

1. Introduction

This document describes how to install and use the InnoSys IATE™ software package, with text-only display, for computers running the UNIX operating system. The text-only version of the terminal uses the “Curses” package for display control.

InnoSys also manufactures a version of IATE for use with the X Window System, a graphical user interface. X IATE is discussed in a separate manual.

IATE can be installed on several manufacturers' UNIX platforms, including Sun, Hewlett-Packard, and DEC systems.

IATE serves as airline reservation terminal, using the TCP/IP protocol for host communication through an InnoSys Gateway. The Gateway is a software and hardware package for Sun or Macintosh computers. Performing the functions of multiple terminal interchange controllers, the Gateway communicates with the airline network over direct data lines, or through a packet-switching network. Separate documents available from InnoSys describe the Gateway products.

IATE uses the ALC 6-bit character codes and the PARS/IPARS line control discipline. The terminal emulator provides functionality similar to that of a Raytheon PTS-100 terminal.

This manual explains how to install, configure, and use the terminal software. The discussion of keyboard commands includes default key assignments. If you wish to reassign any of these commands to different keys, please order the separate technical note “Modifying the PC or Curses IATE Terminal Keyboard Map” (InnoSys document #INTKEYS-1).

DEC and DIGITAL are registered trademarks of Digital Equipment Corporation.

Hewlett-Packard is a registered trademark of Hewlett-Packard Corporation.

Sun is a registered trademark of Sun Microsystems, Inc.

UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

X Window System is a trademark and product of the Massachusetts Institute of Technology.

IATE is a trademark of InnoSys, Inc.

All other product names mentioned in this document are trademarks or registered trademarks of their respective holders.

2. Installing the Terminal Software

Before installing the Terminal software, you must install a Gateway. You must also identify your IA, one or more terminal CRT TAs and possibly printer TAs for communication with the airline host system(s).

2.1 LAN Installation

The Terminal communicates with the Gateway through a TCP/IP LAN. Use the manuals provided by your LAN vendor to set up the LAN before proceeding with installation of the Terminal.

2.2 IATE Object Name

Installation of the Terminal will require an IATE CRT object name, and possibly a Printer object name. You chose one or more object names as part of the Gateway installation procedure. Make a note of the object names before installing the Terminal.

2.3 Copying the Terminal Software from Distribution Media

InnoSys usually distributes the terminal software on floppy disks or tapes. Extract the software using **cpio**, **tar**, or other commands as specified with the distribution. The **cpio** or **tar** and will refer to a device name for the floppy disk drive on your system. (Device names vary between system manufacturers and operating system releases.)

If you received a software update by electronic mail, you may need to use a **uudecode** command in order to extract the IATE files. If the resulting files end in the suffix **.Z**, they are compressed — use **uncompress** to expand them. Contact InnoSys if you have any difficulty extracting the software files.

After extracting the software, you should have the following files:

iate	Terminal executable program.
iate.act	“Special actions” file.
iate.cfg	Configuration file.
iate.pf	Function key definitions file.
termkeys	Keyboard map.
htable	Host definitions table.

Other files may be included with a distribution. One or more files may be omitted in some incremental software updates.

2.4 Setting Up the Terminal Software User Account(s)

Designate one or more UNIX user account names under which your terminal user(s) may access the IATE software. You may wish to use an existing user name, or create new name(s) for this purpose. Locate each user's home directory (this is the current directory after a user logs in). In a user's home directory, use command `ls -a` to get a complete list of existing files. If you find files named `.iate.cfg`, `.iate.pf`, `.htable`, or `.termkeys`, then an IATE installation apparently exists already. Make backup copies of these older files before proceeding.

The IATE distribution contains new files with the names mentioned above, but without the leading periods. If you did not find any of the files mentioned above in a user's home directory, simply copy the new files into the home directory — placing a period at the beginning of each destination file name.

If you did find any of these files in the user's home directory, the existing files are likely work with the new release of the software, except that `.iate.cfg` may require changes. With reference to the configuration information later in this document, it may be appropriate to compare the existing `.iate.cfg` file with the new distribution's `iate.cfg` file, and update the old file manually.

So that IATE users can easily start the terminal, it's important to place a copy of the `iate` executable program file into a directory that all users can access. Check the directory search path for each IATE user. The path is usually set in the `.login` or `.cshrc` file in a user's home directory. If a user's path setting does not already include the directory where you have placed the accessible copy of `iate`, add that directory name to the path setting in `.login` or `.cshrc`.

2.5 Editing IATE Files to Configure and Customize Your Installation

You can use an editor such as `vi` to examine and modify a user's `.iate.cfg` file to specify the configuration settings appropriate to that user. Users can also edit the `.iate.pf` file to define function key macro commands. The available configuration options and the PF key definition commands and syntax are discussed later in this manual.

You can also edit the `.termkeys` file to modify the keyboard map (the assignments of terminal command functions to particular keys). Do so with caution, following the instructions in InnoSys technical note #INTKEYS-1, "Modifying the PC or Curses IATE Terminal Keyboard Map." Note that only recent distributions of the keyboard map file, after July 1994, conform closely to the description in the technical note. Therefore, if you plan to modify the keyboard map, it may be easier to start from the latest version of the file, rather than modifying an older version.

3. Configuring the Terminal

The configuration information for Curses IATE is kept in the `.iate.cfg` file in each terminal user's home directory. As mentioned above, you can use an editor such as `vi` to edit this text file.

You can create alternative configurations under different file names. If you create an alternative configuration file, then in order to use your file instead of `.iate.cfg`, a user must specify the `-cfg` option on the command line when starting the IATE software:

```
iate -cfg filename
```

In a configuration file, each line should begin with a *keyword* specifying a particular configuration item. In most cases, an keyword must be followed by a *parameter* value. The distinction between upper-case and lower-case is important. One or more blank spaces or tabs should separate each keyword from its parameter value. Empty lines (consisting only of a carriage return) are permitted; they will be ignored.

3.1 Configuration Items

Most or all of the items described below will appear in a typical IATE configuration file. The next subsection will present an example of a complete configuration file.

ROWS
(Optional)

The parameter value specifies the number of lines each terminal screen will have. Most airline hosts expect a terminal to have at least 15 lines. You must configure the terminal to have at least as many lines as your host requires. For some systems, you may specify more. However, you may not specify more than the number of lines on your terminal or terminal emulation window, less 2 lines (one for the status area and one for the broadcast service area). In a 24-line window, your maximum allowable `ROWS` parameter value would be 22.

If you enter more than the number of lines allowed, you will be asked if you want to continue with the number of lines available. If you answer "Y" for "yes", then your `ROWS` setting will be reduced to the number of lines available.

This parameter defaults to 15 if you do not specify it.

Note: It is not advisable to resize a terminal emulation window while running IATE. The terminal will not recognize the change, and will continue to assume that the available display area retains its original dimensions. This could cause display problems after resizing a window.

SHOWRETURNS (Optional)	Some airline reservation hosts require that carriage returns (the ends of lines) be visible to the user. Some prefer that carriage returns be invisible. A parameter value of <code>TRUE</code> allows carriage returns that come from the host to be visible on the screen. <code>FALSE</code> hides host carriage returns. All carriage returns entered with the Return key are visible. This parameter defaults to <code>TRUE</code> if you do not specify it.
IDLEMINUTES (Optional)	If you specify this option in the configuration file, IATE will automatically shut down after the specified time period if there has been no keyboard input and no information has been received from the host. Specify a number of minutes (1 or more).
ALLOWTATIMEOUT (Optional)	If you specify this option and if it is also configured at the Gateway, it will enable a 1-minute timeout for communications with the Gateway. With the timeout enabled, the Gateway will automatically monitor the network communication link with the terminal software. If the link fails, the Gateway will allow 1 minute for the link to be restored. If it is not restored within 1 minute, then the Gateway will automatically close the terminal TA connection, making the TA object available for a new connection.
HOSTNAME (Required)	This option specifies the name of the host machine running the IATE Gateway software. (This is not the same as the airline's host system. The terminal will communicate with the airline host system via the IATE Gateway. The Gateway runs on the intermediate host system specified by the <code>HOSTNAME</code> option.)
SERVERNAME (Required)	This option specifies the name of the TCP port through which the terminal software will communicate with the IATE Gateway. This name is usually: <code>ialcserver</code>
LINKNAME (Required)	This option specifies a CRT object name configured at the Gateway. When the terminal software starts up, it will establish an association with the specified Gateway object for host communications. This association will survive until the user shuts the terminal down.
PRINTERNAME (Optional)	This option specifies the name of the destination printer object for a <code>PRINT_SCREEN</code> command. You must specify this option if you want the <code>PRINT_SCREEN</code> command to function. Format: <code>@hostname\portname\objectname</code>

3.2 Sample Configuration File

The contents of a typical IATE configuration file could look like this:

```
ROWS 22
SHOWRETURNS TRUE
ALLOWTATIMEOUT
BREAKFORKB
READWAIT 1000000
HOSTNAME hostmachine
SERVERNAME ialcserver
LINKNAME term1
PRINTERNAME prtr2
```

4. Starting and Stopping the Terminal

4.1 Normal IATE Startup

Each IATE terminal user must log in with the user ID of an IATE user account which you set up during the installation procedure. This is important because the IATE software will seek its configuration file and other files in the logged-in user's home directory.

To start the terminal, enter the command **iate**. The program should reside in a directory listed in the user's login path, as described during the installation procedure. If the "iate" command fails to invoke the terminal software, check the user's **.login** or **.cshrc** file to ensure that the directory containing **iate** is included in the user's path setting.

4.2 Starting IATE with a Special Keyboard Map or Configuration File

IATE will normally look for a file named **.termkeys** in the user's home directory. This file maps the different sequences of keystrokes to terminal functions. IATE will also look for a configuration file named **.iate.cfg**, as discussed earlier. To start IATE using a different key map file, specify the file name on the command line using the **-keys** option. To specify a different configuration file, use the **-cfg** option. For example:

```
iate -keys parskeys
```

would start the Metro terminal using a key map file named **parskeys**.

4.3 Stopping IATE

To exit the IATE software and return to the shell command prompt, press **Ctrl-X** twice in succession. If you have edited the key mapping for the QUIT command, the required key may be different.

If the terminal does not respond to your command to quit, try pressing **Ctrl-C**. After a **Ctrl-C** exit, on some systems, it may be necessary to enter **Ctrl-J reset Ctrl-J** to restore normal keyboard operation.

5. Using the Terminal

When you start IATE, you have one screen available for host entries and responses. There are three primary parts to the display:

- the *terminal emulation area*;
- the *status line*; and
- the *broadcast service line*.

5.1 The Terminal Emulation Area

The *terminal emulation area* is 64 characters wide. The number of rows it contains is set by the `ROWS` parameter in the terminal configuration file discussed earlier. Host entries and responses appear in the emulation area. The terminal sends an entry to the host each time the **Enter** key is pressed. Until the terminal receives a response from the host, the keyboard remains locked and will not respond to operator input, with the exceptions noted below. When the terminal receives a message from the host, the message appears in the terminal emulation area, and the keyboard becomes active again.

It is possible, in the following cases, for the terminal keyboard to be unlocked before the terminal properly receives a message from the host.

- A message to the host may be lost in transit. This can happen for a number of reasons — most commonly, noise on the communication line. Regardless of the cause, the host will not respond to a lost message; this means that without the operator taking special action, the keyboard will remain locked indefinitely. To break out of this locked state, press the **Reset** key to unlock the keyboard.
- A message from the host is received incorrectly (in a garbled form). In this case the “Push Rpt” message appears on the status line and the terminal automatically unlocks the keyboard, allowing the operator to request a retransmission of the message if desired.

5.2 The Status Line

The *status line* presents information about the session with the airline host. Status line labels include:

Sys Avail	The system is available for use.
Sys Unavail	The system is unavailable. Either the host is down, the gateway (multiplexer) is down, or the host/gateway or gateway/terminal connection is down.
KeyBd Locked	A message or command has been sent to the host, but a response has not yet been received. If no response is returned by the host, the Reset key (Esc) may be pressed to unlock the keyboard.
Push Rpt	The terminal has received a message from the host with an error; or, a message from the host is longer than the window size that the operator has selected; or, the message from the host has invalid screen positioning commands. If this message appears, press the Reset key and re-enter the last command.
Umsg	The terminal has received an unsolicited message indicator from the host. Press the Umsg key to display the text of the Unsolicited Message.
Reenter	The information cannot be sent to the host. If this message appears, press Reset and re-enter the last command. This message usually means that there was no SOM on the screen when Enter was pressed.
Insert	Indicates the emulator is in <i>insert</i> mode. Any characters typed will be inserted between the existing characters as long as this message appears in the Status box. Each line contains 64 characters. Any characters pushed beyond the 64th column by the insert function are lost.
Logging	Indicates that entries & responses are being logged to a session file.
Printing	Indicates that the terminal is waiting for a print job to be completed.

5.3 The Broadcast Service Line

The broadcast service line is the bottom line on your display. It is used to support special message functions, and also to display prompts during programmable key function execution.

5.4 Working with Multiple Screens

IATE can have multiple terminal screens open at one time. You may keep as many as 15 screens open at any time. Each screen retains its host entry and response text even when the screen is not visible. Therefore, you may leave one screen to work in another without losing the text you had been working on.

The host does not know that you may be using multiple screens. Host responses will appear in whatever screen you are using when the response arrives. So, for example, if you enter a transaction in Screen 1 and switch to Screen 2 before the response appears, the host response will appear on Screen 2.

The *New Screen* command opens a new, empty terminal screen and places the cursor at the home position. IATE assigns a number to each screen. The screen number is shown in the lower right hand corner of the IATE display. Each screen retains its number until it is deleted. New screens reuse the numbers of deleted screens.

The *Delete Screen* command removes the current screen, deleting its contents entirely. When you have only one screen, "Delete Screen" does nothing. The screen that would be chosen if you issue the "Last Screen" command is made the current screen.

The *Next Screen* command takes you to the next screen available. If there is only one screen, nothing changes. When you have started more than one screen, each time you use the "Next Screen" command, you are taken to the next available screen in ascending numerical order. When you are in the highest numbered screen, you are taken to the first screen again in 'round-robin' fashion.

The *Previous Screen* command toggles between two screens. If there are only two screens, "Last Screen" takes from one to the other. However, when there are more than two screens, "Last Screen" will take you from the current screen to the last screen in which the cursor appeared. This command can be very handy when you are working on two related entries (a fare quote and a flight availability, for instance) and you need to see each group of host responses in order to make entries on the other screen.

The *Split Screen* command allows you to view and use two screens at the same time. However many lines are available for a full-sized screen are split between the two screens. (This number of lines may not necessarily be the same as the number of rows you have configured for your screens.) Entries and responses that lie on rows below the number of rows for split screens will be kept until you change back to full screens. A line appears between the two screens so that you know where the first ends and the second begins. For some hosts (SABRE), when you split screens, the host is notified of the split and will respond that it is allowed. Other hosts (SODA) may then send responses to screens other than the one from which you are making entries. Still other hosts may send too much data for the number of lines available for split screens. This may result in a "Push Rpt" error message. Check with your host administrator for

information about how the host handles split screens.

The *Full Screen* command returns all screens to their configured number of rows. The line separating the split screens is removed from the computer monitor. You see only one terminal screen at a time. Any data that was beyond the number of rows available for the split screens is once again visible. If the host so requires, the terminal will inform it of the return to full size screens, and you may see that full screens are allowed (SABRE). If host responses were being routed to different split screens, they will now be routed only to the current screen (SODA).

5.5 Keyboard Functions

IATE recognizes the following keys as they appear on the keyboard, and will simply display the corresponding character when the key is pressed:

```

ABCDEFGHIJKLMN OPQRSTUVWXYZ
0123456789 - / .

```

A number of special purpose characters and messages are used for airline terminal emulation. These include the following:

<u>Name</u>	<u>Key</u>
<u>Change</u>	<u>⌈</u>
Appears as	ø on keyboard maps, and as _ on screen.
<u>Cross of Lorraine</u>	<u>⌋</u>
Appears as	¥ on keyboard maps, and as \ on screen.
<u>Display</u>	<u>*</u>
Appears as	* on keyboard maps and on screen.
<u>End Item</u>	<u>⌘</u>
Appears as	on keyboard maps, and as " on screen.
<u>Field Mark</u>	<u>^ or `</u>
Appears as	† on keyboard maps, and as ^ on screen.
<u>New Line</u>	<u>⌞</u>
Appears as	⌞ on keyboard maps.
<u>Start of Message</u>	<u>Ctrl-X S</u>
Appears as	> on keyboard maps, and as } on screen.
<u>Set Optional Field</u>	<u>~</u>
Also known as	"lightning bolt".
<u>Unsolicited Message</u>	<u>Ctrl-X U</u>
Abbreviated as	UMG on keyboard maps.

Last Message Ctrl-U L

Copies the last message that you entered, into the current window, starting at the current cursor position; and positions the cursor to the right of the message. May be abbreviated as **Lst Msg** on keyboard maps.

Re-enter Ctrl-E

Re-transmits the last message entered, starting from the latest SOM character on the screen. This command simulates an **Enter** keystroke in order to send the message to the host. Abbreviated as **ReE** on keyboard maps.

Clear Service Ctrl-X B

Clears the Broadcast Service Line.

Protect Ctrl-X \

Toggles SABRE Protected mode on and off.

FOS Toggle Ctrl-X F

Toggles FOS terminal mode on and off.

Arrival Unknown Ctrl-A A

Abbreviated as **ARK** on keyboard maps. *Not available on SABRE hosts.*

Cancel Transaction Ctrl-A E

Abbreviated as **XITN** on keyboard maps. *Not available on SABRE hosts.*

Ignore Ctrl-A I

Abbreviated as **IGN** on keyboard maps. *Not available on SABRE hosts.*

Payment \$

Abbreviated as **\$** on keyboard maps. *Not available on SABRE hosts.*

Repeat Ctrl-A R

Abbreviated as **RPT** on keyboard maps. *Not available on SABRE hosts.*

The following keys control the cursor position and the screen display:

<u>Name</u>	<u>Key</u>
-------------	------------

<u>Arrow Keys</u>	<u>_, =, etc.</u>
-------------------	-------------------

Moves the cursor one row/column in the direction of the arrow.

<u>Back Tab</u>	<u>Ctrl-X Tab</u>
-----------------	-------------------

Moves the cursor immediately to the right of the first host tab character found on the screen to the left of the current cursor position. Abbreviated as **Bck Tab** on the keyboard maps.

- Clear Ctrl-X Backspace
Erases the page displayed in the active window, puts SOM in row 1, column 1 of the active window, and positions the cursor to the right of the SOM. (On SABRE hosts, the SOM character will not be visible.) May be abbreviated as **Clr** on the keyboard maps.
- Clear All Ctrl-A Backspace
Clears the text on all of the terminal's windows, regardless of which window(s) are currently displayed. May be abbreviated as **All** on keyboard maps.
- Home Ctrl-X H
Positions the cursor at row 1, column 1, in the active window.
- Tab Tab
Moves the cursor immediately to the right of the first host tab character that is found on the screen to the right of the current cursor position.
- New Screen Ctrl-N N
Opens a new screen, using a new window number in ascending numerical order. Abbreviated as **New Scr** on keyboard maps.
- Delete Screen Ctrl-N W
Closes the screen currently displayed, unless it is the only screen remaining. Abbreviated as **New Scr** on keyboard maps.
- Next Screen Ctrl-X N
Moves cursor to the next higher screen, in ascending numerical order, whenever more than 1 screen is open. Abbreviated as **Nxt Scr** on keyboard maps.
- Previous Screen Ctrl-A N
Moves cursor to the screen which had been viewed just prior to the screen now in use. Abbreviated as **Lst Scr** on keyboard maps.
- Split Screen Ctrl-A S
Opens a split-screen display, with one screen area occupying the top half of the emulation area, and another using the bottom half. Abbreviated as **Sp1 Scr** on keyboard maps.
- Full Screen Ctrl-X F
Closes a split-screen display, restoring the full-sized screen display in the emulation area. Abbreviated as **Ful Scr** on keyboard maps.

The following keys can be used to edit text on the terminal screen:

<u>Name</u>	<u>Key</u>
<u>Destructive Backspace</u>	<u>Backspace</u>
	Moves the cursor one column to the left, and erases any character that had previously existed at the new cursor location.
<u>Erase to End Of Page</u>	<u>Ctrl-X D</u>
	Erases all characters from the cursor position to the bottom of the page. Abbreviated as EOP on keyboard maps.
<u>Insert</u>	<u>Ctrl-X I</u>
	Toggles <i>character insertion mode</i> on and off. When this mode is toggled on, each character typed will push any characters on its right 1 column further towards the right. Any characters pushed past the end of the line will be lost.
<u>Insert Line</u>	<u>Ctrl-X K</u>
	Inserts a blank line after the line where the cursor is positioned. Text on the last line of the page is deleted. Abbreviated as ILn on keyboard maps.
<u>Delete Character</u>	<u>Del</u>
	Deletes the character at the cursor position; and moves all characters that lie to the right of the cursor, on the same line, one column to the left. Abbreviated as DCh on keyboard maps.
<u>Delete Line</u>	<u>Ctrl-X L</u>
	Deletes the line where the cursor is positioned, and pulls up all of the lines below the deleted line. Abbreviated as DLn on keyboard maps.
<u>Copy</u>	<u>Ctrl-Y</u>
	Saves a copy of the data between the SOM and the current cursor position. The terminal stores the copy in a "clipboard", an area of memory that holds just one copy at a time. Each time the copy command is used, the previous contents of the clipboard will be overwritten. This key is abbreviated as Cpy on keyboard maps.
<u>Paste</u>	<u>Alt-V</u>
	Inserts the current contents of the clipboard onto the screen, starting at the current cursor position. If the data is longer than the current line, it "wraps around" to the next line. Lines of data that were already on the terminal screen, beneath the cursor, will move down to make room for the data contained in the clipboard. The contents of any lines moved down past the bottom of the display will be lost. This key is abbreviated as Pst on keyboard maps.
<u>Print Window</u>	<u>Ctrl-P</u>
	Prints the contents of the current window.
<u>Form Feed</u>	<u>Ctrl-A A</u>
	Generates a "form feed" on the printer, advancing the paper to the next page.

By using the “Copy”, “Paste”, and “Last Message” commands, you can transfer data between terminal screens. For example, you may copy from one screen, switch to a second screen, and then paste. The data from the first screen will be inserted on the second screen, starting from the current cursor position on that screen.

5.6 Programmable Function Keys

The IATE terminal has 30 programmable function keys. For a description of the function key command language, please see Appendix A. To define a key, you will need to create a text file with the text editor of your choice. You may assign this “function key definition file” any legal file name that you choose. Into this file you must place the definitions for the keys you want to load. The *Load Function Keys* (**Ctrl-A L**) command will load the function key definitions. It is possible to create more than one such file, if you wish.

You may include as few as one function key definition in a file, or as many as you please. If you have two definitions for the same key number, the last one encountered in the file will be the one that is loaded.

The Required Format for Key Definitions

All definitions have three elements: a *key number*, an optional *key label*, and the *command(s)* associated with the key. These elements are separated by white space (tabs or spaces) and carriage returns. You may have up to 8 characters in a key label, and up to 255 characters in a key command. Any excess characters will not be loaded.

You may include comments in your function key load file by placing a “#” character at the leftmost column of any line. All such lines are ignored. All lines that do not begin with either a “#” character, a key number, or white space, will end any definition that is being processed.

As an example, a function key definition file with keys 1 through 3 defined could have the following contents:

```

1 INTLFLGT
  ^SR5.IRP-INTERNATIONAL BOARDING PASSES ISSUED AT AIRPORT
  5.IRP-VALID PASSPORT REQUIRED FOR ENTRY PURPOSES^E
# INTLFLGT CAUSES COMMENTS TO BE PRINTED ON ITINERARY

2 LIMO
  ^SR5.IR-LIMO RATE ^V INCLUDES
  5.IR-GRATUITY.**NO SHOWS WILL BE CHARGED**^E
# LIMO PRINTS RATE INFORMATION ON ITINERARY

3 NO CAR
  ^SR5.SR-^V*NO ^V REQUESTED^E

```

Each definition begins at the leftmost column, with a key number from 1 through 30. This tells IATE the key into which you want to load the definition. Optionally, on the same line, you may add a short description of the key: a label of up to 8 characters. You will be able to review a list of these labels by using the *Function Key Labels* command discussed below.

On lines following a key number line, you must place the actual command(s) used in this function key. These lines must not begin at the leftmost column: some white space must precede any characters that you want to load into a key. When processing the file, IATE will add together (concatenate) sequences of such lines to form a command “script” for the specified key — until reaching the 255 character maximum length, a new key number line, or the end of the file.

The carriage returns at the ends of the lines are retained in the loaded script as “end item” characters (). These will be converted to carriage returns when sent to the host.

The *Function Key Labels* command (Ctrl-A V) allows you to see the short labels that you have defined for your function keys. Viewing your labels can help you to remember what you have programmed the function keys to do. When you issue this command, a numbered list of your labels appears in the space along the right edge of the screen, in increasing numerical order from the top down. This command acts as a toggle: The first time you issue the command, your list will be displayed. The next time, the list will be removed from the monitor.

If you have less than 32 lines on your screen, you will not be able to view all of your programmable function keys. The list will attempt to display as many of your key labels as it can while reserving the last two lines of your monitor for the status area and broadcast service line. For instance, if you have a 25 line screen, you will be able to view the first 23 programmable function key labels.

5.6.1 Key Definitions Loaded Automatically at Startup

At startup, IATE will look in the user's home directory for a function key definition file named `.iate.pf`. If this file does exist in the IATE directory, then IATE will automatically load the key definitions from that file, and will display the loaded definitions on the screen. Key definitions in the file should have the format described above.

5.6.2 Commands Executed Automatically at Startup and Shutdown

You can specify commands for IATE to execute automatically whenever you start or stop the terminal software. You can define your startup and shutdown commands in a manner similar to function key definitions, although the procedure is not quite the same.

If you want to define commands for the terminal to execute automatically at startup, you must place your commands in the file `.startup.pf`. If you want to define commands to execute automatically when you quit the terminal, you must place these commands in `.shutdown.pf`. The required format is the same as the *commands* part of a function key definition: simply enter the command into the file. Do not enter function key labels or names into these files. Enter commands only. Here is a simple example of a startup command for a sign-in:

```
^SRSI SIGNIN000
```

If you place the sign-in command shown above into the `.startup.pf` file, then IATE will automatically prepare a "SI SIGNIN000" command when you start the terminal. The sign-in command will appear on the terminal screen immediately. As soon as the host connection becomes available (when "Host Avail" appears on the status line), then you can press **Enter** to send the sign-in command from the screen to the host.

Now suppose that you have also placed the following line into `.shutdown.pf`:

```
^SRSO^E
```

When you quit the terminal, IATE will send the specified "SO" sign-out command to the host, and then exit to DOS.

These are examples only. The startup and shutdown command files are not limited to a single line or command. You can enter multiple commands into these files.

Reminder: Do not include function key labels or names in the startup and shutdown command files. These files do not define function key assignments — they simply define commands for IATE to execute. This is the principal difference between these command files and the function key definition files discussed earlier.

For a complete description of the function key command language, please refer to Appendix A.

5.7 Session Files

It is often useful to have a record of host entries and responses. For instance, such a record can be used to review information that would otherwise be lost (e.g., in the case of “simultaneous changes”). The IATE terminal allows you to create a session file which records all entries and responses.

The *Session File* command (**Ctrl-A S**) requests a session file name, and then starts the capture of data into that file. To stop data capture, issue the command again.

The “View Session File” command (**Ctrl-A V**) allows you to view the contents of the session file. When you choose to view a session file, a special full-screen display will replace your current screen. The data that appears first will be your most recent entries and responses. Use the `+[]` up-arrow and down-arrow keys to move forward and backward within the Session file. (*Note: When you are viewing the beginning of the file, the up-arrow key function cannot move up any further. Likewise, when you reach the end of the file, the down-arrow function will not move down any further.*)

While viewing the Session screen, you will not be able to send or receive messages from the host. To stop viewing your session file and return to normal terminal emulation at any time, press the **Enter** key or the **Reset** key.

5.8 Special Actions File

The *Special Actions File*, **.iate.act**, contains definitions of up to 6 special action keys. Each special action key may contain up to 7 characters. These functions may be mapped to **Ctrl-A 1**, **Ctrl-A 2**, and so forth through **Ctrl-A 6**.

Special action keys are similar to function keys. The differences are that special action key definitions cannot include programming commands, and they are limited to a length of 7 characters. Special action keys are a convenient way to enter special codes that may be used on a specific airline host.

Appendix A. Function Key Commands

This Appendix lists the commands that you may use to define your function keys, and presents examples illustrating the use of these commands.

All function key commands include the *Field Mark* (displayed as a “↑” character on the screen) followed by a letter or number code which represents a command. When defining your function keys, type the “^” character to designate a Field Mark.

A.1 Function Key Command Codes

The following function key command codes are available:

<i>nn</i>	Fixed field length
A	Send entire screen to host
C	Position insertion point ¹ (cursor)
D	Insert optional text
E	Auto Enter
F	Execute PF Key
H	Wait for reply from host
I	Search for tab character
K	Erase to end of page
O	Input optional text
R	Send the next host reply to specified window
S	Move SOM to specified location
T	Display prompting text
V	Fill in variable length field
W	Wait for input
X	Execute commands from a specified file

A.2 Command Code Examples

nn This command copies a fixed number of characters (the number *nn*) from the screen, beginning from SOM position, into the character string generated by pressing a P.F. key. For example, if the contents of PF1 are:

```
FL↑03LAX
```

and the user enters “DFW” at the SOM and then presses the PF1 key, then the screen displays:

```
FDDFWLAX
```

¹ The term *insertion point* refers to the current location of the cursor on the currently displayed terminal screen.

- A** This command is used to include all data in the terminal window in any message to the host. For example, if a form was displayed in the terminal window for the user to fill in, and PF1 was defined as follows:

```
↑A↑E
```

then pressing the PF1 key would send the form to the host.

- C** This command moves the insertion point to the specified location in the terminal window. This command takes the following forms:

↑C* Positions the insertion point at the SOM.

↑Cxx Positions the insertion point at the leftmost column of the specified line xx.

↑C-cc Positions the insertion point at the specified column cc on the current line.

↑Cxx-cc Positions the insertion point at the specified line xx and column cc.

For example:

```
↑C20-45DFWLAX
```

would position the insertion point at line 20, column 45, and write:

```
DFWLAX
```

starting from that point.

Note: Use of the SOM positioning and cursor positioning commands eliminates the requirement that PF keys be pressed only when the insertion point is at the SOM. Data to fill in fields is still taken from the old SOM to the insertion point where the PF Key was pressed, as it is with any PF Key command.

- D** This command allows the user to replace data in a PF key command with data that the user enters. The data in the PF Key command that may be replaced is delimited with a Field Mark. The end of the replacement data (not limited to the number of characters being replaced) must either be delimited by a Field Mark or be at the end of the PF Key data (that is, if the PF Key was pressed with the insertion point at the end of the data). If the user has entered no data, then no replacement occurs.

For example, if the contents of PF1 are:

```
FDDFWLAX↑D-AA
```

and the user enters

```
-DL
```

and then presses the PF1 key when at the SOM, the following command appears

in the terminal window:

```
FDDFWLAX-DL
```

If user presses the PF1 key without entering any data, the following command appears in the terminal window:

```
FDDFWLAX-AA
```

- E** This function sends the characters between the SOM and the current insertion point to the host as if the user had pressed the enter key. For example, if the contents of PF1 are:

```
FDLAXDFW↑E
```

and the user presses the PF1 key with the insertion point at the current SOM, the fare display command

```
FDDFWLAX
```

is displayed on the screen and sent to the host.

- F** This function allows the user to execute a PF Key from inside another PF Key function. PF key is defined as two digits (e.g. as F03 not F3). For example, if the contents of PF1 are:

```
FDDFW↑F24↑E
```

and the contents of PF24 are:

```
LIT
```

and the user presses PF1 at the current SOM, then the fare display command:

```
FDDFWLIT
```

is displayed on the screen and sent to the host.

- H** This function stops execution of the current PF Key until a response is received from the host. This function normally follows the ↑E PF key function. For example, if the contents of PF1 are:

```
FDDFWIAH↑E↑HFDDALHOU↑E
```

and the user presses the PF1 key at the current SOM, the first fare display command:

```
FDDFWIAH
```

is displayed on the screen and sent to the host. When the response is received, the second fare display command:

```
FDDALHOU
```

is displayed on the screen and sent to the host. While waiting for the host reply, the message "Holding for host reply — Press reset to stop" appears on the status line. Pressing the **Reset** key will terminate the PF key function.

- I** This command searches for the next tab character or user-defined tab stop in the terminal window, and moves the insertion point to that location on the current line.

- K** This function erases all text from the current insertion point through the last character in the terminal window.

- O** This command optionally replaces a field in a PF Key command with a replacement field entered by the user. The **O** PF Key command must be followed by a field number, for example, **O0**, **O1**, **O2**, **O3**, etc. Field numbers must be in the range 0 - 9. Each optional field in the PF Key definition is terminated by the Field Mark (**†**). If the user enters an optional field number and no data, then that field is dropped from the PF text.

Replacement field data that the user enters on the terminal screen must be preceded by the **†** character and the number of the field to be replaced, and must be ended by a Field Mark.

For example, if the contents of PF1 are:

```
FD†O1TUL††O2IAH††O3-WN††E
```

and the user presses the PF1 key, then the command

```
FDTULIAH-WN
```

is sent to the host.

As another example, if the user enters

```
O2LAX
```

and presses the PF1 key, the command

```
FDTULLAX-WN
```

is sent to the host.

As a final example, if the user enters

```
O3
```

and presses the PF1 key, the command

```
FDTULIAH
```

is sent to the host.

- Pnn** This command pauses execution of the PF Key for the indicated number of seconds *nn*. The number of seconds must be followed by a Field Mark (**†**).

Rnn This command sends the next host reply to window *nn*. If a host reply command $\uparrow H$ is not subsequently given, the **R** command is ignored. If window *nn* does not exist, the user is asked whether to cancel the function key or send the output to the current window. For example, if the contents of PF1 are:

```
ALAXORD $\uparrow E$  $\uparrow R$ 02 $\uparrow H$ 
```

then the host would be sent

```
ALAXORD
```

and the reply will be sent to window #2.

S This command moves the SOM to the specified location in the terminal window. This command takes the following forms:

- $\uparrow S*$ Position SOM at the current cursor location (only when SOM characters and carriage returns are being displayed — otherwise an error occurs and key execution stops).
- $\uparrow Sxx$ Position SOM at the leftmost column of the specified line *xx*.
- $\uparrow Sxx-cc$ Position SOM at the specified line *xx* and column *cc* (only when SOM characters and carriage returns are being displayed — otherwise an error occurs and key execution stops).
- $\uparrow SR$ Position SOM at the leftmost column of the line below the current line. (If the current line is at the bottom of the window, the SOM wraps around to the top.)

For example, if the contents of PF1 are:

```
 $\uparrow SR$  $\uparrow C$ *ADFWLAX $\uparrow E$ 
```

then pressing PF1 with the insertion point anywhere in a terminal window causes the SOM to be placed at the beginning of the next line. Then the insertion point is placed at the current SOM (see the definition of the **C** PF key command). The following command appears on the screen and is sent to the host:

```
ADFWLAX
```

Note: Use of the SOM positioning and cursor positioning commands eliminates the requirement that PF keys be pressed only when the insertion point is at the SOM. Data to fill in fields is still taken from the old SOM to the insertion point where the PF Key was pressed, as it is with any PF Key command.

- T** This command displays, on the second line of the status box, any text that follows the **T** command. The text must be terminated by either the Field Mark or the end of the key definition. The *Wait for Input* command (**↑w**) often follows the **T** command. For example, suppose the contents of PF1 are as follows:

```
FD↑TENTER DESTINATION↑SFC↑W↑E
```

If the user presses the PF1 key and then enters:

```
DFW↑
```

then the following command is sent to the host:

```
FDSFODFW
```

- V** This command optionally fills in fields in a PF Key command with fields entered by the user. Multiple fields may be replaced. If the user enters multiple fields, they should be separated by Field Marks. For example, suppose the contents of PF1 are as follows:

```
FD↑V↑V↑E
```

If the user enters:

```
DFW↑LAX
```

and then presses the PF1 key, the following command is sent to the host:

```
FDDFWLAX
```

- W** This command stops further processing of a PF key command until the user enters data. The user must terminate the input with a Field Mark. The message "Press Reset or end input with Field Mark" appears on the status line. Pressing the **Reset** key cancels the PF key execution.

For example, if the contents of PF1 are:

```
FDDFW↑W↑E
```

and the user presses the PF1 key, the command:

```
FDDFW
```

is displayed. The user must enter data terminated by a Field Mark in order to complete the PF key execution.

- x** This command executes commands stored in a specified file. The file name immediately follows the **x** command, and is terminated by a Field Mark or the end of the PF key definition. Lines in the file are executed just as if they were part of a PF key definition. The name of the file must be followed by a Field Mark. Inside the file, the Field Mark should be entered as the “^” character. All letters that are part of a message to the host should be entered as upper case letters.

The first characters of the first line in the file should be the SOM/Cursor Reset command `^SR^C*` (or `↑SR↑C*`). The first characters of each subsequent line in the file should be the Cursor Reset command `^C*` (or `↑C*`). Each host command (such as `AMSPORD`) should be followed by the commands `^E^H` (or `↑E↑H`). The message “Holding for reply — Press **Reset (Esc)** to end” appears on the status line while the **x** command is executing. Pressing the **Reset** key cancels key execution. For example, if the contents of PF1 are:

```
↑XBOSTON.PF
```

and the contents of the file “BOSTON.PF” are:

```
^SR^C*AMSPBOS^E^H
^C*A*^E^H
^C*FDMSPBOS24APR^E
```

and if you press the PF1 key, then the following commands will be sent to the host:

```
AMSPBOS
A*
FDMSPBOS24APR
```

The host's responses to each command will be displayed on the screen.

- y** This function sends the characters between the SOM and the current insertion point to an application that has been linked to the terminal. For example, if the contents of PF1 are:

```
FDLAXDFW↑Y
```

and the user presses the PF1 key with the insertion point at the current SOM, then the fare display command:

```
FDDFWLAX
```

is displayed on the screen and sent to the application.

UNIX IATE™
(Curses Version)

Terminal Software

User's Guide

InnoSys
I N C O R P O R A T E D

Copyright © 1995 InnoSys Incorporated

This manual and the software described in it are copyrighted, with all rights reserved. Under the copyright laws, this manual or the software may not be copied, in whole or in part, without the written consent of InnoSys Incorporated.

NO WARRANTIES OF ANY KIND ARE EXTENDED BY THIS DOCUMENT. The information herein and the IATE™ products themselves are furnished only pursuant to and subject to the terms and conditions of a duly executed Product License. INNOSYS SPECIFICALLY DISCLAIMS ALL WARRANTIES, WHETHER IMPLIED OR EXPRESSED, INCLUDING BUT NOT LIMITED TO THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. InnoSys has no responsibility, financial or otherwise, for any result of the use of this document and/or the associated product, including direct, indirect, special and/or consequential damages. The information contained herein is subject to change without notice.

Sun, Sun Workstation, Solaris, Sun OS, and S-Bus are trademarks or registered trademarks of Sun Microsystems, Inc. SPARC is a registered trademark of SPARC International, Inc. Apple, AppleTalk, ImageWriter, LaserWriter, and Macintosh are registered trademarks of Apple Computer, Inc. Finder, MultiFinder, and HyperCard are trademarks of Apple Computer, Inc. NuBus is a trademark of Texas Instruments Corporation. UNIX is a trademark of UNIX Systems Laboratories, Inc.

IATE is a trademark of InnoSys Incorporated. Copyright © 1990-1994 InnoSys Incorporated and Apple Computer, Inc.

INSCC, INSCC-S, and INSCC-LC cards have been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If a computer with INSCC, INSCC-S, or INSCC-LC card(s) installed does cause harmful interference to radio or television reception, which can be determined by turning the computer off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Increase the separation between the computer and receiver (radio or TV)
- Connect the computer into an outlet on a circuit different from that to which the receiver is connected.

Note that shielded cables must be used with INSCC, INSCC-S or INSCC-LC cards in order to meet the FCC Class B emission limits. Changes or modifications to the INSCC, INSCC-S or INSCC-LC cards not expressly approved by InnoSys Inc. could void the customer's right to operate the equipment.

InnoSys Incorporated
3095 Richmond Parkway, Suite 207
Richmond, CA 94806-1900
(510) 222-7717 Voice
(510) 222-0323 FAX

Contents

1	Introduction	1
2	Installing the Terminal Software	2
2.1	LAN Installation	2
2.2	IATE Object Name	2
2.3	Copying the Terminal Software from Distribution Media	2
2.4	Setting Up the Terminal Software User Account(s)	3
2.5	Editing IATE Files to Configure and Customize Your Installation	3
3	Configuring the Terminal	4
3.1	Configuration Items	4
3.2	Sample Configuration File	6
4	Starting and Stopping the Terminal	7
4.1	Normal IATE Startup	7
4.2	Starting IATE with a Special Keyboard Map or Configuration File	7
4.3	Stopping IATE	7
5	Using the Terminal	8
5.1	The Terminal Emulation Area	8
5.2	The Status Line	9
5.3	The Broadcast Service Line	9
5.4	Working with Multiple Screens	10
5.5	Keyboard Functions	11
5.6	Programmable Function Keys	15
5.6.1	Key Definitions Loaded Automatically at Startup	17
5.6.2	Commands Executed Automatically at Startup and Shutdown	17
5.7	Session Files	18
5.8	Special Action File	18
Appendix A	Function Key Commands	19
A.1	Function Key Command Codes	19
A.2	Command Code Examples	19