

TURGEN
Create Atari Tapes
version 9.2.0

Michael Kalouš of the BAKTRA Software

Part I

TURGEN

1 Introduction

1.1 Mission and Features

TURGEN is a versatile utility that allows you to:

- Create your own tapes with software for Atari 8-bit computers.
- Transfer data from your PC or Mac to your Atari 8-bit computer using just a data recorder and a cassette adapter.

1.1.1 Primary Features

- Conversion of digital images of Atari software either to the original standard Atari tape records or to the most popular “turbo” systems.
- Output to WAVE files, tape images, or to the sound card.
- Comprehensive support for conversion of segmented binary files.

1.1.2 Auxiliary Features

- Processing of tape images
- Facilities for automated production of multiple cassettes – batch processing, automatic file name creation, and creation of tape sides
- Tape image extractor tool that extracts data from tape images
- Tool for embedding tokenized BASIC files to binary load files
- Tool for creating monolithic binary load files
- Test pilot tone generator (allows to find optimal volume for a cassette adapter or calibrate tape heads)
- GENCAS command line interface
- Creation of dual track tapes

1.1.3 Support for Standard Tape Records

Standard tape records and unmodified data recorders are fully supported. Tape images with baud, data, and fsk chunks can be read by the *Tape image* plugin. Monolithic binary files, segmented binary files, BASIC files, and plain data files can be converted to standard tape records by the *Standard* plugin.

1.2 Hardware that Can Be Used

- Data recorders with or without turbo upgrades, compact cassettes
- Devices that replace compact cassettes and data recorders, for example MEGA-CD interface (CD-LINK), SIO2PC, and ATART.

Important information. If you are using devices other than data recorders and compact cassettes, please read section 8.5 that contains vital information about how to use such devices. Then read section 4.8.

1.3 Copyright

TURGEN is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

1.4 Support

For technical support, refer to <https://turgen.sourceforge.io/support.html>. This project is not backed by any company or organization, support is provided on a voluntary basis.

1.5 Glossary

Data recorder. A device designed to read or write signal to or from compact cassettes, specially designed to be connected to computers, for example, Atari XC12 or Atari 1010.

Tape recorder. Consumer electronics device designed to read or write electric signals to or from compact cassettes.

Standard Tape Records. Records that the Atari 8-bit computer can read and write using a data recorder that hasn't been modified.

Turbo system. A system designed to speed up data transfer of the original Atari data recorder. A typical turbo system consists of three components:

1. Hardware modification of the data recorder (typically a small circuit bypassing the FSK demodulator)
2. Software that allows for usage of the hardware modification (loaders, tape operating systems)
3. One or more file formats

Program directory. Directory or folder where TURGEN is installed to. The special symbol <TSDIR> is also used to reference this directory.

Configuration directory. Directory or folder where TURGEN stores user configuration files. The special symbol <CFGDIR> is also used to reference this directory.

JRE. Java SE Runtime Environment.

JDK. Java SE Development Kit.

Java home directory. Directory or folder where JRE (or JDK) resides. A special symbol <JAVAHOME> is also used to reference this directory.

1.6 Similar Utilities

TURGEN provides similar functionality as the following utilities: A8CAS, a8cas-util, CAS2WAV, XEX2CAS 2.4, and ATART.

2 Files Processed or Generated by TURGEN

The most important types of files processed and generated by TURGEN are *binary files*, *tape images*, and *WAVE files*. Note that TURGEN does not support disk images or cartridge images.

2.1 Binary Files

TURGEN provides extensive support for processing of *Atari DOS 2 Binary Files*. These files will be referenced as *binary files*. A binary file is a file with precisely defined internal structure that is mainly used to store *programs and data* for the Atari 8-bit computers. Note that the knowledge of the binary files is critical for successful usage of TURGEN. Typical extensions for binary files are the following: COM, XEX, OBJ.

2.1.1 Internal Structure.

A binary file always starts with a two-byte **header**. Both bytes have a value of 255 (0xFF). A binary file contains **segments**. A segment is a block of data prefixed with a **segment header**. The segment header holds two 16-bit addresses (segment start address and segment end address). These two addresses determine the addresses the segment data will be loaded to. The segment header can be optionally prefixed with two bytes of value 255 (0xFF).

2.1.2 Special Vectors.

When a binary file is being loaded, certain addresses have a special meaning. These addresses (special vectors) are described in table 1.

Addresses	Special Vector
736 - 737	RUN vector. When all segments of a binary file are loaded, the binary loader will perform an indirect jump to subroutine (JSR) or jump (JMP) using this vector. This vector defines run address of the binary file.
738 - 739	INIT vector. Whenever a segment that changes this vector is loaded, the binary loader will perform a jump to subroutine (JSR) using this vector. INIT vectors allow to execute code while the binary file is being loaded.

Table 1: Vectors

2.1.3 Segment Types

- DATA Segments. Segments that do not change any of the special vectors.
- INIT segments. Segments only change both bytes of the INIT vector.
- RUN segments. Segments only change both bytes of the RUN vector.

2.1.4 Binary File Types

Segmented Binary File or just **Binary File** is a binary file without any special limitations or restrictions.

Monolithic Binary File is a special case of a binary file. Monolithic binary file that consists of exactly one DATA segment and at most one RUN segment.

2.1.5 Loading Binary Files

Binary Loader. A program, or routine (usually a component of an operating system) that loads and executes binary files.

Binary Load. A name for the process of loading and running of a binary file. This process is performed by the binary loader.

2.2 Tape Images

A tape image is a computer file containing the contents and structure of data stored on a compact cassette. A tape image is usually perfectly replicating the structure and contents of data stored on compact cassette regardless the file format and operating system used to store the data. TURGEN supports CAS, a binary file format originally introduced by Ernest R. Schreurs as the output format of his WAV2CAS utility and later extended by the A8CAS project. The CAS format stores Atari cassettes efficiently. For more information, refer to <http://a8cas.sourceforge.net/format-cas.html>.

2.3 WAVE Files

Waveform Audio File Format (WAVE, or commonly known as WAV due to its filename extension) is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs. The WAV file is an instance of a Resource Interchange File Format (RIFF) defined by IBM and Microsoft. For more information, refer to <https://en.wikipedia.org/wiki/WAV>.

3 Installation

3.1 System Requirements

Operating System. TURGEN requires an operating system supported by Java SE Runtime Environment (JRE) or Java SE Development Kit (JDK).

This includes, but might not be limited to: Microsoft Windows, macOS, GNU/Linux distributions, Solaris, OpenIndiana, FreeBSD.

Java. TURGEN requires Java SE Runtime Environment (JRE) or Java SE Development Kit (JDK), version 9 or newer.

3.2 Overview

TURGEN is distributed in the following packages

- Installer for Microsoft Windows (turgen-n.n.n-installer.exe)
- Binary package (turgen-n.n.n.tar.bz2)

3.3 The Process of Installation

Installation of TURGEN is a two-step process:

1. Installation of Java Runtime Environment (JRE) or Java Development Kit (JDK), if not installed already
2. Installation of TURGEN

3.3.1 Installation of JRE or JDK

Before installing TURGEN, you must install the JRE or JDK.

I recommend downloading from the Eclipse Adoptium project.

For convenient download links, visit the Downloads page of the TURGEN project web.

3.3.2 Installation of TURGEN

To install TURGEN, download and launch the installer for Microsoft Windows or download and extract the binary package.

3.4 Experimental packages

Two experimental installation packages are available for selected versions. These packages are self-contained and include both TURGEN and JRE.

- Full pack installer for Microsoft Windows (turgen-n.n.n-fullpack.msi)
- 64-bit Debian package for GNU/Linux (turgen_n.n.n-1_amd64.deb)

4 Operations Guide

4.1 Starting TURGEN

Starting from Command Line

The program code of the TURGEN is located in the <TSDIR>/dist/turgen.jar file. To run TURGEN, run the Java launcher and specify that the program code is in the turgen.jar file, using the following commands:

<JAVAHOME>\bin\javaw.exe -jar <TSDIR>\dist\turgen.jar under Microsoft Windows or
<JAVAHOME>/bin/java -jar <TSDIR>/dist/turgen.jar on Unix-like systems.

Microsoft Windows

Use the desktop or start menu shortcuts created by the installer or launch the turgen.exe launcher in the program directory.

Other Operating Systems

Start TURGEN from the command line or create launchers or shortcuts using your desktop environment. Alternatively, you can use the turgen.sh BASH script.

Command Line Parameters

One command line parameter is accepted - a file that stores a project.

Log File

Exceptions, traces, or other important diagnostic information are stored in the turgen.log file located in the configuration directory. This file is recycled when it exceeds the size of 4 MB.

4.2 Program Controls

Main Menu and Project

The main menu is located at the top of the program window. Almost all functions are accessible from the main menu.

In the middle of the program window, there is the *project* that stores information on what files are to be converted and how they will be converted.

Main Control Buttons

Under the main menu, there is a toolbar, where the *main control buttons* are located. The most used functions can be invoked using these buttons. Every button has its icon and name. The buttons are depicted in table 2.











Icon	Name	Icon	Name
	Add item to project		Move project items up
	Edit selected project item		Move project items down
	Remove selected project items		Generate WAVE file
	Select all project items		Generate AUDIO directly
	Generate tape image		Wizard for files

Table 2: Main Control Buttons

Worklist

The worklist is located in the bottom part of the program window. It displays work items. Each work item represents work in progress or work that has been completed. Double-click a work item to open the output file associated with the work item. Right-click a work item to display a popup menu with options.

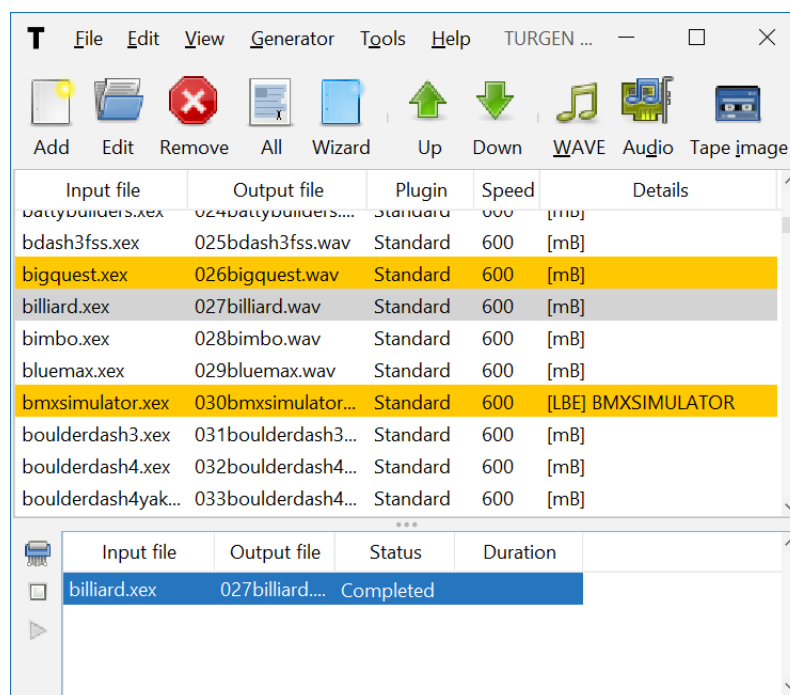


Figure 1: Program Window

4.3 Conversion of Files

4.3.1 Project Items

To convert a file, TURGEN requires information about the conversion that includes the input file name, input and output format, output file name, what loader to prepend, and other parameters. This information is stored in the *project items*. The Project items are elements of the *project*.

4.3.2 Working with Project

To work with the project, use the main control buttons or menu items from the *File* menu.

To add a project item to the project, click the *Add item to project* main control button. To edit project items, click the *Edit selected project item* main control button. To remove selected project items, click the *Remove selected project items* main control button.

To select project items, use your keyboard or your mouse. To select all project items, click the *Select all project items* main control button.

To move project items use the *Move project items up* and *Move project items down* buttons.

To load or save the project, use the *Load project* and *Save project* items from the *File* menu.

4.3.3 Editing and Creating Project Items

TURGEN provides a dialog for creating and editing project items. The dialog is depicted in figure 2.

T Project Item [X]

Plugin: Standard [US Flag] [v]

Input file: C:\utils\8\stuff\Train 1.0 (v3).xex [Browse...]

Output file: C:\utils\8\stuff\Train 1.0 (v3).wav [Browse...]

Conversion type: Binary file to LBE [v] [Check loader]

Binary file options:

Binary loader: LBE Plain [v]

Title: TRAIN 1.0 (V3) [Auto set]

Look and Feel: Background ... [PAL] [NTSC] Luminance: 10 [v]

☐ Hide cursor ☒ Silent I/O ☐ Suppress ATRACT

Silence list: 1,1,1,1,1 [S] [P] [1.0] [v]

Transfer speed: 600 [v]

[OK] [Cancel] [Clear]

Figure 2: Manipulating Project Item

Use the *Plugin* box to select a plugin that will be used for conversion. The national flags help you identify the country of origin of the loading system the plugin represents.

Use the common panel at the top of the dialog to choose the input and the output file. To display a dialog with information about the input file, click the *Input file* label. To populate the output file text field automatically, click the *Output file* label. To display a file chooser dialog, click the *Browse* buttons.

In the central part of the dialog, there are controls specific to the selected plugin. These controls are described in part II.

Use the buttons at the bottom to commit or cancel the changes you have made to the project item. Click the *Clear* button to reset the controls of the dialog.

4.3.4 Special Functions

To copy the project to the clipboard in the “comma separated values” file format, select the *Copy project to the clipboard* menu item from the *Edit* menu.

To populate the project with items by processing a batch file, select the *Process batch file* menu item from the *File* menu. For more information about batch processing, refer to section 7.7.

4.4 Wizard for Files

The Wizard for files provides convenience when converting one or more binary files or tape images. You have two options to start the Wizard:

1. Click the *Wizard for files* main control button
2. Drag file icons from a file manager and drop them in the project area of the main window

4.4.1 Converting Single File

In the first step, specify the following:

- Binary file or tape image you want to convert
- Tape loading system
- If you want to create a silence list (refer to section 4.8 for more information).

Then click the *Next* button to continue. The wizard will process the input file and identify the most appropriate methods of conversion.

T Wizard for files

Input file(s)

File(s): C:\utils\8stuff\Train 1.0 (v3).xex Browse...

Tape loading system

☒ Standard

☐ Turbo 2000 ☐ Super turbo

☐ KSO Turbo 2000 ☐ Turbo 2000F

☐ Atari Super Turbo ☐ Unerring Master ☐ Atari Turbo Tape (ATT)

☐ Lower Silesian Turbo 2000

☐ Turbo Blizzard ☐ Turbo ROM ☐ Hard Turbo

☐ Rambit Turbo Tape

☐ Turbo 6000

Silence list

Generate silence list: None Silence duration: 1.0

Advanced settings

☒ Restrict conversion to methods that do not require special external loaders

<< Back Next >> Finish Cancel

Figure 3: Wizard for Files - Step 1

In the second step, the Wizard displays all possible conversion methods and orders them by their ranking. For each conversion method, the Wizard displays ranking, method name, and the number of pilot tones (binary files only).

Enter the following information:

- Select one of the conversion methods
- Select a transfer speed
- Specify output file (optional)

Then click the Finish button. A project item will be created.

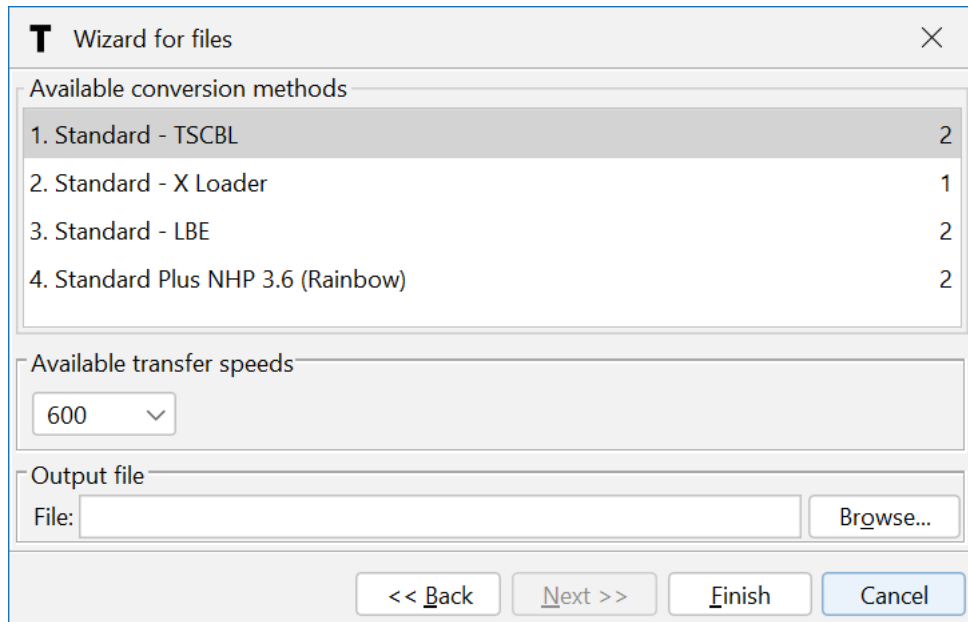


Figure 4: Wizard for Files - Step 2

4.4.2 Converting Multiple Files

In the first step, enter the following:

- Click the *Browse* button. A file chooser appears. Select multiple files as needed. The list of files will appear in the *File(s)* box prefixed with the \$LIST\$ special keyword. The files are separated by semicolons.
- If you need to add more files, hold SHIFT while clicking the Browse button. Files selected with the file chooser will be added to the list.
- Specify the tape loading system and silence list.

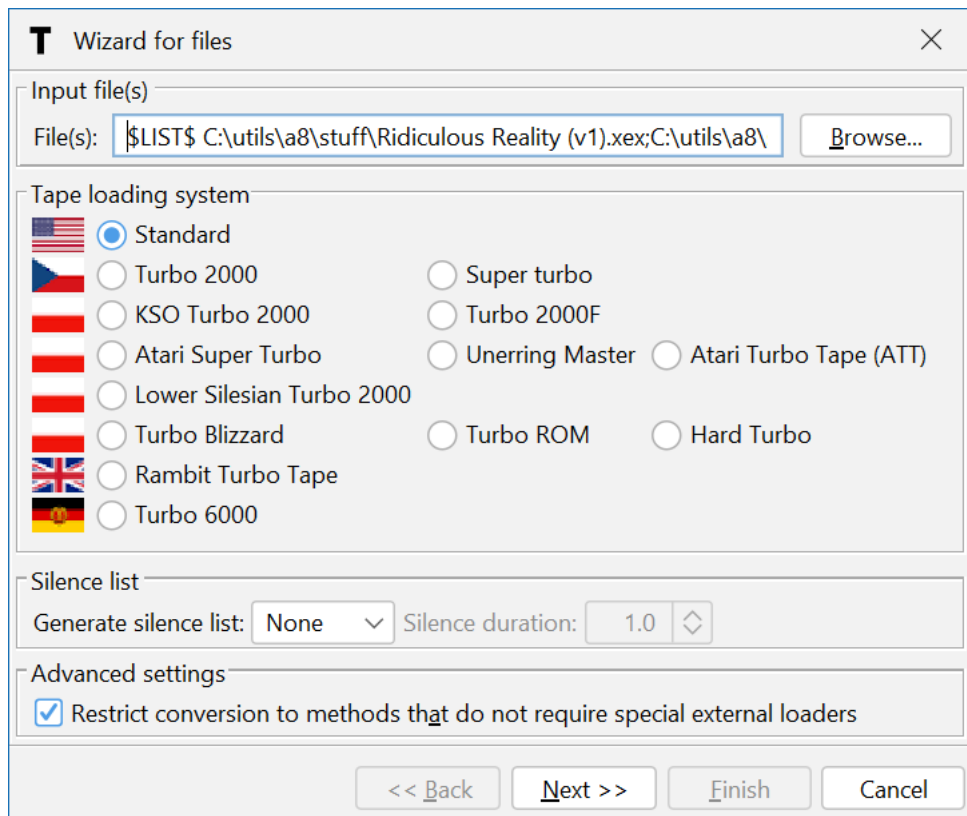


Figure 5: Wizard for Files - Multiple files - Step 1

Then click the Next button to specify additional options.

In the second step, enter the following:

- Output directory. If you do not specify an output directory, all output file names will be empty
- Specify if output file names are prefixed with numbers (Number output files box)
- Specify what is the preferred transfer speed. If the preferred transfer speed cannot be used, then the closest possible transfer speed will be used.
- Select allowed conversion methods. All methods are selected by default.

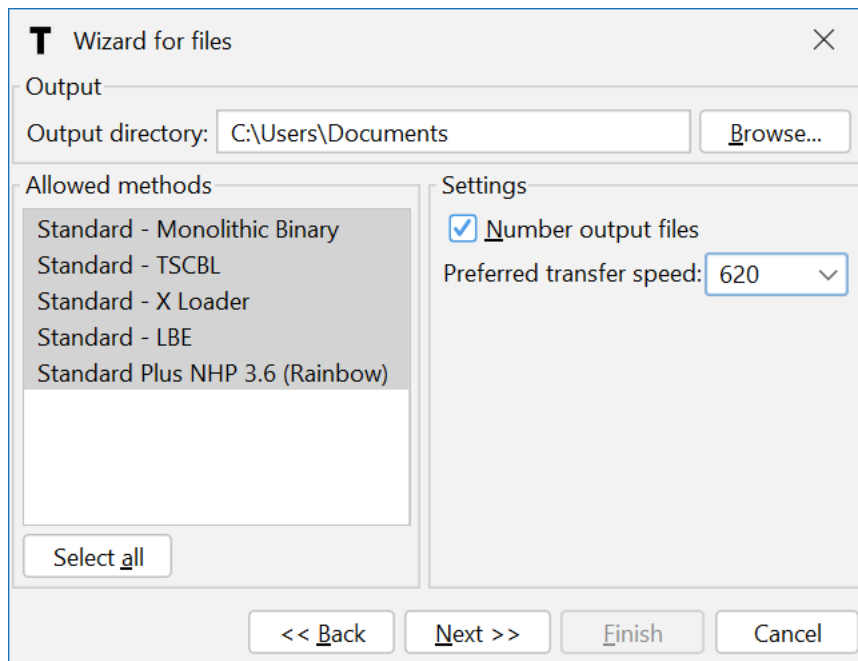


Figure 6: Wizard for Files - Multiple files - Step 2

Then click the Next button to review the conversion results. The conversion method with the best ranking is automatically selected for each input file.

Click the Finish button to add items to the project.

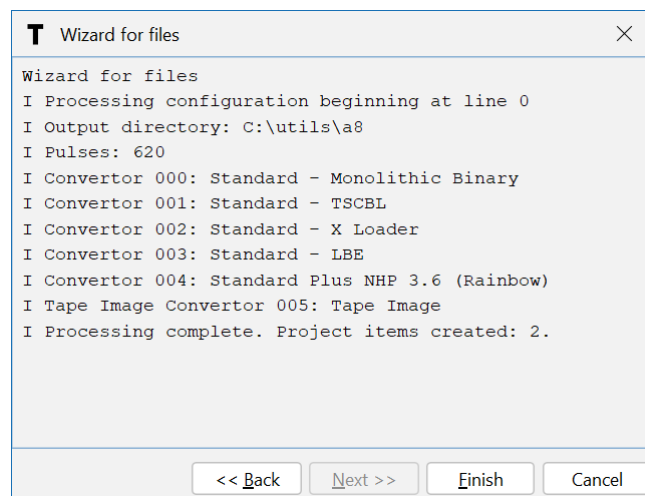


Figure 7: Wizard for Files - Multiple files - Step 3

4.4.3 How Ranking Works

When the Wizard for files calculates ranking to compare the fitness of the conversion methods, it considers the following:

1. Compatibility of the method with the binary load file or tape image (conversion possible or not)
2. Loader availability. Absence of generally available loader results in worse ranking.
3. Nature of the loader. Loaders using non-standard practices (e.g. storing data in RAM under ROM) are given a worse ranking.
4. Time wasted by pilot tones. Fewer pilot tones result in better ranking.

5. Transfer speeds. Ability to use the maximum speed provided by the hardware modification results in better ranking.
6. Auxiliary ranking to resolve ranking ties.

Note: Points 2 to 5 do not apply to tape images.

4.5 Output

4.5.1 Output of Electric Signal into WAVE File

To output electric signals to a WAVE file, select at least one project item and click the *Generate WAVE file* main control button. The WAVE files will be generated in one or more parallel tasks. To stop the generation, click the *Stop* button in the Worklist.

4.5.2 Output of Electric Signal to Sound Card.

To output electric signals to the sound card, select at least one project item and click the *Generate AUDIO directly* main control button. Project items are processed sequentially. To stop the output, click the *Stop* button in the Worklist. To resume an output that has been paused, click the *Resume* button.

4.5.3 Output of Tape Image

To output a tape image, select at least one project item and click the *Generate tape image* main control button. Project items are processed sequentially. The output files are overwritten without a warning.

4.5.4 Waveforms

TURGEN allows you to choose the waveform of the electric signals generated for turbo records. You can select one of the waveforms described in the table below.

Waveform	Description
Auto	Automatic selection. This is the default. Pure Sine Wave for transfer speeds below 4000 bps. Square for transfer speeds above 4000 bps.
Square	Ideal rectangular pulses
7th Harmonic	High precision approximation of rectangular pulses (Fourier series expansion, 7th harmonic)
5th Harmonic	Medium precision approximation of rectangular pulses (Fourier series expansion, 5th harmonic)
3rd Harmonic	Low precision approximation of rectangular pulses (Fourier series expansion, 3rd harmonic)
Pure Sine Wave	Harmonic pulses

4.5.5 Output in Preview Mode

Output in the preview mode allows you to quickly check if your files can be converted and determine the total duration of the generated signal. This is useful when you want to know if the converted files will fit a tape side or estimate how many tape sides will be needed.

To enable the Preview mode, select the *Preview mode* item from the *Generator menu*.

In the preview mode

- Project items are processed
- No output files are created
- No postprocessing is performed
- No electric signals are sent to the sound card
- Durations of the outputs are displayed in the worklist

4.5.6 Sampling Rate

For each output, you can select a sampling rate using the Program configuration facility. For output to the sound card and the wave file, the sampling rate configuration entry determines the actual sampling rate. For tape image output, the sampling rate determines the smallest time unit used for pwm chunks in the tape image.

The default sampling rate is 44100 Hz. It is sufficient for most use cases.

Higher sampling rates (48000 Hz and 96000 Hz) are most useful for turbo systems using transfer speeds above 4000 bd, where more precision is required. Only some plugins take advantage of higher sampling rates, this is noted in the documentation for the plugins.

A lower sampling rate (22050 Hz) is available for legacy hardware and software. In general, usage of this sampling rate is not recommended.

4.6 Input/Output matrix

The conversion capabilities of TURGEN are depicted in the following table.

Input	Wave file	Sound card	Tape image with pwmX chunks
Regular file	YES	YES	YES
Binary file	YES	YES	YES
Tape image (pwmX chunks)	YES	YES	Passed through
Tape image (baud, fsk chunks)	YES	YES	Passed through

4.7 Pulse Corrections

Pulse corrections allow you to permanently adjust the duration of the pulses.

4.7.1 Permanent Pulse Corrections

To make permanent corrections of pulses, use the Program configuration facility. The plugins that support permanent pulse corrections are KSO Turbo 2000, Atari Super Turbo, Turbo ROM, and Turbo Blizzard.

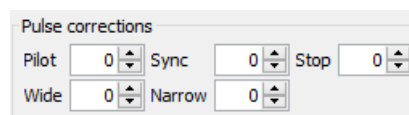


Figure 8: Pulse Corrections

4.8 Silence Lists

4.8.1 Overview

The silence lists allow comfortable usage of devices other than data recorders and compact cassettes (refer to section 8.5).

The silence lists allow the following:

- Insert silence (or a neutral signal) of a defined duration after a block that contains one or more INIT segments
- Pause output of the electric signals to the sound card after a block that contains one or more INIT segments

The silence lists are part of the project items (refer to section 4.3.1).

4.8.2 Silence List Syntax

A silence list consists of silence specifications (SPECs) separated by commas:

`SILENCE_LIST:=<SPEC>,<SPEC>,...,<SPEC>`

A silence specification is a number followed by an optional P suffix or just a P suffix which is an equivalent of 1P syntax:

`SPEC:=n[P]|P`

where n is the generated silence duration in seconds. The P suffix indicates that output of the electric signals to the sound card will be paused. Tenths of seconds can be specified too.

4.8.3 Examples

Silence List	Explanation
1,3,0,4.5	1 second of silence after 1st INIT segment 3 seconds of silence after 2nd INIT segment No silence after 3rd INIT segment 4.5 seconds of silence after 4th INIT segment
3.1,2P,P	3.1 seconds of silence after 1st INIT segment 2 seconds of silence after 2nd INIT segment, output of electric signal to sound card will be paused 1 second of silence after 3rd INIT segment, output of electric signal to sound card will be paused

4.8.4 User Interface

To create a silence list, you can use the Silence list panels. Enter the silence specifications in the text field, or click the S or P buttons to populate the list automatically (the input binary file will be analyzed and a silence specification will be created for each INIT segment found). Use the spinner to set the duration of the automatically generated silence specifications.



Figure 9: Silence List Panel

4.8.5 Cumulation of Silence

Some plugins allow cumulation of silence.

The cumulation can occur when multiple INIT vectors are stored in one data block.

- If the cumulation is enabled, the duration of the resulting silence is calculated as an addition of all silence specifications matching the INIT vectors stored in the affected data block. This is recommended for devices that ignore the MOTOR CTRL signal.
- If the cumulation is disabled, the duration of the resulting silence is the longest of all silence specifications matching the INIT vectors stored in the affected data block. This is recommended for compact cassettes.

4.9 Automatic File Name Creation

Various turbo systems have different rules and limitations for file names. TURGEN provides a configurable automatic file name creation facility that is used by selected plugins.

Each plugin that supports automatic file name creation provides a configuration entry that allows configuring the file name creation.

- Select the *Create extension* check box to always generate file names with extension

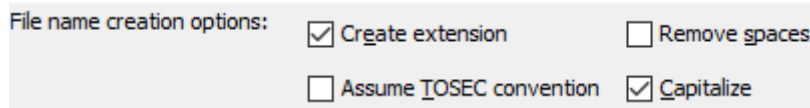


Figure 10: Automatic File Name Creation

- Select the *Remove spaces* check box to generate file names without spaces
- Select the *Capitalize* check box to generate capitalized file names
- Select the *Assume the TOSEC naming convention* check box to create file names from only the first part of the TOSEC file name (up to the first left parenthesis)

5 Program Configuration Facility

The Program configuration facility provides various program modules with configuration capabilities. Program configuration consists of *configuration entries* that are grouped into *configuration classes (or sections)*.

5.1 Changing Configuration

To change the program configuration, select the *Preferences* item from the *Tools* menu. A dialog that allows changing the preferences will appear.

Click the *OK* button to save the changes. Click the *Cancel* button to cancel the changes. Click the *Defaults* button to reset the configuration entries to their default values for the currently selected configuration class.

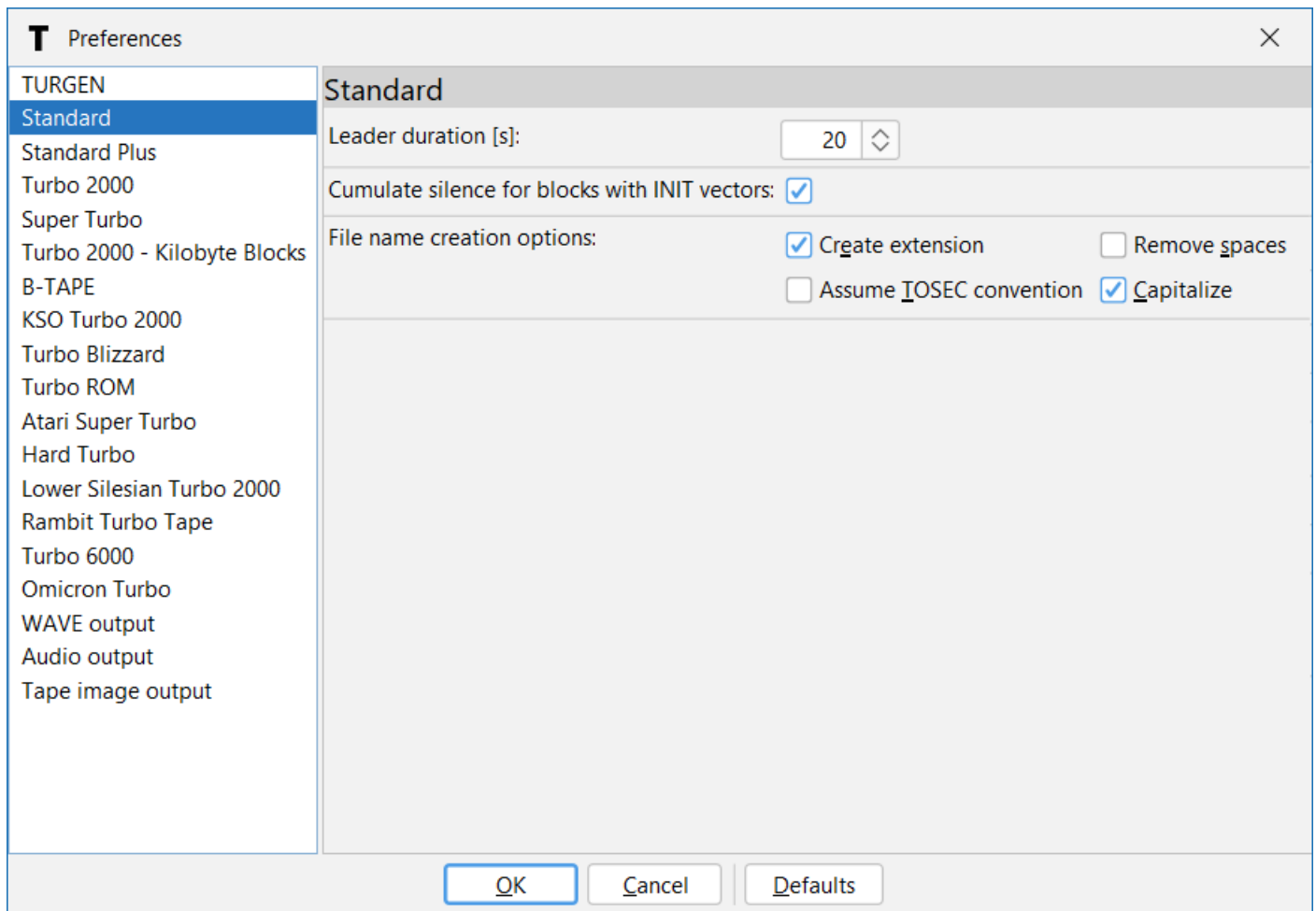


Figure 11: Preferences

5.2 General Configuration Entries

TURGEN/GUI Look and feel

Look and feel of the user interface.

TURGEN/Favorite output directory

Favorite directory for output files (WAVE files and tape images).

5.3 Output of Electric Signals to Sound Card

Audio output/Amplitude

Amplitude of the signal. 0-100%.

Audio output/Waveform

This allows you to select a waveform. For more information, refer to section 4.5.4.

Audio output/Sampling rate

This allows you to select a sampling rate. Allowed values are 44100, 48000, and 96000 Hz. Note that only some plugins can take advantage of sampling rates higher than 44100 Hz.

Audio output/Initial silence

Duration of silence generated at the beginning of the output, one unit is 0.1 second.

Audio output/Terminal silence

Duration of silence generated at the end of the output. This allows to create gaps between programs.

Audio output/Channels

Number of channels.

Audio output/Bits per sample

Number of bits per sample.

Audio output/Signed samples

Indicates whether to use signed samples.

Audio output/Signal in right channel only

If the number of channel is set to two, signal will be generated to the right channel only.

5.4 Output of Electric Signals to WAVE File

WAVE output/Amplitude

Amplitude of the signal. 0-100%.

WAVE output/Waveform

This allows you to select a waveform. For more information, refer to section 4.5.4.

WAVE output/Sampling rate

This allows you to select a sampling rate. Allowed values are 44100, 48000, and 96000 Hz. Note that only some plugins can take advantage of sampling rates higher than 44100 Hz.

WAVE output/Initial silence

Duration of silence generated at the beginning of the WAVE file, one unit is 0.1 second.

WAVE output/Terminal silence

Duration of silence generated at the end of the WAVE file. This allows you to create gaps between programs.

WAVE output/Channels

Number of channels.

WAVE output/Bits per sample

Number of bits per sample.

WAVE output/Signal in right channel only

If the number of channels is set to two, the signals will be generated to the right channel only.

WAVE output/Change extension to .wav

When selected, the extensions of output files will be changed to ".wav" automatically.

WAVE output/Number output files

When selected, every output file name will be prefixed with a sequential number.

WAVE output/Maximum parallel tasks

Maximum number of parallel tasks that will generate wave files. Set to 0 for automatic determination (default, number of processors - 1), or specify 1-8 to set the number of tasks manually.

WAVE output/Postprocessing command

Command line for postprocessing. For more information refer to section 9.1.

5.5 Output of Tape Images

Tape image output/Sampling rate

This allows you to select a sampling rate. Allowed values are 44100, 48000, and 96000 Hz. The sampling rate is used for the pwm chunks to set the smallest time unit.

Tape image output/Change extension to .cas

When selected, the extensions of output files will be changed to ".cas" automatically

Tape image output/Auto create temporary files

If the output file is not specified and this entry is set to true, a temporary file is created automatically.

Tape image output/Postprocessing command

Command line for postprocessing. For more information refer to section 9.2.

6 Advanced Settings

6.1 Manage Plugins

To manage the plugins, select the *Manage Plugins* item from the *Tools* menu. A dialog will appear. In this dialog, you can change order of the plugins and enable or disable the plugins.

Note that all changes will take effect only *after you restart TURGEN*.

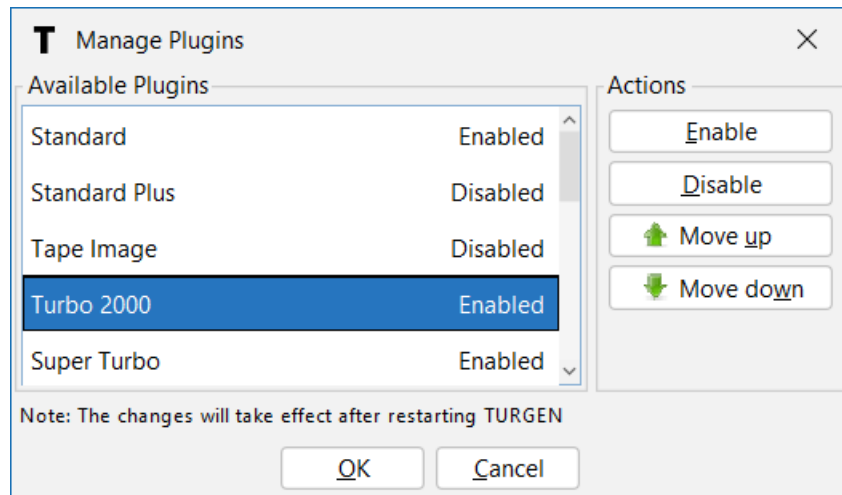


Figure 12: Manage Plugins

6.2 Repository of PWM Pulses

To change the repository of pulses, first, extract the `turgen/etc/pulses.list` file from the `turgen.jar` archive and place it in the program directory. Then you can edit the file. The repository can be modified, directions are present in the file as comments. The changes take effect after TURGEN is restarted. It is not recommended to modify the repository without serious reasons. Using Pulse corrections is usually a better option. For more information, refer to section 4.7.

Note: TURGEN will use `pulses.list` file present in the program directory to read the repository. If the file is not present in the program directory, `pulses.list` in the `turgen.jar` archive will be used instead.

7 Tools

7.1 Tape Image Extractor

A tape image extractor is a tool designed to extract data from tape images. The tape image extractor allows you to do the following:

1. Display basic information about tape image chunks
2. Select tape image chunks for data extraction
3. Determine which portions of tape image chunks will be extracted
4. Extract data from tape image chunks to plain data files
5. Extract boot files to binary files
6. Extract data from tape image chunks to binary files
7. Save selected tape image chunks to a new tape image file

7.1.1 Operations

To open the Tape Image Extractor window, select the *Tape image extractor* item from the *Tools* menu.

To open a tape image, select the *Open tape image* item from the *File* menu.

To extract data from tape image chunks, select one or more tape image chunks. Then select the *Extract data* item from the *Chunks* menu. Select the output file using the displayed file chooser. A report is displayed after the extraction.

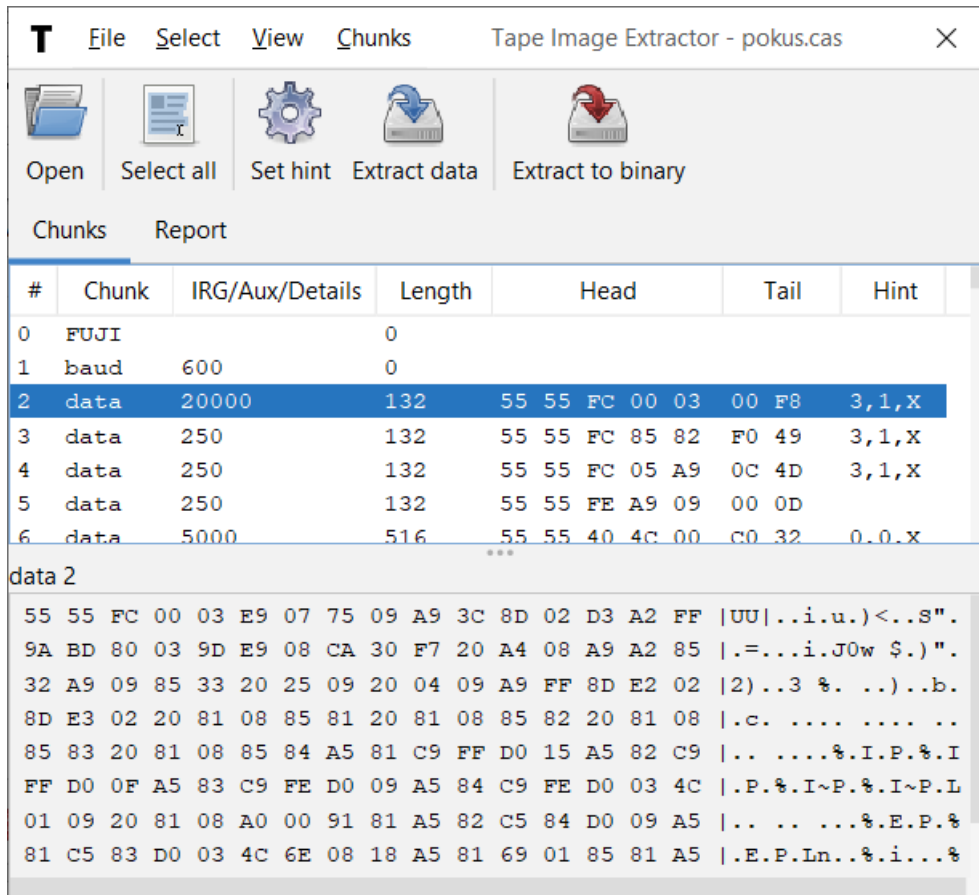


Figure 13: Tape Image Extractor

To extract data from tape image chunks to a binary file, select one or more tape image chunks. Then select the *Extract data to binary file* item from the *Chunks* menu. A dialog that will allow you to select the output file and parameters of the extraction will be displayed. Enter values and click the *Extract* button to perform the extraction.

To determine which portion of a tape image chunk or chunks will be extracted, select one or more tape image chunks. Then select the *Set data extraction hint* from the *Chunk* menu. Enter values and click the *OK* button.

To save selected tape image chunks to a new tape image, select the *Save selected chunks*, or *Save all chunks* item from the *File* menu. Select the output file using the displayed file chooser. Note that a FUJI chunk is inserted into the beginning of the tape image, if needed.

7.2 Creating Tape Sides

This tool allows you to create tape sides from the project items. Enter properties of your tape sides (duration, gaps, etc.). TURGEN then performs all necessary calculations and distributes files among tape sides automatically. The output is one large wave file per side.

To create the tape sides, select the *Create tape sides* item from the *Tools* menu.

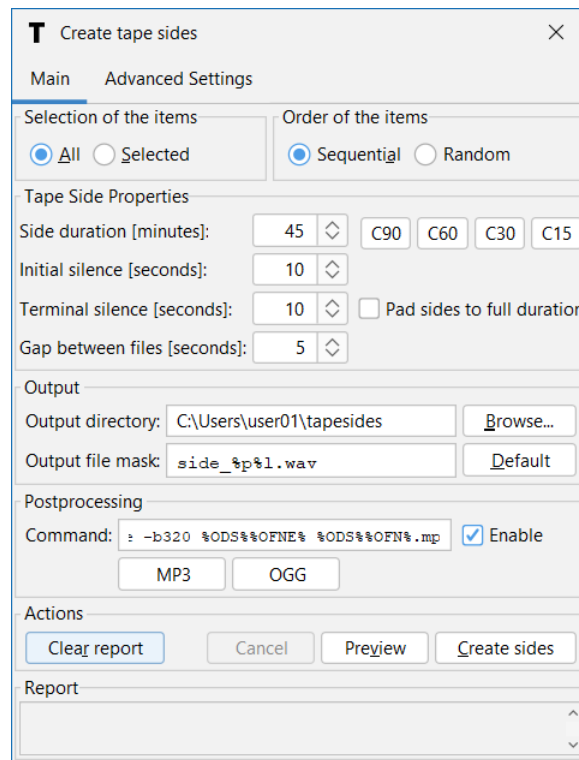


Figure 14: Creating tape sides

Enter the following information on your tape sides:

- Side duration in minutes. Enter how long is one tape side. You can click one of the *Cnn* buttons for predefined tapes. One side of a C90 tape is 45 minutes long.
- Initial silence in seconds. The silence is inserted before the first file on every side. This is used to skip leader tape.
- Terminal silence in seconds. The silence is inserted after the last file on every side. This is used to ensure some reserve at the end of the side.
- Pad sides to the full duration. If selected, then all sides are padded with silence to the full duration.
- Gap between files in seconds. The gaps are inserted after every file, except the last file of a side.
- Output directory. Select an existing directory.
- Output file mask. Enter a mask for output file names. The mask must contain either the %n symbol (absolute side number) or a combination of %p (side pair number) and %l (side letter - A for odd sides or B for even sides) symbols.
- Postprocessing command. The generated files can be passed to an external program. The syntax is described in section 9.1. You can click the *MP3*, *OGG*, and *FLAC* buttons to get predefined postprocessing commands for these file formats.

Then click the *Create sides* or *Preview* button to create sides or to preview the creation. The results (or errors) are reported in the *Report* box. To cancel the tape side creation in progress, click the *Cancel* button. Closing the dialog cancels the operation in progress too.

7.2.1 Advanced Settings

More settings are available in the *Advanced Settings* tabs.

Item list mask. Configure how a project item will appear in the listing of a side. You can use the following substitution symbols:

Symbol	Description
%f, %F	Full file name (original case, capitalized)
%n, %N	File name without extension (original case, capitalized)
%s	Start of the file (minutes, seconds)
%d	Duration of the file (minutes, seconds)
%cs	Start of the file (counter revolutions)
%cd	Duration of the file (counter revolutions)
%a,%A	Atari file name (original case, capitalized)

Tape counter. Configure how TURGEN converts samples to the counter revolutions. Use a stopwatch and your data recorder to measure how long it takes the counter to make 10 revolutions. Note that different recorders can have different counters.

7.2.2 Notes

- Internal calculations are always performed in samples. The values displayed in the report are rounded to minutes and seconds.
- The physical attributes of the generated signal (sampling rate, 8/16 bits, number of channels, waveform, etc.) are given by their respective configuration entries of the Wave output (see section 5.4).
- Some external programs (lame, oggenc) can generate large amounts of output. TURGEN captures the output and displays it in the *Report* box. It is recommended to specify command line options that reduce the verbosity of the output.

7.3 Creating Dual Track Tapes

7.3.1 Overview

This tool allows you to automatically create a “dual track tape” with voice-over or music in the left channel and data in the right channel.

7.3.2 System Requirements

To create dual track tapes, TURGEN requires *SoX* (Sound eXchange), a 3rd party open source tool for processing audio files available at <http://sox.sourceforge.net/>.

Before you begin, download and install *SoX* and ensure the *PATH* environment variable points to the directory where the *SoX* binary file is installed.

7.3.3 Operations

To create a dual track tape, select one project item and then select the *Create dual track* item from the *Tools* menu. A dialog will appear.

T Create dual track [X]

Playlist item and the audio track

Input file: C:\utils\...stuff\trbarchive\0_games_modern\drogawojownika.xex

Audio track: C:\utils\...a8\musicx.wav [Browse]

Output dual track

Output file: C:\utils\...a8\dual.ogg [Browse]

Options

Begin audio track after [s]: 34 [v]

☐ Fade in ☐ Fade out ☐ Repeat short audio tracks

Actions

[Clear report] [Cancel] [Create dual track]

Report

[Empty text area]

Figure 15: Creating a dual track tape

Enter the following information:

- Audio track. Enter a file name or click the *Browse* button to select the file. This file holds the music or voice-over.
- Output file. Enter a file name or click the *Browse* button to select the output file.
- Beginning of the audio track. Determines how many seconds from the beginning of the output file the audio track will start.
- Fade-in and Fade-out. Select when you want these effects applied to the audio track.
- Repeat a short audio track. Select this box when an audio track shorter than the data track should be repeated instead of being padded with silence.

Click the *Create dual track* button to create the dual track tape. The process will begin and messages will be displayed in the *Report* box.

Click the *Cancel* button to cancel the dual track creation. Click the *Clear report* button to clear the Report box.

7.3.4 How a Dual Track Is Created

TURGEN performs the following steps to create a dual track tape:

1. Generate a wave file from the selected project item - a data track.
2. Determine if the data track is long enough to accommodate the audio track.
3. Execute SOX to convert the audio track to a WAVE file.
4. Execute SOX to down-mix, trim, repeat, or pad the audio track, and apply the fade effects if requested.
5. Execute SOX to combine the data track and audio track.

7.3.5 What Audio Tracks Are Supported

- Common digitized sound files supported by SoX (.wav, .flac, .ogg, .mp3, and many others)
- SAP (Slight Atari Player). This requires the installation of the asapconv command-line tool from the ASAP project.

7.3.6 Best Practices

- Dual track tapes are only practically usable with the standard tape records.
- Prepare your audio track in a format supported by your SOX installation. Note that especially MP3 support might not be available or might require further installation steps.
- Always specify the output file with an extension, otherwise SOX won't be able to determine the file format.
- When creating standard project items, use either TSCBL or LBE binary loaders. Select the *Silent I/O* option, so that the audio track is not interrupted by the SIO beeping. Note that the SIO beeping will be disabled after the TSCBL loader is booted or after the LBE loader reads the first long block.
- The audio track should begin after the binary loader is booted, set the audio track's beginning accordingly (around 34 seconds).
- When repeating a short audio track, select the fade-in and fade-out effects.

7.4 Embedding Tokenized BASIC Files to Binary Files

7.4.1 Motivation

Some turbo systems define file formats that cannot store tokenized BASIC files. To circumvent that limitation, TURGEN has a tool that embeds tokenized BASIC files in monolithic binary files. **The purpose of the tool is embedding, not a compilation.** The resulting binary files work only with the original Atari BASIC.

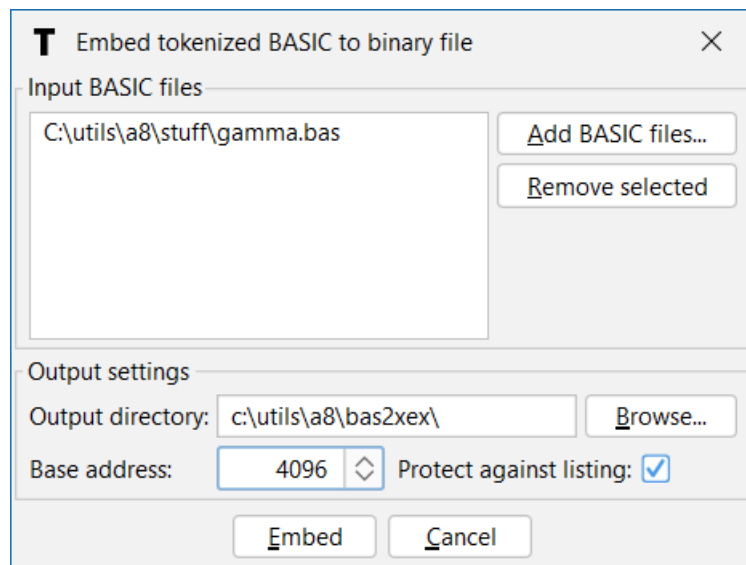


Figure 16: Embedding tokenized BASIC files to binary files

7.4.2 Operations

Open the embedding tool window by selecting the *Embed BASIC to binary file* item from the *Tools* menu. Click the *Add BASIC files...* button to display a file chooser that allows you to select BASIC files to be embedded in binary files. Chosen files are added to a list. You can select multiple files to be embedded. Click the *Remove selected* button to remove BASIC files from the list.

Use the *Output directory* text field to specify into which directory the binary files will be placed. Use the *Browse* button to specify the directory using a file chooser.

Use the *Base address* spinner to specify the address to which the BASIC files will be loaded (tokenized BASIC code is relocatable).

Use the *Protect against listing* check box to protect the BASIC program against listing by disabling the BREAK key and setting the COLDST flag. Note that such protection is only rudimentary and can be broken easily.

To embed the selected BASIC files in binary files, click the *Embed* button. All BASIC files in the list will be embedded in binary files. A report is displayed after the embedding processing finishes.

7.5 Creating Monolithic Binary Files

7.5.1 Introduction

TURGEN provides a tool that creates monolithic binary files by merging segments and replacing INIT vectors with JSR instructions. This allows the conversion of segmented binary files that meet certain conditions to monolithic binary files.

Conditions that the input segmented binary files must meet are the following:

1. Data segments must not overlap
2. Routines called through INIT vectors must work correctly even when all segments of the binary file are already loaded
3. There is a free memory area for the code that replaces the INIT segments

7.5.2 Creation Steps

1. Select the *Create monolithic binary file* item from the *Tools* menu to display the dialog for merging segments of a binary file.
2. Enter a file name in the *Input file* text field and click the *Analyze* button. The list of segments will be populated and attributes of the code that replaces INIT segments will be automatically detected.
3. Enter a file name in the *Output file* text field. Verify the address of the code that replaces the INIT segments and make changes if needed. Note that the code is needed only when the input binary file contains INIT segments.
4. Click the *Create* button to merge the segments and create the output monolithic binary file. Then test the resulting binary file.

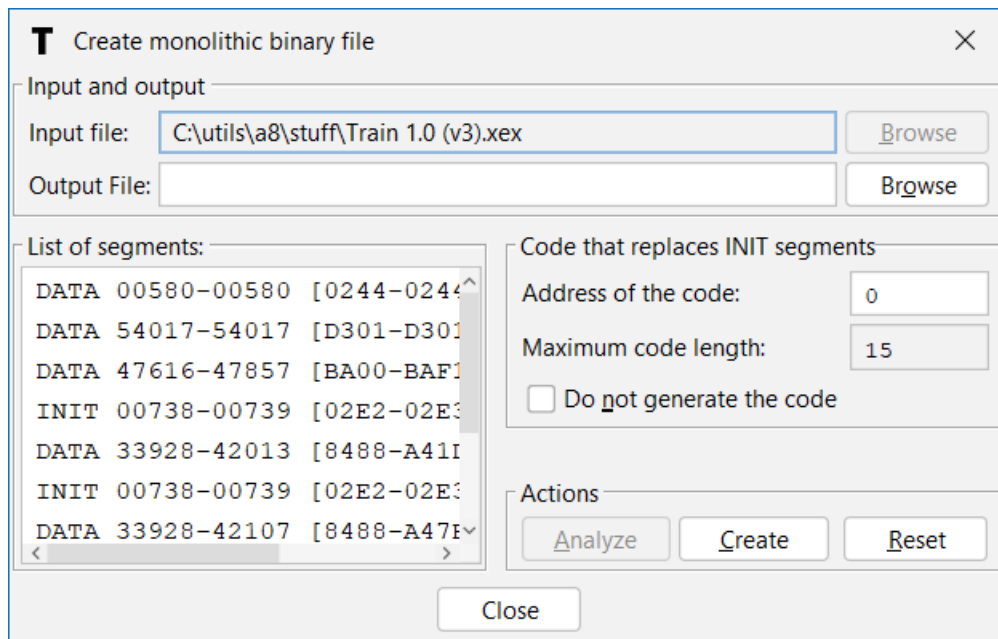


Figure 17: Creating monolithic binary file

7.6 Pilot Tone Test

TURGEN can generate a continuous pilot tone. Select the *Pilot Tone Test* item from the *Tools* menu. The controls allow you to select pulses. The pilot tone of the selected pulses is tested.

The parameters of the signal are those selected in the *Audio Generator* section of the *Preferences* dialog. Use this function to:

- Test your recording equipment
- Find the ideal signal amplitude when using a cassette adapter

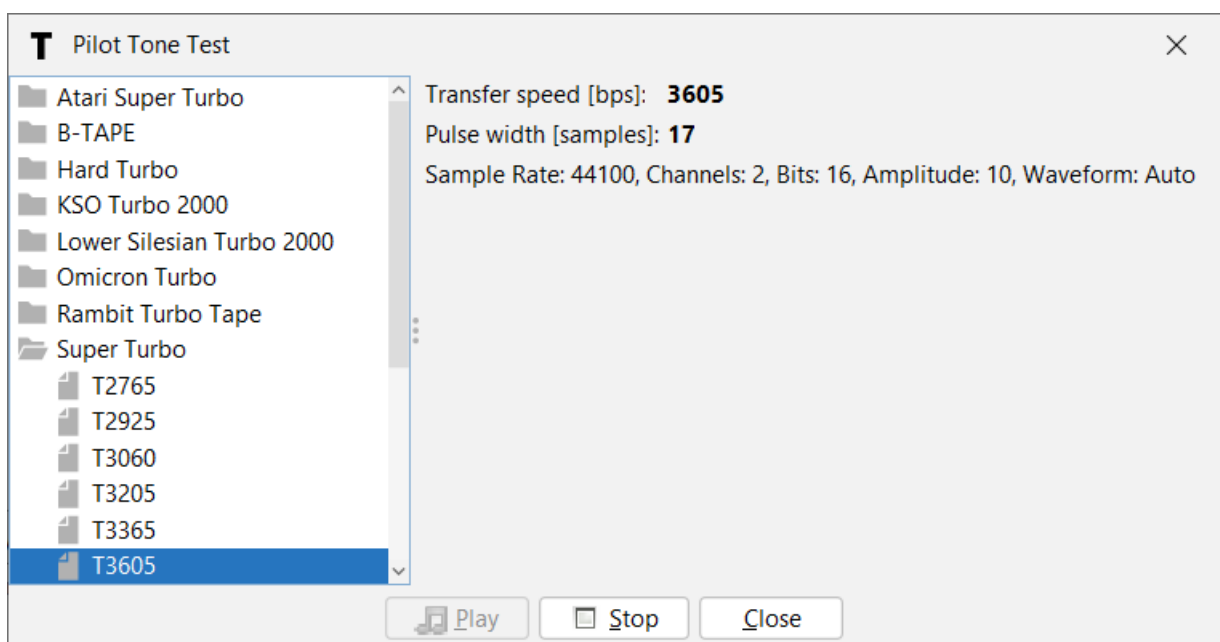


Figure 18: Pilot Tone Test

7.7 Batch Processing

7.7.1 Overview

Batch processing allows converting whole directories of binary files or tape images. The batch processing is performed in three steps:

1. Create a batch file that holds information on what files will be processed and how the files will be converted
2. Process the batch file and create project items
3. Process the project items

7.7.2 Syntax

Batch processing is controlled by a batch file that contains multiple *configurations*.

A configuration is defined by the following syntax:

```
BEGIN
INDIR <directory>
[OUTDIR <directory>]
CONVERTOR <convertor name>*
[SPEED <speed>]
[CONVERTOR-ORDER <LIST|PILOT-TONES|RANK>]
[TAPE-IMAGE-CONVERTOR <convertor name>]
[SORT <NONE|ALPHABETICAL|LISTING>]
[AUTO-SILENCE-LIST <duration>]
END
```

Blank lines and lines beginning with # are ignored. Each keyword must be specified on a separate line.

Keyword	Description
BEGIN	Each configuration must start with the BEGIN keyword.
INDIR	Identifies input directory with binary files. The directory specification can be absolute or relative. Files with the following extensions are processed: .xex,.com,.obj (binary files), and .cas (tape images).
OUTDIR	Identifies output directory (for wave files or tape images). The directory specification can be absolute or relative. If the keyword is not specified, all output file names are left empty.
CONVERTOR	Identifies binary file convertor to be used. You can specify the CONVERTOR keyword multiple times to allow multiple convertors. A binary file is converted by the first convertor that accepts it, unless you specify CONVERTOR-ORDER other than LISTING. Specify the CONVERTOR keywords in order from the most desirable convertor to the least desirable convertor. A convertor is identified by its name. Use the 'List assets' function to get the names of all available convertors.
SPEED	Identifies transfer speed to be used. Specify a decimal number. If the selected binary file convertor doesn't support the transfer speed, the closest available speed will be selected.

Keyword	Description
CONVERTOR-ORDER	Determines how the convertors will be selected to convert binary files. Specify LISTING to select the first convertor that can accept the binary file (this is the default). Specify PILOT-TONES to select a convertor that generates minimum number of pilot tones. Specify RANK to order convertors by ranking (more information on ranking in section 4.4.3).
TAPE-IMAGE-CONVERTOR	Identifies tape image convertor to be used. You can specify this keyword multiple times to allow multiple convertors. A tape image is converted by the first convertor that accepts it. Specify the TAPE-IMAGE-CONVERTOR keywords in order from the most desirable convertor to least desirable convertor. A convertor is identified by its name. Use the 'List assets' function to get the names of all available convertors.
SORT	Determines how files will added to the project. Specify NONE for order given by the file system. Specify ALPHABETICAL for alphabetical order. For full control of the order, specify LISTING for order determined by the listing.txt file located in the input directory. The listing.txt file contains list of file names, one name per line.
AUTO-SILENCE-LIST	Allows to automatically generate silence lists with specified default duration.
END	Each configuration must end with the END keyword.

7.7.3 Sample Batch File

```

BEGIN
INDIR /mnt/win_d/emulator/atari/exebin
OUTDIR ../waves2
CONVERTOR Turbo 2000 - Monolithic Binary
CONVERTOR Turbo 2000 - Blockloading
CONVERTOR-ORDER PILOT-TONES
SPEED 2270
SORT NONE
END

```

This will convert all binary files from the exebin directory. The files will be converted to Turbo 2000 as monolithic binary files if possible. If not, the files will be converted using the Blockloading conversion. Transfer speed will be approximately 2270 bd. The output files will be placed in the waves2 directory.

7.7.4 User Interface

To perform batch processing, select the *Process batch file* menu item from the *File* menu.

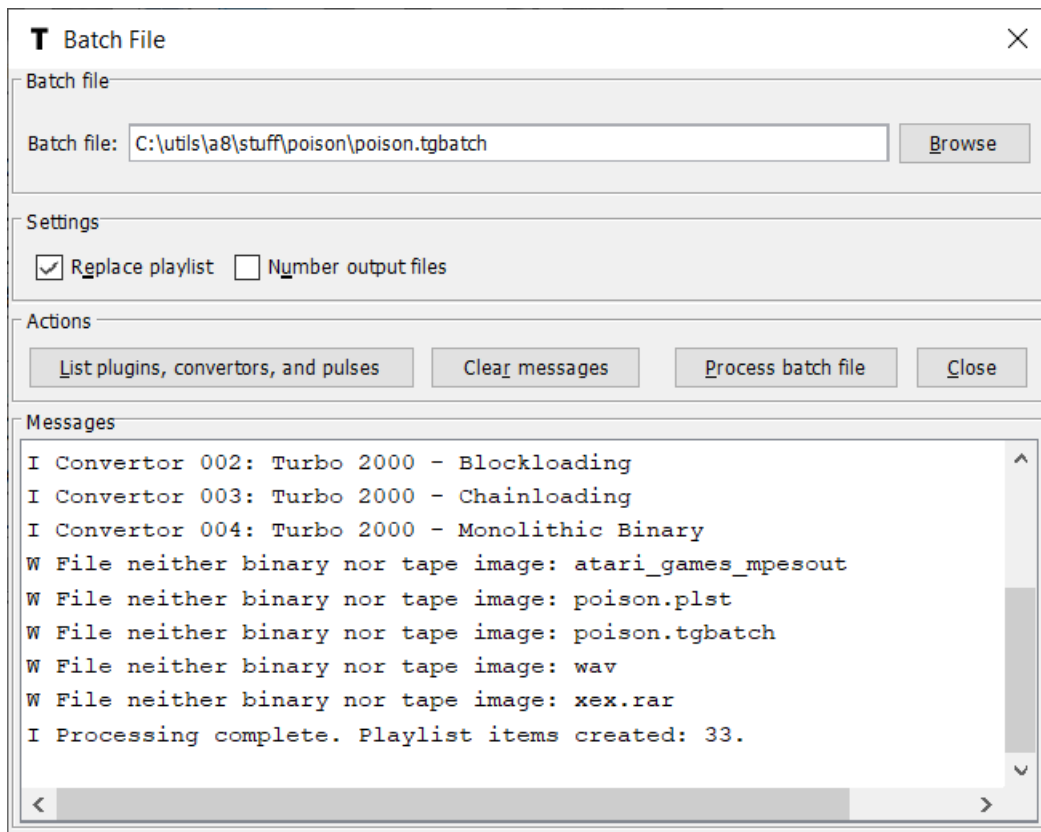


Figure 19: Batch Processing

- Enter the batch file in the *Batch file* text field or click the *Browse* button to select a batch file.
- Click the *Process batch file* button to process the selected batch file and create project items.
- Select the *Replace project* check box to replace all existing project items with items created by the batch processing. Select the *Number output files* check box to create project items with numbered output files.
- Click the *Close* button to close the dialog.
- Click the *Clear messages* button to clear all messages.
- Click the *List assets* button to display a listing of all available plugins, convertors, and transfer speeds.

7.8 GENCAS Command Line Tool

GENCAS is a command line tool that converts files to tape images or wave files. Both standard and turbo records are supported. GENCAS is intended to be a powerful replacement for the XEX2CAS tool.

The syntax is the following:

```
java -cp dist/turgen.jar turgen.gencas.CLIGenCas [-mode:mode] [options...] <input_file>
[output_file]
```

The java launcher syntax can be replaced with gencas.exe or gencas.sh convenience launchers, i.e:

```
gencas [-mode:mode] [options...] <input_file> [output_file]
```

7.8.1 Input and Output Files

- The input file must be specified. Only one input file is supported.
- The output file is optional. If the output file is not specified, it will be named automatically using the input file name and will be given the ".cas" extension or ".wav" extension.

7.8.2 General Syntax Guidelines

- The syntax is case sensitive
- File names, options, and option values that contain spaces or special characters interpreted by the command line interpreter or shell must be quoted with double quotes

7.8.3 Processing Modes

Mode	Description
binary	Convert binary file to standard tape records. This is the default mode, which emulates the behavior of the xex2cas tool.
plain	Convert input file as plain data.
bootbasic	Convert tokenized ATARI BASIC file to a bootable format
full	Full function mode that allows selecting any binary convertor.

7.8.4 General Options

Setup the conversion process in general

Option	Description
-r	Overwrite the output file if it exists.
-a	Append to the output file. The output file must already exist. This option is valid only when the output format is set to tape image. This option is mutually exclusive with the -r option. When this option is specified, no FUJI chunks are appended to the output file.
-spd:<n>	Transfer speed. 400-1200 bd. The option is not valid for the full mode. The default value is 600.
-fuji:<string>	FUJI chunk string. The default is an empty string.
-s	Short leader tone (14 s). The default is 20 s.
-o:<string>	Output format. Specify "cas" for tape image output (the default), or "wav" for wave output.

7.8.5 Binary Processing Mode Options

Set up the conversion of binary load files to standard tape records. Some of the options are fully supported only by the TSCBL and LBE binary loaders.

Option	Description
-ldr:<n>	Select binary loader. 0 no loader; 1 TSCBL (the default); 2 X-Loader, 3 LBE Loader (the input file will be converted to the LBE format)
-bg:<n>	Background color. 0-255. The default is 148.
-fg:<n>	Foreground luminance. 0-15. The default is 10.
-title:<string>	Program title displayed by the binary loader. If not specified, the program title is derived from the input file name.
-crsinh:<n>	Cursor inhibition. 0 Cursor visible; 1 cursor not visible. The default is 0.
-soundr:<n>	Noisy I/O. 0 Silent ; 1 noisy. The default is 1
-noatract	Suppress the ATRACT mode.

7.8.6 Plain Processing Mode Options

Set up the conversion of plain files.

Option	Description
-long	Long gaps between records
-eoftrick:<n>	Use trick with the EOF block. 0 No trick (the default); 1 Data in EOF block trick.

7.8.7 Bootbasic Processing Mode Options

There are no options specific to this mode.

7.8.8 Full Processing Mode Options

Setup conversion of binary files using any of the available binary convertors.

Option	Description
-conv:<string>	Binary convertor name. This option is mandatory.
-spd:<int>	Transfer speed. Specify a decimal number. If not specified, the transfer speed default for the selected binary convertor will be used.
-conf:<string>	Custom configuration file. If not specified, the program configuration file is used.

7.8.9 Notes on the Full Mode

When using the full mode, be advised of the following:

- This mode fully initializes all available plugins, so the conversion can take longer time
- The configuration is loaded from the program configuration file. When performing the conversion, the configuration entries of the respective plugin are fully honored
- When specifying a custom configuration file, you do not have to specify all entries. You can only specify a subset of entries.
- To get a list available convertors and transfer speeds, use the 'List Assets' function described in section 7.7.

7.8.10 Silence List Options

Control the creation of the silence list. These options are valid for the binary and full modes only. These two options are mutually exclusive.

Option	Description
-silencelist:<string>	Use a silence list. For information about syntax of the silence list, refer to section 4.8.
-autosilence:<n>	Automatically generate a silence list. For each INIT segment, generate <i>n</i> seconds of silence or neutral signal.

7.8.11 Examples

Convert a binary file to standard tape records, use the TSCBL loader, display the "HELLO TAPE" title and hide the cursor. Increase transfer speed to 660 bd.

```
gencas.exe -ldr:1 -spd:660 "-title:HELLO TAPE" -crsinh:1 program.xex tape.cas
```

Convert a binary file to the Turbo Blizzard system, default transfer speed.

```
gencas.exe mode:full "-conv:Turbo Blizzard" program.xex tape_blizzard.cas
```

8 Hardware that Can Be Used with TURGEN

8.1 Data Recorders and Compact Cassettes

For the best experience with TURGEN, use a data recorder with a turbo upgrade and quality compact cassettes.

8.2 Cassette Adapters

If you do not wish or cannot use compact cassettes, you can try an affordable and available cassette adapter. This device looks like a compact cassette, but there is no tape inside. Instead of using the tape, the cassette adapter is connected to an external source of the signal using a thin cable with a JACK connector.

The cassette adapter is very versatile, as it can be connected to various devices like CD players, MP3 players, or similar devices. Also do not forget that the cassette adapter will work with data recorders that can no longer spin the reels, but their electrical circuitry is still intact.

The Cassette adapter is most useful when connected to a computer running TURGEN.

To use the cassette adapter, place it into your data recorder (ensure that its top side is oriented correctly) and connect it to an external signal source. Familiarize yourself with the concept of silence lists (see section 4.8) for reasons stated in section 8.5.

8.3 MEGA-CD Interface (CD-LINK)

This interface allows you to plug an external signal source as it would be a data recorder with Czechoslovak Super Turbo upgrade. In essence, it is an external version of Super Turbo circuitry with a pair of RCA connectors and an SIO connector.

You can use the MEGA-CD interface to transfer records for systems that are hardware-compatible with Czechoslovak Turbo 2000 and Super Turbo upgrades.

8.4 SIO2PC and ATART

Both devices allow the transfer of standard tape records. Supporting software can process tape images.

8.5 Loading Binary Files with Replacement Devices

Replacement devices. The replacement devices are all devices that are intended to replace data recorders and/or compact cassettes.

Some examples of typical replacement devices:

- An MP3 player connected to a cassette adapter inserted into the data recorder.
- A SIO2PC device connected to a computer running a data recorder emulation (e.g. AspeQT).
- MEGA-CD interface

MOTOR CTRL. The Atari computer uses a dedicated data recorder specially designed to cooperate with the computer. One advantage of this design is that the Atari computer can programmatically control the data recorder.

There is a dedicated wire (MOTOR CTRL) in the SIO connector that allows the computer to switch the motor of the data recorder on and off. You might have already noticed, that if you just press the PLAY button, the cassette stands still until the computer is instructed to load data from the data recorder (e.g. by entering the CLOAD BASIC command or booting from tape).

When loading a binary file with an INIT segment, the computer will stop the data recorder, execute machine code identified by the INIT segment, and then re-start the data recorder. Execution of the code can take several seconds, or the code can wait for input from the user.

Without MOTOR CTRL. The replacement devices **cannot be controlled by the Atari computer**. These devices are simply not connected to the MOTOR CTRL wire, so they cannot obey the computer's orders to start or stop.

This problem surfaces, when a binary file contains INIT vectors. When the machine code is executed, the replacement device simply keeps sending the signal to the computer (instead of waiting for the MOTOR CTRL signal). This way, blocks or segments can be skipped and the loading fails or the computer just hangs.

Circumvention. To mitigate the consequences of the absence of the MOTOR CTRL signal, TURGEN supports *silence lists*. The silence lists allow to generate silence or pause the signal generator after blocks that contain the INIT vectors. If you are using a replacement device, it is strongly recommended that you make yourself familiar with the silence lists and use them.

For more information about the silence lists, refer to section 4.8.

9 Postprocessing

TURGEN allows the processing of its outputs by external programs. This is called *postprocessing*. After the output file is created, a command line is executed.

To identify the TURGEN's outputs in the command line, some special symbols are introduced. The symbols are described in table 4.

If the command line starts with the exclamation mark (!), the output file is deleted after the external program terminates.

Command lines (after special symbol substitution) are always stored in the log file, so the correctness of command lines can be verified.

Symbol	Meaning
%OD%	Output directory
%ODS%	Output directory followed by file separator
%OFN%	Output file name without last extension
%OFNE%	Output file name with last extension

Table 4: Special symbols

9.1 Postprocessing of WAVE Files

To enable the postprocessing of the WAVE files, select the *Wave postprocessing* check box menu item from the *Generator* menu. Specify the command line in the Wave output/Postprocessing command configuration entry.

9.2 Postprocessing of Tape Images

To enable postprocessing of the tape images, select the *Tape image postprocessing* check box menu item from the *Generator* menu. Specify the command line in the Tape image output/Postprocessing command configuration entry.

9.3 Examples

Example of the special symbols for output file: /mnt/win_d/emulator/atari/wav/river_raid.wav

%OD% = /mnt/win_d/emulator/atari/wav

%ODS% = /mnt/win_d/emulator/atari/wav/

%OFNE% = river_raid.wav

%OFN% = river_raid

Example of conversion to MP3:

```
lame %ODS%%OFNE% %ODS%%OFN%.mp3
```

Example of conversion to OGG Vorbis with deletion of the original file:

```
!oggenc -o %ODS%%OFN%.ogg %ODS%%OFNE%
```

Conversion to MP3 using LAME in the terminal emulator and consequent playback by the xmms media player. The original file is deleted:

```
!konsole -e bash -c "lame %ODS%%OFNE% %ODS%%OFN%.mp3 && xmms %ODS%%OFN%.mp3"
```

10 Solving Common Problems

When you cannot transfer your programs and games to your Atari 8-bit computer, try the following solutions.

Volume. Try to change the volume (signal amplitude). Use available controls (*Amplitude* configuration entries, volume controls provided by your operating system, potentiometers, etc.) to change the volume. Use the Pilot Tone Test tool described in section 7.6 to find the appropriate volume.

The Polarity of the Pulses. Some turbo systems (e.g. Ramba Turbo Tape, Turbo 6000) require certain polarity of the pulses. Devices in your recording chain can sometimes reverse the polarity. Use the *Invert polarity of the pulses* configuration entries to solve this problem.

Waveform. Devices in your recording chain can have maximum signal frequency limits. Choose a waveform that is suitable for the weakest device in your recording chain (it is usually the tape). For more information, refer to section 4.5.4.

Standard Tape Records. Before you try any of the turbo systems, check if you can transfer standard tape records. When you are not able to transfer standard tape records, there is probably a serious problem with your recording chain.

System Sounds. The electric signal can be disturbed by system sounds (notification bubbles, error messages, background tasks). When using the *Generate AUDIO directly* function, check that the system sounds are disabled.

Cassette Adapters. When using a cassette adapter, ensure that it is placed properly in your data recorder.

11 Production of Multiple Tapes

TURGEN offers several facilities that will help you to produce multiple tapes with a high degree of automation. Review the following sections to learn more.

1. Batch Processing - section 7.7
Batch processing allows automatic creation of a project from a directory of binary files and tape images
2. Wizard for files - section 4.4
Wizard allows the processing of multiple files and creating a project.
3. Automatic File Name Creation - section 4.9
This allows automatic generation of good file names that are following the rules of a particular turbo system
4. Creation of tape sides - section 7.2
This tool allows you to easily create tape sides of defined lengths.
5. Copying project as CSV - section 4.3.4
You can copy project items to the system clipboard as comma-separated values. This helps to create printed file listings or album art.

Part II

Plugins

12 Turbo 2000 and Super Turbo

12.1 Overview

The plugins convert input files to Czechoslovak turbo systems **Turbo 2000** and **Super Turbo**. Both plugins also provide special conversion types for segmented binary files, which were not supported by the original file format. The user interface is depicted in figure 20.

The screenshot shows a 'Project Item' dialog box with the following fields and controls:

- Plugin:** Turbo 2000 (with a Czech flag icon and a dropdown arrow).
- Input file:** C:\utils\8\stuff\Train 1.0 (v3).xex (with a 'Browse...' button).
- Output file:** C:\utils\8\stuff\Train 1.0 (v3).wav (with a 'Browse...' button).
- Conversion type:** BlockLoading of segmented binary file (with a 'Check loader' button).
- Header block:**
 - File name:** TRAIN .XEX
 - Type:** (empty text box)
 - Load:** (empty text box) **Length:** (empty text box) **Run:** (empty text box)
 - Buttons:** Auto set header, Load header
- Binary file:**
 - Silence list:** 1,1,1,1,1 (with 'S', 'P', '1.0' buttons and a dropdown arrow).
 - ☐ Do not prepend binary loader
- Transfer speed:** 2270 (with a dropdown arrow).
- Buttons:** OK, Cancel, Clear

Figure 20: Turbo 2000, Super Turbo

12.2 Conversion types

Turbo 2000 and Super Turbo plugins support 6 conversion types. Select the conversion type using the *Conversion type* combo box.

Monolithic binary file to Turbo 2000 or Super Turbo. Conversion of a monolithic binary file (see 2.1.4). The input file must be a monolithic binary file. This is a natural usage of Turbo 2000 or Super Turbo systems.

ChainLoading. Conversion of a segmented binary file to a *chain of Turbo 2000 or Super Turbo block pairs*. A special binary loader (Chainloader 2) that loads the block pairs is prepended before the chain. Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

BlockLoading. Conversion of a segmented binary file to a *chain of Turbo 2000 or Super Turbo blocks*. A special binary loader (Blockloader) that loads the blocks (and stores information where to place the blocks in the memory) is prepended before the chain. The binary file can have up to 62 segments. Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

Tokenized BASIC to Turbo 2000 or Super Turbo. Conversion of a tokenized BASIC program. Note that many loaders cannot load such program directly. To circumvent such limitation, you can embed tokenized BASIC to binary file. See section 7.4.

Plain DATA to Turbo 2000 or Super Turbo. Conversion of plain data. You must enter values in the text fields related to the header block.

Binary file to binary turbo. Conversion of a binary file to the Binary turbo (file type 4) format. Binary files can be stored in this format, but under normal circumstances, you cannot load and run them directly. There are two known programs that provide special functionality for these files:

1. VisiCopy III (a commercial product by JRC) allows you to run such binary files, but only if they do not contain INIT segments.
2. ChainCopy 1.2 (auxiliary utility for TURGEN) can read these files and convert them as the Chain-Loading conversion would do.

ExpressLoading. Conversion of a segmented binary file to a *chain of Turbo 2000 composite blocks*. Each composite block can contain data from multiple segments and ends with either an INIT segment or end of file. A special binary loader (ExpressLoader) is prepended before the composite block chain. Click the *Check loader* button to determine if the binary loader can load the selected input binary file. ExpressLoading is the fastest conversion type for segmented binary files because it utilizes the minimum number of pilot tones possible. ExpressLoading is available only for Turbo 2000.

12.3 Header Block

Use the text fields *File name*, *Type*, *Load*, *Length* and *Run* to specify entries of the Turbo 2000 or Super Turbo header block.

Click the *Auto set header* button to automatically enter values in the text fields. The information is obtained by an analysis of the input file and the selected conversion type. If the input file is incompatible with the selected conversion type, a warning dialog with recommendations is displayed.

Click the *Load header* to load the turbo header from an external file. The header is loaded automatically if it is stored in a file that is named the same as the input file, followed by a *.thead*er suffix. In other cases, a file chooser is displayed.

12.4 Inserting Silence After INIT Segments

Use the *Silence list* panel to create a silence list (refer to section 4.8). This is applicable only for Chain-Loading and BlockLoading conversion types.

12.5 Prepending Universal Turbo Loader

If you do not have a cartridge with Turbo 2000 or Universal Turbo loader, you can prepend a tape version of such loader. Set the *Prepend the Universal Turbo loader* configuration entry to *true*.

12.6 Using Custom Binary Loaders

Select the *Do not prepend binary loader* check box if you intend to use your own custom version of Chain-loader 2, Blockloader or ExpressLoader. The file will be converted, but the loader will not be prepended, so you will be able to use your own.

12.7 Configuration Entries

Turbo 2000/Invert polarity of the pulses

Invert polarity of the pulses

Turbo 2000/Header block pilot tone duration

Number of pilot tone pulses of the header block (256-8192)

Turbo 2000/Data block pilot tone duration

Number of pilot tone pulses of the data block (256-8192)

Turbo 2000/BlockLoading pilot tone duration

Number of pilot tone pulses of data blocks for BlockLoading conversion type (256-8192)

Turbo 2000/Silence after header

Duration of silence that is generated after the header block (0-30). Note that many Universal Turbo loaders do not tolerate any silence after header.

Turbo 2000/Pause after header

If enabled, output of electric signal to the sound card will be paused after the header block. This can be useful for Turbo 2000 only loaders that wait for a key press. Note that many Universal Turbo loaders do not tolerate any interruptions or silence after the header block.

Turbo 2000/Prepend the Universal Turbo loader

Prepend the Universal Turbo loader as a standard tape boot file

Turbo 2000/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

Super Turbo/Invert polarity of the pulses

Invert polarity of the pulses

Super Turbo/Header block pilot tone duration

Number of pilot tone pulses of the header block (256-8192)

Super Turbo/Data block pilot tone duration

Number of pilot tone pulses of the data block (256-8192)

Super Turbo/BlockLoading pilot tone duration

Number of pilot tone pulses of data blocks for BlockLoading conversion type (256-8192)

Super Turbo/Prolongate pilot tone

Increase number of pilot tone pulses with increasing approximate baud rate

Super Turbo/Prolongate pilot tone for BlockLoading

Increase number of pilot tone pulses with increasing approximate baud rate for BlockLoading conversion type

Super Turbo/Silence after header

Duration of silence that is generated after the header block (0-30). Note that many Universal Turbo loaders do not tolerate any silence after header.

Super Turbo/Convert binary loaders to Turbo 2000

If enabled, the Chainloader and Blockloader binary loaders are converted to Turbo 2000 instead of Super Turbo.

Super Turbo/Prepend the Universal Turbo loader

Prepend the Universal Turbo loader as a standard tape boot file

Super Turbo/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

13 Turbo 2000 - Kilobyte Blocks

13.1 Overview

The plugin converts files to the **Turbo 2000 - Kilobyte Blocks** Czechoslovak turbo system. There are no special restrictions on input files.

13.2 User Interface

The user interface is depicted in figure 21.

Use the *File name* text field to specify the file name that will be displayed on screen. Use the *Silence list* panel to create a silence list.

The screenshot shows a 'Project Item' dialog box. The title bar includes a 'T' icon and a close button. The 'Plugin' section shows 'Turbo 2000 - Kilobyte Blocks' with a dropdown arrow. The 'Input file' section has a text field with 'C:\utils\8\stuff\Train 1.0 (v3).xex' and a 'Browse...' button. The 'Output file' section has a text field with 'C:\utils\8\stuff\Train 1.0 (v3).wav' and a 'Browse...' button. The 'Header block' section has a 'File name' text field with 'TRAIN 1.0 (V.XEX)' and an 'Auto set header' button. The 'Binary file' section has a 'Binary loader' dropdown with 'NanoTBL' selected, a 'Check loader' button, a 'Silence list' text field with '1,1,1,1,1', and three buttons labeled 'S', 'P', and '1.0'. The 'Transfer speed' section has a dropdown menu showing '2270'. At the bottom are 'OK', 'Cancel', and 'Clear' buttons.

Figure 21: Turbo 2000 - kilobyte blocks

13.3 Binary Loaders

Turbo 2000 - kilobyte blocks turbo system can store binary files. You can prepend one of the special miniature built-in binary loaders. For this turbo system, you can choose MiniTBL, NanoTBL, NanoTBL[UR] or NanoTBL[U2] loader by using the *Binary loader* combo box. The loader is converted to the Turbo 2000 system.

The MiniTBL loader is a combination of a stripped "T:" device handler that allows only READ operation and code for binary load using CIO.

The NanoTBL loaders are single-purpose binary loaders reading data blocks, not using CIO at all. NanoTBL stores data blocks to the freely available RAM. NanoTBL[UR] stores the data blocks to the beginning of the "RAM under ROM", NanoTBL[U2] to the end of "RAM under ROM".

Click the *Check loader* button to determine if the selected binary loader can load the input binary file..

13.4 Prepending Universal Turbo Loader

If you don't have a cartridge with Turbo 2000 or Universal Turbo loader, you can prepend a tape version of such loader. Set the *Prepend the Universal Turbo loader* configuration entry to *true*.

13.5 Configuration Entries

Turbo 2000 - Kilobyte blocks/Invert polarity of the pulses

Invert polarity of the pulses

Turbo 2000 - Kilobyte Blocks/Header block pilot tone duration

Number of pilot tone pulses of the header block (256-8192)

Turbo 2000 - Kilobyte Blocks/Data block pilot tone duration
Number of pilot tone pulses of the data blocks (256-8192)
Turbo 2000 - Kilobyte Blocks/Silence after header
Number of seconds of the silence inserted after the header block (0-30)
Turbo 2000 - Kilobyte Blocks/Silence after loader
Number of seconds of the silence inserted after the binary loader (0-30)
Turbo 2000 - Kilobyte Blocks/Cumulate silence for blocks with INIT vectors
Indicates whether to cumulate silence. Refer to section 4.8.5.
Turbo 2000 - Kilobyte Blocks/Name the loader same as file
The binary loader will have name same as the converted file
Turbo 2000 - Kilobyte Blocks/Prepend the Universal Turbo loader
Prepend the Universal Turbo loader as a standard tape boot file
Turbo 2000 - Kilobyte Blocks/File name creation options
Configure the automatic file name creation. Refer to section 4.9.

14 B-TAPE

14.1 Overview

The plugin converts files to the **B-TAPE** Czechoslovak turbo system. There are no special restrictions on input files. The B-TAPE system is backward compatible with the Turbo Tape system used by the TT-DOS operating system.

14.2 User Interface

The user interface is depicted in figure 22.

Use the *File name* text field to specify the file name that will be displayed on screen. Use the *Tape mode* combo box to select tape mode. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

14.3 Binary Loaders

The B-TAPE turbo system can store binary files. You can prepend one of the special miniature built-in binary loaders. For this turbo system, you can choose NanoBTAPE, NanoBTAPE[UR] or NanoBTAPE[U2] loader using the *Binary loader* combo box. The loader is converted to the Turbo 2000 system.

The NanoBTAPE loaders are single-purpose binary loaders reading blocks not using CIO. NanoBTAPE stores the blocks to the freely available RAM. NanoBTAPE[UR] stores blocks to the beginning of the “RAM under ROM”, NanoBTAPE[U2] to the end of “RAM under ROM”.

Click the *Check loader* button to determine if the selected binary loader can load the input binary file.

14.4 Transfer Speed

The lowest transfer speed supported by the original B-TAPE and Turbo Tape operating systems and loaders is 2765 bd. The B-TAPE plugin allows you to select lower transfer speeds (matching Turbo 2000), but the original loaders will not be able to load such files. For compatibility with the original loaders, always set transfer speed greater or equal to 2765 bd.

14.5 Prepending Universal Turbo Loader

If you don't have a cartridge with Turbo 2000 or Universal Turbo loader, you can prepend a tape version of such loader. Set the *Prepend the Universal Turbo loader* configuration entry to *true*.

T Project Item

Plugin: B-TAPE

Input file: C:\utils\8\stuff\Train 1.0 (v3).xex Browse...

Output file: C:\utils\8\stuff\Train 1.0 (v3).wav Browse...

Header

File name: TRAIN 10.XEX

Tape mode: [SS] Short gaps, single blocks

Auto set header

Binary file

Binary loader: NanoBTAPE[UR] Check loader

Silence list: 1,1,1,1,1 S P 1.0

Transfer speed

2270

OK Cancel Clear

Figure 22: B-TAPE

14.6 Configuration Entries

B-TAPE/Invert polarity of the pulses

Invert polarity of the pulses

B-TAPE/Pilot tone duration

Number of pilot tone pulses of the blocks (256-8192)

B-TAPE/Prolongate pilot tone

Increase number of pilot tone pulses with increasing approximate baud rate

B-TAPE/Silence after first block

Number of seconds of the silence inserted after the first B-TAPE block (0-30)

B-TAPE/Silence after loader

Number of seconds of the silence inserted after the binary loader (0-30)

B-TAPE/Cumulate silence for blocks with INIT vectors

Indicates whether to cumulate silence. Refer to section 4.8.5.

B-TAPE/Name the loader same as file

The binary loader will have name same as the converted file

B-TAPE/Prepend the Universal Turbo loader

Prepend the Universal Turbo loader as a standard tape boot file

B-TAPE/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

15 KSO Turbo 2000

15.1 Overview

The plugin converts files to the **KSO Turbo 2000** Polish turbo system. The KSO Turbo 2000 system is compatible with various similar turbo systems (e.g., Turbo 2000F).

15.2 Conversion types

15.2.1 Natural Format

This conversion converts any file to the natural file format defined by the KSO Turbo 2000. There are no special restrictions on input files.

15.2.2 Binary File to Speedy2700

This conversion converts binary files to the Speedy2700 format. The Speedy 2700 format is an efficient format for storing and loading binary files. The conversion is performed as follows:

1. The Speedy2700 loader is converted to the natural file format.
2. The input binary file is converted to the Speedy2700 format

15.2.3 Binary File to L3 Format

This conversion converts binary files to a special L3 file format designed for loading of binary files with long segments. The conversion is performed as follows:

1. The L3 Binary loader is converted to the natural file format
2. The input binary file is converted to the special L3 file format

T Project Item

Plugin: KSO Turbo 2000

Input file: C:\utils\8\stuff\Train 1.0 (v3).xex Browse...

Output file: C:\utils\8\stuff\Train 1.0 (v3).wav Browse...

Conversion type: Binary file to Speedy 2700 Check loader

Header block: File name: TRAIN .XEX Auto set header

Binary file: Silence list: 1,1,1,1,1 S P 1.0

Transfer speed: 3060

OK Cancel Clear

Figure 23: KSO Turbo 2000

15.3 User Interface

The user interface is depicted in figure 23.

Use the *Conversion type* combo box to select conversion type. Click the *Check loader* button to determine if the binary loader can load the selected input binary file. Enter a file name in the *File name* text field. Click the *Auto set header* button to automatically create the file name. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

15.4 Configuration Entries

KSO Turbo 2000/Invert polarity of the pulses

Invert polarity of the pulses

KSO Turbo 2000/Header block pilot tone duration

Number of pilot tone pulses of the header block (256-8192)

KSO Turbo 2000/Data block pilot tone duration

Number of pilot tone pulses of the data blocks (256-8192)

KSO Turbo 2000/Long gaps between blocks

Generate long (1.5 s) gaps between the 3-KB blocks. Recommended when using a cassette adapter.

KSO Turbo 2000/Silence after header

Number of seconds of the silence inserted after the header block (0-30)

KSO Turbo 2000/Cumulate silence for blocks with INIT vectors

Indicates whether to cumulate silence. Refer to section 4.8.5.

KSO Turbo 2000/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

KSO Turbo 2000/Convert binary loaders to tape boot files

If enabled, binary loaders (L3, Speedy2700) will be converted to tape boot files

KSO Turbo 2000/Pulse Corrections

Corrections of pulses

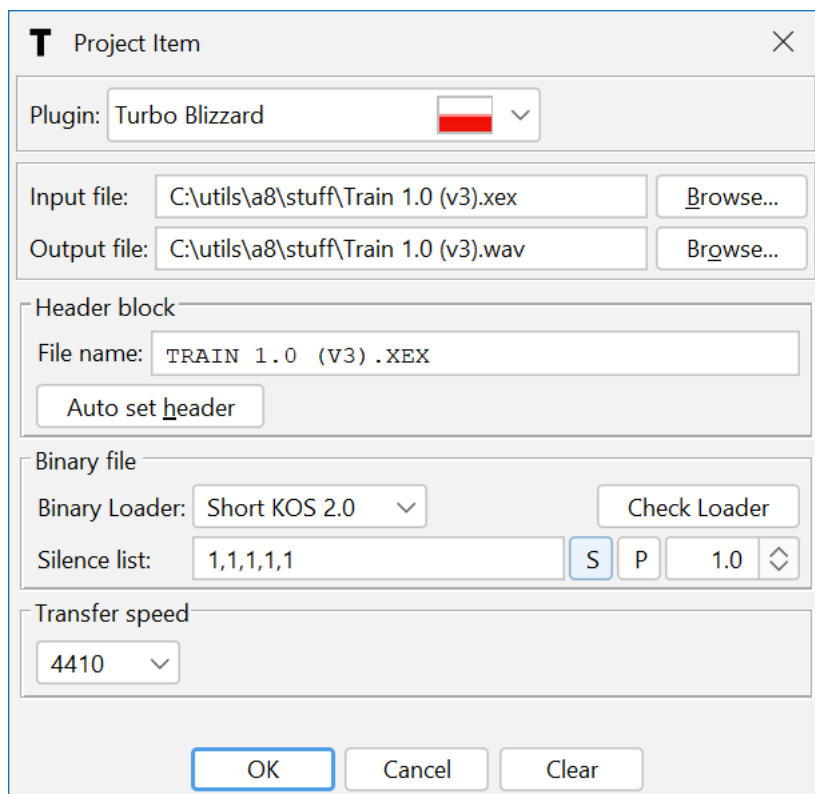
KSO Turbo 2000/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

16 Turbo Blizzard

16.1 Overview

The plugin converts files to the **Turbo Blizzard** Polish turbo system. There are no special restrictions on input files.



The screenshot shows a dialog box titled "Project Item" with a close button (X) in the top right corner. The dialog is organized into several sections:

- Plugin:** A dropdown menu showing "Turbo Blizzard" with a red progress bar and a downward arrow.
- Input file:** A text field containing "C:\utils\8\stuff\Train 1.0 (v3).xex" and a "Browse..." button.
- Output file:** A text field containing "C:\utils\8\stuff\Train 1.0 (v3).wav" and a "Browse..." button.
- Header block:** A section with a "File name:" label and a text field containing "TRAIN 1.0 (V3) .XEX". Below it is an "Auto set header" button.
- Binary file:** A section with a "Binary Loader:" label and a dropdown menu showing "Short KOS 2.0". To the right is a "Check Loader" button. Below this is a "Silence list:" label and a text field containing "1,1,1,1,1". To the right of the text field are three buttons: "S", "P", and "1.0", followed by a small diamond icon.
- Transfer speed:** A section with a dropdown menu showing "4410".

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Clear".

Figure 24: Turbo Blizzard

16.2 User Interface

The user interface is depicted in figure 24.

Use the *File name* text field to specify a file name that will be displayed on screen. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

Use the *Binary loader* combo box to prepend a binary loader. You can choose to prepend none, Microloader 3.0, or ShortKOS 2.0. Click the *Check loader* button to determine if the selected binary loader can load the input binary file.

16.3 Sampling Rate

The Turbo Blizzard plugin take advantage of the sampling rate of 48000 Hz.

16.4 Configuration Entries

Turbo Blizzard/Invert polarity of the pulses

Invert polarity of the pulses

Turbo Blizzard/Long gaps between blocks

Generate long gaps between blocks

Turbo Blizzard/Silence after header

Number of seconds of the silence inserted after the header block (0-30)

Turbo Blizzard/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

Turbo Blizzard/Convert binary loaders to tape boot files

If enabled, the binary loaders are converted to tape boot files. Otherwise, the loaders are converted to Turbo Blizzard.

Turbo Blizzard/Silence after loader

Duration of silence generated after binary loader

Turbo Blizzard/Cumulate silence for blocks with INIT vectors

Indicates whether to cumulate silence. Refer to section 4.8.5.

Turbo Blizzard/Pulse Corrections

Corrections of pulses

Turbo Blizzard/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

17 Turbo ROM

17.1 Overview

The plugin converts files to the **Turbo ROM** Polish turbo system. The following input files are supported:

- Turbo ROM compatible binary files
- Binary files
- Tokenized BASIC files

T Project Item [X]

Plugin: Turbo ROM [v]

Input file: C:\utils\8\stuff\Train 1.0 (v3).xex [Browse...]

Output file: C:\utils\8\stuff\Train 1.0 (v3).wav [Browse...]

Input file type: Binary file [v] [Check loader]

Header block

File name: TRAIN 1.0 (V3) .XEX

[Auto set header]

Binary file

Silence list: 1,1,1,1,1 [S] [P] [1.0] [v]

Transfer speed: 4000 [v]

[OK] [Cancel] [Clear]

Figure 25: Turbo ROM

17.2 User Interface

The user interface is depicted in figure 25.

Use the *File name* text field to specify a file name that will be displayed on screen. Use the *Auto set header* button to automatically set file name and also to verify whether the input file is a Turbo ROM compatible binary file.

Use the *Input file type* combo box to select input file type.

17.3 Conversion of Turbo ROM Compatible Binary Files

These binary files consist of exactly one DATA segment and at most one RUN segment and at most one INIT segment. Files are converted to the Turbo ROM natural format.

17.4 Conversion of Binary Files

17.4.1 Processing

Binary Files are converted to a special format. A special binary loader in the Turbo ROM natural format is prepended. The binary files are normalized before conversion (Combined RUN+INIT segments are split, RUN segment is moved to the end of the file).

17.4.2 User Interface

Click the *Check loader* button to determine if the binary loader can load the selected input binary file. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

17.5 Conversion of Tokenized BASIC Files

Files are converted to the Turbo ROM natural format for tokenized BASIC files.

17.6 Prepending Loaders

If you do not have a cartridge with Turbo ROM loader, you can prepend a tape version of such loader. Set the *Prepend Turbo ROM loader* configuration entry to deserved value. Set the *Convert binary loader to a standard tape boot file* configuration entry to *true*.

17.7 Sampling Rate

The Turbo ROM plugin take advantage of the sampling rate of 48000 Hz.

17.8 Configuration Entries

Turbo ROM/Invert polarity of the pulses

Invert polarity of the pulses

Turbo ROM/Header pilot tone duration

Number of pilot tone pulses of the header block (1024-8192)

Turbo ROM/Data block pilot tone duration

Number of pilot tone pulses of the data block (128-2048)

Turbo ROM/Binary load pilot tone duration

Number of pilot tone pulses of blocks when binary files are converted

Turbo ROM/Silence after header

Number of seconds of the silence inserted after the header block

Turbo ROM/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

Turbo ROM/Silence after binary loader

Number of seconds of the silence inserted after binary loader

Turbo ROM/Convert binary loader to a standard tape boot file

If enabled, the binary loader is converted to a standard tape boot file instead of a Turbo ROM compatible binary file.

Turbo ROM/Prepend Turbo ROM loader

Determines if a Turbo ROM natural format loader will be prepended as a standard tape boot file. The values are the following: "None" – no loader will be prepended, "Standard" – a plain loader will be prepended, "Diagnostic" – a verbose loader that displays hexadecimal dump of the header block will be prepended.

Turbo ROM/LOAD address of a BASIC program

Base address to which BASIC programs will be loaded

Turbo ROM/RUN address of a BASIC program

Address of a routine that executes BASIC programs (default is 41086)

Turbo ROM/Pulse Corrections

Corrections of the pulses

Turbo ROM/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

18 Atari Super Turbo

18.1 Overview

The plugin converts files to the Polish **Atari Super Turbo** and compatible systems. The following conversions are supported:

- Conversion to the **Atari Super Turbo AST** format that can store binary files that have up to 44 segments and no INIT segments.
- Conversion to the **Atari Super Turbo BUT** format that can store binary files that have up to 254 segments of any type. A generic BUT loader is used to load such files. The loader occupies addresses 1926 – 2185.

- Conversion to the **Unerring Master Atari Turbo System**. This turbo system can store binary files.
- Conversion to the **Atari Turbo Tape (ATT) System**. This turbo system can store binary files with up to 53 segments.

The screenshot shows a 'Project Item' dialog box with the following fields and controls:

- Plugin:** Atari Super Turbo (selected from a dropdown menu)
- Input file:** C:\utils\at8\stuff\Train 1.0 (v3).xex (with a 'Browse...' button)
- Output file:** C:\utils\at8\stuff\Train 1.0 (v3).wav (with a 'Browse...' button)
- Conversion type:** Unerring Master (selected from a dropdown menu, with a 'Check loader' button)
- File name:** TRAIN 1.0 (V3) .XEX (with an 'Auto set header' button)
- Binary file:**
 - Silence list:** 1,1,1,1,1 (with 'S', 'P', '1.0', and a diamond icon buttons)
- Transfer speed:** 3500 (selected from a dropdown menu)
- Buttons:** OK, Cancel, Clear

Figure 26: Atari Super Turbo

18.2 User Interface

The user interface is depicted in figure 26.

Use the *File name* text field to specify a file name that will be displayed by the loader on screen. Click the *Auto set header* button to automatically set file name and also to verify whether the input file is compatible with the selected conversion (AST, BUT, Unerring Master or Atari Turbo Tape). Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

18.3 Configuration Entries

Atari Super Turbo/Invert polarity of the pulses

Invert polarity of the pulses

Atari Super Turbo/AST header pilot tone duration

Number of pilot tone pulses of the AST header block (256-16384)

Atari Super Turbo/Data block pilot tone duration

Number of pilot tone pulses of data blocks (256-16384)

Atari Super Turbo/Silence after AST header

Number of seconds of the silence inserted after the header block (0-30)

Atari Super Turbo/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

Atari Super Turbo/BUT loader will wait for any key

Determines whether the BUT loader will require a key press to proceed

Atari Super Turbo/Silence after BUT loader

Duration of silence generated after the BUT loader

Atari Super Turbo/Pulse Corrections

Corrections of pulses

Atari Super Turbo/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

19 Hard Turbo

19.1 Overview

The plugin converts files to the **Hard Turbo** Polish turbo system. Given the nature of this turbo system, the input files must be binary files.

T Project Item

Plugin: Hard Turbo

Input file: C:\utils\8\stuff\Train 1.0 (v3).xex Browse...

Output file: C:\utils\8\stuff\Train 1.0 (v3).wav Browse...

Header

File name: TRAIN 1.0 (V3) .XEX

Auto set header

Binary file

Binary loader: HTBL+ Check loader

Silence list: 1,1,1,1 S P 1.0

Transfer speed

2270

OK Cancel Clear

Figure 27: Hard Turbo

19.2 User Interface

The user interface is depicted in figure 27.

Use the *File name* text field to specify a file name that will be displayed on screen. Use the *Auto set header* to automatically set file name. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

19.3 HTBL+ Loader

You can add HTBL+ loader before the binary file converted.

HTBL+ is an extended binary loader that has the following enhancements

- Data recorder motor is switched off after segment data is loaded. This enables loading of binary files that execute a lot of code from INIT segments
- RUNAD is set in a way that allows RUN segment to be the first segment of the binary file
- The loader never waits for any key under normal circumstances. This makes the loader easier to use with devices that do not support MOTOR CTRL signal

- The loader forces warm start during its initialization. This allows to load and run programs that rely on the display configuration set up by the warm start routines
- BASIC is automatically disabled

To add the HTBL+ loader, select HTBL+ from the *Binary loader* check box. Click the *Check loader* button to determine if the HTBL+ binary loader can load the selected input binary file.

19.4 Configuration Entries

Hard Turbo/Invert polarity of the pulses

Invert polarity of the pulses

Hard Turbo/Header block pilot tone duration

Number of pilot tone pulses of the header block (256-16384)

Hard Turbo/Data block pilot tone duration

Number of pilot tone pulses of data blocks (256-16384)

Hard Turbo/Silence after header

Number of seconds of the silence inserted after the header block (0-30)

Hard Turbo/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

Hard Turbo/Name the loader same as file

The binary loader will have name same as the converted file

Hard Turbo/Convert the HTBL+ loader to a standard tape boot file

If enabled, the HTBL+ binary loader will be converted to a standard tape boot file. This helps when you do not have a cartridge with a loader.

Hard Turbo/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

20 Lower Silesian Turbo 2000

20.1 Overview

The plugin converts binary files to the **Lower Silesian Turbo 2000** turbo system from Poland.

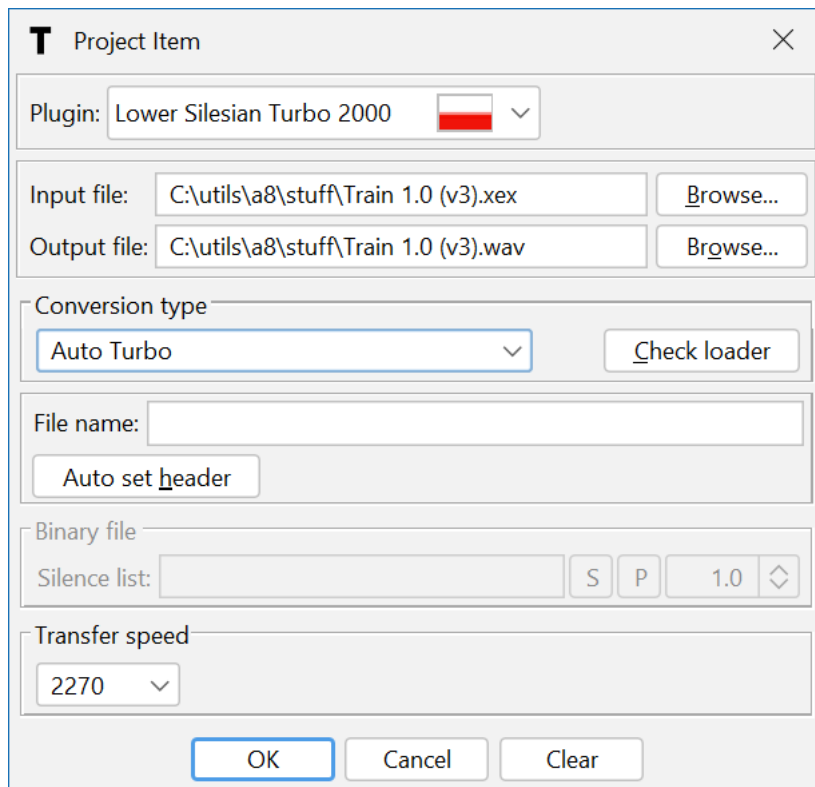


Figure 28: Lower Silesian Turbo 2000

20.2 Conversion Types

Three conversion types are supported.

Auto Turbo. Conversion of an input binary file to the Auto Turbo format (see section 29.6.2). The Auto Turbo format can store any binary file, but only binary files that have no INIT segments can be loaded and executed.

UE Protected Binary. Conversion of an input binary file to the Protected Binary Format (see section 29.6.3) use by the “Unknown Exterminator” copier. A binary loader is converted to the Auto Turbo format. Then the input binary file is converted to the Protected Binary Format.

ExpressLoading of a segmented binary file. Conversion of a segmented binary file to a *chain of Lower Silesian Turbo 2000 composite blocks*. Each composite block can contain data from multiple segments and ends with either an INIT segment or end of file. A special binary loader (ExpressLoader) is prepended before the composite block chain. Click the *Check loader* button to determine if the binary loader can load the selected input binary file. ExpressLoading is the fastest conversion type for segmented binary files, because it utilizes the minimum number of pilot tones possible.

20.3 User Interface

The user interface is depicted in figure 28.

Use the *File name* text field to specify a file name that will be displayed on screen. Click the *Auto set header* button to automatically set the file name. Select conversion type using the *Conversion type* box. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

Click the *Check loader* button to determine if the binary loader used for a given conversion type can load the selected input binary file.

20.4 Configuration Entries

Lower Silesian Turbo 2000/Header block pilot tone duration

Number of pilot tone pulses of the header block (256-8192)

Lower Silesian Turbo 2000/Data block pilot tone duration

Number of pilot tone pulses of the data block (256-8192)

Lower Silesian Turbo 2000/Segment header pilot tone duration

Number of pilot tone pulses of the segment header block (128-768)

Lower Silesian Turbo 2000/Silence after header

Number of seconds of the silence inserted after the header block (0-30)

Lower Silesian Turbo 2000/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

Lower Silesian Turbo 2000/Silence after loader

Duration of silence generated after binary loader

Lower Silesian Turbo 2000/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

21 Rambit Turbo Tape

21.1 Overview

This plugin converts files to the **Rambit turbo system** from the United Kingdom. Given the nature of the Rambit turbo system, only binary files can be converted.

21.2 Conversion Types

Two conversion types are supported.

Monolithic binary file to Rambit. Conversion of monolithic binary file to the natural format for monolithic binary files defined by the Rambit turbo system. A monolithic binary file loader is prepended as a standard tape boot file.

Segmented binary file to Rambit. Conversion of segmented binary file to the natural format for segmented binary files defined by the Rambit turbo system. A segmented binary file loader is prepended as a standard tape boot file.

21.3 User Interface

The user interface is depicted in figure 29.

Use the *Conversion type* box to select conversion type. Use the *File name* text field to specify a file name that will be displayed on screen. Click the *Auto set header* button to automatically set the file name. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

Click the *Check loader* button to determine if the selected binary loader can load the input binary file. Use the *Do not prepend binary loader* box not to prepend the binary loader (use this when using your own binary loader).

21.3.1 Setting Look and Feel of the Loaders

Click the *Color 1* and *Color 2* buttons to set the look and feel of the loaders.

- Original loaders do not allow changing look and feel.
- Modern loaders display horizontal bars of two colors selected with both buttons
- Modern loaders 2 display shades of one basic color selected with the *Color 1* button.

Figure 29: Rabbit

21.4 Loaders

This plugin provides three sets of loaders.

Original Loaders. The original loaders were shipped with the Rabbit turbo system. These loaders display characters on screen to indicate progress of the loading.

Modern Loaders. Modern re-write of the original loaders with improvements and bug fixes. The modern loaders display the name of the loaded program, have adjustable look and feel, and load binary files with INIT vectors properly. These loaders display thick horizontal bars of two colors to indicate progress of the loading.

Modern Loaders 2. Other version of the modern loaders. The capabilities are the same, but the look and feel is different. These loaders display thin horizontal bars to indicate progress of the loading. Shades of a single color are displayed.

It is strongly recommended to use the modern loaders, especially for conversion of segmented binary files. The original loaders handle binary files with INIT segments poorly and also can cause screen display corruption.

21.5 Polarity of the Pulses

The Rabbit turbo system is sensitive to the polarity of the pulses. With incorrect polarity, the files will not load. When the files do not load for no obvious reason, use the *Invert polarity of the pulses* configuration entry to invert the polarity.

21.6 Configuration Entries

Rambit/Invert polarity of the pulses

Invert polarity of the pulses.

Rambit/Loaders

Determines what set of loaders will be used. Modern (default), Modern 2, or Original.

Rambit/Faster Boot

If enabled, the loaders are converted with transfer speed of 720 bd instead of 600 bd.

Rambit/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

22 Turbo 6000

22.1 Overview

This plugin converts files to the **Turbo 6000 system** from the German Democratic Republic.

22.2 Input Files

22.2.1 Compatible Binary Files

Compatible binary file can be converted to the *natural file format* of the Turbo 6000 system. Binary file compatible with Turbo 6000 must meet the following criteria:

- File size is up to 46592 bytes
- The segments of the binary file must be ordered by their first addresses
- File has exactly one RUN segment
- File has at most one INIT segment

22.2.2 Binary Files

Any binary file can be converted to a special file format (Turbo 6000 - ChainLoading) that was designed to allow for loading of binary files with minimum restrictions.

The binary file is converted as follows:

- A special binary loader (Turbo 6000 - ChainLoader) is converted to the natural file format
- Binary file is converted to the special file format

22.3 User Interface

The user interface is depicted in figure 30.

Use the *Input file type* box to select input file type. Enter a file name that will be displayed by the loaders in the *File name* box. Click the *Auto set header* button to automatically set the file name. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

22.4 Polarity of the Pulses

The Turbo 6000 system is sensitive to the polarity of the pulses. With incorrect polarity, the files will not load. When the files do not load for no obvious reason, use the *Invert polarity of the pulses* configuration entry to invert the polarity.

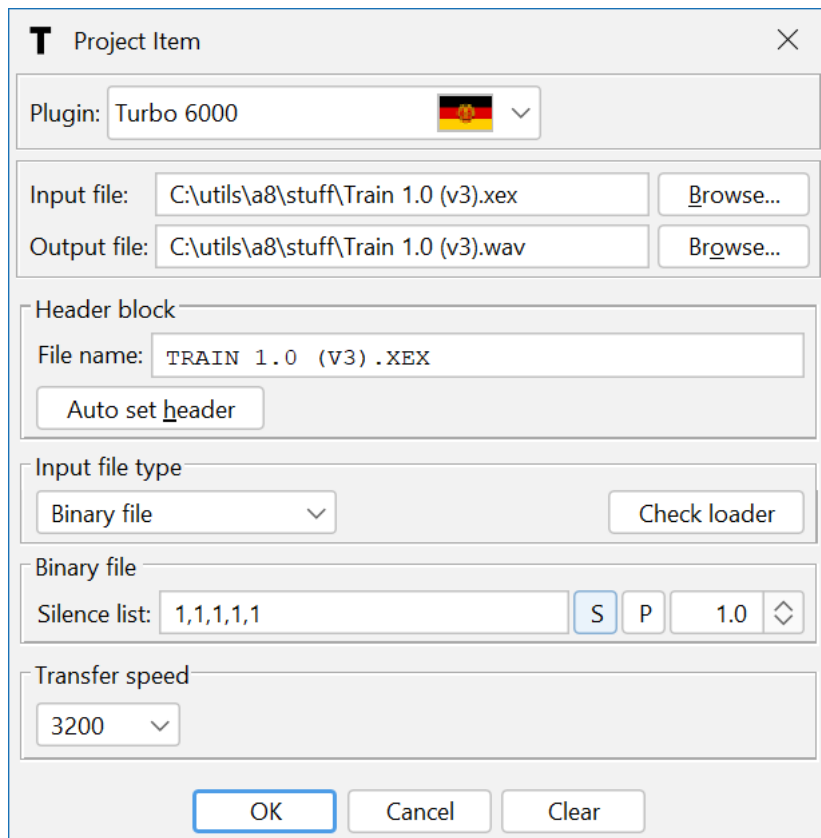


Figure 30: Turbo 6000

22.5 Sampling Rate

The Turbo 6000 plugin takes advantage of the sampling rate of 48000 Hz.

22.6 Configuration Entries

Turbo 6000/Invert polarity of the pulses

Invert polarity of the pulses.

Turbo 6000/Silence after header

Number of seconds of the silence inserted after the header block (0-30)

Turbo 6000/Pause after header

If enabled, output of electric signal to sound card will be paused after the header block

Turbo 6000/Pulse Corrections

Corrections of pulses

Turbo 6000/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

23 Standard

23.1 Overview

The plugin converts input files to the standard (FSK) tape records; therefore, you can use it with a unmodified data recorder. The transfer speed is limited to 600-800 bd.

You can perform the following conversions:

- Monolithic binary files to tape boot files
- Binary files to the standard tape records

- Binary files to the LB-Express (LBE) high-performance file format
- Tokenized BASIC files and BASIC source code files
- Plain data files

23.2 User Interface

The user interface is depicted in figure 31.

Figure 31: Standard

Use the *Conversion type* combo box to select the desired conversion. Use the controls in the *Options* boxes to set additional options for the conversion. Click the *Check loader* button to determine if the input binary file can be loaded with selected options. Use the *Pulses* combo box to select transfer speed.

23.3 Conversion of Monolithic Binary Files to Boot Files

The monolithic binary files are converted to standard tape boot files. The following limitations apply to monolithic binary files:

- The DATA segment cannot occupy memory below address 1536
- Length of the DATA segment must not exceed 32649 bytes

If your monolithic binary file is not within the limitations, convert the file as a (general) binary file.

23.4 Conversion of Binary Files to the Standard Tape Records

23.4.1 Processing

The binary files are converted as follows:

- A binary loader is converted to a tape boot file.
- The binary file is converted to the standard tape records with short IRGs.

23.4.2 Binary Loaders

TSCBL Binary Loader. TSCBL is a general-purpose and customizable binary loader. The look and feel of the loader can be customized. You can set program title, text luminance, background color, SIO beeping, cursor visibility and ATRACT mode suppression.

X Loader. A simple and tiny binary loader contributed by Krzysztof Dudek (xxl).

23.4.3 User Interface

Use the *Binary loader* box to select the desired binary loader.

Use the *Title* text field to set a title that will be displayed by the binary loader. Alternatively, you can enter space-separated hexadecimal ATASCII codes preceded by the \$HEX\$ keyword, e.g., \$HEX\$ 32 FC 24 45 75 to display characters not available on the keyboard, or inverse characters. Click the *Auto set* button to set the title automatically. The binary loader is displaying the characters through the E: system device. The control characters are interpreted, so you can use them to position the title.

Click the *Background color* button to set the background color of the loader. Use the *Luminance* box to select the luminance of the title text. Select the *Hide cursor* box to hide cursor. Select the *Silent I/O* box to disable the SIO beeping (useful when creating dual track tapes). Use the *Silence list* panel to create a silence list. Refer to section 4.8. for more details.

Select the *Suppress ATRACT* box to suppress the ATRACT mode during loading. Note that if the input binary file contains segments larger than 24 KB, these segments are automatically split to smaller ones during the conversion. This occurs in-memory, the original binary file is not touched.

Click the *Check loader* button to determine if the binary loader can load the selected input binary file.

23.5 Conversion of Binary Files to the LBE Format

The LB-Express (LBE) is a special file format that allows you to load binary load files efficiently. The LBE format uses long, 512-byte blocks. Usage of the long blocks significantly reduces the loading time. LBE is highly recommended for publishing tape versions of Atari software.

23.5.1 Processing

The binary files are converted as follows:

- One of the LBE binary loaders is converted to a tape boot file.
- The input binary file is converted to the special LBE format.

23.5.2 Binary Loaders

LBE Plain Loader. The Plain loader loads binary files stored in the LBE format. The look and feel of the loader can be customized. You can set program title, text luminance, background color, SIO beeping and cursor visibility.

LBE PMG Loader. The PMG loader loads binary files stored in the LBE format. The loader displays loading progress using three PMG missiles M0, M1 and M2 in non-DMA mode. The look and feel of this loader can be customized as the Plain loader.

Note that some binary load files can use PMG objects during the loading process, creating a conflict with the loader or vice versa. When the input binary load file is larger than 14 MB, no progress indicator will be displayed.

23.5.3 User Interface

Use the *Binary loader* box to select the desired binary loader.

Use the *Title* text field to set a title that will be displayed by the binary loader. Click the *Auto set* button to set the title automatically. The binary loader skeleton is displaying the characters through the E: system device. The control characters are interpreted, so you can use them to position the title.

Click the *Background color* button to set the background color of the loader. Use the *Luminance* box to select the luminance of the title text. Select the *Hide cursor* box to hide cursor. Select the *Silent I/O* box to disable the SIO beeping (useful when creating dual track tapes). Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

Select the *Suppress ATTRACT* box to suppress the ATTRACT mode during loading.

Click the *Check loader* button to determine if the selected binary loader can load the selected input binary file.

23.6 Conversion of Plain Data files

Plain data files are converted as they are. You can select duration of the IRGs.

Use controls in the *Plain data options* panel to select either short IRGs or long IRGs. Select the *“Use data in EOF block”* trick box to place data to EOF block. Use this special option only when converting tape boot files that require it.

23.7 Conversion of Tokenized BASIC Files

23.7.1 Formats

You can convert tokenized BASIC files to the following formats:

- Format loadable by the CLOAD command. Short IRGs are used.
- Format loadable by the LOAD"C:" or RUN"C:" commands. Long IRGs are used.
- Bootable format. First, a BASIC initializer (LAUNCHBAS) is converted to a boot file. Then the tokenized BASIC file is converted to the format loadable by the CLOAD command. The BASIC initializer forces a RUN"C:" command and also tweaks the OPEN routine to use short IRGs once. This allows you to load and run protected programs. The BASIC initializer displays the B letter on screen during the loading process.

23.7.2 User Interface

Use controls in the *Tokenized BASIC options* box to select the desired file format.

23.8 Conversion of BASIC Source Files

The files are converted to a format loadable by the ENTER"C:" command.

23.9 Sampling Rate

The Standard plugin takes advantage of the sampling rates of 48000 and 96000 Hz.

23.10 Implicit Silence List

The Standard plugin always automatically ensures that IRGs after blocks that hold INIT segments have a minimum duration of 2.5 seconds.

23.11 Configuration entries

Standard/Leader duration

Duration of the leader signal in seconds (14-30).

Standard/Cumulate silence for blocks with INIT vectors

Indicates whether to cumulate silence. Refer to section 4.8.5.

Standard/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

24 Standard Plus

The Standard Plus plugin converts binary files to special systems that are built on the top of the standard tape records (FSK), usable with unmodified data recorders. The supported systems typically offer longer, non-standard block sizes, the ability to re-read a failed block, increased transfer speeds and loading progress indication.

The supported conversion types are the following:

- **NHP 3.6 (Rainbow).** A system from Chile that offers: 1. Increased transfer speed, 2. Increased block size (257 bytes), 3. Ability to re-read a failed block, 4. Loading progress indication by a block countdown. The system supports binary files with INIT segments, but there are certain limitations - e.g.: the loader's screen is renewed after each INIT segment and the loader uses the 'RAM under ROM'.
The 'NHP' stands for '*No Hay Problema*' ('No worries').
- **SITRE.** A system from Chile that offers: 1. Increased transfer speed (760 bd), 2. Increased block size (261 bytes), 3. Ability to re-read a failed block, 4. Loading progress indication by a block countdown. The system supports binary files with INIT segments.
The 'SITRE' stands for '*Sistema Inteligente Turbo con Recuperación de Errores*' ('An Intelligent Turbo System with Error Recovery'). Home page: <http://www.vitoco.cl/atari/sitre/index.html>.

24.1 User Interface

The user interface is divided in two sections.

The first section is common for all conversion types. Use the *Conversion type* combo box to select the desired conversion type. Click the *Check loader* button to verify if the input binary file is compatible with the selected loader.

The second section is specific for each conversion type and is described below.

24.2 NHP 3.6 (Rainbow)

This conversion type converts the input binary file to the NHP 3.6 (Rainbow) system described above.

Specify the program title displayed by the loader in the *Title* text field. Specify the transfer speed using the *Transfer speed* box. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

The 'Project Item' dialog box is shown with the following settings:

- Plugin:** Standard Plus (with a flag icon)
- Input file:** C:\utils\8\stuff\Train 1.0 (v3).xex (with a 'Browse...' button)
- Output file:** C:\utils\8\stuff\Train 1.0 (v3).wav (with a 'Browse...' button)
- Conversion type:** NHP 3.6 (Rainbow) (with a 'Check loader' button)
- NHP Options:**
 - Title:** Train 1.0 (v3) (with an 'Auto set' button)
 - Silence list:** 1,1,1,1,1 (with 'S', 'P', and '1.0' buttons)
 - Transfer speed:** 640 (with a spinner)
- Buttons:** OK, Cancel, Clear

Figure 32: Standard Plus - NHP 3.6 (Rainbow)

24.3 SITRE

This conversion type converts the input binary file to the SITRE system described above.

Specify the program title displayed by the loader in the *Title* text field. Specify the transfer speed using the *Transfer speed* box. Use the *Silence list* panel to create a silence list. Refer to section 4.8 for more details.

The 'Project Item' dialog box is shown with the following settings:

- Plugin:** Standard Plus (with a flag icon)
- Input file:** C:\utils\8\trbarchive\1_games_classic\airstrikeii.xex (with a 'Browse...' button)
- Output file:** C:\utils\8\trbarchive\1_games_classic\airstrikeii.wav (with a 'Browse...' button)
- Conversion type:** SITRE (with a 'Check loader' button)
- SITRE Options:**
 - Title:** AIRSTRIKEII (with an 'Auto set' button)
 - Silence list:** (empty) (with 'S', 'P', and '1.0' buttons)
 - Transfer speed:** 760 (with a spinner)
- Buttons:** OK, Cancel, Clear

Figure 33: Standard Plus - SITRE

24.4 Implicit Silence List

The Standard Plus plugin always automatically ensures that IRGs after blocks that hold INIT segments have a minimum duration of 2.5 seconds.

24.5 Sampling Rate

The Standard Plus plugin takes advantage of the sampling rates of 48000 and 96000 Hz.

24.6 Configuration Entries

Standard Plus/Leader duration

Duration of the leader signal in seconds (14-30). This typically applies only to the first tape boot file.

Standard Plus/Cumulate silence for blocks with INIT vectors

Indicates whether to cumulate silence. Refer to section 4.8.5.

Standard Plus/Allow NHP 3.6 (Rainbow) in the Wizard

Indicates whether the system is available in the Wizard for files

Standard Plus/Allow SITRE in the Wizard

Indicates whether the system is available in the Wizard for files

Standard Plus/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

25 Tape Image

The plugin reads and interprets tape images (refer to section 2.2). The user interface displays a list of supported tape image chunks.

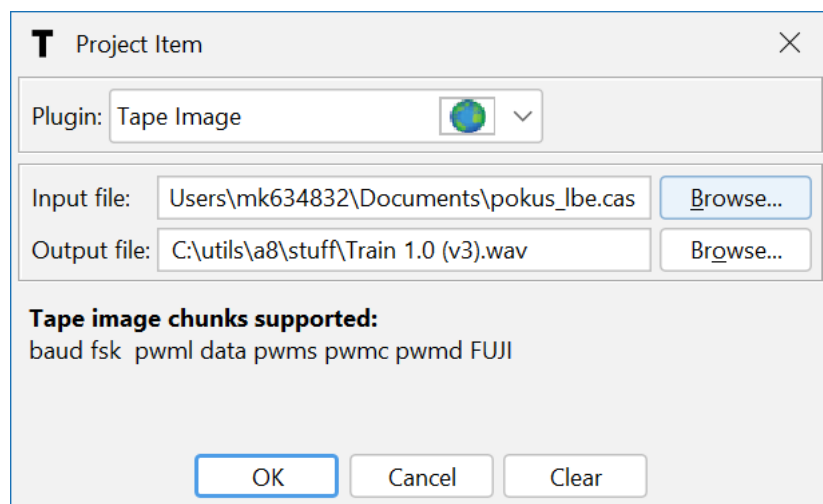


Figure 34: Tape Image

26 Omicron Turbo

Omicron Turbo is a special turbo system that is not associated with a particular turbo upgrade. Its design allows using Omicron Turbo with various existing turbo upgrades.

Omicron Turbo provides a set of unifying file formats and binary loaders that will work with the following turbo upgrades (this list might not be exhaustive):

- Turbo 2000 (CZ/SK), Super Turbo (CZ/SK)
- KSO Turbo 2000 and compatibles (input from joystick port or SIO port)

- Turbo Blizzard
- Atari Super Turbo, Hard Turbo, Lower Silesian Turbo 2000, Unerring Master

Omicron Turbo was designed for creating merchantable gaming software distributions on tape. With Omicron Turbo, you can distribute software on tapes with high transfer speed and great compatibility with various turbo upgrades. The only disadvantage is a tiny standard tape boot file prepended before each program.

26.1 Conversion Types

The Omicron Turbo plugin can convert binary files only.

26.1.1 BlockLoading

The input file is converted to blocks of varying lengths. A block is generated for each segment of the input binary file. The maximum number of segments is 62. A binary loader is prepended before the blocks.

26.1.2 Kilobyte Blocks

The input file is converted to 1024-byte blocks. A binary loader is prepended before the blocks.

26.1.3 Twokilobyte Blocks

The input file is converted to 2048-byte blocks. A binary loader is prepended before the blocks.

26.1.4 ExpressLoading

Conversion of a segmented binary file to a *chain of composite blocks*. Each composite block can contain data from multiple segments and ends with either an INIT segment or end of file. A binary loader is prepended before the composite block chain.

ExpressLoading is the fastest conversion type for segmented binary files because it utilizes the minimum number of pilot tones possible. The maximum transfer speed for ExpressLoading is 2270 bd.

26.2 Processing

The input file is converted as follows

1. Stage 0 loader is converted to a standard tape boot file (16-second leader tone followed by 4 blocks, with data in the EOF block).
2. A binary loader (Stage 1) is converted to a single Omicron Turbo block that is read by the Stage 0 loader. This binary loader can also be loaded directly by cartridge or file version of Omicron Turbo Stage 0 loader.
3. The input binary file is converted to Omicron Turbo

26.3 User Interface

The user interface is depicted in figure 35 .

Click the *Check loader* button to determine if the binary loader can load the selected input binary file. Use the *Silence list* panel to create a silence list. For more details, refer to section 4.8.

Use the *Conversion type* combo box to select conversion type. Click the *Auto Set Header* button to automatically set the program name displayed by the stage 0 loader.

T Project Item

Plugin: Omicron Turbo

Input file: C:\utils\88\stuff\Train 1.0 (v3).xex Browse...

Output file: C:\utils\88\stuff\Train 1.0 (v3).wav Browse...

Conversion type: ExpressLoading Check loader

Header

File name: TRAIN 1.0 (V3) .XEX

Auto set header

Binary file

Silence list: 1,1,1,1,1 S P 1.0

Transfer speed: 2270

OK Cancel Clear

Figure 35: Omicron Turbo

26.4 Configuration Entries

Omicron Turbo/Invert polarity of the pulses

Invert polarity of the pulses

Omicron Turbo/BlockLoading pilot tone duration

Number of pilot tone pulses of data blocks for the BlockLoading conversion type (256-8192)

Omicron Turbo/Pilot tone duration for Kilobyte Blocks

Number of pilot tone pulses of data blocks for the Kilobyte Blocks and Twokilobyte blocks conversion type (256-8192)

Omicron Turbo/Prolongate pilot tone

Increase number of pilot tone pulses with increasing approximate baud rate

Omicron Turbo/Generate loader as a standard tape boot file

When enabled, the stage 0 loader is prepended as a standard tape boot file (default). When disabled, no stage 0 loader is generated (useful when cartridge or file version of Omicron Turbo will be used).

Omicron Turbo/Faster Boot

When enabled, the stage 0 loader is generated with transfer speed of 720 bd and inter-record-gaps (IRGs) that are only 180 ms long.

Omicron Turbo/File name creation options

Configure the automatic file name creation. Refer to section 4.9.

Omicron Turbo/Enable this plugin in the wizard for files

Makes the plugin visible to the wizard for files. Refer to section 4.4.

Part III

Turbo Systems

27 Introduction

This part contains information about various turbo systems that have been created in Czechoslovakia, Poland and other countries. Descriptions of the systems are sufficient to provide ability to write turbo generators, loaders, copiers and turbo decoders.

Note: Some of the information provided is not based on official documentation, but on observations and reverse engineering; thus it might be inaccurate. If you have any corrections to suggest, please visit <https://turgen.sourceforge.io/support.html>.

Information is encoded using pulse width modulation (PWM). In this document, width of a pulse is defined as a distance between two transitions from logical zero to logical one.

28 Turbo Systems from the Czech Republic and Slovakia

28.1 Turbo 2000

28.1.1 Overview

Turbo 2000 was the first turbo system available in former Czechoslovakia. It was developed in 1987 by Jiří Richter, student of Czech Technical University in Prague and member of Prague Atari user club.

In early times, Turbo 2000 loaders were distributed on tapes, stored using the standard 600 baud system. Then loaders on cartridges became popular.

Pulse types	Narrow, wide, pilot tone, sync
File format	Header block (HEADER) followed by data block (DATA)
Block format	Pilot tone (at least 256 pulses), sync pulse, data bytes. At least 2000 pilot tone pulses is recommended for compatibility with Universal turbo loaders
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

28.1.2 Header block

Offset	Description
0	Always 0
1	File type (1 - plain data, 3 - program, 4 - binary file, 255, 254 - tokenized BASIC)
2-11	File name
12-13	Load address
14-15	Length of file
16-17	Start address
18	Check sum = (HEADER[0]) xor ... xor (HEADER[17])

28.1.3 Data block

Offset	Description
0	Always 255
1-?	Data itself
Last	Check sum = (DATA[0]) xor ... xor (DATA[?])

28.1.4 Timing

Standard transfer speed is approximately 2270 bauds. There is a big tolerance for width of all pulses.

Pulse	Center width	Tolerated range
Pilot tone	32/44100 s	(25-47)/44100 s
Wide	26/44100 s	(20-40)/44100 s
Narrow	13/44100 s	(6-19)/44100 s
Sync	10/44100 s	(4-17)/44100 s

28.2 Super Turbo

28.2.1 Description

Super turbo system is an enhancement of Turbo 2000 system, developed by Jiří Richter. Supported transfer speeds are approximately from 2725 to 6411 bauds. The Super Turbo files are loaded by “Universal Turbo” loaders capable of loading both Turbo 2000 and Super Turbo files. To determine the transfer speed and file format, the loader measures the duration of three pilot tone pulses. Some loaders are capable of displaying a progress bar using PMG objects in non-DMA mode.

Pulse types	Narrow, wide
File format	Header block (HEADER) followed by data block (DATA)
Block format	Pilot tone (at least 1200 wide pulses), narrow pulse, data bytes. At least 2000 narrow pulses is recommended for compatibility with Universal turbo loaders
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

28.2.2 Header block

Offset	Description
0	Always 183
1	File type (1 - plain data, 3 - program, 4 - binary file, 255 - tokenized basic)
2-21	File name
22-23	Load address
24-25	Length of file
26-27	Start address
28	Check sum = (HEADER[0]) xor ... xor (HEADER[27])

28.2.3 Data block

Offset	Description
0	Always 237
1-?	Data itself
Last	Check sum = (DATA[0]) xor ... xor (DATA[?])

28.2.4 Timing

Various transfer speeds are supported. For a given speed, the width of the wide pulse is the width of the narrow pulse, simply doubled.

Pulse	Width
Wide	(6/44100 - 22/44100) s
Narrow	(3/44100 - 11/44100) s

28.3 Turbo 2000 - Kilobyte Blocks

28.3.1 Description

Turbo system based on the Turbo 2000, part of various versions of “Turbo operating system” (TOS). The idea was to expose turbo through a custom CIO device (typically T: or D:) for application programs or BASIC.

Pulse types	Narrow, wide, pilot tone, sync
File format	Header block followed by one or more data blocks
Block format	Pilot tone (at least 256 pulses), sync pulse, data bytes.
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

28.3.2 Header block

Offset	Description
0-1	Always 0
2-17	File name
18	Check sum = HEADER[0] xor HEADER[1] xor ... xor HEADER[17]

28.3.3 Data blocks

Offset	Description
0	255=Full block, 250=EOF block. Numbers 251-254 indicate partially filled block. If we subtract 251 from this number, we obtain a difference that will be denoted as Z . This difference can be 0,1,2 or 3 and represents higher byte of number of valid data bytes in the block.
1-1024	Data itself. If the block is a full block, there is 1024 bytes of data. If the block is EOF block, there are all zeros. If the block is a partially filled block, there is data padded with zeros up to the length of 1023 bytes. Last byte is lower byte of number of valid data bytes in the block. If we denote this lower byte as X , the number of valid bytes in the block is $Z * 256 + X$
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1024]

28.3.4 Timing

Standard transfer speed is approximately 2270 bauds. There is a big tolerance for width of pulses.

Pulse	Center width	Tolerated range
Pilot tone	32/44100 s	(25-47)/44100 s
Wide	26/44100 s	(20-40)/44100 s
Narrow	13/44100 s	(6-19)/44100 s
Sync	10/44100 s	(4-17)/44100 s

28.4 Turbo Tape

28.4.1 Description

Advanced turbo system introduced with the TT-DOS operating system sold by the JRC company. Aim of this turbo system was to almost fully replace a disk drive. The usual CIO device installed was B: or D:.

Pulse types	Narrow, wide
File format	One or more data blocks (all 1026 bytes long)
Block format	Pilot tone (at least 1200 pulses), narrow pulse, data bytes.
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

28.4.2 Tape modes

Mode	Description
SS	Short gaps between blocks. First block is written twice, others once.
SD	Short gaps between blocks. All blocks are written twice.
LS	Long gaps between blocks. First block is written twice, others once.
LD	Long gaps between blocks. All blocks are written twice.

The D modes provide data redundancy for safe data storage. The first block is always written twice to provide a convenient support for the READ DIRECTORY CIO function.

28.4.3 Structure of the blocks

Offset	Description
0	Sequential number of the block. Numbering starts from 1. In case of D modes, pairs of blocks have same sequential number.
1	Tape mode: SS=128, LS=0, SD=192, LD=64
2,3	Bits 0-11: Offset of last valid byte in the block (16-1024), this offset will be denoted as B . Bit 15: Last block flag. Byte at offset 2: $B \% 256$ Byte at offset 3: $[B / 256] + [128 * (EOF \text{ is true})]$
4	Undefined number. All blocks of one file should have same number here.
5	Undefined.
6-16	File name and extension padded with spaces. First 8 bytes are devoted to file name, last 3 bytes are devoted to extension.
17-1024	Data itself (1008 bytes). Data must be padded with any bytes.
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1025]

28.4.4 Timing

Timing is compatible with Super Turbo.

28.5 B-TAPE

28.5.1 Description

Advanced turbo system introduced with B-TAPE extension for operating system BW-DOS. The aim of this extension was to almost fully replace a disk drive. B-TAPE allows using both CIO and SIO to exploit data recorder turbo modification. The disadvantage is a big size of the device handler. B-TAPE was designed as an improvement of the Turbo Tape system. The B-TAPE device handler can read files stored using Turbo Tape system. To circumvent problems with the big size of the device handler, a special small binary loader called MICROB was shipped with B-TAPE.

Pulse types	Narrow, wide
File format	One or more data blocks (all 1026 bytes long)
Block format	Pilot tone (at least 1200 pulses), narrow pulse, data bytes.
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

28.5.2 Tape modes

Mode	Description
SS	Short gaps between blocks. First block is written twice, others once.
SD	Short gaps between blocks. All blocks are written twice.
LS	Long gaps between blocks. First block is written twice, others once.
LD	Long gaps between blocks. All blocks are written twice.

D modes provide data redundancy for safe data storage. The first block is always written twice to provide convenient support for the READ DIRECTORY CIO function.

28.5.3 Structure of the blocks

Offset	Description
0	Sequential number of the block. Numbering starts from 1. In case of D modes, pairs of blocks have same sequential number.
1	Tape mode: SS=128, LS=0, SD=192, LD=64
2,3	Bits 0-11: Offset of last valid byte in the block (16-1024), this offset will be denoted as B . Bit 15: Last block flag. Byte at offset 2: $B\%256$ Byte at offset 3: $[B/256] + [128 * (EOF\ is\ true)]$
4	Undefined number. All blocks of one file should have same number here.
5	Random number. All blocks of one file must have same number here. This random number allows to distinguish files with same file name.
6-16	File name and extension padded with spaces. First 8 bytes are devoted to file name, last 3 bytes are devoted for extension.
17-1024	Data itself (1008 bytes). Data must be padded with zeros if needed.
1025	Check sum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[1024]

28.5.4 Timing

Timing is compatible with Turbo 2000 and Super Turbo.

29 Turbo systems from Poland

29.1 KSO Turbo 2000 and Turbo 2000F

29.1.1 Description

Turbo systems used in Poland, originally designed together with the KSO Turbo 2000 tape operating system and then adopted for other tape operating systems.

Pulse types	Narrow, wide, pilot tone
File format	Header block followed by one or more data blocks
Block format	Pilot tone (at least 256 pulses), data bytes.
Bit order	MSB to LSB
Signal input	DATA IN or 2nd JOYSTICK port
Turbo mode activation	COMMAND active or 2nd JOYSTICK port

29.1.2 Header block

Offset	Description
0	Always 0
1	Always 255
2-11	File name (10 characters)
12	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[11]) mod 256

29.1.3 Data block

Offset	Description
0-1	Number of valid bytes in the block, 0-3072.
2-3073	Up to 3072 bytes of data padded with zeros if needed.
3074	Check sum = (BLOCK[0] + ... + BLOCK[3073]) mod 256

The file ends with a data block that has less than 3072 valid bytes. If the total file size can be divided by 3072 without a remainder, the file must end with a block that has 0 valid bytes.

29.1.4 Timing

Pulse	Width
Pilot tone	44/44100 s
Wide	22/44100 s
Narrow	11/44100 s

29.2 Turbo Blizzard

29.2.1 Description

Turbo system used in Poland, suitable for storing of binary files (small blocks, high transfer speed).

Pulse types	Narrow, wide, pilot tone
File format	Synchronization block (3072 pilot tone pulses, two narrow pulses) Header block (302 pilot tone pulses, two narrow pulses, data) One or more data blocks (302 pilot tone pulses, two narrow pulses, data)
Block format	Pilot tone, two narrow pulses, data
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	DATA OUT set to 0

29.2.2 Header block

Offset	Description
0-75	File name
76	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[75]) mod 256
77	Spare byte, always 0. This byte is never read by the loaders.

29.2.3 Data block

Offset	Description
0-1	Number of valid bytes in the block, 0-1024.
2-1025	Up to 1024 bytes of data padded with zeros if needed.
1026	Always 0
1027	Check sum = (HEADER[0] + HEADER[1] + ... HEADER[1026]) mod 256
1028	Spare byte, always 0. This byte is never read by the loaders.

29.2.4 Timing

Pulse	Width
Pilot tone	22/44100 s 24/48000 s
Wide	12/44100 s 12/48000 s
Narrow	8/44100 s 8/48000 s

29.3 Turbo ROM

29.3.1 Description

Turbo system used in Poland, with limitations similar to Czechoslovak Turbo 2000 and Super Turbo Systems. This turbo system can store Turbo ROM compatible binary files (these binary files consist of exactly one DATA segment at most one RUN segment and at most one INIT segment) or tokenized BASIC programs.

Pulse types	Narrow, wide, stop
File format	Header block followed by one data block
Block format	Pilot tone (wide pulses; 4884 for the header block, 516 for data block), narrow pulse, data, 4 wide pulses, stop pulse
Bit order	LSB to MSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

29.3.2 Header Block for Binary Files

Offset	Description
0	Header block check sum = (HEADER[1] ... xor ... HEADER[40])
1-2	Header load address (1537)
3-4	Header length excluding first byte (40)
5	Data block check sum = (DATA[0] xor DATA[1] xor ... DATA[?])
6-7	RUN address
8-9	INIT address
10-11	Load address
12-13	Data block length
14	Padding 0
15-34	File name. Internal code is used.
35	Program type flag. For binary files there is 1.
36	0 - JSR to the INIT address, 1 - No JSR to the INIT address
37-39	Padding zeros
40	RTS opcode (96)

29.3.3 Data Block for Binary Files

Offset	Description
0-?	Data bytes

29.3.4 Header block for BASIC files

A BASIC file consist of two parts. A header part (14 bytes long) and a main part. Data from the header part is stored in the header block and data from the main part is stored in the data block. The tokenized form of the BASIC file is relocatable and can be loaded to any address. Such address is denoted as base address.

Offset	Description
0	Header block check sum = (HEADER[1] ... xor ... HEADER[77])
1-2	Header load address (1537)
3-4	Header length excluding first byte (77)
5	Data block check sum = (DATA[0] xor DATA[1] xor ... DATA[?])
6-7	RUN address (41086). Points to a routine in the BASIC ROM.
8-9	INIT address. Always 0 for a BASIC program.
10-11	LOAD address. Calculated as base address + VNT.
12-13	Data block length. Length of the main part of the BASIC file.
14	Padding 0
15-34	File name. Internal code is used.
35	Program type flag. Always 0 for a BASIC program.
36	1 - No JSR to the INIT address
37-39	Padding zeros
40-59	Routine that copies the header part to the zero page 0xA2, 0x11, 0xBD, 0x3C, 0x06, 0x95, 0x80, 0xCA, 0x10, 0xF8, 0x60
60	Header part of the BASIC file (LOMEM, VNT, VNTE, VVT, STMTAB, STMCUR, STARP)
74-75	Address right past the loaded BASIC file (LOAD address + length of the main part)
76-77	Address right past the loaded BASIC file (LOAD address + length of the main part)

29.3.5 Data Block for BASIC Files

Offset	Description
0-?	Main part of the BASIC file

29.3.6 Timing

Pulse	Width
Wide	14/44100 s 18/48000 s
Narrow	6/44100 s 8/48000 s
Stop	48/44100 s 52/48000 s

29.3.7 Special Notes

The Turbo ROM loading routines calculate the check sum *after a block is fully loaded*. This can cause false check sum errors for blocks that are loaded to the I/O area (0xD000 - 0xD7FF).

29.4 Atari Super Turbo (AST)

29.4.1 Description

Turbo system used in Poland that can store binary files. Note that the following information is incomplete.

Pulse types	Narrow, wide, stop
File format	Various formats. For details, see below.
Block format	Pilot tone (wide pulses), narrow pulse, data, 4 narrow pulses, 128 stop pulses
Bit order	LSB to MSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

29.4.2 AST File Format

This file format can be used to store binary files that have up to 44 segments and no INIT segments. A file consists of several blocks.

1. Header block, 256 bytes of data
2. Data blocks. For each segment of the binary file, there is one data block.

29.4.3 BUT File Format

This file format can be used to store binary files. A file consists of several blocks.

1. BUT loader header block. 6 bytes of data. It appears to be a boot header.
2. BUT loader data block. 576 bytes of data.
3. Pairs of data blocks for each segment. The first block in a pair is a segment header, the second block in a pair holds segment data.
4. Termination segment header¹

29.4.4 Unerring Master Atari Turbo System

This turbo system can be used to store binary files. A file consists of several blocks.

1. Header block
2. Binary loader block
3. Binary file data blocks. Each data block stores one or more segments of a binary file. A block ends either with a special termination header or with an INIT segment.

29.4.5 Atari Turbo Tape (ATT)

This turbo system can be used to store binary files with up to 53 segments. A file consists of several blocks.

1. Header block. Holds the total number of segments, array of segment headers, identification literals and file name.
2. Binary file data blocks. Each data block stores data of one binary file segment.

¹TURGEN is using slightly different format. There is no termination segment header, but the loader knows the number of segments.

29.4.6 AST Header Block

Offset	Description
0	Number of segments (1-44)
1	Header check sum Check sum = (HEADER[2] xor HEADER[3] +xor... HEADER[255])
2-177	44 segment headers (start and end addresses)
178	Always 102
179	Always 85
180-199	File name. Internal code is used.
200	Always 0
201-244	Check sums for each segment block Check sum = (DATA[1] xor DATA[2] +xor... DATA[?])
245-255	Unknown. 0x70, 0x4b, 0x00, 0xd6, 0x47, 0xb4, 0xcf, 0x4b, 0x00, 0xd6, 0x41.

29.4.7 BUT Loader Header Block

Offset	Description
0	Always 0
1	Length of the BUT loader data block in 128 bytes blocks. Always 5.
2,3	Loader load address. 1920.
4,5	Unknown. 182, 9.

29.4.8 BUT Loader Data Block

Offset	Description
0-575	BUT loader code

29.4.9 AST and BUT Data Block

Offset	Description
0-?	Data

29.4.10 BUT Termination Segment Header

Offset	Description
0-3	255, 255, 255, 255

29.4.11 UM Header Block

Offset	Description						
0-255	Header title and file name. All 255 bytes are displayed on screen beginning from SAVMSC. Internal code is used for the characters displayed. The offset for the file name can vary. Offset specified here matches the offset used by the original UM loader and copier. <table><tr><th>Offset</th><th>Description</th></tr><tr><td>214-240</td><td>File name, maximum 27 characters</td></tr><tr><td>56</td><td>Always 'U'. The loader checks this field</td></tr></table>	Offset	Description	214-240	File name, maximum 27 characters	56	Always 'U'. The loader checks this field
Offset	Description						
214-240	File name, maximum 27 characters						
56	Always 'U'. The loader checks this field						
256,257	Load address of the next block						
258,259	Length of the next block						
260,261	Run address. DOSINI is set to this address.						
262	Checksum of the next block (EOR of all bytes of the block)						
263-269	Always 0						

29.4.12 UM Binary Loader Block

Offset	Description
0-?	Binary loader code and data

29.4.13 UM Binary File Data Block

Offset	Description
[This part of the block can repeat
0,1	Segment header 1 (first address of a segment) Special value 0xDE,0xDE indicates end of file, no segment header 2 and no segment data.
2,3	Segment header 2 (last address of a segment).
4-?	Data of the segment
]	End of part that can repeat
?+1	Checksum (EOR of all previous bytes of the block)

29.4.14 Atari Turbo Tape Header Block

Offset	Description
0	Total number of segments of the binary load file
1-213	An array of 4-byte segment headers. The first two bytes hold first address, the last two bytes hold last address
214	Literal 0xAA
215	Literal 0x55
216	File name. 40 characters, internal code.

29.4.15 Atari Turbo Tape Data Block

Offset	Description
0-?	Segment data

29.4.16 Timing

Pulse	Width
Wide	22/44100 s
Narrow	12/44100 s
Stop	72/44100 s

29.5 Hard Turbo

29.5.1 Description

Turbo system used in Poland that can store only binary files.

Pulse types	Narrow, wide, pilot tone
File format	Main header followed by pairs of segment header and segment data blocks. A special segment header block identifies end of file.
Block format	Pilot tone (at least 512 pilot tone pulses), narrow pulse, data
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

29.5.2 Main Header Block

Offset	Description
0	Identification byte, always 0
1-39	File name. ATASCII string terminated with EOL (155)
40	Check sum = (HEADER[0] xor HEADER[1] xor ... HEADER[39])

29.5.3 Segment Header Block

Offset	Description
0	Identification byte, always 255
1,2	First address of the following segment. Value of 65535 (255 255) indicates end of file
3,4	Last address of the following segment (increased by one)
5	Check sum = (SEGHEAD[0] xor SEGHEAD[1] xor ... SEGHEAD[4])

29.5.4 Segment Data Block

Offset	Description
0	Identification byte, always 255
1-?	Segment data
Last	Check sum = (DATA[0] xor DATA[1] xor ... DATA[?])

29.5.5 Timing

Pulse	Width
Pilot	34/44100 s
Wide	22/44100 s
Narrow	12/44100 s

29.6 Lower Silesian Turbo 2000

29.6.1 Description

This turbo system appears to be very similar to the Turbo 2000 from former Czechoslovakia.

Pulse types	Narrow, wide, pilot tone, sync
File format	Various formats. For details, see below.
Block format	Pilot tone (at least 256 pilot tone pulses), sync pulse, data
Bit order	MSB to LSB
Signal input	DATA IN
Turbo mode activation	COMMAND active

29.6.2 Auto Turbo Format

This format is used to store binary files with limitations. Binary files without INIT segments can be stored and executed. Binary files with INIT segments can be stored, but not executed.

File stored using Auto Turbo format consists of two blocks - header block (HEADER) and data block (DATA).

29.6.3 Unknown Exterminator Formats (UE UBF and UE PBF)

These file formats can be used to store binary files. A binary loader is prepended before the file. A file consists of several blocks.

1. A dummy RUN block. 2 bytes of data (UE PBF only).
2. Pairs of data blocks for each segment. The first block in a pair is a segment header (SEGHEAD), the second block in a pair holds segment data (SEGDATA).
3. Termination segment header

29.6.4 Auto Turbo Header block

Offset	Description
0	Always 0
1	File type. Always 4.
2-11	File name
12-13	Always 0
14-15	Length of file
16-17	Always 0
18	Check sum = (HEADER[0]) xor ... xor (HEADER[17])

29.6.5 Auto Turbo Data block

Offset	Description
0	Always 255
1-?	Data of the binary file (includes segment headers)
Last	Check sum = (DATA[0] xor ... xor DATA[?])

29.6.6 UE PBF Dummy RUN block

Offset	Description
0,1	0x02 0xE0
2	Check sum = 0xE2

29.6.7 UE UBF and UE PBF Segment Header Block

Offset	Description
0,1	First address of the following segment
2,3	Last address of the following segment
4	Check sum = (SEGHEAD[0] xor ... xor SEGHEAD[3])

29.6.8 UE UBF and UE PBF Segment Data Block

Offset	Description
0-?	Data
Last	Check sum = (SEGDATA[0] xor ... xor SEGDATA[?])

29.6.9 UE UBF and UE PBF Termination Segment Header

Offset	Description
0-3	0x00,0x00,0x00,0x00
4	Check sum = 0x00

29.6.10 Timing

Standard transfer speed is approximately 2270 bauds. There is a big tolerance for width of all pulses.

Pulse	Center width	Tolerated range
Pilot tone	32/44100 s	(25-47)/44100 s
Wide	26/44100 s	(20-40)/44100 s
Narrow	13/44100 s	(6-19)/44100 s
Sync	10/44100 s	(4-17)/44100 s

30 Turbo Systems from Other Countries

30.1 Rambit Turbo Tape System from UK

Turbo system from the United Kingdom that can store monolithic and segmented binary files.

Pulse types	Narrow, wide. The default polarity is HIGH-LOW.
File format	Two formats. For details, see below.
Block format	Pilot tone (at least 8 wide pulses), data
Bit order	MSB to LSB
Signal input	INTERRUPT
Turbo mode activation	COMMAND active

30.1.1 Standard Boot Format

A file stored using this format consists of one single block (BLOCK). Length of the block, load address, and run address are not stored in the block, but in a loader.

Offset	Description
0	Identification byte. Always 0x5A
2-?	Data (length may vary)
?+1	Check sum = (BLOCK[2] xor ... xor BLOCK[?])
?+2	Termination marker. Always 0x00,0x00,0x00,0x00

30.1.2 Pseudo-Disk Binary File

A file stored using this format consists of one or more blocks (BLOCK). A new block is created after each INIT segment. This allows proper loading of segmented binary files with INIT segments. There is also no check sum.

Offset	Description
0	Identification byte. Always 0x5A
1-2	Always 0x00, 0x00
[This part of the block can repeat
3-4	First address of a segment. If there is special value of 0x??, 0xFF, the block ends right here, INIT segment is executed and new block follows
5-6	Last address of a segment If there is special value of 0x00,0x00, end of file was reached
7-?	Data of the segment
]	End of part that can repeat

30.1.3 Timing

Transfer speed is approximately 3200 bauds.

Pulse	Width
Wide	19/44100 s
Narrow	10/44100 s

30.2 Turbo 6000 from GDR

This turbo system can store compatible binary files (refer to section 22.2).

Pulse types	Narrow, wide. The default polarity is LOW-HIGH.
File format	Header block followed by one data block
Block format	Pilot tone, sync sequence, marker byte, data
Bit order	MSB to LSB
Signal input	PROCEED
Turbo mode activation	COMMAND active

30.2.1 Natural File Format

A file stored in the natural file format consists of a header block (HEADER) and a data block (DATA) that holds a binary file (including binary file header and segment headers).

The Turbo 6000 loader works as follows:

1. Load the data block to a memory area determined by the header
2. Read segments of the binary file from the memory area and place the data of the segments to memory locations determined by the segment headers
3. Execute initialization code via the INITAD vector
4. Run the binary file via the RUNAD vector

This processing imposes severe limitations to the binary file:

1. Maximum binary file size is limited to 46592 bytes
2. The binary file is limited to one INIT segment
3. The binary file must have a RUN segment
4. The segments in the binary file must be ordered by their first addresses

30.2.2 Header Block

Offset	Description
	Pilot tone. Series of at least 64 0x02 bytes.
	Synchronization sequence 0x09,0x08,0x07,0x06,0x05,0x04,0x03,0x02,0x01
	Marker byte. Must be nonzero
0,1	First address of buffer (BUFRLO, BUFRHI)
2,3	Last buffer address (BFENLO, BFENHI) increased by 1
4-23	File name (internal code)
24	Padding byte (always 0)

30.2.3 Data Block

Offset	Description
	Pilot tone. Series of at least 64 0x02 bytes.
	Synchronization sequence 0x09,0x08,0x07,0x06,0x05,0x04,0x03,0x02,0x01
	Marker byte. Must be zero
0-?	Bytes of the binary file (including segment headers)
?+1	Checksum = BLOCK[0] xor BLOCK[1] xor ... xor BLOCK[?]

30.2.4 Timing

Pulse	Width
Wide	14/44100 s 16/48000 s
Narrow	8/44100 s 10/48000 s