transferable to tapes?

When speaking of Atari software, we are speaking of *digital images* of the software.

**The Good News.** There is a lot of Atari software transferable to tapes. Most of the games or entertainment software is available in a form transferable to tape.

**The Bad News.** A lot doesn't mean all. The productivity software packages typically require a disk drive, which shouldn't be a surprise. Some software is also too big, so transferring it to tape is impractical, because of excessive loading time.

What is transferable?

- Binary load files (.XEX, .COM, .OBJ). These files contain executable code and data
- Tape images (.CAS). These files represent an exact image of a tape
- Atari BASIC files (.BAS). These files contain Atari BASIC programs

What is not transferable?

- Disk images (.ATR, .XFD, .DCM). These files represent an exact image of a disk. Obviously, you cannot magically "convert" one medium (disk) to another (tape).
- Cartridge images (.ROM, .CART). These files represent an exact image of a cartridge. Again, you cannot magically convert a cartridge to tape.

Sometimes, you can extract files from a disk image, or convert a cartridge image to a binary load file. These are advanced practices, though.

# Where to find transferable software?

Fortunately, there are large archives of Atari software, where a lot of software is available as binary load files.

Good sources of binary load files are the following:

- a8.fandal.cz (Games and demos, the archive focuses on the quality of the images)
- atarionline.pl (Site hosts two archives of games and demos, the focus is on quantity)

## Binary load files

---

Since the binary load files (or shortly binary files) are most useful for transferring to tapes, it is essential to know about them.

## Purpose and origin

The purpose of the binary load files is to store machine code and data in a compact form that has a simple and well-defined structure. In most cases, the binary load files are used to store executable forms of programs.

A good example of a fully exploited binary load file is a game program. Such binary load file typically contains both machine code (game engine), and data (graphics, character sets, levels, music).

The binary load files as we know them now were introduced early, with Atari's DOS II. The previous version of Atari's DOS also supported binary load files, but their format was different.

There is no dedicated extension for the binary load files. Traditional extensions are: .COM, .OBJ, and .SYS. In the modern era, the .XEX extension is the most popular.

## Structure of the binary load file

**Header.** Every binary load file begins with a header. The header is just two bytes – \$FF \$FF. The header simply says “Hello, I am a binary load file”. If a program is about to process a binary load file and the first two bytes are not \$FF, then you get an error message.

**Segments.** A binary load file consists of one or many segments (sometimes also called sections). A segment represents a contiguous block of data that is to be loaded to a specific area of the computer's memory. Each segment begins with a segment header and holds segment data.

**Segment header.** The segment header is four bytes long and contains two addresses. Address of the first byte of the segment (FIRST), Address of the last byte of the segment (LAST) The segment header determines where to store data of the segment in the computer's memory. A segment header can be optionally preceded with two \$FF bytes.

**Segment data.** Segment data immediately follow the segment header. If you are good at arithmetic, you already know that the data is (LAST-FIRST+1) bytes long. A segment is always at least one byte long.

**Rules.** There is no defined limitation on how many segments a binary load file can contain. You are limited just by the capacity of the external storage media (floppy disk, hard disk). Segments can use overlapping addresses. No one prevents you from loading one segment to addresses \$2000-\$3000 and later loading another segment to addresses \$2100-\$3100.

# Loading Binary Load Files

**Binary load.** The process of loading binary load files to the computer's memory is called *Binary Load*. This process is performed by *binary loaders* – routines or programs that load the binary load files. For example, Atari DOS II contains a binary loader (activated by selecting the L. BINARY LOAD menu item).

The binary loader verifies the header, then reads segments and places segment data to memory locations determined by the segment headers.

**Special Addresses.** There are two vectors (2-byte addresses) with special meanings.

RUNAD - \$02E0, \$02E1

After the binary loader loads all segments of the binary load file (when the work of the binary loader is finished) it performs an indirect jump - JMP (\$02E0). If the binary load file contains a machine code to be executed once the binary load file is completely loaded, then the binary load file contains a segment loaded to RUNAD. The segment contains the address of the first instruction of that machine code.

INITAD - \$02E2, \$02E3

When the binary loader loads a segment that changes bytes of the INITAD vector, it performs an indirect jump JMP (\$02E2) and executes machine code pointed to by the INITAD vector. The machine code can return control to the binary loader. When the control is returned (RTS instruction), the binary loader continues loading of the binary load file. The INITAD vector is used to execute machine code during the binary load process.

What is the difference between RUNAD and INITAD?

- RUNAD is used only once; after all segments of the binary load file are already loaded.
- INITAD is used as many times as there are segments that change bytes of the INITAD vector.

**RUN and INIT segments.** A segment that only loads two bytes to RUNAD is called a RUN segment. A segment that only loads two bytes to INITAD is called an INIT segment.

# Creating your first tape

---

This guide will help you create your first tape with TURGEN.

You will record a small demo program on tape using standard tape records – the original system that Atari used to store data on tapes. This will work with any Atari data recorder.

The demo program runs on every 8-bit Atari computer with at least 16 KB of RAM.

## How the tape is recorded

You are going to create a tape with software for Atari 8-bit computers similar way as it was done by the big software publishers in the golden times of Atari.

Originally, the process of authoring tapes was the following: 1. Create a master recording with an Atari computer and data recorder, possibly using disk to tape copier; 2. Copy the master recording to a high-quality master tape reel; 3. With a big tape duplication machine, copy the signal from the master tape to many cassettes.

With TURGEN, there are just two steps: 1. Create a digital master with TURGEN; 2. Record the master to cassettes using your home tape recorder.

## What is needed

1. An ordinary tape recorder. This can be a tape deck, personal stereo, a decent boombox, etc. The recorder must have some capability to record a signal from an external source (preferably from LINE-IN or AUX input connector, or alternatively from CD or USB flash disk). A good stereo tape deck is always recommended, but a decent personal stereo is appropriate too. It is best (but not necessary) that the device has been manufactured before circa 2010. The newest tape recorders are not of very good quality. You can recognize them, they have a permanent magnet (that is mechanically pushed towards the tape) instead of a traditional eraser head.
2. Tapes. There are still companies selling blank tapes.
3. Cables. You need a cable or cables to connect the output of your sound card with the input of your tape recorder. The cables are available in stores with consumer electronics. Alternatively, you can create WAVE files and burn them as CD audio tracks, or convert them to MP3s and store them on a flash disk or memory card. The use the capability of your tape recorder to read signal from these alternate sources.
4. TURGEN. The tool that helps you with the creation of the tapes.
5. Software for Atari. The software must be in a format that is transferable to tape. So it must be a binary load file (.XEX, .COM, .OBJ), or a tape image (.CAS).

## Before you start

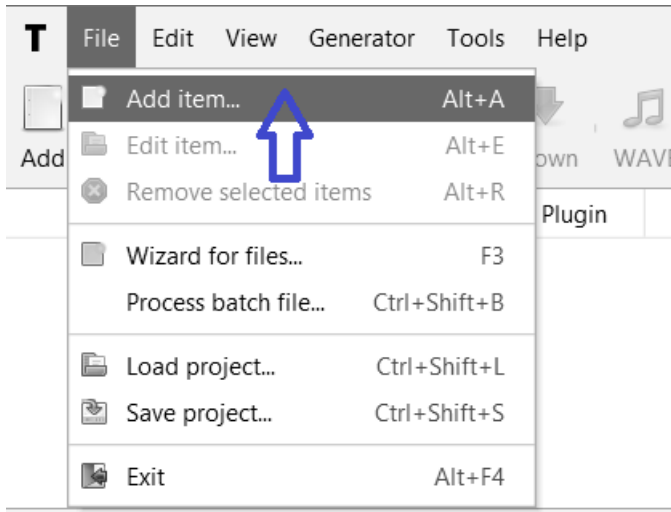
- Download and install the Java Runtime Environment (JRE) or Java Development Kit (JDK). Java is required to run TURGEN. To get JRE and JDK download links, visit the [Downloads](#) page of the project's website.
- Download and install [TURGEN](#), preferably the latest version.

- Locate the test16k.xex binary load file, placed in the *examples* directory of TURGEN's program directory.

## Creating a project item

Run TURGEN and wait until the main window appears.

Go to the *File* menu and select the *Add item* menu item. Alternatively, you can click the *Add* button on the toolbar.



A dialog for adding a project item appears. A project item holds information on how a single file will be stored on a tape. What is the input file, what is the system for storing data on tape and various parameters.

Begin with selecting the system for storing the data on tape. Click the *Plugin* box and select the *Standard* plugin.

This plugin generates data in the format used by the original system that Atari used to store data on tapes, hence the designation "Standard".

Now enter the values as shown in the picture:

**T** Project Item

Plugin: Standard

Input file: C:\utils\as8\turgen\examples\test16k.xex Browse...

Output file: Browse...

Conversion type: Binary file Check loader

Binary file options

Binary loader: TSCBL

Title: TEST16K.XEX Auto set

Look and Feel: Background ... PAL NTSC Luminance: 10

☐ Hide cursor ☐ Silent I/O ☐ Suppress ATRACT

Silence list: S P 1.0

Transfer speed: 600

OK Cancel Clear

Click the *Browse* button in the input file box to select the test16k.xex file you downloaded before.

Select the *Binary file* in the Input file type box. This is usually already selected.

Click the *Auto set* button to automatically set the program title. You can set a different title if you want.

Your project item is ready. The result will be a short tape boot file (that you will load with START+OPTION) followed by the test16k.xex binary load file. The boot file will display the title and then load and run the test16k.xex file.

Click the *OK* button. The project item will appear in the project.

**T** File Edit View Generator Tools Help TURGE... - □ X

Add Edit Remove All Wizard Up Down WAVE Audio Tape image

Input file	Output file	Plugin	Speed	Details
test16k.xex		Standard	600	[B] TEST16K.XEX

...

Input file	Output file	Status	Duration
------------	-------------	--------	----------

# Setting-up output

Before you generate the signal and record it on the tape, you have to take a few setup steps.

## Setting Windows Sounds

Firstly, you have to disable the system sounds in Windows. You wouldn't like your recording to be spoiled by a notification bubble. Go to the control panel and disable the system sounds.

Secondly, you need to disable any special effects produced by your sound card. Go to the control panel of your sound card and disable any 3D effects, bass boost, etc.

## Mono or Stereo

You have to set up TURGEN to generate a mono or stereo signal.

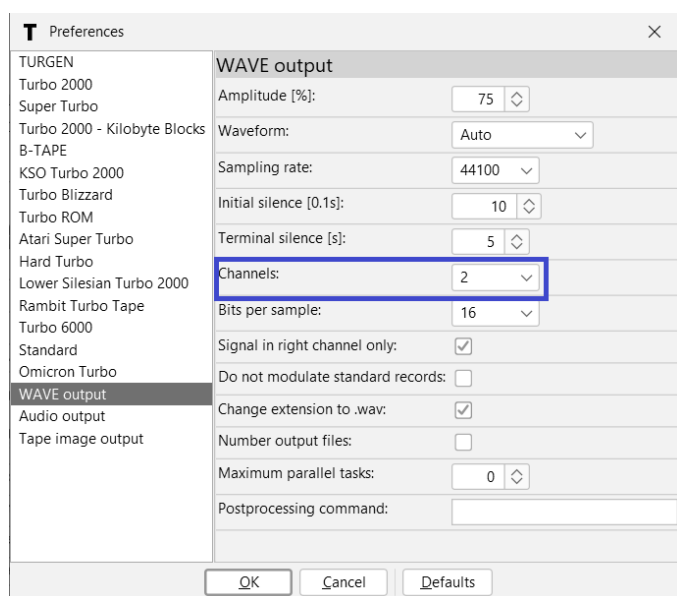
You might already know that all Atari data recorders (410, 1010, XC11, XC12, XL12) use stereo tape heads.

- The left channel is dedicated to voice-overs or music. The signal from the left channel is routed to the AUDIO IN pin of the SIO port and then to the monitor or TV.
- The right channel is dedicated to the data (the signal representing data that sounds like screeching). The signal from the right channel is routed to the FSK demodulation circuit, then to the DATA IN pin, and then to the POKEY chip.

Therefore

- If your ordinary cassette recorder is mono, set TURGEN to generate a mono signal. The Atari data recorders accept the signal recorded with a mono ordinary recorder.
- If your ordinary cassette recorder is stereo, set TURGEN to generate a stereo signal. The left channel will have no signal, the right channel will have the data. This is the default setting.

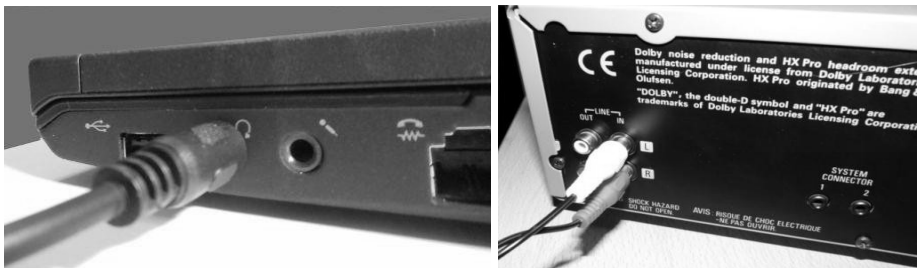
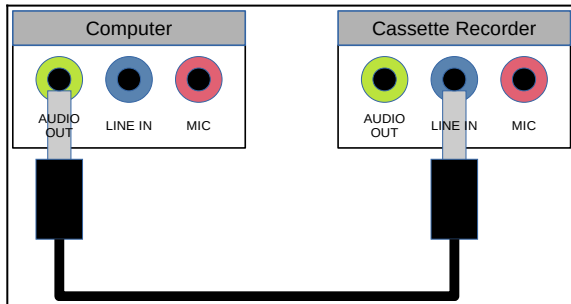
To set up mono or stereo output, go to the *Tools* menu and select *Preferences*. Then navigate to the *Audio Output* section and select the number of channels (either two for stereo or one for mono).





## Wiring up

Now you need to connect your tape recorder to your computer. Connect the output of the sound card to the input of the tape recorder using a cable.



Which input to use on the tape recorder? The best option is to use LINE IN or AUX IN.

The microphone input is typically not the best option (as its circuitry is optimized to capture voice and provides high amplification). Use only if you have no other choice.

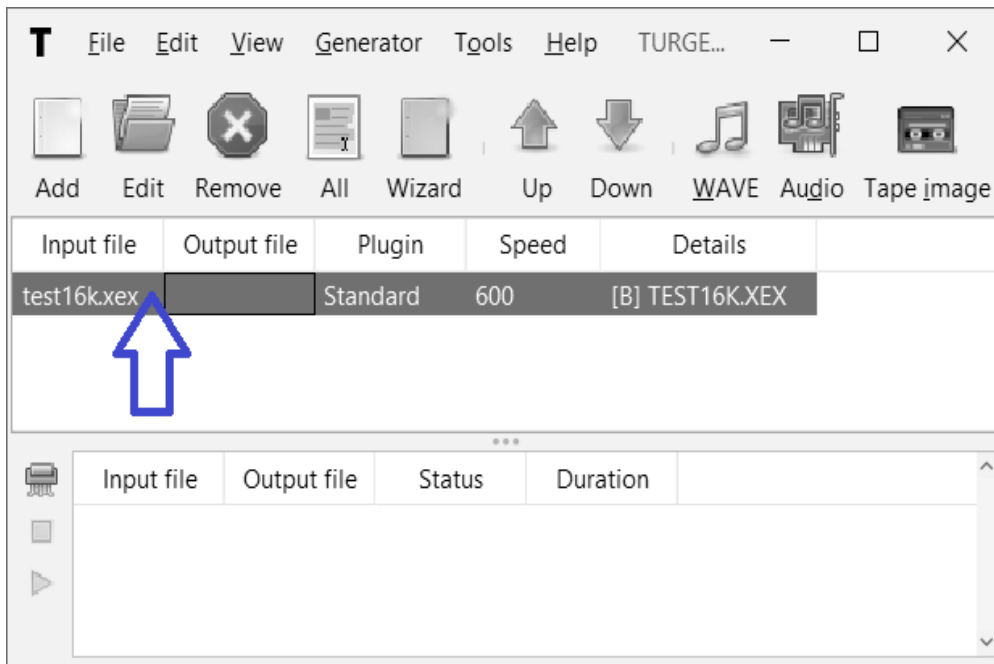
## Recording

Prepare the tape recorder. Insert the tape. If you are recording to the very beginning of a side, ensure you begin the recording on the tape, not on the protective tape leader. Ensure the desired input source is selected on the tape recorder. Also, ensure that any noise reduction features (e. g. DOLBY) are switched off.



Set up the recording level on the tape recorder (if the recorder has such control). Set up the volume of the sound card. It shouldn't be more than 75 % of the maximum.

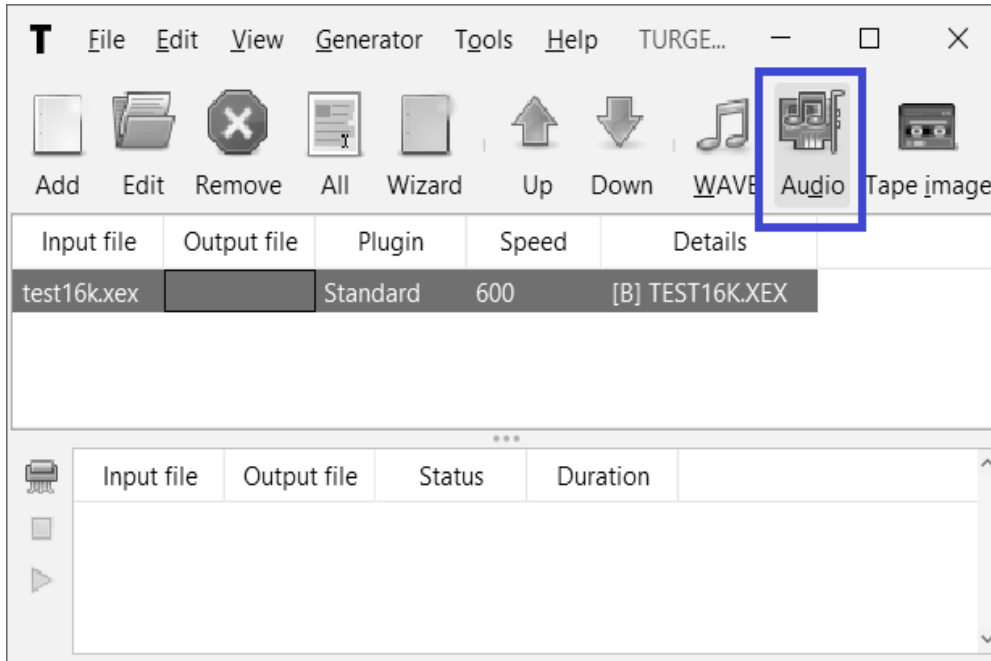
In TURGEN, select the project item you just created by clicking it.



On the tape recorder, start recording by using its controls (typically the PLAY+RECORD keys).



In TURGEN, click the *Audio* button. This will generate the signal and send it to the sound card.



And then wait until the signal is recorded on the tape. Observe the progress bar in the bottom part of the main program window. When done, stop the tape recorder. And that's it.

## Testing the tape

You can test the recorded tape as follows:

1. Rewind the tape while still in the ordinary tape recorder.
2. Switch off the Atari computer, remove all cartridges, connect the data recorder, and insert the rewind tape in the data recorder.
3. Press START and OPTION, and keep holding them.
4. Switch the Atari computer on. Wait for a beep, then release START and OPTION.
5. Press PLAY on the data recorder. Press SPACE BAR.
6. The Atari computer will boot from tape.



## Recording level

If your tape recorder has an indicator of the recording level (either an analog indicator or a digital bar graph), ensure the signal is strong enough (close to the red area), but not too strong (in the red area). If your data recorder doesn't have a such indicator or uses an automatic recording level setting, then your only control is the output volume of the sound card. In this case, find the proper volume by experimentation.



## Creating WAVE file

If you cannot or don't want to record directly to the tape recorder, you can generate a WAVE file. Then you can process it as you see fit (convert it to MP3 or to a CD track, play it back from a portable MP3 player or phone).

When creating a project item, identify also the output file. Then, instead of clicking the *Audio* button, click the *WAVE* button. A WAVE file will be generated.

---