



# RTTYArt Capture Program

by  
William Bytheway



## Forward

Over the years there have been many hours spent creating pictures using only the characters on the keyboard. In November 1970, Don Royer wrote, "We (the XYL Maxine and I) have found that there is much basic art work, available from which RTTY pictures may be made. Cartoons, the comic strips, post cards, magazines, newspapers, center-folds and photographs may all serve as bases for pictures. While these may not be the right size, an inexpensive child's pantograph may be used to enlarge or reduce them. A portrait of Washington was made from the etching on the dollar bill. While it is not that important, if you have a little sketching talent, that will also help (or enlist your wife and friends as I did)."

John Foust, of The Jefferson Computer Museum, has also assembled an excellent dissertation on, and examples of, Teletype and ASCII Art. He has included a number of thumbnails of examples of some highly detailed pieces of this art form. John says, "This early computer graphics output used the visual density of printed characters to approximate gray-scale imagery, sometimes with raw letterforms, sometimes with overstrike techniques. I've collected a few in electronic form, but I'd like some originals, too."

The original pictures were composed using BAUDOT Teletype machines and punched paper tape as the storage medium. Recently an inquiry was made on the Internet to find anyone who has this art stored in his or her basements. To our surprise, there were thousands of paper tapes just waiting to be viewed. There have been various projects to capture the paper tapes to a computer in ASCII form, but no one has ever built an integrated package with a graphical user interface (GUI). This paper will describe such a project.

Sincerely,

*William H. Bytheway*

William Bytheway  
Senior Software Engineer  
Boeing Space and Defense

# RTTYArt Capture Program

by  
William Bytheway

## Introduction

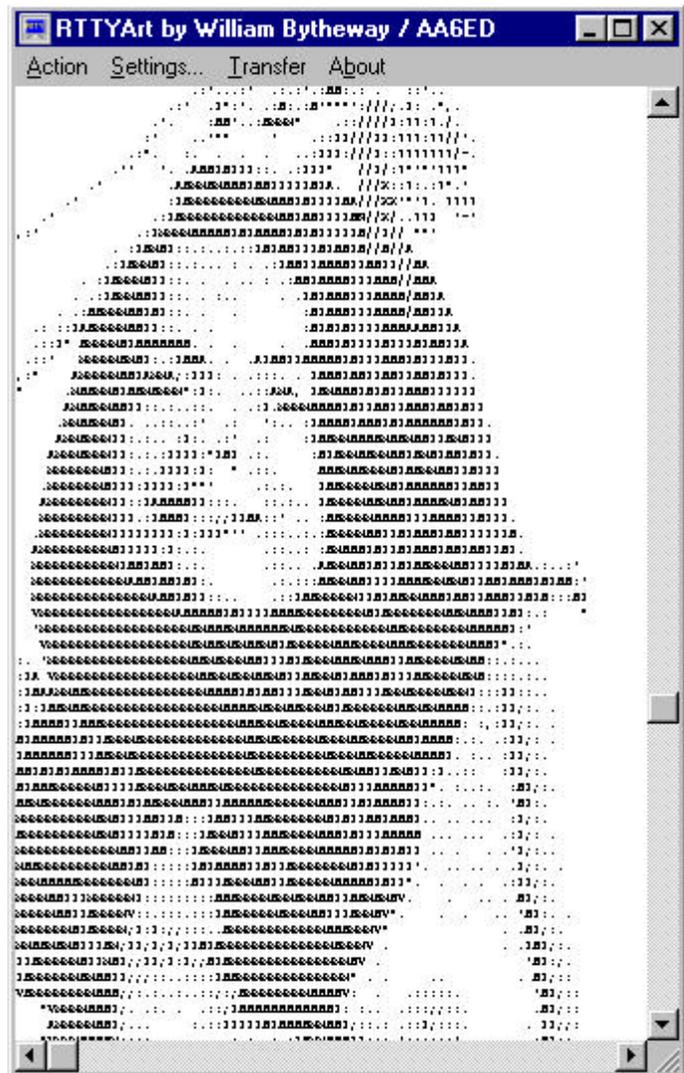
The project concept developed from discussion with various enthusiasts in the BAUDOT Teletype world. It was discovered that there was a need for a software application capable of capturing the slower 5-level BAUDOT code from existing hardware, converting it to 8-level ASCII and storing it in a data file without losing any data integrity. Most Microsoft Windows based editors will reformat text data to meet its carriage return and line feed format. This would destroy any overstrike capability used in the older BAUDOT machines. The capture project had to store the data in it's raw format, mapping each character, line feed and carriage return to it's ASCII equivalence.

## Design

In order to cut the design and development time required for this project, the Microsoft Visual C++ example database was researched for an RS-232 terminal application that provided most of the functionality needed for the example. The Microsoft Source Code Samples provided an application called "TTY" that provided a basic modem-dialing program along with a scrolling text display using graphical writes to the display. With minor modification, this allowed for the viewing of character overstriking, something not available with the standard Microsoft Foundation Class (MFC) CEdit libraries.

## Setting Up Communications

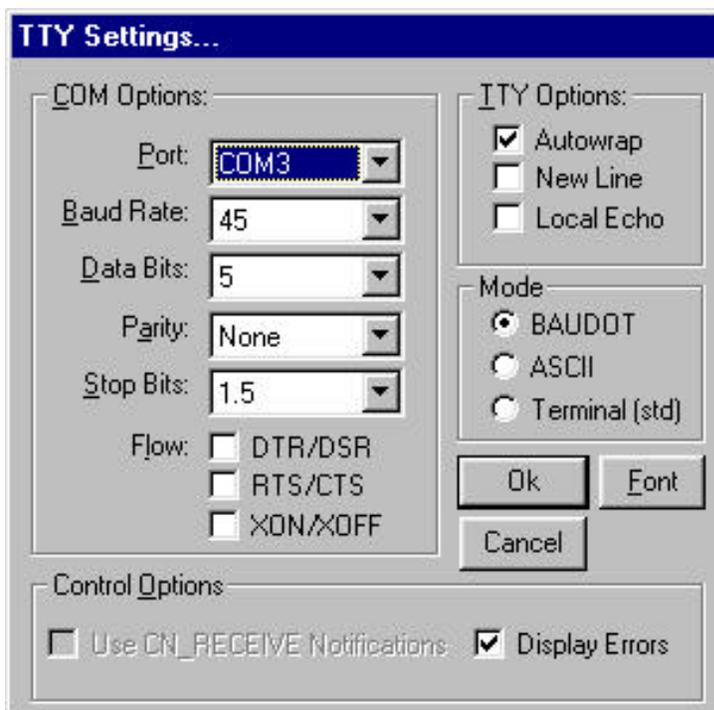
The "RTTYArt" software will allow users to import those paper tapes via a Hal Communications ST-6 modified for RS-232, 60-Mil current loop to RS-232, TNC or other interface. It will also allow a Terminal Node Controller (TNC) to be used if the user disables all carriage/line-feed/auto-return features in the TNC in order to capture files correctly with overstrike. The application runs on a Windows 95 or later computer.



## RTTYArt Capture Program

The TTY settings dialog box allows the user to define the required settings for the RS-232 port to communicate with the Teletype machine. Following is a description of the settings.

1. **Port:** The communications port should be set up for the correct port address and interrupt ID using the appropriate Microsoft Windows configuration utilities. Currently only the first 4 communications ports are supported. The port selection is "Grayed" after you connect so that it cannot be changed until the user disconnects.
2. **Baud Rate:** The default baud rate is 45 Baud, but the user can select from a number of baud rates commonly used for the BAUDOT code and also ASCII modem transmissions.
3. **Data Bits:** The data bits used for the BAUDOT code is 5-Bits, but this application allows the capture of 5,6,7 and 8 bit data. Use the ASCII mode for capturing data in its raw state.
4. **Parity:** This is provided for some interfaces that require parity support. BAUDOT does not have parity, so this box is normally left unchecked.
5. **Stop Bits:** For the BAUDOT code, 1.5-Bits is the standard, but if you are using other data bit configurations, you can select from either 1.0 or 2.0 stop Bits.
6. **Flow:** These should only be used in the Terminal mode. The DTR/RTS signal lines are normally used on the Hal Communications ST-6 and other Terminal Units for transmit and morse code identification.
7. **TTY Options:** These options only affect the displayed data, not the data that may be saved to a file. The functionality is pretty much standard for a common terminal unit.
8. **Mode:** The BAUDOT and ASCII modes do not assert the DTR and RTS lines, so that the user may use them for capturing data in the raw state. The Terminal mode asserts the DTR and RTS lines for communicating with a modem.
9. **OK:** Says you are happy with the selection, and they are saved to the communications port.
10. **Font:** Allows the user to change the character font. It is suggested that the user stay with the Courier, System or Fixed Pitch fonts for proper picture display.
11. **Cancel:** Just as it says, cancels any changes to the communications port.
12. **Display Errors:** Diagnostics for the developer, if there are any display problems, please report them to the software developer.

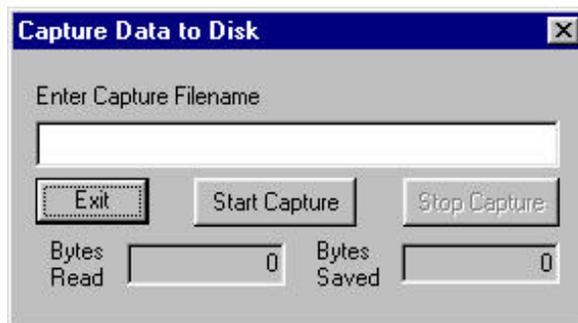


Following is a summary of the user selectable options from this window.

<b>Com Ports:</b>	COM1 thru COM4, configurable via Windows95 and later
<b>Baud Rates:</b>	45,50,57,74,100,110, and all ASCII Baud Rates thru 128K.
<b>Data Bits:</b>	5, 6, 7, 8
<b>Parity:</b>	odd, even, none
<b>Stop Bits:</b>	1, 1.5 and 2 (where 1.5 is used for BAUDOT)
<b>Modes:</b>	ASCII, Terminal and BAUDOT (Military)
<b>Line Ctrl:</b>	Overstrike or automatic New-Line at 80 characters
<b>Local Echo:</b>	On or Off
<b>Fonts:</b>	Selectable for each workstation
<b>Flow Control:</b>	DTR/DSR, RTS/CTS, and XON/XOFF for Terminal mode only

## Capturing Files to Disk

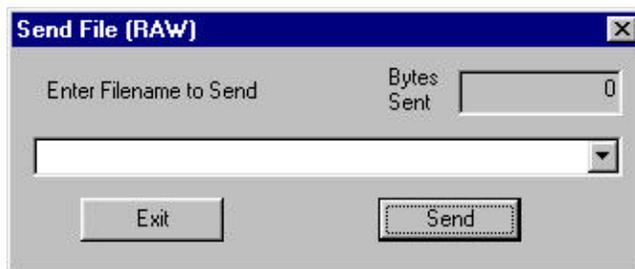
Under the transfer menu, there is the “Capture Files to Disk...” option which allows the user specify a path and file name, start the capture, and stop the capture when finished. The dialog box performing the capture remains in focus during the capture, but allows the display window to scroll and show the picture. This was designed this way so that the user would have direct control over the captured data. Two windows show the bytes read and the bytes saved. The difference between the two will become evident when the letters, figures and nulls get stripped from the incoming data during the save process.



The capture process is quite critical for the recording of the data without errors. When you start the tape, you will notice that the “Bytes Read” does not equal the “Bytes Saved”. That is because the leading Letters and Null characters are stripped from the recorded data. The resultant ASCII file will be a pure recording of only the BAUDOT character set, line feeds and carriage returns. While the data is written to the display, it is also being written to the file exactly as it is being received. As the display scrolls down, certain overstrike information is lost, but the data recorded to file is exactly as received.

## Sending Files

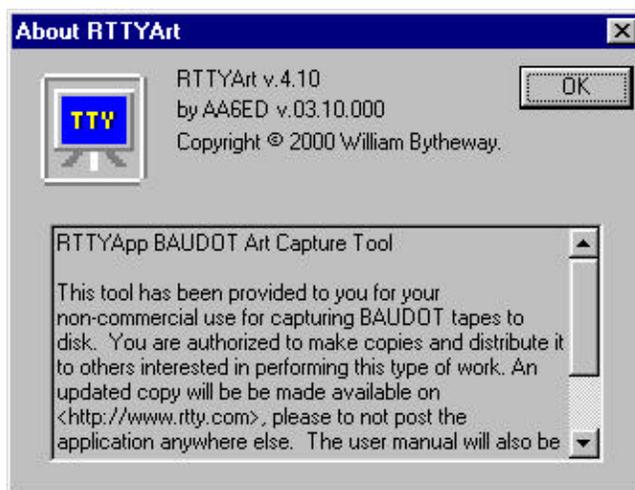
It was also a requirement for a user to send an ASCII file to the RS-232 port in the desired format, either ASCII or BAUDOT. Selecting this menu option brings up a dialog box that allows the user to enter a path and filename. During transmission, the bytes sent are shown, and focus is not returned to the main window until the entire file is transmitted.



## About

The about box brings up a simple dialog box that describes some of the capability and some simple instructions on how to use this application. The text says:

“This tool has been provided to you for your non-commercial use for capturing BAUDOT tapes to disk. You are authorized to make copies and distribute it to others interested in performing this type of work. An updated copy will be made available on <<http://www.rtty.com>>, please to not post the application anywhere else. The user manual will also be made available at that WEB site.”



### Basic Instructions

1. Create a directory for the executable on your hard drive and place the application RTTYArt.exe in it. Run the application.
2. On the title bar, poke the "Settings..." and a dialog box should appear that allows you to set your COM Options, TTY Options, Mode and Font. The default is BAUDOT, COM2 at 45 BAUD. Press OK and continue.
3. Poke on "Action" and select "Connect", you should now be connected to your RS-232 port. The display now operates in a TTY terminal mode and should be able to communicate with your iron horse (teletype machine). The display supports overstrike and 300 lines. It does not auto-scroll and it is a read-only display, no editing.
4. Once you have verified that your connectivity, place a tape in your TD and you are ready to capture.
5. Poke on the "Transfer..." option and select "Capture Files to Disk...". Another dialog box should appear. Type in the name of a output capture file, poke "Start Capture" and start your TD. The counters will show bytes stored, and bytes received. Only characteres are stored, LTRS, FIGS, NULLs are discarded. When you are finished, press "Stop Capture" and the file will be closed. You have now created an ASCII file. Poke Exit for the next demo.
6. Under "Transfer..." select "Send File...". A similar dialog box will appear. A drop down dialog box will show the files currently in the directory. Either select one of those files, or type in the path and filename of the file you want to transmit. Press return or "Send". Your iron horse should now be typing the text.
7. Under "Transfer..." I have also provided a graphical user interface for the famous Tom Jennings ASCII/ITA#2 translation programs. Type in the filename and press the "Do It" button for the action you want. The output file will be created in the same directory as your input. Reference Tom's WEB site for more information about these programs.

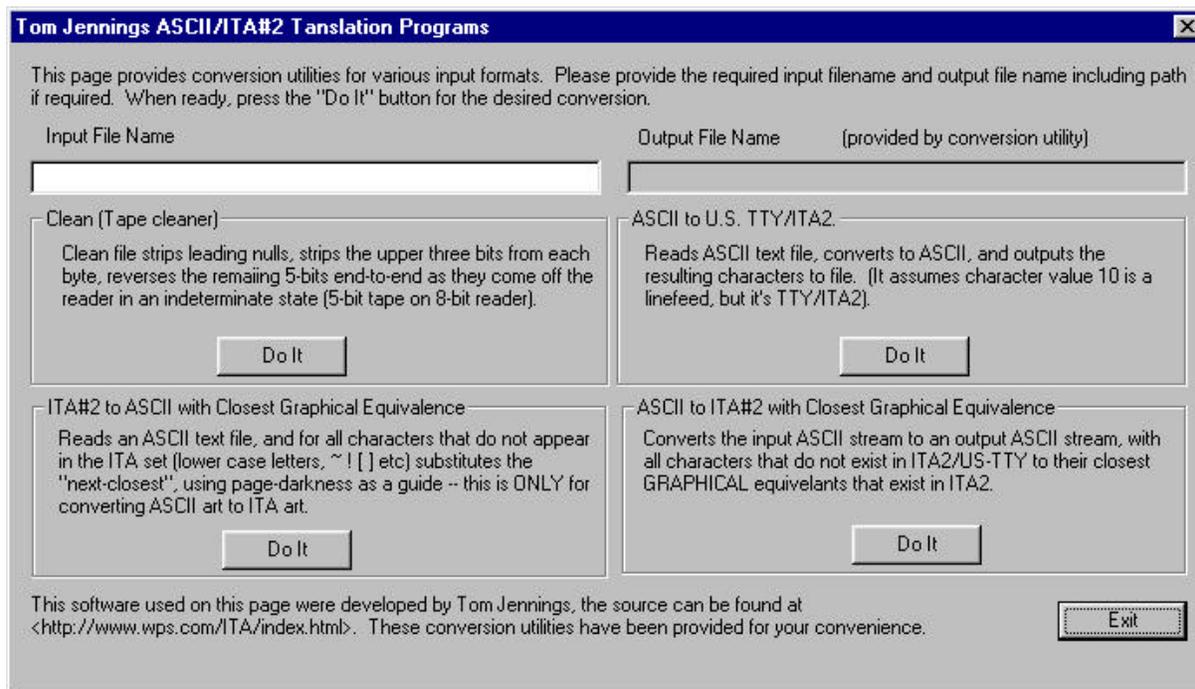


### Tom Jennings ASCII/ITA#2 Translation Code

Another enthusiast, Tom Jennings, wrote several 'C' applications that provided the means of converting data files from one format to another. Among his tools are some simple ASCII/ITA translations programs, written in ANSI C, with DOS executables. They assumed U.S.TTY or ITA2 code.

- **Tape cleaner:** clean file strips leading nulls, strips the upper three bits from each byte, as they come off the reader in an indeterminate state (5-bit tape on 8-bit reader). DOS binary clean.exe; C source clean.c.
- **ASCII to ITA2:** a2b file reads ASCII text file, converts to ITA2, and outputs the resulting characters to file "filename.ITA". DOS binary a2b.exe; C source a2b.c.
- **ITA to ASCII:** b2a file reads ITA text file, converts to ASCII, and outputs the resulting characters to file "filename.ASC". DOS binary b2a.exe; C source b2a.c.
- **ASCII to "ITA" subset of ASCII:** A2ITA-set file reads an ASCII text file, and for all characters that do not appear in the ITA set (lower case letters, ~ ! [ ] etc) substitutes the "next-closest", using page-darkness as a guide -- this is ONLY for converting ASCII art to ITA art. DOS binary A2ITA-set.EXE; C source a2ita-set.c.

His tools were incorporated into the RTTYArt project with an added graphical user interface (GUI) wrapper. The DOS calls were replaced with GUI calls for basic input and output.



The assumption is that the user is able to interface the Teletype 60-mil BAUDOT current loop to an RS-232 input into the computer. In it's raw state, the incoming data will be 5-level BAUDOT code, or 8-level ASCII. Several transfer capabilities were added to the application.

To use the dialog box, the user simply plugs in the input file name and selects the appropriate "Do It" push button. The Output File name box will display the converted file name. Many thanks to Tom Jennings whose WEB site can be found at <http://www.wps.com>.

### **Conclusion**

The project has been considered a success. The final product has been delivered to the users that have expressed an interest in converting the old BAUDOT paper tapes to ASCII. Thus an old form of artwork will be saved for future generations.

It has been distributed to a group of teletype enthusiasts that will be soon saving their tapes to ASCII files which will later be added to the <http://www.rtty.com> art archives. The viewer has already been developed in JAVA and allows the viewer to see the entire picture, regardless of the length.

Future plans are to add an editor to the application that will allow the user to edit the raw data input format including carriage returns and line feeds. Current editors will not allow for the selection of overstrike mode on text, which is used commonly for BAUDOT RTTY art. This project has been fun, please enjoy the results of my work.

Sincerely,

*William H. Bytheway*

William H. Bytheway