

# TIC 3.3 UPDATE



# LABS

TIC version 3.3 represents one of the more important updates to TIC since TIC was released back in 1986. This document explains changes made since the TIC 3.1 documentation was written and should be considered a supplement. Just about everything present in the TIC 3.1 documentation still holds true. We've just added some more features to make TIC more convenient to use.

There are three main areas where TIC 3.3 has changed over TIC 3.1. First, the scripting language has several new commands available and a few of the older commands have added functionality. Secondly, there are some new features in the rest of the program as well as some enhancements to old features. Lastly, there are a few bug fixes here and there.

## BUG FIXES AND TWEAKED FEATURES:

1. A few bugs in the Ymodem Batch Upload Mode have been fixed. These fixes enhance compatibility with some host Ymodem programs that weren't well supported previously.
2. The Text Editor has been updated to version 3.0.3 to fix a few bugs including a bug that caused annoying screen flashing if you tried to use the editor on a mouse equipped Apple IIe.
3. The Text upload feature has been updated to properly handle uploads of AppleWorks version 3.0 word processing files.
4. The recording buffer is now about 2000 bytes smaller to make room for other new features. This leaves about 44K capacity.
5. The delay between sending the X-Off signal and saving the copy buffer has been increased from 1 to 2 seconds to account for slow hosts.
6. Some bugs in the multi-string waitfor statement have been fixed

## NEW COMMAND MODE FEATURES

### Conference Duplex Mode Added

The C-E command is used to switch between TIC's various DUPLEX modes. These modes now include FULL, HALF, CHAT, and now CONF Duplex Mode. CONF duplex mode is a new split-screen mode where the bottom 3 lines of the screen go to inverse video and accept a long line of user input that is not transmitted to the host until the RETURN key is pressed. Control characters, such as ^S or ^C or ^Q are passed to the host immediately. The top 21 (or 20 depending on whether the status line is enabled) lines of the screen are reserved for data received from the host. The point of CONF duplex mode is to allow you to type what you want to send and get it right in spite of simultaneously receiving information from the host. This would be useful if there's too much line noise to reliably edit what you send to the host or situations where you are

in an online conference on GENie, Compuserve or multi-line BBS systems where what other users type would otherwise interfere with your own ability to type messages. HAM radio operators using TIC with packet radio networks would also find this useful.

### Editing Keys

While you are typing your line of input you can press the Left arrow key or the DELETE key to erase what you've typed, one character at a time. The Control-X key will abort the entire line of input, and RETURN or ENTER will send the line of text to the host.

### Side Effects of Conference Duplex

If you write scripts that address the screen using the GOTOXY command, note that Y coordinates that would place the cursor into the user input area at the bottom of the screen will be converted by TIC to keep the cursor in the output area at the top part of the screen. This may mess up formatting of some windows or text displays your scripts may make so you'll need to make adjustments if your scripts will be used while in Conference Duplex Mode.

### Filing Menu

The C-F command brings up a Filing menu where all the file manipulation commands are now gathered (Delete, Rename, Copy, Catalog, Volumes, Type, and Prefix). This frees up the old commands for other uses.

All other TIC commands are supported as usual from Conference Duplex Mode including script file execution commands and macro keys.

## MULTIPLE CONFIGURATION FILES

When you use the C-M command to change various TIC configuration options, you'll be asked what file to save the information to and the TIC.Config filename will be presented as the default value. Pressing RETURN or ENTER here will just save the TIC.Config file as it always did. You may, however, optionally enter the name of another file such as GENie.Config or whatever. Two new script commands have also been added to allow your scripts to save the current configuration to a named file or to load a new configuration from a named file (see section on new scripting commands for syntax). This will be handy to simplify scripts for various services as you'll be able to change multiple configuration settings with a single script command to load a named configuration file rather than setting all of the various configuration options via individual statements. This feature will also let you quickly reload a default configuration file after you've logged off from a service that needed non-standard configuration options.

### Side Effects of Multiple Configuration files

Be careful that you don't try to load a file that isn't really a configuration file. TIC tries to keep this from happening but if you manage to get past TIC's checking you would crash the program by loading garbage data into TIC's configuration variables.



## NEW SCRIPTING FEATURES

### Predefined Variables

A new \$Matched predefined variable has been added. It is used in conjunction with the "Waitfor String" and "On" script statements described later. Basically, it is an integer numeric variable that will save the string number of the most recently matched string from a Waitfor String statement. Example:

**waitfor string "CONNECT" "BUSY" "NO"**

If the first string is matched then the \$Matched variable will expand to "1" and if the second or third string is matched then \$Matched will expand to "2" or "3" respectively. If no string was matched before the timeout then \$Matched will expand to "0".

### Script Statements

#### **If NotExists <filename> <statements>**

This is the opposite of the If Exists (See Page 65 of TIC 3.1 documentation) statement present in TIC 3.1 and is used to conditionally execute <statements> if the <filename> is not found to exist. Example:

**if NotExists myfile do MakeMyFile**

In the above script statement, the subroutine "MakeMyFile" is executed if "myfile" isn't found to exist.

#### **LoadConfig <Config filename>**

This statement loads a TIC configuration file saved previously either from the C-M menu or from the SaveConfig script statement (described later). Example:

**LoadConfig TIC.CONFIG**

The above statement would load the TIC.CONFIG file and update various settings to those contained in the named file. Note that if a fully qualified pathname is not specified, that TIC will assume that you are referring to TIC's Root directory as the location of the configuration file.

#### **On <Number> <Label1> <Label2> <Labeln>**

This is basically a calculated GOTO statement. <Number> would ordinarily be a variable such as "\$1" or "\$Matched" and would contain a number of the Label you want to branch to. In the above example, if <Number> was equal to "2" then <Label2> would be branched to.

#### **Record "String"**

"String" is written directly to the recording buffer. Control characters are allowed as part of "String" just as they are in other strings. One fast way to create a text file would be to use several Record statements followed by a Buffer Write statement.

#### **SaveConfig <Config filename>**

This statement saves TIC's current configuration variables to <Config filename>. The configuration file can later be loaded by the LoadConfig statement described earlier in this document. If the file you save is named TIC.Config then the file will be loaded by TIC the next time TIC is loaded into memory. Note that if a fully qualified pathname is not specified, that TIC will assume that you are referring to TIC's Root directory as the location of the configuration file.

#### **Set CaseSensitive <On/Off>**

This statement determines whether the Waitfor String, If Equal, and If NotEqual statements will be case sensitive (i.e. whether they will require an exact match where MyString does not equal MyString). The default setting is Off. This statement would be typically used for those situations where scripts are being accidentally tripped by similar, though not exact strings that might differ only in the case of various letters within the string.

#### **Set Duplex Conf**

See page 2 for details of Conference Duplex Mode. This statement enables it from a script.

#### **Set ULCapture On/Off**

Setting ULCapture On allows TIC to record the host echo of Text mode uploads into the recording buffer if the recording buffer is turned on. Normally the recording buffer is disabled by TIC during Text mode uploads. Some users requested this feature so they could examine the recording buffer to detect error messages that might come from the host during a Text mode upload. The default is Off.

#### **Set Verbose On/Off**

This statement allows you to suppress most screen messages and dialogues that would otherwise appear on-screen during scripts such as the Set Prefix window or the Buffer Save window and so forth. The effect when Verbose mode is set to Off is that your scripts will look cleaner as they execute. The default is On.

#### **Waitfor String "Str1" "Str2" "Str3"**

This statement has been enhanced to allow your scripts to wait for up to three different strings with a single statement. Nothing has changed if only a single string is specified. A special predefined variable "\$Matched" (see page 5 of this document) is set each time the Waitfor String statement is used to the string number of the matched string or to "0" if no string was matched before the timeout limit. Testing for \$Matched being equal to "0" has the same effect as testing for If Failed. Example:

**waitfor string "CONNECT" "BUSY" "NO"  
if failed goto abort  
on \$matched Login, Redial, Redial**

In the above script statement, if none of the three listed strings are found before the timeout timer expires then the statement following the Abort label will execute. If the "CONNECT" string is received first then the statement following the Login label will execute. If either the "BUSY" or the "NO" strings are received first then the statement following the Redial label will execute.

#### **Set Timer 0**

This statement is supposed to wait forever on strings or keypresses. This didn't before but does now.

#### **Set ULCapture ON**

This statement should capture everything coming from the host during a text upload. It didn't work right before with prompted uploads but it does now.

#### **Create <Dir Name>**

This statement has been added to allow a script to create a subdirectory file.

#### **GetKey <Variable number>**

This statement is used to wait for a keypress or modem byte and save it in the specified variable. The variable is null if a timeout occurs first and FAILED is set to true.

#### **RENAME <file 1> <file 2>**

This statement allows you to rename a file from a script. This can also be done manually from the C-F menu.

#### **COPY <source> <destination>**

This statement lets you copy any non-forked file and you can do it manually from the C-F menu.