© 1999 TC License, Ltd. All rights reserved. TRANSCORE is a registered trademark of TC License, Ltd. All other trademarks listed are the property of their respective owners. Contents are subject to change. Printed in the U.S.A.

For further information, contact:

TransCore
3410 Midcourt Road, Suite 102
Carrollton, Texas 75006 USA

Phone: (214) 461-4031
Fax: (214) 461-6478

Technical Support
Web: transcore.com/rfidsupport
Phone: (505) 856-8007

For comments or questions about this document, e-mail tech.pubs@transcore.com.
WARNING TO USERS IN THE UNITED STATES

FCC RADIO FREQUENCY INTERFERENCE STATEMENT
47 CFR §15.105(a)

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency (RF) energy and may cause harmful interference to radio communications if not installed and used in accordance with the instruction manual. Operating this equipment in a residential area is likely to cause harmful interference, in which case, depending on the regulations in effect, the users may be required to correct the interference at their own expense.

NO UNAUTHORIZED MODIFICATIONS
47 CFR §15.21

CAUTION: This equipment may not be modified, altered, or changed in any way without permission from TransCore, LP. Unauthorized modification may void the equipment authorization from the FCC and will void the TransCore warranty.

USE OF SHIELDED CABLES IS REQUIRED
47 CFR §15.27(a)

NOTE: Shielded cables must be used with this equipment to comply with FCC regulations.

TransCore, LP
USA
Contents

1 Overview

   About this Document ......................................................... 1-3

2 Setting up the Programmer

   Optional Accessories ......................................................... 2-3
   Electrical Power ............................................................... 2-4
   Preventing Electrostatic Discharge Damage .............................. 2-5
      Antistatic Workstation .................................................... 2-6
      Grounding Wire ............................................................. 2-6
      Using the ESD Wrist Strap ............................................... 2-6
      Antistatic Mat ............................................................... 2-7
      Ionized Air Blower ......................................................... 2-7
      ESD-Safe Tags Container .................................................. 2-7
      Seals and Plugs for Tag Programming Ports ........................... 2-7
   Preventing Condensation ..................................................... 2-8
   Tag Programming Head ....................................................... 2-8
   Positioning Brackets ......................................................... 2-8
   Removing/Installing the Tag Positioning Bracket ....................... 2-9
   Connecting the Host Device ................................................. 2-10
      Interfacing the Programmer With the Host Using a Modem .......... 2-11

3 Operating the Programmer

   Indicator Lights ............................................................... 3-4
   Sign-On .............................................................................. 3-4
   Command Execution Protocol ............................................... 3-5
   RS–232 Command Set ........................................................... 3-5
List of Figures

Figure 2-1  Programming Workstation with ESD Controls  .................................................. 2-6
Figure 2-2  Tag Port with ESD Seal  ................................................................................. 2-7
Figure 2-3  Tag Positioning Brackets  ............................................................................... 2-9
Figure 2-4  Serial Interface Connection to Host Computer  ............................................. 2-10
Figure 3-1  AP4110 Programmer Indicator Lights .............................................................. 3-4
Figure 5-1  Pin Block on Tag Programming Head ............................................................... 5-3

List of Tables

Table 2-1  AP4110 Tag Programming Head Kit Options ................................................. 2-4
Table 2-2  Modem Setup Parameters ............................................................................... 2-11
Table B-1  RS-232 Connector Pinout ............................................................................... B-3
Table C-1  6-Bit ASCII Codes ......................................................................................... C-3
Table D-1  4-Bit ASCII Hex Codes .................................................................................. D-3
Table E-1  Programmer Response Codes .......................................................................... E-3
Table F-1  ASCII Security Characters .............................................................................. F-3
Overview
Chapter 1

Overview

The AP4110 tag programmer allows you to enter data into the nonvolatile storage of an Amtech tag for use in automatic equipment identification (AEI), automatic vehicle identification (AVI), or electronic toll and traffic management (ETTM) applications. TransCore/Amtech readers can then read and verify this data, using radio frequency (RF) signals. Appendix A contains the AP4110 tag programmer specifications.

The AP4110 tag programmer, called the programmer, accepts simple RS–232 commands from a host system such as an IBM-compatible personal computer (PC) or from a communications terminal. You can use the programmer to perform tag programming and tag data verification functions.

The programmer operates either in a factory environment with AC power or in the field with external battery power.

About this Document

The AP4110 Tag Programmer User Guide is designed to be used by technical personnel operating AP4110 hardware and related software.

Each AP4110 order includes this manual along with TransCore’s Tag Programming (TP) software and the Tag Programming Software User Guide. The Tag Programming Software User Guide provides instructions for connecting equipment, installing TransCore’s TP software, and programming and verifying tag codes using the TP software.

This AP4110 Tag Programmer User Guide discusses the proper setup of the AP4110 tag programming system including power considerations, indoor and outdoor environmental factors, tag programming head setup, and the host system interface.

This user guide provides instructions on tag programming and tag verification sequences, as well as maintenance and troubleshooting tips. As a reference, Chapter 3, “Operating the Programmer” includes a list of the RS–232 commands recognized by the programmer.

The appendices in this manual contain additional technical reference information, such as programmer specifications, response codes used by the programmer to identify system problems, and descriptions of the data formats used by the programmer.
2

Setting up the Programmer
Chapter 2

Setting up the Programmer

The programmer is shipped with the items listed below. When unpacking your order, make sure all of the following items are included.

- AP4110 tag programmer, with a tag programming head kit. Each kit includes a base plate, a pin block, an interlock switch, and a tag positioning bracket.
- Tag Programming software diskette
- Tag Programming Software User Guide
- Tag programming head
- Power transformer and cable to convert AC voltage to 12 V DC
- Ground wire for programming unit
- Static-suppression wrist strap
- Programmer-to-PC RS–232 connecting cable
- DB9 null modem

If any of the above items are missing from your order, contact your TransCore distributor.

Either a host system — such as a PC — or a communications terminal is required to operate the programmer. TransCore does not supply this equipment for the AP4110 programmer.

TransCore strongly recommends that you use specially-designed plugs and O-rings to seal each tag after you have programmed it. Plugs and O-rings are available at extra cost. See page 8 for more information.

Optional Accessories

The optional accessory items listed below are available at extra cost. Specific programming applications may require one or more of these accessories; however, none is necessary for the basic tag programming sequence described in this manual. These items may be shipped with the programmer but do not require configuration at the factory before shipment.

- Plug extraction kit (model AS8003)
- Plug extraction kit for molded extraction hole plugs (model AS8004)
Plug insertion tool for transportation tags (model AS8010)

- Automobile power adapter cord kit to connect the programmer and, if desired, a notebook or laptop PC to the vehicle 12 V DC power, using the cigarette lighter socket

- Modem for operating the programmer with a remote host system. TransCore recommends the MultiTech® model 224.

- Interchangeable tag programming head kit

Additional programming heads kits allow you to program different types of tags that require different programming pin terminations without buying a separate programmer for each type of tag. For example, with the correct programming head kits you could program both TollTags® and license plate tags with only one programmer. Table 2-1 lists each programming head kit and the types of tags with which each kit can be used.

### Table 2-1 AP4110 Tag Programming Head Kit Options

<table>
<thead>
<tr>
<th>Assembly Number</th>
<th>Final Assy.</th>
<th>Head Type</th>
<th>Tag Models</th>
<th>Tag Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>05117-01</td>
<td>-01</td>
<td>A</td>
<td>AT5110, AT5111, AT5125, AT5530, AT5715</td>
<td>Beam Rail</td>
</tr>
<tr>
<td>05118-01</td>
<td>-02</td>
<td>B</td>
<td>AT5100, AT5540, AT5547</td>
<td>TollTags®</td>
</tr>
<tr>
<td>05119-01</td>
<td>-03</td>
<td>C</td>
<td>AT5531, AT5541, AT5542, AT5544, AT5545</td>
<td>Compact</td>
</tr>
<tr>
<td>05117-01</td>
<td>-04*</td>
<td>A</td>
<td>AT5110, AT5111, AT5125, AT5530, AT5541, AT5715</td>
<td>Rail and Compact</td>
</tr>
<tr>
<td>05094-01</td>
<td>-05</td>
<td>D</td>
<td>AT5110, AT5111, AT5125, AT5530, AT5541, AT5715</td>
<td>Rail and Compact</td>
</tr>
<tr>
<td>05147-01</td>
<td>-06</td>
<td>E</td>
<td>AT5140, AT5145</td>
<td>License Plate</td>
</tr>
<tr>
<td>05160-01</td>
<td>-07</td>
<td>F</td>
<td>AT5510, AT5515</td>
<td>Battery-powered Rail</td>
</tr>
</tbody>
</table>

* Final assembly -04 includes a type C head and a cigarette lighter adapter.

Contact your TransCore distributor for more information on tag programming head kits.

### Electrical Power

The programmer can be powered by either 110 V AC/12 V DC 50/60 Hz or 220 V AC/12 V DC 50/60 Hz power transformers, or by a car battery via the optional cigarette lighter adapter cable.
Any 12 V DC source must meet the electrical requirements stated in Appendix A, Programmer Specifications. The 12 V DC source must have a mating connector with the proper polarity of inner/outer conductors.

For 110 V operation, connect the 110 V AC/12 V DC power transformer to a 110 V 50/60 Hz wall outlet.

For 220 V operation, use the 220 V AC/12 V DC power transformer and connect to a 220 V 50/60 Hz outlet.

Preventing Electrostatic Discharge Damage

Caution

Amtech tags are sensitive to electrostatic discharge (ESD), and precautions are necessary to ensure proper tag programming operation. Use the grounded ESD static-suppression wrist strap whenever using the programmer. Additional protection from ESD is recommended where practicable.

Static is generated by friction. Some causes of static include the following:

- Shoes moving across a carpeted or plastic floor
- Hot air blowing into a room from a hot-air duct
- Rubbing tags together
- Rubbing tags across a table top
- Wearing certain types of clothing

Electrostatic discharge may cause significant damage to the tag and will adversely affect a tag’s operating performance. Typical symptoms of ESD include the following:

- Inability to program the tag
- Greatly reduced operating range
- Tag operating failure
Antistatic Workstation

In a factory or fixed indoor environment, position the programmer in an antistatic workstation. Well-designed workstations use a system of multiple precautions against ESD. Figure 2-1 shows an antistatic workstation.

Figure 2-1  Programming Workstation with ESD Controls

Contact TransCore at (214) 461-4031 if you have any questions concerning the suitability of your programming station or environment.

You can take several precautions to prevent ESD from affecting programming operations. The most effective precautions are listed below. The potential for ESD damage decreases when two or more precautions are used together.

Grounding Wire

Caution

You should always connect your programmer to earth ground before beginning programming operations.

The programmer has circuitry that redirects potential high charges on the tag programming head to the chassis ground of the unit, away from the critical signal processing elements. Use the green ground wire provided with your programmer to connect the ground of the programmer to earth ground.

Using the ESD Wrist Strap

Caution

You should always attach the ESD wrist strap before beginning programming operations in any setting.

1. Be sure the programmer is connected to an earth ground.

2. Connect the banana plug of the ESD wrist strap to the ESD jack on the front of the unit.
3. Attach the ESD strap to your wrist by adjusting the hook and loop material.

**Antistatic Mat**

The antistatic work mat on which your programmer sits should be connected to earth ground through the standard 1 Megohm resistance that should be provided with the mat. The mat should also be clean and free of any electrical path to direct earth ground.

**Ionized Air Blower**

You may choose to set up an ionized air blower at your workstation to provide additional ESD protection. Ionized air blowers neutralize tags from static build-up before programming.

**ESD-Safe Tags Container**

Keep tags in the original box on an antistatic mat or within an ESD-safe container until they are programmed. Grounded metallic containers are best for inhibiting ESD.

**Seals and Plugs for Tag Programming Ports**

Each unprogrammed tag has a removable ESD seal over its programming port. Leave the ESD seal over the tag’s programming port in place (Figure 2-2) until the tag is programmed.

![Electrostatic Discharge Seal](image)

**Figure 2-2 Tag Port with ESD Seal**

At additional cost, TransCore provides plugs and O-ring seals to protect programmed tags. Use one plug and one O-ring seal to cover the tag programming port immediately after programming. Do not leave programmed tags unplugged. Contact your TransCore distributor to order the plugs and O-ring seals.
Preventing Condensation

High humidity environments (greater than 85% humidity) can affect tag programming adversely. Condensation may occur on the tags’ programming pads, preventing successful programming and reading. If tags have been exposed to such an environment, remove the tags from this environment and allow them to dry prior to programming.

Tag Programming Head

The programmer is shipped with a programming head, which is a base plate installed on top of the programmer. The programming head features a positioning bracket to guide the tag into the correct position over spring-loaded contact pins. When the tag is pressed down onto the head, the pins make contact with the tag’s internal printed circuit board.

Each tag programming head incorporates an interlock switch. This switch closes when the tag is adequately seated to ensure good physical contact between the spring-loaded contact pins and the tag circuitry. This switch closure activates the programmer’s tag processor.

The programming head’s contact pins in the pin block are grooved to ensure reliable connection to the pads on the tag circuit board. These pins require regular cleaning. Refer to Chapter 5, “Maintenance” for the recommended cleaning procedure.

Positioning Brackets

Positioning brackets, Figure 2-3, are available for large and small transportation tags, including the TollTag®. For some applications, interchangeable positioning brackets
may be available for programming tags of different sizes on a single large-format tag programming head.

![Large Format Programming Head](image)

![Small Format Programming Head](image)

**Figure 2-3 Tag Positioning Brackets**

Consult your TransCore distributor for more information regarding these positioning brackets. If you have not already received the positioning bracket you need, contact your TransCore distributor.

Ensure that the desired tag positioning bracket is installed on the programmer. If the bracket is not installed, refer to the following section of this document, Removing/Installing the Tag Positioning Bracket.

**Removing/Installing the Tag Positioning Bracket**

Depending upon the screws used to install the head assembly, you will need either a 5/64 in. Allen wrench or a Phillips-head screwdriver to remove or install the positioning bracket.

1. Remove the screws securing the tag positioning bracket.
2. Gently lift the bracket off the unit and set the bracket aside.
3. Position the new bracket on top of the unit.
4. Secure the new bracket to the unit using the original screws.
Connecting the Host Device

The programmer requires connection to a host device — a host computer or communications terminal — for normal operation. Figure 2-4 illustrates the connection from the programmer to the host.

Figure 2-4  Serial Interface Connection to Host Computer

The programmer is configured as a data terminal equipment (DTE) device and may require a null modem connected between the host and the programmer for operating communications. If the host system is a PC or a communications terminal, then a null modem will be required. The null modem serves as an adapter, allowing direct connection between DTE devices by emulating the physical connections of a data communications equipment (DCE) device. TransCore provides a DB9 null modem for this purpose.

Refer to the Interfacing the Programmer With the Host Using a Modem section of this chapter for information on connecting a modem for remote host operations.

The serial connection between the programmer and the host system must meet standard Electronics Industries Association (EIA) RS–232 specifications. Refer to Appendix B for specifications of the programmer’s RS–232 pinouts.

To connect the programmer to the host computer, follow the steps below:

1. Connect the RS–232 cable from the unit to the host computer system, using the null modem if the host system is an IBM-compatible or a conventional terminal.
2. Switch on power to the programmer. You should hear three beeps.
3. The programmer’s data terminal ready (DTR) line will be asserted when the unit has successfully completed its power-on self-test and configuration. At this time, if the data set ready (DSR) is asserted, either from the host or modem, the programmer will assume a full RS–232 interface and will perform hardware
handshaking on all serial communications. If you do not hear the programmer beep or if the host system cannot communicate with the programmer, refer to Chapter 4, “Troubleshooting.”

**Interfacing the Programmer With the Host Using a Modem**

The programmer can be connected to a modem and operated from a remote host system. Configure the modem in stand-alone mode to auto-answer the host system and then to allow direct communications through to the programmer. Connected to the modem, the programmer detects the DSR line and performs hardware flow control with the modem. The programmer-to-modem connection runs at a rate of 9600 baud. The modem-to-host link over the telephone lines runs at the baud rate used by the host system, typically 1200 or 2400 baud.

To set up the modem, follow the steps below.

1. Configure the modem with a terminal using the parameters shown in Table 2-2.
2. Connect the modem to the programmer using the RS–232 cable included with your programmer.
3. Power on the programmer and the modem.

When the host system dials into the modem, the modem should auto-answer and provide a direct connection to the programmer.

*Note:* These parameters are representative for the MultiTech® modem model 224. TransCore recommends the 224 model for use with the AP4110 programmer. Contact your TransCore distributor for information on ordering this modem.

**Table 2-2  Modem Setup Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setup Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore factory defaults</td>
<td>AT&amp;F&amp;W</td>
</tr>
<tr>
<td>Baud rate adjust off</td>
<td>AT&amp;BA0</td>
</tr>
<tr>
<td>Modem reset on DTR</td>
<td>AT&amp;D3</td>
</tr>
<tr>
<td>No command echo</td>
<td>ATE0</td>
</tr>
<tr>
<td>Modem hard flow on</td>
<td>AT&amp;E11</td>
</tr>
<tr>
<td>Modem to DTE baud</td>
<td>AT$SB9600</td>
</tr>
<tr>
<td>Suppress result code</td>
<td>ATQ1</td>
</tr>
<tr>
<td>Answer 1 ring</td>
<td>ATS0=1</td>
</tr>
<tr>
<td>Save as default</td>
<td>AT&amp;W</td>
</tr>
</tbody>
</table>

If communications between the programmer and host are not established, verify that the modem is configured correctly, and check all cable connections.
If the programmer still is unable to communicate with the host, connect the programmer directly to the host using the RS–232 cable. With Amtech Tag Programming software running, the host system will be able to identify any problems with the programmer or system configuration.

For information on loading Tag Programming software and on entering the commands used to configure the host, refer to Chapter 3, “Operating the Programmer.”
Operating the Programmer
This chapter describes how to program user data into a tag and how to verify or read back the user data; it also lists the RS–232 command set and describes the security character operation.

Before programming, the programmer system must be set up as specified in Chapter 2, “Setting Up the Programmer.” The programmer must be connected either to a host system running Tag Programming (TP) software or to a communications terminal. You will need to manually enter the RS–232 control commands if you are using a communications terminal.

To load and use the TP software for programming tags, refer to the Tag Programming Software User Guide provided with the programmer.

To operate the programmer using a communications terminal or a personal computer that does not use TP software, follow the steps in the Tag Programming Sequence and/or Tag Verification Sequence sections of this chapter, as appropriate to your task. Use the programmer commands listed in the RS–232 Command Set in this chapter.

The programmer can be powered by either 110 V AC/12 V DC 50/60 Hz or 220 V AC/12 V DC 50/60 Hz power transformers, or by a car battery via the optional cigarette lighter adapter cable. Refer to Chapter 2, “Setting Up the Programmer,” or to Appendix A, Programmer Specifications, for additional information.
Indicator Lights

Figure 3-1 illustrates the indicator lights used by the programmer during operation.

![Indicator Lights](image)

Figure 3-1 AP4110 Programmer Indicator Lights

- **PROGRAM**: The tag is being programmed.
- **VERIFY**: The programmer has read and verified the tag identification (ID) code.
- **ERROR**: An error has occurred while programming the current tag.
- **POWER**: The programmer is receiving power.
- **READY**: The programmer is ready to program tags.
- **LOW BATTERY**: Indicates the external power source is lower than 10 V. When the external power supply is low, the programming operation is not guaranteed to be reliable.

Sign-On

You should expect to see the following sign-on message on the screen of the host system or a communications terminal upon power-up or after you send the programmer command #30:

```
#Model AP4110 Ver X.XX
#Copyright 1991 TransCore
```

where X.XX is the current software version.
**Command Execution Protocol**

The programmer sends an appropriate acknowledgment to the host device upon the successful or unsuccessful execution of any command received.

All programmer commands are in the following format:

```
#abcxxxxx...xxxxx<cr>
```

where
- `#` start of command character. All commands must start with this character.
- `abc` the command
- `xxx...` user data
- `<cr>` carriage return

If command execution is successful, the programmer sends the response

```
#Done<cr>
```

If command execution is unsuccessful, the programmer sends the response

```
#Error<cr>
```

**RS–232 Command Set**

Programmer commands are listed below. Use these commands following the format described in the Command Execution Protocol section above.

---

**Enable Tone Generator (#120<cr>)**

The default state of the audible tone generator is tone enabled. The tone generator creates audible responses described as beep, triple beep, and solid tone. This command re-enables the programmer's audible response capability after it has been disabled by the #121 command.

Response:

```
#Done<cr>
```

---

**Disable Tone Generator (#121<cr>)**

This command disables the programmer's audible response capability. The default state of the audible tone generator is tone enabled.

Response:

```
#Done<cr>
```
Program Tag Using TransCore 6-Bit ASCII Format (#200xxx...<cr>)

The programmer programs the user’s data (xxx...) into the tag using the TransCore 6-bit ASCII format. This format requires 20 characters of data. Refer to Appendix C for a description of this format.

During execution of this command, a series of beeps indicates that the programmer is waiting for a tag to be placed in the tag positioning bracket. If you do not place the tag in the bracket within a 10-sec limit, the programmer will abort this command.

Response:

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxx<cr>
#Done<cr>
```

or

```
#EndError<cr>
```

Program Tag Using 4-Bit ASCII Hex Format (#201xxx...<cr>)

The programmer programs the user’s data (xxx...) into the tag using the 4-bit ASCII hexadecimal (hex) format. This format requires 30 characters of data. Refer to Appendix D for a description of this ASCII hex format.

During execution of this command, a series of beeps indicates that the programmer is waiting for a tag to be placed in the tag positioning bracket. If you do not place the tag in the bracket within a 10-sec limit, the programmer will abort this command.

Response:

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<cr>
#Done<cr>
```

or

```
#EndError<cr>
```
Read/Verify Tag Data Using TransCore 6-Bit ASCII Format (#210<cr>)
The programmer reads the data from the tag, puts it into the TransCore 6-bit ASCII format, and sends it to the host device. Refer to Appendix C for a description of this 6-bit ASCII format.

During execution of this command, a series of beeps indicates that the programmer is waiting for a tag to be placed in the tag positioning bracket. If you do not place the tag in the bracket within a 10-sec limit, the programmer will abort this command.

Response:

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxx<cr>
#Done<cr>
```
or

```
#Error<cr>
```

Read/Verify Tag Data Using 4-Bit ASCII Hex Format (#211<cr>)
The programmer reads the data from the tag, puts it into the 4-bit ASCII hex format, and sends it to the host device. Refer to Appendix D for this ASCII hex format.

During execution of this command, a series of beeps indicates that the programmer is waiting for a tag to be placed in the tag positioning bracket. If you do not place the tag in the bracket within a 10-sec limit, the programmer will abort this command.

Response:

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<cr>
#Done<cr>
```
or

```
#Error<cr>
```

System Reset (#30<cr>)
The command resets the programmer to its original power-on state with the self-test.

Response (system sign-on message):

```
#Model AP4110 Ver X.XX
#Copyright 1991 TransCore
```

where X.XX is the current software version

or

```
#Error<cr>
```
Display System Status (#40<cr>)
This command requests a 2-digit status response code from the programmer indicating the status of the last operation performed. Refer to Appendix E for a description of the programmer response codes that refer to system status.
Response:

#xx<cr>

where XX is the 2-digit response code. The code 00 indicates that the programmer performed the last operation without error.

Display Tag Interlock Switch Status (#42<cr>)
This command requests a 2-digit status response from the programmer indicating the status of its tag interlock switch. It is designed to be used during remote programming activities when the host computer is not at the site where tags are being programmed.
The response notifies personnel at the host computer site that a tag is present on the programming head, so that tag data can be sent.
Response:

#01<cr>switch depressed
#00<cr>switch not depressed

Programming The Tag

To program a tag, follow the steps below:

1. Send the programmer the appropriate #200 <cr> or #201<cr> command along with the user data you want to program onto the tag. Use command #200 if you are using the TransCore 6-bit format for your data; use command #201 for the 4-bit hex data format.

2. When the programmer begins its 10-sec timer, indicated by an audible beep once every second, place the tag you want to program into the tag positioning bracket and press the tag firmly into place.

   If the tag is placed within the 10-sec period, the programmer programs the user data, verifies the user data, and sends a response to the host system.

   If the tag is not placed within the 10-sec period, the programmer aborts the command, sends the #Error<cr> response to the host, and emits a solid tone.

When the tag is successfully programmed, the programmer responds with a triple beep and with the following response:

   xxx...<cr>
   #Done<cr>

where XXX... is the tag data read from the tag and encoded in the specified format.
If the tag programming process fails, the programmer responds with a solid tone and with 
#Error<cr>.

If you receive this response, begin again at step 1 of this sequence.

**Verifying Tag Data**

To verify the data on a tag, follow these steps:

1. Send the programmer the appropriate #210<cr> or #211<cr> command. Use command #210 if you are using the TransCore 6-bit format for your data; use command #211 for the 4-bit hex data format.

2. When the programmer begins its 10-sec timer indicated by an audible beep once every second, place the proper tag into the tag positioning bracket and press down.
   
   If you place the tag down within the 10-sec period, the programmer reads the tag data, checks for data integrity, and sends the #Done response to the host system.
   
   If you do not place the tag down within the 10-sec period, the programmer aborts the command and sends the #Error<cr> response to the host.

If the verification process is not successful, begin again at step 1.

**Security Characters**

The tag security character function prevents the use of unauthorized tags in any TransCore electronic identification system. All AP4110 tag programmers have the capability, when enabled, to program specific, predefined security characters into a tag. This capability is enabled at the factory if it is specified in an authorized order. Security characters are available for American Trucking Associations (ATA) and International Standards Organization (ISO) tag data formats.

**Note:** The security character function may only be used when you program your tag data in the 4-bit hex format using command #201. The security character function may NOT be used if you program your tag data with the 6-bit ASCII format using command #200, even though the security characters are 6-bit characters.

When enabled, the security character function stores two characters, specific 6-bit patterns designated for security coding purposes, in the programmer's nonvolatile memory. These characters, selected at the factory for each programmer, may be two specific characters from the list of security characters in Appendix F or one character each from Appendix F and Appendix C.

When a tag programmer has tag security enabled, all tags programmed contain the assigned security characters in the designated positions of the tag data. Security characters comprise the second half of the 26th character through the first half of the 29th character when data is in the 4-bit hex format.
The programmer places security characters into their associated tag data locations when security characters in the tag data entered with command #201 match the internal security characters stored in the programmer by factory personnel.

The programmer will not store onto a tag security codes that differ from those stored in the programmer’s internal memory. If you attempt to program the tag with security characters that do not match the internal security characters of the programmer (i.e., incorrect or improperly obtained codes), the programmer automatically programs the space character into both security character locations.

When you input security characters while using TransCore’s Tag Programming software, those security characters will not be visible on the monitor.
Troubleshooting
Listed below are some problems that may arise when you are programming tags at your site, along with suggestions for troubleshooting each problem. If a problem persists, contact your TransCore representative or call TransCore at (214) 461-4031.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Suggested Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System does not beep at power-on.</td>
<td>• Check power transformer.</td>
</tr>
<tr>
<td></td>
<td>• Verify battery connection.</td>
</tr>
<tr>
<td></td>
<td>• Replace the internal 2AG 250mA fuse.</td>
</tr>
<tr>
<td>System powers on, but does not sign on.</td>
<td>• Verify RS–232 connection at the programmer and at the host.</td>
</tr>
<tr>
<td></td>
<td>• Check null modem, if used.</td>
</tr>
<tr>
<td></td>
<td>• Verify that host baud rate is set to 9600.</td>
</tr>
<tr>
<td>Programmer beeper is not heard during tag programming/verify sequence, or system continually beeps until time-out.</td>
<td>• Verify that the programming head interlock switch is depressed.</td>
</tr>
<tr>
<td></td>
<td>• Verify that the programmer pin block is clean and is making contact with each tag's programming pad.</td>
</tr>
<tr>
<td>Tag programming sequence fails repeatedly.</td>
<td>• Refer to the programmer response codes list in Appendix E to identify the problem.</td>
</tr>
<tr>
<td></td>
<td>• Clean programmer head pin block.</td>
</tr>
<tr>
<td></td>
<td>• Improve ESD protection in programming area.</td>
</tr>
<tr>
<td>Requested “verify” data does not match programmed data.</td>
<td>• Check for security code violations.</td>
</tr>
<tr>
<td></td>
<td>• Check programmed data for accuracy.</td>
</tr>
<tr>
<td>System resets intermittently.</td>
<td>• Check for damage to DC power connection.</td>
</tr>
<tr>
<td></td>
<td>• Check for damaged reset switch on rear panel.</td>
</tr>
<tr>
<td></td>
<td>• Check for a loose AC power connection.</td>
</tr>
</tbody>
</table>
5

Maintenance
The programmer requires no calibration or maintenance other than periodic cleaning of the contact pins found in the pin block on top of the tag programming head. Figure 5-1 shows the pin block.

Figure 5-1 Pin Block on Tag Programming Head

These pins are grooved and can collect dust and other materials that may hinder the conduction of the programming signals to the tag. Clean and inspect the block pins for wear every 10,000 tag programming cycles.

TransCore recommends cleaning the programming pins with a cleaner/degreaser specifically designed for electronic equipment. These cleaners can be found at local electronics retailers.

*Note:* Be sure the cleaner is safe for use on plastic materials.
Leaving Tags on Programming Pins

Do not leave tags on top of the programming pins when the programmer is idle. The programming pins are spring-loaded to help ensure firm contact with the tag during programming. Continued depression of the pins over a period of time can reduce the tension in the springs.
Appendix A

Programmer Specifications
**FEATURES**

- PC interface
- User-friendly tag programming software
- Easy data transfer
- Tag data verification
- Protection from electrostatic discharge
- Interchangeable programming brackets available
- Adapter available for mobile operation

---

**AP4110 Programmer**

The AP4110 Programmer transfers 128-bit identification codes to field-programmable Amtech®-brand tags. Programming occurs using a PC interface that allows coding of 20 alphanumeric characters—including optional security characters—into the tag's data frame. The PC interface can create custom features such as automatic sequential numbering or barcode interface. The user or TransCore software development experts may develop software for this purpose.

Included with the AP4110 Programmer is Amtech-brand Tag Programming (TP) software for easy entry of data, creation and modification of data files, transfer of data to tags, verification of tag data, and output of tag information to label printer and tag files. TP software allows users to create table files to expedite data entry of codes for tags with common attributes. The TP system or other software systems may generate input data files.

Transfer of programmed data to the tag requires only a few simple steps. After entering the code, the user issues a command to initiate code transfer. The user presses the tag onto the programming head, and the code automatically transfers to the tag.

The AP4110 Programmer can verify the code programmed into any tag with an unsealed programming port. The TP software initiates the verification sequence and the user presses the tag onto the programming head. The tag ID then automatically appears on the PC screen.

The AP4110 Programmer comes with a programming bracket suitable for the type of tag specified. Additional easily-interchanged brackets are available. A 12V DC power cable with cigarette lighter adapter is available for mobile operation of the AP4110 Programmer.
**AP4110 Programmer**

### HARDWARE FEATURES

**PC-to-Programmer Cable**
RS-232 (with adapters)

**POWER REQUIREMENTS**

**Input Voltage**
110/220V AC
12V DC
(UL-1950 approved transformer supplied)

**Power Consumption**
<1.5W

### PHYSICAL

**Dimensions**
Size: 9.5 x 16.5 x 4.0 in (24.1 x 41.9 x 10.2 cm)
Weight: 7.0 lb (3.2 kg)

### ENVIRONMENTAL

**Operating Temperature**
+32°F to +158°F
(0°C to +70°C)

**Environment**
Dustproof

### STANDARDS

**ISO, AAR, ANSI, ATA and IATA Compatible Coding**
The AP4110 Programmer can code tag data formats compatible with RF identification standards developed by the International Organization for Standardization (ISO), the American National Standards Institute, the Association of American Railroads, the American Trucking Associations, and the International Air Transport Association.

### LICENSING

**Electromagnetic Compatibility**
The AP4110 Programmer complies with the limits established by Part 15 of the Federal Communications Commission (FCC) rules for a Class A digital device.
The AP4110 Programmer complies with all relevant European directives (CEMPRK).

### COMPLIANCE

**RF Interference**
Units have been tested and are verified to Part 15 of the FCC rules for a Class A digital device.

### OPTIONS

**Programming Bracket**
The AP4110 Programmer includes a tag programming bracket as specified by the customer. To order additional tag brackets, please specify tag type when ordering.

**Security Stamp**
The AP4110 Programmer includes a security stamp feature. With this feature, every tag programmed by the security-enabled AP4110 Programmer receives an automatically coded, unique security stamp. This feature may be employed in overall system security to help prevent the use of unauthorized tags.

### ACCESSORIES

**Plug Insertion Tool**
The AS8010 Plug Insertion Tool allows the user to seal multiple tags quickly and securely.

**Plug Removal Tool**
The AS8003 Plug Removal Tool kit allows the user to remove plugs for the reprogramming of transportation tags.

**Grounding Wrist Strap**
The AP4110 Programmer includes a grounding strap to wear when programming tags. The grounding strap prevents damage to tag circuits from electrostatic discharge.

**Carrying Case**
The AP4110 Programmer comes with a durable case for protection and easy carrying.

**Adapter for Mobile Operation**
A 12V DC power cable with cigarette lighter adapter enables mobile operation of the AP4110 Programmer.

### DOCUMENTATION

Tag Programming System Software User Guide
AP4110 Programmer’s Guide

---

**For product information call:** 1.800.923.4824 or 214.461.4031 (outside the U.S.) Fax 214.461.6478

www.transcore.com

© 2008 TC IP, Ltd. All rights reserved. TRANSCORE is a registered trademark of TC IP, Ltd., and is used under license. All other trademarks listed are the property of their respective owners. Contents subject to change. Printed in the U.S.A. Products covered by this document are protected by one or more of the following U.S. patents: 4,075,632; 4,360,810; 4,782,345; 4,835,377; 4,888,591; 5,504,485; and foreign equivalent patents. Other patents pending.

411114008 03/08
RS–232 Connector Pinout
Appendix B

RS–232 Connector Pinout

This table identifies the pinout of the programmer’s DB9 rear panel RS–232 connector.

*Table B-1 RS–232 Connector Pinout*

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCD</td>
</tr>
<tr>
<td>2</td>
<td>RX</td>
</tr>
<tr>
<td>3</td>
<td>TX</td>
</tr>
<tr>
<td>4</td>
<td>DTR</td>
</tr>
<tr>
<td>5</td>
<td>Signal ground</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
</tr>
<tr>
<td>9</td>
<td>No connection</td>
</tr>
</tbody>
</table>
C

TransCore 6-Bit ASCII Format
In TransCore’s 6-bit ASCII format, each ASCII data character is represented by a certain 6-bit pattern. These bit patterns are placed in specific groups of six consecutive bits within the tag. The ASCII characters and their corresponding 6-bit codes are listed in the table below. TransCore's 6-bit ASCII codes are produced by subtracting 00100000 from the character's 8-bit ASCII value.

**Table C-1  6-Bit ASCII Codes**

<table>
<thead>
<tr>
<th>Character</th>
<th>6-Bit Code</th>
<th>Character</th>
<th>6-Bit Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(space)</td>
<td>000000</td>
<td>G</td>
<td>100111</td>
</tr>
<tr>
<td>*</td>
<td>001010</td>
<td>H</td>
<td>101000</td>
</tr>
<tr>
<td>-</td>
<td>001101</td>
<td>I</td>
<td>101001</td>
</tr>
<tr>
<td>.</td>
<td>001110</td>
<td>J</td>
<td>101010</td>
</tr>
<tr>
<td>/</td>
<td>001111</td>
<td>K</td>
<td>101011</td>
</tr>
<tr>
<td>0</td>
<td>010000</td>
<td>L</td>
<td>101100</td>
</tr>
<tr>
<td>1</td>
<td>010001</td>
<td>M</td>
<td>101101</td>
</tr>
<tr>
<td>2</td>
<td>010010</td>
<td>N</td>
<td>101110</td>
</tr>
<tr>
<td>3</td>
<td>010011</td>
<td>O</td>
<td>101111</td>
</tr>
<tr>
<td>4</td>
<td>010100</td>
<td>P</td>
<td>110000</td>
</tr>
<tr>
<td>5</td>
<td>010101</td>
<td>Q</td>
<td>110001</td>
</tr>
<tr>
<td>6</td>
<td>010110</td>
<td>R</td>
<td>110010</td>
</tr>
<tr>
<td>7</td>
<td>010111</td>
<td>S</td>
<td>110011</td>
</tr>
<tr>
<td>8</td>
<td>011000</td>
<td>T</td>
<td>110100</td>
</tr>
<tr>
<td>9</td>
<td>011001</td>
<td>U</td>
<td>110101</td>
</tr>
<tr>
<td>A</td>
<td>100001</td>
<td>V</td>
<td>110110</td>
</tr>
<tr>
<td>B</td>
<td>100010</td>
<td>W</td>
<td>110111</td>
</tr>
<tr>
<td>C</td>
<td>100011</td>
<td>X</td>
<td>111000</td>
</tr>
<tr>
<td>D</td>
<td>100100</td>
<td>Y</td>
<td>111001</td>
</tr>
<tr>
<td>E</td>
<td>100101</td>
<td>Z</td>
<td>111010</td>
</tr>
<tr>
<td>F</td>
<td>100110</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4-Bit ASCII Hex Codes
Appendix D

4-Bit ASCII Hex Codes

In TransCore’s 4-bit hex format, each ASCII character is represented by a certain 4-bit pattern. These bit patterns are placed in specific groups of four consecutive bits within the tag. The ASCII characters and their corresponding 4-bit hex codes are listed in the table below.

*Table D-1  4-Bit ASCII Hex Codes*

<table>
<thead>
<tr>
<th>Character</th>
<th>4-bit Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
</tr>
<tr>
<td>A</td>
<td>1010</td>
</tr>
<tr>
<td>B</td>
<td>1011</td>
</tr>
<tr>
<td>C</td>
<td>1100</td>
</tr>
<tr>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>F</td>
<td>1111</td>
</tr>
</tbody>
</table>
Programmer Response Codes
The programmer returns one of these codes in response to command #40, Display System Status.

**Table E-1  Programmer Response Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Operation Successful/No Error: the last operation performed by the programmer was executed without errors.</td>
</tr>
<tr>
<td>01</td>
<td>Illegal Command: the command sent to the programmer was not a valid command.</td>
</tr>
<tr>
<td>02</td>
<td>Invalid Tag Data: the user data sent to the programmer was in the incorrect format (such as an illegal hex character).</td>
</tr>
<tr>
<td>03</td>
<td>Presence Time-out on Program Cycle: the programmer did not recognize the placement of a tag on the programming head during the 10-sec period prior to a programming sequence.</td>
</tr>
<tr>
<td>04</td>
<td>Presence Time-out on Verify Cycle: the programmer did not recognize the placement of a tag on the programming head during the 10-sec period prior to a verification sequence.</td>
</tr>
<tr>
<td>05, 06, 07</td>
<td>Reserved</td>
</tr>
<tr>
<td>08</td>
<td>Data Compare Error: during a programming sequence, the data read from the tag did not correspond to the user data that was written to the tag.</td>
</tr>
<tr>
<td>09</td>
<td>Locked Tag: the programmer has detected a lock condition of the tag. The tag data is stored permanently and cannot be changed.</td>
</tr>
<tr>
<td>0A</td>
<td>No Frame Marker: the programmer could not decipher a frame marker from the tag. Either the tag is not present, is damaged, or is not making contact with the programming head pins.</td>
</tr>
<tr>
<td>0B</td>
<td>Bad Tag Parity: the checksum bits read from the tag did not correspond to the checksum that the programmer calculated from the tag data.</td>
</tr>
</tbody>
</table>
F

Security Characters
You may enter security characters into an Amtech tag if your programmer has been configured at the factory to perform the security character function. These ASCII security characters are encoded in the Amtech 6-bit data encoding scheme, and are listed in the table below.

**Table F-1  ASCII Security Characters**

<table>
<thead>
<tr>
<th>Character</th>
<th>6-bit Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>000001</td>
</tr>
<tr>
<td>&quot;</td>
<td>000010</td>
</tr>
<tr>
<td>#</td>
<td>000011</td>
</tr>
<tr>
<td>$</td>
<td>000100</td>
</tr>
<tr>
<td>%</td>
<td>000101</td>
</tr>
<tr>
<td>&amp;</td>
<td>000110</td>
</tr>
<tr>
<td>'</td>
<td>000111</td>
</tr>
<tr>
<td>(</td>
<td>001000</td>
</tr>
<tr>
<td>)</td>
<td>001001</td>
</tr>
<tr>
<td>+</td>
<td>001011</td>
</tr>
<tr>
<td>,</td>
<td>001100</td>
</tr>
<tr>
<td>:</td>
<td>010101</td>
</tr>
<tr>
<td>;</td>
<td>010110</td>
</tr>
<tr>
<td>&lt;</td>
<td>011000</td>
</tr>
<tr>
<td>=</td>
<td>011001</td>
</tr>
<tr>
<td>&gt;</td>
<td>011010</td>
</tr>
<tr>
<td>?</td>
<td>011011</td>
</tr>
<tr>
<td>@</td>
<td>011100</td>
</tr>
<tr>
<td>[</td>
<td>110000</td>
</tr>
<tr>
<td>\</td>
<td>110101</td>
</tr>
<tr>
<td>]</td>
<td>111011</td>
</tr>
<tr>
<td>^</td>
<td>111100</td>
</tr>
<tr>
<td>–</td>
<td>111101</td>
</tr>
<tr>
<td>space</td>
<td>010000</td>
</tr>
</tbody>
</table>