Tag Programming Software
User Guide

TransCore
8600 Jefferson Street NE
Albuquerque, New Mexico 87113

January 2012

P/N 411174-006

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

Contents

5 Programming Tags

6 Using Log Files

7 Using Output Files

8 Using Input Files

9 Reading Pre-programmed Tags

A Record Formats for Tag Programming Input/Output Files

B Creating Tag Formats

C Use and Construction of Table Files

List of Figures

List of Tables

1

Introduction

Chapter 1
Introduction

Overview

This guide provides instructions for installing and running the TransCore tag programming (TP) software on a personal computer (PC), and coding and programming TransCore transportation tags using a TransCore tag programmer and a PC. The tag programming software system running on a personal computer supervises and controls the operation of TranScore tag programmers. This software automates the repetitive tag programming function, greatly simplifying the effort necessary for proper encoding of the formats associated with radio frequency identification (RFID) and automatic equipment identification (AEI) standards.

TP provides a screen-oriented, user-friendly interface, which uses simple commands and easy-to-understand prompts. Online help is provided at the touch of a key. Simple to install, the software operates on any IBM-compatible, MS-DOS personal computer.

The TransCore TP system capabilities include converting equipment identification parameters into required formats for the adopted RFID and AEI standards, including the following:

- Association of American Railroads Standard for Automatic Equipment Identification
- International Standards Organization for Freight Containers - Automatic Identification 10374.2
- American Trucking Associations Standard for Automatic Equipment Identification
- American National Standard for Freight Containers - Automatic Identification (ANSI MH5.1.901990)
- International Air Transportation Association Resolution 1640
- Downloading and verifying tag identification data into the tag programmer. TP provides audiovisual confirmation upon successful programming of a tag
- Printing labels for placement on each newly programmed tag
- Creating log files journaling tag programming activity
- Creating data files for subsequent input to TP by field personnel
- Accepting data files created by other software sources for input to TP by field personnel

This guide provides instructions for installing and running the TransCore TP software within the constraints of the tag programmer hardware. The tag programmer hardware does not allow certain security characters to be encoded into a tag unless the programmer has been set up for these security characters, as described in the user guide accompanying the tag programmer hardware.

Before You Use This Guide

Before you use this guide, you should already know how to use a PC and be familiar with:

- booting the PC,
- basic MS-DOS commands,
- using the hard disk drive or floppy disk drives, and
- copying files from diskettes.

This guide includes:

- hardware and software requirements,
- understanding the data entry screen,
- entering tag data,
- transferring data to tags,
- storing tag data to output files, modifying and deleting tag records,
- retrieving tag data from input files,
- creating log files of the programming session,
- printing log, output. and input files,
- importing tag data from other software sources,
- creating format files to customize the data entry interface, and

- using table files to enter significant numbers of equipment having the same physical attributes.

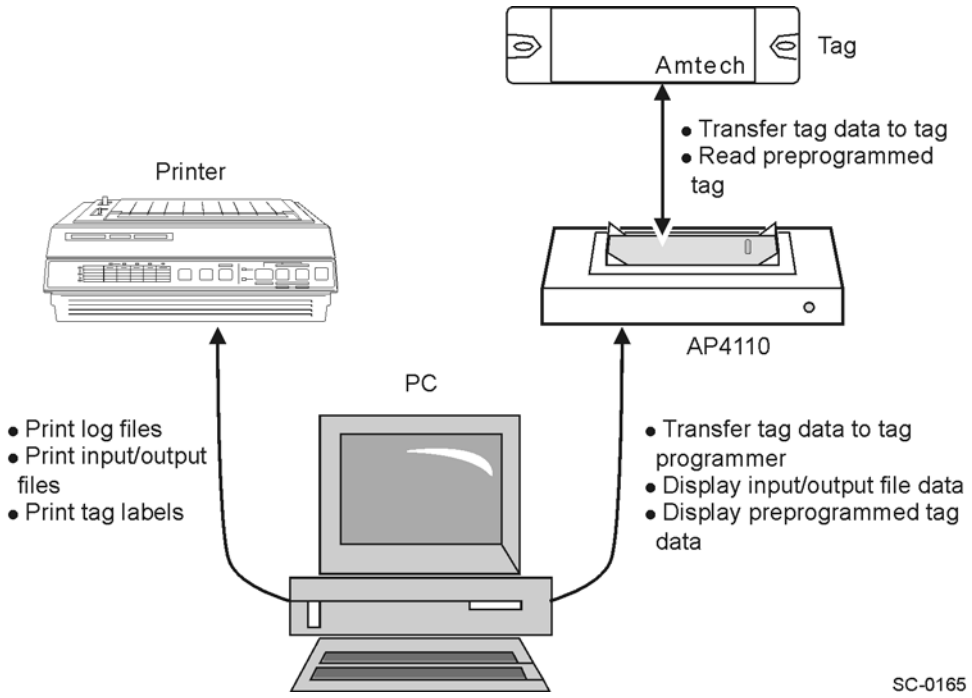

Figure 1-1 Functions of the Tag Programming System

Software Requirements

The tag programming software runs on a personal computer (PC) and requires a DOS operating system (version 3.2 or later) for operation.

The tag programming software is distributed via two diskette media formats:

- 360KB, 5.25-inch, double-sided/double-density, floppy disk, and
- 720KB, 3.5-inch, double-sided/double density, micro diskette.

The tag programmer must be equipped with firmware version 2.10 or higher for the tag programming software to function. Contact TransCore at 214-461-4031 regarding an upgrade if your tag programmer has an older version of the firmware installed. If the software was bundled with the tag programmer, the firmware version is correct.

Hardware Requirements

The tag programming software is designed to run on a variety of personal computer models, including the IBM PC, PC XT, and PC AT, as well as any 100% IBM-compatible personal computer having a minimum of 512KB of memory.

A PC monitor and associated video controller are also required. The tag programming software supports both monochrome and color monitors.

PCs with a hard disk drive need only one floppy disk drive. PCs without a hard disk drive must have two floppy disk drives.

Note: *Hard disk drives are recommended to enhance operation speed.*

Configure the serial RS-232 personal computer communications port as COMl:.

You also need a null model cable with a DB25 (for programmer model 4101) or DB9 (for model 4110) male connector on one end and a PC-compatible connector on the other end.

Optional Hardware

The PC requires additional hardware to print tag labels, tag log files, input and output files, or the tag file report:

- printer,
- printer port,
- printer cable, and
- standard 15/16 in. x 3 1/2 in. single-column label forms.

Note: *When printing labels, set the printer vertical spacing to 6 lines per inch.*

Typographical Conventions

The following conventions are used in this manual:

Table A-1 Typographical Conventions

Convention	Indication
	Concerns about a procedure.
`Code`	Code, including keywords and variables within text and as separate paragraphs, and user-defined program elements within text appear in courier typeface.
Dialog Box Title	Title of a dialog box as it appears on screen.
Function	Start with the characters G4, and are in mixed case with no underscores, and include parentheses after the name, as in G4FunctionName().
Menu Item	Appears on a menu. Capitalization follows the interface.
Note	Auxiliary information that further clarifies the current discussion. These important points require the user's attention. The paragraph is in italics and the word Note is bold.
NUL	Zero-value ASCII character or a zero-value byte.
NULL	Zero-value pointers. Null-terminated string refers to strings of printable ASCII characters with a zero-value byte placed in memory directly after the last printable character of the string.
	This procedure might cause harm to the equipment and/or the user.

2

Connecting the Programmer to the PC

Chapter 2

Connecting the Programmer to the PC

Connecting Programmer Model 4101

The tag programmer connects to the PC via a serial interface on the back panel of the programmer. The tag programmer model determines the type of connection used.

Programmer model 4101 (as seen in Figure 2-1) has an EIA RS-232 serial interface requiring a null modem cable with a DB25 male connector on one end and a PC-compatible connector on the other end. Connect the tag programmer to the COM1: port (the first serial communications port) on the PC using this cable.

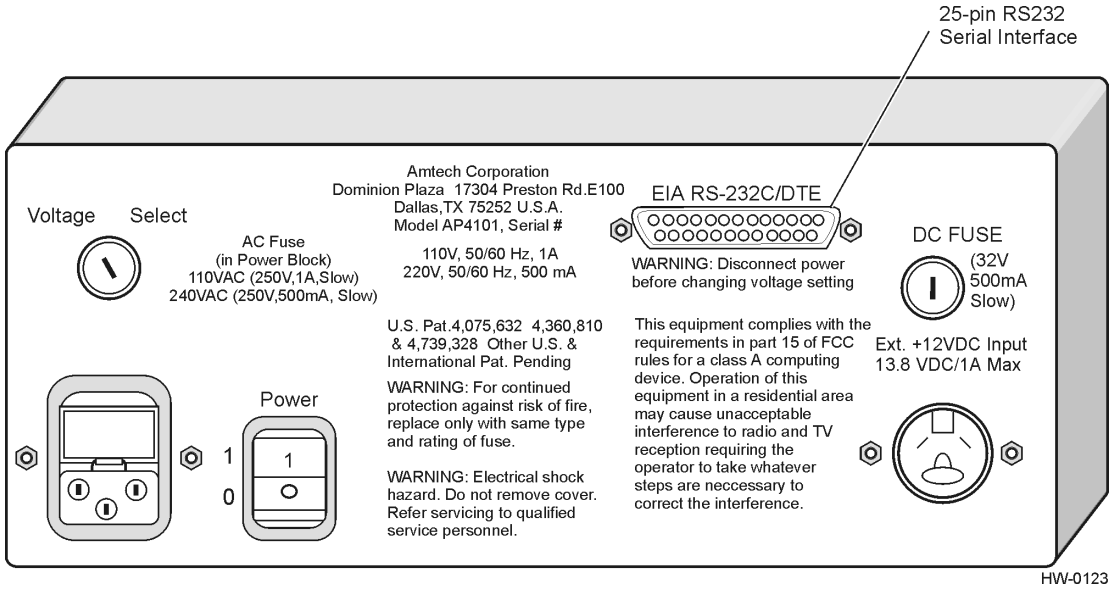

Figure 2-1 Sample View of Programmer Model 4101 Back Panel

Connecting Programmer Model 4110

Programmer model 4110 (as seen in Figure 2-2) has an EIA RS-232 serial interface requiring a null modem cable with a DB9 male connector on one end and a PC-compatible connector on the other end. Connect the tag programmer to the COM1: port (the first serial communications port) on the PC using this cable.

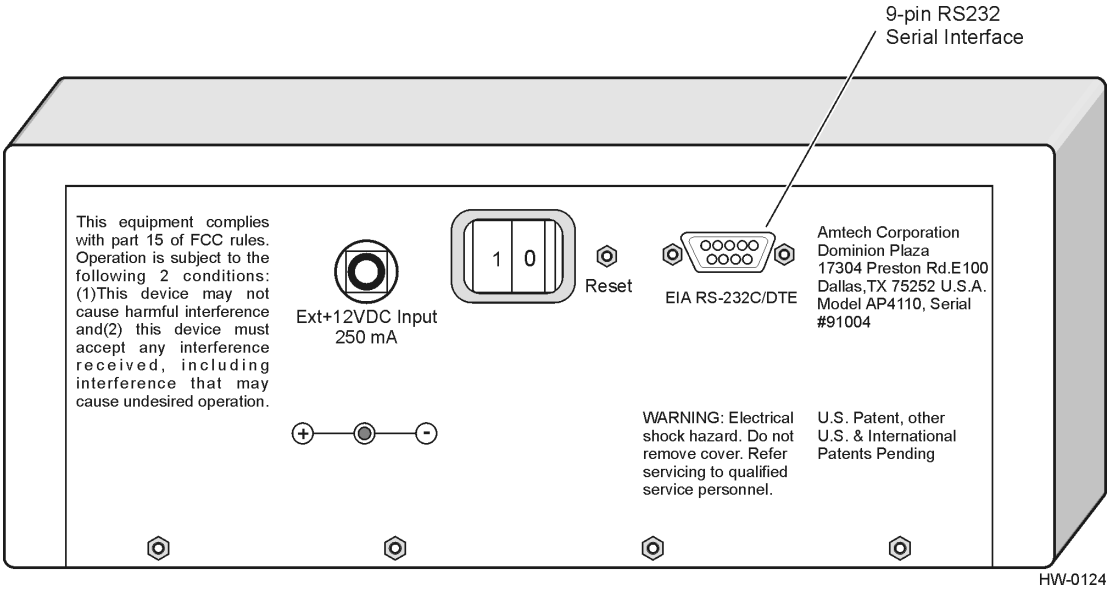

Figure 2-2 Sample View of the Programmer Model 4110 Back Panel

3

Installing and Starting the Tag Programming Software

Chapter 3

Installing and Starting the Tag Programming Software

Hard Disk Installation

Install the tag programming software on the PC after the hardware has been properly connected. The software may be installed on PCs with a hard disk and a floppy disk drive, or on PCs with two floppy disk drives.

To copy the tag programming software and associated files directly to the hard disk

1. Boot the PC.
2. Insert the tag programming software installation diskette into Drive A.
3. Type `A:INSTALL` at the system prompt
4. Press **Enter** to create a directory called TP on the C drive, under the C:/, or root directory.
5. Remove the installation diskette from Drive A.

Floppy Disk Installation

To copy the tag programming software and associated files using two floppy disk drives:

1. Insert the tag programming software installation diskette in Drive A.
2. Insert a blank diskette in Drive B.
3. Type `COPYA:*.* B:`
4. Press **Enter** to copy the contents of the installation diskette in Drive A to the blank diskette in Drive B.
5. Remove the installation diskette from Drive A.
6. Insert the new tag programming software diskette from Drive B into Drive A.
7. Type `DEL TP.BAT` and press **Enter** to delete the file `TP.BAT` from the diskette in Drive A.
8. Store the original software installation diskette in a safe place to be used as a backup.

Starting the Tag Programming Software

Once the tag programming software is installed on the PC, make sure the DOS system prompt displays on the terminal screen:

C:\ > (Hard Disk)

A:\ > (Floppy Disk)

Type TP and press **Enter** to start the tag programming software and display the data entry screen.

4

Understanding the Data Entry Screen

Chapter 4

Understanding the Data Entry Screen

Data Entry Screen

You use the data entry screen to enter information about the vehicle or object being tagged. The data entry screen also displays tag data from input files or from previously programmed tags.

The data entry screen is divided into five sections, or windows, as illustrated in Figure 4-1.

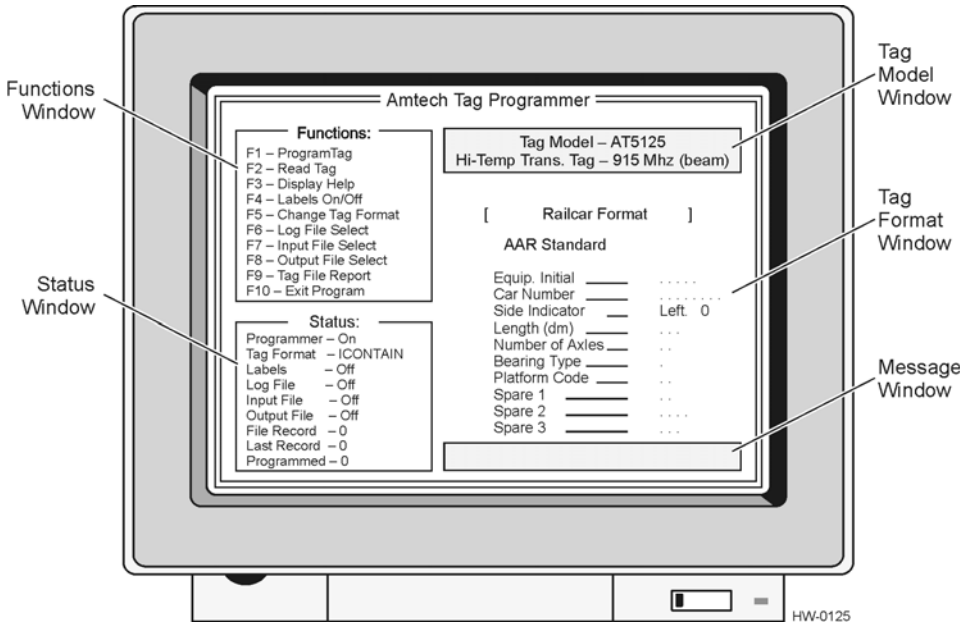

Figure 4-1 Tag Programmer Data Entry Screen

The functions, status, tag model, and message windows display useful information, You enter the tag data in the tag format window.

Functions Window

The **Functions** window, in the upper left quadrant of the data entry screen, lists special function keys and their associated uses. The tag programming software assigns specific tasks to function keys F1 through F10. Pressing the function key causes the corresponding action to occur. The function keys are discussed in Table 4-1.

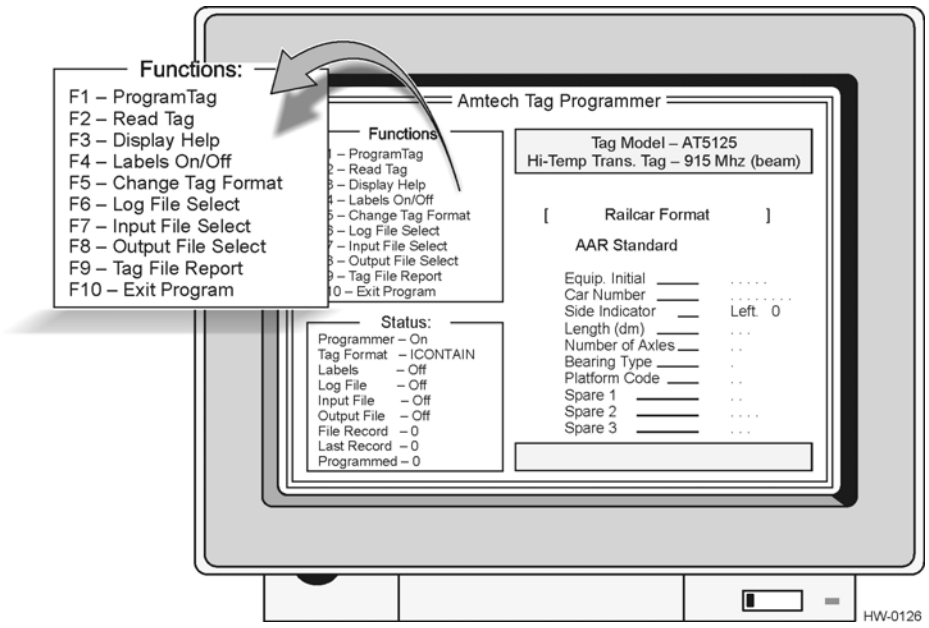

Figure 4-2 The Functions Window

Table 4-1 Function Key Explanations

Function Key	Function	Explanation
F1	Program tag	transfers tag data from the PC, to the tag programmer, to the tag positioned on the programmer. Alternately, if an output file has been designated, F1 sends the completed tag data to the output file so the tags can be programmed at a later time The Home key performs the same function as F1, and may be more convenient to use for certain operations because of its position on the keyboard.

Table 4-1 Function Key Explanations (continued)

Function Key	Function	Explanation
F2	Read tag	reads information from the tag positioned on the programmer and displays the tag data on the fields in the tag format window.
F3	Display Help	displays a help message with additional information about the function and currently active field. The message appears for four seconds at the bottom of the **Tag Format** window.
F4	Labels On/ Off	prints the data field information of the current programmed tag onto standard 15/16 in. x 3 1/2 inch, single-column labels. The label prints after the tag is successfully programmed. If you attempt to turn the Labels function on when no printer is attached to the PC, the program emits an error tone. ***Note:*** *When printing labels, set the printer vertical spacing to 6 lines per inch.*
F5	Change tag format	changes the tag format type. Use **PgUp** or **PgDn** to display the available format types in the message window until the desired format type appears, then press **Enter**. The title and fields in the tag format window display the new format.
F6	Log File Select	prompts for a file name in which to record the date, format type, and all data field information for each successfully programmed tag. Log File entries are made only after a tag is successfully programmed.
F7	Input File Select	prompts for the name of a file containing previously entered tag data and selects the file to program tags. Input files are created using the Output File Select function.
F8	Output File Select	prompts for the name of a file to receive tag data for storage. Output files store tag data, allowing tags to be programmed at a later time using the Input File Select function.
F9	Tag File Report	prompts for a tag file name, then prints a report containing the file name, date, time, record number, tag model, tag format, and tag contents.
F10	Exit Program	exits out of the tag programming software.

Status Window

The **Status** window, located in the lower left quadrant of the screen, displays the values or settings in effect for the current tag programming/data entry session. The status items and their settings are discussed in Table 4-2.

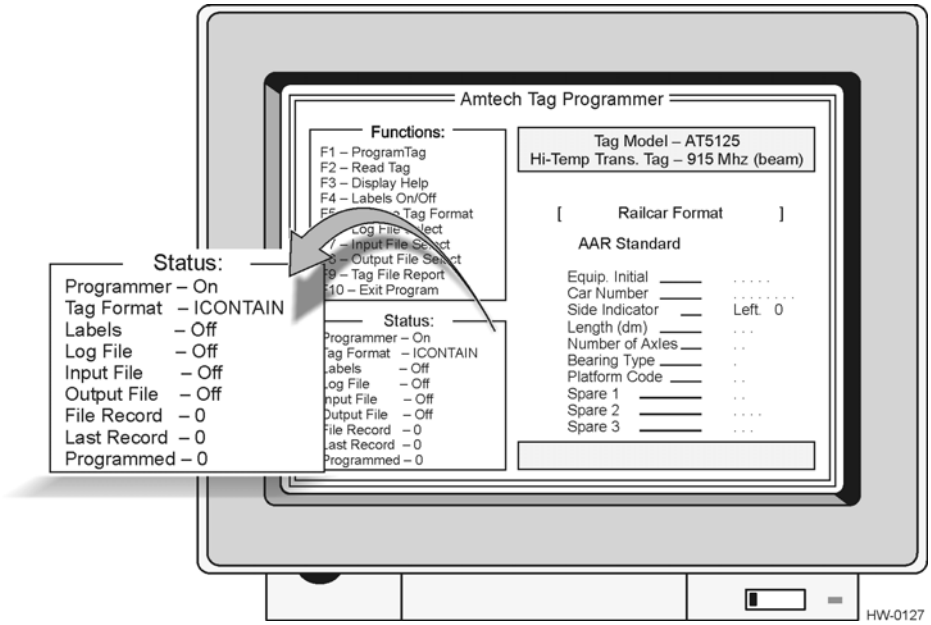

Figure 4-3 The Status Window

Table 4-2 Status Window Field Descriptions

Status Item	Explanation
Programmer	indicates whether a tag programmer is attached and active (ON).
Tag Format	displays the tag format type shown in the Tag Format window.
Labels	indicates whether or not to print the tag data on labels as tags are successfully programmed and verified. Use **F4** to turn label printing off or on.
Log File	displays the name of the log file receiving tag programming records. Use **F6** to select a log file.
Input File	displays the name of the file being used to program tags using previously entered tag data. Use **F7** to select an input file.

Table 4-2 Status Window Field Descriptions (continued)

Status Item	Explanation
Output File	displays the name of the file receiving tag data to be stored. Use **F8** to select an output file.
File Record	displays the number of the current record in an active or output file. Use **PgUp** to view the previous file record, and press **PgDn** to view the next file record in the Tag Format window.
Last Record	displays the number of the last record in an active input or output file.
Programmed	displays the total number of tags programmed during the current session. Exit from the program or press **Ctrl + Z** to reset this number to zero.

Tag Model Window

The Tag Model window, located in the upper right portion of the screen, displays information about the tag model being programmed, stored in an output file, or retrieved from an input file.

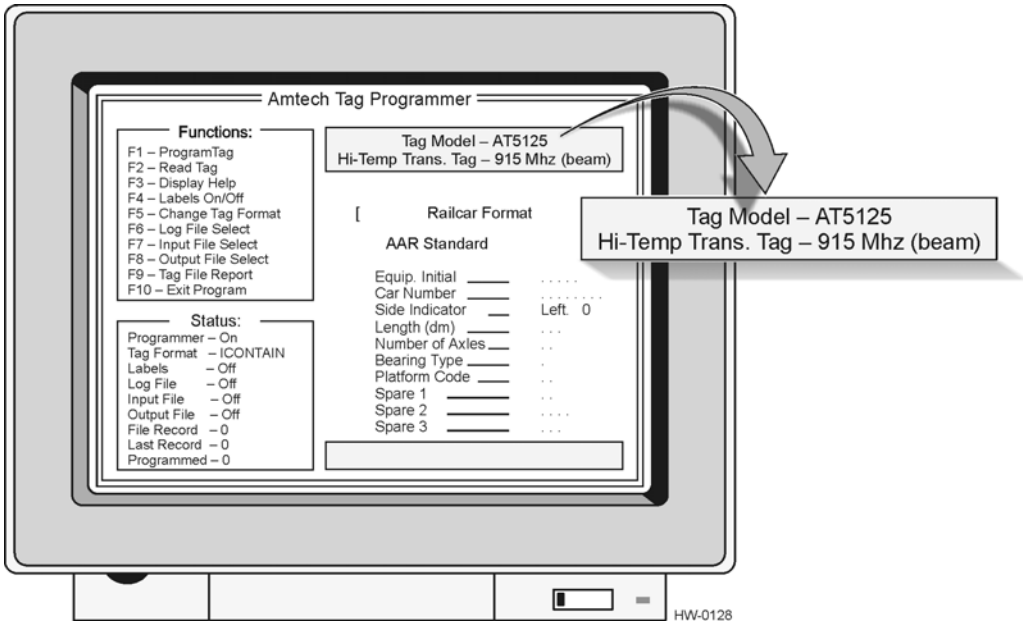

Figure 4-4 The Tag Model Window

Press **Ctrl + S** to select the appropriate tag model when entering tag data to be programmed or stored in an output file. If you enter a tag model, the tag model window displays the tag model and writes it to any selected output file. If you do not enter a

tag model, the tag window displays `Tag Model - UNKNOWN, UNKNOWN Tag Type`.

If you use an input file to program tags, the tag model window displays the tag model information stored in the input file. If the input file contains records with different tag model numbers, a unique tone sounds when the tag models change. This tone alerts you that the tag model of the next tag to program is different from the tag previously programmed.

There is currently no means of detecting the tag model number when reading a tag from the programmer. When reading a tag from the programmer, the tag model window always displays the message, `Tag Model - UNKNOWN, UNKNOWN Tag Type`.

Tag Format Window

The Tag Format window, in the right half of the data entry screen, contains the data entry fields used to program tags, enter data into output files, display data contained in input files, and display data contained in previously programmed tags.

The Tag Format window contains four main components:

- tag format type
- tag format standard status
- data entry field
- data entry field description

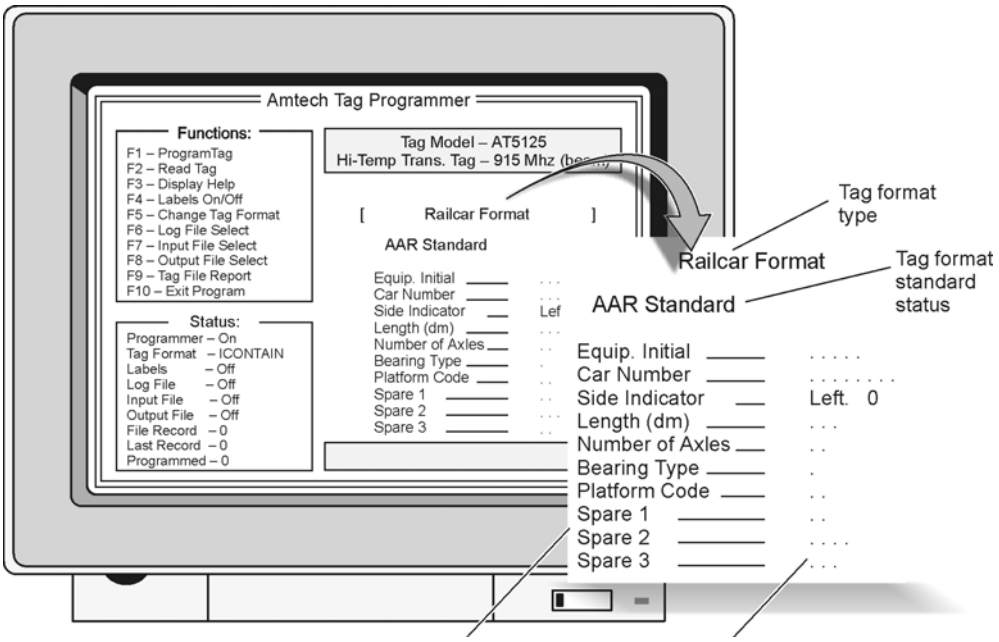

Figure 4-5 The Tag Format Window

Table 4-3 Tag Format Types

Tag Format Types	Explanation
Tag Format Type	is the type of vehicle or object to be tagged. The data entry field descriptions change based on the format selected. To select format type, press **F5**. Press **PgUp** or **PgDn** to display the available format types until the desired format type appears, then press **Enter**.
Tag Format Status	indicates the status of the tag format as a standard or proposed standard recommended by U.S. or international standards committees. The committee name and standard number are given.
Data Entry Field Description	describes the field for which values are to be entered. Data entry field descriptions vary based on the tag format chosen.
Data Entry Field	is the location where data is entered for each field item. The number of dots in this position represent the maximum number of characters that may be entered on that field. The dots are replaced with characters as they are typed in.

Message Window

The Message window is located in the lower right corner of the data entry screen. Unlike the other screen windows, the Message window is not displayed continuously. It is displayed only when there is a help message, field value range message, activity status message, or error message.

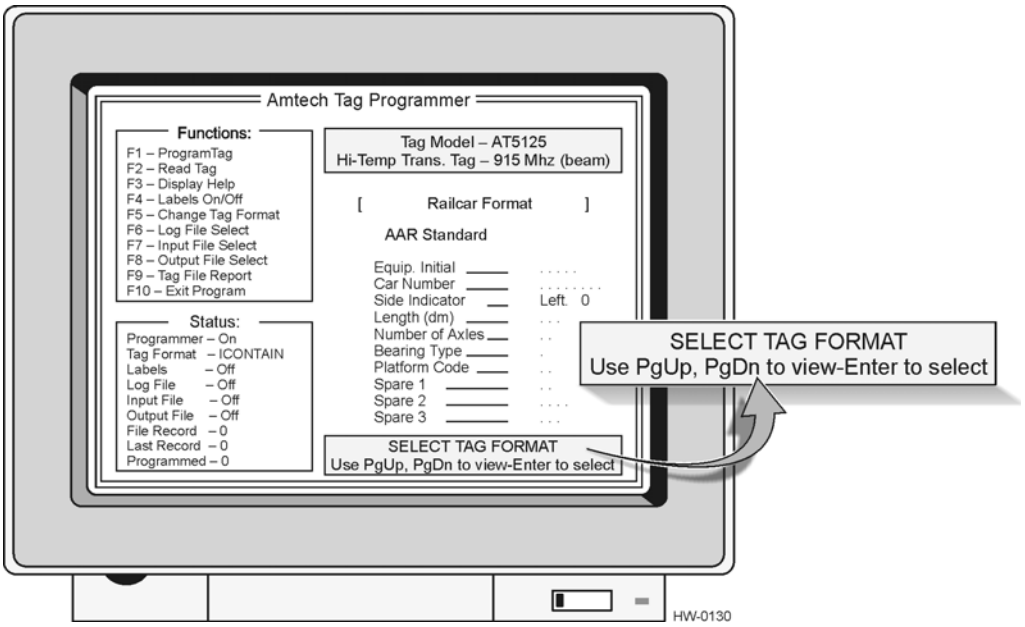

Figure 4-6 The Message Window

Data Entry Techniques

This section describes the conventions and methods that you must follow when entering tag data and file names.

Allowable Field Entries

The tag programming software contains numeric, alphabetic, alphanumeric, and table fields. A field entry inconsistent with the field type, or one outside the range of allowable values for the field, signals with an error tone. An appropriate help message for that field appears in the Message window.

To display the help information for an individual field, position the cursor on the field and press **F3**. The message appears for four seconds at the bottom of the Tag Format window.

Naming Files

Input and output file names may contain up to eight uppercase alphanumeric characters including the following symbols:

! # $ % ^ & ( ) _ -

Note: *Do not use blanks or spaces.*

Typing Numbers

The tag programmer system uses certain keys on the number keypad for cursor control purposes. The number keypad on PC models without a 101-key extended keyboard cannot be used to type numerical characters into data entry fields. Numerical data entry on these PCs is restricted to the number keys at the top of the computer's typewriter keyboard.

Key Functions

Some keys on the keyboard perform very specific functions, such as entering and deleting characters, moving the cursor between fields on the screen, and automatically scrolling field values up and down.

Keyboard Help

The keyboard help screen lists special keys and their functions.

To view the keyboard help screen shown in Figure 4-7, press **F3** twice**.** Pressing any other key while viewing the main help screen displays additional help information (see Table 4-4).

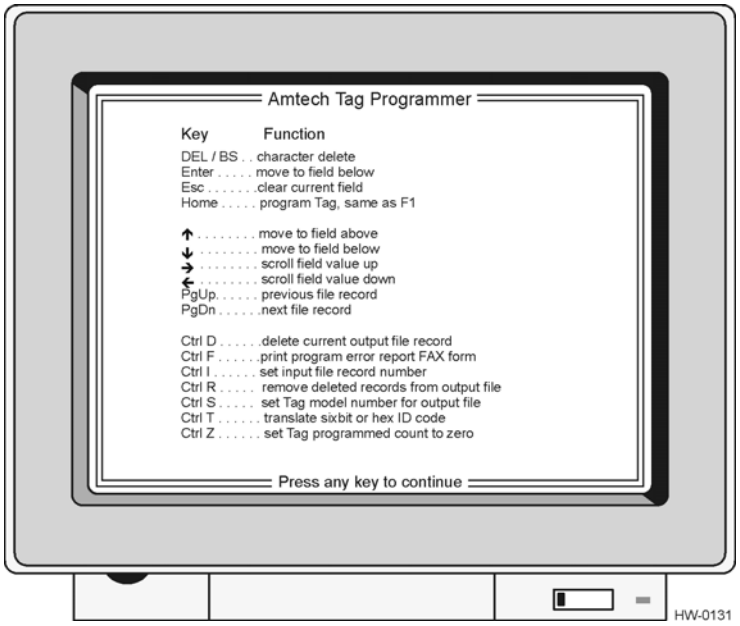

Figure 4-7 Sample Keyboard Help Screen

Table 4-4 Keyboard Help Functions

Keyboard Help Functions	Explanation
Deleting Characters	Position the cursor on the character and press **Delete** or **Backspace** to delete one character at a time. ***Note:*** *Individual characters can only be deleted from a field while the field information is being entered. If changes are required after initial data entry in the field, the entire field must be re-entered.*
Deleting Fields	Position the cursor on the field and press **Esc** to delete the entire field entry.
Deleting Output File Records	Select the record so that it displays on the screen. Select **Ctrl + D** to mark the record as deleted. If the output file is printed, the record is shown as marked for deletion.
Moving Between Fields	Press the **Up** arrow to move the cursor up to the previous field, and the **Down** arrow or **Enter** to move the cursor down to the next field.

Table 4-4 Keyboard Help Functions (continued)

Keyboard Help Functions	Explanation
Printing Error Report FAX Form	The error report fax form reports program errors or problems. 1. Remove labels and insert standard paper in the printer. 2. Press **Ctrl + F** to print out a FAX form. 3. Fill out the form as completely as possible. 4. Fax the completed form to the phone number shown on the form. (See sample Report Form in Figure 4-8).
Removing Deleted Output File Records	Press **Ctrl+R** to remove all deleted records from the current output file selected.
Scrolling Field Values Higher and Lower	Circular numerical tables exist that contain the full range of valid entries for each numerical field. When the cursor is positioned on a numerical field, pressing the right arrow scrolls the field values higher, and pressing the left arrow scrolls the field values lower. When the cursor is initially positioned on one of these fields, the right arrow displays the minimum value allowable for the field. Pressing the right arrow again increases the value. Alternately, pressing the left arrow displays the maximum value for the field and pressing the left arrow again decreases the value. This feature is designed to help minimize typographical errors, and data entry personnel do not have to memorize values for all fields for all format types.
Scrolling file Records	When using input or output files, press **PgUp** to view the previous file record, and press **PgDn** to view the next file record.
Selecting Input File Record Numbers	Press **Ctrl+I** to select the number of the record to display in the Tag Format window.
Setting Tag Model for Output File	Press **Ctrl+S** to select the tag model for tag records being entered in an output file. An output file can contain multiple tag models because the tag model is part of each individual tag record entered in the file.
Setting Tag Programmed Count to Zero	Press **Ctrl+Z** to set the tag programmed count field in the status window to zero.
Translating Six-bit or Hexadecimal ID Codes	Press **Ctrl+T** to decode tag IDs having 20 sic-bit or 30 hexadecimal codes, and translate them into the correct tag format.

Software Performance Report

To: TransCore
8600 Jefferson Street NE
Albuquerque, NM 87113

ATTN: Technical Support, Tag Programmer Software, Version 1.00f

FAX No: 505-857-0715

Date:

Your Name, Phone, Fax:_____
PC Manufacturer and Model: _____
DOS Version: _____
Can the program problem be reproduced at will? Yes / No

Carefully describe the nature of the program problem: _____
_____
_____
_____

Figure 4-8 Sample Error Report Fax Form

5

Programming Tags

Chapter 5

Programming Tags

Overview

This chapter covers the steps involved in programming a tag, including the following:

- Selecting the label print option to turn labels on or off
- Selecting a tag format type
- Entering the tag model number
- Entering the tag data in the PC
- Transferring tag data from the PC to the tag using the tag programmer

Setting the Label Print Option

If you want to print labels for each tag, press **F4** to set Labels **On** and insert standard 15/16 inch, single-column label forms in the printer. Turning labels **On** prints the data field information of the current programmed tag onto a label. The label prints after the tag is successfully programmed.

If you attempt to turn the **Labels** function **On** when no printer is attached to the PC, the program emits an error tone.

Note: *When printing labels, set the printer vertical spacing to 6 lines per inch.*

Selecting a Tag Format Type

The first step in programming a tag is to select a tag format. The tag programming software contains multiple predefined tag formats for each category of vehicle or object to be tagged. Multiple format types may be programmed in the same session. (Refer to the "Tag Format Window" section of Chapter 5.)

To select the tag format

1. Press **F5**.
2. Press **Page Up** or **Page Down** to display the available format types in the message window.
3. Press **Enter** to display the new format title at the top of the tag format window, and change the data entry fields accordingly.

Entering the Tag Model Number

You have the option of selecting and displaying the tag model number and description for each tag record entered. (Refer to “Tag Model Window” on page 4-7).

Note: *The tag model number information is entered into a tag’s output file record, but is not programmed into the tag.*

To enter the tag model number

1. Press **Ctrl** + **S** to display the message `Enter tag model number!`
2. Type in the model number (AT5541, AT5510, AT5125, or AT5510) and press **Enter** to display the tag model number and description in the Tag Model window.

If you do not enter a tag model, the Tag window displays the message, `Tag Model - UNKNOWN, UNKNOWN Tag Type`.

Entering Tag Data

Once you have chosen a tag format and entered the tag model, you then need to enter information about the vehicle or object to be tagged. Selecting a tag format displays field names describing the information required for that format in the tag format window. Following each field name is a dot or series of dots representing the number of characters to be typed in the field. Characters typed in the field overwrite the dots. (Refer to Chapter 5 for data entry techniques.)

End-of-Train Device Format Entries

Figure 5-1 illustrates a sample data entry screen for entering end-of-train device tag information.

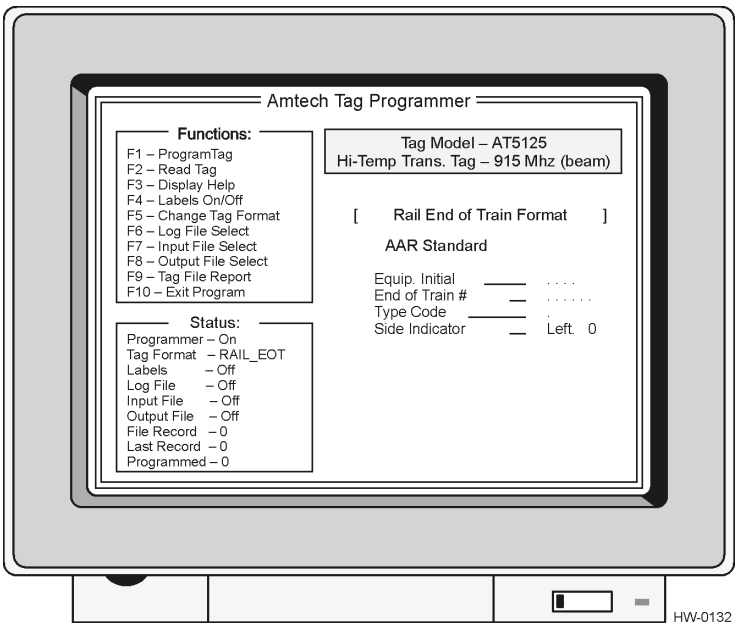

Figure 5-1 End-of-Train Device Data Entry Screen

The end of train device format field entries are described in Table 5-1.

Table 5-1 End of Train Device Format Field Entries

Field Entry	Description
Equip. Initial	Enter up to 4 alphanumeric characters from A to ZZZZ representing the car initial
End of Train Number	Enter up to 6 numbers from 0 to 999999 representing the end of train device number.
Type Code	Enter one of the numbers listed below to provide a generic description of the end of train device: 0 Active EOT, AAR-compatible 1 Active EOT, Not AAR-compatible 2 Passive EOT 3 Not used/unknown
Side Indicator	Enter a 1-digit code indicating whether the tag is to be installed on the left or right side of the end of train device. The right or left side is in reference to a person standing behind and facing an end-of-train device properly mounted on a train. 0 Left side 1 Right side

Locomotive Format Entries

This section provides examples the locomotive format entries.

Figure 5-2 illustrates a sample data entry screen for entering locomotive tag information, and Table 5-2 describes the locomotive format field entries.

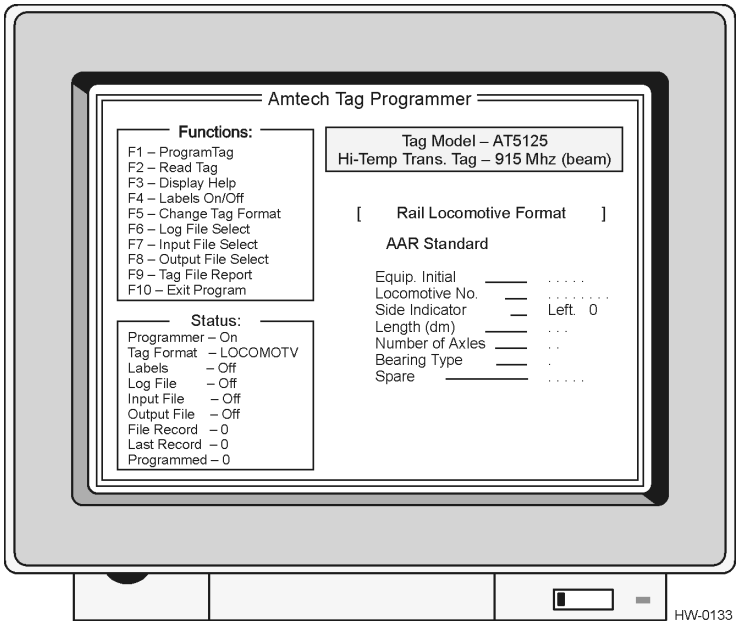

Figure 5-2 Locomotive Data Entry Screen

Table 5-2 Locomotive Format Field Entries

Field Entry	Description
Equip. Initial	Enter up to 4 alphabetic characters from A to ZZZZ representing the car initial.
Locomotive Number	Enter up to 6 numbers from 0 to 999999 representing the locomotive number.
Side Indicator	Enter a 1-digit code indicating whether the tag is to be installed on the left or right side of the rail car. The right or left side is in reference to the person facing the locomotive from the front end of the locomotive. 0 Left side 1 Right side
Length (dm)	Enter the length of the locomotive from coupler to coupler, or as specified by the UMLER specification manual, in feet and inches. The TP software automatically converts this entry into decimeters.
Number of Axles	Enter a number from 1 to 32 indicating the number of axles on the locomotive.

Table 5-2 Locomotive Format Field Entries (continued)

Field Entry	Description
Bearing Type	Enter a 1 for roller bearings (this holds true for 99% of cases). Other possible entries include: 0 Plain bearings 2 Roller bearings 3 Roller bearings, 3 axle truck, 1 axle obstructed ("Buckeye Design") 4 Roller bearings, plain bearing housing
Spare	This field is available for any use by the Owner. For example, it can be used to indicate the locomotive's model number. Up to 5 characters may be entered on this field.

Rail Car Format Entries

This section describes entering rail car tag information into the Data Entry screen.

Figure 5-3 illustrates a sample Data Entry screen for entering rail car tag information.

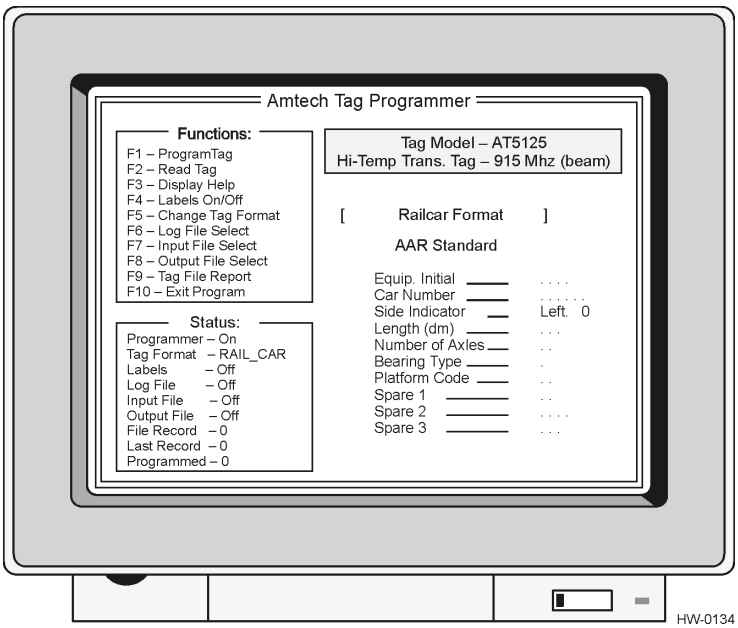

Figure 5-3 Rail Car Data Entry Screen

Table Table 5-3 describes the rail car format field entries.

Table 5-3 Rail Car Format Field Entries

Field Entry	Description
Equip. Initial	Enter up to 4 alphabetic characters from A to ZZZZ representing the car initial.
Car Number	Enter up to 6 numbers from 0 to 999999 representing the car number.
Side Indicator	Enter a 1-digit code indicating whether the tag is to be installed on the left or right side of the rail car. The left or right side is in reference to a person facing the car from the hand brake end (B end) of the car. 0 Left side 1 Right side

Table 5-3 Rail Car Format Field Entries (continued)

Field Entry	Description
Length (dm)	Enter the length of the car, in feet and inches from coupler to coupler, or as specified by the UMLER specification manual. The TP software automatically converts this entry into decimeters.
Number of Axles	Enter a number from 1 to 32 indicating the number of axles on the car.
Bearing Type	Enter a 1 for roller bearings (this holds true for 99% of the cases). Other possibilities include: 0 Plain bearings 2 Roller bearings 3 Roller bearings, 3 axle truck, 1 axle obstructed ("Buckeye Design") 4 Roller bearings, plain bearing housing
Platform Code	Enter a 0 for single platform (non-articulated) cars. Refer to the *AAR Standard for AEI* for articulated car entries.
Spare 1	This field is available for any use by the Owner or may be used to specify the first digit of the AAR Car Type Code.
Spare 2	This field is available for any use by the Owner or may be used to specify the last three digits of the AAR Car Type Code.
Spare 3	This field is available for any use by the Owner.

Transferring Data to Tags

Tags are programmed by transferring the tag data entered in the PC to the tag programmer. Tag data may be programmed into the tag immediately after completing data entry, or you may save the data to an output file and program the tag at a later rime (refer to the Chapter 7, "Using Output Files" and Chapter 8, "Using Input Files."

To transfer data to tags

1. Make sure the tag programmer is connected and active (the Programmer field in the Status window should be set to **On**).
2. Press **F1** or **Home** to transfer the tag data from the PC to the tag programmer and display the message `Place tag in programmer` accompanied by tone pulses one second apart.
3. Place a tag on the tag programming head, aligning the tag's programming port with the pins on the contact block.

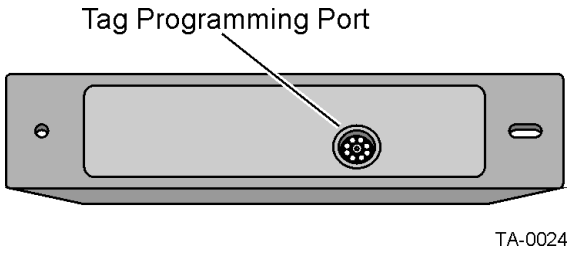

Figure 5-4 Tag to be Programmed

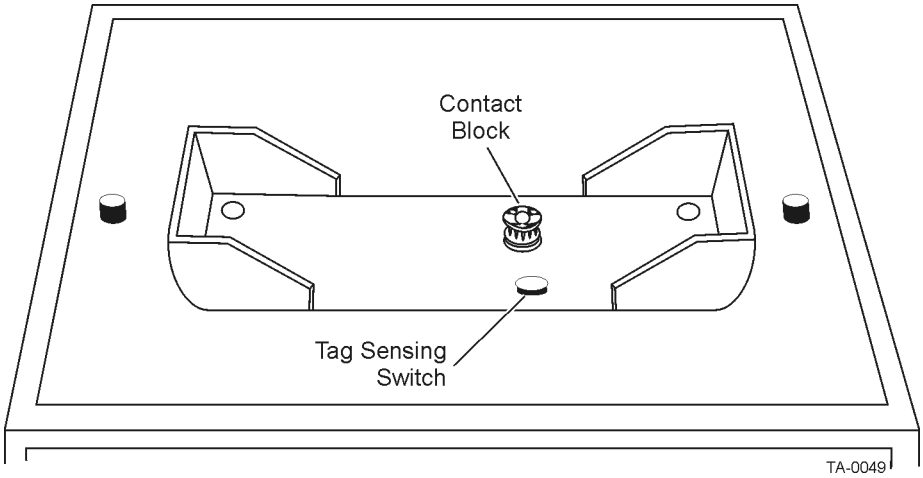

Figure 5-5 Tag Programming Head of the Tag Programmer

4. Place the tag between the positioning brackets and press down firmly and evenly to activate the tag sensing switch, as seen in Figure 5-6.

   The tag programmer allows 10 seconds to position the tag before issuing a time out error to the tag programming software. The tag programmer issues a tone every second while waiting for the tag to be positioned. A long tone and the message, `Error, Try Again` displayed in the message window indicates that the tag was not placed on the tag programming head before the programmer timed out. Return to Step 2 and attempt to program the tag again.

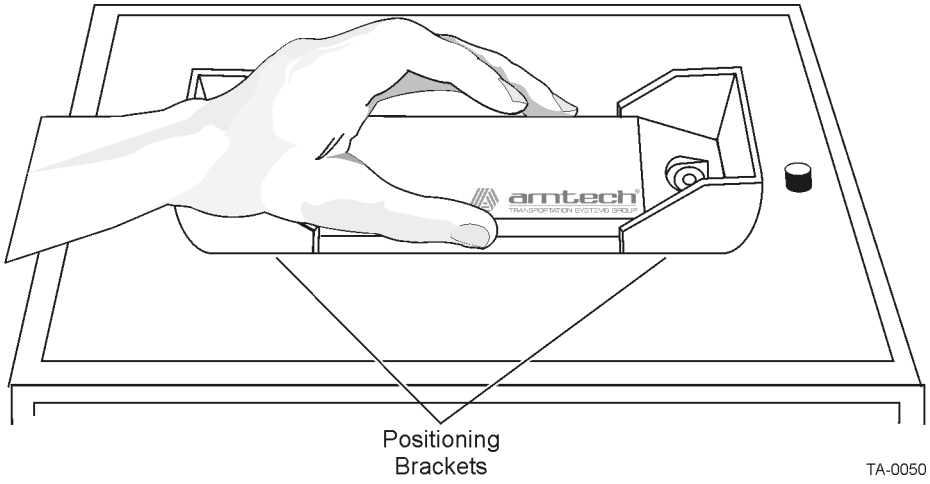

Figure 5-6 Tag in Proper Position

5. A series of quick tones signaling successful data transfer to the tag accompanied by the message, `Tag Code Verified` indicates that the tag data displayed on the screen has been programmed into the tag.
6. Remove the tag from the programming head. If Labels is set to **On**, the tag data prints on the label. Attach the label to the tag.
7. Enter all required data for the next tag and repeat steps 2 through 6.
8. When all tags have been programmed, press **F10** to exit the tag programming system.

Sealing the Tag Programming Port

Make sure that the tag programming port is sealed before mounting the tag on any equipment.

You may seal the tag programming port using the plug insertion tool, or you may insert the plug manually.

Note: *It is highly recommended that you use the plug insertion tool to ensure a tight, waterproof seal between the plug and the tag.*

To seal the tag programming port using the plug insertion tool

1. Wipe any dirt or grit away from the rim of the tag's programming port using a cloth dampened with isopropyl alcohol.

Note: *The plug's O-ring and the tag's programming port must remain free of dirt and other contaminants to ensure a secure, airtight seal.*

2. Inspect the rim of the programming port for gouges or deep scratches. If any gouges or deep scratches are present DO NOT proceed further. Damage to the programming port rim prevents the required airtight seal from forming.

Caution

The O-ring is very fragile and easily cut by sharp errors. Handle with care.

3. Fit an O-ring on the plug.

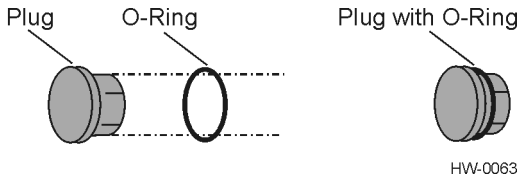

Figure 5-7 Plug and O-Ring

4. Insert the plug in the plug positioning plate so that the top of the plug is flush with the top of the plate.

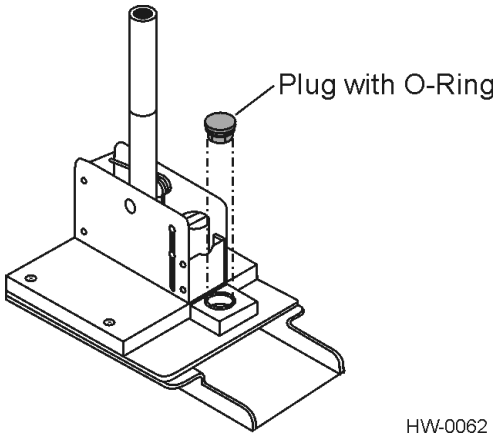

Figure 5-8 Plug in Positioning Plate

5. Place the tag in the positioning tray with the programming port-side up and on the same side of the unit as the plug.

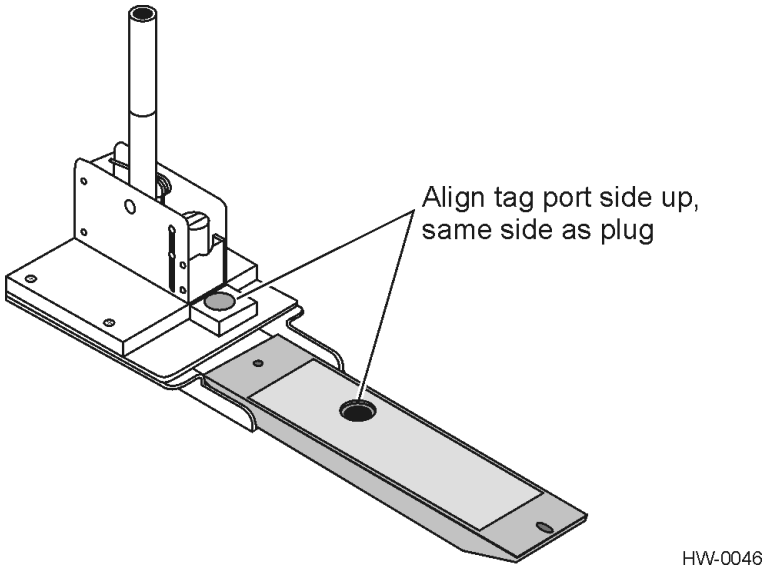

Figure 5-9 Tag in Positioning Tray

6. Slide the tag all the way into the tag positioning tray.

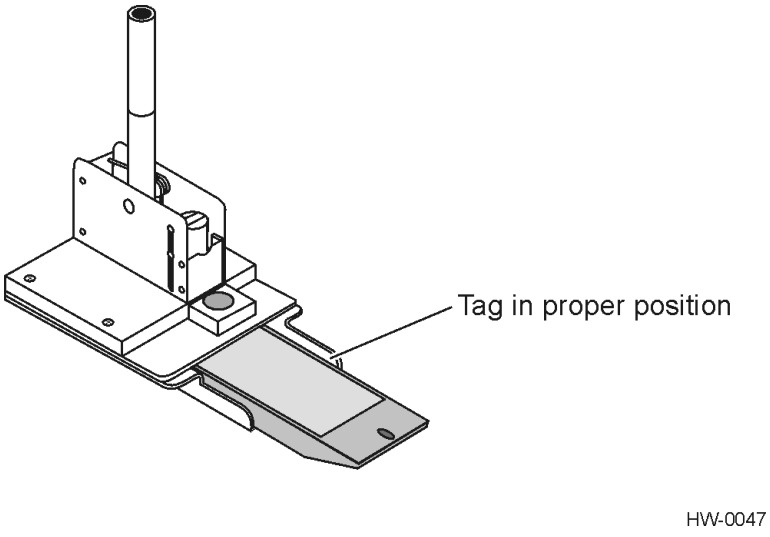

Figure 5-10 Tag in Proper Position

7. Pull the handle down gently but firmly to insert the plug into the programming port.

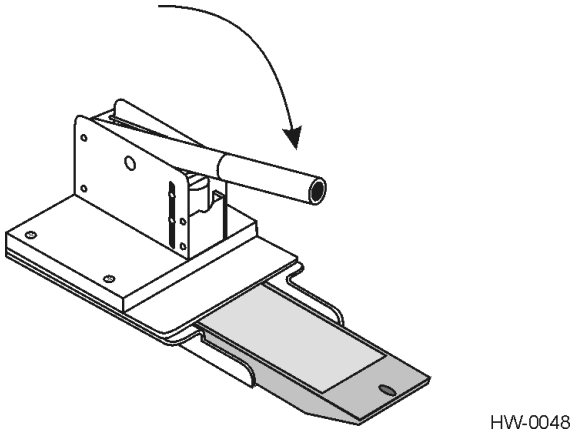

Figure 5-11 Plug Insertion

8. Return the handle to its upright position and remove the tag from the positioning tray.

To seal the tag programming port manually

1. Fit an O-ring on the plug.

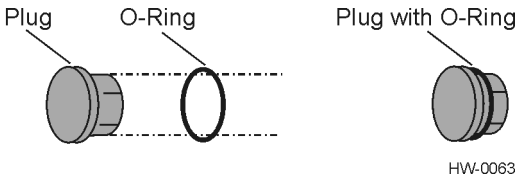

Figure 5-12 Plug and O-ring HW-0063

2. Firmly press the plug and O-ring into the programming port on the tag until the tag snaps into position.

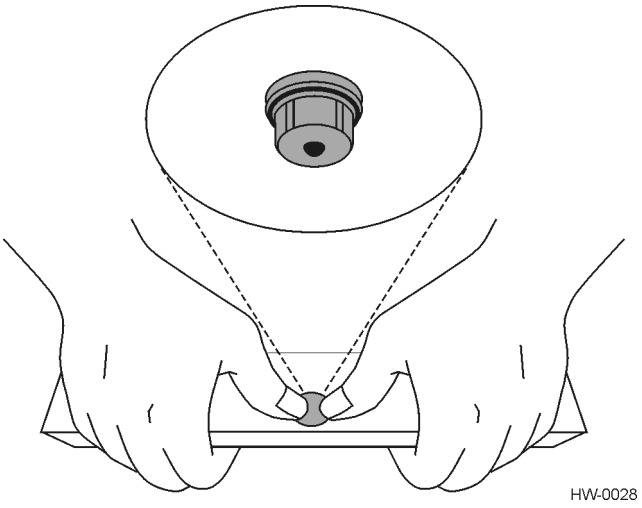

Figure 5-13 Installing the Removable Plug HW-0028

The plug is flush with the surface of the tag when properly inserted.

Attaching Tag Labels

If you have set the Labels On/Off function to **On**, a label prints when the tag is programmed. Affix this label to the tag.

Tag Programming Errors

Tag programming errors are indicated by a long continuous tone, and the message `Error, try again`. The tones continue until the tag programming sequence timer (ten tones) has expired.

Some causes of tag programming errors are:

- tag-to-programming head connection is incomplete
- the data to be transferred is invalid
- attempting to program security characters through a programmer without security settings enabled, or
- programmer timed out before the tag was placed on the programming head.

6

Using Log Files

Chapter 6

Using Log Files

Using Log Files

This chapter explains how to

- create a log file,
- replace a log file with new data,
- append new data to the end of an existing log file, and
- print a log file.

During tag programming, a record of each successfully programmed tag can be written to a log file. The date, format type, and all data field information for each tag is written into the log file.

Note: *Tag information may only be written to a log file during tag programming. No log file entries are made when entering data into an output file.*

Log files are saved as DOS ASCII files on the PC with a `.LOG` filename extension. You can print the log file to verify correct entry of tag data and keep it as a record of tagged vehicles or objects.

To create a log file

1. Press **F6** to display the message `Enter output file name`.

Note: *To cancel press **Esc** or **Enter**. This blanks the F6 - Log File Select field in the Functions window.*

2. Type the log file name and press **Enter** (to cancel press **Esc**). This displays the log file name in the **Log File** field in the Status window.
3. Enter data and program tags.
4. When tag programming is completed, press **F6** again to close the log file and save the contents to the TP directory.

Note: *The file is saved with a .LOG filename extension.*

Appending or Replacing Log Files

You may add new data to the end of an existing log file or replace the file with new data. Individual log file records can not be altered in the TP software; you may, how-

ever, edit the file using standard DOS editing techniques for ASCII files on the PC.

To append or replace log files

1. Press **F6** to display the message `Enter output file name` (To cancel press **Esc** or **Enter**. This blanks the **F6 - Log File Select** field in the Functions window.)
2. Type the log file name and press **Enter**. (To cancel press **Esc**.)

If the log file already exists, a prompt is issued for one of the following selections:

(R)eplace	to replace existing data in the file with new data to be entered.
(A)ppend	to add the new data to the end of the existing data.
(C)ancel	to cancel the Log File Select function

3. Type **R** or **A** and press **Enter**. This displays the log file name in the **Log File** field in the Status window.
4. Enter data and program tags.
5. When tag programming is completed, press **F6** again to close the log file and save the contents to the TP directory.

Note: *The file is saved with a .LOG file name extension.*

Printing Log Files

To print or display the log file on the terminal screen, exit the tag programming software and use standard DOS **PRINT** or **TYPE** commands.

7

Using Output Files

Chapter 7

Using Output Files

Introduction

This chapter explains how to

- create an output file,
- modify output file records,
- delete output file records,
- restore deleted output file records
- remove deleted output file records
- replace an output file with new data
- append new data to the end of an existing output file, and
- print an output file.

Creating an Output File

Tag information may be entered in a file on the PC instead of being immediately programmed into a tag. This allows you to enter tag data into an output file during one session, then program the tags at a later time using the output file for data input.

To create an output file

1. Press **F8** to display the message `Enter output file name`, (to cancel press **Esc** or **Enter**. This positions the cursor on the **F8 - Output File Select** field in the Functions window.)
2. Type the output file name and press **Enter**. (To cancel press **Esc** or **Enter**). This positions the cursor on the **F8- Output File Select** field in the Functions window.
3. Type the output file name and press **Enter**. (To cancel press **Esc**). This displays the **Output File** field in the Status window, and positions the cursor on the first field of the Format window.
4. Select the tag format (refer to Chapter 5 - Programming Tags).
5. Enter the tag model number (refer to Chapter 5 - Programming Tags).
6. Enter the tag data in each field (refer to Chapter 5 - Programming Tags).

7. When field data entry is complete, press **F1** or **Home** to write the current tag record into the output file, and display the messages `Encoding Tag Data`, and then `Tag Data Recorded`.
8. When all tag records have been entered, press **F8** again to close the output file and save the contents.

   The output file status is changed to **Off**.

   Note: *The file is saved with a .TAG file name extension.*

Modifying an Output File Record

This section describes how to modify an output file record.

1. Press **F8** to display the message, `Enter output file name`. (To cancel press **Esc** or **Enter**).
2. Position the cursor on **F8 - Output File Select** field in the Functions window.
3. Type the output file name and press **Enter** to display the message `File already exists. (O)k, (R)eplace, (A)ppend, or (C)ancel?`
4. Type **O** and press **Enter** to display the output file name in the Status window and the message `Enter beginning file record number.`
5. To display the first record, press **Enter**. You can then select **PgUp** to select the previous file record, or press **PgDn** to select the next file record.
6. Make any desired changes in the record fields.
7. Press **F1** to enter the modified record into the output file and display the message `Encoding Tag Data` then `Modifying existing tag data record. Are you sure? (Y/N)?`
8. Type **Y** and press **Enter** to display the message `Tag Data Recorded`.
9. You can select another record and make modifications or press **F8** to close the output file and save the contents. The output file status is changed to **Off**.

Deleting Output File Records

Deleting an output file record marks the record for deletion. The deleted record remains in the file until purged using record removal procedures (refer to “Removing Deleted Output File Records” on page 7-6).

To delete an output file record

1. Press **F8** to display the message `Enter output file name.` To cancel press **Esc** or **Enter**.
2. Position the cursor on the **F8-Output File Select** field in the Functions window.

3. Type the output file name and press **Enter** to display the message `File already exists. (O)k, (R)eplace, (A)ppend, or (C)ancel?`
4. Type **O** and press **Enter** to display the output file name in the Status window, and the message `Enter beginning file record number.`
5. Type in the tag record number to be deleted and press **Enter** to display the selected record in the Tag Format window.
6. To display the first record, press **Enter**. You can then press **PgUp** to select the previous file record, or press **PgDn** to select the next file record.
7. Press **Ctrl+D** to display the message `Record marked as deleted`, and display the next file record in the Tag Format window.

   You may select another record for deletion or press **F8** to close the output file and save the contents. The output file status is changed to **Off**.

Restoring Deleted Output File Records

Deleting an output file record marks the record for deletion, but the deleted record remains in the file until purged using record removal procedures (refer to "Removing Deleted Output File Records" on page 7-6). You may restore any deleted output file records before they have been purged or removed.

1. Press **F8** to display the message `Enter output file name.` (To cancel press **Esc** or **Enter**).
2. Position the cursor on the **F8 - Output File Select** field in the Functions window.
3. Type the output file name and press **Enter** to display the message `File already exists. (O)k, (R)eplace, (A)ppend or (C)ancel?`
4. Type **O** and press **Enter** to display the output file name in the Status window and the message `Enter beginning file record number.`
5. Type in the tag record number to be restored and press **Enter**. To display the first record, press **Enter**. You may then press **PgUp** to select the previous file record, or press **PgDn** to select the next file record.

   This displays the selected record in the Tag Format window, along with the message `Record marked as deleted.`
6. Press **F1** to enter the restored record into the output file and display the message Encoding Tag Data, then m`odifying existing tag data record. Are you sure? (Y/N)?`

   You may select another record for restoration or press **F8** to close output file and save the contents. The output file status is changed to **Off**.

Removing Deleted Output File Records

Once output file records are marked for deletion, they remain in the file until the removal procedures described in this section are performed.

To remove deleted output file records

1. Press **F8** to display the message `Enter output file name.` To cancel press **Esc** or **Enter**.
2. Position the cursor on the **F8-Output File Select** field in the Functions window.
3. Type the output file name and press **Enter** to display the message `File already exists. (O)k, (R)eplace, (A)ppend, or (C)ancel?`
4. Type **O** and press **Enter** to display the output file name in the Status window and the message `Enter beginning file record number`.
5. Type any tag record number and press **Enter** to display the selected record in the Tag Format window.
6. Press **Ctrl+R** to display the message `Removing deleted records.`
7. When all records marked for deletion in the file are removed, the message `File processing complete` displays.
8. You may modify or delete additional records or press **F8** to close the output file and save the contents. The output file status is changed to **Off**.

Printing Output Files

You can print the contents of an output file at any time.

To print output files

1. Press **F9** to display the message `Enter tag file name`.

   Note: *The cursor must NOT be positioned in a field containing sizes (length, width, or height).*
2. Type the file name and press **Enter** to display.
3. The message `Printing report` appears while the report prints.

8

Using Input Files

Chapter 8
Using Input Files

This chapter explains how to program tags using input files, and print input files.

Programming Tags Using Input Files

Output files created using previously entered tag data, or files created from other software sources (refer to Appendix A - "Record Formats for Tag Programming Input/ Output Files") are used as input files to input and transfer data to tags through the tag programmer.

To program tags using input files

1. Press **F7** to display the message `Enter input file name.`

Note: *To cancel press Esc or Enter, and position the cursor on the F7 - Input File Select field in the Functions window.*

2. Type the input file name and press **Enter** to display the input file name in the **Input File** field in the Status window. The message `Enter beginning file record number` appears. (To cancel press **Esc**.)
3. To select the first record in the file, press **Enter**. If you are resuming programming tags from a previous session and need to select a tag record other than the first, type in the tag record number to be programmed and press **Enter**.
4. Press **Enter** to display the selected record in the Tag Format window and the tag model information in the Tag Model window.

   The fields in the Tag Format window contain the tag data for the requested file record number.
5. Press **F1** or **Home** to transfer the tag data to the tag programmer and to display the message `Place tag in programmer`, which is accompanied by a series of tones.

6. Place a tag on the programming head, aligning the tag's programming port with the pins on the contact block.

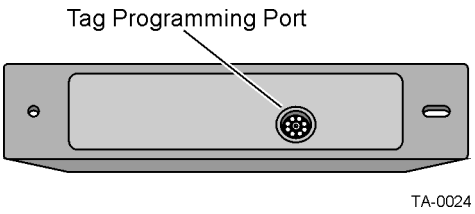

Figure 8-1 Tag to be programmed

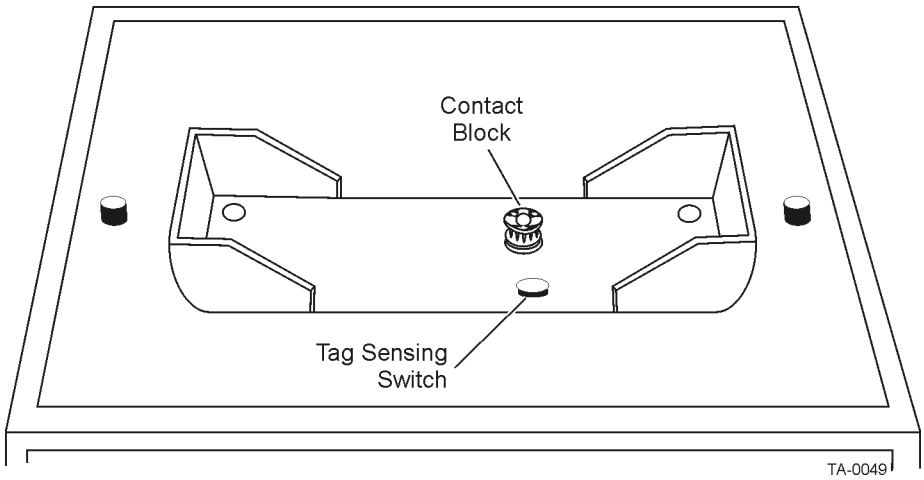

Figure 8-2 Tag programming head of the tag programmer

7. Place the tag between the positioning brackets and press down firmly and evenly to activate the tag sensing switch.

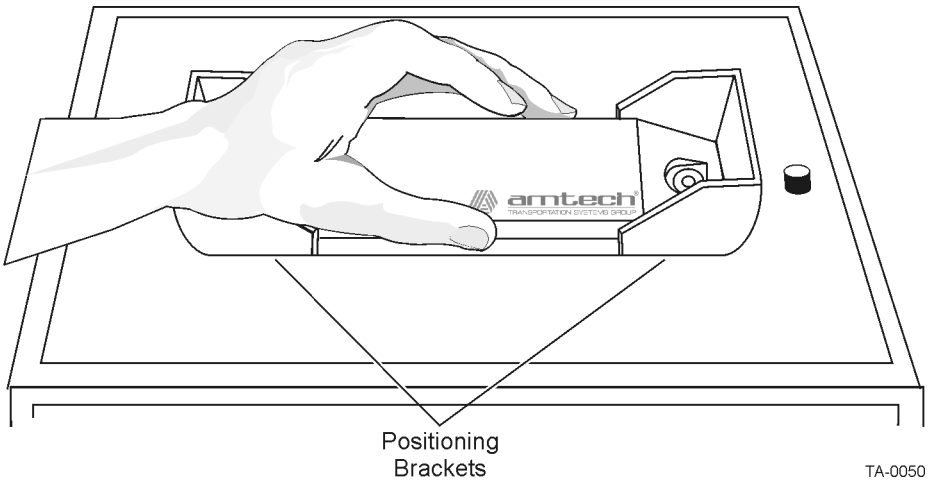

Figure 8-3 Tag in Proper Position

A series of quick tones signals successful data transfer to the tag, and displays the message `Tag code verified`. This indicates that the tag data displayed on the screen has been programmed into the tag.

The tag data for the next tag record in the file is automatically displayed on the tag data fields in the window. If the tag model changes, a unique tone sounds to alert you that the tag model has changed.

8. Remove the tag from the programming head.

   You may seal the tag programming port using the plug insertion tool, or insert the plug manually. It is highly recommended that you use the plug insertion tool to ensure a tight, waterproof seal between the plug and the tag. (Refer to "Sealing the Tag Programming Port" on page 5-12.

9. If you are printing labels, attach the label to the tag.

10. Repeat steps 5 through 8 to progam any remaining tags.

11. When tag programming is completed, press **F7** to close the input file and save the contents.

    If you are terminating an input file tag programming session before all tags have been programmed, write down the record number of the last tag programmed.Enter this record number plus 1 when you resume programming. For example, if the last record number programmed was 150, you would resume programming with record number 151.

Caution:

Always observe the File Record filed in the Status window when editing or programming tags from an input file. This field displays the input file's record number for the tag code currently on the screen. The input file is handled in a circular fashion. That is, when you have programmed the last tag record in the input file, the software displays the first record in the input file.

Printing Input Files

This section describes how to print the contents of an input file.

To print input files

1. Press **F9** to display the message `Enter tag file name`.

Note: *The cursor must not be positioned in a field containing sizes (length, width, or height).*

2. Type the file name and press **Enter** to display the message `Printing report`.

   The report prints.

9

Reading Pre-programmed Tags

Chapter 9

Reading Pre-programmed Tags

This chapter explains how to read pre-programmed tags.

Introduction

To read a pre-programmed tag

1. Remove the plug and O-ring from any sealed tags so that the tag programming port is exposed.

   Note: *It is recommended to use the TransCore AS8001 or AS8002 plug removal tool to remove the plug and O-ring.*

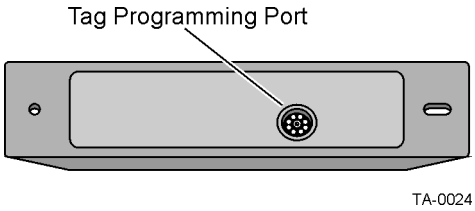

Figure 9-1 TA-0024 or TA-0027 Tag with the Plug Removed

2. Press **F2** to display the message `Place tag in programmer.`

   A series of tones indicates that the programmer is ready to transfer data from the tag.

3. Place a tag on the tag programming head, aligning the tag's programming port with the pins on the contact block.

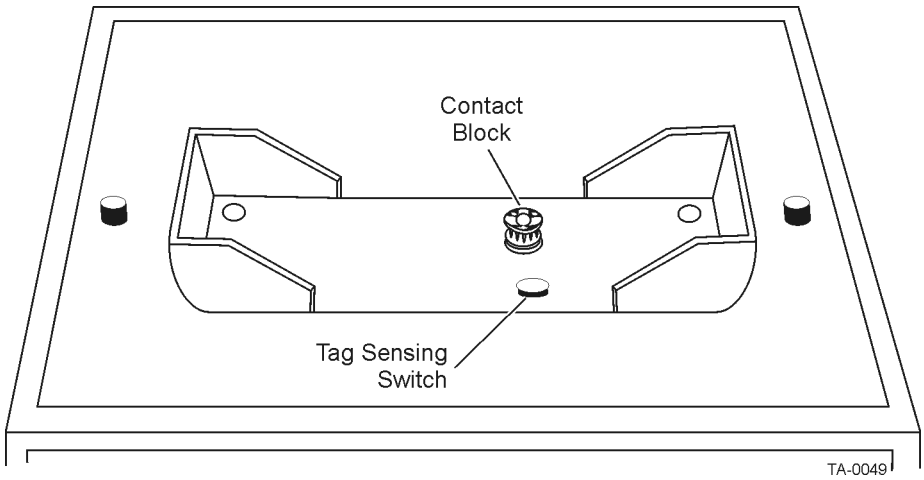

Figure 9-2 Tag Programming Head of the Tag Programmer

4. Place the tag between the positioning brackets and press down firmly and evenly to activate the tag sensing switch, as shown in Figure 9-3.

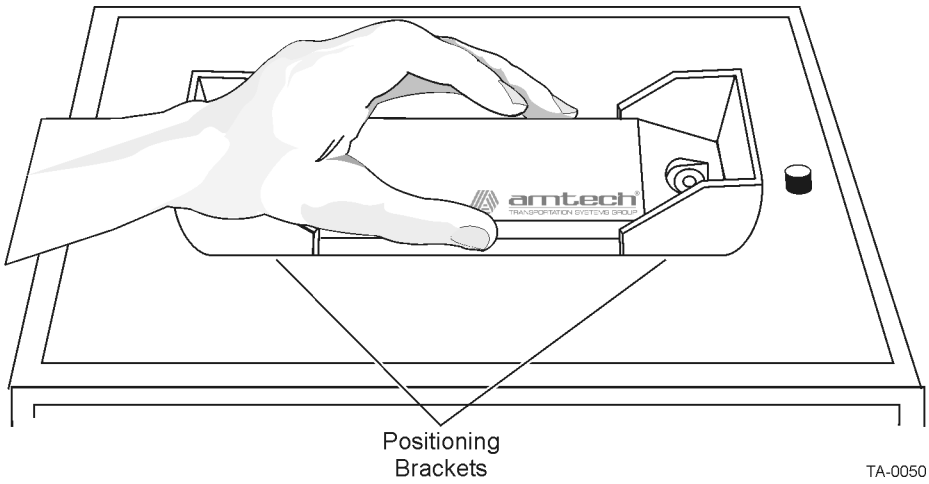

Figure 9-3 Tag in Proper Position

If the tag-to-programming head connection is not adequate, the tones continue until the timer has expired (ten tones).

A long continuous tone, along with the message `Error, try again` indicates tag read failure.

A series of quick tones signals the successful reading of the tag code and displays the tag data on the screen.

5. When tones and screen message indicate the tag read is complete, remove the tag from the programmer.

A

Record Formats for Tag Programming Input/Output Files

Appendix A

Record Formats for Tag Programming Input/Output Files

This chapter explains the TP input data record formats required for the Association of American Railroads, American Trucking Association, International Standards Organization, American National Standards Institute, and the International Air Transport Association standards, for each supported equipment type.

Programming Tags Using Input Files

The Tag Programming (TP) software accepts data input from files as well as from the keyboard. Using input data files can improve the tag programming procedure by reducing potential data entry errors.

The input files can be created either as TP output files, or as files generated by a software program run independently of TP. In order to create such an input file, the record format required by TP must be followed precisely. Because TP is used to program several standard tag formats, different record formats are required for different equipment types. The description of the formats are included in data files supplied with each TP distribution. By storing these variable record descriptions in data files, new formats can be supported by TP without software program modification.

The input file name must consist of at least one, but no more than eight characters. The file name extension mst be `.TAG` (e.g. `12345678.TAG`). The file must reside in the TP directory.

General Record Description

Each input record used by TP consists of 316 bytes. These bytes are divided into 32 fields as described in Table A-1.

Table A-1 Record Input Field Descriptions

Bytes	Fields
Bytes 1-6	Tag Type (e.g. AT5110)
Bytes 7-16	Field format name. The field format name is the name of the file TP uses to initialize the remaining fields, and is based upon equipment type. For AAR, ATA, ISO, ANSI, and IATA standard tags, the following field names are valid: • RAIL_CAR Rail car equipment • RNON_REV Rail car non-revenue equipment • CHASSIS Chassis • LOCOMOTV Locomotive • ICONTAIN Intermodal container • TRAILER Trailer • DOLLY Dolly • TRACTOR Tractor • RAIL_EOT End of train device • ULD Unit Load Device
Bytes 17-316	The remainder of the record is divided into 30 fields, consisting of 10 characters each. The arrangement of the fields is dependent upon the equipment type. As with the field format name, if a field does not use all 10 characters, it is padded with trailing ASCII "space" characters. If less than 30 fields are required for the equipment type, all unused fields are "space" filled.

Rail Car Record Layout

Table A-2 lists the record definitions for rail car record layout.

Table A-2 Rail Car Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	8	RAIL_CAR	RAIL_CAR	7-14
Filler	2	SPACES	SPACES	15-16
Equipment Code	2	19	19	17-18

Table A-2 Rail Car Record Layout (continued)

FIELD NAME	LENGTH	MIN	MAX	POSITION
Filler	8	SPACES	SPACES	19-26
Owner Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
Car Number	6	000001	999999	47-52
Filler	4	SPACES	SPACES	53-56
Side Indicator	1	0 (left)	1 (right)	57-57
Filler	9	SPACES	SPACES	58-66
Car Length	4	0000	4095	67-70
Filler	6	SPACES	SPACES	71-76
Number of Axles	2	01	32	77-78
Filler	8	SPACES	SPACES	79-86
Bearing Type	1	0	7	87-87
Filler	9	SPACES	SPACES	88-96
Platform Indicator	2	00	15	97-98
Filler	8	SPACES	SPACES	99-106
Spare 1 OR Car Type 1	1	SPACE	7	107-107
Filler	9	SPACES	SPACES	108-116
Spare 2 OR Car Type 2	3	SPACES	999	117-119
Filler	7	SPACES	SPACES	120-126
Spare 3	3	SPACES	127	127-129
Filler	7	SPACES	SPACES	130-136
Reserved	3	SPACES	511	137-139
Filler	7	SPACES	SPACES	140-146
Security	2	SPACES	ZZ	147-148
Filler	8	SPACES	SPACES	149-156
Industry Code	1	S	S	157-157
Filler	9	SPACES	SPACES	158-166
Filler	150	SPACES	SPACES	167-316

Non-Revenue Rail Car Record Layout

Table A-3 lists the record definitions for non-revenue rail car record layout.

Table A-3 Non-revenue Rail Car Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	8	RNON_REV	RNON_REV	7-14
Filler	2	SPACES	SPACES	15-16
Equipment Code	2			17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Owner Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
Car Number	6	000001	999999	47-52
Filler	4	SPACES	SPACES	53-56
Side Indicator	1	0 (left)	1 (right)	57-57
Filler	9	SPACES	SPACES	58-66
Car Length	4	0000	4095	67-70
Filler	6	SPACES	SPACES	71-76
Number of Axles	2	01	32	77-78
Filler	8	SPACES	SPACES	79-86
Bearing Type	1	0	7	87-87
Filler	9	SPACES	SPACES	88-96
Platform Indicator	2	00	15	97-98
Filler	8	SPACES	SPACES	99-106
Spare 1 OR Car Type 1	1	SPACE	Z	107-107
Filler	9	SPACES	SPACES	108-116
Spare 2 OR Car Type 2	3	SPACES	999	117-119

Table A-3 Non-revenue Rail Car Record Layout (continued)

FIELD NAME	LENGTH	MIN	MAX	POSITION
Filler	7	SPACES	SPACES	120-126
Spare 3	3	SPACES	127	127-129
Filler	7	SPACES	SPACES	130-136
Reserved	3	SPACES	511	137-139
Filler	7	SPACES	SPACES	140-146
Security	2	SPACES	ZZ	147-148
Filler	8	SPACES	SPACES	149-156
Industry Code	1	S	S	157-157
Filler	9	SPACES	SPACES	158-166
Filler	150	SPACES	SPACES	167-316

Chassis Record Layout

Table A-4 lists the record definitions for chassis record layout.

Table A-4 Chassis Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	7	CHASSIS	CHASSIS	7-13
Filler	3	SPACES	SPACES	14-16
Equipment Code	2	27	27	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Chassis Mark	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
Chassis Number	6	000001	999999	47-52
Filler	4	SPACES	SPACES	53-56
Type Detail Code	2	00	15	57-58

Table A-4 Chassis Record Layout (continued)

FIELD NAME	LENGTH	MIN	MAX	POSITION
Filler	8	SPACES	SPACES	59-66
Tare Weight in 100 kg	2	00 or 15	77	67-68
Filler	8	SPACES	SPACES	69-76
Height in cm	3	000 or 040	166	77-79
Filler	7	SPACES	SPACES	80-86
Tandem Width Code	1	0	3	87-87
Filler	9	SPACES	SPACES	88-96
Forward Extension in cm	3	000 or 030	154	97-99
Filler	7	SPACES	SPACES	100-106
Kingpin Setting in cm	3	000 or 030	154	107-109
Filler	7	SPACES	SPACES	110-116
Axel Spacing in dm	2	00 or 10	40	117-118
Filler	8	SPACES	SPACES	119-126
Running Gear Locat. in dm	2	00 or 13	43	127-128
Filler	8	SPACES	SPACES	129-136
Number of lengths	1	0	7	137-137
Filler	9	SPACES	SPACES	138-146
Minimum Length in cm	4	0000	2046	147-150
Filler	6	SPACES	SPACES	151-156
Spare	1	0	3	157-157
Filler	9	SPACES	SPACES	158-166
Maximum Length in cm	4	0000	2046	167-170
Filler	6	SPACES	SPACES	171-176
Industry Code	1	S	S	177-177
Filler	9	SPACES	SPACES	178-186
Filler	130	SPACES	SPACES	187-316

Locomotive Record Layout

Table A-5 lists the record definitions for locomotive record layout.

Table A-5 Locomotive Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	8	LOCOMOTV	LOCOMOTV	7-14
Filler	2	SPACES	SPACES	15-16
Equipment Code	2	05	05	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Owner Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
Locomotive Number	6	000001	999999	47-52
Filler	4	SPACES	SPACES	53-56
Side Indicator	1	0 (left)	1 (right)	57-57
Filler	9	SPACES	SPACES	58-66
Length	3	000	510	67-69
Filler	7	SPACES	SPACES	70-76
Number of Axels	2	01	32	77-78
Filler	8	SPACES	SPACES	79-86
Bearing Type	1	0	7	87-87
Filler	9	SPACES	SPACES	88-96
Spare	5	SPACES	ZZZZZ	97-101
Filler	5	SPACES	SPACES	102-106
Reserved	3	SPACES	255	107-109
Filler	7	SPACES	SPACES	110-116
Security	2	SPACES	ZZ	117-118
Filler	8	SPACES	SPACES	119-126

Table A-5 Locomotive Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Industry Code	1	S	S	127-127
Filler	9	SPACES	SPACES	128-136
Filler	180	SPACES	SPACES	137-316

Intermodal Container Record Layout

Table A-6 lists the record definitions for intermodal container record layout.

Table A-6 Intermodal Container Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	8	ICONTAIN	ICONTAIN	7-14
Filler	2	SPACES	SPACES	15-16
Equipment Code	2	10	10	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Owner Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
Identification Number	6	000001	999999	47-52
Filler	4	SPACES	SPACES	53-56
Check Digit	1	0	9	57-57
Filler	9	SPACES	SPACES	58-66
Length in cm	4	0000	2000	67-70
Filler	6	SPACES	SPACES	71-76
Height in cm	3	000	500	77-79
Filler	7	SPACES	SPACES	80-86
Width in cm	3	200	300	87-89
Filler	7	SPACES	SPACES	90-96

Table A-6 Intermodal Container Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Container Type	3	000	127	97-99
Filler	7	SPACES	SPACES	100-106
Max. Weight in 100 kg	3	045	455	107-109
Filler	7	SPACES	SPACES	110-116
Tare Weight in 100 kg	2	00	91	117-118
Filler	8	SPACES	SPACES	119-126
Spare	1	SPACE	3	127-127
Filler	9	SPACES	SPACES	128-136
Security	2	SPACES	ZZ	137-138
Filler	8	SPACES	SPACES	139-146
Industry Code	1	S	S	147-147
Filler	9	SPACES	SPACES	148-156
Filler	160	SPACES	SPACES	157-316

Trailer Record Layout

Table A-7 lists the record definitions for trailer record layout.

Table A-7 Trailer Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6SPACES	AT9999	1-6
Format Name	7	TRAILER	TRAILER	7-13
Filler	3	SPACES	SPACES	14-16
Equipment Code	2	21	21	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Owner (SCAC) Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46

Table A-7 Trailer Record Layout (continued)

FIELD NAME	LENGTH	MIN	MAX	POSITION
Identification Number	8	00000001	ZZZZZZZZ	47-54
Filler	2	SPACES	SPACES	55-56
Length in cm	4	0000	2047	57-60
Filler	6	SPACES	SPACES	61-66
Width Code	1	0	3	67-67
Filler	9	SPACES	SPACES	68-76
Tandem Width Code	1	0	3	77-77
Filler	9	SPACES	SPACES	78-86
Type Detail Code	2	00	15	87-88
Filler	8	SPACES	SPACES	89-96
Forward Extension in cm	3	000 or 030	284	97-99
Filler	7	SPACES	SPACES	100-106
Tare Weight in 100 kg	3	000 or 015	141	107-109
Filler	7	SPACES	SPACES	110-116
Height in cm	3	000	511	117-119
Filler	7	SPACES	SPACES	120-126
Industry Code	1	S	S	127-127
Filler	9	SPACES	SPACES	128-136
Filler	180	SPACES	SPACES	137-316

Tractor Record Layout

Table A-8 lists the record definitions for tractor record layout.

Table A-8 Tractor Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	7	TRACTOR	TRACTOR	7-13
Filler	3	SPACES	SPACES	14-16
Equipment Code	2	17	17	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Owner (SCAC) Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
Identification Number	8	00000001	ZZZZZZZZ	47-54
Filler	2	SPACES	SPACES	55-56
Number of Axels	1	0	7	57-57
Filler	9	SPACES	SPACES	58-66
Tare Weight in 100 kg	3	000	255	67-69
Filler	7	SPACES	SPACES	70-76
Wheelbase in dm	2	00 or 26	64	77-78
Filler	8	SPACES	SPACES	79-86
5th Wheel Offset in dm	1	0	8	87-87
Filler	9	SPACES	SPACES	88-96
Tare Weight on Steering Axle in 100 kg	2	0 or 20	50	97-98
Filler	8	SPACES	SPACES	99-106
Drive Axle Spread in dm	2	00	26	107-108
Filler	8	SPACES	SPACES	109-116

Table A-8 Tractor Record Layout (continued)

FIELD NAME	LENGTH	MIN	MAX	POSITION
Reserved	1	0	0	117-117
Filler	9	SPACES	SPACES	118-126
Security	2	SPACES	ZZ	127-128
Filler	8	SPACES	SPACES	129-136
Industry Code	1	S	S	137-137
Filler	9	SPACES	SPACES	138-146
Filler	170	SPACES	SPACES	147-316

Rail End-of-Train Record Layout

Table A-9 lists the record definitions for rail end-of-train record layout.

Table A-9 Rail End-of-Train Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	8	RAIL_EOT	RAIL_EOT	7-14
Filler	2	SPACES	SPACES	15-16
Equipment Code	2	06	06	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
Owner Code	4	AAAA	ZZZZ	37-40
Filler	6	SPACES	SPACES	41-46
EOT Number	6	000001	999999	47-52
Filler	4	SPACES	SPACES	53-56
EOT Type Code	1	0	3	57-57
Filler	SPACES	SPACES		58-66
Side Indicator	1	0 (left)	1 (right)	67-67
Filler	9	SPACES	SPACES	68-76

Table A-9 Rail End-of-Train Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Spare Field	7	SPACES	SPACES	77-83
Filler	3	SPACES	SPACES	84-86
Reserved	2	SPACES	ZZ	87-88
Filler	8	SPACES	SPACES	89-96
Security	2	SPACES	ZZ	97-98
Filler	8	SPACES	SPACES	99-106
Industry Code	1	S	S	107-107
Filler	9	SPACES	SPACES	108-116
Filler	200	SPACES	SPACES	117-316

Unit Load Device Record Layout

Table A-10 lists the record definitions for unit load device record layout.

Table A-10 Unit Load Device Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Tag Model #	6	6 SPACES	AT9999	1-6
Format Name	3	ULD	ULD	7-9
Filler	7	SPACES	SPACES	10-16
Equipment Code	2	01	01	17-18
Filler	8	SPACES	SPACES	19-26
Tag Type	1	2	2	27-27
Filler	9	SPACES	SPACES	28-36
ULD Type Code	3	000	ZZZ	37-39
Filler	7	SPACES	SPACES	40-46
ID Alpha Code	1	0	Z	47-47
Filler	9	SPACES	SPACES	48-56
ID Number Code	4	0000	9999	57-60
Filler	6	SPACES	SPACES	61-66

Table A-10 Unit Load Device Record Layout

FIELD NAME	LENGTH	MIN	MAX	POSITION
Owner Code	3	000	ZZZ	67-69
Filler	7	SPACES	SPACES	70-76
Spare Field	7	SPACES	ZZZZZZZ	77-83
Filler	3	SPACES	SPACES	84-86
Security	2	SPACES	ZZ	87-88
Filler	8	SPACES	SPACES	89-96
Industry Code	1	A	A	97-97
Filler	9	SPACES	SPACES	98-106
Filler	210	SPACES	SPACES	107-316

B

Creating Tag Formats

Appendix B

Creating Tag Formats

Overview

The heart of the tag programming software is a collection of files called *format files*, which define to TP the organization of tag data as defined by standards-setting bodies such as the AAR. The format files included with the TP software are current with the various standards set by such agencies as the American Association of Railroads (AAR), American Trucking Association (ATA), International Standards Organization (ISO), American National Standards Institute (ANSI), and the International Air Transport Association (IATA). These formats are stringently tested and verified for correctness with the published standards.

You may modify or add supported tag format files without software changes. Sophisticated TP users may find modifying TransCore-supplied format files brings added efficiency to tag programming operations. For example, you may wish to use data bits that are not used by a particular standard (such as the spare fields in the AAR rail car tag format). These bits, which would otherwise not be accessed by TP, can be used when appropriately defined in a format file.

Caution:
Be very careful when creating or modifying TP format files. When creating format files for TP, it is extremely important that no mistakes are made. If a mistake is made in a format file, tags will be encoded incorrectly by TP. If these incorrectly coded tags are installed in the field, a considerable expense will be incurred in de-installing the tags and returning them for new encoding.

Any newly created or modified format should be tested thoroughly. Each bit must be accounted for correctly. Any format file creation or modification is done at your own risk.

Any screen editor that handles simple character data may be used to create or modify format files. It is recommended that you do not use word processing software to create or modify format files. If you do choose to use word processing software as an editor, be very careful that it does not insert control characters such as tabs, page breaks, margin settings, or printer controls. TP will not process these control characters, which will cause the format files to be used incorrectly.

The following rules apply to all format files:

- Each format file must reside in the TP directory.
- Each format file must be uniquely named. The format file name must consist of 1

to 8 characters with .FMT as the file extension (e.g. RAIL_CAR.FMT).

- Each format file must be registered in the TP.INI file, as described later in this document. The TP.INI file acts as the catalogue for TP formats.
- Only one format per unique equipment type is allowed in the TP.INI file. When TP reads a tag, it uses the equipment type to choose the correct format from the TP.INI file for decoding the tag. If more than one format exists for the same equipment type, TP may pick the wrong format file for decoding the tag. Equipment types are described in the various standard documents.
- The format file layout must be followed without exception. Any deviation may produce unexpected results and erroneously encoded tags.
- All formats must use or account for all 120 user-programmable bits of the tag.

Adding a New Format to the TP.INI File

The TP.INI file acts as the catalogue for TP formats. This file resides in the TP directory and contains information about the various formats that TP uses to encode and decode tags. Any character text editor software may be used to add new records or modify existing records in the TP.INI file.

Caution:
Word processing software, such as WordPerfect or Microsoft Word, interjects hidden control characters such as margin settings, page breaks and tabs, which prevent TP from performing as expected. We strongly discourage using a word processor to edit the TP.INI file.

TPI.INI File Layout

The TPI.INI file has a fixed layout. Adhere to this layout exactly. The TP.INI file is a comma-delimited ASCII file. This means that each file is separated by a comma. Terminate each record with a CR (carriage return character, ASCII 13). Most text editors append a carriage return at the end of every record. Shown below is the actual TP.INI file distributed with the standard release of TP:

```
RAIL_EOT,Rail End of Train Format,S,6,*STD
RNON_REV,Non Revenue Railcar Format,S14,*STD
CHASSIS,Chassis Format,S,27,*STD
LOCOMOTV,Rail Locomotive Format,S,5,*STD
RAIL_CAR,Railcar Format,S,19,*STD
ICONTAIN,Intermodal Container Format,S,10,*STD
TRAILER,Truck Trailer Format,S,21,*STD
```

```
DOLLY,Truck Dolly Format,S,20,*STD
TRACTOR,Tractor Format,S,17,*STD
ULD,Unit Load Device Format,A,1,*STD
SIXBIT,Sixbit Tag Format," ",0,*STD
CONVERT,Metric Conversion," ",0,*STD
```

TP.INI Field Descriptions

Each record contained in the `TP.INI` file has five fields, as described in Table B-1.

Table B-1 TP.INI Fields and Descriptions

Field	Description
Format Name	This field must contain the name of the specific format file that the record pertains to, without the .FMT extension. For example, RAIL_CAR in the AP.INI file is the record entry for the RAIL_CAR.FMT file. This field must be entered in all upper case characters.
Format Description	This field contains a brief description of the format for which the entry stands. This field may be entered in both upper and lower case characters.
Industry Code Indicator	This field is very important. Many decisions within TP are based on the industry code. The value "S" is the standard entry for any formats listed within the AAR, ATA, ISO, and ANSI standard documents. The value "A" is the standard entry for any format in the IATA standard document. If the format being added does not match any of the established standards, enter the value of a blank within double quotation marks into this field as evidenced in the SIXBIT and CONVERT examples above. If a character is entered, it must be in upper case.
Equipment Identification Code	This field is also very important; it determines how TP will decode tags. TP finds the equipment identification code in the tag and searches the TP.INI file for a match. If a match is found, TP gets the format name from field one above and uses the format file to decode the tag. If no match is found, TP assumes the SIXBIT format and decodes the tag accordingly. The correct value for the equipment identification codes can be found in the various standard documents published by the AAR, ATA, ISO, ANSI, and IATA.
Format Type Indicator	This field pertains mainly to future possible enhancements to TP. Any format now used in TP should have "*STD" entered in this field as shown in the example above.

Note: *The TP.INI file must have an end-of-file (EOF) mark after the carriage return character of the last record. Any good text editor wills supply this EOF mark automatically.*

Creating a New Format File

The format files act as the heart of the TP software. These files allow the addition and modification of tag encoding schemes without necessitating software modification to TP. Much care should be taken in adding or modifying the format files. Mistakes can result in encoding erroneous information into the tags, causing costly recall of tags from the field and lost data due to tags which can not be decoded.

You can create or modify format files using any screen editor that handles simple character data. It is recommended that you do not use word processing software to add or modify table files. If you do choose to use word processing software as an editor, be very careful that it does not insert control characters such as tabs, page breaks, margin settings, and printer controls. TP will not process these control characters, which will cause the table files to be used incorrectly.

The format file must be named uniquely. The file name must contain from 1 to 8 characters with a `.FMT` extension (e.g. `RAIL_CAR.FMT`). As previously stated, the format file must be registered in the `TP.INI` file if TP is to use it.

The tag format file is divided into two sections: field and screen definition and field default values. The field and screen definition section deals with the attributes of the different fields required for the particular tag encoding scheme. The field default values section deals with the non-modifiable values of certain fields, as well as handling tables and special field handling. Both sections are contained in the same file. Each record in both sections must be terminated by a carriage return (CR, ASCII 13 character). Most text editors will automatically place a CR at the end of each record. Each record is a comma-delimited (,) ASCII record. This means that a comma separates each field value.

Figure B-1 shows the actual locomotive format as set forth by the AAR.

Field and Screen Definition Section

```
LOCOMOTV
FORMAT, D, 0, 0, 0, 0, 8, 60, 1, 0, 39, "[ Rail Locomotive Format ]"
FORMAT2, D, 0, 0, 0, 0, 10, 60, 0, 0, 39, "AAR Standard"
EQUIPID, NI, 5, 0, 31, 0, 0, 60, 1, 2, 39, "Equipment Code __"
TAGTYPE, NI, 2, 1, 4, -1, 0, 60, 1, 1, 39, "Tag Type _____"
OWNERCD, AL, 19, A, ZZZZ, 0, 12, 60, 1, 4, 39, "Equip. Initial__"
ID, NL, 20, 0, 999999, 0, 13, 60, 1, 6, 39, "Locomotive No. ____"
SID, NL, 1, 0, 1, 0, 14, 60, 1, 5, 39, "Side Indicator __"
LENGTH, NFD, 9, 0, 510, 0, 15, 60, 1, 3, 39, "Length (dm) _____"
AXLES, N, 5, 1, 32, -1, 16, 60, 1, 2, 39, "No. of Axles _____"
BEARING, N, 3, 0, 7, 0, 17, 60, 1, 1, 39, "Bearing Type _____"
SPARE, S, 30, 0, 99999, 0, 18, 60, 1, 5, 39, "Spare _____"
RESERVED, NI, 8, 0, 255, 0, 19, 60, 1, 3, 39, "Reserved _____"
SECURITY, ZI, 12, 0, 0, 0, 20, 60, 1, 2, 39, "Security _____"
TFORMAT, SI, 6, 0, 0, 0, 21, 60, 1, 1, 39, "Format Code ____"
*DONE*
```

Field Default Values Section

```
FORMAT, , 0,
FORMAT2, , 0,
EQUIPID, 05, 0,
TAGTYPE, 2, 0,
OWNERCD, . . . . , 0,
ID, . . . . . ,0,
SID, Left, 0,
SID, Right, 1,
LENGTH, . . . , 0,
AXLES, . . , 0,
BEARING, . , 0,
SPARE, . . . . . , 0,
RESERVED, 0, 0,
SECURITY, NA, 0,
TFORMAT, S, 0,
*DONE*
```

Figure B-1 Sample Tag Format File for Locomotives

Field and Screen Definition Section

The field and screen definition section, as illustrated in Figure B-1, defines all the fields necessary to properly encode a tag. This section also defines how TP lays out the display screen format and how TP will process each field.

The first record is simply the name of the format without the `.FMT` extension. This name must be entered in all upper case letters.

The *DONE* record approximately halfway down the file designates the end of the first section. This record is very important, as it tells TP that all fields have been defined.

Field and Screen Definition Fields

All records between the format name record (record 1) and the first *DONE* record have the same record layout. This layout consists of twelve fields. These twelve fields are defined as:

Field Name This field consists of 1 to 8 characters, all upper case. The name must be unique for each field within the same format file.

Field Type This field consists of 1 to 5 characters which designate the type of field and how the field is to be processed. The first (leftmost) character designates the type of field and must be one of the following:

- N - Numeric only
- A - Alphanumeric Modulo (base) 27
- B - Alphanumeric Modulo (base) 37
- S - Alphanumeric upper case only six-bit
- Z - Alphanumeric upper case only six-bit (seldom used)
- P - Protected field, no entry allowed
- X - Alphanumeric, upper and lower case (seldom used)
- C - Check digit
- D - For display only to screen

The second through fifth characters designate how the field is to be processed. The following process handling codes are field type specific:

N - Numeric only

- * - Convert numeric data to Modulo (base) 37
- E - Convert data to even numbers

F - Prompt for input in feet and inches or pounds and ounces

D - Convert feet and inches to decimeters

C - Convert feet and inches into centimeters

K - Convert pounds or ounces into kilograms

S - Alphanumeric upper case only six-bit

! - Numeric entry only allowed

C - Check digit

1 through 9 -Numeric digit for the number of previous fields to use for check digit calculation.

The following process handling codes are not field type specific and may be used with any field type:

L - Print the field value on a label if labels are being printed

I - Do not display and do not accept input, use default values

T - Accept input and use equipment identification, owner code, and identification number to perform a table lookup. This is a very format specific function requiring special table file construction (refer to "Appendix C - Use and Construction of Table Files").

Below are some examples of how the process handling codes can be used in conjunction with the field types:

NFD This is a numeric field. Prompt for feet and inch input. Convert the feet and inches to decimeters and store.

NI This is a numeric field. Do not display or prompt for input. Store default value.

C2 Calculate check digit based on 2 previous fields.

ZI This is a six-bit field. Do not display or prompt for input. Store default as six-bit data.

AL This is an alphanumeric field. Prompt for alphanumeric data and convert to Modulo 27 data and store. Print the value on the label, if labels are being printed.

NFDLE This is a numeric field. Prompt for numeric data in feet and inches. Convert the feet and inch data into even decimeters and store. Print the value on the label if labels are being printed.

S! This is a six-bit field. Allow only numeric characters to be entered.

Number of Bits	This field contains the total number of bits this field is to occupy in the tag. The value in this field must be greater than zero and less than sixty-one. The only exception is fields that are used for display only, such as the *Format* and *Format2* fields. These fields can contain zero, as they are not stored in the tag.
Minimum Value	This field contains the minimum value allowed for entry into this field. The minimum value is established by the applicable standard document.
Maximum Value	This field contains the maximum value allowed for entry into this field. The maximum value is established by the applicable standard document.
Field Offset	This field contains the value to be added to or subtracted from the entered value before storage into the tag. In some cases, the maximum value for a field is greater than the value that can be stored in the available bits. For example, the AAR standard document specifies that the tag type in each format must contain a value from 1 to 4, but this value must fit within 2 bits. Two bits can only contain values from 0 to 3. The field offset designated for tag type is -1. One is subtracted from the value when the tag is encoded and one is added to the value when the tag is decoded.
Display Row	This field contains the value of the row on the screen where the input prompt is to be displayed and the input accepted.
Input Column	This field contains the value of the column on the display row as defined above where the input will be accepted from.
Index	This field is reserved for future use. The *Format* and *Format2* fields should contain a zero. All other fields should contain a one (1).
Field Length	The field contains the value of the number of characters allowed for entry. For example, the AAR standards document states that the value for the number of axles on a rail car can be from 1 to 32, so the Field Length value is 2.
Input Prompt Column	This field contains the value for the column on the display row, as described above, where the input prompt will begin displaying.
Input Prompt Text	This field contains the actual input prompt text, enclosed in double quotation marks, that will be displayed at the input prompt column on the display row.

All 120 bits of the tag must be accounted for by records in the field and screen definition section of the format file. The tag actually has 128 bits, but only 120 bits are user-programmable. The various standards documents reference the following non-programmable fields:

First Check Sum	Bits 60 - 61
Reserved Frame Marker	Bits 62 - 63

Second Check Sum	Bits 124 - 125
Frame Marker	Bits 126 - 127

These fields are handled by the Amtech tag programmers and readers. They are transparent to the user and should not be considered in the tag format. The tag format should assume 120 contiguous bits starting with bit 1 and ending with bit 120. If the fields do not consume all 120 bits, spare fields should be used to consume the remaining bits. If the format does not use all 120 bits, TP will display the error message

```
Improper Tag ID Length, XX bits
```

in the message window. The XX in the message denotes the number of bits the format attempted to use.

Field Default Values Section

The records in the format file that follow the first *DONE* record comprise the field default values section. There must be at least one record in the field default values section to match each record in the field and screen definition section discussed above. There can be more than one record in the field default values section for each record in the field and screen definition section, if the field is intended to be a scrollable table. A scrollable table is a field with a fixed number of values such as the Side Indicator (SID) in the locomotive format file example above or the Type Detail Code in the Chassis format file. When a field is set up as a scrollable table, TP will load values into the field by having the user press the right or left arrow keys.

Field Default Values Fields

The records in the field default values section consist of the four following fields:

Field Name	This field value must match exactly the value entered in the corresponding record's field name in the field and screen definition record above. If the field is a scrollable table, multiple records will have the same field name value as shown in the locomotive format file side indicator (SID) example above.
Initial Display Value	This field contains the initial value to be displayed on the screen for this field when the format is first loaded by TP. After the first tag is programmed, TP will keep the last value entered for the next tag, unless the user enters another value.
Initial Storage Value	This field contains the initial value to be stored in the tag unless the user enters a value. If multiple records are used for a scrollable table, this field value must be unique for each record and correspond to the value to be stored in the tag.
Field Description	This field contains the description to be displayed on the screen if the record is part of a scrollable table. If the field is not part of a scrollable table, this field should contain blanks only.

In the locomotive format example shown in Figure B-1, the *Initial Display Value* field for the *Format* and *Format2* records contains a blank. Any field which is for display only (a "D" in the field type in the field and screen definition section record) should contain only a blank in the *Initial Display Value* field.

Any field that is not to be displayed (an "I" as one of the characters in the field type in the field and screen definition section record) must contain the value in the *Initial Display Value* field which is to be stored in the tag. For example, look at the *Reserved*, *Security*, and *Tformat* fields in the locomotive format file example in Figure B-1. In this example, "O" is to be stored for the Reserved field, "NA" for Security, and "S" for Tformat. The value in the Initial Display Value field for Tformat must always match the industry code indicator in the TP.INI file record for the format file referenced by the TP.INI record.

The final record in the format file is another *DONE* record. This second and final *DONE* record tells TP that the entire format file has been loaded for processing.

C

Use and Construction of Table Files

Appendix C

Use and Construction of Table Files

Using Table Files

The tag programming system accepts tag programming information from disk files as well as data entered from the PC keyboard. Programming tags using disk files instead of keyboard entry lets you produce the files in advance, and saves time and reduces data entry errors when the tags are actually programmed. There are two types of input disk files for TP: (1) standard tag input files as documented in Appendix A, and (2) table files.

Table files allow you to enter an appropriate identification number into TP and have the informational field values retrieved automatically. The individual records in the table file contain a starting and ending range of identification numbers of equipment with the same physical attributes.

Caution:
Table files are only practical when a significant number of pieces of equipment have the same physical attributes and are sequentially numbered.

General Guidelines

The following rules govern the use of table files:

- Table files can only be used for formats which contain a 4-character alphanumeric owner identification or equipment initial code and a numeric identification number, such as chassis, intermodal containers, locomotives, rail cars, rail end of train devices, and non-revenue rail cars. Formats that mandate alphanumeric identification numbers such as follies, tractors, trailers, and unit load devices can not use table files.
- A table file can only pertain to one type of equipment. All records within the table file must be for the same Equipment Group Type as defined in the AAR< ATA, ISO, ANSI, or IATA standards document. Different table files can be built for different types of equipment.
- Only one table file per owner identification or equipment initial code and unique equipment type code may be constructed. This means that only one table file for rail cars with owner identification code "TEST" may exist.
- The applicable TP format file must contain the letter "T" in the field type descriptor of the identification number field. Refer to "Appendix B - Creating Tag

Formats" for additional information.

- The table file must reside on disk in the TP directory.
- The identification number range within each table file record must be unique within the table file.
- The first record (header record) in the table file must contain the owner identification or equipment initial code, the applicable format file name, and the appropriate equipment type code in all upper case letters delimited by commas.
- Each record within the table file must be terminated with a carriage return (ASCII 13).
- An end of file mark must terminate the table file.

Caution:
Using table files can save time and prevent errors during tag information data entry; however, if the table file is constructed with erroneous information, the tags will be encoded incorrectly. Much care should be exercised in the development and use of table files. If the table file information is incorrect, a large number of tags can be encoded incorrectly in a short period of time. Erroneously encoded tags can incur great expense by requiring de-installation, re-encoding, and re-installation of the tag. Valuable data can also be lost if tags are encoded incorrectly and installed in the field. The development and use of table files is done at your own risk.

Creating Table Files

You may create or modify table files using any screen editor which handles simple character data. It is recommended that you do not use word processing software to add or modify table files. If you use word processing software as an editor, be very careful that it does not insert control characters such as tabs, page breaks, margin settings, and printer controls. TP will not process these control characters, which will cause the table files to be used incorrectly.

File Naming Conventions

The table file must reside in the TP directory. The table file name consists of the 4-character owner identification or equipment initial code, (eg. AMTC), the numeric equipment type code as assigned by the ISO, ATA, AAR, ANSI, and IATA standards, and a `.TBL` extension. This name is important as the TP software looks for specific table files based on these entered values.

The following are examples of valid table file names:

AMTC05.TBL	Amtech Corporation Locomotives
AMTC06.TBL	Amtech Corporation Rail End of Train Devices
AMTC10.TBL	Amtech Corporation Intermodal Containers

AMTC14.TBL	Amtech Corporation Non-revenue Rail Cars
AMTC17.TBL	Amtech Corporation Tractors
AMTC19.TBL	Amtech Corporation Rail Cars
AMTC20.TBL	Amtech Corporation Dollies
AMTC21.TBL	Amtech Corporation Trailers
AMTC27.TBL	Amtech Corporation Chassis

where AMTC is the owner identification or equipment initial code of the equipment being tagged, and 05, 06, 10, 14, 17, 19, 20, 21, and 27 are the equipment type codes for locomotives, end of train devices, intermodal containers, non-revenue rail cars, tractors, rail cars, dollies, trailers, and chassis, respectively. If you create your own table files, replace the "AMTC" in the above examples with the owner identification or equipment initial code assigned to the equipment you are tagging.

Table File Records

The actual contents of the table file depend on the particular format being used. The table file is a comma-delimited ASCII file, which means that each field, or item, in a record is separated with a comma (",").

The first record in the table file consists of the owner identification or equipment initial code, the format name, and the equipment type code for the specific type of equipment. The subsequent records contain the beginning identification number, the ending identification number, and a value for each field in the format. The number and order of the fields depends upon the format being used as described in "Appendix A - Record Formats for Input/Output Files." There can be as many of these records as needed to cover the different ranges of equipment attributes.

Caution

The beginning and ending identification numbers must not overlap from one record to another, and no duplication of ranges are allowed.

Creating a Sample Table File

Table C-1 is an example of how to build a valid rail car table file for equipment initial code "TEST":

1. Use a text editor to create a file named TEST19.TBL.
2. Enter the first or, "header", record. This record consists of:

Table C-1 Rail Car Table File Example

Field Description	Field Value
The standard equipment initial code	TEST
A comma	,
The actual format file name without the ".FMT" extension	RAIL_CAR
A comma	,
The standard equipment group code	19

3. Enter the actual table records. There is no limit on the number of table records that can be entered. Each table record consists of the same fields, but the values may vary. Table C-2 is an example of a rail car table record:

Table C-2 Rail Car Table Record Example

Field Description	Field Value
The starting range identification number	000001
A comma	,
Ending range identification number	000099
A comma	,
Side Indicator	0
A comma	,
Length in decimeters	1025
A comma	,
Number of Axles	06
A comma	,
Bearing Type	1
A comma	,

Table C-2 Rail Car Table Record Example

Field Description	Field Value
Platform Code	01
A comma	,
Spare 1 Value	00
A comma	,
Spare 2 Value	0000
A comma	,
Spare 3 Value	000
A comma	,
Reserved Value	000
A comma	,
Security (usually "NA")	NA
A comma	,
Data Format Code or Industry Code	S

Repeat this record entry until all of the table file records are entered.

Each record must be terminated with a carriage return (ASCII 13). Any good text editor will place the carriage return at the end of the record automatically. An end of file mark must be placed after the last record in the table file. Again, any good text editor will automatically place the EOF mark at the end of the file.

The following is a valid table file for rail cars:

```
TEST,RAIL_CAR,19
000001,000099,0,1024,06,1,01,00,0000,000,000,NA,S
000100,000199,0,0820,04,1,01,00,0000,000,000,NA,S
000200,000499,0,0960,06,1,01,15,1023,127,000,NA,S
000500,000999,0,4095,16,1,01,00,0000,000,000,NA,S
```

Modify the Appropriate Format File

After creating the table file, you must modify the appropriate format file so that TP can use the table file. Add the "T" descriptor to the format file's identification number field to invoke use of the appropriate table file (refer to the Field Type field in the "Field and Screen Definition Fields" section of "Appendix B - Creating Tag Formats").

Once you have made this modification to the format file, re-run TP. Entering the 4-

character owner identification or equipment initial code and identification number in the tag format window for data entry instructs TP to search for the matching table file. If this table file is found, TP searches the file for the range of identification numbers where the entered number falls. If the range is found, TP automatically fills in the rest of the fields in the tag format window from the table file values. You can then modify any of these values or program the tag with these table file values.

If the table file does not exist, or the entered identification number does not fall within any of the ranges in the table file records, the message

```
No matching table entry for value.
```

is displayed and a tone sounded. You can then type in the field values as if there were no table file present.