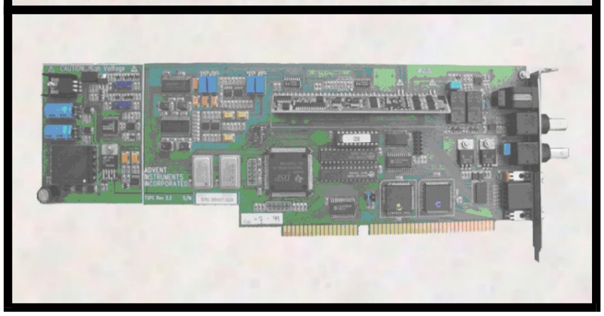
CID1500

FSK Caller ID Simulator Software



User Guide & Reference Manual

Advent Instruments Inc.

Release 7 - May 2004

Copyright 2004 - Advent Instruments Inc. All rights reserved.

Printed in Canada

Advent Instruments Inc.
111 - 1515 Broadway Street
Port Coquitlam, BC, V3C6M2
Canada

Internet: techsupport@adventinst.com
sales@adventinst.com

Web Site: http://www.adventinstruments.com

Telephone: (604) 944-4298
Fax (604) 944-7488

■ Overview

Table of Contents

■ Section 1 Installation & Setup

Verifying the System Requirements

The CID1500 Caller ID Simulator requires the following minimum system setup for the target computer.

- Intel 486 PC computer, although a pentium class (or equivalent) is highly recommended
- VGA type monitor
- Microsoft Windows 95 / 98 or NT 4.0 / 2000 operating system
- Sixteen Megabytes of RAM
- One full length 16 bit ISA expansion slot (*)

Note: The ISA expansion slot is only required when using the TSPC (Telephone Signal Processing Card) hardware. If the CID1500 software is used with the AI-7280 Central Office Line Simulator, then only a free RS-232 communications port is required.

Installing the Telephone Signal Processing Card

Before installing the Telephone Signal Processing Card (TSPC), verify that the correct base I/O address is set. The base I/O address used is determined by the settings of two jumpers shown in figure 1. The factory default settings includes both jumpers, as shown below, for a base I/O address of 0x0380. This address should be free of conflict from most other peripheral cards; however, in case of conflict, change the jumpers to select an alternate base I/O address. The CID1500 program will automatically scan all four possible base I/O address for a TSPC. Note, that the TSPC is designed to avoid interrupt and DMA channel conflicts, and as such, these parameters do not need to be configured. The only possible conflict between peripheral cards is with the base I/O address selected.

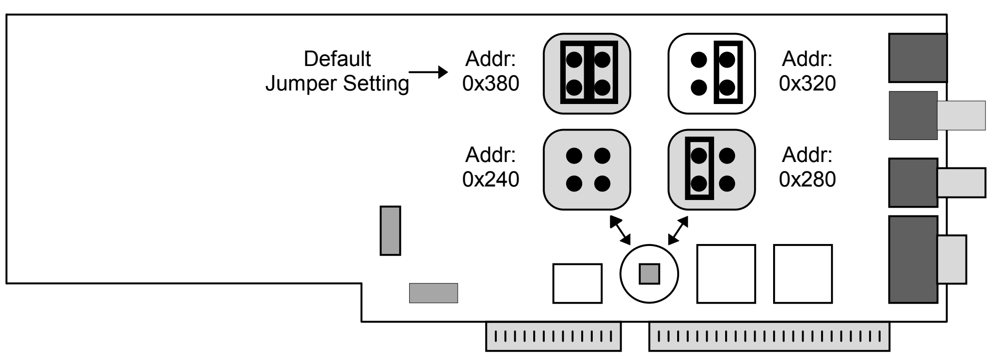


Figure 1. Verifying Base I/O Address

To install the TSPC:

1. Turn off the computer, including all external peripherals
2. Leave the power cable connected to a grounded outlet, so that the system is grounded
3. Remove the cover from the computer
4. Locate an unused 16 bit ISA expansion slot in your system
5. Insert the TPSC into the expansion slot
6. Replace the cover from the computer

Note: The TSPC can create high voltages. Ensure adequate space between the TSPC and other adjacent peripheral cards in the computer. Also, always re-attach the cover to the computer before turning it on in order to insure that no contact can be made to the high voltage portions of the TSPC.

Installing the Windows Software

To install the software, turn on the PC and launch either the Windows 95 / 98 or NT 4.0 / 2000 / XP operating system. A few moments after inserting the supplied CD, a setup program should be running which guides you through the installation procedure. Follow the instructions presented by the setup program and when prompted to select for which programs to install, chose the “CID1500: FSK Based Caller ID”.

The setup program will extract all the necessary files and install them into the directory you select. The default directory is set to “C:\AI\CID1500”; however, alternate directories may be chosen.

The Setup program will automatically create a program group called “Advent Instruments” and program icon within the Windows program folder.

Note: If the setup program does not execute automatically when the CD is inserted into the computer, it must be started manually. To do this, click the mouse on the START button within the task bar, then select the RUN option. A small dialog box appears requesting the name of the program to run. Type “x:\setup.exe”, where “x” is the drive letter of the computer’s CD ROM.

Note: The software does not require any modifications to the AUTOEXEC.bat or CONFIG.sys files. However, the WIN.ini file is modified to contain certain program initialization values.

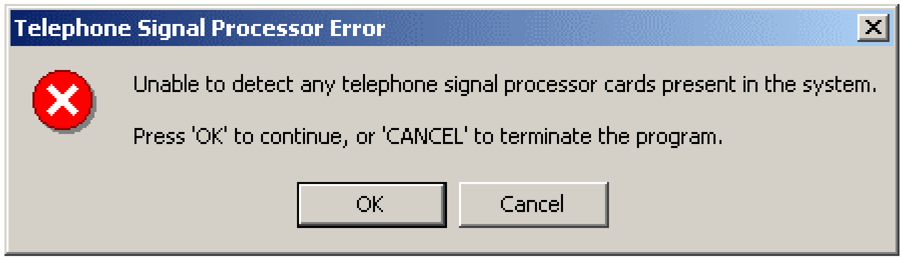
Using Windows NT 4 and Windows 2000 / XP

The standard release of the CID1500 software contains TSPC (Telephone Signal Processing Card) drivers that support Windows 95 and 98. The Windows NT & 2000 operating systems are not compatible with the standard CID1500 drivers. However, a version of the driver that operates under Windows NT & 2000 is available from the installation CD (NTsupport directory) or our web site at www.adventinstruments.com. The driver package includes instructions for its installation in a Windows NT 4.0 & 2000 system. If problems are encountered during the installation, please contact us for assistance.

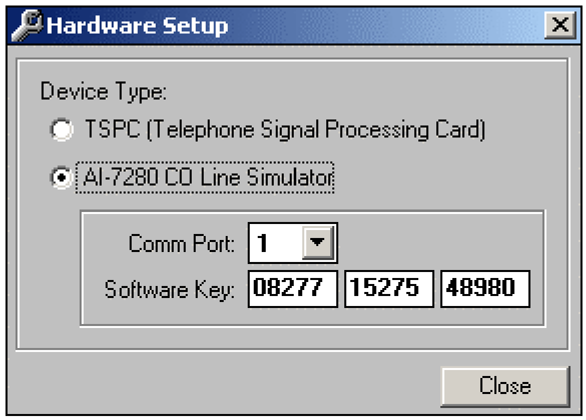
Note: If using the CID1500 software with the AI-7280 Central Office Line Simulator additional drivers for Windows NT4, 2000, or XP are not required..

Using the AI-7280 CO Line Simulator

The CID1500 software operates with either the TSPC (Telephone Signal Processing Card), or the AI-7280 hardware. If using the AI-7280, it must be connected to one of the PC’s RS-232 serial communication ports. Following the CID1500 installation, the software expects to use the TSPC hardware by default. As such, if the PC does not contain a TSPC an error is displayed when the CID1500 program is launched. The following figure shows an example of this error message.



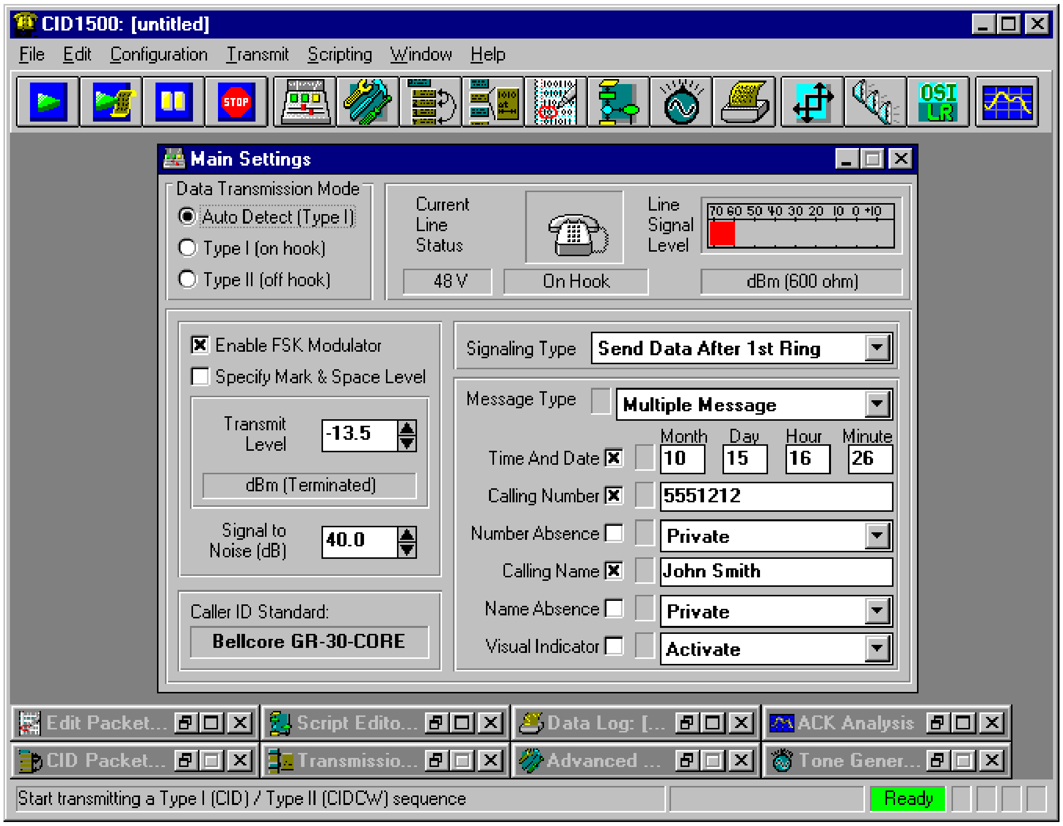
The CID1500 must be configured to search for an AI-7280 instead of a TSPC at start up. To change the setting, click the 'OK' button for the above message and then select the [HARDWARE SETUP] command from the [CONFIGURATION] menu. This displays the following window, from which the CID1500 software can be told which hardware device to use.



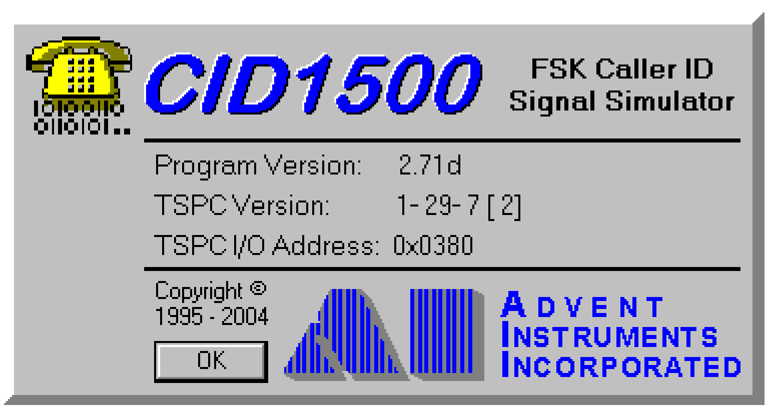
Click the mouse on the "AI-7280 CO Line Simulator" option as shown in the above figure. The PC communication port number is selected from the drop-down list labeled "Comm Port". This setting ranges from port number 1 to port number 8. As a final step, enter the software key value. The key is composed of three groups of five numbers. Each AI-7280 using the CID1500 requires a separate key. Once the "Close" button is clicked, the CID1500 will attempt to communicate with the AI-7280 using the specified communication port. If the AI-7280 can not be found or the key is incorrect, an error message is displayed.

Testing the Installation

Once you have finished installing the hardware and software, you can verify the correct setup by launching the "CID1500" program from the Window Program Manager. Once the program has finished loading itself in the system, the screen should display the following.



To verify the software version and the TSPC version codes, select the HELP menu by either clicking on HELP or pressing ALT-H. Select the "ABOUT CID1500" option within the HELP menu. The following window will appear until you press the CLOSE button.



The window will display the software version code, TSPC product, revision, and option codes, and the TSPC base I/O address being used. If any problems are encountered in the use of this product, include the above information in any correspondence with technical support.

■ Section 2 Getting Started

Starting Up

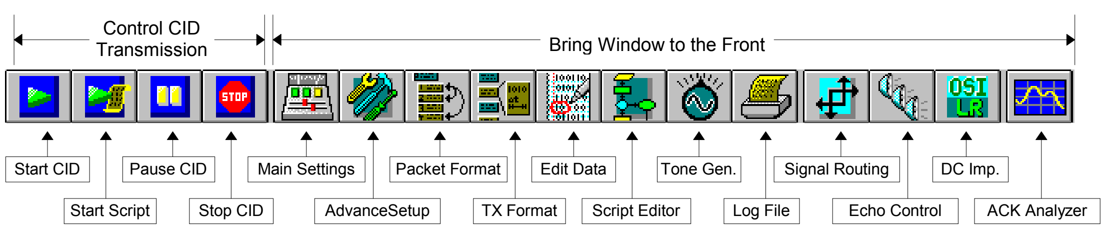
Once the CID1500 Caller ID Simulator program has started, it is ready to begin sending Caller ID data transmissions. Simply connect a telephone up to the telephone jack on the rear of the Telephone Signal Processing Card (TSPC) installed in the computer, and press F5.

While this program is written with a Windows Graphical User Interface for ease of use, the program allows for sophisticated test sequences in order to deal with some of the complex aspects of testing Caller ID products thoroughly.

The program is structured around the use of multiple sub-windows, each which allows you to control various aspects of CID data transmission, all contained within one master window. The master window contains the menu, toolbar, and status line, which allows you to control the CID test sequences and monitor its progress.

The Menu & Toolbar

Use the menu to load/save various files used in the program, send output to the printer, select the Caller ID standard to follow, start/stop/pause Caller ID transmissions, display various sub-windows, show help files, and to exit the program. To speed up selecting common actions that use the menu, a tool bar has been created. Each button in the toolbar is equivalent to its menu selection.

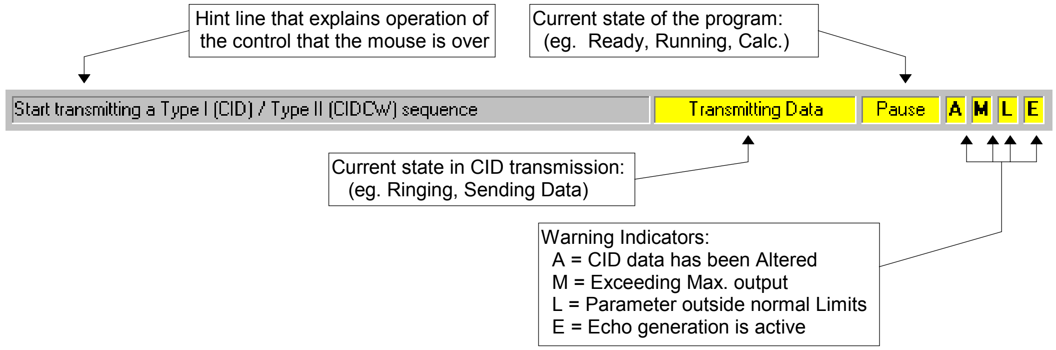


The first four buttons control the transmission of the Caller ID data. Pressing the START button, launches a Caller ID transmission. PAUSE will suspend its state indefinitely, while STOP will terminate it. START SCRIPT will start the execution of scripting programs, which help to automate complex test sequences.

Clicking any of the following ten buttons on the toolbar will cause the program to bring the selected sub-window up to the fore front.

The Status Bar

At the bottom of the master window, is the status line. Here you can instantly determine the current state of the program. The following diagram helps to illustrate some examples.



The first frame, the Hint Line, helps to explain the operation of most of the controls in the sub-windows, whenever the mouse is placed over them. The next frame shows the current state of any Caller ID transmissions taking place. When a transmission is active, it will be highlighted in yellow, with the text indicating the current action being taken. The following frame indicates the status of the program. It may display “READY” at idle times. During data transmissions, it will display either “RUNNING” or “PAUSE”. Also, when performing calculations, it may display “CALC.”.

The last four frames are warning indicators. These help remind the user that certain settings have been set to unusual values, or are outside recommended limits. The first indicator will illuminate with an “A” to signify an altered CID data stream. If any of the data packets have been byte or bit wise edited, an “A” will appear. The second indicator displays a “M” whenever the TSPC has been asked to generate simultaneous tones at a level that exceeds its output capability. The third indicator will display a “L” anytime certain transmission parameters fall outside the Bellcore, ETSI, or Australian specifications. Finally, the fourth indicator displays an “E” whenever the signal echoes option has been enabled.

Note: The rest of the document assumes that the Bellcore standard is the current operational standard. To change standards, Select the [CONFIGURATION] menu, and then the [CALLER ID STANDARD] option.

Note: The software remembers the Caller ID standard between sessions. So if you exit the program with the ETSI standard selected, the next time the program starts, it will be set to the same standard. The software also remembers its last screen position on the monitor.

The Main Settings Window

The sub-window first displayed when the program starts is called the Main Settings Window. This window is a collection of the most commonly changed parameters, and they have been collected together in one window for convenience. A sample snap shot of the Main Settings sub-window is as follows.

The key areas of the Main Settings sub-window are the Data Transmission Mode, Line Status, and CID Transmission Parameters.

The buttons in the Data Transmission Mode area determine what type of CID signals are sent. The “Type I” selection refers to the sending of CID data while the phone is on hook. The “Type II” selection, or called Caller ID with Call Waiting (CIDCW), refers to sending CID data while the phone is off hook. The default setting is the “Auto Detect” mode. In this setting if the phone is on hook, the Type I format is used; otherwise, if the telephone is off hook, the Type II format will be used.

The Line Status area of the Main Settings sub-window shows the current status of the telephone line. Indicated are the hook switch status, line voltage or loop current levels, and the signal level present on the telephone line.

The remainder of the window contains various controls where the Caller ID signal transmit level, the signal to noise ratio of that signal, and parameters determining what data is sent down the line are view and changed. Also shown is what standard any CID transmissions will conform to. These being either Bellcore, the European ETSI standards, or the Australian caller ID system.

Depending on the selected standard, you can choose from various signaling methods from the “Signaling Type” drop-down list. For the Bellcore Type I standard, these consist of either sending the data after the first ring, or sending the data without any ringing. The ETSI standard allows for many different options in the signaling type. These include preceding the data with a Dual Tone Alert Signal (DTAS), short ringing burst, and/or telephone line polarity reversal. The signaling types of the Australian system are generally similar to the ETSI signaling, with the additional option of using OSI (open switching intervals) events.

To change the message type, click on the drop down list and select the desired message format. For the Bellcore standard, the options are Single Message, Multiple Message, Single Message Waiting, and Multiple Message Waiting. A CID message is composed of one or many individual packets. The information contained in these packets is shown below the Message Type drop down list. For the Multiple Message format, they are Date & Time, Calling Number, Number Absence, Calling Name,

Name Absence, and Visual Message Waiting Indicator. Beside each packet title is a check box. To enable or disable the packet from being sent, click on the check box.

In the example shown above, the Calling Number packet, and the Message Type have “A” illuminated beside them. This shows that the Calling Number packet has been edited and changed. This may have been done by changing a byte in the packet to test how the telephone decoding software will handle the exception. Since the packet has been altered, and the packet a part of the message, the message is by default also flagged as altered. Likewise, the “A” will also be illuminated in the status bar of the main program since, if the message is altered, the entire CID transmission is classified as altered.

At this point, you can send most of the common Caller ID transmissions, by selecting the various message types, and packets. After changing any field, just click the START button on the toolbar, or select the [TRANSMIT] [START TRANSMISSION] from the menu, or press F5 to send the Caller ID data to the telephone under test.

■ Section 3 Program Operation

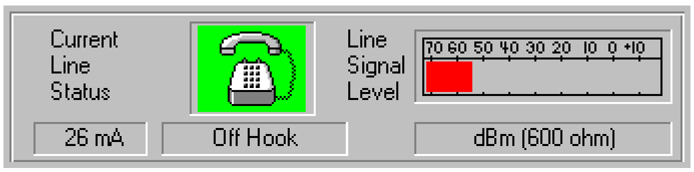
This section describes in more detail how to control and use the capabilities of the CID1500 simulator to perform Caller ID Testing. Major topics covered include the following:

1) Viewing the current status of the CPE under test
2) Controlling the progress of the Caller ID transmission
3) Setting the Caller ID signaling method to use
4) Modifying the message that is sent to the CPE
5) Changing various Caller ID transmission parameters
6) How to use the tone generation functions, and
7) Configuring the signal flow and echo generation
8) Using the ACK tone analyzer
9) Viewing the contents of the Log File

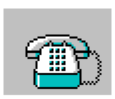
■ Section 3-1 Telephone Status

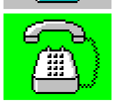
Viewing Current Telephone Status

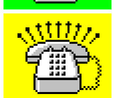
The current status of the telephone line is displayed in the MAIN SETTINGS window. This information is always being updated, regardless of the current program operation. The figure below is representative of the information displayed concerning the current telephone line conditions.

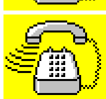


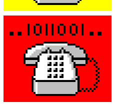
The telephone icons display various states or conditions experienced by the telephone under test. All of the possible telephone icons are shown below:

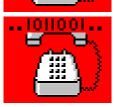

Telephone is On Hook


Telephone is Off Hook


Ringing Generator is Active


SAS/CAS or Alert Signal Active


Transmitting Data On Hook

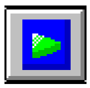

Transmitting Data Off Hook

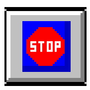
If the telephone is in an on hook state, the current line voltage is displayed to the left of the status icon. Likewise, if the telephone is in an off hook state, the loop current is displayed to the left of the status icon.

The Line Signal Level meter displays the current signal level present on the telephone line. The units of measurement is in dBm referred to 600 ohms.

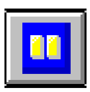
■ Section 3-2 Controlling Transmissions

Controlling a Caller ID Transmission

 **Starting,**



Stopping, or

 **how to Pause a Caller ID Transmission**

Starting a Caller ID transmission to a telephone under test is as simple as clicking on the Start icon in the toolbar. The menu option [TRANSMIT] [START TRANSMISSION] or pressing F5 will also start a Caller ID transmission.

Once started, the program status frame in the status line will display RUNNING. When a transmission sequence has been started, changing any parameters, or any aspect of the Caller ID data will have no effect on the current Caller ID transmission in progress. Any changes made will be reflected in any subsequent transmissions. If the tone generator had been active before starting the Caller ID transmission, it will be suspended until the transmission sequence is completed.

Once the transmission has finished its sequence the program status will return to READY, or TONE ON if the tone generator is enabled.

Pressing the Pause icon in the toolbar, or selecting the menu command [TRANSMIT] [PAUSE TRANSMISSION], or pressing F7, while a Caller ID transmission is active, will freeze the transmission sequence. PAUSE will be displayed in the program status frame in the status line. In the pause state, the Start command will allow the Caller ID transmission to continue until its completion. The Pause command has no effect unless a transmission is currently active.

During an active transmission, or in Pause mode, selecting the Stop command in the toolbar will terminate the transmission. The Stop toolbar button, or the menu command [TRANSMIT] [STOP TRANSMISSION], or the key F8 has no effect unless a Caller ID transmission is actively running, or is suspended via the Pause command.

Another option for starting a Caller ID transmission is by pressing F4, or selecting the menu command [TRANSMIT] [INVERT CHECKSUM and START]. This command will simply bit-wise invert the current message checksum and then start a Caller ID transmission. This provides an easy way to send invalid messages in order to determine how various CPE’s handle this situation. If the message had a valid checksum, after pressing the F4 key, the message will be flagged as altered due to the checksum inversion. Pressing the F4 key again will invert the checksum again resulting in valid message. However, if the Auto-Increment of Name/Number fields option is enabled, the checksum will be recalculated at the end of every Caller ID transmission as the name or number field is updated.

Enabling or Disabling the FSK Modulator

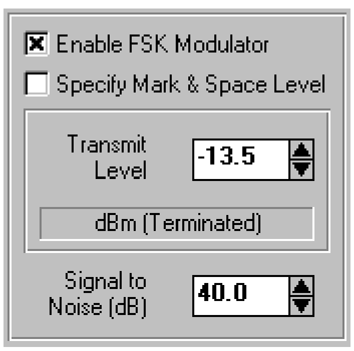
The FSK modulator used to send the Caller ID data to the CPE under test, can be enabled or disabled by clicking on the "Enable FSK Modulator" check box in the "Main Settings" window. If disabled, the operation of a Caller ID transmission will proceed exactly as if the FSK modulator is enabled, except for the lack of an FSK modulated signal sent to the CPE. The timing of any ringing bursts, SAS/CAS tones, line reversals, and alert tones are not effected by the status of the FSK modulator. The programmed noise level or interfering tone level during the FSK portion of the Caller ID transmission will also be unaffected if the FSK modulator is disabled. Disabling the FSK modulator can be useful in testing the CPE's response to calls without the associated Caller ID information being sent.

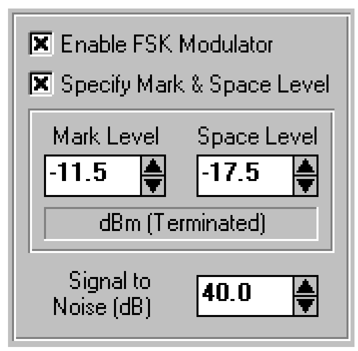
If the FSK modulator is disabled, the transmit level and signal-to-noise ratio of the FSK modulator can not be changed. The default state for the FSK modulator is to be enabled. Also, upon restoring the default parameters or changing operational standards, the FSK modulator will be re-enabled, if previously disabled.

FSK Modulator Signal Levels

The level of the FSK modulated data signal can be specified as either the total FSK signal level, or mark and space tone levels. Since many of the Caller ID standards state the mark and space tone levels, it may be more convenient to directly enter the levels as opposed to calculating the total level and twist level.

The FSK modulator controls, located on the Main Settings window, include a check box that, if enabled, displays the mark and space tone levels instead of the total FSK signal level. The following figures show the two possible modes of specifying the FSK signal level(s).





At anytime the total FSK transmit level is changed, the mark and space tone levels are re-calculated to reflect the new total level and maintaining the existing twist level. If the FSK twist setting is changed, the mark and space tone levels are re-calculated to reflect the new twist settings and maintaining the existing total level setting.

Changing the mark tone level will force a change of the total level and the twist setting, but maintain the space level. Likewise when changing the space level.

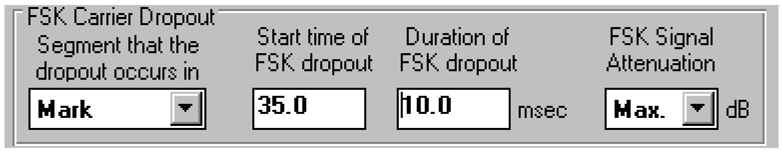
In either of the total FSK or mark/space level modes, the signal level unit is displayed just below the text boxes containing the level values. Three different unit systems can be specified under the More Options window. They are dBm (600 ohms), dBV, and mVrms. In addition, the levels can referenced to either an unterminated telephone line, or a telephone line terminated into 600 ohms. The More Options window is displayed by selecting the [CONFIGURATION] [MORE OPTIONS] menu command, or by pressing the CTRL-Z key combination.

Note: Additional FSK modulator settings, such as twist level, baud rate, mark/space tone frequencies can be viewed and changed in the Advanced Setup Window.

Controlling Signal Drop-outs of the FSK Modulator

The FSK modulator can be programmed to "drop-out" at a specified time within the FSK data stream. A "drop-out" is a period of time for which the FSK modulator produces either no output signal, or is attenuated in level. Normally the FSK modulator generates either a mark or space tone depending on the current bit value. At the time of the drop-out the FSK modulator signal level is set to zero, or attenuated, until the end of the drop-out when the level returns to its previous value. This feature is useful for testing a CPE's ability to withstand FSK drop-outs during the Preamble, or the Mark portion of a Caller ID transmission, as per Bellcore's recommendation in SR-3004. Attenuating the FSK signal is required when testing to the ETSI ETS 300 778-1 (Terminal Equipment Requirements, Part I: Off-line data transmission) standard, annex C.10 during the Caller ID message.

Programming a FSK modulator drop-out is done in the Transmission Format window. The following graphic shows how to program a drop-out during the Mark segment for a duration of 10 msec, starting 35 msec after the beginning of the Mark segment. The level of FSK attenuation can be set from 0 dB to 60 dB in 1 dB steps, in addition to a maximum attenuation setting in which the FSK modulator’s output level is set to zero.



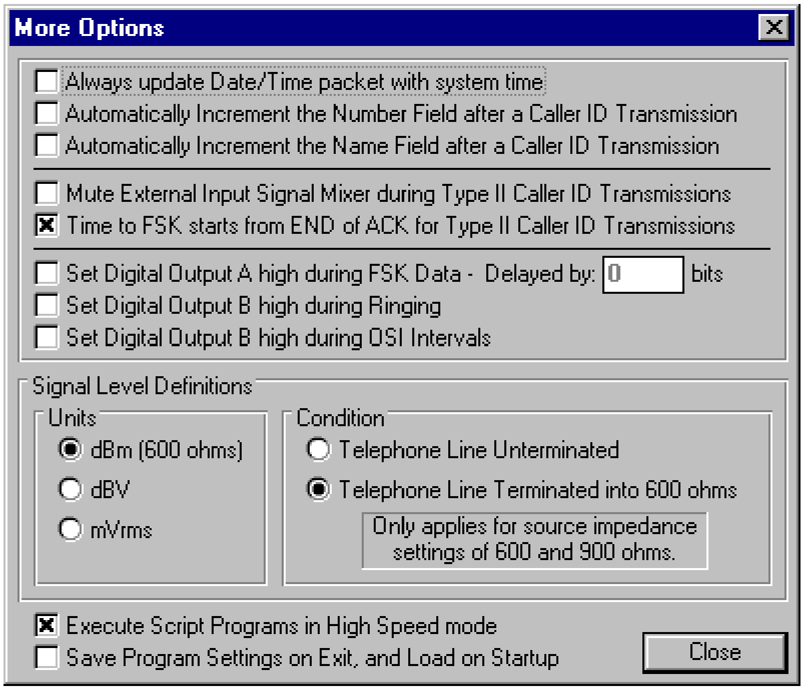
To program a specific drop-out interval, first select the segment the drop-out is to occur in from the drop down list box. The start time of the drop-out and duration of the drop-out is then entered in the two text boxes. An error message will be generated if the selected segment for a drop-out is unavailable, as is the case for the Preamble segment during Type II (CIDCW) transmissions. Also, if the start time for a drop-out exceeds the entire transmission time for that segment, an error message will be displayed, and no drop-out will occur. If the duration specified for a drop-out exceeds the length of the selected segment, the drop-out will continue into the next segment(s). In this case a warning message will be displayed, indicating that the drop-out will carry in to the following segment(s). To disable the drop-out feature, select the "(none)" segment in the drop down list box.

Note: The start time and duration of a drop-out will always be rounded to the closest bit time. As such, drop-outs will always start and end on bit boundaries. The exact start and duration times will then depend on the programmed baud rate.

Note: If any segments are selected for a FSK modulator drop-out, except for "(none)", the Caller ID transmission if be flagged as "altered" and an "A" will appear in program status line.

Continuously Updating the Date and Time

Selecting [CONFIGURATION] [MORE OPTIONS] in the program menu bar, displays various options and features that can be enabled or disabled. These include the ability to alter the Caller ID data sent to the CPE, activation of auxiliary digital output signals, and defining the system on level units to use. The first set of options deals with automatically changing the Caller ID information sent to the CPE under test.



The first option, "Always update Date/Time packet with system time", controls how the Date and Time packet is updated. With the option disabled, the Date and Time packet is updated with the current computer system time whenever the Message Type is changed, Operating Standard is changed, or when the Restore Default settings command is selected. Otherwise, the information contained in the Date/Time fields will stay the same. Enabling this option, will update the fields every minute, so that the date and time information sent with the Caller ID transmission represents the current system time.

The default condition for this option is disabled.

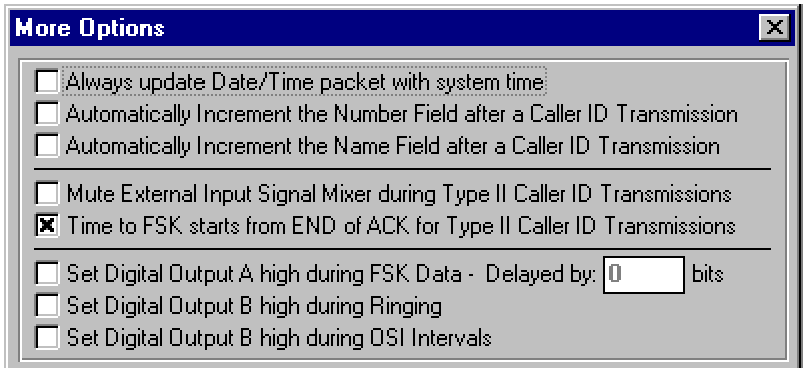
Note: The background color of the Date/Time text boxes will change to a light blue color if this feature is enabled. This serves as a reminder that the Date/Time field will be updated with the system time every minute.

Note: When the option is enabled, every minute, the Date/Time fields will be updated. If the Date/Time packet is selected as part of the Caller ID transmission, this will then force a recalculation of the entire message checksum, and the bit pattern of the Caller ID transmission. If you are planning to manipulate the Caller ID data at the byte or bit level, it is recommended to disable this option, otherwise the recalculation will over write your changes at the byte or bit level. For more information on how the program calculates the Caller ID data, see the section: Hierarchical Caller ID Data Structure

---

Enabling Automatic Name/Number Increment

Selecting [CONFIGURATION] [MORE OPTIONS] in the program menu bar, displays various options and features that can be enabled or disabled. These include the ability to alter the Caller ID data sent to the CPE, activation of auxiliary digital output signals, and defining the system on level units to use. The first set of options deals with automatically changing the Caller ID information sent to the CPE under test.



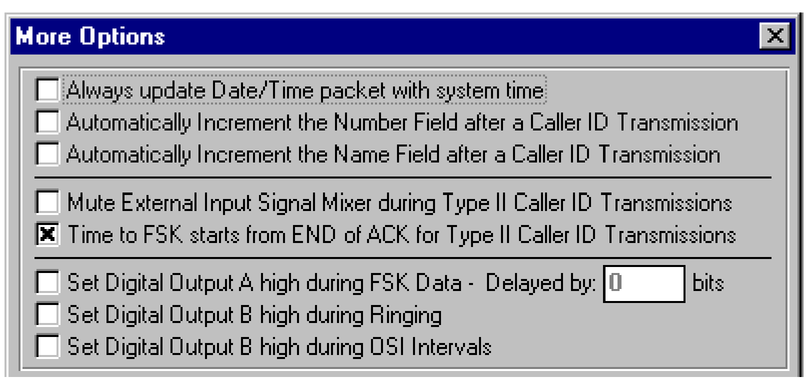
Two of the options are used to enable or disable a feature that will automatically increment the name or number text fields after a Caller ID transmission has been sent. This can be useful in testing some CPE's, where you wish to insure that the data sent to the CPE is different every time. The method of incrementing either the name or number fields is the same. The right most character is incremented by 1. If that character rolls over, then the next character to the left is incremented by 1. Any character that rolls over causes the adjacent character to the left to be incremented. Numbers between 0 and 8 will be incremented by 1, while the number 9 rolls over to 0. Letters between A to Y will be incremented by 1 (i.e. A goes to B, B goes to C,...), while the letter Z rolls over to the letter A. Characters present in the text field that are neither numbers or letters will be unaffected by the automatic increment. It should be noted that the automatic increment feature removes any trailing spaces present in the name or number fields. The default state for this option is disabled.

Note: The background color of the name and, or number text boxes will change to a light blue color if the respective option is enabled. This serves as a reminder that the field(s) will be incremented at the end of a Caller ID transmission.

Note: When the option(s) are enabled, at the end of every Caller ID transmission, the respective fields will be incremented. If the name or number packet is selected as part of the Caller ID transmission, this will then force a recalculation of the entire message checksum, and the bit pattern of the Caller ID transmission. If you are planing to manipulate the Caller ID data at the byte or bit level, it is recommended to disable this option, otherwise the recalculation will over write your changes at the byte or bit level. For more information on how the program calculates the Caller ID data, see the section: Hierarchical Caller ID Data Structure

Muting the External Input Signal Mixer

Selecting [CONFIGURATION] [MORE OPTIONS] in the program menu bar, displays various options and features that can be enabled or disabled. These include the ability to alter the Caller ID data sent to the CPE, activation of auxiliary digital output signals, and defining the system on level units to use.



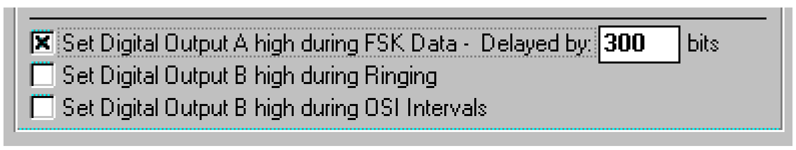
The forth option "Mute External Input Signal Mixer during Type II Caller ID Transmissions", controls the behavior of the signal mixer during Type II transmission. The external input signal mixer will allow the user to combine the signal present at the external input BNC connector, with the tones and signals generated for Caller ID transmissions. This option, when enabled, will mute the signal mixer at the start of a Type II (CIDCW) Caller ID Transmission, and then unmute at the end of the transmission. For Type II Caller ID, the central office will normally mute the "far voice" before sending the SAS and CAS tone. With this option enabled, simulated "far voice" signals can be injected into the external BNC input connector, which are passed to the CPE. Once a Type II transmission is started, the "far voice" signals will be muted, thus simulating the action of a central office switch. If the option is disabled, then the signal mixer will not mute during Type II Caller ID transmissions, or any other time.

The default state for this option is disabled.

For more information on the External Input Signal Mixer see the section: Signal Flow and Routing

Controlling the Auxiliary Digital Outputs

The More Options panel includes a feature to automatically control two digital output signals during Caller ID transmissions. The output signals, termed Output A and Output B, can be accessed from the rear DB9 connector on the TSPC. The output signals can be programmed to indicated the presence of FSK data within the Caller ID transmission, along with indicating periods of ringing or an OSI event.



By clicking the mouse at the appropriate check box above, the output A or B will output a high level when that event is active. This feature can be useful in triggering external equipment such as oscilloscopes, logic analyzers, and emulators during key times of the Caller ID transmission.

Output A, which can be enabled to be active during FSK data transmission, can also be delayed a programmable number of bits. When enabled, output A will go to a high logic level after the FSK modulator starts and returns low when the last bit has been generated. The point at which output A transitions to a high logic level can be delayed by any number of bits, while it still returns to a low level after the last data bit.

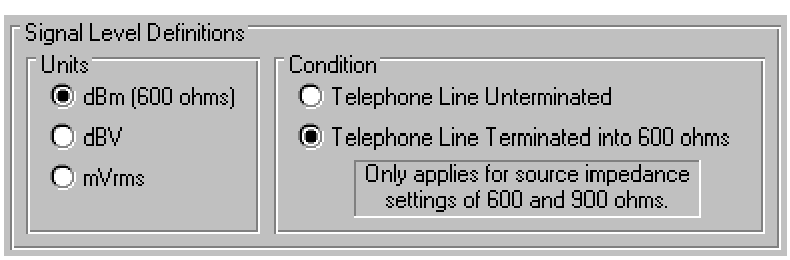
The option of using Output B to indicate OSI events can be used to overcome a hardware limitation of earlier TSPC's (revision 2.2b and older) that do not support the generation of an OSI. By using the Output B signal to control an external relay that disconnects the tip and ring leads from a CPE under test, an OSI event can be simulated by the earlier hardware.

For more information on the digital output signals for the TSPC and the pin definitions of the rear DB9 connector, see Section 5: Auxiliary Digital Inputs and Outputs.

Note: When using the AI-7280 Central Office Line Simulator, the ability to set Output A or Output B during FSK, Ringing, or OSI is not supported.

Defining the Signal Level Units

The units used throughout the program to define the various signal levels can be changed in the More Options panel under the Configuration menu. The signal levels can be specified in terms of dBm, dBV, or mVrms. Additionally, these levels can be referred to telephone lines that are either unterminated or terminated into 600 ohms.



The units of dBm specify that all signals will have their level defined relative to 1 milliwatt of power into a 600 ohm load. The dBV unit specifies levels relative to 1 Vrms, while the unit of mVrms is self-explanatory. Irrespective of what units the levels are specified in, they can refer to a telephone line terminated into 600 ohms, or an unterminated telephone line.

If the unterminated case is selected, then the level of any signal specified will only be accurate when the telephone line is unterminated. A termination will reduce the signal level by the relationship between the terminating impedance and the selected source impedance.

In the situation where the terminated option is selected, then the level of any signal specified will only be accurate when the telephone line is terminated into 600 ohms. If the telephone line is unterminated, then the measured level will be 6 dB higher if the source impedance is set to 600 ohms, or 7.8 dB high if set to 900 ohms.

Since the optional complex source impedance varies in magnitude and phase over frequency, the program can not accurately adjust the specified signal level for a terminated line. Thus, even if the “Telephone Line Terminated into 600 ohms” option is selected, it will not apply when the source impedance has be set to complex. For the complex source impedance, the level measured at the telephone line will only be accurate when the telephone line is unterminated, regardless of the setting here.

Type II Delay-to-FSK Data Synchronization

For Type II (off-hook) Caller ID transmissions, the CID1500 software is required to detect the ACK tone sent by the CPE. Following the detection of the ACK tone, the FSK data is then sent to the CPE. The time delay before sending the FSK data can be either referenced to the detection point of the ACK tone (see figure A), or to the end of the ACK tone (see figure B). In both figures, the delay to the start of the FSK data is represented by the time interval T4.

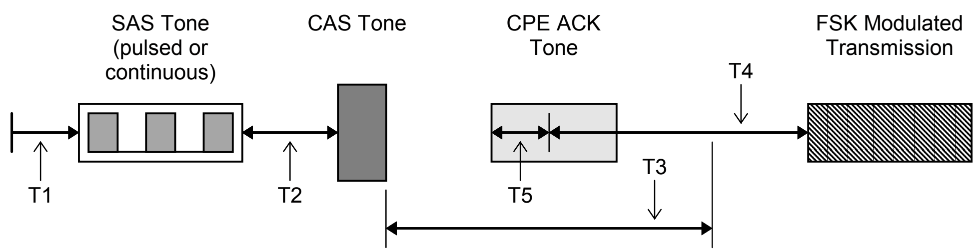


Figure A: Time delay to FSK data referenced to ACK detection

The ACK detection time, T5, represents a timing uncertainty, as it can vary due to various conditions. The time required for the ACK detection (T5) normally ranges from 10 to 20 msec; however, this can depend on the DC stability of the telephone line and the ACK tone purity. In situations where a CPE generates large transient signals due to a parallel set detection algorithm, or at the start of the ACK tone, the time required to properly qualify the ACK tone’s frequency and level characteristics may be extended. Additionally, an ACK tone that experiences a slow or uneven ramp-up or large frequency jitter results in longer detection intervals.

The timing uncertainty introduced by the ACK detection can become a problem when testing to various standards such as the TIA-777 (Type 2 Caller Identity Equipment Performance Requirements). In the above mentioned document, the uncertain T5 delay can cause difficulties in verifying CPE conformance. As such, the CID1500 software contains an option which changes the reference point for the time delay to FSK data. If enabled, the T4 delay is referenced from the end of the ACK tone, as shown below in figure B.

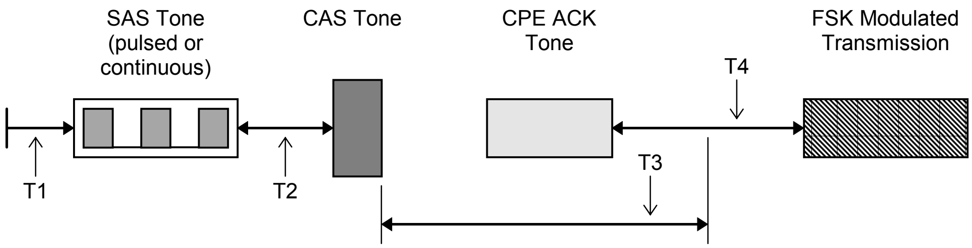
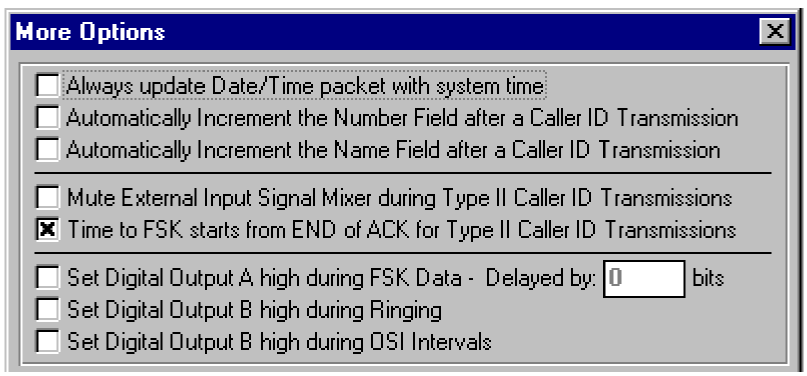


Figure B: Time delay to FSK data referenced to end of ACK

In this mode of operation, the CID1500 must still detect the ACK tone within the time-out interval of T3 by qualifying the ACK tone's level and frequency characteristics. However, once detected, the time delay to the start of the FSK data will not begin until the ACK tone has ended. It will be deemed ended when either of the low or high group tone level falls below half the power level measured at the detection point.

The timing mode used is shown in the More Options window. If the check box labeled "Time to FSK starts from END of ACK for Type II Caller ID Transmissions" is checked, then the time delay T4 will start at the end of the ACK tone instead of when the ACK tone is detected.



The More Options window can be viewed by selecting the [CONFIGURATION] [MORE OPTIONS] menu command, or by pressing the CTRL-Z key combination. The default setting is enabled.

■ Section 3-3 Setting The Signaling Type

The signaling type defines the sequence of events that forms a complete Caller ID transmission. This can include ringing before or after the FSK data is sent, or the generation of special alerting signals. The choices for the various signaling methods depends on the current operational Caller ID standard. For the Bellcore standard, four signaling methods are defined. Three for Type I (CID) transmissions, and one for Type II (CIDCW) transmissions. The ETSI standard defines eight different signaling types. Seven for Type I (CID) and one for Type II (CIDCW) transmission. If the Australian standard is selected, then seven Type I and one Type II signaling methods are available. The following list shows all of the available signaling options.

Bellcore: Type I (on-hook)
- Transmit data after first ring (during long silence interval)
- OSI Alert (no ringing)
- Transmit data without any ringing

Bellcore: Type II (off-hook)
- Send SAS and CAS tone, wait for returning ACK, then transmit data

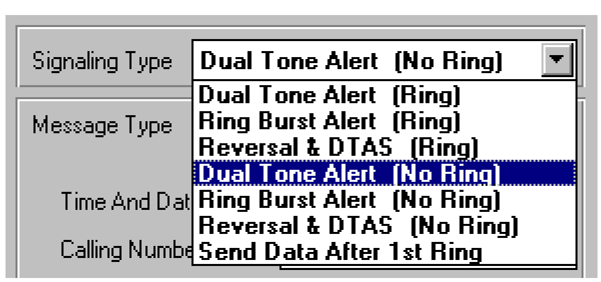
ETSI: Type I (on-hook)
- Transmit data after Dual Tone Alert Signal (DTAS), followed by ringing
- Transmit data after line reversal and DTAS, followed by ringing
- Transmit data after a single ringing burst, followed by ringing
- Transmit data after first ring (during long silence interval)
- Transmit data after DTAS, with no subsequent ringing
- Transmit data after line reversal and DTAS, with no subsequent ringing
- Transmit data after a single ringing burst, with no subsequent ringing

ETSI: Type II (off-hook)
- Send DTAS, wait for returning ACK, then transmit data

Australia: Type I (on-hook)
- Transmit data after a single ringing burst, followed by ringing
- Transmit data after an OSI, followed by ringing
- Transmit data after a line reversal, followed by ringing
- Transmit data immediately, followed by ringing
- Transmit data after an OSI, with no subsequent ringing
- Transmit data after a line reversal, followed by a 2nd line reversal
- Transmit data with no pre-data or post-data events

Australia: Type II (off-hook)
- Send SAS and CAS tone, wait for returning ACK, then transmit data

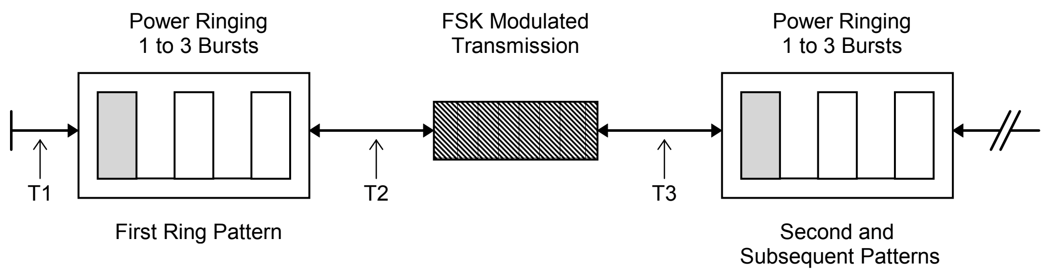
The signaling type used by the CID1500 program is selected by the Signaling Type drop-down list box shown on the Main Settings window, as shown below.



The list of available signaling options depends on the selected standard, and the state of the CPE's hook switch. If the CPE is on-hook, then only the Type I signaling options will be shown. In the case of an off-hook CPE, then the Type II options are displayed.

Bellcore (Type I): Send Data After 1st Ring

For the Bellcore Type I (CID) standard, the FSK data is sent after the first ringing pattern as shown in the figure below. The ringing pattern sent depends on the state of the ring generator parameters. This pattern can consist of a single burst, two bursts, or three bursts. This is set independently of the signaling type.



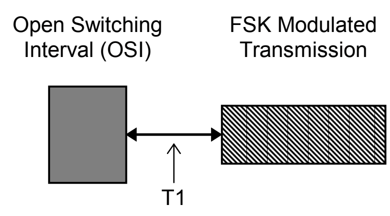
The time intervals T1, T2, and T3 can be adjusted in order to create a wide range of possible test conditions. T1 refers to the point in time when the Caller ID transmission has been started and the beginning of the 1st ringing pattern. Its value is nominally zero. T2 defines the interval from the end of the ringing burst to the start of the FSK data transmission, and T3 gives the time to subsequent ringing. However if the number of ringing cycles has been set to 1, then no subsequent rings will be generated. The parameter names that correspond to the time intervals are as follows:

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Ring*	*0 msec*
T2	*CID Timing*	*Time to Data*	*500 msec*
T3	*n/a*	*n/a*	*n/a*

Time interval T3 can not be directly set, as it is computed by the program as the long silence interval time minus T2 and the duration of the FSK transmission data. If the duration of the FSK transmission exceeds the long silence interval, the start of the second ringing pattern will be pushed back in time and start immediately once the FSK transmission is complete.

Bellcore (Type I): Transmit data after an OSI

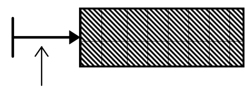
By selecting the “OSI Alert (No Ring)” signaling method, in the Main Settings windows, the FSK modulated data is sent following an OSI. The OSI represents a period of time in which the DC feeding voltage is removed from the tip and ring leads. The duration of this time period is programmable under the Advanced Setting windows with a nominal value of 250 msec. Following a time delay after the OSI (T1), the FSK data transmission begins. The default delay value for T1 is 500 msec, which is programmable in the Advanced Settings window. No ringing will follow the FSK data with this signaling method.



	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Data*	*500 msec*

Bellcore (Type I): Send Data without Ringing

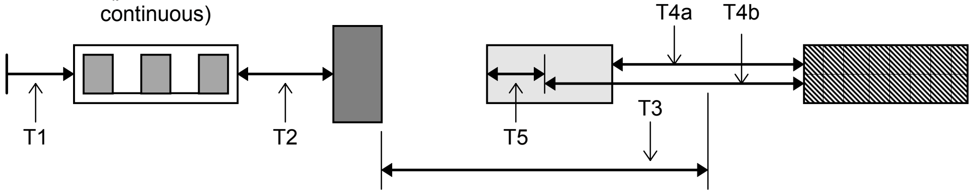
If this signaling type is selected, the FSK data will be sent without any ringing. The only timing parameter that can be adjusted is T1, which represents the time to data transmission.



	Parameter Category	Parameter Name	Default Value
T1	*CID Timing*	*Time to Data*	*500 msec*

Bellcore (Type II): Send SAS/CAS tone, wait for ACK

For Type II (CIDCW) Caller ID transmissions, only one signaling method is available. This involves sending a SAS tone, followed by the CAS tone. Once the CAS tone has been sent, the program waits for a valid ACK tone to be returned by the CPE. If this tone is detected in time and meets the criteria specified, the FSK data will be transmitted. The basic sequence of events is shown in the figure below.



T1 represents the time from the start of the Caller ID transmission to the beginning of the SAS tone. The SAS tone can be programmed to be continuous or pulsed. Once the SAS is completed, T2 defines the interval to the beginning of the CAS tone. Once the CAS tone has been generated, the CPE ACK tone must be received within the time T3. T5 represents the time required by the program to accurately measure both the level and frequency components of the ACK tone. This can vary between the approximate range of 10 to 20 msec. If the ACK tone is detected and the level and frequency components meet the specified limits, and the timer interval T3 has not expired, then the FSK data will be sent. The time delay to the FSK data is represented by either T4a or T4b depending on the timing selection displayed by the More Options window.

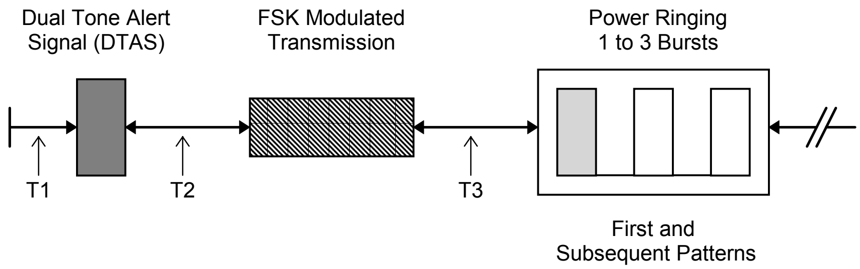
	Parameter Category	Parameter Name	Default Value
T1	*CIDCW Timing*	*Time to SAS Tone*	*0 msec*
T2	*CIDCW Timing*	*Time to CAS Tone*	*25 msec*
T3	*CIDCW Timing*	*Timeout for ACK*	*165 msec*
T4	*CIDCW Timing*	*Time to Data*	*250 msec*
T5	*n/a*	*n/a*	*10 to 20 msec*

Note: It is possible to allow the FSK transmission to commence even if the ACK tone was not detect. This is done by enabling the parameter “Tx Even After Timeout”.

ETSI (Type I): Transmit data after Dual Tone Alert Signal

The ETSI standard allows for more signaling options for type I Caller ID transmissions than the Bellcore standard. This particular signaling type uses a special alerting signal that precedes the FSK data. The dual tone alerting signal (DTAS) can consist of either one or two tones, which can be programmed to arbitrary frequency and level. Following the DTAS signal, the FSK modulated data is sent. Following the FSK data, a ringing signal can be generated depending on which of the two

“Transmit data after DTAS” signaling types is selected. One will enable the ringing generator, while the other does not.

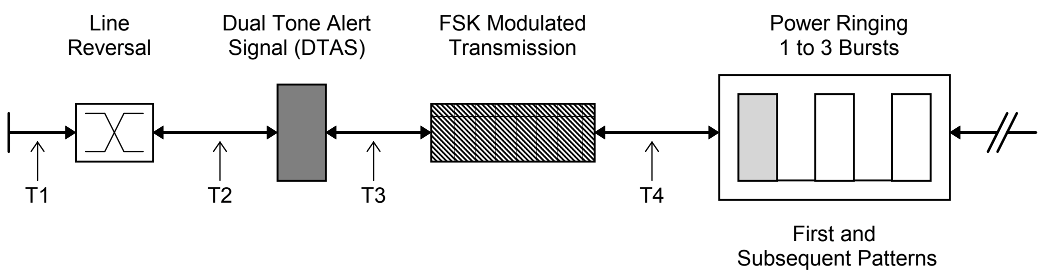


The time interval T1 represents the beginning of the Caller ID transmission and the generation of the DTAS tone. The length of the DTAS tone is independently set. Following the DTAS, is the interval T2 which precedes the FSK data transmission. Once the FSK data has been sent T3 defines the time to ringing, provided it has been enabled. The parameter category, name, and default value for each of these parameters is as follows:

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Dual Tone Alert*	*0 msec*
T2	*CID Timing*	*Time to Data Transmit*	*250 msec*
T3	*CID Timing*	*Time to Ringing*	*350 msec*

ETSI (Type I): Transmit data after Line Reversal & DTAS

A variation of the above signaling type will precede the DTAS signal with a telephone line polarity reversal. Other than the added line reversal, the sequence of events remains the same. As with the previous signaling type, the ringing after the data transmission may be enabled or disabled depending on which to the two signaling types has been selected.

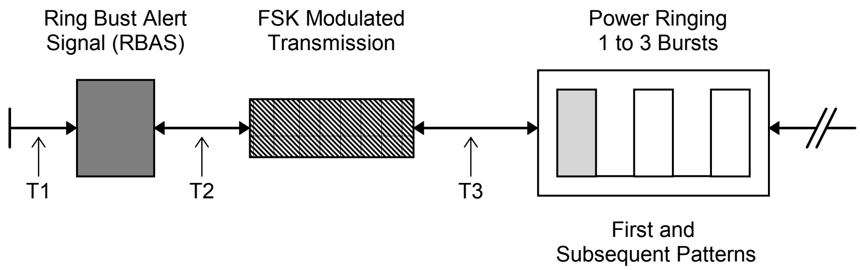


T1 is the time delay before the reversal of the telephone line, while T2 is the time interval between the line reversal and the generation of the DTAS tone. The interval between the DTAS and the FSK data is defined by T3. If ringing has been enabled, then T4 specifies the delay between the FSK data and the start of ringing.

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Reversal*	*0 msec*
T2	*CID Timing*	*Time to Dual Tone Alert*	*150 msec*
T3	*CID Timing*	*Time to Data Transmit*	*250 msec*
T4	*CID Timing*	*Time to Ringing*	*350 msec*

ETSI (Type I): Transmit data after Ring Burst Alert Signal

An alternate signaling method uses a short ringing burst instead of the low level dual tone alerting signal (DTAS). The duration, level, and frequency of the ring burst can be set independently of the normal ringing parameters. The ring burst alerting signal (RBAS) is generated before the transmission of the FSK data. Following the FSK data, a ringing signal can be enabled or disabled. This depends on which of the two “Transmit data after RBAS” signaling types is selected. One will enable the ringing generator, while the other does not.

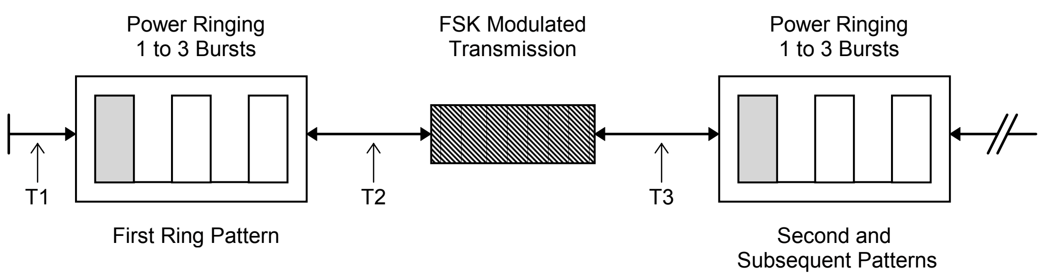


The time interval T1 represents the beginning of the Caller ID transmission and the generation of the RBAS tone. The length of the RBAS tone is independently set. Following the RBAS, is interval T2 which precedes the FSK data transmission. Once the FSK data has been sent T3 defines the time to ringing, provided it has been enabled. The parameter category, name, and default value for each of these parameters is as follows:

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Ring Burst Alert*	*0 msec*
T2	*CID Timing*	*Time to Data Transmit*	*650 msec*
T3	*CID Timing*	*Time to Ringing*	*350 msec*

ETSI (Type I): Send Data After 1st Ring

Like the Bellcore signaling type of the same name, the FSK data is sent after the first ringing pattern as shown in the figure below. The ringing pattern sent depends on the state of the ring generator parameters. This pattern can consist of a single burst, two bursts, or three bursts. This is set independently of the signaling type.



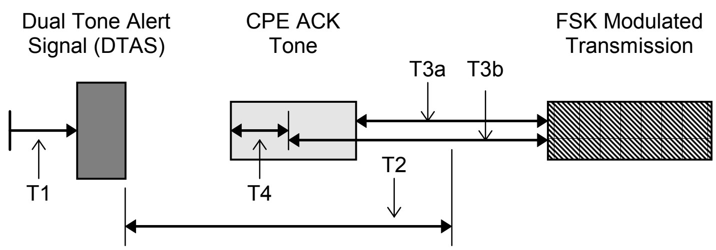
The time intervals T1, T2, and T3 can be adjusted in order to create a wide range of possible test conditions. T1 refers to the point in time when the Caller ID transmission has been started and the beginning of the 1st ringing pattern. Its value is nominally zero. T2 defines the interval from the end of the ringing burst to the start of the FSK data transmission, and T3 gives the time to subsequent ringing. However if the number of ringing cycles has been set to 1, then no subsequent rings will be generated. The parameter names that correspond to the time intervals are as follows:

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Ring*	*0 msec*
T2	*CID Timing*	*Time to Data*	*250 msec*
T3	*n/a*	*n/a*	*n/a*

Time interval T3 can not be directly set, as it is computed by the program as the long silence interval time minus T2 and the duration of the FSK transmission data. If the duration of the FSK transmission exceeds the long silence interval, the start of the second ringing pattern will be pushed back in time and start immediately once the FSK transmission is complete.

ETSI (Type II): Send DTAS tone, wait for ACK

Similar to the Bellcore standard, only one signaling method is available for Type II (CIDCW) Caller ID transmissions. This involves generating a DTAS tone and waiting for the CPE to respond with a valid ACK tone. If this tone is detected in time and meets the criteria specified, the FSK data will be transmitted. The basic sequence of events is shown in the figure below.



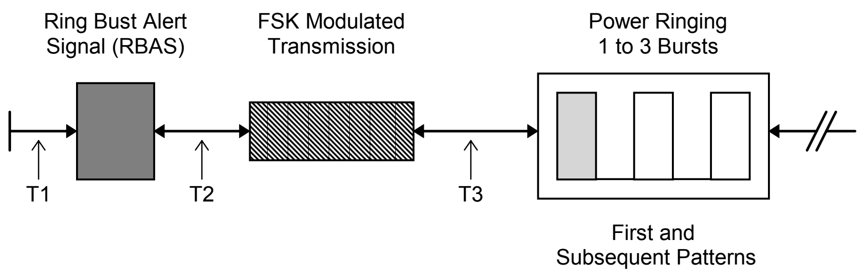
T1 represents the time from the start of the Caller ID transmission to the beginning of the DTAS tone. Once the DTAS tone has been generated, the CPE ACK tone must be received within the time T2. T4 represents the time required by the program to accurately measure both the level and frequency components of the ACK tone. This can vary between the approximate range of 10 to 20 msec. If the ACK tone is detected and the level and frequency components meet the specified limits, and the timer interval T2 has not expired, then the FSK data will be sent. The time delay to the FSK data is represented by either T3a or T3b depending on the timing selection displayed by the More Options window.

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CIDCW Timing*	*Time to DTAS Tone*	*0 msec*
T2	*CIDCW Timing*	*Timeout for ACK*	*165 msec*
T3	*CIDCW Timing*	*Time to Data*	*250 msec*
T4	*n/a*	*n/a*	*10 to 20 msec*

Note: It is possible to allow the FSK transmission to commence even if the ACK tone was not detect. This is done by enabling the parameter “Tx Even After Timeout”.

Australia (Type I): Transmit data after Ring Burst

This signaling method is virtually identical to the ETSI method of using a ringing burst prior to sending the FSK data. The duration, level, and frequency of the ring burst can be set independently of the normal ringing parameters. The ring burst alerting signal (RBAS) is generated before the transmission of the FSK data. Following the FSK data, a ringing pattern is generated that can be composed from up to three ringing bursts per cycle.

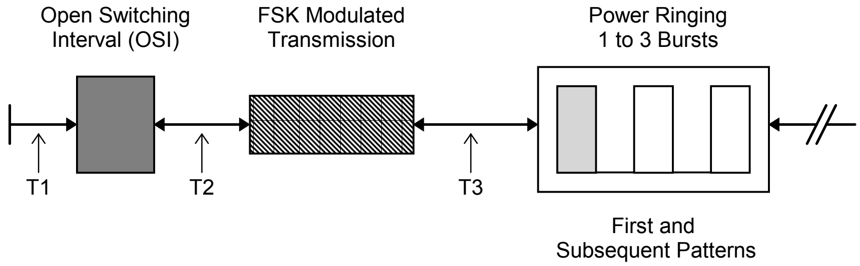


The time interval T1 represents the beginning of the Caller ID transmission and the generation of the RBAS tone. The length, frequency, and level of the RBAS tone is independently set. Following the RBAS, is interval T2 which precedes the FSK data transmission. Once the FSK data has been sent T3 defines the time to ringing. The parameter category, name, and default value for each of these parameters is as follows:

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Ring Burst Alert*	*0 msec*
T2	*CID Timing*	*Time to Data Transmit*	*700 msec*
T3	*CID Timing*	*Time to Ringing*	*500 msec*

Australia (Type I): Transmit data after an OSI

As opposed to generating a ringing burst prior to the FSK data transmission, this signaling method causes an open switching interval (OSI) to precede the FSK data. Following the FSK data, a ringing signal can be enabled or disabled. This depends on which of the two “Transmit data after OSI” signaling types is selected. One will enable the ringing generator, while the other does not.



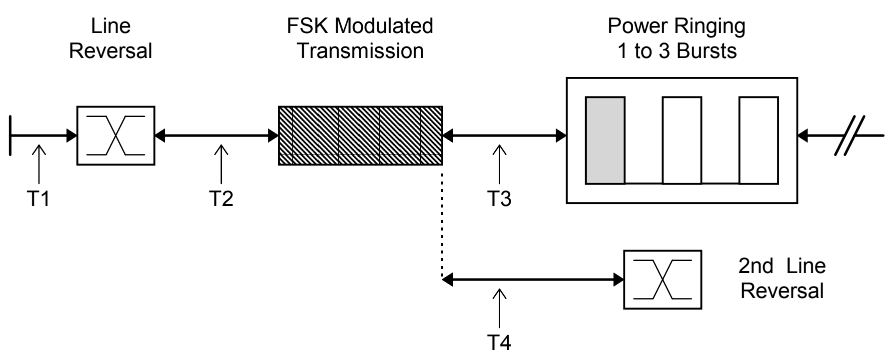
The time interval T1 represents the beginning of the Caller ID transmission and the generation of the OSI. The length of the OSI can be set to any arbitrary time interval. Following the OSI, is interval T2 which precedes the FSK data transmission. Once the FSK data has been sent T3 defines the time to ringing, provided it has been enabled. The parameter category, name, and default value for each of these parameters is as follows:

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to OSI*	*0 msec*
T2	*CID Timing*	*Time to Data Transmit*	*650 msec*
T3	*CID Timing*	*Time to Ringing*	*350 msec*

Note: Early TSPC’s (revision 2.2b and older) did not have the capability to generate an OSI event. However, by using an external relay controlled by one of the auxiliary digital outputs, an OSI can be created. See the section: Auxiliary Digital Inputs and Outputs for more information.

Australia (Type I): Transmit data after Line Reversal

This signaling method uses a telephone line polarity reversal before sending the FSK data. After the FSK data has been sent, either power ringing will begin, if enabled, otherwise a 2nd polarity reversal will occur. The following figure outlines the events for this signaling type, with and without ringing enabled.

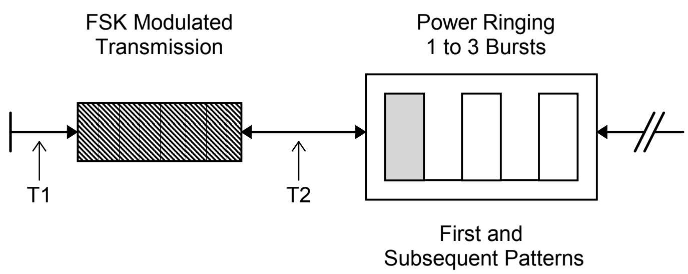


T1 is the time delay before the reversal of the telephone line, while T2 is the time interval between the line reversal and the FSK data. If ringing has been enabled, then T3 specifies the delay between the FSK data and the start of ringing. Otherwise, a second line reversal occurs at a time interval T4 after the FSK data has been sent.

	Parameter Category	***Parameter Name***	***Default Value***
T1	*CID Timing*	*Time to Reversal*	*0 msec*
T2	*CID Timing*	*Time to Data Transmit*	*700 msec*
T3	*CID Timing*	*Time to Ringing*	*500 msec*
T4	*CID Timing*	*Time to Post Reversal*	*500 msec*

Australia (Type I): Transmit data with no Alerting Signal

The following signaling method immediately generates the FSK data without any type of alerting signal to the CPE. Following the FSK data, a ringing signal can be enabled or disabled. If the "No Alert (Ring)" signaling type has been selected, ringing will be generated after the FSK data. Selecting "No Alert (No Ring)" prevents the ringing and ends the Caller ID transmission once the FSK data has been sent.

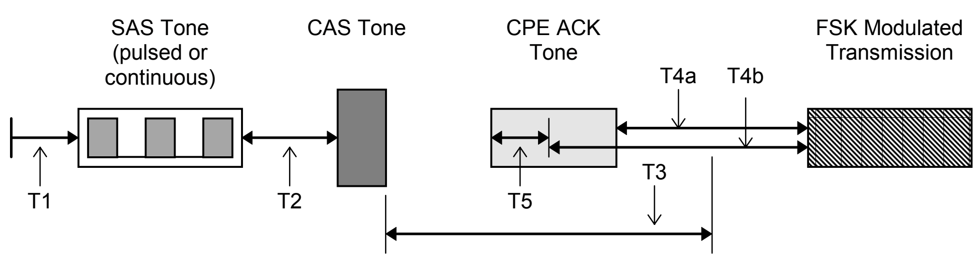


The time interval T1 represents the beginning of the Caller ID transmission and the start of the FSK data transmission. If ringing has been enabled, then following the time interval T2, the ring generator will be started. The parameter category, name, and default value for each of these parameters is as follows:

	Parameter Category	*Parameter Name*	*Default Value*
T1	*CID Timing*	*Time to Data Transmit*	*700 msec*
T2	*CID Timing*	*Time to Ringing*	*500 msec*

Australia (Type II): Send SAS/CAS tone, wait for ACK

For Type II (CIDCW) Caller ID transmissions, only one signaling method is available. This involves sending a SAS tone, followed by the CAS tone. Once the CAS tone has been sent, the program waits for a valid ACK tone to be returned by the CPE. If this tone is detected in time and meets the criteria specified, the FSK data will be transmitted. The basic sequence of events is shown in the figure below.



T1 represents the time from the start of the Caller ID transmission to the beginning of the SAS tone. The SAS tone can be programmed to be continuous or pulsed. Once the SAS is completed, T2 defines the interval to the beginning of the CAS tone. Once the CAS tone has been generated, the CPE ACK tone must be received within the time T3. T5 represents the time required by the program to accurately measure both the level and frequency components of the ACK tone. This can vary between the approximate range of 10 to 20 msec. If the ACK tone is detected and the level and frequency components meet the specified limits, and the timer interval T3 has not expired, then the FSK data will be sent. The time delay to the FSK data is represented by either T4a or T4b depending on the timing selection displayed by the More Options window.

	Parameter Category	*Parameter Name*	*Default Value*
T1	*CIDCW Timing*	*Time to SAS Tone*	*0 msec*
T2	*CIDCW Timing*	*Time to CAS Tone*	*25 msec*
T3	*CIDCW Timing*	*Timeout for ACK*	*165 msec*
T4	*CIDCW Timing*	*Time to Data*	*100 msec*
T5	*n/a*	*n/a*	*10 to 20 msec*

Note: It is possible to allow the FSK transmission to commence even if the ACK tone was not detect. This is done by enabling the parameter "Tx Even After Timeout".

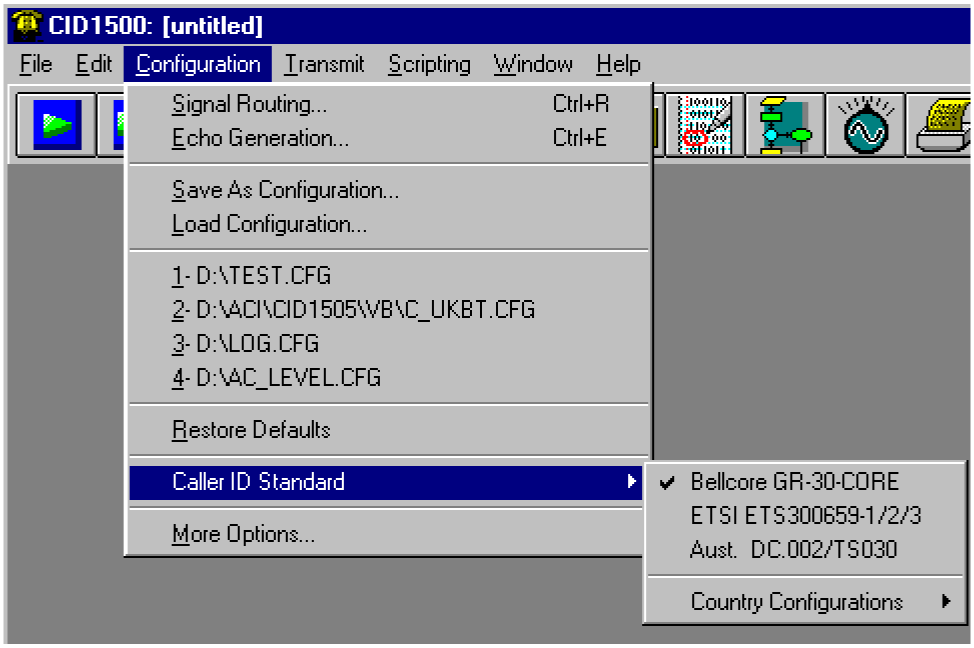
■ Section 3-4 Modifying the Message to Send

This section deals with controlling and changing the messages that are sent to the CPE under test. There are six main aspects to the program that effect the Caller ID message that is sent. They are:

1) The operational standard that is currently selected
2) The transmission mode in use
3) The message type selected
4) The contents of the data packets
5) How to edit the data packets, and
6) How to edit the segments

Setting the Caller ID Standard

Selecting the desired Caller ID standard should be the first action taken when testing various Caller ID devices. Changing the standard will cause all the parameters to return to the default value associated with the selected standard. Likewise all message data packets and segments will be reset to default values. Since the script file and data log file will be cleared of any contents it is important to save this information before the standard is changed. The standard currently selected is displayed in the Main Settings window.

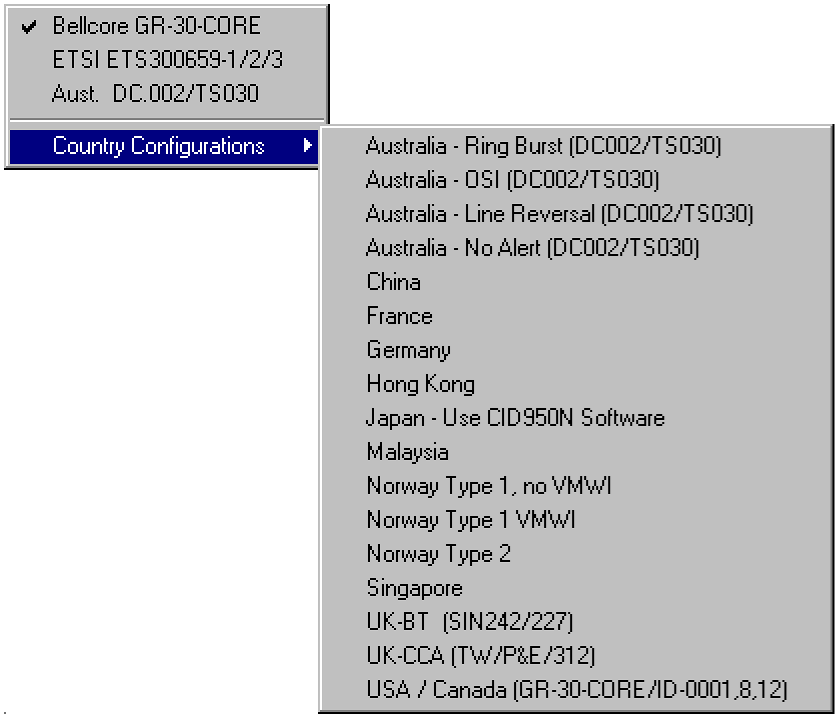


Changing the standard is simply accomplished by selecting the Caller ID Standard option under the Configuration menu, as shown above. The three different standards that can currently be selected are Bellcore GR-30-CORE, ETSI (European Telecommunications Standards Institute) ETS 300659-1/2/3, and the Australian Caller ID System based upon DC.002 and TS 030 (1997).

The standard currently selected is displayed in the Main Settings window.

Predefined Country Configuration Files

To simplify setting up the CID1500 program to the Caller ID systems used by various countries, a number of pre-defined configuration files have been included. These configurations are loaded by selecting the [CONFIGURATION] [CALLER ID STANDARD] menu. All of the country configurations use either the Bellcore, ETSI, or Australian standards as a basis, and modify (if any) the signaling system used, along with various other parameter settings. Since the Japanese NTT Caller ID system is significantly different from most other FSK based Caller ID standards, a separate program has been developed for its support. The CID950N program can be downloaded from our web site for users requiring NTT support.



Note: The CID1500 program remembers the selected standard between sessions, such that, when the program is started at a future time, the program will automatically reload the last standard selected.

Reference Documents:

Bellcore:
GR-30-CORE, Issue 1, December 1994
TR-NWT-000031, Issue 4, December 1992
TR-NWT-000575, Issue 1, October 1992
TR-NWT-001188, Issue 1, December 1991
TR-NWT-001401, Issue 1, September 1993
SR-TSV-002476, Issue 1, December 1992
SR-3004, Issue 2, January 1995

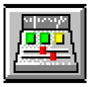
ETSI:
ETS 300 659-1, December 1996
ETS 300 659-2, September 1996
Draft EN 300 659-3, October 2000 (v1.3.1)

Australia:
DC.001, Issue 4, July 1997
DC.002, Issue 2, July 1997
DC.017, Issue 1, June 1997
DC.018, Issue 1, July 1997
TS 030, 1997

Other related documents that might be useful include:

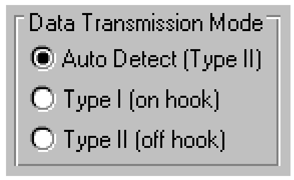
British Telecom:
Suppliers Information Note: SIN242 Issue 2, November 1996
Suppliers Information Note: SIN227 Issue 2, April 1996

Cable Communications Association (CCA)
TW/P&E/312, Issue 4, April 1997



Setting the Transmission Mode

The Transmission Mode sets whether the Caller ID transmission will be sent with the telephone in the on hook or off hook state. The options available will depend on the standard currently being used. Changing the operational mode is as simple as clicking on the desired selection in the Main Settings window.



The Type I mode refers to Caller ID transmissions with the telephone in the on hook state, and usually accompanied with power ringing. Type II mode refers to Caller ID with Call Waiting (CIDCW) operation, with the telephone in the off hook state. If the Auto Detect mode has been enabled, the operational mode will change depending on the current state of the hook switch. If on hook, Type I Caller ID transmissions will be sent. If off hook, Type II (CIDCW) transmissions will be sent.

If selecting the Type I (on-mode) of operation and the connected CPE is in the off-hook state, a warning message is displayed and the Caller ID transmission will be canceled. However, the CID1500 program will allow Type II Caller ID transmissions even if the connected CPE is in the on-hook state. This ability can become useful during the development stages of Type II capable CPE devices. To send on-hook Type II Caller ID transmissions, the Data Transmission Mode selection in the Main Settings window must be set to “Type II (off-hook)”. If a Caller ID transmission is then started and the connected CPE is on-hook, a message is displayed warning the user of the CPE on-hook state. At this point, the Caller ID transmission can be aborted or continued in the on-hook state. Note that if a Caller ID transmission is started by the script language “START” command, no warning message is displayed. If the Data Transmission Mode is set to Type II then a Caller ID transmission will be started regardless of the CPE hook switch state.

Note: When the transmission mode is changed manually, or automatically when Auto Detect is enabled, the number of bits transmitted in the Preamble and Mark segments are changed to match the specified values for the Caller ID standard currently selected. As such, if you are changing the number of bits, or editing the bits in the Preamble or Mark segments, it is best not to use the Auto Detect transmission mode. This is because if the status of the telephone hook switch changes for any reason, the Preamble and Mark segments will be recalculated, and any changes made will be lost. The re-calculation also occurs at the end of every Caller ID transmission in the Auto-Detect mode.

Setting the Message Type

What are the Message Types?

The message type refers to the different data structures used to convey the Caller ID information to the telephone under test. Both the Bellcore and ETSI standards specify data transmission via a 1200 baud FSK modulated tone using ASCII characters in a serial format. However, the structure of the ASCII characters sent defines the message type. Four message types are supported in the Bellcore standard, along with four message types for the ETSI standard.

Note: The message layer for the Australian Caller ID system is similar to that of Bellcore. As such, any reference to Bellcore message or packet types applies equally when operating under the Australian standard.

Bellcore Message Types:

Single Message
Multiple Message
Single Message Waiting
Multiple Message Waiting

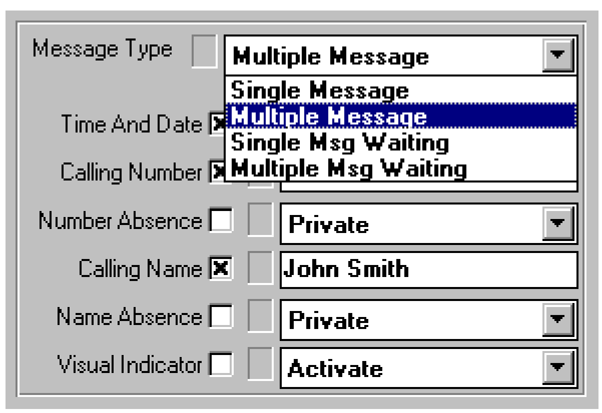
ETSI Message Types:

Call Setup
Message Waiting
Advice of Charge
Short Message Service

All of the ETSI message types along with the Bellcore Multiple Message and Multiple Message Waiting types have a very similar data structure. Only the Bellcore Single Message and Bellcore Single Message Waiting types use a significantly different data structure.

How to Change the Message Type:

The Message Type is changed in the Main Settings window. The drop down list box displays the available message types depending on the standard currently selected.



Changing the message type will reset all of the packet data (Time & Date, Number, Name, ect.) to their default values for that message. Also, some packet types are not supported with certain message types. These packets will be disabled if such a message type is selected. For example, the Bellcore Single Message format restricts the packet types to the Time & Date, Number, or Number Absence only.

Note: If a yellow "A" appears beside the Message Type drop down list box, then the current message type has been flagged as altered. This means that the message has been altered by either editing the packet data, inserting or coping new packets, or re-arranging the packet order in a manner that is not consistent with the chosen specification.

Bellcore: Single Message Format

Description:
The Single Message Format is a simple message structure that will support the transmission of date/time, and calling number/absence information only. The structure of the Single Message Format consists of a Message Type byte, Message Length byte, Message Word bytes, and then the Checksum byte. The Message Type byte is defined to be the value 0x04. The Message Length byte contains the number of Message Word bytes to follow (the Checksum byte is excluded). The first eight bytes of the Message Words will contain the date/time information. Following this is the calling number or reason for number absence data. Finally, the Checksum is the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Single Message Format in the message type drop down list on the Main Settings window enables only the Time & Date, Calling Number, and Number Absence packet checkboxes. The user has complete control to enable or disable the inclusion of all of the above mention packets. However, for a properly structured message, the Time & Date packet should be enabled along with either the Calling Number or Number Absence, but not both. The order of the data within the Message Words is always Time & Date, Calling Number, and Number Absence, if present.

The Single Message Format structure is treated as a single entity when using the Packet/Segment Edit window to alter or view its contents. This is unlike the Multiple Message Format, where each packet within the message can be altered or viewed independently.

Example:
The following example shows the contents of a Single Message Format containing the information of number 555-1212 calling on June 23rd, at 12:45 PM.

Byte Count	Description	Example Value
1:	Message Type	0x04 (only value allowed)
2:	Message Length	0x0F (15 Message Words)
3:	Message Words (month)	0x30 ASCII 0
4:	Message Words (month)	0x36 ASCII 6 (06 = June)
5:	Message Words (day)	0x32 ASCII 2
6:	Message Words (day)	0x33 ASCII 3 (23rd day)
7:	Message Words (hour)	0x31 ASCII 1
8:	Message Words (hour)	0x32 ASCII 2 (12th hour)
9:	Message Words (minute)	0x34 ASCII 4
10:	Message Words (minute)	0x35 ASCII 5 (45th minute)
11:	Message Words (number)	0x35 ASCII 5
12:	Message Words (number)	0x35 ASCII 5
13:	Message Words (number)	0x35 ASCII 5
14:	Message Words (number)	0x31 ASCII 1
15:	Message Words (number)	0x32 ASCII 2
16:	Message Words (number)	0x31 ASCII 1
17:	Message Words (number)	0x32 ASCII 2 (5551212)
18:	Checksum	0xF1

The following example shows the contents of a Single Message Format containing the information of a private number calling on October 17th, at 10:59 PM.

Byte Count	Description	Example Value
1:	Message Type	0x04 (only value allowed)
2:	Message Length	0x09 (9 Message Words)
3:	Message Words (month)	0x31 ASCII 1
4:	Message Words (month)	0x30 ASCII 0 (10 = October)
5:	Message Words (day)	0x31 ASCII 1
6:	Message Words (day)	0x37 ASCII 7 (17th day)
7:	Message Words (hour)	0x32 ASCII 2
8:	Message Words (hour)	0x32 ASCII 2 (22nd hour)
9:	Message Words (minute)	0x35 ASCII 5
10:	Message Words (minute)	0x39 ASCII 9 (59th minute)
11:	Message Words (number absence reason)	0x50 ASCII P (private number)
12:	Checksum	0x08

Bellcore: Multiple Message Format

Description:
The Multiple Message Format is a more complex structure that allows for more information to be transmitted than the Single Message Format. Like the Single Message Format structure, the Multiple Message Format consists of a Message Type byte, Message Length byte, Message Word bytes, and then the Checksum byte. The Message Type byte is defined to be the value 0x80 for call setup. The Message Length byte contains the number of Message Word bytes to follow (the Checksum byte is excluded).

Unlike the Single Message Format, the Message Words of the Multiple Message Format are structured as list of packets. Each packet contains a Packet Type byte, Packet Length byte, and Packet Word bytes. These packets can contain Time & Date, Calling Number, Number Absence, Calling Name, Name Absence, Call Redirection, Call Qualifier, and DDN information.

Following the last Packet Word of the last packet, is the Checksum byte. The Checksum is calculated as the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Multiple Message Format in the message type drop down list on the Main Settings window, enables all of the packets. Each individual packet can be included for transmission by selecting its associated checkbox. Likewise, suppressing the transmission of a packet is accomplished by de-selecting its associated checkbox. The Call Redirection, Call Qualifier, and DDN packet checkbox are located within the CID Packet Format window. Also in this window, each selected packet that will be part of the multiple message is displayed in the Packet Ordering list box. All selected packets can be viewed or altered in the Edit Packet/Segment window, by selecting the desired packet from the drop down list box.

Note: When a packet is enabled by clicking on its associated checkbox, that packet is added to the packet ordering list second from the last. The checksum packet will remain the last packet in transmission order.

Note: The Bellcore standard does not specify the ordering of packets within the message. As such, the CPE should be able to accept the packets in any order.

Example:
The following example shows the contents the packets in a Multiple Message Format if John Smith at 5551212 called at 8:31 AM on July 25th.

Multiple Message Header packet:

Byte Count	Description	Example Value
1:	Message Type	0x80
2:	Message Length	0x1F (31 Message Words)

Date & Time packet:

Byte Count	Description	Example Value
1:	Packet Type	0x01 (date & time)
2:	Packet Length	0x08 (8 Packet Words)
3:	Packet Words (month)	0x30 ASCII 0
4:	Packet Words (month)	0x37 ASCII 7 (07 = July)
5:	Packet Words (day)	0x32 ASCII 2
6:	Packet Words (day)	0x35 ASCII 5 (25th day)
7:	Packet Words (hour)	0x30 ASCII 0
8:	Packet Words (hour)	0x38 ASCII 8 (8th hour)
9:	Packet Words (minute)	0x33 ASCII 3
10:	Packet Words (minute)	0x31 ASCII 1 (31st minute)

Calling Number packet:

Byte Count	Description	Example Value
1:	Packet Type	0x02 (Calling Number)
2:	Packet Length	0x07 (7 Packet Words)
3:	Packet Words (number)	0x35 ASCII 5
4:	Packet Words (number)	0x35 ASCII 5
5:	Packet Words (number)	0x35 ASCII 5
6:	Packet Words (number)	0x31 ASCII 1
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x31 ASCII 1
9:	Packet Words (number)	0x32 ASCII 2 (number 555 1212)

Name packet:

Byte Count	Description	Example Value
1:	Packet Type	0x07 (name)
2:	Packet Length	0x0A (10 Packet Words)
3:	Packet Words (name)	0x4A ASCII J
4:	Packet Words (name)	0x6F ASCII o
5:	Packet Words (name)	0x68 ASCII h
6:	Packet Words (name)	0x6E ASCII n
7:	Packet Words (name)	0x20 ASCII (space)
8:	Packet Words (name)	0x53 ASCII S
9:	Packet Words (name)	0x6D ASCII m
10:	Packet Words (name)	0x69 ASCII i
11:	Packet Words (name)	0x74 ASCII t
12:	Packet Words (name)	0x68 ASCII h (name John Smith)

Checksum:

Byte Count	Description	Example Value
1:	Message Checksum	0x8B

Bellcore: Single Message Waiting Format

Description:
The Single Message Waiting Format can be used to send activate or deactivate message waiting commands to the CPE under test. Like the Single Message Format for calling number transmission, the Single Message Waiting Format is similar in structure. The structure of the Single Message Waiting Format consists of a Message Type byte, Message Length byte, Message Word bytes, and then the Checksum byte. The Message Type byte is defined to be the value 0x06. The Message Length byte contains the number of Message Word bytes to follow, which will always be three. The Message Words will be 0x42, 0x42, 0x42 (ASCII 'BBB'), or 0x6F, 0x6F, 0x6F (ASCII 'ooo') to activate the message waiting indicator, or to deactivate the message waiting indicator respectively. Finally, the Checksum is the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Single Message Waiting Format in the message type drop down list on the Main Settings window enables only the Visual Indicator packet checkbox. The Visual Indicator packet checkbox can not be disabled. The packet option can be selected from the drop down list box, which is either "activate" or "deactivate".

The Single Message Waiting Format structure is treated as a single entity when using the Packet/Segment Edit window to alter or view its contents. This is unlike the Multiple Message Format, where each packet within the message can be altered or viewed independently.

Example:
The following example shows the contents of a Single Message Waiting Format structure containing the activate command.

Byte Count	**Description**	**Example Value**
1:	Message Type	0x06 (only value allowed)
2:	Message Length	0x03 (3 Message Words)
3:	Message Words (activate)	0x42 ASCII B
4:	Message Words (activate)	0x42 ASCII B
5:	Message Words (activate)	0x42 ASCII B
6:	Checksum	0x31

The following example shows the contents of a Single Message Waiting Format structure containing the deactivate command.

Byte Count	Description	Example Value
1:	Message Type	0x06 (only value allowed)
2:	Message Length	0x03 (3 Message Words)
3:	Message Words (deactivate)	0x6F ASCII o
4:	*Message Words (deactivate)*	*0x6F ASCII o*
5:	*Message Words (deactivate)*	*0x6F ASCII o*
6:	*Checksum*	*0xAA*

Bellcore: Multiple Message Waiting Format

Description:
The Multiple Message Waiting Format can be used to send activate or deactivate message waiting commands to the CPE under test. Like the Multiple Message Format structure, the Multiple Message Waiting Format consists of a Message Type byte, Message Length byte, Message Word bytes, and then the Checksum byte. The Message Type byte is defined to be the value 0x82 for message waiting. The Message Length byte contains the number of Message Word bytes to follow (the Checksum byte is excluded).

The Message Words will contain a single "Visual Indicator" packet. This packet is similar to all packet structures. In that they start with a Packet Type byte, then Packet Length byte, and Packet Word bytes. The Packet Type byte is defined as 0x0B and the Packet Length will always be 1. The Packet Word byte will be 0xFF to activate message waiting or 0x00 to deactivate message waiting.

Following the Packet Word, is the Checksum byte. The Checksum is calculated as the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Multiple Message Waiting Format in the message type drop down list on the Main Settings window will enable the Visual Waiting packet check-box. The other packets will be disabled, but can be enabled by clicking the mouse on the appropriate check-box. Normally, only the Visual Waiting packet is sent to the CPE; however, there is no restriction in sending other packet types. The Visual Waiting packet option can be selected from the drop down list box, which is either "activate" or "deactivate".

Example:
The following example shows the contents of a Multiple Message Waiting Format structure containing the activate command.

Byte Count	Description	Example Value
1:	Message Type	0x82
2:	Message Length	0x03 (3 Message Words)
3:	Packet Type	0x0B Visual Indicator
4:	Packet Length	0x01
5:	Packet Word (activate)	0xFF
6:	Checksum	0x70

The following example shows the contents of a Multiple Message Waiting Format structure containing the deactivate command.

Byte Count	Description	Example Value
1:	Message Type	0x82
2:	Message Length	0x03 (3 Message Words)
3:	Packet Type	0x0B Visual Indicator
4:	Packet Length	0x01
5:	Packet Word (deactivate)	0x00
6:	Checksum	0x6F

ETSI: Call Setup Message Format

Description:
The Call Setup message format for ETSI is very similar to the Bellcore Multiple Message Format. Like the Bellcore Multiple Message Format structure, the Call Setup Message consists of a Message Type byte, Message Length byte, Message Word bytes, and then the Checksum byte. The Message Type byte is defined to be the value 0x80. The Message Length byte contains the number of Message Word bytes to follow (the Checksum byte is excluded).

The Message Words of the Call Setup Message are structured as a list of packets. Each packet contains a Packet Type byte, Packet Length byte, and Packet Word bytes. The ETSI message formats have more packet types available than that of the Bellcore message formats. The list of supported ETSI packets include Time & Date, Calling Number, Number Absence, Calling Name, Name Absence, Visual Indicator, Called Line, Call Type, Network Status, Complementary Calling Line, First Called Line, Type of Call, Type of User, Redirecting Number, and an Extension for Network Operator packet.

Following the last Packet Word of the last packet, is the Checksum byte. The Checksum is calculated as the two's complement of the sum of all the Message bytes sent.

Usage:
The Call Setup Message type is the default message type when the ETSI standard has been selected. Each individual packet can be included for transmission by selecting its associated checkbox. Likewise, suppressing the transmission of a packet is accomplished by de-selecting its associated checkbox. The Main Settings window contains most of the commonly used packet types such as Time & Date, Calling Number, and Calling Name. Many of the other ETSI packet types can be found in the CID Packet Format window. Also in this window, each selected packet that will be part of the Call Setup message is displayed in the Packet Ordering list box. All selected packets can be viewed or altered in the Edit Packet/Segment window, by selecting the desired packet from the drop down list box.

Note: When a packet is enabled by clicking on its associated checkbox, that packet is added to the packet ordering list second from the last. The checksum packet will remain the last packet in transmission order.

Note: The ETSI standard does not specify the ordering of packets within the message. As such, the CPE should be able to accept the packets in any order.

Example:
The following example shows the contents the packets in a Call Setup message if John Bull at 071 250 7587 called at 8:31 AM on July 25th.

Message Header packet:

Byte Count	Description	Example Value
1:	Message Type	0x80
2:	Message Length	0x23 (35 Message Words)

Date & Time packet:

Byte Count	Description	Example Value
1:	Packet Type	0x01 (date & time)
2:	Packet Length	0x08 (8 Packet Words)
3:	Packet Words (month)	0x30 ASCII 0
4:	Packet Words (month)	0x37 ASCII 7 (07 = July)
5:	Packet Words (day)	0x32 ASCII 2
6:	Packet Words (day)	0x35 ASCII 5 (25th day)
7:	Packet Words (hour)	0x30 ASCII 0
8:	Packet Words (hour)	0x38 ASCII 8 (8th hour)
9:	Packet Words (minute)	0x33 ASCII 3
10:	Packet Words (minute)	0x31 ASCII 1 (31st minute)

Calling Number packet:

Byte Count	Description	Example Value
1:	Packet Type	0x02 (Calling Number)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words (number)	0x30 ASCII 0
4:	Packet Words (number)	0x37 ASCII 7
5:	Packet Words (number)	0x31 ASCII 1
6:	Packet Words (number)	0x20 ASCII (space)
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x35 ASCII 5
9:	Packet Words (number)	0x30 ASCII 0
10:	Packet Words (number)	0x20 ASCII (space)
11:	Packet Words (number)	0x37 ASCII 7
12:	Packet Words (number)	0x35 ASCII 5
13:	Packet Words (number)	0x38 ASCII 8
14:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Name packet:

Byte Count	Description	Example Value
1:	Packet Type	0x07 (name)
2:	Packet Length	0x09(9 Packet Words)
3:	Packet Words (name)	0x4A ASCII J
4:	Packet Words (name)	0x6F ASCII o
5:	Packet Words (name)	0x68 ASCII h
6:	Packet Words (name)	0x6E ASCII n
7:	Packet Words (name)	0x20 ASCII (space)
8:	Packet Words (name)	0x42 ASCII B
9:	Packet Words (name)	0x75 ASCII u
10:	Packet Words (name)	0x6C ASCII l
11:	Packet Words (name)	0x6C ASCII l (name John Bull)

Checksum:

Byte Count	Description	Example Value
1:	Message Checksum	0x15

ETSI: Message Waiting Format

Description:

The ETSI Message Waiting format is basically identical to the Bellcore Multiple Message Waiting format. This message format can be used to send activate or deactivate message waiting commands to the CPE under test. Like the Call Setup structure, the Message Waiting format consists of a Message Type byte, Message Length byte, Message Word bytes, and then the Checksum byte. The Message Type byte is defined to be the value 0x82 for message waiting. The Message Length byte contains the number of Message Word bytes to follow (the Checksum byte is excluded).

The Message Words will contain a single "Visual Indicator" packet. This packet is similar to all packet structures. In that they start with a Packet Type byte, then Packet Length byte, and Packet Word bytes. The Packet Type byte is defined as 0x0B and the Packet Length will always be 1. The Packet Word byte will be 0xFF to activate message waiting or 0x00 to deactivate message waiting.

Following the Packet Word, is the Checksum byte. The Checksum is calculated as the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Message Waiting format in the message type drop down list on the Main Settings window will enable the Visual Waiting packet check-box. The other packets will be disabled, but can be enabled by clicking the mouse on the appropriate check-box. Normally, only the Visual Waiting packet is sent to the CPE; however, there is no restriction in sending other packet types. The Visual Waiting packet option can be selected from the drop down list box, which is either "activate" or "deactivate".

Example:
The following example shows the contents of a Message Waiting format structure containing the activate command.

Byte Count	Description	Example Value
1:	Message Type	0x82
2:	Message Length	0x03 (3 Message Words)
3:	Packet Type	0x0B Visual Indicator
4:	Packet Length	0x01
5:	Packet Word (activate)	0xFF
6:	Checksum	0x70

The following example shows the contents of a Message Waiting format structure containing the deactivate command.

Byte Count	Description	Example Value
1:	Message Type	0x82
2:	Message Length	0x03 (3 Message Words)
3:	Packet Type	0x0B Visual Indicator
4:	Packet Length	0x01
5:	Packet Word (deactivate)	0x00
6:	Checksum	0x6F

ETSI: Advice of Charge Message

Description:
The ETSI Advice of Charge message is used to send information relating to the charge of the call. When using this message type, the Charge packet must be included within the message. Additional packets may be sent, including:

Date & Time
Calling Line Identity or Reason for Absence of Calling Line Identity
Called Line Identity
Complementary Calling Line Identity
Additional Charge
Duration of the Call
Network Provider Identity
Carrier Identity
Selection of Terminal Function
Display Information
Extension for Network Operator use

Like the Call Setup structure, the Advice of Charge message consists of a Message Type byte, Message Length byte, Message Word bytes, and then ending with the Checksum byte. The Message Type byte is defined as the value 0x86. The Message Length byte contains the number of Message Word bytes to follow (excluding the Checksum byte). Following the last Message Word data byte, is the Checksum byte. The Checksum is calculated as the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Advice of Charge in the message type drop down list on the Main Settings window will enable the Charge packet check-box. All of the other packets are disabled, but can be enabled by clicking the mouse on the appropriate check-box. Within the Charge packet, information regarding the currency units, type of charge, charge amount or units, along with various other flags are sent

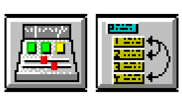
ETSI: Short Message Service

Description:
The ETSI Short Message Service is used to send short text messages to the CPE or terminal equipment. When using this message type, the Display Information packet must be included within the message. Additional packets may be sent, including:

- Date & Time
- Calling Line Identity or Reason for Absence of Calling Line Identity
- Calling Party Name or Reason for Absence of Calling Party Name
- Complementary Calling Line Identity
- Type of Calling User
- Network Provider Identity
- Selection of Terminal Function
- Service Information
- Extension for Network Operator use

Like the Call Setup structure, the Short Message Service format consists of a Message Type byte, Message Length byte, Message Word bytes, finally ending with the Checksum byte. The Message Type byte is defined as the value 0x89. The Message Length byte contains the number of Message Word bytes to follow (excluding the Checksum byte). Following the last Message Word data byte, is the Checksum byte. The Checksum is calculated as the two's complement of the sum of all the Message bytes sent.

Usage:
Selecting the Short Message Service in the message type drop down list on the Main Settings window enables the Display Information packet check-box. All of the other packets are disabled, but can be enabled by clicking the mouse on the appropriate check-box. The Display Information packet contains data regarding the type of text message being sent along with a short text message.



Setting the Packet Data to Send

What are Data Packets?
The data packets are a collection of bytes that contain the information (date & time, telephone number, name, ect.) transmitted in the Caller ID message. The Bellcore Multiple Messages, and the ETSI messages can contain more than one data packet. The list below shows the various packet types available with the Bellcore Multiple Message format, and the ETSI message formats.

The Bellcore Single Message Format and Single Message Waiting Format are treated by this program as a single data packet, as opposed to the many data packets that make up the Bellcore Multiple Message formats, and the ETSI message formats.

Note: For information on the structure of the various message types, see: Setting the Message Type

Note: The message layer for the Australian Caller ID system is similar to that of Bellcore. As such, any reference to Bellcore message or packet types applies equally when operating under the Australian standard.

Bellcore Data Packet Types:

Date & Time
Calling Line Number
Reason for Absence of Number
Calling Name
Reason for Absence of Name
Visual Indicator
Dialable Directory Number
Call Qualifier
Reason for Redirection

ETSI Packet Types:

Date & Time
Calling Line Number
Reason for Absence of Number
Calling Name
Reason for Absence of Name
Visual Indicator
Called Line Identity
Call Type
Network Message System Status
Complementary Calling Line Identity
First Called Line Identity
Type of Forwarded Call
Type of User Calling
Redirecting Number
Message Identification
Last Message CLI (Calling Line Identity)
Complementary Date and Time
Charge
Additional Charge
Duration of the Call
Network Provider Identity
Carrier Identity
Selection of Terminal Function
Display Information
Service Information
Extension for Network Operator Use

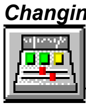
Packet Type: Date & Time

What standard does this packet apply to:

Bellcore and ETSI. Both standards use the same format for conveying the date and time information.

The Date & Time packet contains the month, day of the month, hour, and minute information of when the Caller ID transmission was sent. The information is encoded as numeric ASCII characters, with the month ranging from '00' to '12', the day of the month from '00' to '31', the hour from '00' to '23', and the minute from '00' to '59'.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the Main Settings window. If the Date & Time checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Date & Time information will not be sent. The month, day, hour, and minute information is individually entered in the four text boxes. Only two characters are allowed in each of the text boxes and the characters entered must be numeric (from '0' to '9'). Illegal times and dates can be sent by having months over 12, days over 31, hours over 23, and minutes over 59.

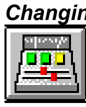
If the option "Always update Date/Time packet with system time" is enabled, the values in the Date and Time text boxes will be updated with the current system time every minute. This option can be enabled or disabled by selecting [CONFIGURATION] [MORE OPTIONS] from the menu. For more information on this option and its implications, see: Enabling Continuous Time Updates (Section 3-2).

The structure of the packet:
The example below is for the date: July 25th, 8:31 AM. The text boxes would look as follows: Month: '07', Day: '25', Hour: '08', Minute: '31'.

Byte Count	Description	Example Value
1:	Packet Type	0x01 (date & time)
2:	Packet Length	0x08 (8 Packet Words)
3:	Packet Words (month)	0x30 ASCII 0
4:	Packet Words (month)	0x37 ASCII 7 (07 = July)
5:	Packet Words (day)	0x32 ASCII 2
6:	Packet Words (day)	0x35 ASCII 5 (25th day)
7:	Packet Words (hour)	0x30 ASCII 0
8:	Packet Words (hour)	0x38 ASCII 8 (8th hour)
9:	Packet Words (minute)	0x33 ASCII 3
10:	Packet Words (minute)	0x31 ASCII 1 (31st minute)

Packet Type: Calling Number Identity

What standard does this packet apply to:
Bellcore and ETSI. Both standards use the same format for conveying the Calling Number information.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the Main Settings window. If the Calling Number checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Calling Number information will not be sent. The number sent is contained within the text box. The characters in the text box are not limited to numbers. All characters including spaces will be transmitted as part of the packet. Up to 50 characters can be entered.

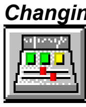
The structure of the packet:
The example below is for the phone number: 5551212. The text box would contain the following: "5551212" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0x02 (Number)
2:	Packet Length	0x07 (7 Packet Words)
3:	Packet Words (number)	0x35 ASCII 5
4:	Packet Words (number)	0x35 ASCII 5
5:	Packet Words (number)	0x35 ASCII 5
6:	Packet Words (number)	0x31 ASCII 1
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x31 ASCII 1
9:	Packet Words (number)	0x32 ASCII 2 (number 5551212)

Packet Type: Reason for Absence of Number

What standard does this packet apply to:
Bellcore and ETSI. Both standards use the same format for conveying the Reason for Absence of Number information.

Changing the packet data:


The data contained in this packet is displayed and can be changed in the Main Settings window. If the Reason for Absence of Number checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Reason for Absence of Number information will not be sent. The reason for number absence is selected from the drop down list, and can be either "Private" or "Out of Area". It is possible to enable both the number and number absence packets. This would be a very unusual condition; however, no restriction in the standards prevents its occurrence.

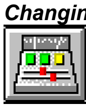
The structure of the packet:
The example below is for the Reason for Absence of Number selected as: Private. In the case of "Out of Area" being selected, the Packet Word would contain 0x4F for ASCII 'O'.

Byte Count	Description	Example Value
1:	Packet Type	0x04 (Number Absence)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word (reason)	0x50 ASCII 'P'

Packet Type: Calling Name

What standard does this packet apply to:
Bellcore and ETSI. Both standards use the same format for conveying the Calling Name information.

Changing the packet data:


The data contained in this packet is displayed and can be changed in the Main Settings window. If the Calling Name checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Calling Name information will not be sent. The calling name sent is contained within the text box. Up to 50 characters can be entered.

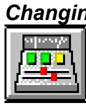
The structure of the packet:
The example below is for the phone number: John Smith. The text box would contain the following: "John Smith" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0x07 (name)
2:	Packet Length	0x0A (10 Packet Words)
3:	Packet Words (name)	0x4A ASCII J
4:	Packet Words (name)	0x6F ASCII o
5:	Packet Words (name)	0x68 ASCII h
6:	Packet Words (name)	0x6E ASCII n
7:	Packet Words (name)	0x20 ASCII (space)
8:	Packet Words (name)	0x53 ASCII S
9:	Packet Words (name)	0x6D ASCII m
10:	Packet Words (name)	0x69 ASCII i
11:	Packet Words (name)	0x74 ASCII t
12:	Packet Words (name)	0x68 ASCII h (name John Smith)

Packet Type: Reason for Absence of Name

What standard does this packet apply to:
Bellcore and ETSI. Both standards use the same format for conveying the Reason for Absence of Name information.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the Main Settings window. If the Reason for Absence of Name checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Reason for Absence of Name information will not be sent. The reason for name absence is selected from the drop down list, and can be either "Private" or "Out of Area". It is possible to enabled both the name and name absence packets. This would be a very unusual condition; however, no restriction in the standards prevents its occurrence.

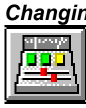
The structure of the packet:
The example below is for the Reason for Absence of Name being: Out of Area. In the case of "Private" being selected, the Packet Word would contain 0x50 for ASCII 'P'.

Byte Count	Description	Example Value
1:	Packet Type	0x08 (Name Absence)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word (reason)	0x4F ASCII 'O'

Packet Type: Visual Indicator

What standard does this packet apply to:
Bellcore and ETSI. Both standards use the same format for conveying the visual indicator information.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the Main Settings window. If the Visual Indicator checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Visual Indictor information will not be sent. The drop down list is used to select either “activate” or “deactivate” for the packets contents. Normally this packet is only sent with the message waiting formats. However, there is no restriction in sending the Visual Indicator information in the call setup messages.

The structure of the packet:
The example below shows the Visual Indicator packet when sending: Activate. In the case of “Deactivate” being selected, the packet word would contain 0x00 instead of 0xFF.

Byte Count	Description	Example Value
1:	Packet Type	0x0B (Visual Indicator)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word	0xFF (Activate)

Packet Type: Dialable Directory Number/Called Line Identity

What standard does this packet apply to:
Bellcore and ETSI. Both standards use the same format for conveying the Dialable Directory Number (DDN)/Called Line Identity information.

Changing the packet data:

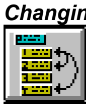
The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the DDN/Called Line checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the DDN/Called Line information will not be sent. The number sent is contained within the text box. The characters in the text box are not limited to numbers. All characters including spaces will be transmitted as part of the packet. Up to 50 characters can be entered.

The structure of the packet:
The example below is for the phone number: 071 250 7587. The text box would contain the following: "071 250 7587" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0x03 (DDN/Called Line)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words (number)	0x30 ASCII 0
4:	Packet Words (number)	0x37 ASCII 7
5:	Packet Words (number)	0x31 ASCII 1
6:	Packet Words (number)	0x20 ASCII (space)
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x35 ASCII 5
9:	Packet Words (number)	0x30 ASCII 0
10:	Packet Words (number)	0x20 ASCII (space)
11:	Packet Words (number)	0x37 ASCII 7
12:	Packet Words (number)	0x35 ASCII 5
13:	Packet Words (number)	0x38 ASCII 8
14:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Packet Type: Call Qualifier

What standard does this packet apply to:
Only the Bellcore standard.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Call Qualifier checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Call Qualifier information will not be sent. The call qualifier is selected from the drop down list and is only defined for the "Long Distance" selection.

The structure of the packet:
The example below is for the Call Qualifier selected as: Long Distance.

Byte Count	Description	Example Value
1:	Packet Type	0x06 (Call Qualifier)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word (Call Qual.)	0x4C ASCII 'L' (Long Distance)

Packet Type: Call Redirection

What standard does this packet apply to:
Only the Bellcore standard.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Redirection checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Redirection information will not be sent. The call redirection is selected from the drop down list, and can be either "Call Fwd Universal", "Call Fwd Busy", or "Call Fwd Unanswered".

The structure of the packet:
The example below is for the Call Redirection selected as: Call Fwd Universal. In the case of "Call Fwd Busy" or "Call Fwd Unanswered" being selected, the Packet Word would contain 0x01 or 0x02 respectively.

Byte Count	Description	Example Value
1:	Packet Type	0x05 (Call Redirection)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word	0x00 (Call Fwd Universal)

Packet Type: Call Type

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Call Type checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Call Type information will not be sent. The call type is selected from the drop down list, and can be any of the following:

Type of Call	Value
Voice Call	0x01
Ring back when Free	0x02
Calling Name Delivery	0x03
Call Return	0x04
Alarm Call	0x05
Download Function	0x06
Reverse Charging Call	0x07
External Call	0x10
Internal Call	0x11
Monitoring Call	0x50
Message Waiting Call	0x81

The structure of the packet:
The example below is for a Call Type of "Voice Call".

Byte Count	Description	Example Value
1:	Packet Type	0x11 (Call type)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word (Call Type)	0x01 (Voice Call)

Packet Type: Network Message System Status

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:

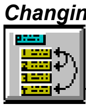
The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Network Status checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Network Status information will not be sent. The value of the network system status is a binary encoded value ranging from 0 to 255 representing the number of messages waiting in the message system. This value can be selected from the drop down list box.

The structure of the packet:
The example below is for the a Network System Status value of 1.

Byte Count	Description	Example Value
1:	Packet Type	0x13 (Network System Status)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word (Status)	0x01 (Value)

Packet Type: Complementary Calling Line Identity

What standard does this packet apply to:
Only the ETSI standard.

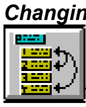
Changing the packet data:
The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Comp Calling Line checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Complementary Calling Line Identity information will not be sent. The number sent is contained within the text box. The characters in the text box are not limited to numbers. All characters including spaces will be transmitted as part of the packet. Up to 50 characters can be entered.

The structure of the packet:
The example below is for the phone number: 071 250 7587. The text box would contain the following: "071 250 7587" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0x10 (Comp Calling Line)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words (number)	0x30 ASCII 0
4:	Packet Words (number)	0x37 ASCII 7
5:	Packet Words (number)	0x31 ASCII 1
6:	Packet Words (number)	0x20 ASCII (space)
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x35 ASCII 5
9:	Packet Words (number)	0x30 ASCII 0
10:	Packet Words (number)	0x20 ASCII (space)
11:	Packet Words (number)	0x37 ASCII 7
12:	Packet Words (number)	0x35 ASCII 5
13:	Packet Words (number)	0x38 ASCII 8
14:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Packet Type: First Called Line Identity

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:
The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the First Called Line checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the First Called Line Identity information will not be sent. The number sent is contained within the text box. The characters in the text box are not limited to numbers. All characters including spaces will be transmitted as part of the packet. Up to 50 characters can be entered.

The structure of the packet:
The example below is for the phone number: 071 250 7587. The text box would contain the following: "071 250 7587" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0x12 (First Called Line)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words (number)	0x30 ASCII 0
4:	Packet Words (number)	0x37 ASCII 7
5:	Packet Words (number)	0x31 ASCII 1
6:	Packet Words (number)	0x20 ASCII (space)
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x35 ASCII 5
9:	Packet Words (number)	0x30 ASCII 0
10:	Packet Words (number)	0x20 ASCII (space)
11:	Packet Words (number)	0x37 ASCII 7
12:	Packet Words (number)	0x35 ASCII 5
13:	Packet Words (number)	0x38 ASCII 8
14:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Packet Type: Type of Forwarded Call

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:
The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Type of Forward checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Type of Forwarded Call information will not be sent. The Type of Forwarded Call is selected from the drop down list, being one of the following selections.

Unavailable or unknown forwarded call type
Forwarded call on busy
Forwarded call on no reply
Unconditional forwarded call
Deflected call after alerting
Deflected call immediate
Forwarded call on inability to reach mobile subscriber

The structure of the packet:
The example below is for the Type of Forwarded Call setting of: “Forwarded call on busy”.

Byte Count	Description	Example Value
1:	Packet Type	0x15 (Forwarded Call Type)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word	0x01 (Forwarded call on busy)

Packet Type: Type of Calling User

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:
The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Type of User checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Type of User information will not be sent. The Type of User is selected from the drop down list, being one of the following selections.

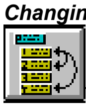
Type Calling User	Value
Unknown or Unavailable	0x00
Voice Call	0x01
Text Call	0x02
Virtual Private Network	0x03
Mobile Phone	0x04
Mobile Phone and VPN	0x05
Fax Call	0x06
Video Call	0x07
E-Mail Call	0x08
Operator Call	0x09
Ordinary Calling	0x0A
Priority Subscriber	0x0B
Data Call	0x0C
Test Call	0x0D
Telemetric Call	0x0E
Payphone	0x0F

The structure of the packet:
The example below is for the Type of User setting of: "Mobile phone and VPN".

Byte Count	Description	Example Value
1:	Packet Type	0x16 (Type of User)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word	0x05 (Mobile phone and VPN)

Packet Type: Redirecting Number

What standard does this packet apply to:
Only the ETSI standard.

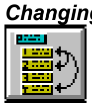
Changing the packet data:

The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Redirecting Number checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Redirecting Number information will not be sent. The number sent is contained within the text box. The characters in the text box are not limited to numbers. All characters including spaces will be transmitted as part of the packet. Up to 50 characters can be entered.

The structure of the packet:
The example below is for the phone number: 071 250 7587. The text box would contain the following: "071 250 7587" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0x1A (Redirecting Number)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words (number)	0x30 ASCII 0
4:	Packet Words (number)	0x37 ASCII 7
5:	Packet Words (number)	0x31 ASCII 1
6:	Packet Words (number)	0x20 ASCII (space)
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x35 ASCII 5
9:	Packet Words (number)	0x30 ASCII 0
10:	Packet Words (number)	0x20 ASCII (space)
11:	Packet Words (number)	0x37 ASCII 7
12:	Packet Words (number)	0x35 ASCII 5
13:	Packet Words (number)	0x38 ASCII 8
14:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Packet Type: Message Identification

What standard does this packet apply to:
Only the ETSI standard.

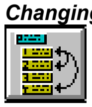
Changing the packet data:

The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Messge Ident. check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Message Identification information is not sent. The packet always sends three data bytes. The first data byte indicates the status of a message as either Removed (0x00), Added (0xFF), or sending Message Reference only (0x55). The second and third data bytes contain the message reference as a value from 0 to 65535. The most significant byte is sent before the least significant byte.

The structure of the packet:
The example below sends a message reference only with the value of 1234.

Byte Count	Description	Example Value
1:	Packet Type	0x0D (Message Identification)
2:	Packet Length	0x03 (3 Packet Words)
3:	Packet Words	0x55 (Message Reference Only)
4:	Packet Words	0x04 (MSB)
5:	Packet Words	0xD2 (LSB) 0x04D2 = 1234

Packet Type: Last Message CLI (Calling Line Identity)

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:

The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Last Message CLI check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Last Message CLI information is not sent. This packet sends the calling line identity representing the party who left the last message in the message system. The number sent is contained within the text box. The characters in the text box are not limited to numbers. All characters including spaces will be transmitted as part of the packet. Up to 50 characters can be entered, even though the standard specifies a maximum of 20 characters.

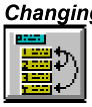
The structure of the packet:
The example below is for the phone number: 071 250 7587. The text box would contain the following: "071 250 7587" (without the quotation marks).

Byte Count	Description	Example Value
1:	Packet Type	0xOE (Last Message CLI)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words (number)	0x30 ASCII 0
4:	Packet Words (number)	0x37 ASCII 7
5:	Packet Words (number)	0x31 ASCII 1
6:	Packet Words (number)	0x20 ASCII (space)
7:	Packet Words (number)	0x32 ASCII 2
8:	Packet Words (number)	0x35 ASCII 5
9:	Packet Words (number)	0x30 ASCII 0
10:	Packet Words (number)	0x20 ASCII (space)
11:	Packet Words (number)	0x37 ASCII 7
12:	Packet Words (number)	0x35 ASCII 5
13:	Packet Words (number)	0x38 ASCII 8
14:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Packet Type: Complementary Date and Time

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:


The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Comp. Date & Time check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Complementary Date and Time information is not sent. Similar to the Date & Time packet, the Complementary Date and Time packet sends information as ASCII characters representing the month, day, hour, and minutes. Optionally, the Complementary Date and Time information can include two extra characters representing the seconds. As such, the packet length byte will either be 8 or 10 depending of whether the seconds information is sent.

Normally, the CID1500 will include the seconds in the packet. If the seconds are not to be sent, simply select the seconds text field in the Additional ETSI Packet window, and delete both characters. As long as the text field is empty, the seconds information is not sent.

The structure of the packet:
The example below is for the date: July 25th, 8:31 AM. The text boxes would look as follows: Month: '07', Day: '25', Hour: '08', Minute: '31'.

Byte Count	Description	Example Value
1:	Packet Type	0x0F (complementary date & time)
2:	Packet Length	0x08 (8 Packet Words)
3:	Packet Words (month)	0x30 ASCII 0
4:	Packet Words (month)	0x37 ASCII 7 (07 = July)
5:	Packet Words (day)	0x32 ASCII 2
6:	Packet Words (day)	0x35 ASCII 5 (25th day)
7:	Packet Words (hour)	0x30 ASCII 0
8:	Packet Words (hour)	0x38 ASCII 8 (8th hour)
9:	Packet Words (minute)	0x33 ASCII 3
10:	Packet Words (minute)	0x31 ASCII 1 (31st minute)

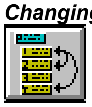
Adding a seconds value of 57 changes the data as follows:

Byte Count	Description	Example Value
1:	Packet Type	0x0F (complementary date & time)
2:	Packet Length	0x0A (10 Packet Words)
3:	Packet Words (month)	0x30 ASCII 0
4:	Packet Words (month)	0x37 ASCII 7 (07 = July)
5:	Packet Words (day)	0x32 ASCII 2
6:	Packet Words (day)	0x35 ASCII 5 (25th day)
7:	Packet Words (hour)	0x30 ASCII 0
8:	Packet Words (hour)	0x38 ASCII 8 (8th hour)
9:	Packet Words (minute)	0x33 ASCII 3
10:	Packet Words (minute)	0x31 ASCII 1 (31st minute)
11:	Packet Words (second)	0x35 ASCII 5
12:	Packet Words (second)	0x37 ASCII 7 (57th second)

Packet Type: Charge

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:


The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Charge check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Charge information is not sent. The Charge packet always consists of

16 bytes. The first and second bytes are the packet type and length respectively, and the remaining bytes (14) contain the charge information. The Charge information is broken into a number of bytes as follows:

Bytes 1 to 3:	Three ASCII characters representing the currency.			
Byte 4:	Bit 1	clear: Normal Charging		set: Free of Charge
	Bit 2:	clear: Total		set: Subtotal
	Bit 3:	clear: Normal Charging		set: Credit/Debit
	Bit 4:	clear: Information Avail.		set: No Information
	Bit 5:	clear: Currency Amt.		set: Units
	Bit 6 & 7:		00: Current Charge Call	
			01: Accumulated Charge	
			10: Extra Charge	
Bytes 5 to 14:	ASCII characters representing cost or,			
Bytes 5 to 9:	ASCII characters representing number of units			
Bytes 10 to 14:	ASCII characters representing price per unit			

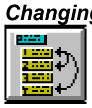
The structure of the packet:
The example below is for a charge of 23,45 FRF.

Byte Count	Description	Example Value
1:	Packet Type	0x20 (Charge)
2:	Packet Length	0x0E (14 Packet Words)
3:	Packet Words (currency)	0x46 ASCII F
4:	Packet Words (currency)	0x52 ASCII R
5:	Packet Words (currency)	0x46 ASCII F
6:	Packet Words	0x00 Total Normal Current Charge
7:	Packet Words (cost)	0x30 ASCII 0
8:	Packet Words (cost)	0x30 ASCII 0
9:	Packet Words (cost)	0x30 ASCII 0
10:	Packet Words (cost)	0x30 ASCII 0
11:	Packet Words (cost)	0x30 ASCII 0
12:	Packet Words (cost)	0x32 ASCII 2
13:	Packet Words (cost)	0x33 ASCII 3
14:	Packet Words (cost)	0x2C ASCII ,
15:	Packet Words (cost)	0x34 ASCII 4
16:	Packet Words (cost)	0x35 ASCII 5

Packet Type: Additional Charge

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:



The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Additional Charge check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Additional Charge information is not sent. The Additional Charge packet always consists of 16 bytes. The first and second bytes are the packet type and length respectively, and the remaining bytes (14) contain the charge information. The Additional Charge information is broken into a number of bytes as follows:

Bytes 1 to 3:	Three ASCII characters representing the currency.			
Byte 4:	Bit 1	clear: Normal Charging		set: Free of Charge
	Bit 2:	clear: Total		set: Subtotal
	Bit 3:	clear: Normal Charging		set: Credit/Debit
	Bit 4:	clear: Information Avail.		set: No Information
	Bit 5:	clear: Currency Amt.		set: Units
	Bit 6 & 7:		00: Current Charge Call	
			01: Accumulated Charge	
			10: Extra Charge	
Bytes 5 to 14:	ASCII characters representing cost or,			
Bytes 5 to 9:	ASCII characters representing number of units			
Bytes 10 to 14:	ASCII characters representing price per unit			

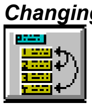
The structure of the packet:
The example below is for an additional charge 23 units without price per units.

Byte Count	Description	Example Value
1:	Packet Type	0x21 (Additional Charge)
2:	Packet Length	0x0E (14 Packet Words)
3:	Packet Words (currency)	0x2D ASCII -
4:	Packet Words (currency)	0x2D ASCII -
5:	Packet Words (currency)	0x2D ASCII -
6:	Packet Words	0x10 Charged Units, Total Normal Current
7:	Packet Words (units)	0x30 ASCII 0
8:	Packet Words (units)	0x30 ASCII 0
9:	Packet Words (units)	0x30 ASCII 0
10:	Packet Words (units)	0x32 ASCII 2
11:	Packet Words (units)	0x33 ASCII 3 (23 units)
12:	Packet Words (price per unit)	0x2D ASCII -
13:	Packet Words (price per unit)	0x2D ASCII -
14:	Packet Words (price per unit)	0x2D ASCII -
15:	Packet Words (price per unit)	0x2D ASCII -
16:	Packet Words (price per unit)	0x2D ASCII - (no price per unit)

Packet Type: Duration of the Call

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:



The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Call Duration check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Duration of the Call information is not sent. The duration is represented by 6 ASCII characters signifing the hours (00 to 99), minutes (00 to 59), and seconds (00 to 59).

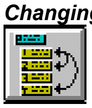
The structure of the packet:
The example below represents a call duration of 1 hour, 27 minutes, and 43 seconds.

Byte Count	Description	Example Value
1:	Packet Type	0x23 (Duration of the Call)
2:	Packet Length	0x06 (6 Packet Words)
3:	Packet Words (hour)	0x30 ASCII 0
4:	Packet Words (hour)	0x31 ASCII 1 (1 hour)
5:	Packet Words (minute)	0x32 ASCII 2
6:	Packet Words (minute)	0x37 ASCII 7 (27 minutes)
7:	Packet Words (second)	0x34 ASCII 4
8:	Packet Words (second)	0x33 ASCII 3 (43 seconds)

Packet Type: Network Provider Identity

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:



The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Provider Identity check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Network Provider Identity information is not sent. Though nominally 20 characters is the longest string that should be sent with this packet, the text box allows up to 50 characters.

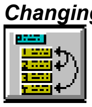
The structure of the packet:
The example below show the bytes sent with an identity of “Provider Name”.

Byte Count	Description	Example Value
1:	Packet Type	0x30 (Network Provider Identiy)
2:	Packet Length	0x0D (13 Packet Words)
3:	Packet Words	0x50 ASCII P
4:	Packet Words	0x72 ASCII r
5:	Packet Words	0x6F ASCII o
6:	Packet Words	0x76 ASCII v
7:	Packet Words	0x69 ASCII i
8:	Packet Words	0x64 ASCII d
9:	Packet Words	0x65 ASCII e
10:	Packet Words	0x72 ASCII r
11:	Packet Words	0x20 ASCII (space)
12:	Packet Words	0x4E ASCII N
13:	Packet Words	0x61 ASCII a
14:	Packet Words	0x6D ASCII m
15	Packet Words	0x6E ASCII e

Packet Type: Carrier Identity

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:



The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Carrier Identity check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Carrier Provider Identity information is not sent. Though nominally 20 characters is the longest string that should be sent with this packet, the text box allows up to 50 characters.

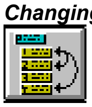
The structure of the packet:
The example below show the bytes sent with an identity of “Carrier Name”.

Byte Count	Description	Example Value
1:	Packet Type	0x31 (Carrier Identity)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words	0x43 ASCII C
4:	Packet Words	0x61 ASCII a
5:	Packet Words	0x72 ASCII r
6:	Packet Words	0x72 ASCII r
7:	Packet Words	0x69 ASCII i
8:	Packet Words	0x65 ASCII e
9:	Packet Words	0x72 ASCII r
10	Packet Words	0x20 ASCII (space)
11:	Packet Words	0x4E ASCII N
12:	Packet Words	0x61 ASCII a
13:	Packet Words	0x6D ASCII m
14	Packet Words	0x6E ASCII e

Packet Type: Selection of Terminal Function

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:

 The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Terminal Function check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Selection of Terminal Function information is not sent. The length of packet ranges from 4 to 23 bytes. The first payload byte (following the packet length byte) signifies the type of information to send as either connection type (0x01), multiple subscriber number (MSN) (0x02), or sub-address (0x03). In the case of specifing the connection type, the following byte qualifies the connection as one of the following:

Connection Type	Value
Not Identified	0x00
Voice Call	0x01
Fax Call	0x02
Data Call	0x03
Video Call	0x04
E-Mail Call	0x05
Telemetric Call	0x06
Text Call	0x07

Otherwise, if the MSN or sub-address information is to be sent, then the following bytes contain a text string of up to 20 characters.

The structure of the packet:

The example below specifies a connection type for a voice call.

Byte Count	Description	Example Value
1:	Packet Type	0x40 (Selection of Terminal Function)
2:	Packet Length	0x02 (2 Packet Words)
3:	Packet Words	0x01 (Connection Type Information)
4:	Packet Words	0x01 (Voice Call)

The example below specifies a Multiple Subscriber Number of 071 250 7587. The text box would contain the following: "071 250 7587" (without the quotation marks).

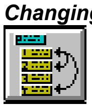
Byte Count	Description	Example Value
1:	Packet Type	0x40 (Selection of Terminal Function)
2:	Packet Length	0x0D (13 Packet Words)
3:	Packet Words	0x02 (MSN information)
4:	Packet Words (number)	0x30 ASCII 0
5:	Packet Words (number)	0x37 ASCII 7
6:	Packet Words (number)	0x31 ASCII 1
7:	Packet Words (number)	0x20 ASCII (space)
8:	Packet Words (number)	0x32 ASCII 2
9:	Packet Words (number)	0x35 ASCII 5
10:	Packet Words (number)	0x30 ASCII 0
11:	Packet Words (number)	0x20 ASCII (space)
12:	Packet Words (number)	0x37 ASCII 7
13:	Packet Words (number)	0x35 ASCII 5
14:	Packet Words (number)	0x38 ASCII 8
15:	Packet Words (number)	0x37 ASCII 7 (number 071 250 7587)

Packet Type: Display Information

What standard does this packet apply to:

Only the ETSI standard.

Changing the packet data:

 The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Display Information check box is checked, the packet will be sent with the

next Caller ID transmission. Otherwise the Display Information is not sent. The packet can send a text string of up to 252 characters. The data byte prior to the first character qualifies the packet as follows:

Bits 1 to 7:	Value =	0: Unknown or Other
		1: Positive Acknowledgement
		3: Negative Acknowledgement
		4: Advertisement
		5: Network Provider Information
		6: Remote User Provided Information
Bit 8:	clear:	No stored information
	set:	Stored information

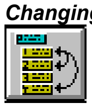
The structure of the packet:
The example below sends the text string "Hello World" as unknown non-stored information.

Byte Count	Description	Example Value
1:	Packet Type	0x50 (Display Information)
2:	Packet Length	0x0C (12 Packet Words)
3:	Packet Words	0x00 (unknown & not stored)
4:	Packet Words (text)	0x48 ASCII H
5:	Packet Words (text)	0x65 ASCII e
6:	Packet Words (text)	0x6C ASCII l
7:	Packet Words (text)	0x6C ASCII l
8:	Packet Words (text)	0x6F ASCII o
9:	Packet Words (text)	0x20 ASCII (space)
10:	Packet Words (text)	0x57 ASCII W
11:	Packet Words (text)	0x6F ASCII o
12:	Packet Words (text)	0x72 ASCII r
13:	Packet Words (text)	0x6C ASCII l
14:	Packet Words (text)	0x64 ASCII d

Packet Type: Service Information

What standard does this packet apply to:
Only the ETSI standard.

Changing the packet data:



The data contained in this packet is displayed and can be changed in the Additional ETSI Packets window. If the Service Information check box is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Service Information is not sent. The Service Information is selected from the drop down list, and can be either "Active" or "Not Active".

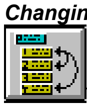
The structure of the packet:
The example below represents a Service Information packet sending "Not Active".

Byte Count	Description	Example Value
1:	Packet Type	0x55 (Service Information)
2:	Packet Length	0x01 (1 Packet Word)
3:	Packet Word	0x00 (not active)

Packet Type: Extension for Network Operator Use

What standard does this packet apply to:
Only the ETSI standard.

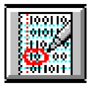
Changing the packet data:

The data contained in this packet is displayed and can be changed in the CID Packet Format window. If the Ext. for Network checkbox is checked, the packet will be sent with the next Caller ID transmission. Otherwise the Extension for Network Operator Use information will not be sent. The packet data is composed of three fields. These being the country code, network operator code, and the version code. The country code and the version code can have up to 3 characters, while the network operator code will allow up to 4 characters. If the maximum number of characters are not used, the data sent will have extra 'spaces' (0x20) appended to the three codes.

The structure of the packet:

The example below is for a country code of 62, operator code of 175 and version code of 1.

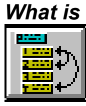
Byte Count	Description	Example Value
1:	Packet Type	0xE0 (Extension for network operator)
2:	Packet Length	0x0A (10 Packet Words)
3:	Packet Words (country)	0x36 ASCII 6
4:	Packet Words (country)	0x32 ASCII 2
5:	Packet Words (country)	0x20 ASCII (space)
6:	Packet Words (operator)	0x31 ASCII 1
7:	Packet Words (operator)	0x37 ASCII 7
8:	Packet Words (operator)	0x35 ASCII 5
9:	Packet Words (operator)	0x20 ASCII (space)
10:	Packet Words (version)	0x31 ASCII 1
11:	Packet Words (version)	0x20 ASCII (space)
12:	Packet Words (version)	0x20 ASCII (space)



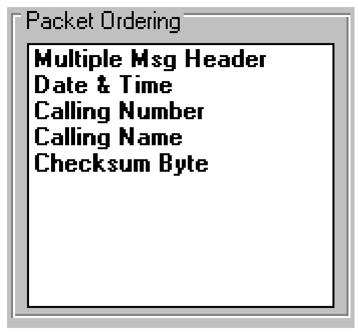
Editing the Packets

Changing the Transmission Order of the Packets

What is Packet Ordering?



Caller ID messages that are transmitted to CPE's may be composed of one or more packets. For the Bellcore Multiple Message Format and the ETSI Message Formats, the messages are composed of more than one packet. All of the packets that make up the current message are displayed in the Packet Order list box displayed in the CID Packet Order window. An example of this list box is shown below for the Bellcore Multiple Message Format.



The transmission order of the packets is from the top to the bottom. In this case the Multiple Msg. Header (which contains the Message Type and Message Length bytes) is the first packet to be transmitted. Following this is the Date & Time, Calling Number, and Calling Name packets. The last packet is called the Checksum Byte which contains a single byte that represents the checksum of all the packets listed.

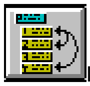
Note: In order for the Caller ID message to be consistent with the selected standard, the Multiple Msg. Header must be first in the list, and the Checksum Byte last. If this is not true, the Message will be flagged as "Altered" in the status bar. The Multiple Msg. Header and Checksum Byte packets are only present if the message type is either the Bellcore Multiple Message formats, or the ETSI message formats.

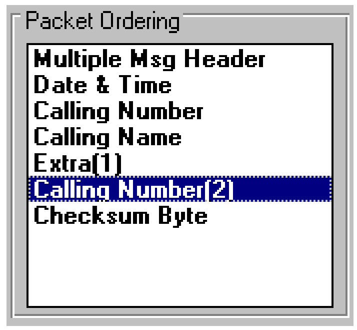
How to Change the Order?
The packet order can be changed by using either the mouse or keyboard. Using the mouse, simply press the mouse button on the packet you wish to move, and drag it to the desired position. Once the packet is in the desired position in the list, release the mouse button. To change the order using the keyboard, select the packet to move with the up/down arrow keys. Once the packet is selected, press and hold the SHIFT key. Then use the up/down arrow keys to move that packet up or down. Once the packet is in its desired position release the shift key.

Note: For either the Bellcore Multiple Message formats, or the ETSI message formats, when a packet is enabled by clicking on its associated checkbox, that packet is added to the packet ordering list second from the last. The checksum packet will remain the last packet in transmission order.

Note: The Bellcore and ETSI standards do not specify the ordering of packets within the message. As such, the CPE should be able to accept the packets in any order.

Inserting, Deleting, Coping Non-Standard Packets

It is possible to insert, delete, or copy packets that make up the Caller ID Message. All of the packets that make up the current message are displayed in the Packet Ordering list box within the CID Packet Order window. The example below shows a common Bellcore Multiple Message Format with a new packet inserted, and a packet copied, or duplicated.



Packets added to the list are shown with a numeric tag appended to the packet name. Up to 6 additional packets can be added to the list by either inserting new packets, named as "extra", or coping existing packets. If packets are copied, the name of the new packet is the same as the existing packet with the numeric tag appended. Packets that are inserted contain no data bytes when added to the list, while packets that are copied contain the identical data bytes as the original packet. To modify the data in a packet, see: Byte-wise Editing of any Packet.

How to Insert a Packet?
To insert an extra packet in the message, select the Packet Ordering list box in the CID Packet Order window. This enables the [INSERT PACKET] command in the menu. Select the [EDIT] [INSERT PACKET] command from the menu and a new packet titled "extra" will be inserted. The new packet will be inserted just before the Checksum Byte for the Bellcore Multiple Message formats and the ETSI message formats, or at the end of the list for the other message types. See the note at the bottom to determine the effects on the message checksum.

How to Copy a Packet?
To copy an existing packet in the message, select the Packet Ordering list box in the CID Packet Order window. This enables the [COPY PACKET] command in the menu. Select the packet to be copied and then select the [EDIT] [COPY PACKET] command from the menu and a new packet with the same title, with numeric tag appended, will be inserted just before the Checksum Byte for the Bellcore Multiple Message formats and the ETSI message formats. For other message types, it will be inserted at the end of the packet list. See the note at the bottom to determine the effects on the message checksum.

How to Delete a Packet?
To delete a packet in the message, select the Packet Ordering list box in the CID Packet Order window. This enables the [DELETE PACKET] command in the menu. Select the packet to be deleted and then select the [EDIT] [DELETE PACKET] command from the menu. If the Multiple Msg. Header or Checksum Byte packets are deleted, the message will be shown as altered in the status bar, since the message is not consistent with either the Bellcore or ETSI standard.

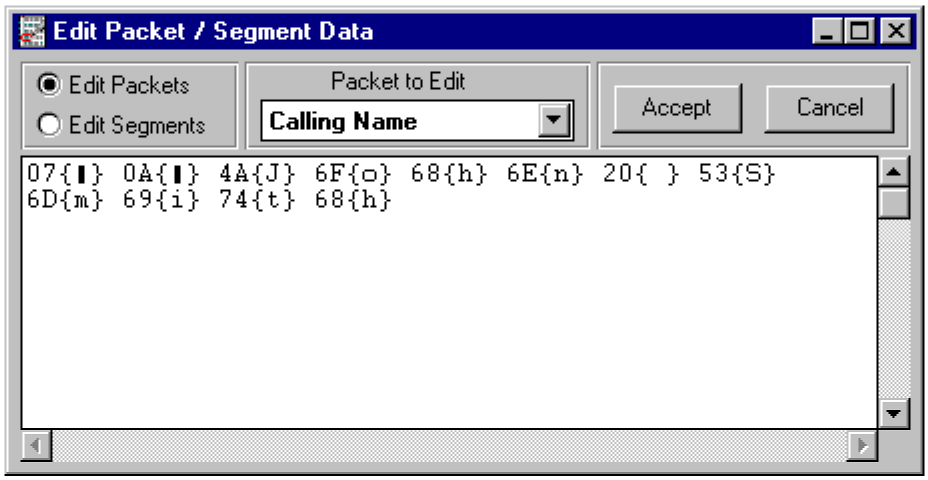
Note: When a packet is inserted, copied, or deleted, the Checksum Byte is not recalculated. This is independent of the "Automatic Checksum Calculation" control setting. For information on how to force a re-calculation, see: Automatic Checksum Calculation

---

Byte-Wise Editing of any Packet

Editing Window:

Any of the packets that make up the current Caller ID message can be viewed or edited by selecting the Edit Packet/Segment window. The editing window can be set to edit either the data packets, or the segments of the Caller ID transmission. As shown below, the two buttons in the upper left corner of the editing window select between the packets or segments.



How to Edit a Packet?

To edit a packet, first verify that the "packet" button is selected, and not the "segment" button. Then pick the packet to view or edit from the drop down list box. In the editing area, the contents of the packet will be displayed. In the example above, the Name packet has been selected. The bytes making up the packet are listed in the editing window. Following each hexadecimal value is its ASCII character enclosed in {}. If the ASCII character is unprintable, the character displayed is a solid thick vertical line. Changes can be made to the packet by either over writing the existing hexadecimal data, or inserting new hexadecimal data. Any text inserted between the {} characters is ignored by the editor. When the changes have been made, press the ACCEPT button. This will scan the editing window for errors, and if none have been detected, the new packet will be re-formatted and re-displayed in the editor window. To ignore any changes made in the editing window, press the CANCEL button and the original packet will be re-displayed in the window in its original form. For information on how editing packets effects the checksum byte, see: Automatic Checksum Calculation

---

Note: If errors are detected in the format of the packet, a message will be displayed and the offending portion will be highlighted.

Rules for Making Changes:

1) The value for the bytes must be entered in hexadecimal format
2) No more than two hexadecimal digits for a byte
3) The only characters allowed outside the {} characters are "0"-"9" and "a"-"f" ("A"-"F")
4) All text between { and } characters is ignored
5) For every { character, there must be a } character, and vise versa
6) Nesting of the {} characters is not allowed

Automatic Message Checksum Calculation

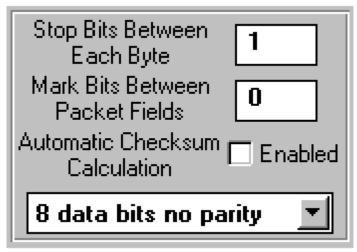
Checksum Basics:

All of the message formats use the last byte transmitted as a checksum byte. The byte is the two's complement of the sum of all the message bytes, except the checksum byte of course. The checksum is the last byte in the packet for all the Bellcore single message formats. In the multiple message formats and the ETSI message formats, the checksum byte is contained within its own packet.

The checksum byte is **always** calculated if one of the packet's checkbox has been enabled or disabled in the Main Settings, or CID Packet Format windows. It will also be calculated **always** if the packet's data has been changed in the Main Settings, or CID Packet Format windows.

The checksum will **not** be re-calculated if packets have been inserted, copied, or deleted from the Packet Ordering list box in the CID Packet Format window. Although, if any packet is edited and the "Automatic Checksum Calculation" checkbox is enabled, the new checksum will reflect the inserted, copied, or deleted packet. See below.

Optional Checksum Re-calculation:
In cases where the packets are edited, whether or not the checksum byte is re-calculated after a change is controlled with the "Automatic Checksum Calculation" checkbox in the CID Packet Format window, as shown below.

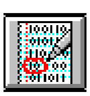


If a packet is edited in the Edit Packet/Segment Data window and the "Automatic Checksum Calculation" checkbox is **disabled**, the checksum byte **is not** re-calculated. Accordingly, if the checkbox is **enabled**, the checksum byte **is** re-calculated once the ACCEPT button is pressed in the Edit Packet/Segment Data window (assuming no formatting errors were found).

For the Bellcore Multiple Message formats, and the ETSI message formats, the checksum is calculated from all of the bytes in all of the packets displayed in the Packet Ordering list box in the CID Packet Format window. This includes packets that have been inserted, copied, or deleted using the [INSERT PACKET], [COPY PACKET], or [DELETE PACKET] menu command.

For the Bellcore Single Message formats, the checksum byte is part of the single packet. Specifically, the checksum is the last byte in the packet. If the contents of the packet are edited and the "Automatic Checksum Calculation" checkbox is enabled, the checksum will be re-calculated. However, if additional packets have been inserted, or copied using the [INSERT PACKET] or [COPY PACKET] menu commands, the updated checksum value will not reflect the data in the additional packets.

Note: If the "Automatic Checksum Calculation" checkbox is enabled, and the checksum byte has been changed using the editor, a warning message will be displayed indicating that the checksum has been re-calculated and has over written any changes made.



Editing the Segments

What Are the Segments?

The segments represent the various bit patterns generated by the FSK modulator during a Caller ID transmission. The FSK modulator not only generates the bit pattern of the data message being sent to the CPE, but also other bit patterns sent before or after the data message according to the selected standard. The Bellcore and ETSI standards define a preamble (channel seizure) segment, mark segment, and data message segment. The program defines three other segments useful in testing the robustness of the CPE's decoding software. These other segments are termed the Start Burst, Mark Out, and End Burst.

During a Caller ID transmission, the FSK modulator generates all the bits defined in the six segments in the following order:
1) Start Burst Segment
2) Preamble (Channel Seizure) Segment (not defined for Type II CIDCW Caller ID)
3) Mark Segment
4) Data Message Segment
5) Mark Out Segment
6) End Burst Segment

Start Burst Segment:
The Start Burst segment is not used during normal Caller ID transmission. As such its transmission time is set to zero upon resetting the program, changing the Caller ID standard, or changing the message type. It is possible to define an arbitrary bit pattern for the Start Burst by editing the bit pattern of the segment. This can be useful in testing the CPE's ability to reject spurious bit patterns before a qualified Preamble, or Mark segment has been sent.

Since the length of the Start Burst segment is normally zero, anytime this segment has a length other than zero, the Caller ID transmission will be flagged as altered in the program's status bar.

Preamble Segment:
The Preamble or Channel Seizure segment is defined as an alternating zero/one bit pattern preceding the transmission of the Mark segment and Data Message Segment. For Type II (CIDCW) Caller ID transmissions, the Preamble segment is not sent, and as such will be displayed as being disabled in the Transmission Format window. Using the segment editor, the bit pattern of the preamble can be changed as desired. However, a change in the standard or message type will set the Preamble segment to an alternating one/zero pattern.

The Preamble segment will be flagged as altered in the Transmission Format window if the bit pattern of the segment is not an alternating zero/one pattern, which starts with a zero. The altered segment will also flag the Caller ID transmission as altered in the program's status bar.

Mark Segment:
The Mark segment follows the Preamble segment (if enabled), and precedes the data message segment. Normally, the segment consists of all ones (mark bits). Like all segments, the bit pattern may be edited; however, changing the standard or message type will reset the Mark segment to an all ones pattern.

The Mark segment will be flagged as altered in the Transmission Format window if the bit pattern contains anything other than all ones (mark bits). The altered segment will also flag the Caller ID transmission as altered in the program's status bar.

Note: A change in the Caller ID transmission mode, either automatically or manually, will set the length of the Mark segment to 180 bits for Type I and 80 bits for Type II (CIDCW). The mode change will also cancel any editing done and return the Mark segment to all ones. As such, when altering the characteristics of the Mark segment, it is suggested not to select the Auto Detect Mode for Caller ID transmissions, as any change in the hook switch state will reset the Mark segment as described above.

Data Message Segment:
The Data Message segment follows the Mark segment and contains all of the data contained within the packets (such as date, time, number, name). This segment too can be edited to any desired bit pattern. The transmission time of this segment can not be changed arbitrary like the other segments, since its length depends the data contain in the message packets. Any changes to the standard, message type, or packet data will force a recalculation of the Data Message segment, and cancel any bits that have been edited.

The Data Message segment will be flagged as altered in the Transmission Format window, if the bits have been altered in any way that the data no longer reflects the packet data or the settings of the number of Stop Bits or Mark Stuffing Bits. The altered segment will also flag the Caller ID transmission as altered in the program's status bar.

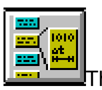
Mark Out Segment:
The Mark Out segment follows the Data Message segment, and normally contains all ones (mark bits). Like all segments, the bit pattern may be edited; however, changing the standard or message type will reset the Mark Out segment to an all ones pattern.

The Mark Out segment will be flagged as altered in the Transmission Format window if the bit pattern contains anything other than all ones (mark bits). The altered segment will also flag the Caller ID transmission as altered in the program's status bar.

End Burst Segment:
The End Burst segment is not used during normal Caller ID transmission. As such its transmission time is set to zero upon resetting the program, changing the Caller ID standard, or changing the message type. It is possible to define an arbitrary bit pattern for the End Burst by editing the bit pattern of the segment.

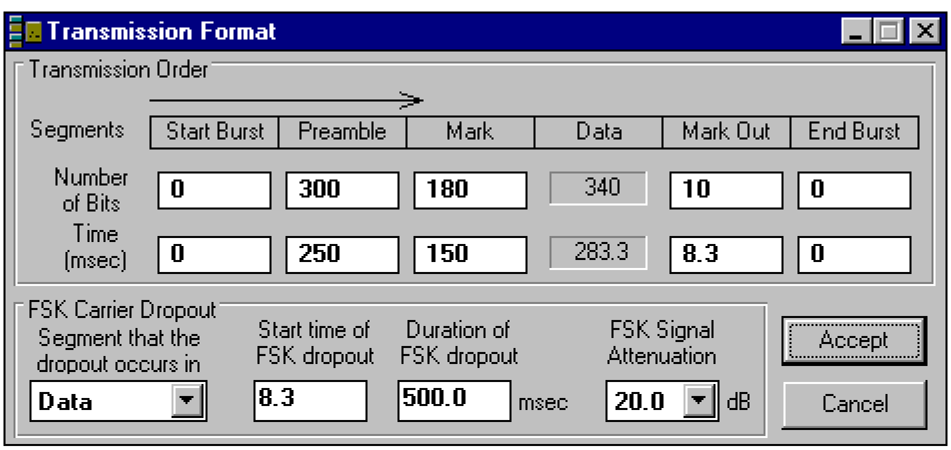
Since the length of the End Burst segment is normally zero, anytime this segment has a length other than zero, the segment will be flagged as altered in the Transmission Format window. The altered segment will also flag the Caller ID transmission as altered in the program's status bar.

Changing the Transmission Time of Each Segment


The Transmission Format window is used to display and/or change the transmission time of the segments. As shown in the example below, the transmission time and number of bits in each segment is displayed. The relationship between the transmission time and number of bits in a segment is defined by the Baud rate. The Baud rate can be changed in the Advance Setup window, under the category of FSK Modulator.

To change the number of bits in a segment, simply type a new value in the "Number of Bits" text box for the desired segment. Pressing the accept button will then update that segment's transmission time value.

Likewise, to change the transmission time of a segment, type a new value in the "Time" text box for the desired segment. The time value should be entered in units of milliseconds. Pressing the accept button will then update the number of bits in the segment.



The number of bits in a segment must be an integer value. As such, when entering a transmission time value and pressing the accept button, the time displayed may be slightly different as the program will satisfy the condition that the number of bits in a segment be an integer value.

Note: The transmission time, or number of bits in the Data Message segment can not be changed in the Transmission Format window. Since the Data Message segment is determined from the data packets, only changing the data contained within the packets will alter the number of bits in the Data Message segment. The only exception is if the Data Message segment is bit-wise edited.

Altered Segments:
If any segment has been bit wise edited in a manner that is inconsistent with the normal state for that segment, it will be flagged as altered. This is accomplished by highlighting the segment's name in yellow in the Transmission Format window. Of course, if any segment has been altered, the entire Caller ID transmission is deemed to be altered and is reflected in the program status bar.

To determine what constitutes an altered segment see the section: What are the Segments

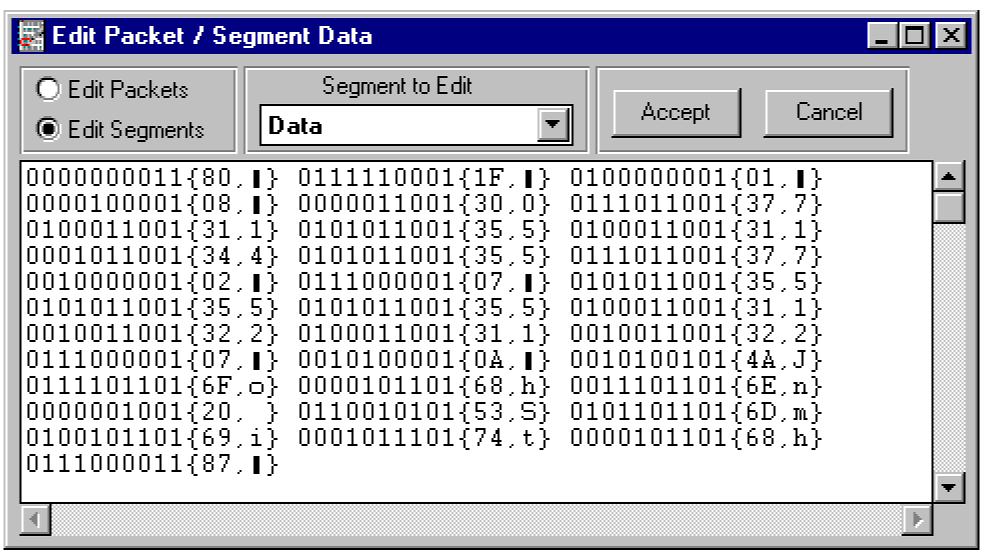
Note: Changing the number of bits or transmission time of a segment will cancel any edits performed on that segment. The segment bit pattern will be set to its default state. For more information on how changing the various message types, packets, and segments effect each other, see the section: Hierarchical Caller ID Data Structure

For information on how to specifiy FSK carrier dropouts, see: Controlling the Caller ID Transmission (Section 3-2)

Bit-Wise Editing of any Segment

Editing Window:

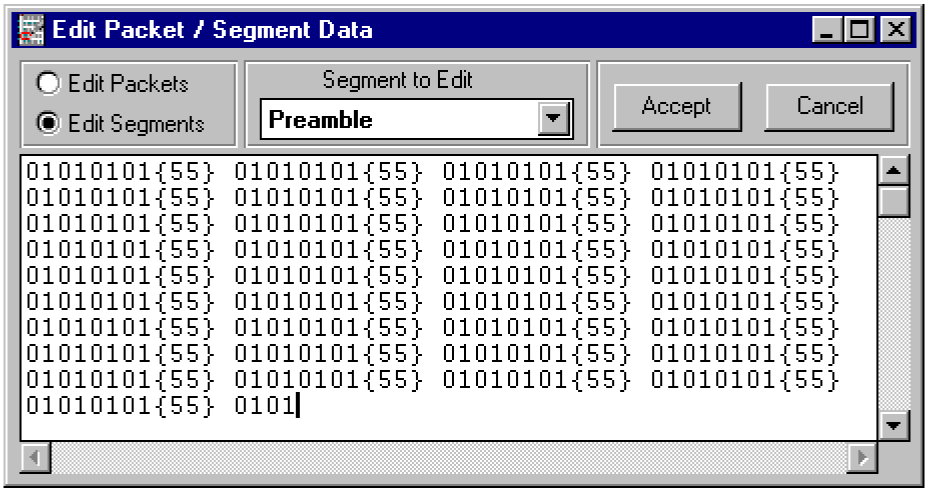
Any of the segments that make up the current Caller ID transmission can be viewed or edited by selecting the Edit Packet/Segment window. The editing window can be set to edit either the Caller ID transmission segments, or the Caller ID message packets. As shown below, the two buttons in the upper left corner of the editing window select between packets and segments.



How to Edit a Segment?
To edit a segment, first verify that the "segment" button is selected, and not the "packet" button. Then pick the segment to view or edit from the drop down list box. In the editing area, the contents of the segment will be displayed. In the example above, the Message Data segment has been selected.

Two different formats for displaying the segment data are used depending on the segment type being displayed. For the Message Data segment, as displayed above, the data bytes that make up the message are displayed starting with a zero (start bit), followed by the data bits (LSB first), then ending in one or more ones (stop bits). Following the bit pattern for each data byte is its hexadecimal value, and ASCII character equivalent (if printable) enclosed in {} characters.

For all the other segments, the segment bits are displayed in groups of eight with the hexadecimal equivalent shown enclosed in {} characters. An example of the Preamble segment is shown below.



Changes can be made to the segments by either over writing the existing bits, or inserting extra bits. Any text inserted between the {} characters is ignored by the editor. When the changes have been made, press the ACCEPT button. This will scan the editing window for errors, and if none have been detected, the new segment will be re-formatted and re-displayed in the editor window. To ignore any changes made in the editing window, press the CANCEL button and the original segment will be re-displayed in the editing window.

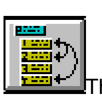
If a segment has been altered and its bit pattern changed from its default setting, that segment will be highlighted as altered within the Transmission Format window. As a result, the entire Caller ID transmission will also be classified as altered. This will illuminate the altered flag on the program status bar. For a definition of what is the normal bit pattern for each segment, see the section titled: What are the Segments.

Note: If errors are detected in the format of the segment, a message will be displayed and the offending portion will be highlighted.

Rules for Making Changes:

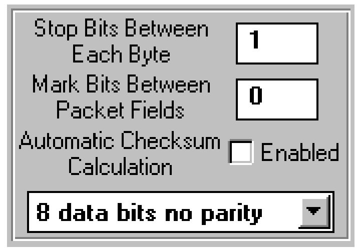
1) The value for the bits (mark or space) are represented by the characters "1" or "0"
2) The only characters allowed outside the {} characters are "0" and "1"
3) All text between { and } characters is ignored
4) For every { character, there must be a } character, and vise versa
5) Nesting of the {} characters is not allowed

Changing the Number of Stop or Mark Stuffing Bits in the Data Segment

The number of stop bits or mark stuffing bits used when creating the Message Data segment can be changed within the CID Packet Format window. The stop bit(s) are appended after the MSB of every data byte in the Caller ID Message. These stop bit(s) are used to signal to the data decoder in the CPE the end of a data byte. The mark stuffing bits are extra stop bits inserted at the end of specific data bytes in the Caller ID Message. This feature is provided in order to mimic a

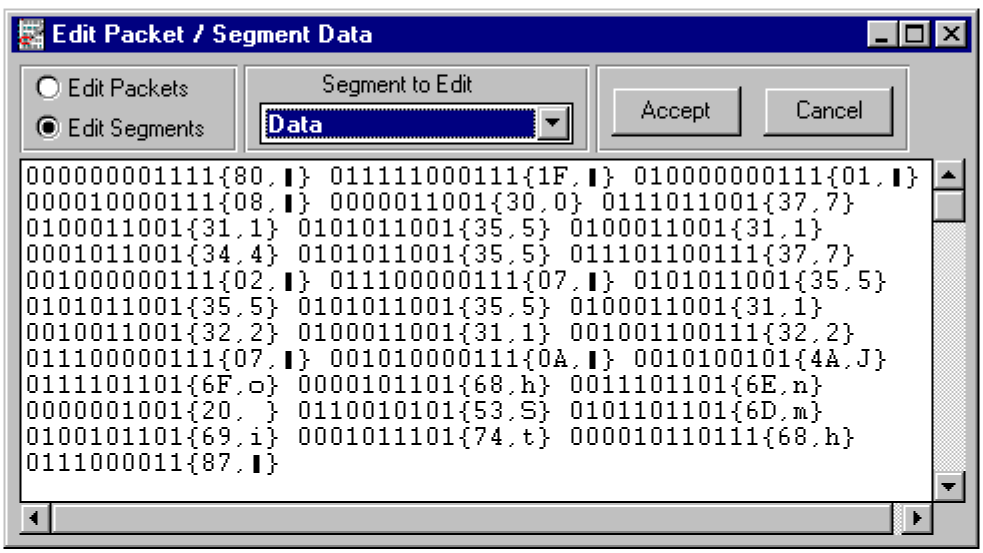
SPCS that inserts extra stop bits between specific bytes in the Caller ID Message. The specific bytes where a SPCS is allowed to insert these extra stop bits are as follows:

1. Message Type byte
2. Message Length byte
3. Packet Type byte
4. Packet Length byte
5. Last byte in a Packet

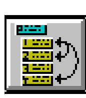


The current number of stop bits and mark stuffing bits are displayed in the text boxes as displayed above. To change either of the values, simply select the text box and enter a new value. The Message Data segment will be updated immediately upon pressing "ENTER", or automatically updated during any further actions.

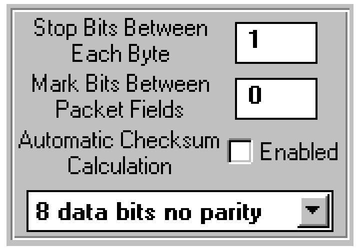
The following figure shows the contents of the Message Data segment with the number of mark stuffing bits set to 2. As shown, the bytes that correspond to the five types listed above (Message Type, Message Length, Packet Type, Packet Length, last packet byte) are formatted with two extra stop bits, for a total number of stop bits of three.



Setting the Message Data Parity

The parity encoding used with any of the Caller ID message types can be changed between the default 8 bits no parity, to 7 bits odd or even parity. Though most Caller ID systems use the 8 bit no parity setting, some Caller ID standards may use 7 bits with parity. Changing the parity is done by selecting the desired option from the appropriate drop-down list box shown in the CID Packet Format window. It is important to note that the parity settings do not effect all the bytes in a Caller ID message. Only the sections of the Caller ID message that include ASCII text data are effected by the parity settings. This includes the date and time information, calling number, and calling name

among others. The message type, parameter type, parameter length, checksum byte, and all non-ASCII data is unaffected.



■ Section 3-5 Transmission Parameters

What are the Transmission Parameters?

Transmission parameters are a collection of properties that define how a Caller ID transmission will be sent. These parameters exist as three basic types. The first is a numeric parameter which can be set to any value between its specified minimum and maximum values. The second, is a binary parameter, which can be set in only one of two possible states. These are typically used to enable or disable certain options. Third is an enumerated parameter, from which the setting is made from a fixed list of possibilities. Certain parameters are common between the Caller ID standards, while others are specific to the selected standard.

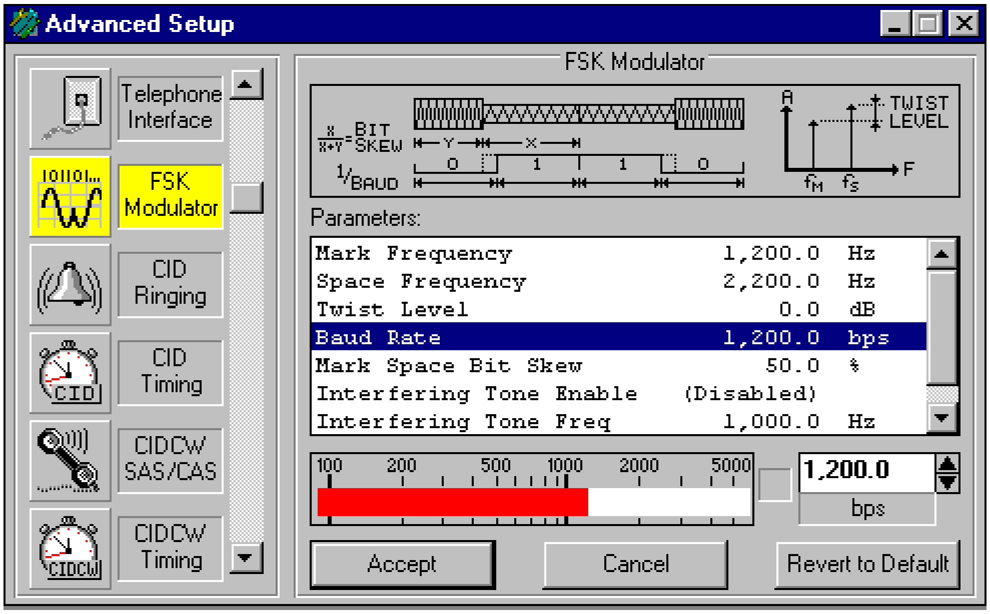
How to Change the Parameters

All of the parameters, except the FSK Transmit Level and FSK Signal-to-Noise Ratio (SNR), are modified, changed, or viewed within the Advanced Setup window. The two exceptions listed above can be changed within the Main Settings window.

The Advanced Settings window is used to adjust the parameters that deal with the physical data transmission layer. All of the parameters are grouped into categories that are displayed on the left side of the window. There are six basic categories that define all the parameters for the Bellcore, ETSI, and Australian Caller ID Standards. These categories are:

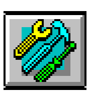
i) Telephone Interface Parameters,
ii) FSK modulator Parameters,
iii) Ringing Generator Parameters,
iv) Time Sequencing for Type I (CID) Parameters,
v) SAS/CAS and Alerting Tones, and
vi) Time Sequencing for Type II (CIDCW) Parameters.

Select the general category of the parameter you wish to change by either clicking on the icon or by using the vertical scroll bar. The example below shows how to change the baud rate of the FSK modulator. First click the FSK modulator category, which displays a list of all parameters within that category, and at the top of the screen a graphical representation of the parameters.



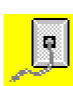
From the list of parameters, select the one labeled "Baud Rate". The selection shows the current setting along with its units. Below the list, a bar graph graphically represents the current value of the selected parameter, along with a text box displaying its value. To change the parameter's value, either type in a new value in the text box, or use the up/down spin box to increment or decrement the current value, or click on the bar graph the approximate value you wish to enter. Once you have entered a new value, press the "Accept" button, or press ENTER and the new parameter value will be entered into the parameter list. To abort making any changes, press the "Cancel" button. If the "Revert to Default" button is pressed, the parameter's value changes back to the standard's default value.

If the value of parameter is outside specified limits, the letter "L" will be illuminated beside the text box, along with the letter "L" in the status line of the master window. This is to remind you that a parameter value is outside the Standard's specified limit.



List of Transmission Parameters

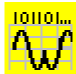
A list of all transmission parameters is given here by category. For a description of each parameters function, its default value, maximum and minimum values, and script language reference name, see Appendix A: Transmission Parameters.



Parameter Category: Telephone Interface

Parameters Contained within this Category:

Telephone Line Voltage
Telephone Loop Current
Telephone Line Impedance (600 ohms, 900 ohms, or Complex)
Telephone Line Polarity



Parameter Category: FSK Modulator

Parameters Contained within this Category:

Transmit Level
Signal-to-Noise Ratio
Mark Frequency
Space Frequency
Twist Level
Baud Rate
Mark/Space Bit Skew
Interfering Tone Enable
Interfering Tone Frequency
Interfering Tone Level

Parameter Category: Ringing Generator

Parameters Contained within this Category:

Ringer Frequency
Ringer Level
Ringer Sequence
Ringer On Time #1
Ringer Off Time #1
Ringer On Time #2
Ringer Off Time #2
Ringer On Time #3
Ringer Off Time #3
Number of Ringing Cycles

Parameter Category: Type I CID Timing

Parameters Contained within this Category:

Bellcore:

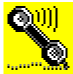
Time to Ringing
Time to Data Transmission

ETSI:

Time to Polarity Reversal
Time to Dual Tone Alert SignalTime to Ring Burst Alert SignalTime to Data TransmissionTime to Ringing

Australia

Time to Ring Burst Alert
Time to Line Reversal
Time to OSI
Time to Data Transmission
Time to Ringing
Time to Post Reversal



Parameter Category: SAS/CAS and Alert Tones

Parameters Contained within this Category:

Bellcore

SAS Tone Frequency
SAS Tone Level
SAS On Time
SAS Off Time
SAS Duration
CAS Tone #1 Frequency
CAS Tone #2 Frequency
CAS Tone #1 Level
CAS Tone #2 Level
CAS Duration
Noise Generator Enable
Noise Generator Level

ETSI

Number of Alert Tone Frequencies
Dual Tone Alert #1 Frequency
Dual Tone Alert #2 Frequency
Dual Tone Alert #1 Level
Dual Tone Alert #2 Level
DTAS Duration for Type IDTAS Duration for Type II
Noise Generator Enable
Noise Generator LevelRing Burst FrequencyRing Burst LevelRing Burst Duration

Australia

SAS Tone Frequency
SAS Tone Level
SAS On Time
SAS Off Time
SAS Duration
CAS Tone #1 Frequency
CAS Tone #2 Frequency
CAS Tone #1 Level
CAS Tone #2 Level
CAS Duration
Noise Generator Enable
Noise Generator Level
Ring Burst Frequency
Ring Burst Level
Ring Burst Duration
OSI Duration

Parameter Category: Type II CIDCW Timing

Parameters Contained within this Category:

Bellcore:

Time to SAS Tone
Time to CAS Tone
Time-out for ACK Tone
Time to Data Transmission
Transmit Data Even After ACK Time-outACK Low Group Minimum Level
ACK High Group Minimum Level
ACK Frequency Tolerance
Valid ACK Digit

ETSI:

Time to DTAS Tone
Time-out for ACK Tone
Time to Data Transmission
Transmit Data Even After ACK Time-outACK Low Group Minimum Level
ACK High Group Minimum Level
ACK Frequency Tolerance
Valid ACK Digit

Australia

Time to SAS Tone
Time to CAS Tone
Time-out for ACK Tone
Time to Data Transmission
Transmit Data Even After ACK Time-outACK Low Group Minimum Level
ACK High Group Minimum Level
ACK Frequency Tolerance
Valid ACK Digit

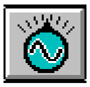
See Also

Other Aspects of the Data Transmission that can be changed:

1) Changing the Preamble and Mark Segment Transmission Time

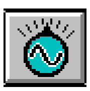
2) Changing the Number of Stop or Mark Stuffing Bits in the Data Segment

■ Section 3-6 Tone Generator Functions



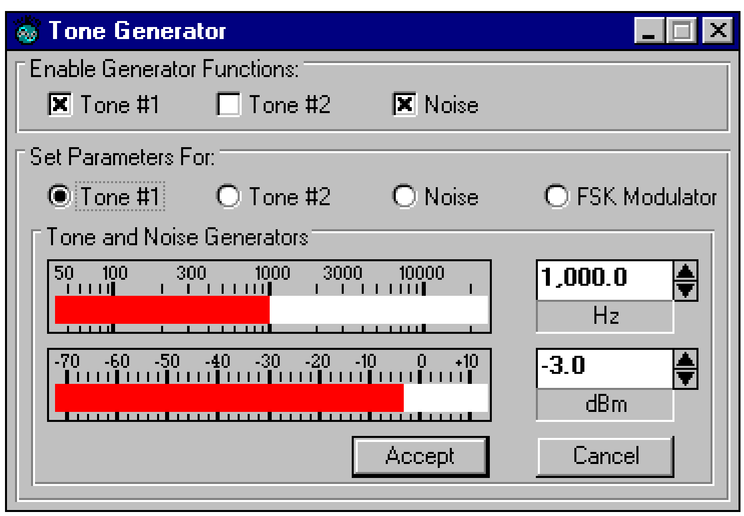
Using the Tone Generator

What is the Tone Generator?
The Tone Generator window provides access to a flexible signal generator that can be used for various purposes. The signal generator can supply up to two pure tones at an arbitrary frequency and amplitude plus a broad band noise signal simultaneously. One of the pure tone generators can be configured as a FSK modulator with all of the flexibility of the FSK modulator used for Caller ID transmissions. Starting a Caller ID transmission will suspend the tone generator until the transmission is completed. Once completed, the tone generator will be activated again. The exception to this is the FSK modulator. If the FSK modulator function is running when a Caller ID transmission is started, the FSK modulator will be terminated, and will not be restarted at the end of the Caller ID transmission. It is possible to enable an interfering tone while sending a CID Transmission. See the parameter: <u>FSK Interfering Tone Enable</u>.



Generating Sinusoidal Tones & Noise

Clicking the tone generator icon, or selecting [WINDOW] [TONE GENERATOR] for the menu will bring the Tone Generator window to the forefront. The top three check boxes shown in the Tone Generator window are used to enable or disable the two tone generators, Tone #1, Tone #2, and the noise generator. If any one of the signal generators are selected as enabled, the program status line will display TONE ON or FSK MOD.



To change the settings of either tone #1, tone #2, or the noise generator, click the button associated with that tone or noise signal in the "Set Parameters For:" section of the window.

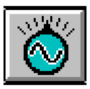
For tone #1 and tone #2, two bar graphs will display the current frequency and amplitude settings for the selected tone. For the noise signal, only an amplitude bar graph will be displayed. Beside the bar graph(s) a text box will show the numeric value for the frequency or amplitude. The settings can be changed by either:

1) clicking the mouse at the desired numeric value on the bar graph,
2) typing a new value into the text box, or
3) using the spin buttons to increment or decrement the current value.

After setting the new frequency and amplitude value, click the ACCEPT button to update the tone generator's settings, or click the CANCEL button to return the settings to the previous values. Note that pressing the ENTER key is equivalent to clicking the ACCEPT button.

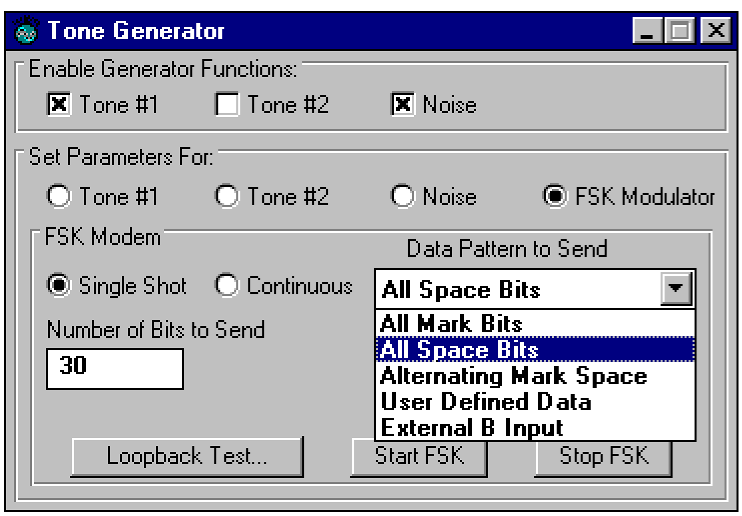
Note: The units for the tone levels can be specified in either dBm, dBV, or mVrms depending on the setting in the More Options panel. To view or change the current setting, select the [MORE OPTIONS] item from the [CONFIGURATION] menu. For more information, see: Section 3-2 Defining the Signal Level Units.

Note: The level of the noise generator is specified as the noise level delivered into the frequency band of 200 Hz to 3200 Hz. Since the noise produced is broadband over a 20 kHz bandwidth, its total signal level will exceed the value specified, if measured over a bandwidth greater than 200 Hz to 3200 Hz.



Generating a FSK Modulated Signal

Clicking the tone generator icon, or selecting [WINDOW] [TONE GENERATOR] for the menu will bring the Tone Generator window to the forefront. The top three check boxes shown in the Tone Generator window are used to enable or disable the two tone generators, Tone #1, Tone #2, and the noise generator. If any one of the signal generators are selected as enabled, the program status line will display TONE ON or FSK MOD.

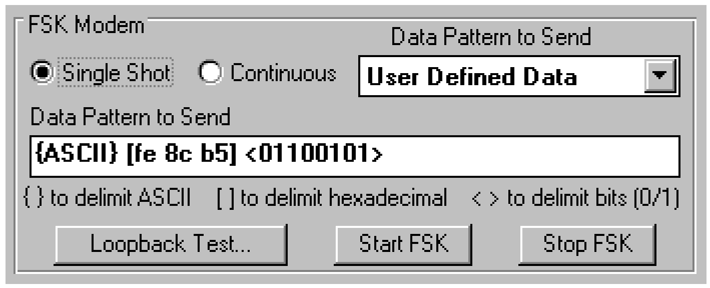


To use the FSK modulator, click on the Button labeled "FSK Modulator" in the "Set Parameters For:" section of the window.

Displayed in the window will be two buttons selecting between "Single Shot" and "Continuous" mode of operation, a "Data Pattern" list box, and a "Number of Bits to Send" text box. The data pattern that is transmitted with the FSK modulator can be all Mark bits, all Space bits, an alternating Mark/Space bit pattern, a user defined bit pattern, or based upon the logic level of a digit input signal. For each pattern (except for External B Input), the modulator can be run in a continuous or single shot mode. Single shot mode transmits the pattern specified, with the specified number of bits once, then it shuts off. In continuous mode, the selected bit pattern will be repeated indefinitely.

For the "User Defined Data" pattern, a text string is entered in the displayed text box which represents the bit pattern to send to the FSK modulator. This bit stream can be specified as ASCII characters sent in a serial fashion (one start bit, 8 data bits LSB first, then one stop bit), or hexadecimal values sent serially MSB first, or a binary pattern sent left to right. ASCII characters are specified by surrounding them with {} brackets, hexadecimal values are surrounded by [] brackets, and binary bit patterns are surrounded by <> brackets. In the example shown below, the data pattern would be ASCII characters 'A', 'S', 'C', 'I', and 'I', followed by 11111110 (fe), 10001100 (8c), and 10110101 (b5), followed by 01100101.

Note: The delimiting brackets used to specify a bit pattern {}, [], <> can **not** be nested inside each other.



The option to use an external digital input as the modulation source can be very useful in testing a FSK decoder in conjunction with bit error rate testers. Commercially available testers can perform a comprehensive suite to tests when coupled with the FSK generator portion of this program. The FSK generator can be programmed over a wide range of mark or space levels and frequencies, along with generating an interfering tone or noise. When using the External B Input, the FSK modulator will generate a mark tone when a high logic level is present at the B input at the rear DB9 connector. Correspondingly, a space tone is generated with a logic low level. For more information on how to use the auxiliary digital inputs and outputs see: Section 5, Auxiliary Digital Inputs and Outputs.

To start the FSK modulator, press the START button. The STOP button will halt the FSK modulator. If the single shot mode has been selected, the FSK modulator will automatically stop when it has reached the end of its specified bit pattern. If the FSK modulator is running, and a Caller ID transmission is started, the FSK modulator and the tone #1, and noise generator will be suspended. Once the Caller ID transmission has finished, tone #1 and the noise signal will be automatically restarted; however, the FSK modulator will not be restarted.

The parameters that specify the FSK modulator characteristics for Caller ID transmissions also apply in this usage of the FSK modulator. As such, use the Advance Setup window to modify the FSK parameters (mark frequency, space frequency, twist level, baud rate, and bit skew). See FSK Modulator Parameters. The output level of the modulator is controlled with the Transmit Level control on the Main Settings window.

Note: The FSK Signal-to-Noise ratio control on the Main Settings Window has no effect when using the FSK modulator in this manner.

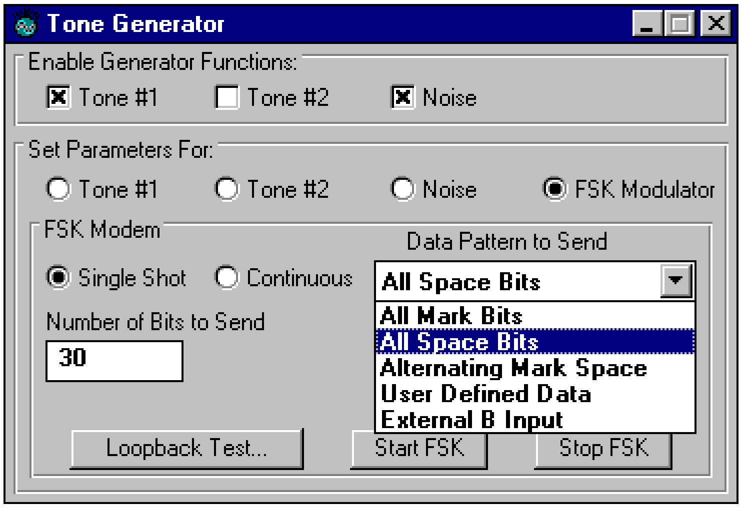
Note: The FSK modulator uses the same resources as tone #2. As such, if tone #2 is enabled, the FSK modulator can not be started. Likewise, if the FSK modulator is active, tone #2 can not be started.

FSK Loop Back Testing

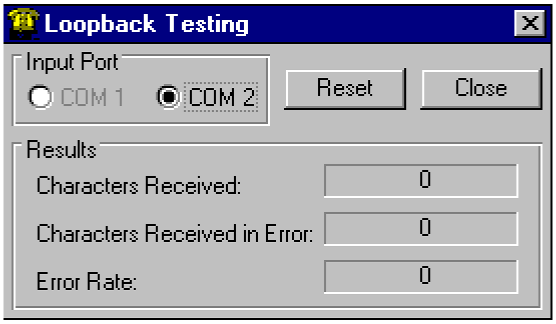
FSK loop back testing is accomplished by setting the FSK modulator to generate an alternating mark/space bit pattern, which is sent to a telephone under test. The output of the FSK demodulator (in the telephone under test) is routed back to the RX Data pin of Com Port 1 or Com Port 2. The data read from the Com Port is then analyzed for errors.

To set up the FSK modulator for alternating mark/space bit pattern, see Generating a FSK Modulated Signal.

Once the FSK modulator is setup, display the loop back results window by clicking on the "Loopback Test" button found in the FSK modulator parameters section of the Tone Generator window, as shown in the diagram below. Clicking the tone generator icon, or selecting [WINDOW] [TONE GENERATOR] for the menu will bring the Tone Generator window to the forefront.



For loop back testing, the FSK modulator is expected to generate an alternating mark/space bit pattern. If this is not the selected data pattern selected for the FSK modulator, a warning message will be displayed when clicking the "Loop back Test" button. The alternating mark/space pattern will be read by the Com Port as an ASCII character 'U'. If any other character is received by the Com Port, it is treated as an error, and will be displayed.

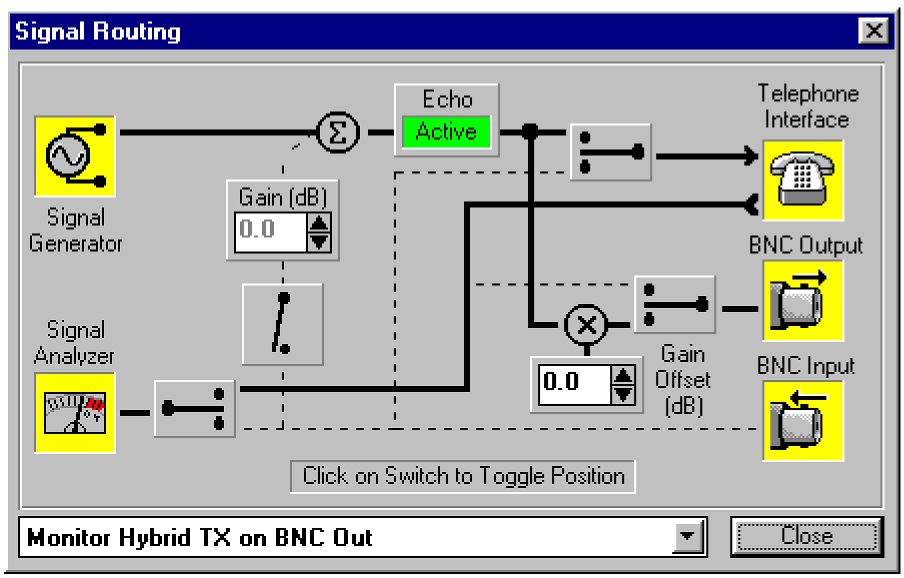


The Com Port for the returning data signal can be selected in the loop back results window, via the two control buttons. However, if a mouse is installed, or other serial device is connected, the Com Port(s) may be disabled, since they are currently being used. When the FSK modulator is started, the number of character received by the Com Port will be displayed, along with the number of characters that are not ASCII 'U'. From these two values, an error rate is calculated. This is simply the number of characters received in error divided by the total number of characters received.

Pressing the RESET button will set the number of characters received, and number of characters received in error to zero. The CLOSE button will close the loop back results window.

■ Section 3-7 Signal Flow & Echo Control

Various options are available for controlling the signal flow through the Telephone Signal Processing Card (TSPC). The current setting can be displayed and changed by opening the Signal Routing window. This is accomplished by either clicking on the Signal Routing icon in the toolbar, selecting [CONFIGURATION] [SIGNAL ROUTING] from the menu, or pressing CTRL-R. As shown below, the window gives a graphical representation of the signal paths between the signal generator, signal analyzer, telephone interface, and external BNC input and output connectors.



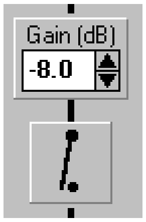
By changing the signal flow, the BNC connectors can used to monitor the transmit or receive signal to and from the telephone. Or to provide for a loop through configuration for the transmit or receive path where externally applied signals or filtering can be added to the signals being generated by the TSPC or analyzed by the TSPC. The BNC connectors can also be configured so that the user has direct control over the signals being sent to and from the telephone interface.

The signal path can be changed by either clicking on the graphical switches, which will toggle their position, or use the drop down list box to select from a predefined list of setups.

Using the programmable gain offset for the BNC output, signals from the tone generators can be increased or decreased by up to 20 dB relative to the signal level at the telephone interface. This feature can be handy when using the BNC output connector for development work, as adding a gain offset eliminates the need for remembering the level differences between the telephone interface and BNC output. Note that the gain offset only effects signals originating from the on board tone generators or the BNC input (via the signal mixer). If the BNC output is routed to the RX path of the telephone interface, the gain offset value has no effect.

External Input Signal Mixer

The external input signal mixer can be used to combine the output from the audio generators with the signal present at the input BNC connector. This can be useful for injecting custom signals, shaped noise, or other interference during a Caller ID transmission. One example would be to inject speech signals in order to perform Talk-Off and Talk-Down tests on the CPE's CAS tone detector.



To enable the signal mixer, click the mouse on the graphical switch. This will toggle the state of the signal mixer between on and off. The mixing gain can be adjusted using the up and down spin controls, or by entering a numeric value. The default state for the mixer is off, and upon restoring program defaults, or changing operational standards, the mixer will be turned off.

The signal mixer can be set to automatically mute itself during Type II Caller ID Transmissions, to simulate the operation of a central office switch. For more information on the use and implications of this option, see: Muting the External Input Signal Mixer (Section 3-2).

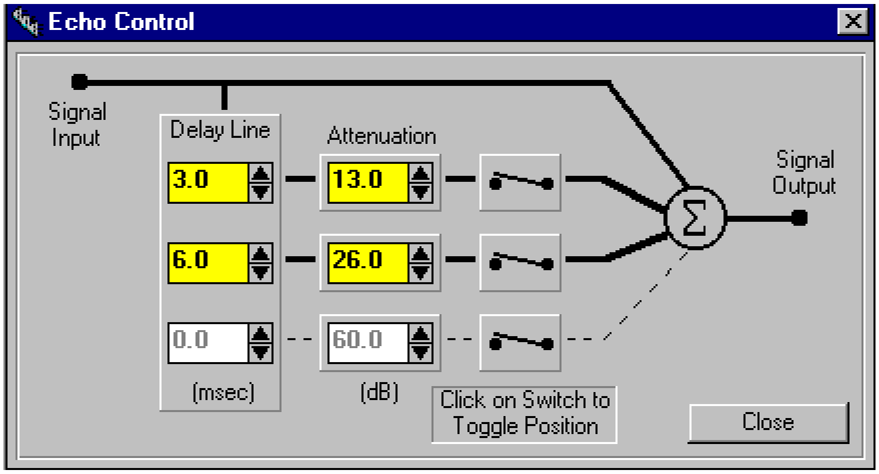
For more information about the signal levels present at the BNC connectors, see: External BNC Signal Levels (Section 5).

Echo Control and Generation

The CID1500 program also has the capability to generate echoes of any signal generated by the software, or injected into the BNC input connector using the external signal mixer. The signals with the added echoes can then be routed to the telephone interface or the BNC output connector.

The following panel is used to control the echo settings. This panel can be made visible by either,

i) selecting the [ECHO GENERATION] item from the [CONFIGURATION] menu, or
ii) clicking the mouse on the Echo toolbar button, or
iii) double clicking the mouse on the echo panel shown in the Signal Routing window.



The Echo Control window displays a graphical representation of the echo parameters. Up to three different echo generations can be enabled, with independent delays and attenuation settings relative to the input signal. To turn on or off a delay path, simply click the mouse on one of the toggle switch icons. This will alternate the echo path between active or inactive. The delay value for any active path can range from 0 to 20 milliseconds, while the attenuation value can range from 0 to 60 dB relative to the input signal. All of the active echo paths are summed together with the input signal to form the output signal. This output signal can be routed to the telephone interface and/or the BNC output.

Note: The delay and attenuation values for each echo path are always relative to the original signal.

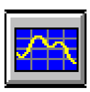
Note: If any of the echo taps are active, the status panel will display the letter 'E" on the status line. This serves as a reminder that at least one echo path is active and will effect the signals generated.

■ Section 3-8 The ACK Tone Analyzer

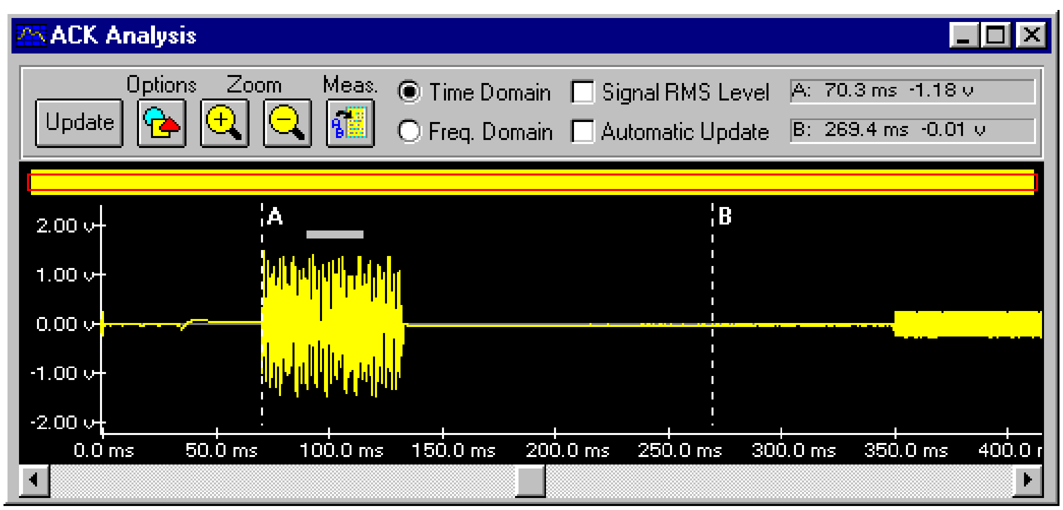
The ACK Tone Analyzer provides an easy means for verifying the timing and signal characteristics of the CPE generated ACK tone. For both the Bellcore and ETSI Type II (CIDCW) Caller ID transmissions, the CPE is required to generate an acknowledge (ACK) tone in response to detecting a CAS or DTAS tone. The ACK tone is usually the DTMF digit A or D, and only after being detected by the simulator, will the FSK data transmission be initiated. The ACK analyzer provides a means to determining the timing relationships of the signals along with analyzing the signal characteristics in both the time and frequency domains. This greatly simplifies the qualification and testing of the ACK tone, as well as giving the user a tool to assist in determining failure modes. The ACK tone analyzer is an option that can be added to any AI 150 package.

Basic Operation

The ACK analyzer functions by recording and the processing the signals generated by the CPE starting at the time the CAS/DTAS tone has ended. Slightly more than 400 msec of data is captured by the program. This data can be viewed much like a digital storage oscilloscope in the time domain, or a Fast Fourier Transform (FFT) can be generated in order to view the frequency domain aspects of the signals.



The ACK analysis window can be displayed by simply pressing the appropriate button on the tool bar, or by selecting the [WINDOW] [ACK ANALYSIS] menu command.



In the example shown above, an ACK tone from a CPE was captured and is displayed in its full span. In this case the ACK tone was generated 70 msec after the end of the CAS/DTAS tone. A small transient signal can be seen to be generated by the CPE at a time of 40 msec. The lower level signal starting at a time index of 350 msec is the beginning portion of the FSK data being transmitted to the CPE.

For closer analysis of the waveform, the graph can be zoomed in or out by pressing one of the two zoom buttons. The top rectangular bar displays the relationship between the span of the data displayed on the graph to the entire data record captured. In the full span view, the rectangular bar runs across the entire length of the graph. As the waveform is zoomed in, the length of the bar will decrease, with its position and size representing the displayed waveform. The bottom scroll bar is used to pan the waveform either left or right. By using the zoom buttons and the scroll bar, any section of the captured data can be displayed. In the time domain view, the horizontal scale is always in units of milliseconds with the time of zero representing the end of the CAS/DTAS pulse (default setting).

Two cursors are provided to assist in making signal measurements. These cursors are labeled A and B. They can be moved to any position on the graph by using the mouse buttons. By pressing and holding the left mouse button, cursor A can be moved to any position. Likewise, pressing and holding the right mouse button controls the position of cursor B. The position of the two cursors, and the data value at that position is always displayed in the upper right corner of the window.

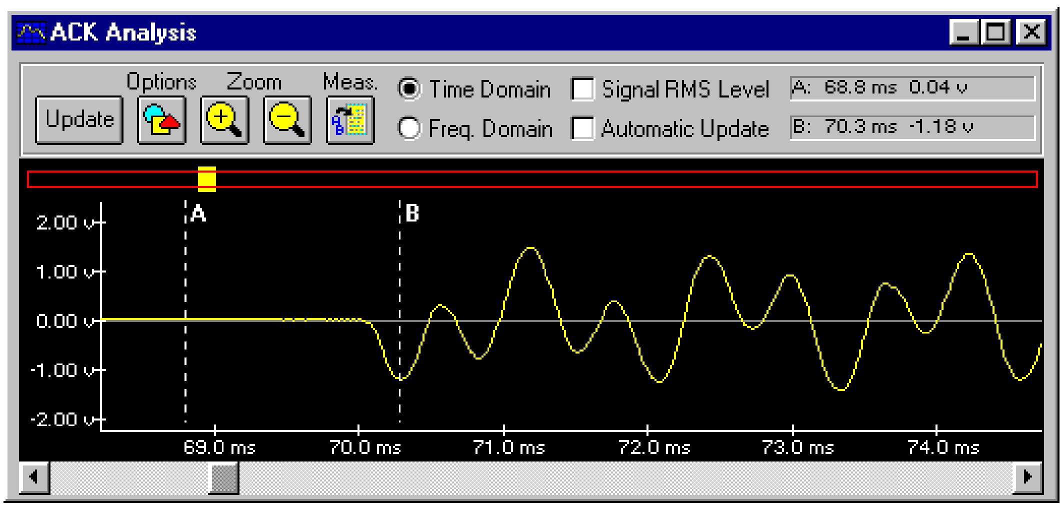
The waveform displayed can be printed by selecting the menu command [FILE] [PRINT ACK ANALYSIS WINDOW]. The printed output will match the same scaling and zoom factor as what is displayed in the ACK analysis window, along with showing the cursor positions and their data values.

Using the Analyzer

The analyzer will capture the signals following the CAS/DTAS tone for every type II Caller ID transmission generated. However, the displayed graph will not normally be updated automatically. This is done for two main reasons. First depending on the speed of the computer, there can be a long processing delay in updating the displayed graph. It is recommended that the PC on which the software resides on is at least a 486DX class machine, as this option is quite intensive in its floating point use. PC's without a floating point processor or a processor speed below 33 MHz will still operate with this option, however the program operation will appear to be quite slow at times. The second reason is that it can be more useful to display only specific events in certain circumstances. For example, it may be of more interest to focus on only the Caller ID transmissions were ACK recognition failed. In this case, only when a missed event occurred would the user update the waveform. In any case, the Update button is used to display any new data. After a Type II transmission, the Update button will be enabled. Pressing this button will cause the display to show the new data.

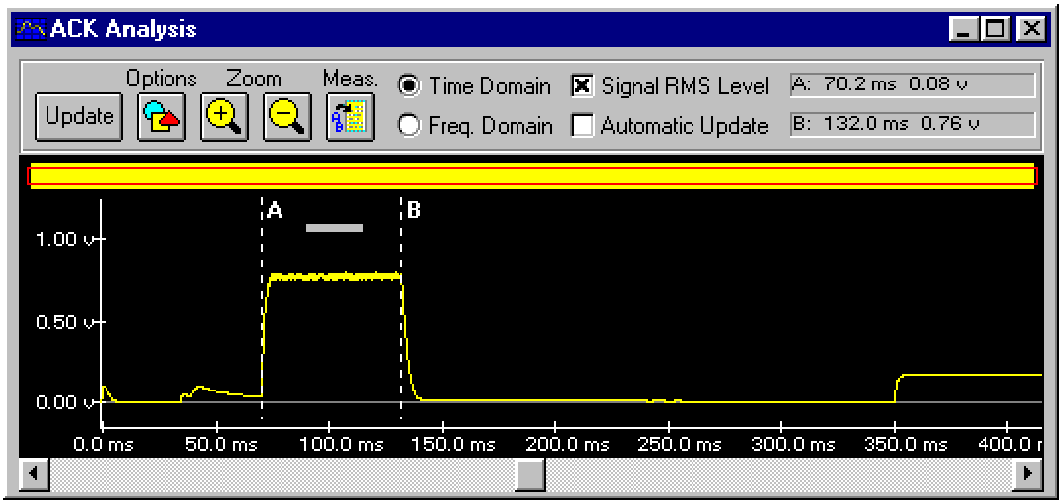
If the user wishes to see the collected data of every type II Caller ID transmission and the computer is reasonable fast enough, it is possible to force an automatic updating of the displayed graph by enabling the Automatic Update option on the ACK analyzer window. With this check box enabled, after every type II transmission, the graph will be updated with the new data.

Once the data has been captured, it can be analyzed in a number of methods. The time domain view is useful in observing the timing relationships between the end of the CAS/DTAS tone and ACK tone. In most cases the beginning portion of the FSK data transmission is also captured. From the time domain view an accurate assessment of the ACK tone's start time and duration can be reached. The time domain view will also reveal any transient events. This can become important in some CPE designs, as large transients generated in muting the CPE may interfere with ACK detection or even possible FSK data reception. The previous figure shows a slight transient at a time of approximately 40 msec after the end of the CAS/DTAS tone. This could be due to muting the CPE's audio path. However no transient signals are shown just before, during, or after the ACK tone. The next figure shows a close up view of the start of the ACK tone. The ACK tone generated by this CPE can be seen to have a very fast and clean start with little distortion.



The time domain view can also show the captured data in terms of its RMS signal level. This is done by enabling the check box labeled Signal RMS Level on the ACK Analyzer window. If this check box is enabled, the vertical axis of the time domain view will represent the RMS signal level of the captured data.

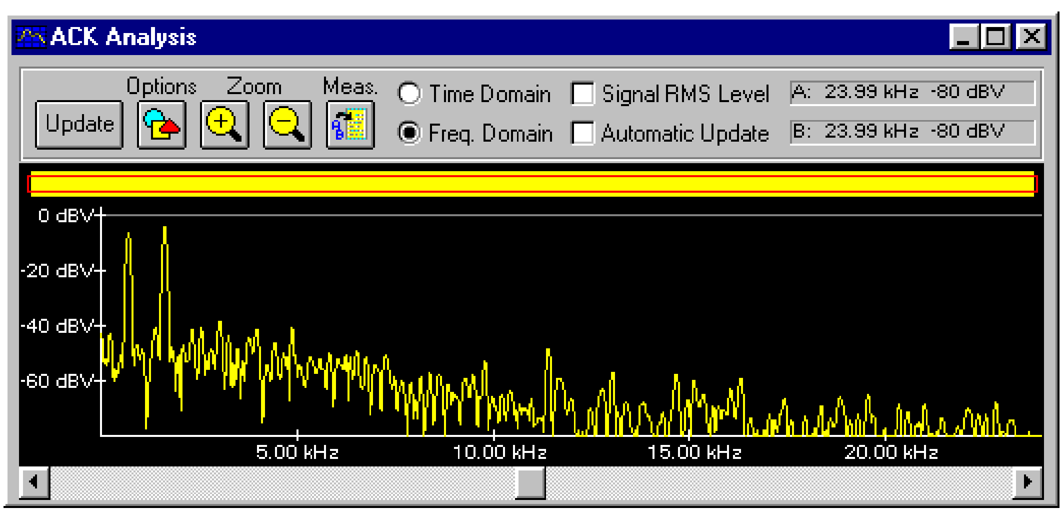
The example below shows the RMS signal level of the ACK tone. This view also indicates a slight transient occurring at approximately 40 msec, and the FSK data signal starting at a time index of 350 msec after the end of the CAS/DTAS tone. Displaying the RMS signal level can be useful in verifying that the ACK signal level is constant over its entire period. Such that there are no slow ramp ups, or fluctuations in the signal level during the ACK tone.



Frequency Domain Analysis

An FFT can be performed on the captured data to reveal its frequency domain characteristics. The default record length of the FFT computed is 1024 data points. This spans a time interval of approximately 21 msec. As such, an FFT can be computed over any 21 msec span. The thick horizontal bar displayed on the time domain graph represents the FFT time span. In the previous figure, the FFT bar was positioned over the center of the ACK tone. Moving the FFT bar is accomplished by simply positioning the mouse on the bar and pressing the left mouse button. While holding the left mouse button down, the FFT bar can be moved to any position on the graph. If the mouse position is not close enough to the FFT bar when the left button is pressed, the cursor A will be selected instead, and its position can be moved. Since the FFT assumes that the data contained within the 21 msec span is periodic, for accurate ACK analysis, the FFT should be taken over a portion of the ACK tone that is transient free and has a stable level.

Using the same data captured as from the previous examples, the FFT bar is positioned over the middle of the ACK tone. Clicking the mouse on the frequency domain button on the ACK Analysis window, changes the graph from the time domain view to the frequency domain. The horizontal axis is now in units of kHz, while the vertical axis is in units of dBV. Like as with the time domain view, the waveform can be zoomed in and out of in the same manner.



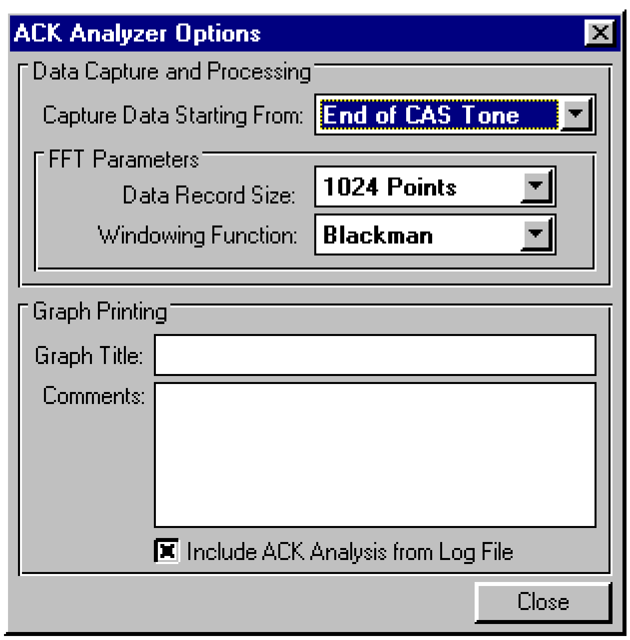
In the above figure, the FFT reveals the spectral components of the ACK tone. The above figure shows the full span view from 0 kHz to 24 kHz. Two distinct spectral peaks are clearly visible representing the two frequency components of the ACK tone. Since this particular CPE responds with a DTMF A tone, the spectral components are at a frequency of 697 Hz and 1633 Hz. Similar to the time domain view, the cursors A and B can be used to measure the level of the two DTMF tones or measure any other spectral peak of interest. By selecting the [FILE] [PRINT ACK ANALYSIS WINDOW] menu command, a printed copy of the FFT data will be generated.

ACK Analyzer Options

Clicking the mouse on the Options button displays a dialog window from which various features can be controlled. As shown in the figure below, the first series of controls affects how the data is captured and processed. The start time, within a Type II Caller ID transmission, can be selected from a range of positions. Normally, the data capture starts when the CAS tone has stopped. However, by clicking the mouse on the appropriate drop-down list box, that start position can be changed to the following:

Start of SAS Tone
Start of CAS Tone
End of CAS Tone (default setting)
When ACK is Detected
Start of FSK Data Transmission

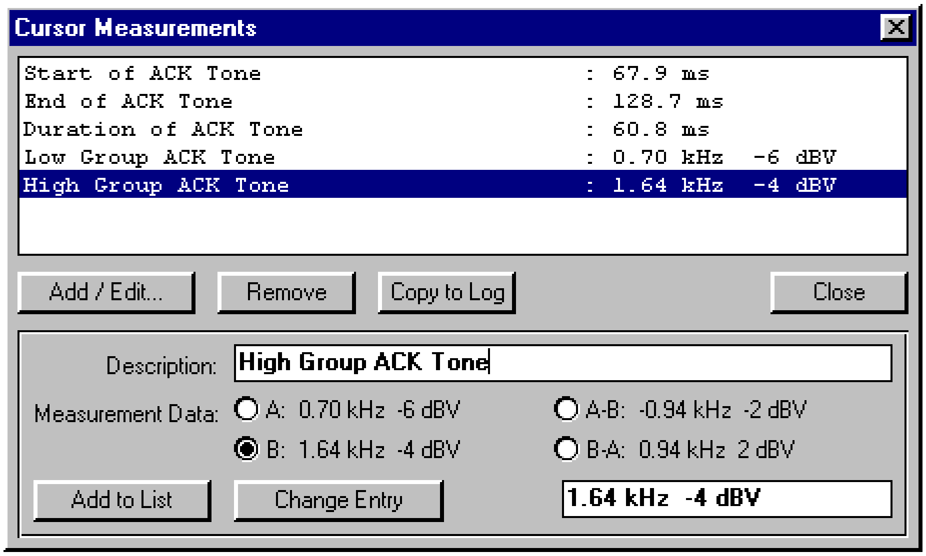
This allows for examining other sections of the Type II Caller ID transmission sequence. The size of the FFT record can be changed from either 1024 points to 2048 points. While the default record size is 1024 points, using a larger record size increases the frequency resolution. However, the record time span also increases from approximately 21 msec to 42 msec. As such, too obtain accurate results any signal measured will have to be stable for a longer time period. In addition to the variable record size, different windowing functions can be specified for the FFT. Aside from the default Blackman window, the Hanning, Hamming, Nutall, and rectangular windows can be used. The different windows provide a basic tradeoff between FFT frequency resolution and side lobe attenuation. The Blackman window provides excellent side lobe attenuation, but its main lobe is wider than the Hanning or Hamming windows. The rectangular window provides the best frequency resolution due to its narrow main lobe; however, its poor side lobe attenuation makes it difficult to resolve other low level signals.



In addition to the data capture and FFT processing options, a graph title and comment can be entered. These two fields will be printed along with the waveform displayed, by selecting the [FILE] [PRINT ACK ANALYSIS WINDOW] menu command. If the check box labeled “Include ACK Analysis from Log File" is enabled, then the ACK tone results from the last Type II Caller ID transmission will be included in the print out.

Recording Cursor Measurements

Cursor measurements can be recorded and included in the ACK analysis printout, or copied to the data log file. Pressing the "Meas." button on the ACK Analysis window displays the Cursor Measurement window. A list of up to 10 arbitrary measurements can be created, edited, or cleared. To add new entries to the list, or to edit any entries, click the mouse on the Add/Edit button. This makes the bottom half of the window visible. From here a description of the measurement can be entered, along with any measurement values.



The values of cursors A and B are shown along with the value of cursor A minus B and cursor B minus A. To use any of the four cursor measurements, simply click the mouse on the appropriate button. This copies the cursor value to the measurement text box at the bottom right corner of the window. Pressing the Add to List button will add a new entry with the specified description and measurement value. If the Change Entry button is pressed, the currently selected measurement will be changed to the new description and measurement value.

All of the measurements defined will be included in any ACK analysis printout. Also, pressing the Copy to Log button will transfer all of the measurements to the data log file.

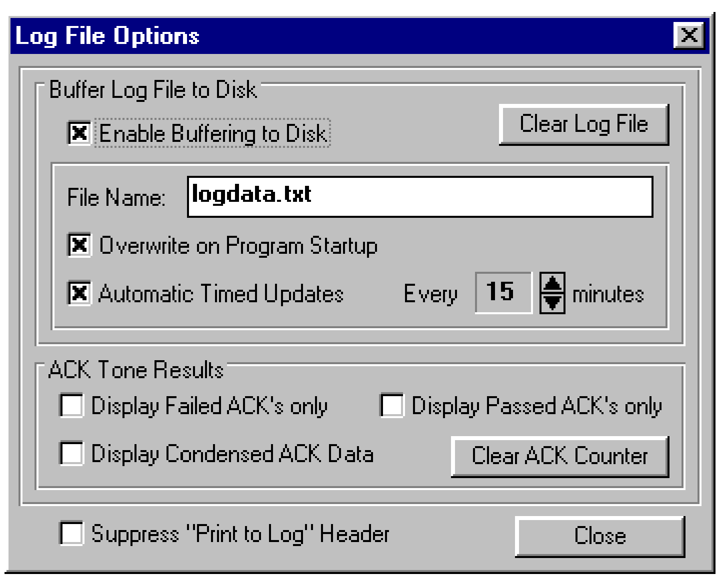
■ Section 3-9 The Data Log

During the course of sending Caller ID transmissions and executing script files, various program messages and analysis data are written to the Data Log window. This information is displayed by clicking the mouse on the Data Log toolbar button, or selecting the [WINDOW] [DATA LOG] command from the menu. The Data Log can be saved to disk as a separate file or sent to a printer by using the [FILE] [SAVE AS DATA LOG] or [FILE] [PRINT DATA LOG] commands from the menu bar. The contents of the log window will also be saved as part of a configuration file. Thus, loading in a saved configuration file, will also restore the Data Log. During the course of running the program, the size of the data log will grow as Caller ID transmissions are sent and script files are executed. There is a maximum size limit to the text of up to 32k bytes. Once this limit has been reached, new messages are still entered into the data log, but the earlier messages will be flushed out. However, all of the data may be buffered to disk if the “Enable buffering to disk” option is enabled.

The Data Log window along with displaying the data generated by the scripting program, also includes an ACK counter. This counter keeps track of the number of Type II Caller ID transmission started, ACK tones received that met the specified criteria, and the number of ACK tones that did not meet the specified criteria, or were never generated by the CPE.

Every time a Type II Caller ID transmission is started, the ACK Total count is incremented by one. Depending on the received ACK tone, either the Passed or Failed count is incremented. The timing, level, and frequency criteria governing the ACK tone reception is controlled by the parameters in the Advanced Setup window.

The Data Log window also includes two buttons that are used to clear the contents of the data log text, or bring up a control panel of options for how data is stored and displayed in the data log. Pressing the Clear Log button will erase the contents of the data log, while pressing the Options button displays the following.



The Log File Options control panel defines if and how the data log will be buffered to a disk file and the display format for Type II Caller ID transmission. Since the data log window will only display the last 32 kbytes of data, an option can be enabled to write all of the data log contents to a file on disk. This will prevent any loss of data that can not be held in the data log window. The log file size is only limited by the space available on the selected storage drive. The file name and path can be specified as well as an option to overwrite the log file at program startup and to force timed updates. If no drive letter is specified for the file name, then the buffering file will be stored in the same directory as the CID1500 application. The option to overwrite the log file at program startup keeps the log file size in check, otherwise it will continue to grow as the program is used, and eventually exceeding the available storage space. Normally, the log file will be written to disk when more than 10 kbytes of new data has been collected. However, the Automatic Timed Updates will force the log file to be updated at least once within the specified time interval. This interval can range from 1 to 30 minutes.

Also included as part of the options panel, are settings to control the display format for Type II Caller ID transmissions. By changing the check boxes labeled “Display Failed ACK’s only” and “Display Passed ACK’s only”, the ACK tone data will be sent to the log file when:

ACK passed or failed (always reports data) (default setting)
ACK failed (only reports failed ACK’s)
ACK passed (only reports passed ACK’s)
Never (does not report any data)

The “Display Condensed ACK Data” setting, if enabled, reduces the amount of information written to the log file to a single text line. Also, the “Print to Log” header, that is included every time the script PRINTLOG command is used, can be enabled or disabled by selecting the appropriate check box at the bottom of the options window.

At the conclusion of either a Type II (CIDCW) transmissions, or ETSI Type I transmissions, various results are written to the Data Log. The Bellcore Type I transmission does not generate any messages, since no signal is received from the CPE. It is possible that some ETSI compatible CPE’s will draw loop current briefly after receiving the alerting tone. This will be detected by the simulator and the start, stop, and duration of the pulse will be displayed in the log file.

An example of the contents of the Data Log file after a Type II (CIDCW) transmission follows below. Included in the message is an analysis of the ACK tone generated by the CPE, and timing information concerning hook switch flashes that may have been generated by the CPE. This detailed analysis is only generated if the “Display Condensed ACK Data” option is not selected. Otherwise only a single line of text is generated, displaying what DTMF digit was received (if any) and the time of the occurrence

```
Type II CIDCW Transmission at: 11:29 AM  Tuesday, May 29, 1997
   ACK Tone Analysis:  Valid ACK tone digit D was received.
      Low group tone:  948.7 Hz @ -3.9 dBm  [0.8 % frequency error from  941 Hz]
      High group tone:  1645.2 Hz @ -2.0 dBm  [0.8 % frequency error from  1633 Hz]
      Tone first detected:  80 to 90 msec after the CAS tone ended
      And analyzed until :  100 to 110 msec after the CAS tone ended
      FSK data was sent, since ACK tone was analyzed before the time limit of:  160 msec
Hook Switch Analysis (Parallel Set Detect)
      Telephone went on hook:  126 msec after start of CAS
      Telephone went off hook:  133 msec after start of CAS
      For a duration of      :  7 msec
```

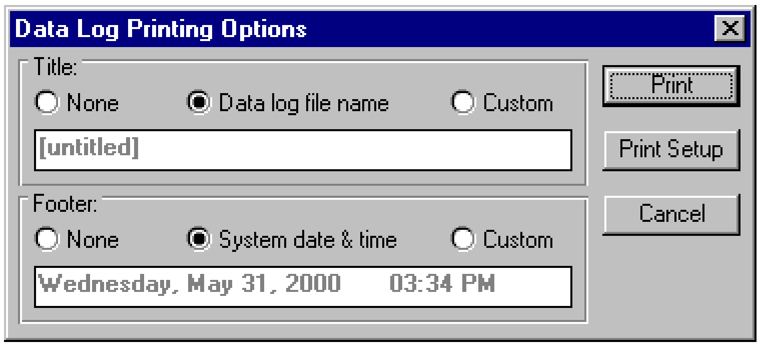
Since the simulator measures the ACK tone frequencies and levels instead of simply detecting a DTMF digit, a longer processing delay is incurred. Conventional DTMF detectors can detect a DTMF tone within a few milliseconds; however, to perform the signal measurements accurately more time is required. As such the simulator requires a minimum ACK tone duration of 30 milliseconds to guarantee detection and signal analysis. As such the timing values presented in the log file are accurate to the ten’s of milliseconds. For more precise timing information, use the ACK tone analyzer, which will display the exact timing of the ACK tone.

Messages are also sent to the Data Log file during the execution of a script file. The start time and end time for the script are logged, as well as any script program warnings that were generated during the execution of the script program. Note that an option exists that will stop the script program execution if a warning is encountered. If this option is set, which is the default state, then no warning message will be sent to the Data Log, since the execution of the script program will stop, and the warning is displayed on the screen to inform the user.

The example message below shows the execution of a script file that generated one warning message during its execution. The warning was generated by setting the parameter FSK_LEVEL to a value beyond its maximum limit. A list of script warning messages can be found in Appendix B.

```
Executing Script Program: [untitled], starting at: 9:43:38 AM  Thursday, November 20, 1997
Warning...  The new parameter value exceeds its maximum allowable value.
            In the line:  Parameter  FSK_LEVEL  +=  6
Script Program ended at: 9:43:40 AM  Thursday, November 20, 1997
-------------------------------------------------------------------------------------------------
```

The contents of the data log can be printed by selecting the [FILE] [PRINT DATA LOG] menu command. This will display a window similar to that shown below. Normally, the file name and system date & time are used for the printed title and footer text respectively. However, this can be changed to any user defined text, or to disable the printing of the title and footer text.



In addition, a script program can change the printing options via the LOG command. Any changes made to the settings are stored within a configuration file.

■ Section 3-10 DC Line Impairments

During the Caller ID transmissions, the CID1500 software can be programmed to induce various DC line impairments. Under certain signaling methods, the central office switch may cause the DC condition on the telephone line to undergo a polarity reversal or an OSI (open switching interval). This is unlike some signaling methods, which use a polarity reversal or OSI to inform the CPE of an impending data transmission, in that the change in the DC line conditions are only an artifact of the central office switch. As such, they do not represent any signaling information, but rather, a condition that the CPE may or may not experience. The Caller ID reception of a CPE should not be affected by the possibility of these impairments.

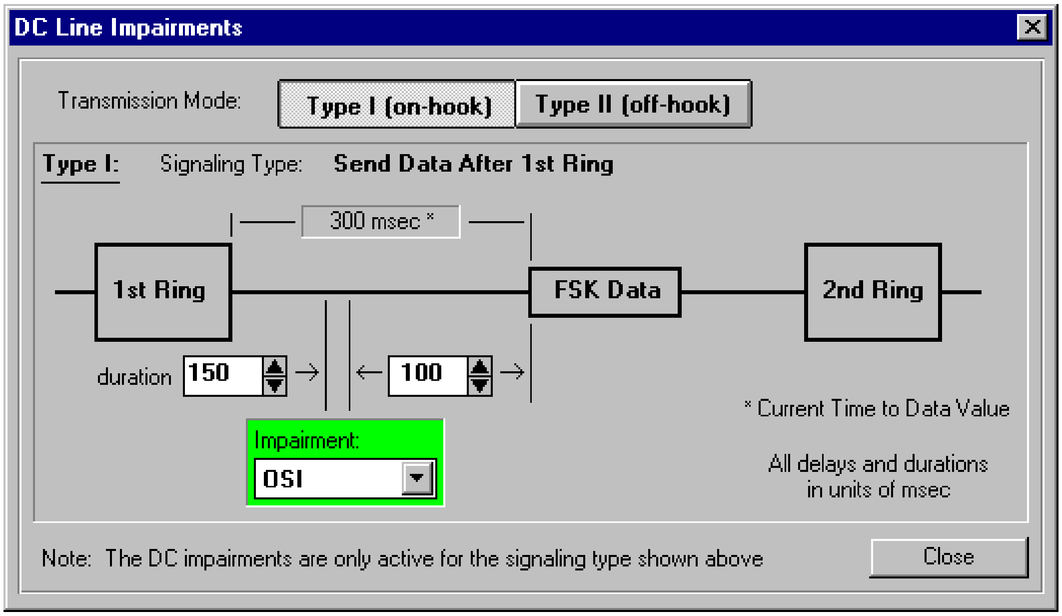
The Bellcore GR-30-CORE (Issue 1, December 1994) allows for a possible OSI between the end of the first ring and the start of the FSK data transmission for Type I (on-hook) Caller ID transmissions. In the Type II (off-hook) case, an OSI may precede the SAS/CAS tone and may also follow after the end of the FSK data transmission.

The DC Line Impairments options within the CID1500 software allow either an OSI or line polarity reversal as an impairment for both Type I and Type II Caller ID transmissions. In the case of Type I, the DC impairments are only active for the signaling method in which the FSK data is sent after the first ring, or the No Ringing signaling method (Bellcore standard). Other Type I signaling types that utilize line reversals, OSI's, DTAS tones, and short ringing bursts are unaffected by the DC Line Impairment settings.

DC Line Impairments active for:

Standard	**Signaling Type**	**Mode**
Bellcore/ETSI	Send Data After 1st Ring	(Type I)
Bellcore	No Ringing	(Type I)
Bellcore/ETSI	Send CAS, Wait for ACK	(Type II)

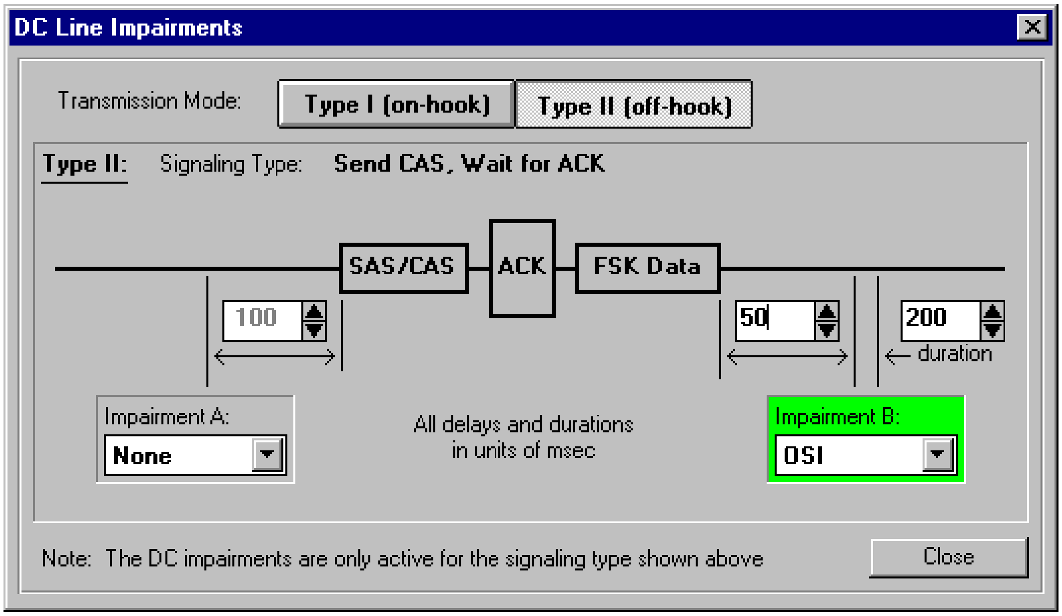
The DC Line Impairments window is displayed by clicking the mouse on the appropriate toolbar button (similar to that shown above), selecting the [CONFIGURATION] [DC LINE IMPAIRMENTS] menu, or pressing the CTRL-I key combination. In either case, a window similar to the following is shown.



A graphic diagram is used to simplify the timing relationship between the various parameters for both the Type I (on-hook) and Type II (off-hook) cases. The two Transmission Mode buttons, at the top center of the window, control whether the Type I or Type II conditions are displayed.

In the Type I case, as shown above, an impairment can be inserted between the end of the 1st ring and the start of the FSK data. The impairment type is selected from the drop-down list as, either an OSI, Line Reversal, or none. For an OSI, two timing parameters must be specified. These are the duration of the OSI and the delay from the end of the OSI to the start of the FSK data. In the case of a line reversal, only the delay from the line reversal to start of the FSK data is required. The time between the end of the first ring and the start of the FSK is determined by the “Time to Data Transmit” parameter in the Advanced Setup window. The value of this parameter is shown in the above figure as 300 msec. It is possible to set the impairment delay and duration values such that the timing will conflict with the “Time to Data Transmit” parameter. In the example figure above, if the OSI duration were to be increased to 250 msec from 150 msec, then the OSI would conflict with the ringing. In these situations, the “Time to Data Transmit” value, will be highlighted in a red background color and a warning message issued when the close button is pressed. If the conflict is not corrected, the DC impairments will not be generated during subsequent Caller ID transmissions.

By clicking the mouse on the Type II (on-hook) Transmission Mode button, the displayed timing graphic changes to a figure similar to below.



In this case, two independent impairments can be selected to occur before the SAS and CAS tones and after the FSK data transmission. As with the Type I, settings, the impairment is selected from the drop-down list box, and can be either an OSI, line reversal, or none.

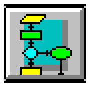
Note: that if a single line reversal is selected as an impairment, then at the end of the Caller ID transmission, a second reversal is performed to ensure that the line polarity is the same as prior to the start of the Caller ID transmission.

■ Section 4 Scripting Language

Basic Operation of Script Programs:
The scripting language capability gives the user a method in which to automate complex test procedures. A script program is composed from a series of commands that can modify, change, and control various aspects of a Caller ID transmission. Other commands are available that control the flow of the script program, such as looping and branching commands.

Script programs can be composed with the built in editor. The Script Editor window includes functions to simplify the generation of script programs, such as context sensitive script command help. Once a script program has been entered, clicking the run script button on the toolbar, or selecting the [TRANSMIT] [RUN SCRIPT FILE] command will first scan the script program for syntax errors, and then begin to execute the script program.

The contents of this section:



The Basics of Script Program Syntax

The following example program is used to explain some of the basics in writing a script program.

Example Program:

```
1:    ***********************************************************
2:    *  Type I CID Test Program
3:    *  May 18, 1997
4:    *  Version 1.0
5:    ***********************************************************
6:
7:    Label START_HERE
8:    Parameter FSK_LEVEL = -36
9:    Start                              ;start CID transmission
10:   PrintScreen "Program Paused, press F6 to cont.  F9 to repeat"
11:   Pause START_HERE                   ;pause program
12:
13:   End                                ;end script program
```

Comment Fields:
If the first non-space character on any line starts with the "*" or ";" character, the entire line will be interpreted as a comment line and ignored during program execution. In the example, the first five lines will be treated as comment fields since they start with the "*" character. If within a program line, a ";" character is encountered, the rest of that line will be treated as a comment field, and ignored during program execution. As shown in lines 9, 11, and 13 in the example program. Blank lines, or lines with only spaces will also be ignored.

Program Commands:
Any line that does not start with "*" or ";" will be treated as what should be a valid command. The commands or data that follow are not case sensitive. So it does not matter if the program is written in upper case, lower case, or a combination of the two. It is important to use spaces (1 or more) between the command words and any data fields in a program line. Spaces are used by the script file interpreter to separate the command and data fields from each other. The only exception is any text that is placed within quotation marks "".

What the Example Program Does:
The example program would execute as follows. The first 6 lines would be ignored since the first 5 start with the "*" character, and the 6th is a blank line. The first command encountered is the LABEL command. This command marks that point in the program with the label START_HERE. Labels are used in conjunction with the PAUSE command. When the program is in the paused state, the user may cause a branch to the label specified in the PAUSE command. Line 8 contains PARAMETER FSK_LEVEL = -36, which sets the transmit level of the FSK modulator to -36 dBm (Bellcore standard) or -36 dBV (ETSI standard). The PARAMETER command can be used to set any transmission parameter to a new value. Next in line 9, the START command starts a Caller ID transmission just like clicking on the Start button in the toolbar. Once the Caller ID transmission has finished, line 10 is executed. The PRINTSCREEN command will display the following text string, enclosed in quotation marks, on the screen. This allows the programmer to prompt the operator while the script file is executing. The PAUSE command on line 11 will suspend operation of the script program. Here the user may restart the script program, terminate the script program, or if the PAUSE command is followed by a label, restart the script program at the specified label. In this case, branching to the label START_HERE will repeat the Caller ID transmission. Line 12 is ignored, since it is blank. Line 13 will stop and terminate the script program with the END command. The END command is not explicitly needed, since the program would automatically end at the last line.

Note: It is important to understand that any changes made to the parameters, or message data will be in effect after the program finishes. In the case above, the FSK transmit level will be set to -36 dBm/dBV when the program finishes. It will not be returned to its value prior to running the script program. The same applies to any changes made to the message data.

How the Script Program Editor works

The script editor provided contains some features to simplify writing script programs. This includes the ability to support multiple text modules, and a command builder that simplifies constructing the script command lines.

Multi-Module Support

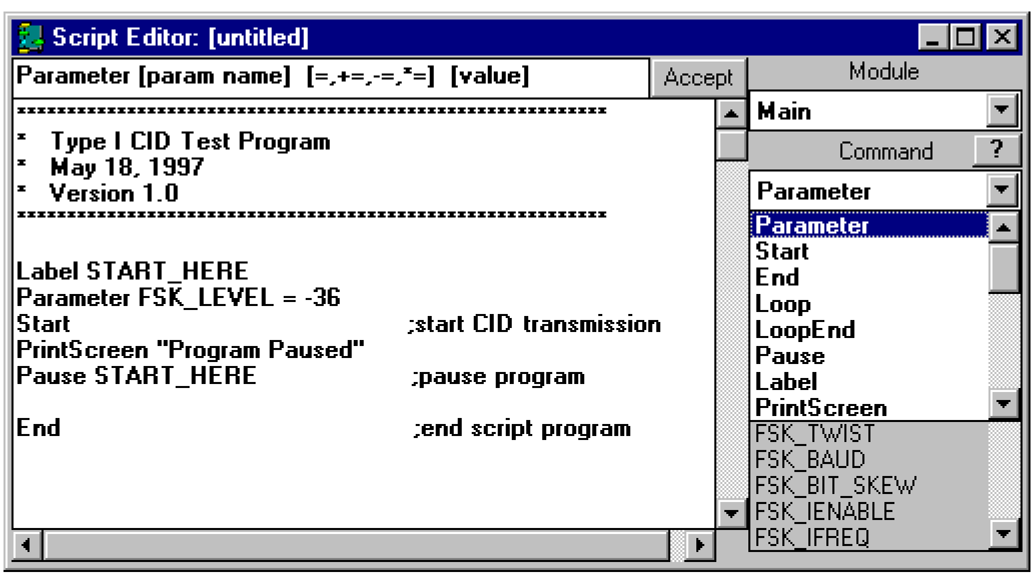
The latest version of the CID1500 software supports multiple text modules while earlier version of the program only allowed for one text module. Complex scripting programs can be divided up into separate modules for ease of code maintenance. Modules can be added, renamed, or removed by using the appropriate menu item, under the [SCRIPTING] menu. The module termed "Main" is always present and can not be removed or renamed. It is this module where script program execution always starts from. To pass program control between the modules the branching command can be used. The BRANCHIF command will transfer the program execution point to the specified label irrespective of what module that label is located in.

Script programs can be composed of up to 32 different modules with each module containing up to 32 kbytes of program text. A new file structure is used to save the multi-module script program to disk; however, the CID1500 program will still be able to read script programs composed on earlier versions for backward compatibility. Older script programs with the .scr extension can only consist of one code module, named Main. Script programs will only be saved in the new file format with the .scx extension. This format supports the multi-module structure.

The module drop-down list in the upper right corner of the script editing window contains a list of all the modules that are part of the current script program. To view or edit another module, simply selected the desired module from the drop-down list.

Command Builder

Below the module list, is the command drop-down list. All of the commands available are displayed here. By selecting a command from the list, a list of operands or other options is displayed below the command drop-down list. The composing text line at the top of the editor is filled with a template of the selected command. In the example below, selecting the PARAMETER command shows all of the parameter names available in the list box and at the same time the command template is shown in the composing text line at the top of the window.



To enter the parameter name, scroll the list box until you find the parameter you wish to modify. Double click on this parameter. The parameter name will be transferred to the composing text line. Now the list box fills with further options for the parameter. As before double clicking on the item in the list box will transfer that selection to the composing text line and complete the template with the correct values.

Once the command has been composed, clicking the ACCEPT button will add the new command line to the program as shown in the large text area.

The operation of the command drop down list box and composing text line is basically the same for all the commands. This command composing feature eliminates the need to memorize the command names, or the structure of the command.

Note: The maximum size of the script editor text window is 32k bytes. The editor will truncate any characters beyond its 32k byte limitation.

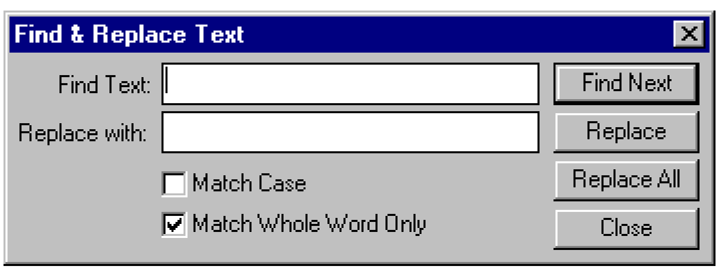
Shortcuts:

1) When selecting the command operand or options in the list box, double clicking or pressing the ENTER key will transfer the selection to the composing text line.
2) For some commands where text information is required to be entered (like PRINTSCREEN), the command data list box will be empty. If the current focus is still with the command data list box, you can type the text field without changing focus to the composing text line.
3) If the current focus is with the command data list box, pressing the right mouse button will transfer the text in the composing line to the script program just like the ACCEPT button.

A short-cut to the on-line help provides direct access to the script command help topics. Clicking the mouse on the button marked with a question mark displays the on-line help topic for the currently selected command. The help button is located within the Script Editor window above and to the left side of the command drop-down list box.

Find & Replace Text

Two menu commands allow for searching and replacing text within the script program editing window. Anytime the cursor is inside the editing area, the [EDIT] [FIND TEXT] and [EDIT] [REPLACE TEXT] menu commands are enabled. Both commands display a window that is similar to the following.



Any text that is highlighted in the script editing window is automatically copied to the “Find Text” field. Clicking the mouse on the Find Next button will start a search for matching text beginning from the cursor’s position in the editing window. If a match is found, the text is highlighted in the editing window. Once the text search reaches the end of the editing window, it will continue from the beginning automatically.

Clicking the mouse on the Replace button will over write any highlighted text in the editing window with whatever text is present in the “Replace With” field. The Replace feature is only available if the [EDIT] [REPLACE TEXT] command was selected.

The Replace All button will automatically start from the beginning of the script editing window and search for matching text. If a match is found, it replaces it with the text in the “Replace With” field. This process is continued until the end of the editing window is reached.

Normally, the text search ignores differences between lower case and upper case letters. By selecting the “Match Case” check box, the search will only find text if the case of each letter matches. The “Match Whole Word Only” check box, if enabled, ignores word fragments and only finds text that matches as complete words. For the purposes of the search, words may contain letters, numbers, and the underscore character.

Script Command Reference

The script command reference describes the function of the script command words. All of command words currently supported are listed below.

Command Word List:

START	Starts a Caller ID transmission
PAUSE	Suspends operation of the script program
END	Ends operation of the script program
LOOP	Marks the start of a loop within the program
LOOPEND	Marks the end of a loop within the program
IF-ELSE-ENDIF	Conditional program execution
LABEL	Defines a label within the program for branching
BRANCHIF	Performs a branch to a label
CALL	Jump to a subroutine
RETURN	Return from a subroutine
DECLARE	Define a script variable
ASSIGN	Assign a value to a script variable
INPUT	Display a user prompt for input
PARAMETER	Changes any of the transmission parameter values
MODE	Sets the Caller ID transmission mode
SIGNALING	Sets the Caller ID signaling mode
MESSAGE	Sets the Caller ID message type
PACKET	Enables/Disables or sets the value of data packets
SEGMENT	Changes the time or number of bits in a segment
STOPBITS	Sets the number of stop bits
MARKBITS	Sets the number of mark stuffing bits
SET	Sets various program options
CLEAR	Clears or restores to default various settings
LOG	Change the settings of the data log file
NETTONE	Execute a network tone script file
DCIMP	Modify DC Line Impairment settings
PRINTSCREEN	Displays a text string on the screen for user prompting
PRINTLOG	Writes a text string to the log file
PRINTFILE	Writes a text string to an ASCII text file
READFILE	Read data from a file into a script variable
DELETEFILE	Erase a file
CHANGEBYTE	Changes any byte of any packet to any value
CHANGEBIT	Changes any bit of any segment
TONEGEN	Controls the dual tone and noise generator
FSKGEN	Controls the FSK modulator
WAIT	Suspends the script program for a specified time
DO	Perform specific operation

Script Command: START

Description:
The START command initiates a Caller ID transmission. Its action is identical to that of clicking on the start icon in the toolbar, selecting from the menu [TRANSMIT] [START TRANSMISSION], or pressing F5.

Syntax:
START

Example:
The example program simply starts one Caller ID transmission using the START command, then ends.

```
************************************************************
*   Type I CID Test Program
*   May 18, 1996
*   Version 1.0
************************************************************

Start                                   ;start CID transmission

End                                     ;end script program
```

Script Command: PAUSE

Description:
The PAUSE command is used to suspend execution of the script program. While program execution is suspended in pause mode, the user can restart execution by pressing the START button or press F6, stop and reset the script program by pressing the STOP button, or branch to the label specified in the pause command by pressing F9. The label field of the PAUSE command is optional. However, if a label has been specified is must not contain any spaces in its name. Also, the same label name must be defined elsewhere within the program via the LABEL command.

A modification of the basic PAUSE command is to add “[EXT]” as a suffix. By adding this keyword at the end of the PAUSE command, the script program can also be restarted by asserting a high logic level at the digital input ports at the rear DB9 connector of the TSPC. With the [EXT] suffix, a logical high at Input A will restart the script program execution as if the Start button was pressed. Likewise, if a label has been specified, a logical high at Input B will branch to the label and restart the script program execution. For more information on the digital input and outputs, see Section 5: Auxiliary Digital Inputs and Outputs.

Syntax:
PAUSE
PAUSE [label]
PAUSE [EXT]
PAUSE [label] [EXT]

Example:
The example program starts the Caller ID transmission, then enters pause mode. If F9 is pressed, the program will branch to the BEGIN_PROGRAM label in line 1 and start executing from there. If F6, or the start button, had been pressed the program would continue after line 5. The second pause command in line 8, has no label specified, so pressing F9 will have no effect.

```
1:    Label  BEGIN_PROGRAM
2:    Start                          ;start CID transmission
3:
4:    *  when paused press F6 to continue, or F9 to branch to label
5:    Pause  BEGIN_PROGRAM           ;pause command with label
6:
7:    *  when paused press F6 to continue
8:    Pause                          ;pause command without label
9:    End                            ;end script program
```

Script Command: END

Description:
The END command stops execution of the script program. Once stopped the script program is reset and can only be started from the beginning. The END command is optional in that script program execution will automatically be stopped at the last line of the script program. END can be used in the middle of a script program to stop execution of the remainder of the program.

Syntax:
END

Example:
The example program starts a Caller ID transmission due to the START command in line 2, then stops at the END command in line 4. The START command in line 7 will not be executed.

```
1:    Label  BEGIN_PROGRAM
2:    Start                                ;start CID transmission
3:
4:    End                                  ;end script program
5:
6:    *  The following commands will not be executed
7:    Start                                ;start CID transmission
```

Script Command: LOOP

Description:
The LOOP command in conjunction with the LOOPEND command allow for certain sections of the program to be repeated a specified number of times. The LOOP command marks the start of the loop, while the LOOPEND command marks the end of the loop. Up to 50 loops can be used in any script program module. Loops can also be nested within each other. An error will be reported if a LOOP command is present without a corresponding LOOPEND command.

Syntax:
LOOP (number of loops)

Example:
The example program contains two loops in a nested fashion. The program will start 32 Caller ID transmissions with FSK transmit levels ranging from -10 to -58 and FSK signal-to-noise ratios ranging from 40 dB to 10 dB. Lines 5 and 12 mark the outer loop with the LOOP and LOOPEND commands. This outer loop will be executed 8 times as specified in line 5. The inner loop is marked by lines 7 and 10. This loop will be executed 4 times for every outer loop cycle. As such, lines 7 to 10 will be executed a total of 32 times.

```
1:    *  This program will cycle through various FSK transmit levels and
2:    *  Signal-to-Noise ratios as an example of the loop command.
3:
4:    Parameter FSK_LEVEL = -10          ;set FSK transmit level
5:    Loop 8                             ;repeat outer loop 8 times
6:       Parameter FSK_SNR = 40          ;set FSK SNR level
7:       Loop 4                          ;repeat inner loop 4 times
8:          Start                        ;start CID transmission
9:          Parameter FSK_SNR -= 10      ;decrease SNR by 10 dB
10:      Loopend                         ;inner loop end mark
11:      Parameter FSK_LEVEL -= 12       ;decrease level by 12 db
12:   Loopend
13:
14:   End                                ;end script program
```

Script Command: LOOPEND

Description:
The LOOPEND command in conjunction with the LOOP command allow for certain sections of the program to be repeated a specified number of times. The LOOP command marks the start of the loop, while the LOOPEND command marks the end of the loop. Up to 50 loops can be used in any script program module. Loops can also be nested within each other. An error will be reported if a LOOPEND command is present without a corresponding LOOP command.

Syntax:
LOOPEND

Example:
The example program contains two loops in a nested fashion. The program will start 32 Caller ID transmissions with FSK transmit levels ranging from -10 to -58 and FSK signal-to-noise ratios ranging from 40 dB to 10 dB. Lines 5 and 12 mark the outer loop with the LOOP and LOOPEND commands. This outer loop will be executed 8 times as specified in line 5. The inner loop is marked by lines 7 and 10. This loop will be executed 4 times for every outer loop cycle. As such, lines 7 to 10 will be executed a total of 32 times.

```
1:     * This program will cycle through various FSK transmit levels and
2:     * Signal-to-Noise ratios as an example of the loop command.
3:
4:     Parameter FSK_LEVEL = -10          ;set FSK transmit level
5:     Loop 8                             ;repeat outer loop 8 times
6:        Parameter FSK_SNR = 40          ;set FSK SNR level
7:        Loop 4                          ;repeat inner loop 4 times
8:           Start                        ;start CID transmission
9:           Parameter FSK_SNR -= 10      ;decrease SNR by 10 dB
10:       Loopend                         ;inner loop end mark
11:       Parameter FSK_LEVEL -= 12       ;decrease level by 12 db
12:    Loopend
13:
14:    End                                ;end script program
```

Script Command: IF-ELSE-ENDIF

Description:
A series of commands can be used to implement conditional code execution. These commands are: IF, ELSE, and ENDIF. Using the command syntax shown below, if the specified condition is evaluated as TRUE, then any statements following the IF command are executed. In the alternative case where the condition is evaluated to FALSE, then only statements following the ELSE command are executed. The ENDIF command is used to mark the end of the conditional statements. Though the ELSE command is optional, the ENDIF command is required for every IF command.

The IF command may be nested to a maximum of 32 levels deep. Beyond this, an error will be generated.

Syntax:
```
IF  (condition)  THEN
   [ optional statements to execute if condition is TRUE ]
[ELSE
   [ optional statements to execute if condition if FALSE ]]
ENDIF
```

The possible conditions are:

ALWAYS	Always evaluate to TRUE
ACKPASS	Evaluates to TRUE, if the last ACK was valid
ACKFAIL	Evaluates to TRUE, if the last ACK was invalid
ONHOOK	Evaluates to TRUE, if the CPE is on-hook
OFFHOOK	Evaluates to TRUE, if the CPE is off-hook
INPUTA	Evaluates to TRUE, if digital input A is high
INPUTB	Evaluates to TRUE, if digital input B is high
NOTinputA	Evaluates to TRUE, if digital input A is low
NOTinputB	Evaluates to TRUE, if digital input B is low
NetTone	Evaluates to TRUE, if a NetTone file is running
NOTnetTone	Evaluates to TRUE, if a NetTone file is not running
(variable)	Evaluates the script variable as either TRUE or FALSE

If using a declared script variable as the condition, it will be evaluated according to its data type. If it is a numeric variable, it is evaluated as TRUE for non-zero values, otherwise it represents FALSE. For string variables, the condition is evaluated as TRUE if the text contained within the variable is “TRUE”, otherwise it represents FALSE.

Example:
The example program uses two nested IF statements to first verify that a connected CPE is off-hook. If so, it starts a Type II Caller ID transmission and then displays to the user if the ACK tone was received or not.

```
If  OffHook  Then
   Printscreen  "The CPE is On-Hook:  Starting Type II CID"
   Start
   If  ACKPass  Then
       PrintScreen  "Type II ACK received"
   Else
       PrintScreen  "*** ACK tone not received from CPE"
   EndIf
Else
   Printscreen  "The CPE is Off-Hook:  Can't send Type II CID"
EndIf
```

Note: if a BRANCHIF command is used within an IF command's conditional statements, it's target label must not be located beyond the current ENDIF statement or prior to the current IF statement, otherwise it will cause a error.

Script Command: LABEL

Description:
The LABEL command is used to mark a location in a script program to branch to using the PAUSE command. The label specified must contain no spaces in its name. A maximum limit of 200 labels for the script program exists.

Syntax:
LABEL (label name)

Example:
The example program starts the Caller ID transmission, then enters the pause mode. If F9 is pressed, the program will branch to the BEGIN_PROGRAM label in line 1 and start executing from there. If F6, or the start button, had been pressed the program would continue after line 5. The second pause command in line 8, has no label specified, so pressing F9 will have no effect.

```
1:    Label  BEGIN_PROGRAM
2:    Start                          ;start CID transmission
3:
4:    *  when paused press F6 to continue, or F9 to branch to label
5:    Pause  BEGIN_PROGRAM           ;pause command with label
6:
7:    *  when paused press F6 to continue
8:    Pause                          ;pause command without label
9:    End                            ;end script program
```

Script Command: BRANCHIF

Description:
The BRANCHIF command can be used to execute a program branch based on various conditions. If the specified condition is evaluated as true, then the script program will branch to the specified label. The target label can occur in the same module, or in any of the other possible script modules. The labels are defined using the LABEL command. If a script variable is used as the condition, a branch occurs if the variable is set to TRUE. For numeric variables, any non-zero value is considered true, while for string variables, the variable must contain the text "true" (case insensitive) for the branch to occur. The syntax for the BRANCHIF command is as follows:

Syntax:
BRANCHIF   (condition)   (label)

The possible conditions are:

ALWAYS	Causes an unconditional branch to the label
ACKPASS	Causes a branch if the last ACK received was valid
ACKFAIL	Causes a branch if the last ACK received was not valid
ONHOOK	Causes a branch if the CPE is currently on-hook
OFFHOOK	Causes a branch if the CPE is currently off hook
INPUTA	Causes a branch if digital input A is at a high state
INPUTB	Causes a branch if digital input B is at a high state
NOTinputA	Causes a branch if digital input A is at a low state
NOTinputB	Causes a branch if digital input B is at a low state
NetTone	Causes a branch if a NetTone file is running
NOTnetTone	Causes a branch if a NetTone file is not running
(variable)	Causes a branch if the specified variable is TRUE

Example:
The example program prompts the user to take the CPE off-hook to start a Type II Caller ID transmission. Before the START command in line 6, the BRANCHIF command in line 4 checks to make sure the CPE is off-hook. If it is on-hook, then the program branches back to the beginning. Assuming the CPE is off-hook, after the Caller ID transmission, if the received ACK tone was invalid, a message is shown to the user. Otherwise the program ends. The BRANCHIF command in line 7 is used to check to see if the ACK tone was valid or not. If invalid, it causes a program jump to the label "No_Good".

```
1:     Label  BEGIN_PROGRAM
2:     PrintScreen  "Take CPE off-hook and press F6 to start"
3:     Pause
4:     Branchif   OnHook  Begin_Program
5:
6:     Start                          ;start CID transmission
7:     Branchif   ACKfail  No_Good
8:     End                            ;end script program
9:
10:    Label  NO_GOOD
11:    PrintScreen   "Unsuccessful.  Press F9 to retry."
12:    Pause   Begin_Program
13:    End                            ;end script program
```

Script Command: CALL

Description:
The CALL command is used to pass program control the point in the script program marked by the specified label. The labels are set by the existing LABEL command. Script program execution will continue from the specified label until a RETURN command is encountered. At that point, the execution point will be returned to the next statement after the original CALL command. The two commands work together to form script program subroutines. Subroutines can be nested up to 32 levels deep.

Syntax:
CALL (label)

Example:
The example program executes a series of similar tests by combining the basic testing loop into a subroutine. The subroutine "Do_Test" will send 5 Caller ID transmission with varying FSK levels between -12 and -36 dBm. The subroutine is called three times, in lines 2, 4, and 6, with different FSK signal parameter values. At each CALL command, program execution is transferred to the label "Do_Test" at line 8. The RETURN command at line 15 then returns the program execution to the next line following the CALL command. In this case, it would be lines 3, 5, and 7.

```
1:    Parameter   FSK_MFREQ  =  1188
2:    Call   Do_Test
3:    Parameter   FSK_MFREQ  =  1212
4:    Call   Do_Test
5:    Parameter   FSK_BAUD = 1188
6:    Call   Do_Test
7:    End
8:
9:    Label   Do_Test
10:   Parameter   FSK_LEVEL = -12
11:   Loop 5
12:      Start
13:      Parameter   FSK_LEVEL -= 6
14:   LoopEnd
15:   Return
```

Script Command: RETURN

Description:
The RETURN statement is used to mark the end of subroutines. When this command is encountered, program execution will return to the statement after the last CALL command. If a RETURN command is processed before any CALL commands, an error will occur and the script program will be stopped.

Syntax:
RETURN

Example:
The example program executes a series of similar tests by combining the basic testing loop into a subroutine. The subroutine “Do_Test” will send 5 Caller ID transmission with varying FSK levels between -12 and -36 dBm. The subroutine is called three times, in lines 2, 4, and 6, with different FSK signal parameter values. At each CALL command, program execution is transferred to the label “Do_Test” at line 8. The RETURN command at line 15 then returns the program execution to the next line following the CALL command. In this case, it would be lines 3, 5, and 7.

```
1:    Parameter   FSK_MFREQ  =  1188
2:    Call   Do_Test
3:    Parameter   FSK_MFREQ  =  1212
4:    Call   Do_Test
5:    Parameter   FSK_BAUD = 1188
6:    Call   Do_Test
7:    End
8:
9:    Label   Do_Test
10:   Parameter   FSK_LEVEL = -12
11:   Loop 5
12:      Start
13:      Parameter   FSK_LEVEL -= 6
14:   LoopEnd
15:   Return
```

Script Command: DECLARE

Description:
The DECLARE command is used to create a script variable, which can then be used throughout the script program. The variable can be defined as either a numeric or string type. String variables contain ASCII text, while the numeric type contain floating point numbers. When using variables with other scripting commands, the variable type (string or numeric) must match the command requirements. Some commands only work with numeric variables, while others require string variables, and some commands can function with either. The variable name specified can contain up to 32 characters. However, the name must start with a letter and can only contain letters, numbers, and the underscore symbol. Also, the script variable names are case insensitive. The DECLARE command can be placed anywhere within the script program, and does not need to positioned before any statements using the variable.

Syntax:
DECLARE   Numeric   (variable name)
DECLARE   String   (variable name)

Example:
The following example program declares one numeric variable, called Num_TwoPi and one string variable called Str_TwoPi, in the first two program lines. Line 4 is used to assign the value of 6.283 to the numeric variable Num_TwoPi, while line 5 sets the string variable Str_TwoPi to “The value 2pi is equal to: 6.283”. Finally line 7 displays the contents of the Str_TwoPi variable to the screen.

1:   **Declare   Numeric   Num_TwoPi**
2:   **Declare   String   Str_TwoPi**
3:
4:    Assign Num_TwoPi To 3.1416 * 2
5:    Assign Str_TwoPi To “The value 2pi is equal to: “ + [Num_TwoPi]
6:
7:    Printscreen   [Str_TwoPi]

Script Command: ASSIGN

Description:
The ASSIGN command is used to change the contents of a variable to the result of an expression. The expression can be as simple as a number or text string, or more complex using other variables and operators. If the target variable type is defined as numeric, then all of the operands used in the expression must be numeric in nature. Likewise, for string variables, the operands in any expression must all be strings. The variable types can not be mixed together. However, when literals are used in the expression, the ASSIGN command will attempt a conversion to the appropriate variable type. A numeric variable can be converted to string by enclosing the variable name in square brackets [ ]. As such, numeric values can be converted to string for use in any place where a string variable or literal is required. If operators are used in the expression, care must be taken in the order of operations. The ASSIGN command evaluates expressions from left to right with NO precedence of operators. For example, the expression 5 + 2 * 3 would evaluate to 21 taken from left to right. Also, for string variables the only the +, =, and <> operators are defined. Any other operator contained in a string expression causes an error. Comparison operators such as =, <>, >, >=, < , <= result in a true or false value which can be used by the BRANCHIF command.

Syntax:
ASSIGN   (variable name)   TO   (expression

Expression:
(operand)
(operand)   (operator)   (operand)
(operand)   (operator)   (operand)   (operator)   (operand)
and so on...

Operand: (variable name)
(numeric literal)
(string literal)

Operators:

+	(numeric add or string concatenate)
-	(numeric subtract)
*	(numeric multiply)
/	(numeric divide)
=	(numeric or string equality comparison)
<>	(numeric or string inequality comparison)
>	(numeric greater than comparison)
>=	(numeric greater or equal than comparison)
<	(numeric less than comparison)
<=	(numeric less than or equal comparison)
AND	(numeric logical and operator)
OR	(numeric logical or operator)

Example:
The following shows a number of examples using the ASSIGN command. Note that lines 12 and 19 will cause a script program error, due to a mix of string and numeric types. All operands in an expression must be either numeric or string in nature.

```
Declare   Numeric  N1
Declare   Numeric  N2
Declare   Numeric  N3
Declare   Numeric  N4
Declare   String  ST1
Declare   String  ST2

Assign  ST1  To  "Two Pi is:  "           ;set ST1 equal to "Two Pi is:  "
Assign  N1  To  3.14                      ;set N1 equal to 3.14
Assign  N2  To  N1 * 2                    ;set N2 equal to 6.28
Assign  ST2  To  3.14                     ;set ST2 to the string "3.14"
Assign  ST2  To  ST1  +  N2  ;INVALID since N2 is numeric
Assign  ST2  To  ST1  +  [N2];this works:  ST2 = "Two Pi is:  6.28"
Assign  N3  To  N2  >  N1                 ;N3 is set to true (-1)
Assign  N4  To  N2  =  N1                 ;N4 is set to false (0)
Assign  N4  To  N3  AND  N4 ;N4 is set to (true AND false) = false
```

Along with variables that are defined using the DECLARE command, all of the internal program parameters can be used as operands, such as the FSK_LEVEL parameters shown below.

```
Assign  N1  To  6 + FSK_LEVEL         ;use built in parameter values
Assign  ST1  To  [FSK_LEVEL]          ;convert parameter value to string
Assign  N2  To  LINE_IMP     ;INVALID, since LINE_IMP is string only
Assign  ST2  To  [LINE_IMP] ;this works, since ST2 is a string
```

Assuming that the FSK level is set to -13 dBm, then N1 will be set to the value of -7. ST1 is set to the string conversion of the FSK level, which would be "-13 dBm". The string conversion of any internal parameter will always include the units of the parameter. The statement with variable N2 causes an error, since some parameters like the telephone line impedance are not numeric in nature, but treated as strings. As such, they can not be assigned to a numeric variable. However, they can be assigned to a string variable as shown with ST2.

Script Command: INPUT

Description:

The INPUT command is used to allow user input of numeric or text string information. This input data is assigned to variables and can be used to control program flow or set various program settings. Two fundamentally different types of user prompts can be displayed to the user. The first presents a text field for the user to enter information from the keyboard, while the second displays up to four buttons from which the user can select one. The command syntax for the INPUT command is given below. The style keyword determines the type of prompt to display to the user. This can be either "FPnumber", "Integer", TextLine", "MultiLine", "Buttons", and "PopUpMenu". The first three styles are very similar in that the user is displayed a single line text box in which to enter data. However, the FPnumber style will check to insure that that value entered is numeric and bound between the specified minimum and maximum limits. The Integer style does a further check to see if the value entered is an integer. If non-numeric, or text string information is needed, use the TextLine or MultLine styles. The TextLine style allows the user to enter up to 1 line of information, while the MultiLine style allows multiple text lines to be entered. In both cases, a minimum and maximum string length can be specified as part of the command.

Following the style type in the command line, a variable name must be specified. The variable should only be a numeric type with the FPnumber and Integer prompt styles. String variables can be used with any style. If a numeric variable is used with the TextLine or MultiLine style and the user enters a non-numeric value, a run time error will be issued and the script program stopped.

After specifying the variable name in the command line, a caption to display is enclosed in quotation marks. The caption is displayed on the user prompt and can be used to explain to the user what information is being requested. Finally, following the caption, the minimum and maximum limits for the input data is given. For the FPnumber and Integer styles, the entered value is compared with the minimum and maximum limits. If outside the range, the user will be prompted to enter another value. For the TextLine and MultiLine styles, the minimum and maximum limits defined the number of characters allowed in the returning text string. The user can not enter more characters than the maximum limit, but any string less than the minimum limited will cause a prompt to the user indicating that more characters are required.

If the user selects the CANCEL button on the prompt window, then the variable will be set to a null string for the TextLine and MultiLine styles or the value of zero for the FPnumber and Integer styles.

The Button style is significantly different from the others and requires a slightly different syntax. With this style the user prompt will show from 1 to 4 buttons with programmable captions and colors. When the user presses one of the buttons, the button number pressed is returned to the specified variable in either a numeric or string format. The syntax for this style of user prompt requires a text string describing the button captions following the variable name. Each caption should be separated by a comma, with up to four captions allowed. Optionally, the color of each button can be defined by following the captions with the | character and a number representing the color for each button. As with the captions, the color for each button is separated by a comma.

Finally, the PopUpMenu style, displays a simple menu with 1 to 20 captions. If the user selects one of the selections, the selection number (from 1 to 20) is returned in the specified variable. In the case the user does not click the mouse on the pop up menu, the variable returns the value of zero.

Syntax:
INPUT (style) (variable name) "caption" (min) (max)

Style:
- FPnumber (floating point number between min and max)
- Integer (integer number between min and max)
- TextLine (single text line between min and max)
- MultiLine (multiple text lines between min and max)

INPUT Buttons (variable name) "caption" "button description"

Button Description:
"(button caption(s)) | (button color(s))"

Buttons Captions:
"caption1"
"caption1, caption2"
"caption1, caption2, caption3"
"caption1, caption2, caption3, caption4"

Buttons Colors:
"color1"
"color1, color2"
"color1, color2, color3"
"color1, color2, color3, color4"

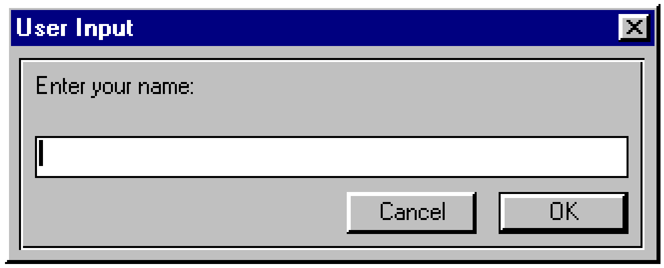
Color Numbers:

0	Black	8	Gray
1	Dark Blue	9	Light Blue
2	Dark Green	10	Light Green
3	Dark Cyan	11	Light Cyan
4	Dark Red	12	Light Red
5	Dark Magenta	13	Light Magenta
6	Dark Yellow	14	Light Yellow
7	Light Gray	15	Bright White

INPUT PopUpMenu (variable name) "menu 1" ["menu 2" ... "menu 20"]

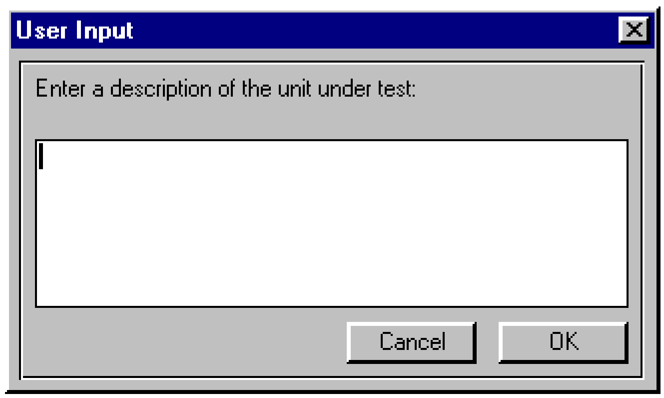
Example:
The following section shows a prompt for the user to enter his name on a single text line. The contents of the text line will be returned to the variable declared as ST1. The values of 0 and 50 are used limit the text string length to between 0 and 50 characters. Since the TextLine style is meant to return a text string, it should be used with a string variable and not a numeric variable. The FPnumber and Integer prompt styles are identical to the following figure, with the exception that a numeric value must be entered.

1: Declare String ST1
2: Input TextLine ST1 "Enter your name:" 0 50



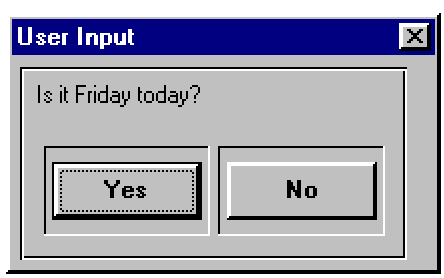
The next command uses the MultiLine style to allow the user to enter a text string that can be made up of more than a single line. In this case, the minimum text length is set to 10 characters, so that the user if forced to enter something. As with the previous example, the variable specified should be a string type. If a numeric variable type is specified, and the user enters a non-numeric value, then a run-time error will be generated.

1: Declare String ST1
2: Input MultiLine ST1 “*Enter a description of the unit under test*” 10 256



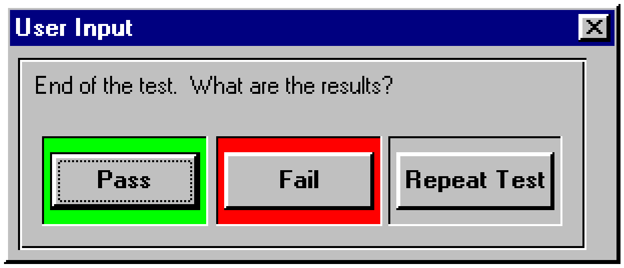
The Buttons style of user input displays a limited number of options to the user. In this example, a simple Yes/No selection. The number of buttons can range from 1 to 4 and is determined by the number of button captions defined. In this case, only 2 are defined. These being “Yes” and “No”. Commas are used to separate the button captions. The prompt returns a numeric value depending on the button pressed. The first button will return the number 1, while the second will return 2 and so on. Normally, the variable name specified will be numeric; however, it is possible to use a string variable. In this case, selection the first button returns the string “1”, while the second button returns the string “2”, and so on. A keyboard shortcut can be used instead of pressing the buttons with the mouse. The shortcut is the key 1 for the first button, 2 for the second, 3 for the third, and 4 for the forth button.

1: Declare Numeric N1
2: Input Buttons N1 “*Is it Friday today?*” “Yes, No”



The following command is another example of the Buttons style with three buttons and using the color option. Similar in form to the above example, the only significant difference is defining a third button and colors. Colors for the buttons are specified with numbers ranging from 0 to 15 following a | character in the caption string. In this case, the first button is assigned color 10, which is green. The second button is assigned color 12, which is red. The third button is not assigned as color, so it appears as the default gray.

```
1:      Declare   Numeric   N2
2:      Input   Buttons   N2   "End of the test.  What are the results?"
                "Pass, Fail, Repeat  |  10, 12"
```



Script Command: PARAMETER

Description:
The PARAMETER command is used to change the setting for any of the transmission parameters. Since there are two basic types of parameters, there are accordingly two different syntax's. The first parameter type is numeric, in which the parameter contains a numeric value between the specified minimum and maximum values. The second type is a multiple choice type of parameter, in which the parameter can take on only one of two or more possible options as defined uniquely by that parameter. The syntax for numeric parameters is specified as the parameter name, then an operator, and then a numeric quantity. Four possible operators can be used. They are "=" which sets the parameter value to the quantity specified, "+=", "-=", "*=", which modify the existing parameter value by either adding, subtracting, or multiplying the numeric quantity respectively. If the resulting parameter value is outside the minimum to maximum range for that parameter, is value is clamped to the closest minimum or maximum value. Multiple choice parameter types need no operator as their value can be only one of a fixed number of options. The desired option of the parameter simply follows the parameter name.

A special case of the parameter command resets all of the parameter values to their default values. This can be useful before the execution of a complex script program, since it sets all the parameter values back to a known value. The syntax to reset the parameters is simply following the parameter command word with "reset".

Syntax:
PARAMETER (numeric parameter name) (= , += , -= , *=) (numeric value)
PARAMETER (multiple choice parameter name) (selected option)
PARAMETER RESET

Example:
The example program cycles through various FSK transmit levels, then cycles though various interfering tone levels. Line 4 sets the multiple choice parameter FSK_IENABLE to the disabled state. This turns off the interfering tone during FSK transmission. Line 5 and 8 control the numeric parameter FSK_LEVEL in the first program loop. Line 5 sets the level to -10 dBm, while inside the loop, line 8 decreases the level 5 dB for every time through the loop. Once the first loop has finished, lines 11 to 13 enable the interfering tone and sets its frequency to 60 Hz, with a level of -60 dBm. Line 16 increases the interfering tone level by 15 dB for every cycle in the loop.

```
*  This program will cycle through various FSK transmit levels then
*  cycle through various interfering tone levels

Parameter FSK_IENABLE Disabled     ;turn off interfering tone
Parameter FSK_LEVEL = -10          ;set FSK transmit level
Loop 8                             ;repeat 8 times
   Start                           ;start CID transmission
   Parameter FSK_LEVEL -= 5        ;decrease level by 5 db
Loopend

Parameter FSK_IENABLE Enabled      ;turn on interfering tone
Parameter FSK_IFREQ = 60           ;set tone freq to 60 Hz
Parameter FSK_ILEVEL = -60         ;set level to -60 dBm
Loop 4                             ;repeat 4 times
   Start                           ;start CID transmission
   Parameter FSK_ILEVEL += 15      ;increase level by 15 dB
Loopend

End                                ;end script program
```

Script Command: MODE

Description:
The MODE command is used to set the Caller ID transmission mode. This can be either Type I (on hook data transmission), Type II (CIDCW off hook data transmission), or Auto (automatically changes between Type I and Type II depending on hook switch status of telephone). If, when starting a Caller ID transmission, the telephone's hook switch state is not compatible with the mode of operation, an error will be reported. For example, if the mode has been set for a type II transmission and the CPE is on-hook, then an error will be reported since the mode is incompatible with the CPE's state. The changing of the mode setting within a script program is only in effect until the program ends. Once the script program terminates, then the mode is set back to its setting before the script program had been executed.

Syntax:
MODE (TypeI, TypeII, Auto)

Example:
The example program tests a telephone in both Type I and Type II Caller ID modes.

```
*  This program will test both a Type I and Type II CID transmission

Printscreen "Make sure telephone is on hook for Type I testing"
Pause
Mode TypeI
Start                              ;start CID transmission

Printscreen "Make sure telephone is off hook for Type II testing"
Pause
Mode TypeII
Start                              ;start CID transmission

End                                ;end script program
```

Script Command: SIGNALING

Description:
The SIGNALING command changes the signaling type of the Caller ID transmission. For the Bellcore standard, there are three different signaling types. Two for Type I CID, and one for Type II CIDCW. The ETSI standard defines seven signaling types for Type I CID, and one for Type II CIDCW. Likewise, the Australian Caller ID system also has seven Type I CID signaling methods, and one for Type II CID. Any of these signaling options can be set with the signaling command.

Syntax:

SIGNALING (Data_after_Ring, OSI_no_Ring, No_Ringing, Send_CAS_wait_ACK)
SIGNALING (DTAS_with_Ring, RBAS_with_Ring, Rev_DTAS_with_Ring, DTAS_no_Ring,
RBAS_no_RING, Rev_DTAS_no_Ring, Data_after_Ring, DTAS_wait_ACK)
SIGNALING (RBAS_with_Ring, OSI_with_Ring, Reversal_with_Ring, No_Alert_with_Ring,
OSI_no_Ring, Reversal_no_Ring, No_Alert_no_Ring, Send_CAS_wait_ACK)

Example:

The example program first sends a Type I Caller ID transmission with power ringing, and then follows it with a Type I message waiting transmission with no ringing.

```
*  This program will test both Bellcore Type I signaling methods

Signaling   Data_After_Ring
Start                                   ;start CID transmission

Message    Multiple_Msg_Waiting
Packet    Visual_Indicator    Enabled
Packet    Visual_Indicator    Value   Activate

Signaling   No_Ring
Start                                   ;start CID transmission

End                                     ;end script program
```

Script Command: MESSAGE

Description:

The MESSAGE command sets the message type to use when sending the Caller ID data. The message type that is available will depend on the current standard selected. For both the Bellcore and ETSI standard, four message types are available. For the ETSI standard, the message types are very similar in structure to the Bellcore multiple message formats. ETSI does not support the older Bellcore single message formats. The Australian CID system uses the same message and packet types as the Bellcore standard.

Syntax:

Bellcore Standard:

```
MESSAGE   Single_Message
MESSAGE   Multiple_Message
MESSAGE   Single_Msg_Waiting
MESSAGE   Multiple_Msg_Waiting
```

ETSI Standard:

```
MESSAGE   Call_Setup
MESSAGE   Message_Waiting
MESSAGE   Advice_of_Charge
MESSAGE   Short_Message_Service
```

Example:

The example program sends two Caller ID transmissions. The first using the Single Message format, while the second uses the Multiple Message format.

```
*  This program will test both a Single and Multiple Message
*  Caller ID transmission

Message Single_Message
Start                                   ;start CID transmission

Message Multiple_Message
Start                                   ;start CID transmission

End                                     ;end script program
```

Script Command: PACKET

Description:
The PACKET command is used to control the data that is sent in a Caller ID transmission. Various packets, such as the date & time, calling number, name can be enabled, disabled, or their value changed. All of the possible packets for both the Bellcore and ETSI standards are listed below. The Australian CID system uses the same message and packet types as the Bellcore standard.

Using the PACKET command, it is also possible to change the value of the text based packets to the value of any parameter. For example, when testing a CPE at many different FSK signal levels, it is possible to set the Calling Name packet to the value of the FSK level. The CPE will then display the FSK level in the name field of its Caller ID display. This is accomplished by enclosing the desired paramater name in square brackets [] as the value of a parameter. For example, if the FSK level was set to -13 dBm, then the script line:

PACKET Calling_Name Value [FSK_LEVEL]

would set the contents of the calling name packet to “-13 dBm”.

Syntax:
PACKET (name) (Enabled, Disabled, Value (contents))
PACKET (name) (Enabled, Disabled, Value ([parameter name])

Bellcore:

PACKET   Date_&_Time (Enabled, Disabled, Value "mmddhhnn")
    mm = month   dd = day   hh = hour   nn = minute
PACKET   Calling_Number (Enabled, Disabled, Value)
    Value = up to 50 character string
PACKET   Number_Absence (Enabled, Disabled, Value)
    Value = Private, or Out_Of_Area
PACKET   Calling_Name (Enabled, Disabled, Value)
    Value = up to 50 character string
PACKET   Name_Absence (Enabled, Disabled, Value)
    Value = Private, Out_Of_Area
PACKET   Visual_Indicator (Enabled, Disabled, Value)
    Value = Activate, Deactivate
PACKET   Qualifier (Enabled, Disabled, Value)
    Value =- Long_Distance
PACKET   Redirection (Enabled, Disabled, Value)
    Value = Call_Fwd_Universal, Call_Fwd_Busy, Call_Fwd_Unanswered
PACKET   DDN (Enabled, Disabled, Value)
    Value = up to 50 character string

ETSI:

PACKET   Date_&_Time (Enabled, Disabled, Value "mmddhhnn")
    mm = month   dd = day   hh = hour   nn = minute
PACKET   Calling_Number (Enabled, Disabled, Value)
    Value = up to 50 character string
PACKET   Number_Absence (Enabled, Disabled, Value)
    Value = Private, or Out_Of_Area
PACKET   Calling_Name (Enabled, Disabled, Value)
    Value = up to 50 character string
PACKET   Name_Absence (Enabled, Disabled, Value)
    Value = Private, Out_Of_Area
PACKET   Visual_Indicator (Enabled, Disabled, Value)
    Value = Activate, Deactivate
PACKET   Called_Line (Enabled, Disabled, Value)
    Value = up to 50 character string
PACKET   Call_Type (Enabled, Disabled, Value)
    Value = Voice_Call, Ring_Back_When_Free, Calling_Name_Delivery,
    Message_Waiting_Call
PACKET   Network_Sys_Status (Enabled, Disabled, Value)
    Value = 0_Messages, 1_Messages, .. , 255_Messages
PACKET   Comp_Calling_Line (Enabled, Disabled, Value)
    Value = up to 50 character string
PACKET   First_Called_Line (Enabled, Disabled, Value "071 250 7587")
PACKET   Forwarded_Call_Type (Enabled, Disabled, Value)
    Value = Unavailable_or_Unknown, Forward_on_Busy,
    Forward_on_No_Reply, Unconditional_Forward,
    Deflected_after_Alerting, Deflected_Immediate, Can’t_Reach_Mobile
PACKET   Calling_User_Type (Enabled, Disabled, Value)

Value = Unavailable_or_Unknown, Virtual_Private_Network, Mobile_Phone, Mobile_Phone_and_VPN, Ordinary_Calling, Priority_Subscriber, Data_Call, Test_Call, Payphone

PACKET Redirecting_Number (Enabled, Disabled, Value)
Value = up to 50 character string

PACKET Network_Operator_Ext (Enabled, Disabled, Value "cccnnnnvvv")
ccc = country code, nnnn = network code, vvv = version

PACKET Message_Identification (Enabled, Disabled, Value, Type)
Value = number from 0 to 65535
Type = Remove, Reference_Only, Add

PACKET Last_Message_CLI (Enabled, Disabled, Value)
Value = up to 50 character string

PACKET Comp_Date_&_Time (Enabled, Disabled, Value "mmddhhnn" or "mmddhhnnss")
mm = month, dd = day, hh = hour, nn = minute, ss=second

PACKET Charge (Enabled, Disabled, Currency, Type, Cost, Units, PricePerUnit, FreeOfCharge, SubTotal, CreditDebit, NoInfo, ChargeUnits)
Currency = 3 characters representing currency code
Type = Current, Accumulated, Extra
Cost = up to 10 characters
Units = up to 5 characters
PricePerUnit = up to 5 characters
FreeOfCharge = Yes, No
SubTotal = Yes, No
CreditDebit = Yes, No
NoInfo = Yes, No
ChargeUnits = Yes, No

PACKET Additional_Charge (Enabled, Disabled, Currency, Type, Cost, Units, PricePerUnit, FreeOfCharge, SubTotal, CreditDebit, NoInfo, ChargeUnits)
Currency = 3 characters representing currency code
Type = Current, Accumulated, Extra
Cost = up to 10 characters
Units = up to 5 characters
PricePerUnit = up to 5 characters
FreeOfCharge = Yes, No
SubTotal = Yes, No
CreditDebit = Yes, No
NoInfo = Yes, No
ChargeUnits = Yes, No

PACKET Call_Duration (Enabled, Disabled, Value = "hhmmss")
hh = hours, mm = minutes, ss = seconds

PACKET Provider_Identity (Enabled, Disabled, Value)
Value = up to 50 character string

PACKET Carrier_Identity (Enabled, Disabled, Value)
Value = up to 50 character string

PACKET Terminal_Function (Enabled, Disabled, Mode, Type, Value)
Mode = Connection_Type, Subscriber_Number, SubAddress
Type = Not_Identified, Voice_Call, Fax_Call, Data_Call, Video_Call, E-mail_Call, Telemetric_Call, Text_Call
Value = up to 50 characters for subscriber number of sub-address

PACKET Display_Information (Enabled, Disabled, Type, Stored, Text)
Type = Unknown, Positive_Ack, Negative_Ack, Advertisement, Network_Information, User_Information
Stored = Yes, No
Text = up to 252 character string

PACKET Service_Information (Enabled, Disabled, Value)
Value = Not_Active, Active

Example:
The example program sends two Caller ID transmissions using the multiple message format. The first sends the calling number as 16045551212, while the second disables the calling number transmission, enables the number absence packet with a setting of "Out of Area", and with a time stamp of 11:45 PM on Dec. 31.

```
*  This program will send various packet combinations

Message Multiple                          ;use multiple message type

Packet Number_Absence Disabled
Packet Calling_Number Enabled
Packet Calling_Number Value "16045551212"
Start                                     ;start CID transmission

Packet Calling_Number Disabled
Packet Number_Absence Enabled
Packet Number_Absence Value Out_Of_Area
Packet Date_&_Time Value "12312345"       ;11:45 PM on Dec. 31
Start                                     ;start CID transmission

End                                       ;end script program
```

Script Command: SEGMENT

Description:
The SEGMENT command is used to adjust the length of any of the segments, except the Message Data segment, whose length is controlled by the packet data. The segment name is positioned after the SEGMENT command word, and can be one of the following: Start_Burst, Preamble, Mark, Mark_Out, and End_Burst. The new length of the segment can be specified in two ways. Either in terms of the number of bits (using the "NumberBits" keyword) or time (using the "Time" keyword). Note that the time of a segment is specified in milliseconds.

Syntax:
SEGMENT (name) (NumberBits, Time) (Numeric Value)

Example:
The example program sends two Caller ID transmissions with different preamble segment lengths. Line 4 sets the length of the preamble to 200 msec, while line 7 sets the length to 400 bits.

```
1:     *  This program will send two message with different preamble
2:     *  segment lengths
3:
4:     Segment Preamble Time 200           ;200 msec of preamble
5:     Start                               ;start CID transmission
6:
7:     Segment Preamble NumberBits 400     ;400 bits of preamble
8:     Start                               ;start CID transmission
9:
10:    End                                 ;end script program
```

Script Command: STOPBITS

Description:
The STOPBITS command sets the number of stop bits each byte of message data is encoded with. Each message byte is encoded in a serial format of one start bit, eight data bits, and then a number of stop bits as defined with the STOPBITS command. The numeric quantity following the command key word must be between 1 and 60.

Syntax:
STOPBITS (1,2,...,60)

Example:
The example program sends two Caller ID transmissions with a different number of stop bits. Line 4 sets the number of stopbits to 1, while line 7 sets the number to 10.

```
1:    *  This program will send two messages with a different
2:    *  number of stopbits in each message
3:
4:    StopBits 1
5:    Start                    ;start CID transmission
6:
7:    StopBits 10
8:    Start                    ;start CID transmission
9:
10:   End                      ;end script program
```

Script Command: MARKBITS

Description:
The MARKBITS command sets the number of mark stuffing bits to use when encoding the message bytes into the serial bit stream. Mark stuffing bits are extra stop bits inserted in various places within a multiple message and single message bit streams. The locations of the mark stuffing bits for a multiple message are after:

i) message type byte,
ii) message length byte,
iii) packet type bytes, iv) packet length bytes,
v) the last packet data byte.

For the single message format, the mark stuffing bits are located after:

i) message type byte,
ii) message length byte,
iii) the last message byte (before the checksum byte).

The numeric quantity following the command key word must be between 0 and 150.

Syntax:
MARKBITS (0,1,...,150)

Example:
The example program sends two Caller ID transmissions with a different number of mark stuffing bits. Line 4 sets the number of mark stuffing bits to 0, while line 7 sets the number to 10.

```
1:    *  This program will send two message with a different
2:    *  number of mark stuffing bits in each message
3:
4:    Markbits 0
5:    Start                    ;start CID transmission
6:
7:    Markbits 10
8:    Start                    ;start CID transmission
9:
10:   End                      ;end script program
```

Script Command: SET

Description:
The SET command can be used to perform a number of functions. It controls various program options such as the FSK modulator enable, FSK modulator drop-outs, FSK mark and space tone levels, echo generation, digital outputs, and what system of level units to use throughout the program. The syntax for all the variations of this command are as follows:

Syntax:
a) FSK Modulator Signal Drop-outs
This use of the SET command controls the location and position of a FSK modulator signal drop-out during a Caller ID transmission. The segment names can be either: “Start_Burst”, “Preamble”, “Mark”, “Mark_Out”, “End_Burst”, “Data”, and “(none)”. The start time and duration of the signal drop-out is given in units of milliseconds. Optionally, a signal attenuation value can be included following the duration. This determines the reduction in the FSK level in units of dB from 0 to 60, or “Max”. Specifying “Max” sets the FSK level to

zero during the drop-out period. If the attenuation value is not included in the command line, then the last attenuation value specified is used.

SET DROPOUT (segment name) (start time) (duration) [attenuation]

b) FSK Modulator Enable
The FSK signal can be enabled or disabled during a Caller ID transmission. This use of the SET command has the same effect as changing the “Enable FSK Modulator” check box on the Main Settings window.

SET FSK (on/off)

c) FSK Mark and Space Tone Levels
Normally, the FSK modulator mark and space tone levels are defined by their total signal level (FSK_LEVEL) and twist level (FSK_TWIST). However, at times it is more convenient to be able to set the mark and space tone levels directly. The following syntax of the SET command allows this. The command will cause the current settings of the total FSK signal level and twist level to be re-calculated for the specified mark tone level and space tone level.

SET MSLEVEL (mark level) (space level)

d) Enable or Disable Echo Generation
The following syntax for the SET command can control the settings for the echo generator. Each of the three taps can be set to a delay and attenuation value, or turned off. The units for the echo delay is in milliseconds and the attenuation units are in dB.

SET ECHOTAP (1/2/3) (delay value) (attn. value)
SET ECHOTAP (1/2/3) OFF

e) Set the Signal Level Units
The units used to define the various signal levels throughout the program can be changed using the following command. The units can be set to either dBm, dBV, or mVrms referring to either a terminated or unterminated telephone line. See Section 3-2, Defining the Signal Level Units for more information on what effect changing the level units has on the program operation. When specifying signal levels with the PARAMETER, TONEGEN, or SET MSLEVEL commands, the units will be assumed the same as what is set with this command.

SET LEVELUNITS (dBm/dBV/mVrms) (term/unterm)

f) Set the Digital Output A
The function of digital output A, at the rear DB9 pin connector, can be programed to serve a number of purposes. In addition to being able to set output A high or low with the On and Off options, the output level can be toggled between high and low with the Toggle option, or set to be active during FSK data transmission. If the FSK option is selected, output A will be at a low state until the FSK modulator is active during a Type I or Type II Caller ID transmission. Once active, output A will go high after a pre-determined number of bits have been generated. This delay is set by the FSKdelay option. Once all the bits have been generated, output A will return to its low logic level.

SET OUTPUTA (options)

Options:		
	On	Sets output A to a high logic level
	Off	Sets output A to a low logic level
	Toggle	Inverts the logic level of output A
	FSK *	Set to a high logic level during FSK

SET OUTPUTA FSKdelay (delay in bits)

See Section 5: Auxiliary Digital Outputs and Inputs for more information on the digital I/O port usage for the TSPC.

Note: The 'FSK' keyword is not supported when using the AI-7280 Central Office Line Simulator.

g) Set the Digital Output B

Similar to the output A key words, digital output B can be set on or off and toggle its output state. However, output B has the option to be set to a high logic level during ringing or OSI (open switching intervals). This is accomplished by specifying the RING or OSI option with the command.

SET OUTPUTB (options)

Options:		
	On	Sets output B to a high logic level
	Off	Sets output B to a low logic level
	Toggle	Inverts the logic level of output B
	Ring	Set to a high level during Ringing
	OSI	Set to a high level during OSI

See Section 5: Auxiliary Digital Outputs and Inputs for more information on the digital I/O port usage for the TSPC.

Note: The 'Ring' and 'OSI' keywords are not supported when using the AI-7280 Central Office Line Simulator.

h) Message Checksum Calculation Control
The automatic checksum calculation control, found in the CID Packet Format window, can be enabled or disabled using this variant of the SET command. Normally, this setting is off, which means that any changes to the message data with the Changebyte command will not force a recalculation of the Checksum byte. However, with the setting turned on, any modifications with the Changebyte command will force a checksum calculation.

SET AUTOXSUM (On/Off)

i) Signal Flow and Routing
The ROUTING keyword is a new selection and gives control over the signal flow between the tone generators, signal analyzer, telephone interface, and BNC inputs and outputs. Following the ROUTING keyword, one of four controls is specified. This can be either LineTX, Analyzer, BNCoutput, or Mixer. The LineTX control determines the signal source for the transmit path of the telephone interface. This can be from the internal signal generators (default), or the BNC input connector. The Analyzer control selects the signal analyzer source from either the telephone interface or the BNC input connector. The BNCoutput controls determines what signal will be routed to the external BNC output connector. This can be from the internal signal generator or the telephone interface. The GAIN option can be used to specify a level offset for the BNC output in units of dB. Finally, the Mixer control selects if the signal mixer is on or off, and the gain of the signal mixer.

SET ROUTING (control) (option)

Control:		Options:
	LineTX	Generator
		BNCinput
	Analyzer	LineRX
		BNCinput
	BNCoutput	Generator
		LineRX
		Gain (level in dB)
	Mixer	On
		Off
		Gain (level in dB)

j) Message Data Parity Control
The PARITY key word selects the encoding method for any ASCII data that is included as part of the Caller ID message. The parity selection can be either 8 data bits no parity, 7 data bits with odd parity, or 7 data bits even parity. The default setting is for 8 data bits with no parity; however, some Caller ID standards may use the other settings.

SET PARITY (8none, 7odd, 7even)

k) Caller ID Standard
The STANDARD keyword selects the Caller ID standard to use. Using this command is identical to that of selecting the [CONFIGURATION] [CALLER ID STANDARD] menu command, with the exception that the script program in memory is not cleared. To use the SET command to change the Caller ID standard setting, use the following syntax:

SET STANDARD (Bellcore, ETSI, Australia)

Example:
The following example uses the SET command the change various program settings before starting a Caller ID transmission. Line 1 enables the FSK modulator, while line 2 programs a FSK signal dropout for 10.0 msec to occur 40.0 msec into the Mark bits segment. Lines 3 and 4 set the level units to mVrms, and fix the FSK mark and space level to 300 mVrms and 100 mVrms respectively. Lines 5 to 7 setup the signal echo to product 2 echoes at delays of 3 and 6 msec with attenuation's of 13 and 26 dB respectively. Finally, line 9 starts the Caller ID transmission.

```
 1:    Set  FSK  ON                       ;enabled FSK modulator
 2:    Set  DROPOUT  MARK  40.0  10.0     ;dropout in Mark segment
 3:    Set  LEVELUNITS  mVrms  Unterm     ;levels specified in mVrms
 4:    Set  MSLEVEL  300  100             ;set FSK mark and space
 5:    Set  ECHOTAP  1  3.0  13.0         ;echo at 3 msec, 13 dB down
 6:    Set  ECHOTAP  2  6.0  26.0         ;echo at 6 msec, 26 dB down
 7:    Set  ECHOTAP  3  OFF               ;only 2 signal echoes
 8:
 9:    Start                              ;start CID transmission
10:    End                                ;end script program
```

Script Command: CLEAR

Description:
Similar to the SET command, the CLEAR command is used to clear or reset various program options such as the ACK counter, data log, and all global program settings. The syntax for all the variations of this command are as follows:

Syntax:

a) Reset the ACK counter
The ACK counter displayed in the data log window, which counts the number of good or bad ACK tones received from the CPE, can be reset to zero using the following command.

CLEAR ACKCOUNTER

b) Restore all Program Default Settings
The following command restores all of the default settings associated with the currently selected standard. It resets all of the parameter values as well as the message layer settings. This command mimics the [RESTORE DEFAULTS] option in the [CONFIGURATION] menu, except that it does not clear the script program from memory, data log, or ACK counter.

CLEAR ALL

c) Clear the Contents of the Data Log
The Data Log window and its associated buffer file can be cleared with the following command.

CLEAR LOG

d) Clear the contents of the Digital Input B pulse counter
The InBcount reserved variable returns the total number of pulses detected at the digital input B pin, since last reset. The clear command resets this reserved variable to a value of zero.

CLEAR INBCOUNT

e) Clear the timer
This command clears the TIMER reserved variable. The TIMER reserved variable returns the number of seconds that have elapsed since it was last cleared.

CLEAR TIMER

Example:
The following program uses the CLEAR command to restore all the CID1500 program settings back to their default value, for the currently selected standard. This can be useful in assuring that the script program starts from a "known state". Lines 1, 2, and 3 are used to restore all the default settings and clear the ACK counter and data log file. The remainder of the program generates 100 Type II

transmissions with a low CAS tone level of 10 mVrms. At the end of the 100 transmissions, the results of the ACK counter are displayed on the screen.

```
Clear   All
Clear   AckCounter
Clear   Log

Set   LevelUnits   mVrms   unterm
Parameter   CAS_LEVEL1 = 10
Parameter   CAS_LEVEL2 = 10
Mode   TypeII
Loop 100
    Start                                        ;start CID transmission
Loopend
PrintScreen  [ACKCOUNTER]                        ;show the results

End                                              ;end script program
```

Script Command: LOG

Description:
The LOG command can be used to control some of the settings for the data log file as well as saving the current contents of the log file to disk. The syntax for all the variations of this command are as follows:

Syntax:

a) Save the Data Log File
The SAVEFILE keyword following the LOG command will save the current contents of the log file to the file name specified. If the file name does not include a full path with drive letter, then it will be saved to the same directory as the CID1500 program.

LOG SAVEFILE “filename

b) Format of the ACK Tone Results
Following a type II Caller ID transmission, the results of the ACK tone analysis can be printed to the log file under various conditions. This is determined by the settings in the Log File Options control panel. The LOG command with the ACKdisplay key word can be used to change these settings that determine how the ACK results are printed to the log file.

LOG ACKdisplay (option)

The possible options are:

FailOnly	Show in log only when ACK tone fails
PassOnly	Show in log only when ACK tone passes
All	Always show ACK tone results in log file
None	Never show ACK tone results in log file
Condensed	Only show the condensed ACK tone report
Verbose	Print the full ACK tone report in the log file

c) Enable or Disable the “Print to Log” Header
Normally, using the PRINTLOG command to output data to the log file includes the “Print to Log:” text string at the beginning of the text line. The HEADER key word to the LOG command can be used to enabled or disable this header. The syntax for this command is shown below.

LOG HEADER (On/Off)

d) Data log printing options
Printing the data log supports optional titles as well as specifying text for the footer on each page. These options can be set by a script program using the LOG command with the following syntax:

LOG PRINTING TITLE ([NONE], [DEFAULT], “custom”)
LOG PRINTING FOOTER ([NONE], [DEFAULT], “custom”)

Printing the data log can include a title on each page that is either the data log file name, or any user specified text. If [NONE] is used as the value field then no title is printed, while

[DEFAULT] will print the file name of the data log. As an alternative, a custom title is specified by using a text string or script variable in the value field.

A footer is always printed on each page showing the page number in the bottom right corner. However, by using the "Footer" aspect, the left side text in the footer can be either blank by specifying [NONE], the system date and time by specifying [DEFAULT], or custom text by specifying a text string or script variable.

Example:
The following program sends 100 Type II Caller ID transmissions with a low CAS tone level of 10 mVrms. Before the test starts, line 1, sets the log file options so that only failed ACK's are recorded. At the end of the test, line 12, saves the contents of the data log to the file name of "test_1.log".

```
1:      Log   ACKdisplay   FailOnly
2:
3:      Set   LevelUnits   mVrms   unterm
4:      Parameter   CAS_LEVEL1 = 10
5:      Parameter   CAS_LEVEL2 = 10
6:      Mode   TypeII
7:      Loop 100
8:          Start                              ;start CID transmission
9:      Loopend
10:     PrintScreen  [ACKCOUNTER]              ;show the results
11:
12:     Log   SaveFile   "test_1.log"
13:     End                                    ;end script program
```

Script Command: NETTONE

Description:
The NETTONE command is used to execute a special network tone command file (*.nts) created by the NetTone program. These network tone files can be used to generate virtually any network tone with arbitrary frequencies, levels, and cadences. This includes dial tone, stutter dial tone, ring back tone, busy tone, and others. The NetTone program can be used to develop and execute these scripting files. Though the CID1500 program can not edit or modify a network tone script file, it can execute them via the NETTONE command.

In addition to simply executing the NetTone command files, parameters may be passed from the CID1500 scripting language to the NetTone command file. By using parameter passing, the CID1500 script programs can modify and/or control the NetTone command file. Up to 26 parameters can be specified by appending them to the end of the NetTone script command. Each parameter can be a numeric value, text string, user declared variable, or reserved variable. If a variable is used as a parameter, it is first evaluated to either a numeric value or text string, then passed to the NetTone file. Within the NetTone file, the use of the '%' character followed by a letter (A to Z), marks the parameter. For example, if parameters are included in the command line, the CID1500 will scan the NetTone file for '%A' and replace it with the first parameter. A second parameter, if included, is substituted into any occurrences of '%B' within the NetTone file. This process is repeated for up to 26 parameters.

The CID1500 script program will normally be suspended until the NetTone command file has finished executing. Once the NetTone command file has finished, any following CID1500 script commands will then continue execution. However, if the NOWAIT keyword is placed after the NetTone file name, the CID1500 will continue to execute script commands while the NetTone command file is running.

Syntax:
NETTONE (filename) [NOWAIT] [parameter list]

Where:

(filename) is the name of the file to process. If a complete path is not supplied, the file is assumed to be in the same directory as the CID1500 application.

NOWAIT, if included in the command line, it allows the CID1500 program to continue executing script commands while the NetTone program is running. If NOWAIT is not included, then no further script commands are executed until the specified NetTone program is finished.

[parameter list] is up to 26 items (separated by a space character), that are passed into the specified NetTone file. The items can text strings, numbers, declared variables, or reserved variables.

Example:
The following program generates normal dialtone and stutter dialtone in an alternating sequence every time the telephone goes off-hook. Line 3 executes the Dialtone.nts file, which is setup to wait until the telephone goes off-hook, then generates dialtone. Lines 4 and 5, then wait until the telephone goes on-hook. Once on-hook, line 7 executes the Stutter.nts file, which waits for the telephone to go off-hook, and then generates a stutter dialtone. If the telephone then goes on-hook, the program branches to the start, and the cycle continues indefinitely. The network tone script files (dialtone.nts, stutter.nts) are supplied with the NetTone program.

```
 1:    Label   Program_Start
 2:
 3:    NetTone   "Dialtone.nts"
 4:    Label   Wait1
 5:    Branchif  OffHook  Wait1
 6:
 7:    NetTone   "Stutter.nts"
 8:    Label   Wait2
 9:    Branchif  OffHook   Wait2
10:
11:    Branchif  Always  Program_Start
```

Additionally, the following example uses the CID1500 to generate a stutter dial tone. By using parameter passing, both the number of dial tone pulses generated and the tone on/off times are programmable. The script command below uses two parameters, being the numeric values of 100 and 10 respectively.

NETTONE   Example.nts   100 10

The first parameter (100) is substituted into every occurrence of '%A' found within the "Example.nts" file. As shown below, this controls the duration of the dial tone pulse as well as the duration between dial tone pulses. Likewise the second parameter (10) is substituted into every occurrence of '%B'. This parameter controls the number of loops executed.

```
*  Stutter dial tone (send %b pulses then
continuous dial tone)
LevelUnits   dBm                ;define lines in units
of dBm
Wait  OffHook                   ;wait until off-hook
Loop  %b                        ;do %b loops
  ToneA  350.0  -20.0           ;generate 350 Hz at
-20 dBm
  ToneB  440.0  -20.0           ;generate 440 Hz at
-20 dbm
  Wait  %a                      ;tone on for %a
msec
  ToneA   Off
  ToneB   Off
  Wait  %a                      ;tone off for %a
msec
LoopEnd
ToneA  350.0  -20.0             ;generate 350 Hz at
-20 dBm
ToneB  440.0  -20.0             ;generate 440 Hz at
-20 dbm
```

Example NetTone File: Example.nts

If parameters are included on the NetTone command line but the specified file does not contain any parameter markers ('%A' to '%'Z), then an error message is displayed. Likewise, if no parameters are included on the NetTone command line, but the specified file contains parameter markers, an error message will be displayed.

Script Command: DCIMP

Description:
The DCIMP script command is used to control the settings of any optional DC line impairments generated during a Caller ID transmission. The DC Line Impairments options within the CID1500 software allow either an OSI or line polarity reversal as an impairment for both Type I and Type II Caller ID transmissions. In the case of Type I, the DC impairments are only active for the signaling method in which the FSK data is sent after the first ring. Other Type I signaling types that utilize line reversals, OSI's, DTAS tones, and short ringing bursts are unaffected by the DC Line Impairment settings.

Syntax:
DCIMP (impairment) (parameter) (value)

Impairment: *AfterRing* (For Type I Caller ID)
BeforeSASCAS (For Type II Caller ID)
AfterFSK (For Type II Caller ID)

Parameter: *Type* (specify impairment type)
Delay (specify impairment delay)
Duration (specify OSI impairment duration)

Value: For (parameter) = Type:
None (turn off the impairment)
Reversal (generate a line reversal)
OSI (generate an OSI)

For (parameter) = Delay:
(time value in integer msec from 0 to 1000)

For (parameter) = Duration:
(time value in integer msec from 1 to 1000)

Example:
As an example, the following program enables an OSI after the first ring and before the FSK data for a Type I Caller ID transmission, with an OSI duration of 150 msec occurring 75 msec before the start of the FSK data. Once the DC impairment conditions have been set a Caller ID transmission is started. After it has finished, line 10 disables the OSI DC impairment.

```
1:    *   Turn on an OSI impairment before the FSK data
2:    DCimp   AfterRing   Type   OSI
3:    DCimp   AfterRing   Delay   75
4:    DCimp   AfterRing   Duration   150
5:
6:    *   Start the Caller ID Transmission
7:    Start
8:
9:    *   Disable the OSI
10:   DCimp   AfterRing   Type   None
```

Script Command: PRINTSCREEN

Description:
The PRINTSCREEN command is used to display text strings on the screen during the execution of a script program. This can be used to display various test conditions, or what the user should expect of the telephone under test. In addition to displaying literal text strings, the command can show the value of any parameter setting, script variable, or reserved variable. The contents of any declared variable can be printed by specifying its name in the item list. Enclosing the parameter settings, or script variables in [ ] brackets, performs a conversion to a text string. This is required, since the PRINTSCREEN command can only display text strings. If the variable is a string type, then the [ ] brackets are optional. Reserved variables such as [DATE], [TIME], and [NEWLINE] can be used to display/print the current system date, time, or start a new line respectively. The | character can be used as a shortcut to the [NEWLINE] variable, as it performs the same function. It is important to remember that each item included on the PRINTSCREEN command line must be separated by a space.

For example, if the current FSK level setting was -13 dBm, the script line

PRINTSCREEN “The level is: “ [FSK_LEVEL]

Would display the following on the screen:

The level is: -13 dBm

Syntax:
PRINTSCREEN item1 item2 ... item n

Items: “literal text in quotations”
[declared string or numeric variable]
[parameter setting]
[reserved variable]
|, to output a CR+LF character combination
{x} to output the ASCII character represented by x

Example:
The example program uses the PRINTSCREEN command to first indicate to the operator what parameters are being changed, as in line 7. The PRINTSCREEN command in line 11, tells the operator what options are available

```
1:    ************************************************************
2:    *  Type I CID Test Program
3:    *  May 18, 1997
4:    *  Version 1.0
5:    ************************************************************
6:    Label START_HERE
7:    Parameter FSK_LEVEL = -36
8:    PrintScreen  "The FSK transmit level is:  "  [FSK_LEVEL]
9:
10:   Start                              ;start CID transmission
11:   PrintScreen  "Program Paused, press F6 to cont.  F9 to repeat"
12:   Pause START_HERE                   ;pause program
13:
14:   End                                ;end script program
```

Script Command: PRINTLOG

Description:
The PRINTLOG command writes the text string enclosed in quotation marks to the data log file. Like the PRINTSCREEN command, it can also write parameter settings and script variables to the data log file. The two commands have the same syntax structure, with the primary difference that the PRINTLOG command sends its output to the log file, while the PRINTSCREEN command sends its output to the screen. One other difference between the two commands is that the PRINTLOG command will accept a comma “,” as a tab character. This is helpful in formatting output values in the log file.

Syntax:
PRINTLOG item1 item2 ... item n

Items: “literal text in quotations”
[declared string or numeric variable]
[parameter setting]
[reserved variable]
, to output a tab character (ASCII code 9)
| to output a CR+LF character combination
{x} to output the ASCII character represented by x

Example:
The example program uses the PRINTLOG command to write to the data log file the changes made to the CAS tone level parameters. This simplifies identifying multiple Type II Caller ID transmissions ACK tone analysis in the log file, since the PRINTLOG command will indicate which transmission corresponded to which parameter change.

```
Mode TypeII

Parameter CAS_LEVEL1 = -40
Printlog "CAS tone # 1 level is:  "  [CAS_LEVEL1]
Start                                 ;start CID transmission

Parameter CAS_LEVEL2 = -40
Printlog "CAS tone # 2 level is:  "  [CAS_LEVEL2]
Start                                 ;start CID transmission

End                                   ;end script program
```

Script Command: PRINTFILE

Description:
Similar in nature to the PRINTSCREEN and PRINTLOG commands, the PRINTFILE command will output a text string to a specified file. In addition to displaying literal text strings, the command can write the value of any parameter setting, script variable, or reserved variable. The contents of any declared variable can be written to a file by specifying its name in the item list. Enclosing the parameter settings, or script variables in [ ] brackets, performs a conversion to a text string. This is required, since the PRINTFILE command can only write text strings. Reserved variables such as [DATE], [TIME], and [NEWLINE] can be used to write the current system date, time, or start a new line respectively. The | character can be used as a shortcut to the [NEWLINE] variable, as it performs the same function. It is important to remember that each item included on the PRINTFILE command line must be separated by a space.

Note that the PRINTFILE command does not overwrite an existing file. Instead it will append the text strings to the end of any existing file. To ensure a new file is created with the PRINTFILE command, use the DELETEFILE command to erase it first. Also, the PRINTFILE command will not add any line feed or carriage return characters unless explicitly told to do so with the [NEWLINE] reserved variable or the | character. For example the following script:

PRINTFILE test.txt "The current time is:"
PRINTFILE test.txt [TIME]

Would generate a file containing one line of text without any line feeds or carriage return characters, as in the following.

The current time is:1:50:19 PM

Syntax:
PRINTFILE (filename) [item1 item2 ... item n]

Filename: Name and path of file to output text to.
If a complete path is not supplied, the file will be written to the same directory as the CID1500 application.

Items:
"literal text in quotations"
[declared string or numeric variable]
[parameter setting]
[reserved variable]
, to output a tab character (ASCII code 9)
| to output a CR+LF character combination
{x} to output the ASCII character represented by x

Example:
The example program uses the PRINTFILE command to create a file called “Output.txt”, containing the recorded data from the last ACK tone received. Line 3 writes a title to the file, with lines 4 to 6 writing the ACK low frequency value, ACK high frequency value, and ACK digit code.

```
1:    DeleteFile  Output.txt          ;erase the file, it is exists
2:
3:    PrintFile  Output.txt  “Results of last Type II ACK Tone:”  |  |
4:    PrintFile  Output.txt  “Low Tone Freq=“  [ACKLOWFREQ]  |
5:    PrintFile  Output.txt  “High Tone Freq=“  [ACKHIGHFREQ]  |
6:    PrintFile  Output.txt  “Digit Code=“  [ACKDIGIT]  |
7:
8:    End
```

Script Command: READFILE

Description:
Data can be read from files by using the READFILE script command. Any files read must be in an ASCII text format using a structure common to Windows .INI type files. This structure consists of groups of “sections”, each containing a list of “keywords” and “values” in the following format:

```
[section 1]
keyword 1 = value 1
...
keyword n = value n
```

An example of this is:

```
[FSKPARAMS]
MarkFreq=1200
SpaceFreq=2200
[MESSAGEDATA]
CallingName="John Smith"
CallingNumber="6045551212"
```

In which the file contains two sections called “FSKPARAMS” and “MESSAGEDATA”, each further containing two keywords. The READFILE script command is used to read any of the values following a keyword into a declared variable.

Syntax:
READFILE (filename) (section) (key) (variable)

Filename: Name and path of file to read data from.
If a complete path is not supplied, the file will be read from the same directory as the CID1500 application.

Section: The text representing the section to read, without the enclosing [ ] brackets, and is case insensitive.

Keyword: The text preceding the equal sign and is case insensitive.

Variable: The name of a declared script variable which will receive the specified value from the file.

Example:
Using the above example data file (called test.txt), the following script program will read the value of the MarkFreq keyword and the CallingName keyword. Upon reading the data into the Mfreq and Cname variables, it will display their contents using the PrintScreen command.

```
Declare  Numeric  Mfreq
Declare  String  Cname
ReadFile  test.txt  FskParams  MarkFreq  Mfreq
ReadFile  test.txt  MessageData  CallingName  Cname

Printscreen  [Mfreq]  “and”  [Cname]
```

The value of the MarkFreq keyword (1200) is assigned to the numeric variable Mfreq, while the value of the CallingName keyword (“John Smith”) is assigned to the string variable Cname.

If the specified file name, section name, or keyword name does not exist an error message is generated and the execution of the script program is stopped.

Script Command: DELETEFILE

Description:
The DELETEFILE command is used to delete any file. It is normally used in conjunction with the PRINTFILE command to ensure that the text strings written with the PRINTFILE command are directed to a new file and not appended to an existing file. The syntax for this command is as follows:

Caution should be taken in using this command, as it can erase any file. If the specified file is not found or does not exist, no error will be reported.

Syntax:
DELETEFILE (filename)

Filename: Name and path of file to delete.
If a complete path is not supplied, the file will be assume to be in the same directory as the CID1500 application.

Example:
The above script example queries the user if they wish to delete the file called "OutLog.txt" by selecting either the Yes or No button. Line 2 sets the variable KillFile to either 1(yes) or 2(no), depending on the button selected. Line 3, then tests whether the yes button was pressed. If so, KillFile will be evaluated as TRUE in the IF statement on line 5, which then executes line 6 deleting the file.

```
1:    Declare  Numeric KillFile
2:    Input  Buttons  KillFile  "Erase the file:  OutLog.txt?"  "Yes,No"
3:    Assign  KillFile  To  KillFile = 1
4:
5:    If KillFile Then
6:        DeleteFile  OutLog.txt
7:    EndIf
```

Script Command: CHANGEBYTE

Description:
The CHANGEBYTE command can be used to change a single byte in any packet in the Caller ID Message to any value. This can be useful in sending messages to the CPE with invalid message types, packet types, or incorrect checksums and then verifying that the CPE rejects the corrupted message. The syntax of the command consists of specifying the packet name to modify, followed by the byte number of modify, and then finally the new value of the byte. The byte number must be an integer between 1 and 256, otherwise an error will be generated. The byte value specified must be an integer between 0 and 255, or a special modifier. These can be the key words INVERT, INC, or DEC. They are used in place of a byte value. These will cause the byte in question to be bit-wise inverted, or incremented by 1, or decremented by 1 respectively. During the execution of the command, if the packet name specified is not currently enabled, a warning will be generated. Likewise, if the byte number specified is outside the range of the packet, a warning will be generated. For example, this can happen if you wish to change byte #6 in a five byte packet.

Note: If the Automatic Message Checksum Calculation feature is enabled, then a warning will be generated if the CHANGEBYTE command is used to change the checksum byte of a message. The feature should enabled if the checksum is to be re-calculated after any byte in the message has been changed manually or via the CHANGEBYTE command. However, with this feature enabled, you can't modify the checksum byte. For more information on how the checksum is calculated, and its ramifications, see the section: Automatic Message Checksum Calculation.

Syntax:
CHANGEBYTE (packet) # (byte number) to (value, INVERT, INC, DEC)

Example:
The example program uses the CHANGEBYTE command to send invalid messages to the CPE. The program first sets the message type for the Multiple Message Format and sends the message to the CPE. The CPE should correct display the message. Next the CHANGEBYTE command in line 3 is used to set the message type to an invalid value. The CPE should reject this message. Following this, the message is reset to its original value with the "MESSAGE Multiple" command in line 6. Again, a correct Caller ID message is sent, followed by the CHANGEBYTE command in line 8, which inverts the checksum byte. When this message is sent, the CPE should reject it because of its incorrect checksum.

```
1:    Message Multiple
2:    Start                                    ;start CID transmission
3:    ChangeByte Multiple_Msg_Header # 1 to 47
4:    Start                                    ;start CID transmission
5:
6:    Message Multiple
7:    Start                                    ;start CID transmission
8:    ChangeByte Checksum_Byte # 1 to INVERT
9:    Start                                    ;start CID transmission
10:
11:   End                                      ;end script program
```

Script Command: CHANGEBIT

Description:
The CHANGEBIT command can be used to change a single bit in any segment in the Caller ID Message. The selected bit can be forced to a zero, one, or inverted from its current value. This can be useful in sending messages to the CPE with bit errors located in selected locations. The syntax of the command consists of specifying the segment name to modify, followed by the bit position to modify, and then finally the new value of the bit. The bit position can be specified from either the start of the segment or the end of the segment. If the "#" symbol is used, the position is relative to the start of the segment. Alternatively, using "FromEnd" causes the bit position to be calculated relative to the end of the segment. Scripting variables can be used for both the bit position and value fields. If variables are used, they must be numeric and not string variables. Also, variables for the value field must evaluate to either the number 0 or 1, otherwise an error will be generated. If the bit position value is greater than the maximum number of bits in a segment, a warning is issued at runtime.

Using the CHANGEBIT command will flag the segment modified as "Altered". As such, a yellow "A" is displayed in the programs status bar. This serves as a reminder that data layer modifications have been made.

Due to the hierarchical structure used to display and format the Caller ID data, any changes to the operating standard, message type, and message contents will cause a recalculation of the data segment and nullify changes made. Likewise, changing from Type I (on-hook) to Type II (off-hook) transmission modes resets the preamble and mark segments.

Syntax:
CHANGEBIT (segment) # (bit position) TO (0,1,toggle)
CHANGEBIT (segment) FromEnd (bit position) TO (0,1,toggle)

Example:
The example program uses the CHANGEBIT command to toggle the fourth bit in the preamble. This changes the start of the preamble from 01010101 to 01000101. Also, the stop bit of the checksum byte is set to zero from its normal value of one.

```
Clear All
Message Multiple

Changebit preamble # 4 to Toggle
Changebit data FromEnd 1 to 0

Start                                    ;start CID transmission
```

Script Command: TONEGEN

Description:
The TONEGEN command is used to control the arbitrary dual tone generator. Up to two arbitrary tone frequencies and levels along with the noise generator can be controlled using this command. The syntax is similar to the PARAMETERS command, which is used to adjust the Caller ID transmission characteristics.

Syntax:
TONEGEN (parameter name) (operator) (value)

The parameter name can be one of the following

TONE1_LEVEL	Level of tone #1 in the currently selected units
TONE1_FREQ	Frequency of tone #1 in Hz
TONE1_ENABLE	Turns on or off tone #1
TONE2_LEVEL	Level of tone #2 in the currently selected units
TONE2_FREQ	Frequency of tone #2 in Hz
TONE2_ENABLE	Turns on or off tone #2
NOISE_LEVEL	Level of noise generator in the currently selected units
NOISE_ENABLE	Turns on or off the noise generator

The operator, which effects how the selected parameter value will be changed, can be one of the following.

=	Set to specified value
+=	Increment by specified value
-=	Decrement by specified value
*=	Multiply by specified value

Example:
The example program generates a frequency sweep using the tone generator. Lines 1 to 3 initially disable the two tones and noise, assuming that they were active. Then lines 4 to 8 sets tone #1 to a level of 0 dBm into a terminated telephone line at an initial frequency of 300 Hz. Finally lines 10 to 13 froms a loop which waits half a second for each tone before incrementing the frequency by 100 Hz. Once completed, line 15 turns off the tone.

```
1:    ToneGen   TONE1_ENABLE   disabled
2:    ToneGen   TONE2_ENABLE   disabled
3:    ToneGen   NOISE_ENABLE   disabled
4:
5:    Set   LevelUnits   dBm    term
6:    ToneGen   TONE1_LEVEL  =  0.0
7:    ToneGen   TONE1_FREQ  =  300.0
8:    ToneGen   TONE1_ENABLE   enabled
9:
10:   Loop 30
11:       Wait   0.50
12:       ToneGen   TONE1_FREQ  +=  100.0
13:   Loopend
14:
15:   ToneGen   TONE1_ENABLE   disabled
```

Script Command: FSKGEN

Description:
Like the TONEGEN command, the FSKGEN command is used to control the operation of the FSK modulator at times when no Caller ID transmissions are in progress. This command can be used to control the bit pattern generated by the FSK modulator as well as starting or stopping it. The level, frequencies, baud rate and other parameters of this idle mode FSK generator are still controlled by the same parameters that define its operation during a Caller ID transmission. These can be changed in the Advanced Setting window, or by the PARAMETERS script command. The following aspects of the FSK modulator can be controlled by the FSKGEN command.

Syntax:
a) Mode of Operation
The FSK modulator can operate in one of two basic modes of operation. These are single shot or continuous mode. In single shot mode, the FSK modulator will generate the specified data stream and then terminate after the last bit has been sent. In the continuous mode of operation, once the modulator reaches the end of the specified bit pattern, it

begins to repeat the same bit pattern over indefinitely. The setting of the FSK modulator's mode effects the execution of the script program. In single-shot mode, once the FSK modulator has started, the script program execution is halted until the bit pattern has been sent. At that time further script commands are processed. In the continuous mode of operation, the script program execution continues to process commands while the FSK modulator is active.

FSKGEN MODE (single/continuous)

b) Type of Bit Pattern
The TYPE keyword specifies what type of bit pattern is to be generated by the FSK modulator. This can be selected from one of 5 different possibilities. These are:

MARK	Send Mark Bits Only
SPACE	Send Space Bits Only
ALTERNATE	Alternate Mark and Space Bits
CUSTOM	Send a User Defined Bit Pattern
EXTERNAL	External Data Modulation

FSKGEN TYPE (mark/space/alternate/user/external)

The USER type of bit pattern is defined with the DATA keyword. If the EXTERNAL type of bit pattern is selected, then the state of the FSK modulator depends on the input level at digital Input B (pin 9). If Input B is high, a mark tone will be generated, otherwise a space tone will be generated.

c) User Defined Bit Pattern
For the USER bit pattern, the DATA keyword is used to specify what bit pattern is to be generated. The syntax for the command is.

FSKGEN DATA "text string

The text string is used to define the bit pattern at an ASCII, hexadecimal, or binary level. For example, the following text string:

"{ASCII} [fe 8c b5] <01100101>"

Defines a bit pattern that starts with 5 ASCII characters ("A", "S", "C", "I", and "I") starting with 1 start bit and ending in 1 stop bit with the data sent LSB to MSB. This is immediately followed by the hexadecimal bit pattern given by 0xfe, 0x8c, and 0xb5 (MSB to LSB), which is then immediately followed by the binary pattern of 01100101.

The type of brackets used in the text string define what encoding method is used.

{ } ASCII Characters
[ ] Hexadecimal Bytes (0 to 9, and a, b, c, d, e, f)
< > Binary Bits (0 or 1)

Note that "nesting" of data types is not allowed.

d) Specifying the Number Of Bits to Send
If either the MARK, SPACE, or ALTERNATE bit pattern types has been selected, and the FSK modulator mode is single shot, then the BITS keyword is used to specify how many bits should be generated. The following syntax is used to define the number of bits to generate from 1 to 8192.

FSKGEN BITS (number of bits)

e) Starting or Stopping the FSK Modulator
The FSK modulator is started or stopped with the following command lines. If the FSK modulator is in single shot mode, then the STOP command has no purpose, as the script program execution will be halted until the last bit has been sent. Then the FSK modulator will automatically turn off and the script program execution will continue. However, in the continuous mode of operation, the script program will continue to execute statements after the GO command. In this case the STOP command can be used to terminate the FSK modulator.

FSKGEN GO
FSKGEN STOP

Note: that starting a Caller ID transmission will terminate the FSK modulator, if it is active. Also, once the FSK modulator has been started with the GO command, any changes to its settings will not take effect until the next time it is started.

Example:
The example program causes the FSK modulator to generate a constant alternating mark space bit pattern at various baud rates. Lines 1 to 3 set the FSK generator mode and initial baud rate. Line 5 starts the FSK generator in a continuos mode. The loop defined by 7 to 13 increments the baud rate by 5 Hz and pauses the program unit the user presses the start button. Once the start button is pressed, the baud rate increases by 5 Hz. If F9 is pressed, then the program jumps to the end.

```
1:     FSKGen   Mode   Continuous
2:     FSKGen   Type   Alternate
3:     Parameter  FSK_BAUD  =  1150
4:
5:     FSKGen   Go
6:
7:     Loop  20
8:         Parameter   FSK_BAUD  += 5
9:         PrintScreen  "Press F6 to change baud rate to  "  [FSK_BAUD]
10:        Pause   End_of_Prog
11:        FSKGen   Stop
12:        FSKGen   Go
13:    Loopend
14:
15:    Label   End_of_Prog
16:    FSKGen  Stop
```

Script Command: WAIT

Description:
The WAIT command can be used to insert a programmed delay into the script program. The time delay can range from 0 to 30 seconds, with a resolution of approximately 50 milliseconds. The syntax for this command is.

Syntax:
WAIT (time delay in seconds)

Example:
The example program generates a frequency sweep using the tone generator. Lines 1 to 3 initially disable the two tones and noise, assuming that they were active. Then lines 4 to 8 sets tone #1 to a level of 0 dBm into a terminated telephone line at an initial frequency of 300 Hz. Finally lines 10 to 13 froms a loop which waits half a second for each tone before incrementing the frequency by 100 Hz. Once completed, line 15 turns off the tone.

```
1:     ToneGen   TONE1_ENABLE   disabled
2:     ToneGen   TONE2_ENABLE   disabled
3:     ToneGen   NOISE_ENABLE   disabled
4:
5:     Set   LevelUnits   dBm    term
6:     ToneGen   TONE1_LEVEL  =  0.0
7:     ToneGen   TONE1_FREQ  =  300.0
8:     ToneGen   TONE1_ENABLE   enabled
9:
10:    Loop 30
11:        Wait   0.50                              ;wait 500 msec
12:        ToneGen   TONE1_FREQ  +=  100.0
13:    Loopend
14:
15:    ToneGen   TONE1_ENABLE   disabled
```

Script Command: DO

Description:
The DO command can perform two different types of actions. The first is generating an OSI (Open Switching Interval) of various durations. The second is connecting and disconnecting the DC feed

and AC termination from the telephone interface. The syntax structure consists of the DO keyword, followed by the action keyword and any additional parameters. In the specific case of performing an OSI, the additional parameter must specify its duration in milliseconds. During the OSI period, the script program execution is halted. It is restarted again, once the OSI period has finished. The action keyword of "TipRing" is used to control the DC feed and AC termination. If it is followed by "Disconnect", the TSPC's rear panel RJ-11 jack is completely isolated from the telephone interface. using the "Connect" keyword restores the DC feed and AC termination.

Syntax:
DO (action) (parameters)

Action:	OSI	Generates an Open Switching Interval
	TipRing	Connects or disconnects DC feed
Parameter:	OSI	duration in msec
	TipRing	either "Connect" or "Disconnect"

Example:
The following script example generates an OSI that lasts for 500 msec. Once the OSI has finished, the following PrintScreen command is executed.

1:	* Create an OSI that is 500 msec in duration
2:	**Do OSI 500**
3:	PrintScreen "Done"
4:	End

Using Script Variables

The scripting language uses two different types of variables. These are the built-in variables and the user defined variables. The user defined variables must be defined by the DECLARE command somewhere within the script program before they are used. These variables can contain numeric or text string information depending on their definition. The other class of variables, built-in variables, do not need to be defined before using them. These variables represent all the various physical layer parameters, which can be changed with the PARAMETER command, along with others that represent the current system date, system time, and ACK tone pass/fail counter. The list of supported built-in variables is as follows:

DATE	Returns the current system date.
TIME	Returns the current system time.
ACKCOUNTER	Returns a text string containing the total number of Type II Caller ID transmissions sent, including the number of ACK tones detected and missed.
ACKLOWFREQ	Returns DTMF low group tone frequency of the last ACK in units of Hz.
ACKHIGHFREQ	Returns DTMF high group tone frequency of the last ACK in units of Hz.
ACKLOWLEVEL	Returns DTMF low group tone level of the last ACK in the current level units defined.
ACKHIGHLEVEL	Returns DTMF high group tone level of the last ACK in the current level units defined.
ACKDIGIT	Returns single character string with last ACK digit.
ACKTIME	Returns time from end of CAS to when the ACK tone was detected in units of msec.
ACKPASS	Returns the number of times the ACK tone was successfully detected.
ACKFAIL	Returns the number of times the ACK tone was not successfully detected.
ACKTOTAL	Returns the total number of times a Type II Caller ID transmission was sent.
ONHOOKTIME	Returns the time from the end of CAS to when the CPE went on-hook in order to perform a parallel set detect.
OFFHOOKTIME	Returns the time from the end of CAS to when the CPE returned to the off-hook state after performing a parallel set detect.
INBCOUNT	Returns the number of pulses detected at the Digital Input B pin since the last time it was reset with the CLEAR INBCOUNT command.
TIMER	Returns the number of seconds that have elapsed since it was last reset by the CLEAR TIMER command.
NEWLINE	Returns an ASCII CR+LF character combination.
TRUE	Returns either 1 for numeric representation or TRUE for string representation.

FALSE	Returns either 0 for numeric representation or FALSE for string representation.
(Parameter Name)	Returns either the numeric or string representation of the specified parameter depending on the context.

Some of the reserved variables can only be represented as strings. This includes the DATE, TIME, ACKCOUNTER, ACKDIGIT, and NEWLINE variables. As such they should always be enclosed in [ ] brackets. The TRUE and FALSE built-in variables find their use in logical comparisons with the ASSIGN command. These two variables can be used with either numeric or text string logical comparisons. As with all variables, to use them as text strings, they must be enclosed in [ ] brackets. Finally, all of the parameters accessible with the PARAMETER command can be treated as variables. Most parameters can be used as numeric or text strings as in the following example:

FSK_BAUD	contains the default numeric value of 1200
[FSK_BAUD]	contains the default text string of “1200.0 bps”

However, some parameters are not numeric in nature, as with the telephone line polarity. This parameter can only be set to “Normal” or “Reversed”. As such, it can be used as a text string, but not as a numeric value.

LINE_POLARITY	invalid, since parameter can only be a text string
[LINE_POLARITY]	contains the default text string of “Normal”

To change the value of a user defined variable, the ASSIGN or INPUT commands must be used. However, for the built-in variables, these commands will not work. To change any of the parameter variables, use the PARAMETER command. The other built-in variables such as [DATE], [TIME], [NEWLINE], [ACKCOUNTER], TRUE, FALSE can not be written too. They are in effect read only variables. Though the ACK tone pass fail counter can be cleared with the CLEAR command.

The user defined variables has a large effect on the pre-execution error scanning ability of the script language interpreter. Since the exact value of any variable is unknown until runtime, when an initial syntax scan of the script program is performed, it will be unable to catch any illegal uses for variables, except for the type checking. As such, run time errors may occur when a command uses a variable or literal value that is outside of its valid range.

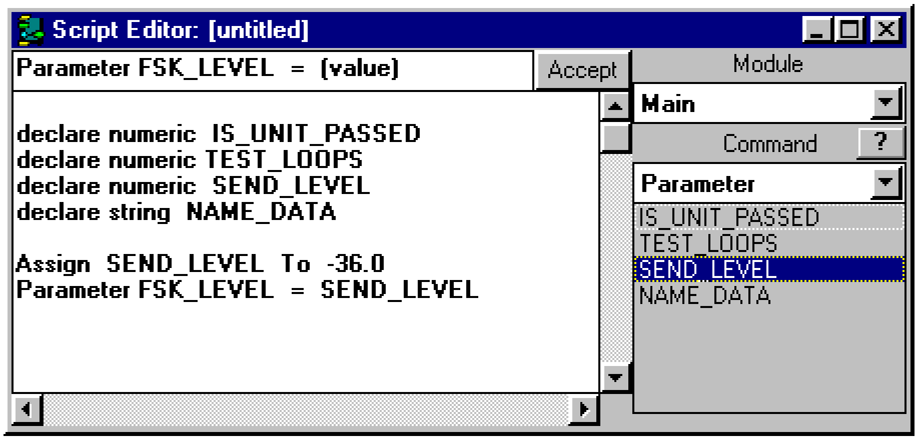
The following list contains all of the possible scripting commands. In many of the commands, variables may be used instead of literal numeric or string constants. The underlined portion of the commands shown below indicate where variables can be specified. It is important to match the type of variable to the requirement for the script command. For example, the first command in the list that can accept a variable is the loop command. If a variable is used to specify the number of loops, it must be a numeric type and not a string variable. If the type of variable does not match the requirement of the command, then an error will be displayed. Commands that can utilize numeric variables are shown below with a single underline. Likewise, commands that can utilize string variables are shown below with a dotted underline. Some commands can use either numeric or string variable types, and these are specified with a double underline.

Note that enclosing a numeric variable in [ ] brackets uses the string equivalent of that variables contents. This allows numeric variables to be used anywhere string variables are required, since a number can always be converted into a string. However, the reverse is not always true.

START	
PAUSE	(optional label) ([EXT] optional)
END	
LOOP	<u>(number of loops)</u>
LOOPEND	
IF	<u>(condition)</u> THEN
LABEL	(label name)
BRANCHIF	<u>(condition)</u> (label)
CALL	(program label)
RETURN	
DECLARE	STRING <u>(variable name)</u>
DECLARE	NUMERIC <u>(variable name)</u>
ASSIGN	<u>(variable name)</u> TO <u>(expression)</u>
INPUT	(style) <u>(variable name)</u> “caption” <u>(min)</u> <u>(max)</u>
PARAMETER	(parameter name) (operator) <u>(value)</u>
PARAMETER	(parameter name) <u>(value)</u>

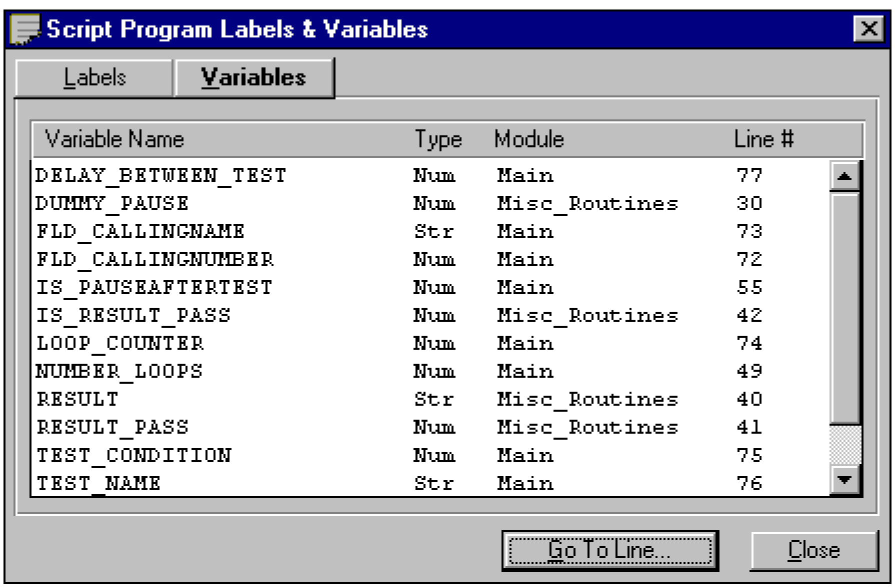
MODE	(auto/typeI/typeII)
SIGNALING	(signaling type)
MESSAGE	(message type)
PACKET	(packet type) (enabled/disabled/value) (value)
SEGMENT	(segment type) (numberbits/time) (value)
STOPBITS	(number of bits)
MARKBITS	(number of bits)
SET	DropOut (segment name) (start time) (duration)
SET	FSK (on/off)
SET	MSLEVEL (mark level) (space level)
SET	ECHOTAP (1/2/3) (delay value) (attn value)
SET	ECHOTAP (1/2/3) OFF
SET	LEVELUNITS (dBm/dBV/mVrms) (term/unterm)
SET	OUTPUTA (on/off/toggle/FSK/FSKdelay)
SET	OUTPUTB (on/off/toggle/Ring/OSI)
SET	AUTOXSUM (on/off)
SET	ROUTING (control) (option)
SET	PARITY (8none/7odd/7even)
SET	STANDARD (Bellcore/ETSI/Australia)
CLEAR	(ACKcounter/all/log/InBcount/Timer)
LOG	SAVEFILE (filename)
LOG	ACKdisplay (option)
LOG	HEADER (on/off)
LOG	PRINTING (Title, Footer) ([NONE], [DEFAULT], “custom”)
PRINTSCREEN	“text string” [(variable name)]
PRINTLOG	“text string” [(variable name)]
PRINTFILE	(filename) “text string” [(variable name)]
READFILE	(filename) (section) (keyword) (variable name)
DELETEFILE	(filename)
CHANGEBYTE	(packet type) # (address) TO (invert/inc/dec/value)
CHANGEBIT	(segment) # (bit position) TO (0,1,toggle)
CHANGEBIT	(segment) FromEnd (bit position) TO (0,1,toggle)
TONEGEN	(parameter name) (operator) (value)
TONEGEN	(parameter name) (value)
FSKGEN	MODE (single/continuous)
FSKGEN	TYPE (mark/space/alternate/user/external)
FSKGEN	DATA “text string”
FSKGEN	BITS (number of bits)
FSKGEN	GO
FSKGEN	STOP
WAIT	(time delay in seconds)
NETTONE	(filename) [NOWAIT] [parameter list]
DCIMP	(impairment) (parameter) (value)
DO	OSI (duration)
DO	TipRing Connect
DO	TipRing Disconnect

When creating a script program using variables the command builder list box will show, for most commands, any user defined variables that have been declared. For example, in the following figure four parameters have been defined using the DECLARE statement. At any time after the script program has been executed or scanned for errors, an internal list of user defined variables is created. This list is shown in the command builder list box for most commands that accept variable data. In this case, the PARAMETER command is used to set the FSK transmit level, and the list box will show all user defined variables. To select one of the variables, simply double click on it. Note that the list contains all variables, including numeric and string types. For this example, selecting a string type variable will cause an eventual error, since the FSK transmit level is numeric in nature and can not be assigned a text string value.

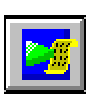


Tracking Labels & Variables

As script programs become large and complex, it becomes difficult to track a growing number of program labels and variables. To assist in managing the various labels and variables, a complete list of all the script program's labels and variables can be displayed by selecting the [SCRIPTING] [LABELS & VARIABLES] menu command. This shows a window similar to below.



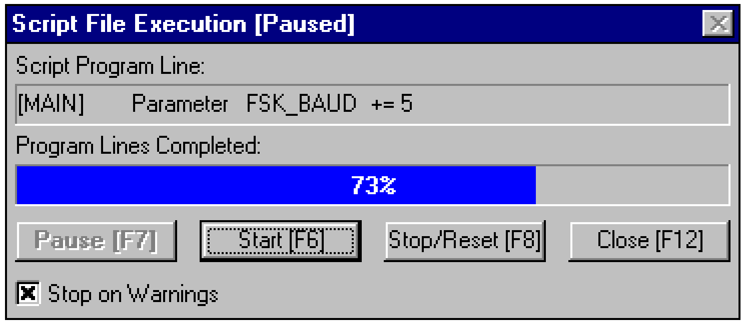
All of the current program's labels and variables are listed in alphabetical order. In addition, the module name and declaring line number are also included in the list. The above figure shows a listing of variables for an example script program. To view the defined labels, simply click the mouse on the "Labels" button at the top of the window. Likewise, clicking "Variables" lists all the declared variables. Clicking the "Go To Line" closes the window and moves the script editor cursor to the selected label or variable declaration.



Running a Script Program

When clicking the run script program icon, selecting [TRANSMIT] [RUN SCRIPT FILE] from the menu, or pressing F6, the current script file contained in the script editor is scanned for errors, then executed. See Appendix C: Script Error Reference for a list of script file syntax errors. If an error is encountered, a message will appear and the offending program line will be highlighted in the script file.

If no errors has been found, the Script Execution window will be displayed on the screen, similar to the figure below.



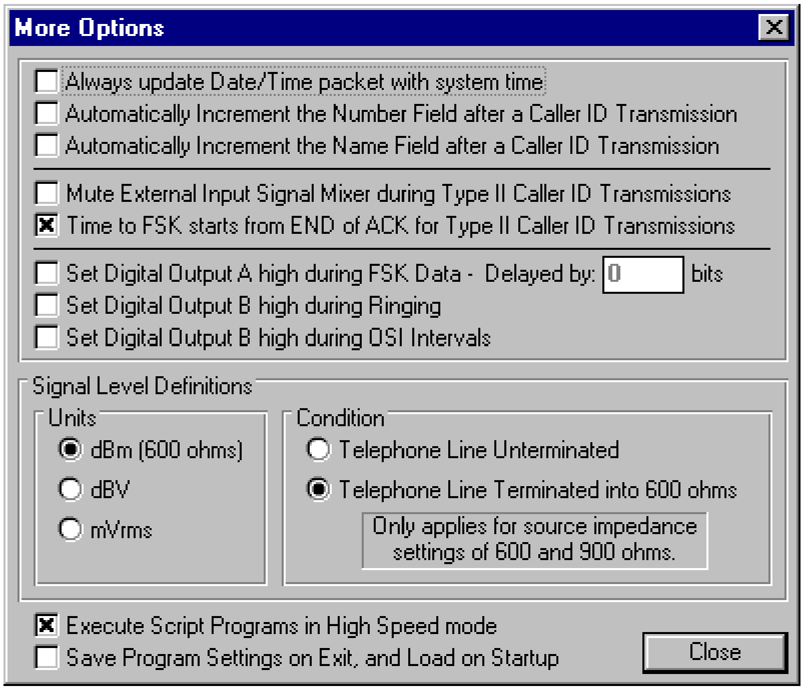
The Script Execution window shows the current script program line being processed, along with a scale showing percentage of program lines completed. This scale is only accurate assuming that no branches have been taken during the execution of the script program.

The script program execution can be controlled via the four buttons along the bottom edge of the window. The PAUSE button will suspend operation of the script program after the current command has finished its processing. Once in pause mode, the program may be restarted by pressing the START button, or pressing F6, or stopped completely by pressing the STOP button. Once stopped, the program can be restarted from the beginning with the START button, or pressing F6. The CLOSE button will remove the Script Execution window and stop any script programs that may be running.

Script Program Execution Speed

Script programs can execute at either a fixed command rate or at the maximum rate possible with the current computer system. CID1500 program versions prior to 2.31d executed script commands at a nominal rate of one command every 50 msec. This resulted in relatively consistent performance for a wide variety of computer processor speeds. However, when executing large and complex scripting programs, the nominal 50 msec command execution rate caused significant program run times. The new option allows a script program's command execution rate to be determined by the inherent speed of the computer running the CID1500 program. Depending on the processor speed, this can translate into an increase of approximately ten fold for most Pentium class processors.

The high speed script execution option is located in the More Options windows. To display the window, select the [CONFIGURATION] [MORE OPTIONS] menu command, or press the CTRL-Z key combination. The figure below shows the More Options window, with the high speed script execution option located near the bottom of the window. The default mode for the option is enabled.



Stop on Warning Option:
When the Stop on Warning option is enabled, any program warnings that are generated will cause the script program to be stopped. If disabled, program warnings will not stop the execution of the script program; however, a warning message will be written to the data log file.

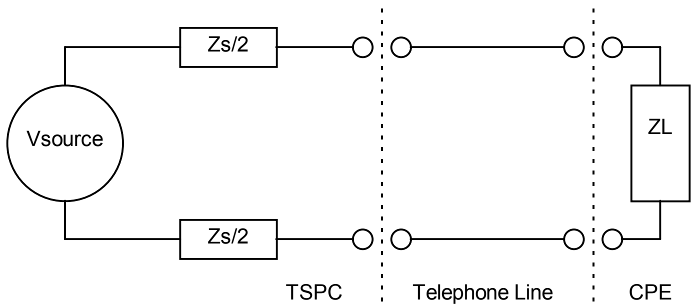
Script Program Errors:
See Appendix C: Script Error Reference for a list of script file syntax errors.

■ Section 5 Additional Information

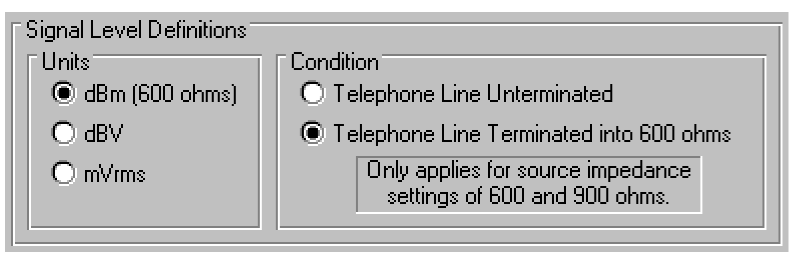
Telephone Line Signal Levels

Signal levels are a common source of confusion with most signal generation equipment. This section aims to explain how signal levels are calculated within the program.

It is important to understand how the Telephone Signal Processing Card (TSPC) delivers its AC signals to the CPE under test. The figure below shows a simplified version of the AC signal path from the TSPC to the CPE. The TSPC consists of basically a programmable AC voltage source followed by two impedance's. The two impedance's combined represent the total source impedance (Zs) of the TSPC which can be set to 600 ohms, 900 ohms, or a complex impedance in the Advance Setup window (under the category of Telephone Interface). The CPE under test is represented by the impedance ZL.



The signals generated by the software can be specified in terms of three different units. These being dBm, dBV, and mVrms. Also, the signal levels can be specified assuming a terminated telephone line of 600 ohms or an unterminated (open circuit) telephone line. These settings can be viewed and changed by selecting the [MORE OPTIONS] item under the [CONFIGURATION MENU] as shown below.



Changing the Caller ID standard setting of the program, resets the signal level definitions back to a default value associated with that standard. These default settings are:

Standard	***Units***	***Condition***
Bellcore	dBm	Terminated into 600 ohms
ETSI	dBV	Unterminated
Australia	dBm	Terminated into 600 ohms

The units of dBm in the context of this program always refers to a power level delivered into a 600 ohm load. As such, a setting of 0 dBm for a signal level will produce a voltage level equivalent to 1 milliwatt of power dissipated into a 600 ohm resistor. This voltage level works out to be 0.7746 Vrms, or -2.218 dBV. This gives an easy conversion factor when changing from dBm to dBV and vice versa, since the difference between the two unit systems is always 2.218 dB.

In the case that the signal levels are referred to an unterminated telephone line, then the level of the AC voltage source driving the telephone line will be exactly the voltage level specified any where in the program. For example, setting the tone generator to produce an output level of 0 dBm will cause the AC voltage source to product a voltage of 0.7746 Vrms. Or a setting of -20 dBV will translate into 0.100 Vrms. However the voltage measured at the telephone line may be different depending on the loading of the CPE.

The voltage at the CPE can be calculated simply as:

$$Vcpe = Vsource * \frac{ZL}{Zs + ZL}$$

where: Vcpe is the voltage level at the CPE in Vrms
Vsource is given in Vrms
ZL is the load impedance of the CPE
Zs is the source impedance of the TSPC

Alternatively, if the program is set to refer all signal levels to a terminated telephone into 600 ohms, the voltage levels will be calculated differently. This setting delivers the specified signal level to the CPE, provided that the CPE's load impedance is exactly 600 ohms. For example, if the tone generator is programmed for an output level of 0 dBm, the TSPC will delivery 1 milliwatt of signal power to the CPE provided its impedance (ZL) is 600 ohms. Or if specified in dBV, a setting of -20 dBV will cause the voltage level at the CPE to be 0.100 Vrms. This is independent of whether the TSPC's source impedance has been set to 600 ohms or 900 ohms, as the program will compensate for the different source impedance.

If the CPE's impedance (ZL) is not 600 ohms, the power level delivered to the CPE will be different from what is specified. The voltage level can be calculated in any case from the following formula:

$$Vcpe = Vspecified * \frac{Zs + 600}{600} * \frac{ZL}{Zs + ZL}$$

where: Vcpe is the voltage level at the CPE in Vrms
Vspecified is given in Vrms
ZL is the load impedance of the CPE
Zs is the source impedance of the TSPC

As can be seen from the previous formula, if the telephone line is unterminated, then the voltage levels will be 6 dB (2 times) higher than specified if the source impedance is set to 600 ohms, and 7.96 dB (2.5 times) higher if the source impedance is set to 900 ohms.

In the case that the TSPC's source impedance is set to the complex setting, the program can not compensate for the source impedance since its magnitude and phase response varies with frequency. When using the complex impedance, the signal levels will always be calculated assuming an unterminated telephone line. The above formula can still be used, but substituting a value of zero for Zs.

FSK Modulator & Noise Generator Signal Levels

FSK Modulator Levels:

The FSK modulator generates one of two different frequency tones depending on the logic level of the current data bit being transmitted. These two frequency tones are generally termed the Mark and Space tones. The Mark tone corresponds to a logic one, while the Space tone corresponds to a logic level of zero.

The signal levels for the FSK modulator can be specified in terms of either total transmit level and twist level, or mark tone level and space tone level. This selection is made on Main Settings window. The relationship between the total/twist and mark/space tone levels are defined as follows:

There are two formulae for this calculation. The first assumes that the FSK Bit Skew parameter is set to 50% (equal Mark and Space bit times). The second is more general, and allows for arbitrary bit skew values:

TxLevel = sqrt((MarkLevel^2 + SpaceLevel^2) / 2) voltage levels

Since desired Mark/Space levels and transmit levels are generally expressed in dB's, the formula becomes:

Txlevel = 10 Log10 ((10 ^ (MarkLevel / 10) + 10 ^ (SpaceLevel / 10)) / 2)

For arbitrary bit skew values the transmit level is:

TxLevel = sqrt((SK/100) * MarkLevel^2 + ((100 - SK) / 100) * SpaceLevel^2)

Expressed in the logarithmic form, it becomes:

Txlevel = 10 Log10 (10 ^ ((MarkLevel - skm) / 10) + 10 ^ ((SpaceLevel - sks) / 10))

where: skm = 10 Log10 (SK/100)

and: sks = 10 Log10 ((100-SK) / 100)

Note: SK is the FSK Bit Skew parameter value

The twist value can be calculated as the following, regardless of whether the Bit Skew parameter is taken into account.

Twist(V) = MarkLevel(V) / SpaceLevel(V)

or expressed in the logarithmic form, it becomes:

Twist(dB) = MarkLevel(dB) - SpaceLevel(dB)

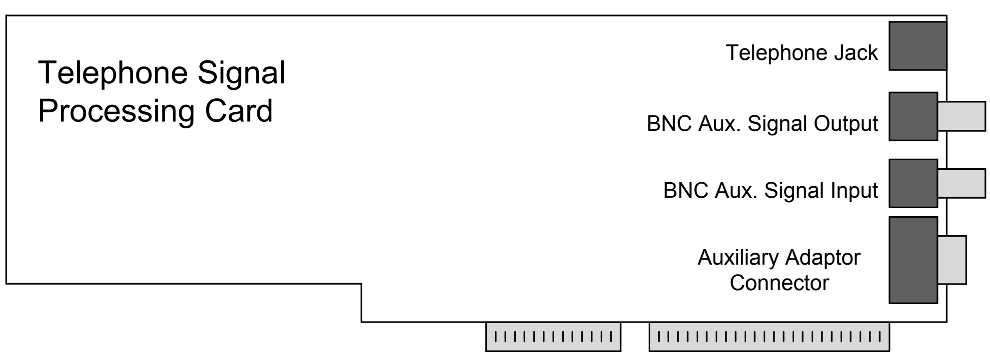
Note: FSK Twist level is defined as the difference in signal level between the Space and Mark tone. Positive twist (in dB's) is when the level of the Space tone is less than the level of the Mark tone. Correspondingly, a Space tone level higher than the Mark tone is defined as negative twist (in dB's).

Noise Generator Levels:

The TSPC noise generator produces a spectrally flat noise bandwidth to a maximum frequency of approximately 22 kHz. However, noise level specified in either dBm, dBV, or mVrms refers to the total noise power or voltage in the band of 200 Hz to 3200 Hz. As such the total noise power generated by the TSPC will be greater than the noise power specified by a factor of approximately 8.6 dB. This correction factor is also applied to the Signal-to-Noise Ratio (SNR) parameter of the FSK modulator when transmitting Caller ID data. As such, the Signal-to-Total Noise Ratio will be approximately 8.6 dB less that what is specified since the total noise power is greater than the noise power in the band of 200 Hz to 3200 Hz.

External BNC Signal Levels

The two BNC connectors on the rear of the Telephone Signal Processing Card (TSPC) can be used to monitor various signals to and from the CPE, and to inject signals directly to the CPE. The figure below shows the BNC signal output connector being located below the telephone jack, and just above the BNC signal input connector.



BNC Output Connector:
The auxiliary signal output has an output impedance of 600 ohms, and can either be set to monitor

the signal tones, FSK modulator, and noise signals generated by the TSPC, or the signals being received from the CPE connected via the telephone jack. When monitoring the signals generated by the TSPC, the output level present at the BNC connector (with a high impedance load) is equal to one quarter (12 dB less) of the AC voltage source driving the telephone interface.

For example, setting the Tone Generator for an output level of 0 dBm (terminated) with a telephone interface impedance (Zs) setting of 600 ohms, the level at the BNC connector will be -8.22 dBV into a high impedance. This is because to deliver 0 dBm of power across a 600 ohm load requires a voltage level of -2.22 dBV across the load. With a 600 ohm source impedance, the generator level must be 6 dB higher, or +3.78 dBV. Since the level at the BNC connector is 12 dB less than the AC voltage source driving the telephone line, its level will be 3.78 dBV minus 12 dB, or -8.22 dBV. If the telephone interface impedance (Zs) is changed to 900 ohms, the level at the BNC connector will be increased by 1.96 dB, as the AC voltage source driving the telephone line must be increased in order to maintain a level of 0 dBm across the load. In this case the BNC level is -8.22 dBV plus 1.96 dB equaling -6.26 dBV.

See the section dealing with Telephone Line Signal Levels for a more in-depth look at the signal levels within the telephone interface. If monitoring signals generated from the CPE, the signal level present at the BNC connector (with a high impedance load) is 6 dB less than the voltage level across the CPE. So if a level of 0 dBm is present at the CPE terminals and the CPE represents a 600 ohm load the voltage level will be -2.22 dBV. Routing this signal to the BNC output connector will give it a level of -2.22 dBV minus 6 dB equaling -8.22 dBV.

BNC Input Connector:
The auxiliary signal input has an input impedance of approximately 100 kohms. Signals injected in this input can be routed to the CPE, or to the level analyzer within the TSPC. If set to route the input signal to the CPE, the voltage generated by the AC voltage source driving the telephone interface will be four times higher than the voltage level injected at the BNC input connector.

For example, if the CPE is disconnected (open circuit on tip and ring), the voltage at the telephone line would be 12 dB more than the voltage present at the BNC input connector. If the CPE presents a 600 ohm load and the telephone interface impedance (Zs) is 600 ohms, the voltage level across the CPE would be 6 dB more than the voltage level at the BNC input connector. Changing the telephone interface's source impedance (Zs) to 900, reduces the voltage level across the CPE by 1.96 dB.

As above, see the section dealing with Telephone Line Signal Levels for a more in-depth look at the signal levels within the telephone interface. If the BNC input is routed to the level analyzer within the TSPC, the voltage read by the level meter will be the signal level at the BNC connector.

For more information on how to change, and configure the function of the auxiliary BNC connectors, see the section: Signal Flow and Routing

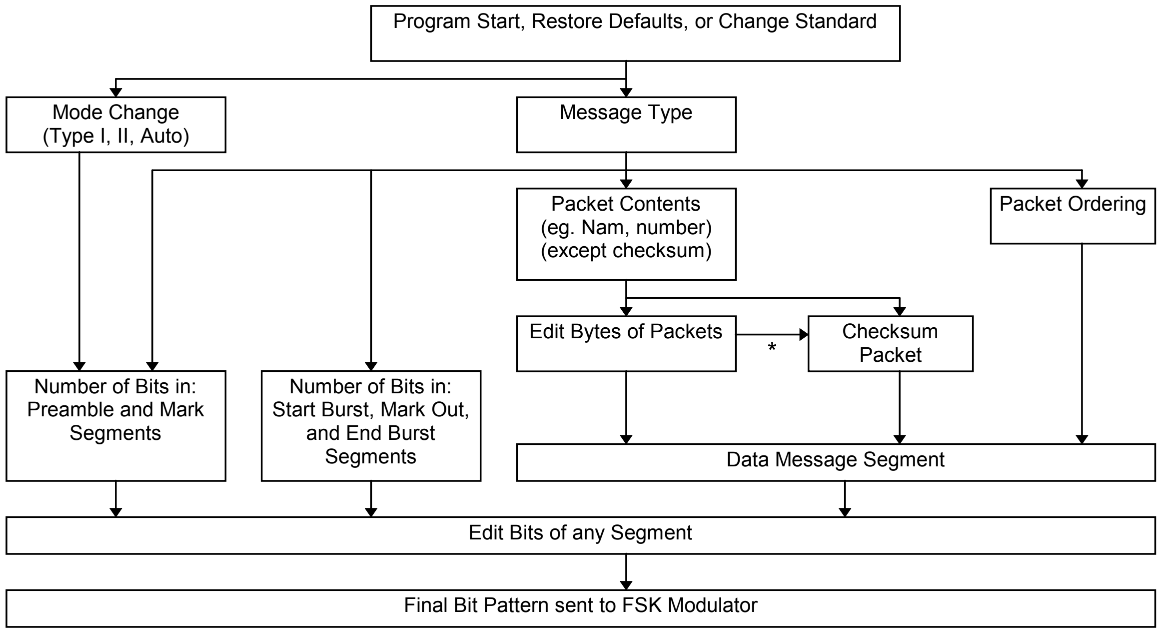
Hierarchical Caller ID Data Structure

The section explains the hierarchical relationship within the Caller ID data structure.

In order to provide the user of the CID1500 Caller ID Simulator an extremely large degree of flexibility over how and what data is sent over the telephone line, and still provide simple usage for casual users, a hierarchical data structure has been implemented.

This structure defines the relationship between the Message Types, Packets, Segments, and how editing or changing them effect each other and the final bit stream sent to the CPE. The diagram below shows how changes made to the Caller ID data effect the various messages, packets, and segments.

A simple rule of interpreting the diagram is that a change in any block will effect a change in any block with an arrow to it. For example, the highest level change is either starting the program, restoring default settings, or changing the operational standard. A change here will reset the Message Type, which resets the Packet contents, which resets the Data Message segment.



* If Auto CheckSum Calculation is Enabled

The following observations can be made from the above diagram:

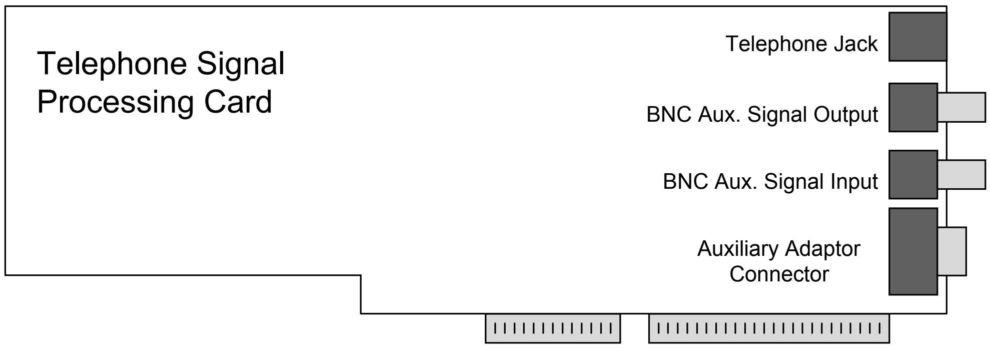
1. Changing the Message Type will reset the Packet settings
2. Changing the Packet settings (enabled, disabled, or value) will reset any bytes edited for that packet, re-calculate the message checksum, and re-calculate the Data Message segment
3. Editing the bytes of a packet will re-calculate the Data Message segment, and may re-calculate the message checksum if the Auto Checksum calculation feature is enabled
4. Changing the packet ordering will re-calculate the Data Message segment
5. A change in the transmission mode will reset the Preamble and Mark segments, which will reset any bits edited in those two segments

---

Note: If the option "Always update Date/Time packet with system time" is enabled, the values in the Date and Time text boxes will be update with the current system time every minute. As such, every minute the Date and Time packet will be recalculated, which will recalculate the message checksum, and the bit pattern of the message segment. If you are manipulating the message data at the byte of bit level, it is then recommend that this option should be disabled. This option can be enabled or disabled by selecting [CONFIGURATION] [MORE OPTIONS] from the menu. For more information on this option and its implications, see: Enabling Continuous Time Updates (Section 3-2).

Auxiliary Digital Inputs and Outputs

The DB9 pin connector at the rear of the Telephone Signal Processing Card (TSPC) provides access to general purpose digital outputs and inputs that can be used for various functions throughout the program.



The DB9 female connector allows for the use of up to 2 digital outputs, 2 digital inputs, and a +5V supply from the PC. The digital outputs are only active when Pin 7 is connected to ground. This serves as a output port enable. The outputs are driven from 5V 'HC CMOS logic. Pin 3 is defined as Output A, while pin 4 is Output B. The two available inputs are at pins 8 (Input A) and 9 (Input B). The input voltages should be limited to ground and +5V in order to prevent damage to the internal ‘HC CMOS buffers. Both Input A and Input B have 10 kohm resistors to ground in order to prevent floating inputs. The pin assignment for the DB9 connector is as follows:

Pin 1: +5V (can draw up to 0.5A from this pin) (internally fused)
Pin 2: Reserved Output (do not use)
Pin 3: External Output A (‘HC CMOS 5V output)
Pin 4: External Output B (‘HC CMOS 5V output)
Pin 5: Ground
Pin 6: Reserved Output (do not use)
Pin 7: Digital Output Enabled (connect to ground to enable outputs)
Pin 8: External Input A (‘HC CMOS 5V input)
Pin 9: External Input B (‘HC CMOS 5V input)

The two digital outputs (A & B) can be programmed high or low by scripting commands, or they can be set to indicate certain activities such as FSK data transmission, ringing, or OSI signaling. This feature can be enabled or disabled in the [MORE OPTIONS] settings panel under the [CONFIGURATION] menu. Output A can be set to be active during FSK data transmissions, while output B can be set active during times of ringing or OSI signaling. For earlier hardware that does not support the generation of OSI for the Australian standard, output B can be used to control an external relay that breaks the circuit to the CPE in order to simulate an OSI. Other uses for these output signals include the ability to trigger external equipment (oscilloscopes, logic analyzers, emulators) during the FSK or ringing portions of a Caller ID transmission.

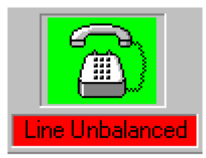
The two digital inputs can be used to control the execution of a script language program. The BRANCHIF command can be used to branch the execution point of the script program if either of the digital inputs are currently at a high level.

Telephone Line Unbalance and Grounding

As with any system of instruments and devices, proper grounding is essential in order to minimize circuit hum and susceptibility to interference. It is important to understand the circuit grounds of all the devices in the test setup to ensure their proper connection.

Like most PC based instruments the TSPC's ground point is referenced to the PC's ground, which should be connected to earth ground. The ground conductor on the BNC input and output and the DB9 connector are at earth ground potential, assuming the computer is properly grounded. The tip and ring leads of the telephone interface have a negative potential with respect to ground depending on the programmed line voltage. At the default setting of 48 volts, the voltages present on tip and ring are approximately -52 volts and -4 volts with respect to the computer's ground.

Care must be taken with CPE's that do not maintain isolation from the telephone interface. Most Caller ID adjunct units fall into this category. This can cause grounding problems when connecting the ground leads of oscilloscopes or other equipment that is normally earth grounded. The most common cause is that the CPE connects either tip or ring to its circuit ground via a bridge rectifier. If the CPE's circuit ground is then connected to earth ground by use of an oscilloscope ground lead, an emulator, or logic analyzer, then this effectively shorts either the tip or ring line to earth ground. The imbalance in loop current is detected by the TSPC and will be shown as a line unbalanced condition.



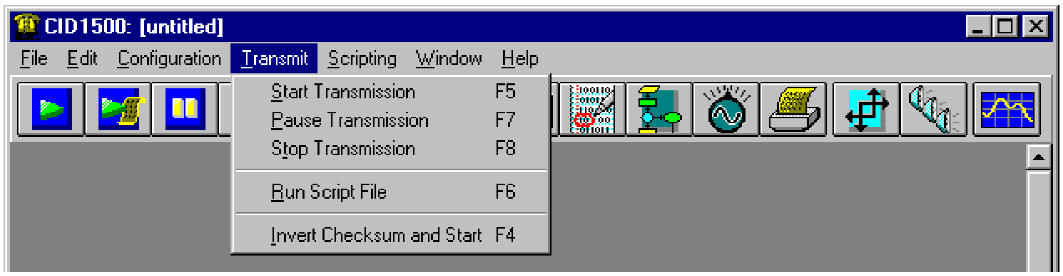
If the CPE has an isolated telephone interface, then this condition should not occur, since the CPE's circuit ground is not connected to its telephone interface circuitry. If this is not the case, and external equipment must be connected to the CPE under test that forces its ground to be connected with earth ground, there are a few possible solutions available. In some cases, the telephone interface is not a requirement for the testing situation, and the BNC inputs or outputs can be used, since the BNC's are ground referenced unbalanced signals. If the telephone interface is required, then the grounding "loop" must be broken. This can mean isolating the PC that contains the TSPC, or the test equipment connected to the CPE, or the telephone line itself. Depending on what testing is required, the telephone line can be isolated via an AC coupling transformer and a DC feeding bridge. Of course, the DC supply would have to be isolated from earth ground, but this is generally easier to do than isolating the PC.

Note: Disconnecting the earth ground wire from the computer is both an improper and potentially dangerous method of isolating the computer from the remainder of the test setup. The AC isolation achieved is generally poor due to the large parasitic capacitance's to earth ground. An alternative approach may be the use of a UPS. UPS's are an excellent method of achieving near perfect isolation between circuit grounds.

■ Section 6 Program Reference

Menu Options

This section lists each command available at the program's command line, along with its function.



The program menu bar contains the following headings.

1) FILE menu heading
2) EDIT menu heading
3) CONFIGURATION menu heading
4) TRANSMIT menu heading
5) WINDOW menu heading
6) SCRIPTING menu heading
7) HELP menu heading

The FILE Menu Options

The options under the FILE menu are the following:

[NEW SCRIPT FILE] This option erases the current script file contained in the script file editor window, such that a new script file can be entered.

[OPEN SCRIPT FILE] Selecting this options opens up a dialog box where a saved script file can be loaded into the script file editor window. The program will only work with one script file at any given time, so the original script file will be lost, if not saved.

[SAVE SCRIPT FILE] This menu item will save the current script file contained in the script editor window with the current script file name. If no previous file name has been given, then a dialog box will allow the user to enter the file name for the script program.

[SAVE AS SCRIPT FILE] This menu item will save the current script file contained in the script editor window. A dialog box will allow the user to enter the file name used to save the script program.

[SAVE AS LOG FILE] The menu item will save the current log file as contained in the log file window. A dialog box will allow the user to enter the file name used to save the log file.

[PRINT PARAMETER SETTINGS] Selecting this option sends a snapshot of all the parameter settings to the printer.

[PRINT SCRIPT PROGRAM] This option prints a copy of the current script file contained in the script file editor window.

[PRINT DATA LOG] This option prints a copy of the current data log file contained in the data log file window.

[PRINT ACK ANALYSIS WINDOW] Selecting this menu item will print the contents of the current ACK analysis window. This option will be disabled if the ACK analysis window does not contain any data to print. Since a large amount of data can be contained within the ACK analysis window, depending on the printer's speed, the output can take significant time to generate.

[PRINTER SETUP] If various print devices are connected, this option allows the user to select the preferred printer.

[EXIT] This option will shut down the telephone signal processing card, and end the program's operation. Before terminating, the program will save its window position on the screen, currently selected standard, and the last four configuration files used. When the program is restarted at a later time, this information will be restored.

The EDIT Menu Options

The options under the EDIT menu are the following:

[CUT] Any text that is currently selected or highlighted will be removed and placed into the Windows clipboard. Once inside the clipboard, the text may be transferred to other Windows programs by PASTE'ing it out of the clipboard.

The shortcut key for this command is: **CTRL-X**

[COPY] Any text that is currently selected or highlighted will be copied into the Windows clipboard. Once inside the clipboard, the text may be transferred to other Windows programs by PASTE'ing it out of the clipboard.

The shortcut key for this command is: **CTRL-C**

[PASTE] If any text is contained within the Windows clipboard, it can be placed into the current text region that has the focus at the insertion point. If text within the text region has been selected, the text from the clipboard will replace the selected text.

The shortcut key for this command is: **CTRL-V**

[SELECT ALL] This option will select or highlight all of the text in the current text region.

The shortcut key for this command is: **CTRL-A**

[CLEAR] Any text that is currently selected will be deleted. This option does not effect the text within the Windows clipboard.

[INSERT PACKET] This command will insert a new packet into the Packet Ordering list in the CID Packet Ordering window. This command is used to create and insert a new packet into the message. No data is contained within the newly created packet. To add data into this packet, use the packet editor and insert the data bytes to be contained in the new packet.

Note: This command will only be enabled if the Packet Ordering list in the CID Packet Ordering Window has been selected.

[COPY PACKET] Similar to the [INSERT PACKET] command, this option will copy the selected packet in the Packet Ordering list. The copied packet will contain the same data as the original packet it was copied from.

Note: This command will only be enabled if the Packet Ordering list in the CID Packet Ordering Window has been selected.

[DELETE PACKET] This command will cause the currently selected packet in the Packet Ordering list in the CID Packet Ordering Window to be deleted. All the data associated with that packet will be erased from the Caller ID message.

Note: This command will only be enabled if the Packet Ordering list in the CID Packet Ordering Window has been selected.

[FIND TEXT] Only enabled when the cursor is present in the script editing window. This command displays a dialog box that can search the script editor for text strings. See section 4: How the Script Program Editor Works for more information.

The shortcut key for this command is: **CTRL-F**

[REPLACE TEXT] Only enabled when the cursor is present in the script editing window. This command displays a dialog box that can search the script editor for text strings and replace found text fragments with other text See section 4: How the Script Program Editor Works for more information.

The CONFIGURATION Menu Options

The options under the CONFIGURATION menu are the following:

[SIGNAL ROUTING] Selecting this option will display the Signal Routing Window. From this window, the user can configure the telephone signal processing card to various types to signal flow conditions.

The shortcut key for this command is: **CTRL-R**
See Also: Signal Flow and Routing (Section 3-6)

[ECHO GENERATION] This menu item displays the Echo Control window, which can be used to enable or disable the generation of signal echoes.

The shortcut key for this command is: **CTRL-E**
See Also: Signal Flow and Routing (Section 3-6)

[DC LINE IMPAIRMENTS] This menu item displays the DC Line Impairment window. Various dc line conditions are specified in this window, for certain signaling types.

The shortcut key for this command is: **CTRL-I**
See Also: DC Line Impairments (Section 3-10)

[SAVE AS CONFIGURATION] This option will display a dialog box where the user can select a file name for saving the program's configuration. The resulting configuration file will contain a snap shot of all parameters, packet data, segment data, script program, data log file, and program options. This configuration file can then be loaded into the program at any time to quickly return to the same state as when it was saved.

[LOAD CONFIGURATION] This option will display a dialog box where the user can select a configuration file to be loaded into the program. The configuration files contain an exact snap shot of all of the parameters, packet data, segment data, script program, data log file, and program options the program contained at the time the configuration file was saved.

[1,2,3,4] The four menu options contain the names of the last four configuration files used. Selecting the menu option with the desired configuration file, will load that configuration file into the program. This provides a shortcut method of loading configuration files that are frequently used.

[RESTORE DEFAULTS] This option will set all parameters, packets, and segment data back to their default value for the selected standard. The script and data log files will be cleared and the telephone signal processing card reset to the default settings associated with the selected standard. Effectively, this command results in the same actions as quitting and restarting the program.

[CALLER ID STANDARD] The Caller ID standard is set with this menu option. Selecting a new standard will reset all parameter values, packets, and segment data back to their default value for the selected standard. The script and data log files will also be cleared.

[MORE OPTIONS] This selection presents additional program options. A feature to continuously update the Date and Time packet fields with the current system time can be enabled or disabled. Also, a feature to mute the external input signal mixer during Type II Caller ID transmission and be enabled or disabled.

See also: Muting the External Input Signal Mixer and Enabling Continuous Time Updates (Section 3-2)

The shortcut key for this command is: **CTRL-Z**

[VIEW CONFIGURATION FILE NOTES] Selecting this menu item displays any ASCII text notes associated with the current configuration file. The contents of the notes can be view and altered, along with setting options controlling if they are displayed automatically when loading and saving the file.

The TRANSMIT Menu Options

The options under the TRANSMIT menu are the following:

[START TRANSMISSION] Selecting this option starts a Caller ID transmission using the current parameters, packet data, and segment data options. Once the transmission has started, changing any of the parameters, packet, or segment data options will no effect on the current Caller ID transmission underway. Any changes will take effect when the next transmission is started. While a Caller ID transmission is in progress, selecting this option will have no effect.

The shortcut key for this command is: **F5**

[PAUSE TRANSMISSION] This option will suspend a Caller ID transmission currently in progress. If no transmission is taking place, this command will have no effect. Once paused, the transmission can be resumed by selecting [START TRANSMISSION], or terminated by selecting [STOP TRANSMISSION]. In the suspended state, changing any parameters, packet, or segment data options will have no effect on the transmission when it is restarted.

The shortcut key for this command is: **F7**

[STOP TRANSMISSION] Selecting this option will terminate any Caller ID transmission that is running or in a paused state. If no Caller ID transmission is taking place, this command will have no effect.

The shortcut key for this command is: **F8**

[RUN SCRIPT FILE] This command starts the execution of the script file program contained in the Script File Window. The script file will first be scanned for syntax errors, before execution begins. Once executing, the Script File Execution window will be displayed. From this window, the progress of the script file can be monitored along with pausing, or stopping its execution.

The shortcut key for this command is: **F6**

See Also: Running Script Files (Section 4)

[INVERT CHECKSUM and START] This menu command functions identically to the regular start command (F5), except that before the Caller ID transmission is initiated, the checksum byte of the message is bit-wise inverted.

The shortcut key for this command is: **F4**

The SCRIPTING Menu Options

The options under the SCRIPTING menu are the following:

[ADD NEW MODULE] The menu option is used to add another code module to the scripting program. The maximum number of code modules supported is 32. When adding a new module, a name for the module will be required. The maximum length of the name is 15 characters.

[RENAME CURRENT MODULE] Select this menu item to change the name of a scripting code module. This menu item will be disabled if the current module is titled 'Main', as this is the primary module and can not be renamed.

[REMOVE CURRENT MODULE] This menu option is used to delete scripting code modules. The currently selected module will be deleted if this menu options is selected. If the 'Main' module is selected, then this menu item will be disabled, as the primary code module can not be deleted.

[SCAN CURRENT MODULE FOR ERRORS] Selecting this menu item will cause a syntax scan of the currently selected scripting code module. All of the commands will be checked for proper syntax. However, references to label will not be checked, as they may reside outside the current module being scanned.

[SCAN ALL MODULES FOR ERRORS] This menu option will check all of the scripting code modules for syntax errors. Unlike the scanning of only the current module, all label references defined in the entire script program will verified by this menu option.

The WINDOW Menu Options

The options under the WINDOW menu are the following:

[TILE HORIZONTAL] This menu option will arrange all open windows in a horizontal pattern across the main program window.

[CASCADE] This menu option will arrange all open window in a cascading pattern from top left to bottom right.

[ARRANGE ICONS] This menu option will arrange all minimized windows along the bottom of the main program window.

[1 to 9] Selecting any of these menu options will bring the associated window to the front, and place all other windows behind it.

The HELP Menu Options

The options under the HELP menu are the following:

[CONTENTS] This menu option displays the help window table of contents. From the table of contents, specific areas of program help can be reached by clicking on the underlined topic heading.

The shortcut key for this command is: **F1**

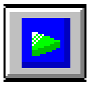
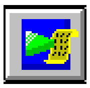
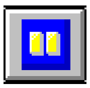
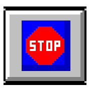
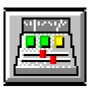
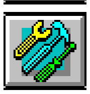
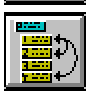
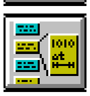
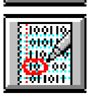
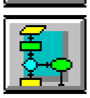
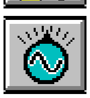
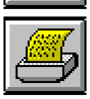
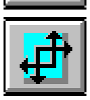
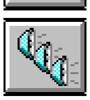
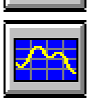
[SEARCH FOR HELP ON] This menu option displays the Windows Help Search dialog box. By typing key words, the help program will display help topics that relate to the key word entered.

[TECHNICAL SUPPORT] This menu option shows how you can contact us for questions concerning the product.

[ABOUT CID1500] This menu option displays the program version code, along with the revision and product codes for the telephone signal processing card. The base I/O address used to communicate with the telephone signal processing card is also displayed.

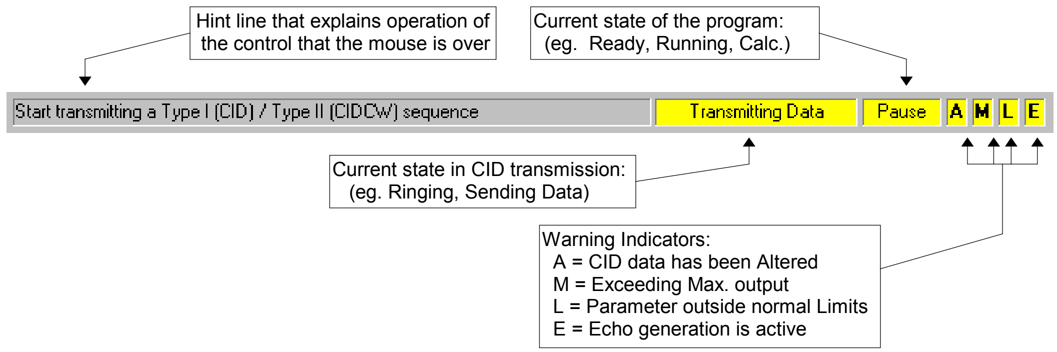
The Toolbar

The toolbar gives a shortcut method for accessing various program functions contained in the menus. The function of each toolbar button is described below. For more information on each menu item, see the section: Program Reference: Menu Options

ICON	***Description of Action***	***Equivalent Menu Option***
	Start Caller ID Transmission:	[TRANSMIT] [START TRANSMISSION]
	Start Executing the Script Program	[TRANSMIT] [RUN SCRIPT FILE]
	Pause Caller ID Transmission	[TRANSMIT] [PAUSE TRANSMISSION]
	Stop Caller ID Transmission	[TRANSMIT] [STOP TRANSMISSION]
	Main Settings Window	[WINDOW] [MAIN SETTINGS]
	Advance Setup Window	[WINDOW] [ADVANCE SETTINGS]
	Packet Format Window	[WINDOW] [CID PACKET FORMAT]
	Transmission Format Window	[WINDOW] [TRANSMISSION FORMAT]
	Packet/Segment Editor Window	[WINDOW] [EDIT PACKET/SEGMENT DATA]
	Script Editor Window	[WINDOW] [SCRIPT EDITOR]
	Tone Generator Window	[WINDOW] [TONE GENERATOR]
	Log File Window	[WINDOW] [DATA LOG]
	Signal Routing Window	[CONFIGURATION] [SIGNAL ROUTING]
	Echo Control Window	[CONFIGURATION] [ECHO GENERATION]
	DC Impairment Window	[CONFIGURATION] [DC LINE IMPAIRMENTS]
	ACK Analysis Window	[WINDOW] [ACK ANALYSIS]

The Status Line

At the bottom of the master window, is the status line. Here you can instantly determine the current state of the program. The following diagram helps to illustrate some examples.



The first frame, the Hint Line, helps to explain the operation of most of the controls in the sub-windows, whenever the mouse is placed over them. The next frame shows the current state of any Caller ID transmissions taking place. When a transmission is active, it will be highlighted in yellow, with the text indicating the current action being taken. The following frame indicates the status of the program. It will display “READY” at idle times. During data transmissions, it will display either “RUNNING” or “PAUSE”. Also, when performing calculations, it may display “CALC.”. "TONE ON" will be indicated if the tone generator is currently active. If the FSK modulator feature is being used, "FSK MOD" will be shown in the status frame.

The last four frames are warning indicators. These help remind the user that certain settings have been set to unusual values, or are outside recommended limits.

The first indicator will illuminate with an “A” to signify an altered CID data stream. This means that the CID data has been changed in an unusual manner. Possible causes are editing the data bytes in a packet, editing the bits in segment, having extra packets inserted or copied into the Packet Ordering list, or re-ordering the packets in a manner that results in an invalid data message. Also, the Caller ID message will be considered as altered if a FSK Modulator drop-out is specified for a segment. This is done in the Transmission Format window.

The second indicator displays a “M” whenever the Telephone Signal Processing Card (TSPC) has been directed to generate simultaneous tones at a level that exceeds its maximum undistorted output capability.

The third indicator will display a “L” anytime transmission parameters fall outside the Bellcore or ETSI specifications. If a parameter value has been changed to a value that is outside the range specified by the selected standard, the "L" indicator will be shown. Not all transmission parameters are defined with specification maximums or minimums. To determine what are the specification limits, if any at all, located the desired parameter under the Help Section: <u>Changing the Transmission Parameters</u>

Finally, the fourth indicator will show an “E” if any signal echoes are being created. Any signals generated by the program can have up to three time delayed echoes of varying amplitude also generated. In order to reminder the user that echoes are being generated, the “E” will be displayed in the status panel.

Shortcut Keys

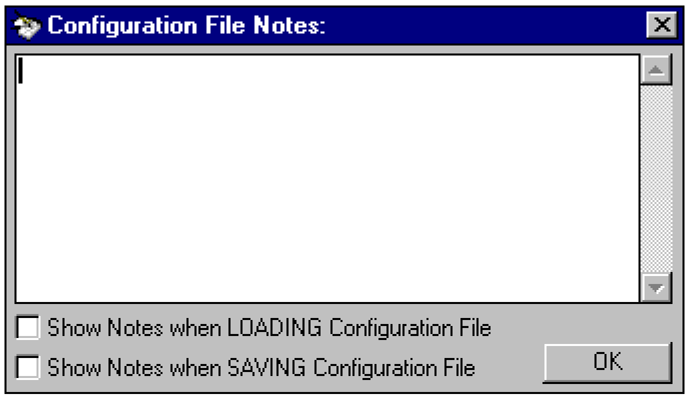
Pressing the shortcut keys are equivalent to selecting its associated menu option.

File - Save Script File	**CTRL-S**
Edit - Cut	**CTRL-X**
Edit - Copy	**CTRL-C**
Edit - Paste	**CTRL-V**
Edit - Select All	**CTRL-A**
Edit - Find Text	**CTRL-F**
Signal Routing Window	**CTRL-R**
Echo Control Window	**CTRL-E**
DC Impairments Window	**CTRL-I**
More Options Window	**CTRL-Z**
Help	**F1**
Start CID Transmission	**F5**
Pause CID Transmission	**F7**
Stop CID Transmission	**F8**
Invert Checksum and Start	**F4**
Start Script Program	**F6**
Pause Script Program	**F7 ***
Stop Script Program	**F8 ***
Script File Program Branch	**F9 ***
Terminate Script Program	**F12 ***
View Labels & Variables	**CTRL-L**

* Only available when running a script program

Viewing Configuration File Notes

Text notes can be stored as part of the configuration files. The notes can contain any text information of up to 10,000 characters that is always contained within the configuration file. Two options are included that if set, display the text notes when the configuration file is loaded, or when the configuration file is saved. Otherwise, the notes can be viewed and edited by selecting the [CONFIGURATION] [VIEW CONFIGURATION FILE NOTES] menu command.



The options to view the text notes during loading and saving the configuration file are controlled by the two checkboxes at the bottom of the previous figure. These settings are stored as part of the configuration file. As such, if saving a configuration with the option to view on loading is enabled, then anytime the file is loaded, the notes are displayed. This is independent of the previous option settings. Likewise for the option to display the text notes when saving the configuration file.

Loading Script or Config Files at Program Start

It is possible to load a script program or configuration file automatically at the program startup. This is done by specifying the file to load on the program command line. For example, the default command line that windows uses to run the CID1500 program is:

CID1500

However, this can be changed to specify the immediate loading of a script or config file at program start by appending either “s=(script file name)”, or “c=(config file name)”. Two examples of this are given below.

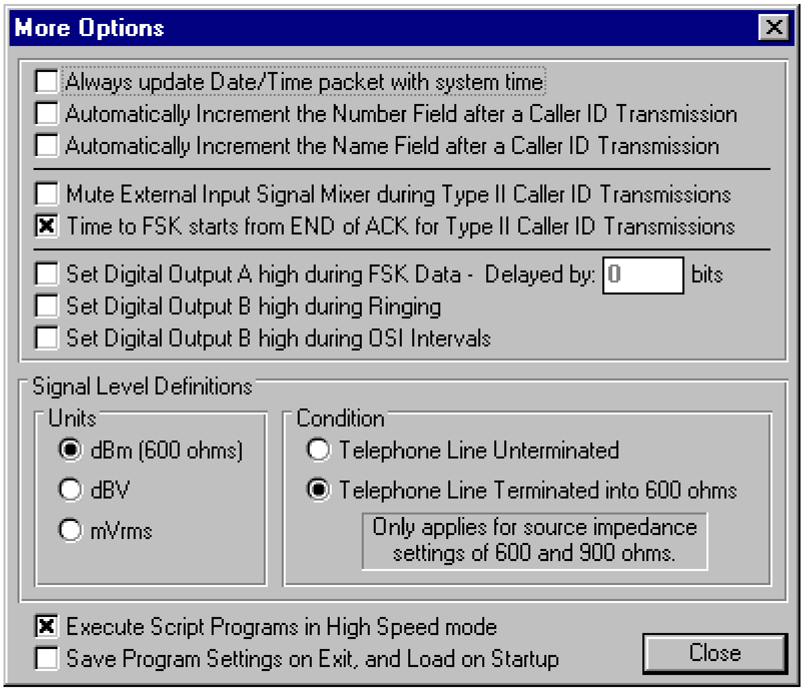
CID1500 s=sr3004_A.scr
CID1500 c=c:\test1.cfg

The first example loads the script file called “sr3004_A.scr” that is located in the same program directory as the CID1500 program. The second example loads the configuration file “test1.cfg” from the root directory of C drive.

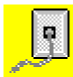
It is also possible to automatically load and begin executing a scripting program at startup. By adding either “sr=(script file name)”, or “cr=(config file name)” to the command line, once the scripting or configuration file has loaded, the script program will start to run.

Automatic Configuration Save and Load

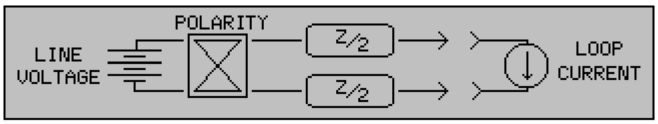
The CID1500 program can be set to automatically save its entire configuration at shutdown, and then retrieve its last settings during program startup automatically. This feature is enabled or disabled with the check box at the bottom of the More Options window. If enabled, when the program is closed, it will save its configuration to a file called “startup.cfg”. The next time the CID1500 program is started, it will load that configuration file (if present).



■ Appendix A Transmission Parameters



Parameter Category: Telephone Interface



Parameter: Telephone Line Voltage

Parameter Description:
This parameter sets the voltage present on the telephone line when the telephone is in an on hook state (drawing no current). Once the telephone goes off hook, a constant current mode of operation is engaged, where the setting of this parameter is not relevant. Any changes made in the parameter will take effect immediately.

Details:

Title:	Telephone Line Voltage		
Script Name:	LINE_VOLTAGE		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Volts	Volts	Volts
Default Value:	**48**	**48**	**48**
Maximum Value:	52	52	52
Minimum Value:	20	20	20
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Telephone Loop Current

Parameter Description:
This parameter sets the loop current flowing through the telephone line when the telephone is in an off hook state (drawing current). Once the telephone goes on hook, a constant voltage mode of operation is engaged, where the setting of this parameter is not relevant. Any changes in the parameter will take effect immediately.

Note: In the off hook state, the constant current mode of operation will only be maintained if the DC voltage across the CPE is less than the setting of the parameter: Telephone Line Voltage. If this condition is not met, then the loop current will fold back to a lower level. For example, setting a loop current of 40 mA, with a CPE that has 600 ohms of DC resistance will cause a line voltage of 40mA x 600 ohms or 24 Volts. However, if the Telephone Line Voltage parameter is set to 20 Volts, then it would be impossible to feed 40 mA through the CPE.

Details:

Title:	Telephone Loop Current		
Script Name:	LOOP_CURRENT		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mA	mA	mA
Default Value:	**26**	**26**	**26**
Maximum Value:	40	40	40
Minimum Value:	20	20	20
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Telephone Line Impedance

Parameter Description:
This parameter selects the AC impedance presented by the telephone line. The valid settings can be either a real 600 ohms, 900 ohms, or a complex impedance. For more infomation on the default complex impedance see Section 1, Installation and Setup.

Details:

Title:	Telephone Line Impedance		
Script Name:	LINE_IMP		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**900 ohms**	**600 ohms**	**600 ohms**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

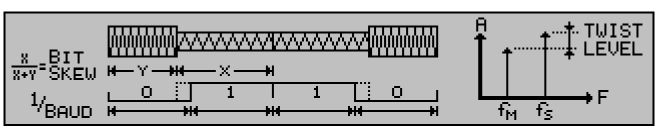
Parameter: Telephone Line Polarity

Parameter Description:
The Line Polarity parameter selects the voltage and current polarity of the telephone interface. The two options are normal and reversed

Details:

Title:	Telephone Line Polarity		
Script Name:	LINE_POLARITY		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**Normal**	**Normal**	**Normal**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter Category: FSK Modulator



Parameter: FSK Transmit Level

Parameter Description:
This parameter sets the level of the FSK signal used to transmit the Caller ID data. Unlike most parameters, this parameter can be changed in the Main Settings window, as opposed to the Advanced Setup window.

Details:

Title:	Transmit FSK Signal Level		
Script Name:	FSK_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	mVrms	mVrms
Default Value:	**163.7**	**316.2**	**163.7**
Maximum Value:	1000	1000	1000
Minimum Value:	0	0	0
Maximum Standard Value:	194.5	398.1	194.5
Minimum Standard Value:	12.3	10	12.3

Parameter: FSK Transmit Signal-to-Noise Ratio

Parameter Description:
This parameter specifies the signal-to-noise ratio between the transmit FSK level and the noise generator. The larger the value, the lower the noise level is with respect to the FSK signal. Note, the noise level is defined as the noise power in the frequency band from 200 Hz to 3200 Hz. However, the noise spectrum produced by the noise generator is flat to 20 kHz. As a result the broad band noise power produced by the noise generator will be higher than that within the 200 Hz to 3200 Hz bandwidth.

Details:

Title:	Transmit FSK Signal to Noise Ratio		
Script Name:	FSK_SNR		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	dB	dB	dB
Default Value:	**40**	**40**	**40**
Maximum Value:	40	40	40
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	25	25	25

Parameter: FSK Mark Tone Frequency

Parameter Description:
This parameter specifies the frequency of the Mark tone produced by the FSK generator.

Details:

Title:	FSK Mark Tone Frequency		
Script Name:	FSK_MFREQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	Hz	Hz
Default Value:	**1200**	**1300**	**1200**
Maximum Value:	5000	5000	5000
Minimum Value:	100	100	100
Maximum Standard Value:	1212	1320	1212
Minimum Standard Value:	1188	1281	1188

Parameter: FSK Space Tone Frequency

Parameter Description:
This parameter specifies the frequency of the Space tone produced by the FSK generator.

Details:

Title:	FSK Space Tone Frequency		
Script Name:	FSK_SFREQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	Hz	Hz
Default Value:	**2200**	**2100**	**2200**
Maximum Value:	5000	5000	5000
Minimum Value:	100	100	100
Maximum Standard Value:	2222	2132	2222
Minimum Standard Value:	2178	2070	2178

Parameter: FSK Twist Level

Parameter Description:
The FSK Twist Level defines the difference in signal level between the Mark and Space tones that make up the FSK signal. Positive twist is defined to be a Mark signal level that exceeds the Space signal level. Likewise, for a negative twist, the Space signal level exceeds the Mark signal level.

Details:

Title:	FSK Twist Level		
Script Name:	FSK_TWIST		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	dB	dB	dB
Default Value:	**0**	**0**	**0**
Maximum Value:	60	60	60
Minimum Value:	-60	-60	-60
Maximum Standard Value:	10	6	10
Minimum Standard Value:	-10	-6	-10

Parameter: FSK Baud Rate

Parameter Description:
This parameter specifies the baud, or symbol rate of the FSK modulator. For the modulation used, the baud rate is equivalent to the bit rate.

Details:

Title:	FSK Baud Rate		
Script Name:	FSK_BAUD		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	bps	bps	bps
Default Value:	**1200**	**1200**	**1200**
Maximum Value:	5000	5000	5000
Minimum Value:	100	100	100
Maximum Standard Value:	1212	1212	1212
Minimum Standard Value:	1188	1188	1188

Parameter: FSK Mark/Space Bit Skew

Parameter Description:
This parameter specifies any timing skew between Mark and Space bits. It is defined as a percentage value, where 50% represents equal bit duration between Mark and Space bits. A value greater than 50% increases the bit duration of the Mark bits at the expense of the Space bits. Likewise, a value of less than 50% decreases Mark bit duration's, with an increase in Space bit duration's. The bit skew can be calculated as the ratio of Mark bit time to the sum of Mark and Space bit times.

Details:

Title:	FSK Bit Skew		
Script Name:	FSK_SKEW		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	%	%	%
Default Value:	**50**	**50**	**50**
Maximum Value:	75	75	75
Minimum Value:	25	25	25
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: FSK Interfering Tone Enable

Parameter Description:
This parameter enables or disables an interfering tone that is generated in conjunction with the FSK modulated signal.

Details:

Title:	FSK Interfering Tone Enable		
Script Name:	FSK_IENABLE		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**Disable**	**Disable**	**Disable**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: FSK Interfering Tone Frequency

Parameter Description:
This parameter specifies the frequency of an interfering tone that is generated simultaneously with the FSK modulated signal. The interfering tone will not be produced unless it has been enabled with the FSK Interfering Tone Enable parameter.

Details:

Title:	FSK Interfering Tone Frequency		
Script Name:	FSK_IFREQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	Hz	Hz
Default Value:	**1000**	**1000**	**1000**
Maximum Value:	10000	10000	10000
Minimum Value:	20	20	20
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

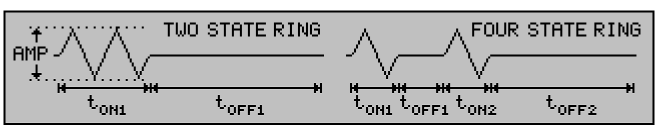
Parameter: FSK Interfering Tone Level

Parameter Description:
This parameter specifies the level of an interfering tone that is generated simultaneously with the FSK modulated signal. The interfering tone will not be produced unless it has been enabled with the FSK Interfering Tone Enable parameter.

Details:

Title:	FSK Interfering Tone Level		
Script Name:	FSK_ILEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	mVrms	mVrms
Default Value:	**0**	**0**	**0**
Maximum Value:	2000	2000	2000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter Category: Ringing Generator



Parameter: Ring Generator Frequency

Parameter Description:
This parameter defines the frequency of the Ringing Generator.

Details:

Title:	Ring Generator Frequency		
Script Name:	RING_FREQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	Hz	Hz
Default Value:	**22**	**22**	**22**
Maximum Value:	1000	1000	1000
Minimum Value:	10	10	10
Maximum Standard Value:	68	68	68
Minimum Standard Value:	13	13	13

Parameter: Ring Generator Level

Parameter Description:
This parameter specifies the level of the Ringing Generator. The maximum ringing level that can be generated is 80 Vrms; however, this is dependent on the setting of the Telephone Line Voltage. Below a setting of 48 Volts for the Telephone Line Voltage, the maximum unclipped ring generator level must be de-rated. At the minimum Telephone Line Voltage level of 20 V, the maximum unclipped ringing voltage is 60 Vrms.

Details:

Title:	Ring Generator Level		
Script Name:	RING_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Vrms	Vrms	Vrms
Default Value:	**50**	**50**	**50**
Maximum Value:	80	80	80
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	40	40	40

Parameter: Ring Generator Sequence

Parameter Description:
This parameter determines whether the Ring Generator pattern is either a 2 state or 4 state ringing pattern. The two state ringing is a single ringing burst followed by silence. The four state ringing is two ringing bursts followed by silence. Similarly, six state ringing allows for three ringing bursts.

Details:

Title:	Ring Generator Sequence		
Script Name:	RING_SEQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**2 State**	**4 State**	**4 State**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Ring Generator On Time #1

Parameter Description:
This parameter defines the length of the ringing burst generated for the two state ringing pattern. For the four and six state pattern, this parameter defines the length of the **first** ringing burst.

Details:

Title:	Ring Generator On Time #1		
Script Name:	RING_ON1		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**2000**	**700**	**400**
Maximum Value:	100000	100000	100000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Ring Generator Off Time #1

Parameter Description:
This parameter defines the length of the silence interval for the two state ringing pattern. For the four and six state pattern, this parameter defines the length of the time between the first and second ringing bursts.

Details:

Title:	Ring Generator Off Time #1		
Script Name:	RING_OFF1		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**4000**	**600**	**200**
Maximum Value:	100000	100000	100000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Ring Generator On Time #2

Parameter Description:
This parameter is used in conjunction with the four or six state ringing pattern. It specifies the duration of the second ringing burst.

Details:

Title:	Ring Generator On Time #2		
Script Name:	RING_ON2		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**500**	**600**	**400**
Maximum Value:	100000	100000	100000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Ring Generator Off Time #2

Parameter Description:
This parameter is used in conjunction with the four or six state ringing pattern. It specifies the duration of the silence interval after the second ringing burst.

Details:

Title:	Ring Generator Off Time #2		
Script Name:	RING_OFF2		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**4000**	**4000**	**2000**
Maximum Value:	100000	100000	100000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Ring Generator On Time #3

Parameter Description:
This parameter is only used in conjunction with the six state ringing pattern. It specifies the duration of the third ringing burst.

Details:

Title:	Ring Generator On Time #3		
Script Name:	RING_ON3		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**500**	**500**	**500**
Maximum Value:	100000	100000	100000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Ring Generator Off Time #3

Parameter Description:
This parameter is only used in conjunction with the six state ringing pattern. It specifies the duration of the silence interval after the third and last ringing burst.

Details:

Title:	Ring Generator Off Time #3		
Script Name:	RING_OFF3		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**4000**	**4000**	**4000**
Maximum Value:	100000	100000	100000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

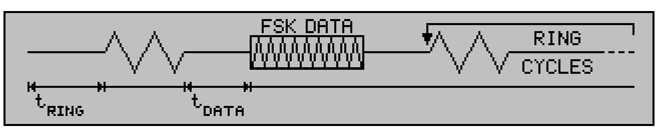
Parameter: Ring Generator Number of Cycles

Parameter Description:
This parameter specifies the number of complete ring cycles to generate for a Type I Caller ID Transmission.

Details:

Title:	Ring Generator Number of Cycles		
Script Name:	RING_CYCLES		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	cycles	cycles	cycles
Default Value:	**2**	**2**	**2**
Maximum Value:	100	100	100
Minimum Value:	1	1	1
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter Category: Type I CID Timing



Parameter: Type I CID Timing: Time to Start Ringing

Parameter Description:
The Time to Start Ringing parameter specifies the time interval that precedes the start of ringing. For the Bellcore standard, this is the time delay between starting a Caller ID transmission and the beginning of the first ringing pattern. For the ETSI and Australian standards, this is the time delay between the end of transmitting the FSK data and the beginning of the following ringing (if any). The exception to this is for the ETSI “Send Data after Ring” signaling type, in which the parameter has the same meaning as for the Bellcore case.

Details:

Title:	Time to Start Ringing		
Script Name:	TIME_RING		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**0**	**200**	**500**
Maximum Value:	20000	20000	20000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	1000
Minimum Standard Value:	n/a	200	200

Parameter: Type I CID Timing: Time to Data Transmission

Parameter Description:
This parameter specifies the time interval leading up to the start of the FSK data transmission. For the Bellcore standard, this is the delay between the end of the first ringing pattern to the start of the FSK data. For the ETSI and Australian standards this time interval is from the end of the selected alerting signal (DTAS, ring burst, OSI, or line reversal) to the beginning of the FSK data transmission. In the case no alerting signal is used, then this parameter defines the time between starting the Caller ID transmission to when the FSK data is sent.

Details:

Title:	Time to Data Transmission		
Script Name:	TIME_DATA_CID		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**500**	**250**	**700**
Maximum Value:	20000	20000	20000
Minimum Value:	0	0	0
Maximum Standard Value:	3950	n/a	1000
Minimum Standard Value:	250	45	500

Parameter: Type I CID Timing: Time to Line Reversal

Parameter Description:
This parameter is only used for the ETSI and Australian standards, and it specifies the time interval from the start of the Caller ID transmission to the beginning of the telephone line polarity reversal. For this parameter to have any effect, the signaling type must include a polarity reversal.

Details:

Title:	Time to Line Reversal		
Script Name:	TIME_REVERSE		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	msec
Default Value:	**n/a**	**0**	**0**
Maximum Value:	n/a	20000	20000
Minimum Value:	n/a	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type I CID Timing: Time to Ring Burst Alert Signal

Parameter Description:
This parameter is only used for the ETSI and Australian standards, and it specifies the time interval from the start of the Caller ID transmission to the beginning of a short ringing burst (used to alert the CPE to impending FSK data transmission). For this parameter to have any effect, the signaling type must include a ringing burst before sending the FSK data.

Details:

Title:	Time to Ring Burst Alert Signal		
Script Name:	TIME_RBAS		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	msec
Default Value:	**n/a**	**0**	**0**
Maximum Value:	n/a	20000	20000
Minimum Value:	n/a	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type I CID Timing: Time to Dual Tone Alert Signal

Parameter Description:
This parameter is only applicable to the ETSI standard. It specifies the time interval prior to the Dual Tone Alert Signal (DTAS). If the telephone line polarity reversal is enabled, then this time interval starts after the line reversal. Otherwise, the time interval starts when the Caller ID transmission is initiated.

Details:

Title:	Time to Dual Tone Alert Signal		
Script Name:	TIME_DTAS		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	n/a
Default Value:	**n/a**	**0**	**n/a**
Maximum Value:	n/a	20000	n/a
Minimum Value:	n/a	0	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type I CID Timing: Time to OSI

Parameter Description:
The parameter is only applicable to the Australian standard. It defines the time delay from the start of a Caller ID transmission to the beginning of an open switching interval (OSI). For this parameter to have any effect, the signaling type must be set to generate an OSI.

Details:

Title:	Time to OSI		
Script Name:	TIME_OSI		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	msec
Default Value:	**n/a**	**n/a**	**0**
Maximum Value:	n/a	n/a	20000
Minimum Value:	n/a	n/a	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

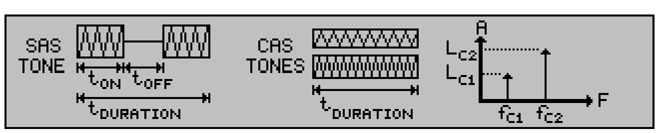
Parameter: Type I CID Timing: Time to Post Reversal

Parameter Description:
This parameter is only applicable to the Australian standard. In the case that no ringing is associated with the Caller ID transmission and a line polarity reversal is used as the signaling type; then this parameter defines the time interval from the end of the FSK data transmission to a second line polarity reversal. This second line reversal is used to restore the polarity of the telephone line to its previous state.

Details:

Title:	Time to Post Line Reversal		
Script Name:	TIME_POST_REV		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	msec
Default Value:	**n/a**	**n/a**	**500**
Maximum Value:	n/a	n/a	20000
Minimum Value:	n/a	n/a	0
Maximum Standard Value:	n/a	n/a	1000
Minimum Standard Value:	n/a	n/a	200

Parameter Category: SAS/CAS and Alert Tones



Parameter: SAS Tone Frequency

Parameter Description:
This parameter specifies the frequency of the Subscriber Alerting Signal (SAS) tone. This tone is generated before the CPE Alerting Signal (CAS) tone during Type II (CIDCW) Caller ID transmissions.

Details:

Title:	SAS Tone Frequency		
Script Name:	SAS_FREQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	n/a	Hz
Default Value:	**440**	**n/a**	**440**
Maximum Value:	10000	n/a	10000
Minimum Value:	50	n/a	50
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: SAS Tone Level

Parameter Description:
This parameter specifies the level of the Subscriber Alerting Signal (SAS) tone. This tone is generated before the CPE Alerting Signal (CAS) tone during Type II (CIDCW) Caller ID transmissions.

Details:

Title:	SAS Tone Level		
Script Name:	SAS_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	n/a	mVrms
Default Value:	**77.5**	**n/a**	**77.5**
Maximum Value:	2000	n/a	2000
Minimum Value:	0	n/a	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: SAS Tone On Time

Parameter Description:
This parameter specifies the On time of the Subscriber Alerting Signal (SAS) tone. The SAS On Time parameter is used in conjunction with the SAS Off Time and SAS Duration parameter to fully specify the SAS tone timing. The total SAS tone interval is determined by the Duration parameter. If the sum of the SAS tone On Time and Off Time parameters is less than the Duration, the On Time and Off Time cycle is repeated until the Duration time is reached. In this way, a pulsing SAS tone can be generated. For a continuous SAS tone, specify the On Time to be equal or greater than the Duration.

Details:

Title:	SAS Tone On Time		
Script Name:	SAS_ONTIME		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	n/a	msec
Default Value:	**300**	**n/a**	**300**
Maximum Value:	20000	n/a	20000
Minimum Value:	0	n/a	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: SAS Tone Off Time

Parameter Description:
This parameter specifies the Off time of the Subscriber Alerting Signal (SAS) tone. The SAS Off Time parameter is used in conjunction with the SAS On Time and SAS Duration parameter to fully specify the SAS tone timing. The total SAS tone interval is determined by the Duration parameter. If the sum of the SAS tone On Time and Off Time parameters is less than the Duration, the On Time and Off Time cycle is repeated until the Duration time is reached. In this way, a pulsing SAS tone can be generated. For a continuous SAS tone, specify the On Time to be equal or greater than the Duration.

Details:

Title:	SAS Tone Off Time		
Script Name:	SAS_OFFTIME		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	n/a	msec
Default Value:	**300**	**n/a**	**300**
Maximum Value:	20000	n/a	20000
Minimum Value:	0	n/a	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: SAS Tone Duration

Parameter Description:
This parameter specifies the total duration of the Subscriber Alerting Signal (SAS) tone. The SAS Duration parameter is used in conjunction with the SAS On Time and SAS Off Time parameter to fully specify the SAS tone timing. The total SAS tone interval is determined by the Duration parameter. If the sum of the SAS tone On Time and Off Time parameters is less than the Duration, the On Time and Off Time cycle is repeated until the Duration time is reached. In this way, a pulsing SAS tone can be generated. For a continuous SAS tone, specify the On Time to be equal or greater than the Duration.

Details:

Title:	SAS Tone Duration		
Script Name:	SAS_DURATION		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	n/a	msec
Default Value:	**300**	**n/a**	**300**
Maximum Value:	20000	n/a	20000
Minimum Value:	0	n/a	0
Maximum Standard Value:	350	n/a	1000
Minimum Standard Value:	250n/a	n/a	250

Parameter: CAS Tone #1 Frequency

Parameter Description:
This parameter specifies the frequency of one of the two tones used within the CPE Alerting Signal (CAS) tone. The CAS signal is composed of two simultaneous tones.

Details:

Title:	CAS Tone #1 Frequency		
Script Name:	CAS_FREQ1		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	n/a	Hz
Default Value:	**2130**	**n/a**	**2130**
Maximum Value:	10000	n/a	10000
Minimum Value:	50	n/a	50
Maximum Standard Value:	2141	n/a	2141
Minimum Standard Value:	2119	n/a	2119

Parameter: CAS Tone #2 Frequency

Parameter Description:
This parameter specifies the frequency of one of the two tones used within the CPE Alerting Signal (CAS) tone. The CAS signal is composed of two simultaneous tones.

Details:

Title:	CAS Tone #2 Frequency		
Script Name:	CAS_FREQ2		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	Hz	n/a	Hz
Default Value:	**2750**	**n/a**	**2750**
Maximum Value:	10000	n/a	10000
Minimum Value:	50	n/a	50
Maximum Standard Value:	2764	n/a	2764
Minimum Standard Value:	2736	n/a	2736

Parameter: CAS Tone #1 Level

Parameter Description:
This parameter specifies the level of one of the two tones used within the CPE Alerting Signal (CAS) tone. The CAS signal is composed of two simultaneous tones.

Details:

Title:	CAS Tone #1 Level		
Script Name:	CAS_LEVEL1		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	n/a	mVrms
Default Value:	**61.5**	**n/a**	**122.7**
Maximum Value:	1000	n/a	1000
Minimum Value:	0	n/a	0
Maximum Standard Value:	154.5	n/a	154.5
Minimum Standard Value:	19.5	n/a	19.5

Parameter: CAS Tone #2 Level

Parameter Description:
This parameter specifies the level of one of the two tones used within the CPE Alerting Signal (CAS) tone. The CAS signal is composed of two simultaneous tones.

Details:

Title:	CAS Tone #2 Level		
Script Name:	CAS_LEVEL2		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	n/a	mVrms
Default Value:	**61.5**	**n/a**	**122.7**
Maximum Value:	1000	n/a	1000
Minimum Value:	0	n/a	0
Maximum Standard Value:	154.5	n/a	154.5
Minimum Standard Value:	19.5	n/a	19.5

Parameter: CAS Tone Duration

Parameter Description:
This parameter specifies the duration of the CPE Alerting Signal (CAS) tone. The CAS tone is generated after the SAS tone and before the reception of the CPE's ACK tone.

Details:

Title:	CAS Tone Duration		
Script Name:	CAS_DURATION		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	n/a	msec
Default Value:	**80**	**n/a**	**80**
Maximum Value:	20000	n/a	20000
Minimum Value:	0	n/a	0
Maximum Standard Value:	85	n/a	85
Minimum Standard Value:	75	n/a	80

Parameter: Number of DTAS Tone Frequencies

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the number of tones used within the Dual Tone Alert Signal (DTAS). The options can either be one or two tones.

Details:

Title:	Number of DTAS Tones		
Script Name:	DTAS_TYPE		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**n/a**	**Two**	**n/a**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Dual Tone Alert #1 Frequency

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the frequency of DTAS tone number 1.

Details:

Title:	Dual Tone Alert #1 Frequency		
Script Name:	DTAS_FREQ1		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	Hz	n/a
Default Value:	**n/a**	**2130**	**n/a**
Maximum Value:	n/a	10000	n/a
Minimum Value:	n/a	50	n/a
Maximum Standard Value:	n/a	2153	n/a
Minimum Standard Value:	n/a	2107	n/a

Parameter: Dual Tone Alert #2 Frequency

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the frequency of DTAS tone number 2. If only one tone is specified in the Number of DTAS Tones parameter, then this parameter value has no effect on the Caller ID transmission.

Details:

Title:	Dual Tone Alert #2 Frequency		
Script Name:	DTAS_FREQ2		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	Hz	n/a
Default Value:	**n/a**	**2750**	**n/a**
Maximum Value:	n/a	10000	n/a
Minimum Value:	n/a	50	n/a
Maximum Standard Value:	n/a	2780	n/a
Minimum Standard Value:	n/a	2720	n/a

Parameter: Dual Tone Alert #1 Level

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the signal level for DTAS tone number 1.

Details:

Title:	Dual Tone Alert #1 Level		
Script Name:	DTAS_LEVEL1		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	mVrms	n/a
Default Value:	**n/a**	**316.2**	**n/a**
Maximum Value:	n/a	1000	n/a
Minimum Value:	n/a	0	n/a
Maximum Standard Value:	n/a	794	n/a
Minimum Standard Value:	n/a	10	n/a

Parameter: Dual Tone Alert #2 Level

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the signal level for DTAS tone number 2. If only one tone is specified in the Number of DTAS Tones parameter, then this parameter value has no effect on the Caller ID transmission.

Details:

Title:	Dual Tone Alert #2 Level		
Script Name:	DTAS_LEVEL2		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	mVrms	n/a
Default Value:	**n/a**	**316.2**	**n/a**
Maximum Value:	n/a	1000	n/a
Minimum Value:	n/a	0	n/a
Maximum Standard Value:	n/a	794	n/a
Minimum Standard Value:	n/a	10	n/a

Parameter: Dual Tone Alert Duration for Type I

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the duration of the DTAS tone generated when used for Type I Caller ID transmissions.

Details:

Title:	Dual Tone Duration for Type I		
Script Name:	DTAS_DUR_CID		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	n/a
Default Value:	**n/a**	**100**	**n/a**
Maximum Value:	n/a	20000	n/a
Minimum Value:	n/a	0	n/a
Maximum Standard Value:	n/a	110	n/a
Minimum Standard Value:	n/a	90	n/a

Parameter: Dual Tone Alert Duration for Type II

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the duration of the DTAS tone generated when used for Type II Caller ID transmissions.

Details:

Title:	Dual Tone Duration for Type II		
Script Name:	DTAS_DUR_CIDCW		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	n/a
Default Value:	**n/a**	**80**	**n/a**
Maximum Value:	n/a	20000	n/a
Minimum Value:	n/a	0	n/a
Maximum Standard Value:	n/a	85	n/a
Minimum Standard Value:	n/a	75	n/a

Parameter: Ring Burst Frequency

Parameter Description:
This parameter is only applicable to the ETSI and Australian standards, and defines the frequency of the Ring Burst Alerting Signal (RBAS).

Details:

Title:	Ring Burst Frequency		
Script Name:	RBAS_FREQ		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	Hz	Hz
Default Value:	**n/a**	**22**	**22**
Maximum Value:	n/a	1000	1000
Minimum Value:	n/a	10	10
Maximum Standard Value:	n/a	68	68
Minimum Standard Value:	n/a	13	13

Parameter: Ring Burst Level

Parameter Description:
This parameter is only applicable to the ETSI and Australian standards, and defines the level of the Ring Burst Alerting Signal (RBAS). The maximum ringing level that can be generated is 80 Vrms; however, this is dependent on the setting of the Telephone Line Voltage. Below a setting of 48 Volts for the Telephone Line Voltage, the maximum unclipped ring generator level must be de-rated. At the minimum Telephone Line Voltage level of 20 V, the maximum unclipped ringing voltage is 60 Vrms.

Details:

Title:	Ring Burst Level		
Script Name:	RBAS_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	Vrms	Vrms
Default Value:	**n/a**	**50**	**50**
Maximum Value:	n/a	80	80
Minimum Value:	n/a	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	40	40

Parameter: Ring Burst Duration

Parameter Description:
This parameter is only applicable to the ETSI and Australian standards, and defines the duration of the Ring Burst Alerting Signal (RBAS).

Details:

Title:	Ring Burst Duration		
Script Name:	RBAS_DURATION		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	msec
Default Value:	**n/a**	**250**	**500**
Maximum Value:	n/a	20000	20000
Minimum Value:	n/a	0	0
Maximum Standard Value:	n/a	300	650
Minimum Standard Value:	n/a	200	200

Parameter: OSI Duration

Parameter Description:
This parameter is only applicable to the Australian standard, and defines the duration of the open switching interval (OSI).

Details:

Title:	OSI Duration		
Script Name:	OSI_DURATION		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	msec
Default Value:	**n/a**	**n/a**	**250**
Maximum Value:	n/a	n/a	20000
Minimum Value:	n/a	n/a	0
Maximum Standard Value:	n/a	n/a	150
Minimum Standard Value:	n/a	n/a	350

Parameter: Alert Tone Noise Generator Enable

Parameter Description:
This parameter enables or disables the noise generator during the transmission of the alerting SAS/CAS and DTAS tones. For the Bellcore and Australian standards, the script language parameter name is CAS_NOISE_ENABLE; while for the ETSI standard, the parameter name is DTAS_NOISE_ENABLE. The noise generated during the FSK data transmission is not determined by this parameter, but rather the FSK Signal-to-Noise ratio parameter in the Main Settings window.

Details:

Title:	Noise Generator Enable		
Script Name:	CAS_NOISE_ENABLE / DTAS_NOISE_ENABLE		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**Disabled**	**Disabled**	**Disabled**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

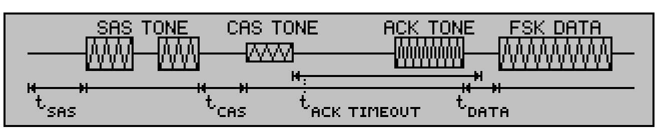
Parameter: Alert Tone Noise Generator Level

Parameter Description:
This parameter specifies the level of the noise generator during the transmission of the alerting SAS/CAS and DTAS tones. For the Bellcore and Australian standards, the script language parameter name is CAS_NOISE_LEVEL; while for the ETSI standard, the parameter name is DTAS_NOISE_LEVEL. No noise will be generated during the alert tones unless enabled with the Noise Generator Enable parameter. The noise generated during the FSK data transmission is not determined by this parameter, but rather the FSK Signal-to-Noise ratio parameter in the Main Settings window.

Details:

Title:	Noise Generator Level		
Script Name:	CAS_NOISE_LEVEL / DTAS_NOISE_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	mVrms	mVrms
Default Value:	**0**	**0**	**0**
Maximum Value:	1000	1000	1000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter Category: Type II CIDCW Timing



Parameter: Type II CIDCW Timing: Time to SAS Tone

Parameter Description:
This parameter is only applicable to the Bellcore and Australian standards, and specifies the time from when the Caller ID transmission is started to the time the SAS tone generation starts.

Details:

Title:	Time to SAS Tone		
Script Name:	TIME_SAS		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	n/a	msec
Default Value:	**0**	**n/a**	**0**
Maximum Value:	20000	n/a	20000
Minimum Value:	0	n/a	0
Maximum Standard Value:	n/a	n/a	60
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type II CIDCW Timing: Time to CAS Tone

Parameter Description:
This parameter is only applicable to the Bellcore and Australian standards, and specifies the time from the end of the SAS tone to the beginning of the CAS tone generation.

Details:

Title:	Time to CAS Tone		
Script Name:	TIME_CAS		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	n/a	msec
Default Value:	**25**	**n/a**	**25**
Maximum Value:	20000	n/a	20000
Minimum Value:	0	n/a	0
Maximum Standard Value:	80	n/a	50
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type II CIDCW Timing: Time to DTAS Tone

Parameter Description:
This parameter is only applicable to the ETSI standard, and specifies the time from the start of the Type II Caller ID transmission to the beginning of the DTAS tone.

Details:

Title:	Time to DTAS Tone		
Script Name:	TIME_DTAS_CIDCW		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	msec	n/a
Default Value:	**n/a**	**0**	**n/a**
Maximum Value:	n/a	20000	n/a
Minimum Value:	n/a	0	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type II CIDCW Timing: Time-out for ACK Tone

Parameter Description:
This parameter specifies the maximum time duration for the reception of the ACK tone generated by the CPE. The CPE upon detecting the CAS/DTAS tone should respond with the ACK tone. The time-out period starts at the end of the CAS/DTAS tone generation.

Details:

Title:	Time-out for ACK Tone		
Script Name:	TIMEOUT_ACK		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**160**	**160**	**160**
Maximum Value:	0	0	0
Minimum Value:	20000	20000	20000
Maximum Standard Value:	165	165	160
Minimum Standard Value:	155	155	n/a

Parameter: Type II CIDCW Timing: Time to Data Transmission

Parameter Description:
This parameter specifies the time interval between the reception of the ACK tone and the start of the FSK data transmission. If no ACK tone is received and the Transmit Data even after ACK Time-out Parameter is set to enabled, then this parameter defines the time between the expiration of the time-out interval and the start of the FSK data transmission.

Details:

Title:	Time to Data Transmission		
Script Name:	TIME_DATA_CIDCW		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	msec	msec	msec
Default Value:	**250**	**250**	**100**
Maximum Value:	0	0	0
Minimum Value:	20000	20000	20000
Maximum Standard Value:	500	500	100
Minimum Standard Value:	50	50	50

Parameter: Type II CIDCW Timing: Transmit Data even after ACK Time-out

Parameter Description:
This parameter enables or disables a feature where the FSK data transmission may be started even if no ACK tone has been received from the CPE. If enabled, the FSK data transmission will start after the time-out period has expired and the delay specified in the Time to Data Transmission parameter. If disabled, no FSK data will be sent unless an ACK tone is received within the time-out period.

Details:

Title:	Transmit Data even after time-out		
Script Name:	TIMEOUT_TX		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**Disable**	**Disable**	**Disable**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type II CIDCW Timing: ACK Low Group Tone Minimum Level

Parameter Description:
This parameter specifies the minimum acceptable level of the low group CPE ACK tone. Unless the returned ACK tone has its low frequency group level greater than the specified value, the ACK tone will be ignored.

Details:

Title:	ACK Low Group Tone Minimum Level		
Script Name:	ACK_LOWG_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	mVrms	mVrms
Default Value:	**100**	**100**	**100**
Maximum Value:	2000	2000	2000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type II CIDCW Timing: ACK High Group Tone Minimum Level

Parameter Description:
This parameter specifies the minimum acceptable level of the high group CPE ACK tone. Unless the returned ACK tone has its high frequency group level greater than the specified value, the ACK tone will be ignored.

Details:

Title:	ACK High Group Tone Minimum Level		
Script Name:	ACK_HIGHG_LEVEL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	mVrms	mVrms	mVrms
Default Value:	**100**	**100**	**100**
Maximum Value:	2000	2000	2000
Minimum Value:	0	0	0
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

Parameter: Type II CIDCW Timing: ACK Frequency Tolerance

Parameter Description:
This parameter defines the upper bound for the ACK tone frequency tolerance. Both the low group tone and high group tone must have their frequency within the specified tolerance, or the ACK tone will be ignored.

Details:

Title:	ACK Frequency Tolerance		
Script Name:	ACK_FREQ_TOL		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	%	%	%
Default Value:	**2**	**2**	**2**
Maximum Value:	100	100	100
Minimum Value:	0	0	0
Maximum Standard Value:	1.5	1.5	1.5
Minimum Standard Value:	3.5	3.5	3.5

Parameter: Type II CIDCW Timing: Valid ACK Digit

Parameter Description:
This parameter defines what DTMF digit, or range of digits will be accepted as a valid ACK tone from the CPE. The possible selctions for this parameter can be: Digit A, B, C, D, A and D, A to D, or any DTMF digit.

Details:

Title:	Valid ACK Digit		
Script Name:	ACK_DIGIT		
Standard:	***Bellcore***	***ETSI***	***Australia***
Units:	n/a	n/a	n/a
Default Value:	**A and D**	**A and D**	**A and D**
Maximum Value:	n/a	n/a	n/a
Minimum Value:	n/a	n/a	n/a
Maximum Standard Value:	n/a	n/a	n/a
Minimum Standard Value:	n/a	n/a	n/a

■ Appendix B Script Command Summary

Commands

START	
PAUSE	(optional label) ([EXT] optional)
END	
LOOP	(number of loops)
LOOPEND	
LABEL	(label name)
BRANCHIF	(condition) (label)
CALL	(label)
RETURN	
DECLARE	(string/numeric) (variable)
ASSIGN	(variable) TO (expression)
INPUT	(style) (variable) "caption" (min) (max)
PARAMETER	(parameter name) (operator) (value)
MODE	(auto/typeI/typeII)
SIGNALING	(signaling type)
MESSAGE	(message type)
PACKET	(packet type) (enabled/disabled/value) (value)
SEGMENT	(segment type) (numberbits/time) (value)
STOPBITS	(number of bits)
MARKBITS	(number of bits)
SET	DropOut (segment name) (start time) (duration)
SET	FSK (on/off)
SET	MSLEVEL (mark level) (space level)
SET	ECHOTAP (1/2/3) (delay value) (attn value)
SET	ECHOTAP (1/2/3) OFF
SET	LEVELUNITS (dBm/dBV/mVrms) (term/unterm)
SET	OUTPUTA (on/off/toggle/FSK*/FSKdelay)
SET	OUTPUTB (on/off/toggle/Ring*/OSI*)
SET	AUTOXSUM (on/off)
SET	ROUTING (control) (option)
SET	PARITY (8none/7odd/7even)
CLEAR	(ACKcounter/all/log/InBcount/Timer)
LOG	SAVEFILE (filename)
LOG	ACKdisplay (option)
LOG	HEADER (on/off)
LOG	PRINTING (Title/Footer) ([NONE]/[DEFAULT]/"text")
PRINTSCREEN	"text string" [(variable)] ...
PRINTLOG	"text string" [(variable)] ...
CHANGEBYTE	(packet type) **#** (address) TO (invert/inc/dec/value)
CHANGEBIT	(segment) **#** (bit position) TO (0/1/toggle)
CHANGEBIT	(segment) FromEnd (bit position) TO (0/1/toggle)
TONEGEN	(parameter name) (operator) (value)
FSKGEN	MODE (single/continuous)
FSKGEN	TYPE (mark/space/alternate/user/external)
FSKGEN	DATA "text string"
FSKGEN	BITS (number of bits)
FSKGEN	GO
FSKGEN	STOP
WAIT	(time delay in seconds)
NETTONE	(filename)
DO	OSI (duration)
DO	TipRing (connect/disconnect)

Note: (*) Available with the TSPC only (not supported with AI-7280).

Parameters

Parameters common to all standards:

Script Name	***Range of Values (levels are in mVrms)***
LINE_VOLTAGE	20 to 52 Volts
LINE_CURRENT	20 to 40 mA
LINE_POLARITY	normal, reversed
LINE_IMP	600_ohms, 900_ohms, complex
FSK_LEVEL	0 to 1000 mVrms
FSK_SNR	0 to 40 dB
FSK_MFREQ	100 to 5000 Hz
FSK_SFREQ	100 to 5000 Hz
FSK_TWIST	-60 to 60 dB
FSK_BAUD	100 to 5000 bps
FSK_BIT_SKEW	20 to 75 %
FSK_IENABLE	enabled, disabled
FSK_IFREQ	50 to 10000 Hz
FSK_ILEVEL	0 to 2000 mVrms
RING_FREQ	10 to 1000 Hz
RING_LEVEL	0 to 80 Vrms
RING_SEQ	2_State, 4_State, 6_State
RING_ON1	0 to 100000 msec
RING_ON1	0 to 100000 msec
RING_ON2	0 to 100000 msec
RING_OFF2	0 to 100000 msec
RING_ON3	0 to 100000 msec
RING_OFF3	0 to 100000 msec
RING_CYCLES	1 to 100

Parameters defining Bellcore Caller ID characteristics

Script Name	***Range of Values (levels are in mVrms)***
TIME_RING	0 to 20000 msec
TIME_DATA_CID	0 to 20000 msec
SAS_FREQ	50 to 10000 Hz
SAS_LEVEL	0 to 2000 mVrms
SAS_ONTIME	0 to 20000 msec
SAS_OFFTIME	0 to 20000 msec
SAS_DURATION	0 to 20000 msec
CAS_FREQ1	50 to 10000 Hz
CAS_FREQ2	50 to 10000 Hz
CAS_LEVEL1	0 to 1000 mVrms
CAS_LEVEL2	0 to 1000 mVrms
CAS_DURATION	0 to 20000 msec
CAS_NOISE_ENABLE	enabled, disable
CAS_NOISE_LEVEL	0 to 1000 mVrms
TIME_SAS	0 to 20000 msec
TIME_CAS	0 to 20000 msec
TIMEOUT_ACK	0 to 20000 msec
TIME_DATA_CIDCW	0 to 20000 msec
TIMEOUT_TX	enabled, disabled
ACK_LOWG_LEVEL	0 to 2000 mVrms
ACK_HIGHG_LEVEL	0 to 2000 mVrms
ACK_FREQ_TOL	0 to 100 %
ACK_DIGIT	A, B, C, D, A and D, A to D, Any DTMF

Parameters defining ETSI Caller ID characteristics

Script Name	***Range of Values (levels are in mVrms)***
TIME_REVERSE	0 to 20000 msec
TIME_DTAS	0 to 20000 msec
TIME_RBAS	0 to 20000 msec
TIME_DATA_CID	0 to 20000 msec
TIME_RING	0 to 20000 msec
DTAS_TYPE	one, two
DTAS _FREQ1	50 to 10000 Hz
DTAS _FREQ2	50 to 10000 Hz
DTAS _LEVEL1	0 to 1000 mVrms
DTAS _LEVEL2	0 to 1000 mVrms
DTAS_DUR_CID	0 to 20000 msec
DTAS_DUR_CIDCW	0 to 20000 msec
DTAS_NOISE_ENABLE	enabled, disable
DTAS_NOISE_LEVEL	0 to 1000 mVrms
RBAS_FREQ	10 to 1000 Hz
RBAS_LEVEL	0 to 80 Vrms
RBAS_DURATION	0 to 20000 msec
TIME_DTAS_CIDCW	0 to 20000 msec
TIMEOUT_ACK	0 to 20000 msec
TIME_DATA_CIDCW	0 to 20000 msec
TIMEOUT_TX	enabled, disabled
ACK_LOWG_LEVEL	0 to 2000 mVrms
ACK_HIGHG_LEVEL	0 to 2000 mVrms
ACK_FREQ_TOL	0 to 100 %
ACK_DIGIT	A, B, C, D, A and D, A to D, Any DTMF

Parameters defining Australian Caller ID characteristics

Script Name	***Range of Values (levels are in mVrms)***
TIME_RBAS	0 to 20000 msec
TIME_REVERSE	0 to 20000 msec
TIME_OSI	0 to 20000 msec
TIME_DATA_CID	0 to 20000 msec
TIME_RING	0 to 20000 msec
TIME_POST_REV	0 to 20000 msec
SAS_FREQ	50 to 10000 Hz
SAS_LEVEL	0 to 2000 mVrms
SAS_ONTIME	0 to 20000 msec
SAS_OFFTIME	0 to 20000 msec
SAS_DURATION	0 to 20000 msec
CAS_FREQ1	50 to 10000 Hz
CAS_FREQ2	50 to 10000 Hz
CAS_LEVEL1	0 to 1000 mVrms
CAS_LEVEL2	0 to 1000 mVrms
CAS_DURATION	0 to 20000 msec
CAS_NOISE_ENABLE	enabled, disable
CAS_NOISE_LEVEL	0 to 1000 mVrms
RBAS_FREQ	10 to 1000 Hz
RBAS_LEVEL	0 to 80 Vrms
RBAS_DURATION	0 to 20000 msec
OSI_DURATION	0 to 20000 msec
TIME_SAS	0 to 20000 msec
TIME_CAS	0 to 20000 msec
TIMEOUT_ACK	0 to 20000 msec
TIME_DATA_CIDCW	0 to 20000 msec
TIMEOUT_TX	enabled, disabled
ACK_LOWG_LEVEL	0 to 2000 mVrms
ACK_HIGHG_LEVEL	0 to 2000 mVrms
ACK_FREQ_TOL	0 to 100 %
ACK_DIGIT	A, B, C, D, A and D, A to D, Any DTMF

■ Appendix C Script Language Errors

The following is a list of script program errors:

The script program contains an unknown command.
The first word in a script program line, that is not a comment, must be a command key word. The command must match the spelling as given in the script command reference. Note, the command words are **not** case sensitive See the section: Script Command Reference

The program line's syntax is inconsistent with the command.
The command has been recognized, but the line's syntax in incorrect. This is usually due to an incorrect number of operands on the program line. Remember that spaces are required between operands.

The PARAMETER specified is unknown.
The parameter name that must be specified after the PARAMETER command is unknown. Check the spelling of the parameter name. Note, the parameter names are **not** case sensitive.

The operator being used is invalid.
For the parameter command, the operator specified is not one of the four accepted types. The valid operators are "=" "+=" "-=" "*=" with at least one space before and after the operator.

The value specified for the PARAMETER is unknown or invalid.
The value given for the parameter may not be spelled correctly if it is a binary type parameter.

The program line contains a text string without matching quotation marks.
There must be two quotation marks to correctly specify a text string.

The loop count value must be an integer value greater than zero.
The numeric value given after the LOOP command must be an integer value and be greater than zero.

Too many loops are being used in the script program.
A maximum limit of 50 loops may be used in any one script program. If this limit has been exceeded, some of the LOOP and LOOPEND commands must be removed.

A LOOP command has been found without a matching LOOPEND command.
For every LOOP command, a matching LOOPEND command must be present. This error indicates that there are more LOOP commands than LOOPEND commands.

A LOOPEND command has been found without a matching LOOP command.
For every LOOPEND command, a matching LOOP command must be present. This error indicates that there are more LOOPEND commands than LOOP commands.

The MODE type indicated is unknown.
The mode type given is an unknown or an invalid type. Check the spelling for the correct mode. The valid mode types are: TypeI, TypeII, and Auto. Note that the mode type is **not** case sensitive.

The number of STOP BITS must be between 1 and 60.
The quantity specified after the STOPBITS command is not between 1 and 60.

The number of MARK BITS must be between 0 and 150.
The quantity specified after the MARKBITS command is not between 0 and 150.

The MESSAGE type specified is unknown.
The message type given is unknown or an invalid type. Check the spelling for the correct message . For the Bellcore standard, the valid message types are: Single_Message, Multiple_Message, Single_Msg_Waiting, and Multiple_Msg_Waiting. For the ETSI

standard, the valid message types are Call_Setup, and Message_Waiting. Note that the message type is **not** case sensitive.

The PACKET name specified is unknown.
The packet name given is unknown or an invalid type. Check the spelling for the correct packet name . See the Script Command Reference on the PACKET command for the valid packet names. Note that the packet name is **not** case sensitive.

See: PACKET Command Reference

The operator specified for the packet is invalid.
The key word given after the packet name must be either "enabled", "disabled", or "value". Check the spelling in the program line.

The segment name specified is unknown or invalid.
The segment name given is unknown or an invalid type. Check the spelling for the correct segment name . See the Script Command Reference on the SEGMENT command for the valid segment names. Note that the segment name is **not** case sensitive.

See: SEGMENT Command Reference

The operator specified for the segment is invalid.
The key word given after the segment name must be either "NumberBits", or "Time". Check the spelling in the program line.

The value specified can not be negative.
The numeric value specified in the program line is negative. This is an illegal value.

The LABEL command requires a single label to follow it, without any spaces in the label.
The LABEL command has more than one label specified after the LABEL command word, or it has no label specified.

The maximum number of labels has been exceeded in this program.
Only a maximum of 99 labels can be used in a script program. The number of labels must be reduced before the program can be executed.

Can't find the label specified in the PAUSE, BRANCHIF, or CALL command.
The label given in the PAUSE does not match any of the labels specified with the LABEL command. Check the spelling of the offending label.

The address specified must be between 1 and 256 and an integer value.
The address specified in the CHANGEBYTE command must meet the conditions of being between 1 and 256, and an integer value.

The data value specified for a byte must be either INVERT, INC, DEC, or between 0 and 255 and an integer value.
When specifying the new value of the byte with the CHANGEBYTE command, it must be either an integer between 0 and 255, or one of the three modifying key words. The valid modifiers are INVERT, INC, or DEC.

The signaling mode specified is unknown or invalid for the current operational standard.
The signaling type specified is either unknown, spelt incorrectly, or not valid for the current operational standard. The signaling type for the Bellcore standard can not be used for the ETSI standard or for the Australian standard, and vise-versa. Note that the signaling type specified is **not** case sensitive.

The specified value can not be accepted as it is not a numeric value.
The value being specified is not a numeric value, which the selected command requires.

The value specified does not match the requirements for this command.
The value given for the command does is not one of the valid keywords associated with that command. This is most likely due to an incorrect spelling.

The file specified with the NETTONE command does not exist or has an invalid path or the device is not available.
The NETTONE script command is unable to load the network script tone file specified. This can occur if the file does not exist, the path is incorrect, or the source drive is not ready or unavailable.

A syntax error exists within the specified NetTone script file.

The NETTONE script command has detected a syntax error within the specified network tone script file. This prevents execution of the network tone script file. Use the NetTone program to view, edit, and debug the file in error.

Too many CALL's without RETURN's. The return address stack space is full.
Subroutines may be nested up to 32 levels deep. On the 33rd CALL command, this error will be issued because of a limited amount of data space to store the return address locations. To remove this error, reduce the subroutine depth to below 32 levels.

A RETURN command was encountered before a CALL command.
A RETURN command must always occur after a CALL command. If the RETURN command is encountered first, this error is displayed.

The variable name in the DECLARE statement is invalid. It must be no more than 32 characters, with only letters and numbers.
The variable name given with the DECLARE command can not be used, as it must be less than 32 characters in length and use only letters and numbers (with the exception of the underscore character). The variable name must also start with a letter.

The variable name in the DECLARE statement has been previously declared elsewhere.
An attempt was made to DECLARE a script variable of the same name of a previously declared variable. Script variable names must be unique.

The variable name in the DECLARE statement is a reserved variable.
The variable name given with the DECLARE command is the same as a reserved variable name. A different name for the variable must be used.

This command or function requires a text string, not a numeric value.
The script command in the line causing the error expects to see a text string as its operand. Not a numeric value. This can be caused by specifying a script variable that is a numeric type and not a string type. However, a numeric variable can still be used by enclosing it in [ ] brackets. This performs a conversion to its equivalent text string.

The literal or variable text string specified is too long for this command.
The length of the text string used with this command is too long. Use a shorter text string.

The literal or variable text string specified is too short for this command.
The length of the text string used with this command is too short. Use a longer text string.

The literal or variable specified must be a numeric value, not a text string.
The script command in the line causing the error expects to see a numeric value instead of the text string specified. If using script variables, ensure that the variable used is a numeric type and not a string type.

The literal or variable specified exceeds the maximum permissible limit for this command.
The numeric value specified for this command is too large. Use a smaller value.

The literal or variable specified is less than the minimum permissible limit for this command.
The numeric value specified for this command is too small. Use a larger value.

The literal or variable specified must be an integer value for this command.
The numeric value specified for this command must be an integer value.

The reserved variable specified is only defined as a text string value. Enclose the variable in [ ] characters.
Many of the reserved variables can only be accessed as text strings. As such, they should always be specified surrounded by [ ] brackets. This ensures that the text equivalent of variable is used.

The target user variable that will contain the result of the expression does not exist.
The variable name specified with the ASSIGN or INPUT command has not been declared. Check that the variable name used matches the one in a DECLARE statement.

The syntax for the expression is invalid..
The syntax for the expression in the ASSIGN command is invalid. Expression are made up of operands (variables or literals) and operators. An operator must always be surrounded by two operands. For more information on the syntax of expression, consult the section explaining the ASSIGN command.

The operator specified is unknown. The valid operators are: +, -, /, *, >, >=, <, =<, =, <>, AND, OR.

The operator used within the expression of the ASSIGN command is unknown. Insure that the operator used is found in the list given above.

Two of the operands are of different types. All operands in an expression must be either all numeric in nature or text strings.

The operands in an expression can not be mixed in type. They must be either all numeric or text strings. Note that to automatically convert a numeric variable to a text string, enclose the variable name in [ ] brackets.

The specified operator is not defined for text string operands. For strings, the operator must be either: +, =, <>.

An expression with text strings is limited to one the three operators shown above.

There are too many elements on a single command line.

The maximum number of items on a single command line has been exceeded. This error can occur with variable length script commands such as ASSIGN. Reduce the size of the expression to eliminate this error.

Too many nested IF commands have been used. Reduce the number of nested IF statements.

Script programs can only support up to 32 nested IF statements. This error will occur upon exceeding that limit. In addition, if the BRANCHIF statement is used to jump outside of an IF statement, the ENDIF statement will never execute, causing this error to occur on the 32nd instance.

The ENDIF command does not have a matching IF command. The IF statement must always come before the ENDIF statement.

An ENDIF command must always follow an IF command, within the same module. If the ENDIF statement is missing, this error is generated.

The ELSE command does not have a matching IF command. The IF statement must always come before the ELSE statement.

If the ELSE statement is used, it must always come after the IF statement and before the ENDIF statement.

More than one ELSE command follows the IF command. This is an illegal condition, as only one ELSE may follow the IF statement.

Only once ELSE statement may follow an IF statement.

No ENDIF command found to match an IF command. Every IF statement must be followed by an ENDIF statement.

This error is generated when no ENDIF statement is found of occur after an IF statement.

Nothing was read from the file. Either the file name, section header, or keyword was invalid.

When using the READFILE command, if the file specified does not exist, or the section header does not exist, or the keyword does not exist, this error is generated.

The impairment type or parameter has not been correctly specified, or is unknown. Check the command syntax for valid impairment types and parameters.

This error is generated when using the DCIMP command with an impairment type or parameter type that is unknown or unsupported.

The number of parameters exceeds the number of markers found in the specified NetTone file.

If using the NETTONE command to start a NetTone command file with parameter passing, but the specified file does contain any or enough parameter markers, this error is generated. Check to make sure the number of parameters passed matches the number of parameter markers in the NetTone command file.

Warnings that may be issued during the execution of a script file are listed here.

The new parameter value exceeds its maximum allowable value.

The parameter value, changed with the PARAMETER command, is greater than its maximum allowed value. As such, it will be clamped to its maximum value.

The new parameter value is below its minimum allowable value.

The parameter value, changed with the PARAMETER command, is less than its minimum allowed value. As such, it will be clamped to its minimum value.

The byte address specified with the CHANGEBYTE command is greater than the number of bytes in the packet.

The CHANGEBYTE command is being used to change a byte of a packet that does not exist. The address specified in the command is greater than the total number of bytes in the packet. For example, this warning will be generated if the sixth byte in a five byte packet is being changed.

The packet specified with the CHANGEBYTE Command is not currently enabled.

The CHANGEBYTE command is attempting to change the byte of a packet that is not enabled, and is currently not part of the Caller ID Message. For example, this warning will be generated if the CHANGEBYTE command is being used to change a byte in the NAME packet, but the NAME packet is not currently enabled.

The byte altered with the CHANGEBYTE command is the Checksum byte, but the Auto Checksum Calculation feature has been enabled, which overrides the change.

The CHANGEBYTE command can not be used to change the Checksum byte of a Caller ID Message, if the Auto Checksum Calculation feature is enabled. The Auto Checksum Calculation feature will always re-calculate the message checksum after any changes to the message. However, this prevents the CHANGEBYTE command from then altering the value of the checksum byte. The Auto Checksum Calculation feature and be enabled or disabled in the Packet Format window.

The bit number specified is outside the valid range for the given segment.

This warning is generated when using the CHANGEBIT command with a bit position that does not exist. For example, if the mark segment only has 180 bits, then attempting to change the 181st bit will cause this warning.

■ Appendix D Glossary of Terms

ACK Tone
Acknowledge Tone - The term given to the signal generated by the CPE after it has detected a CAS tone. The Acknowledge tone is equivalent to the DTMF 'D' tone (low group frequency of 941 Hz and high group tone of 1633 Hz). Some CPE's may use an ACK tone equivalent to DTMF 'A' instead of 'D'. In this case, the low group frequency is 697 Hz, and the high group tone remains the same at 1633 Hz.

Alert Tone
A signal consisting of one or two tones that are used to inform the CPE that a Caller ID transmission of data will commence shortly. The alert tone applies only to Caller ID transmission using the ETSI standard.

Auto Detect
A mode of operation where the CID1500 program switches between Type I and Type II Caller ID transmission modes automatically, depending on the state of the CPE's hook switch.

CAS Tone
CPE Alerting Signal - A signal consisting of two tones that are sent to the CPE to alert it of an impending Caller ID message. The CAS tone is part of the Type II Caller ID transmission sequence.

Channel Seizure
A pattern of alternating mark/space bits that are sent to the CPE prior to the Mark Segment and Data Message segment. This bit pattern is represented by the Preamble Segment in the CID1500 program.

Checksum Byte
A byte that is used to detect the presence of transmission errors in other data bytes. It is the last byte of a Caller ID Message and it represents the two's complement of the sum of all the other data bytes in the message.

CID
Caller ID - Generally refers to a Type I Caller ID transmission, where the data is sent with the CPE being in the on-hook state.

CIDCW
Caller ID with Call Waiting - Generally refers to a Type II Caller ID transmission, where the data is sent with the CPE being in the off-hook state.

CLIP
Calling Line Identification Presentation - A data message type for use with the British Telecom SIN242/227 standard. This message type is very similar to the Bellcore Multiple Data Message Format and the ETSI Call Setup mesage.

CPE
Customer Premises Equipment - A device connected to the telephone network that resides at the customers location.

DDN
Dialable Directory Number - A packet type, in which the number contained within it can be dialed out with no modification to its format.

FSK
Frequency Shift Keying - The method of signal modulation used for data transmission. In the specific case of Caller ID data transmission, a signal carrier's frequency is shifted between two values representing either a mark or space bit.

Mark Bit
A single bit that represents a logical one. Its complement is the Space bit, which represents a logical zero.

Mark Segment
A group of Mark bits that are sent to the CPE after the Preamble or Channel Seizure segment, and before the Data Message segment.

Mark Stuffing Bits
A small group of Mark bits that may be sent after the Stop bit of certain bytes within the Data Message segment. These bytes are the Message Type byte, Message Length byte, Packet Type bytes, Packet

	Length bytes, and the last byte in the packets.
Message Segment	The bit pattern containing all of the Message bytes to be transmitted to the CPE. Each byte in the message is sent first with a Start bit, then the eight data bits of the byte (LSB first), and then a Stop bit. Certain data bytes may contain Mark Stuffing Bits inserted after the Stop Bit.
Preamble Segment	A pattern of alternating mark/space bits that are sent to the CPE prior to the Mark Segment and Data Message segment.
SAS Tone	Subscriber Alerting Signal - A tone sent to the CPE in the off-hook state to alert the user that another caller is trying to call the user. This tone precedes the CAS tone in a Type II Caller ID transmission.
SNR	Signal-to-Noise Ratio - A numerical value that represents the ratio of the power of a specific signal, and the power of the noise present at the measuring point. Usually specified in logarithmic values (dB).
Space Bit	A single bit that represents a logical zero. Its complement is the Mark bit, which represents a logical one.
SPCS	Stored Program Control System - Generally referred to a telephone switch used in a central office which links the CPE to the telephone network.
Start Bit	A logical zero bit that precedes the LSB of a data byte. Used to represent the start of a Data Message byte.
Stop Bit	A logical one bit that follows the MSB of a data byte. Used to represent the end of a Data Message byte.
Twist Level	The ratio of signal powers between two different tones. A positive twist is defined as the lower frequency tone having a power level greater than the higher frequency tone.
Type I	Caller ID transmission occurring while the CPE is in the on-hook state.
Type II	Caller ID transmission occurring while the CPE is in the off-hook state. Also referred to CIDCW.

■ Appendix E Modifying the Complex Impedance

The telephone line impedance that is presented to the CPE can be set to ,under program control, to one of three possible values. These are:

i) 600 ohms (real)
ii) 900 ohms (real)
iii) Complex impedance ZR

Where the complex impedance ZR is modeled by the following schematic diagram:

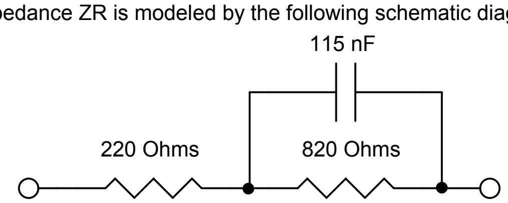


Figure 2. ZR Impedance

However, it is possible to change various components on the TSPC in order to present other custom or user defined impedance's. Figure 3 represents a network of components present on the TSPC that determine the value of the complex line impedance. The line impedance presented to the CPE will be 0.1 times the impedance of the network shown in figure 3. As such, the default complex impedance of ZR can be changed to other values simply by changing values of components R1, R2, C1, C2, and C3.

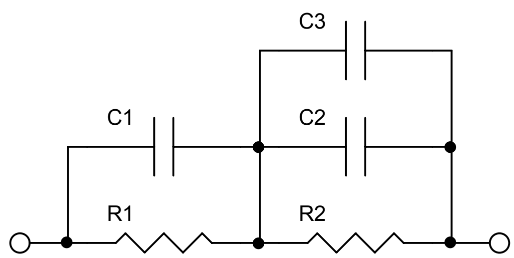


Figure 3. Complex Impedance Network

The default values for the components, in order to present a ZR impedance, are:

R1 = 2200 ohms
R2 = 8200 ohms
C1 = open (not used)
C2 = 10.0 nF
C3 = 1.50 nF

The position of these components on the TSPC is shown in figure 4.

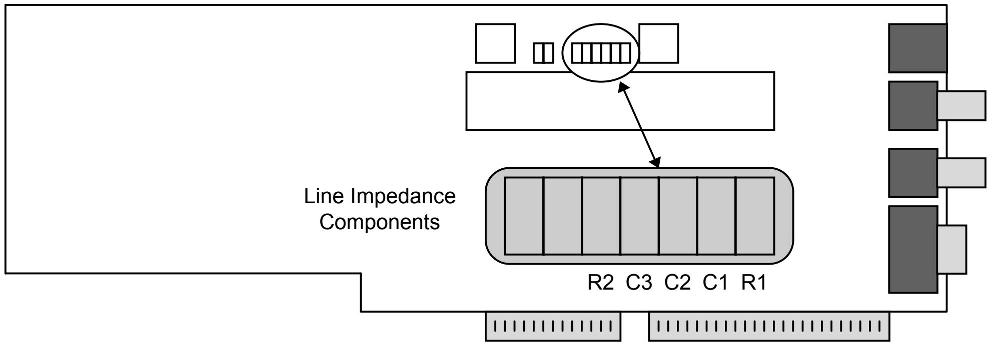


Figure 4. Complex Line Impedance Components

If the component values are to be changed, make sure that standard ESD precautions are taken during the removal of the existing components and soldering of the new components. Of course, the PC should be turned off and the TSPC removed from the PC before any changes are made.

■ Appendix F General Specifications

Note, the following specifications refer to the TSPC (Telephone Signal Processing Card) only. For technical information and specifications regarding the AI-7280, consult the “AI-7280 User Guide”.

Tone Generator			
	Output Level	-70 dBm to +6 dBm +/- 0.3 dB	
	Frequency Range	50 Hz to 10 kHz Flatness	+/- 0.3 dB
	THD+N	0.09% C-message	
	Harmonic Distortion	> 65 dBc	
	Frequency Accuracy	0.015%	
FSK Generator			
	Output Level	-60 dBm to 0 dBm +/- 0.3 dB	
	Frequency Range	100 Hz to 5 kHzFlatness	+/- 0.1dB
Noise Generator			
	Output Level	-60 dBm to -8 dBm +/- 0.3 dB	
Ring Generator			
	Output Level	0 Vrms to 80 Vrms.	
	Frequency Range	10 Hz to 600 HzFlatness	+/- 0.2 dB
	THD+N	0.1%	
	Frequency Accuracy	0.015%	
	Ringer Load	5 REN	
Telephone Line			
	Output Impedance	600 or 900 ohms +/- 2%	
	Output Impedance	complex ZR +/- 3%	
	Loop Voltage	20 - 52 Volts +/- 1V	
	Loop Current	20 - 40 mA +/- 10%	
Level Meter			
	Level Accuracy	+/- 0.2 dB	
	Frequency Range	10 Hz to 10 kHzFlatness	100 Hz to 5 kHz +/-
	0.2 dBMaximum Input	+14 dBm	
	Residual Noise	<-70 dBmC	
ACK Analyzer			
	Level Accuracy	+/- 0.2 dB	
	Maximum Input	+6 dBm per tone	
	Minimum Input	-20 dBm	
	Frequency Resolution	0.1 Hz	

■ Appendix G Technical Support

If you encounter problems while using the CID1500 Caller ID simulator, please contact us so that we can provide assistance. You may reach us in any one of the following manners:

Email: techsupport@adventinst.com

In North America:

Tel: (604) 944-4298
Fax: (604) 944-7488

Mail: Advent Instruments Inc.
111 - 1515 Broadway Street
Port Coquitlam, BC, V3C 6M2
Canada

In Asia:

Tel: (852) 8108-1338
Fax: (852) 2900-9338

Mail: Advent Instruments (Asia) Limited
Unit No. 7, 9/F, Shatin Galleria
18 – 24 Shan Mei Street
Fotan, Shatin, N.T.
Hong Kong

Note: Please include the program version, TSPC product code, revision code, option code, and TSPC base I/O address as given in the [HELP] [ABOUT CID1500] window in your correspondence. If using the AI-7280 hardware instead of the TSPC, please provide the unit's serial number.

Software updates are made available on our web site at:
www.adventinstruments.com