Tag Programming Software (Windows Version) User Guide

TransCore
8600 Jefferson Street NE
Albuquerque, New Mexico

January 2012

P/N 411670-005

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

4 Programming Tags and Reading Pre-programmed Tags

5 Using Files

A Record Formats for TP Input/Output Files

B Creating Tag Formats

C Constructing and Using Table Files

Index

List of Figures

List of Tables

1

Getting Started

Chapter 1

Getting Started

This chapter describes this guide's purpose and intended audience. It provides a list of topics covered in each section and the symbols and typographical conventions used.

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 TP software system running on a PC 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.

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.

TP provides a screen-oriented, user-friendly interface, which uses simple commands and easy-to-understand prompts. Online help is provided by clicking a menu button. Simple to install, the software operates on any PC using Windows 95, 98, 2000, or NT.

Figure 1-1 shows the components of a typical TP system.

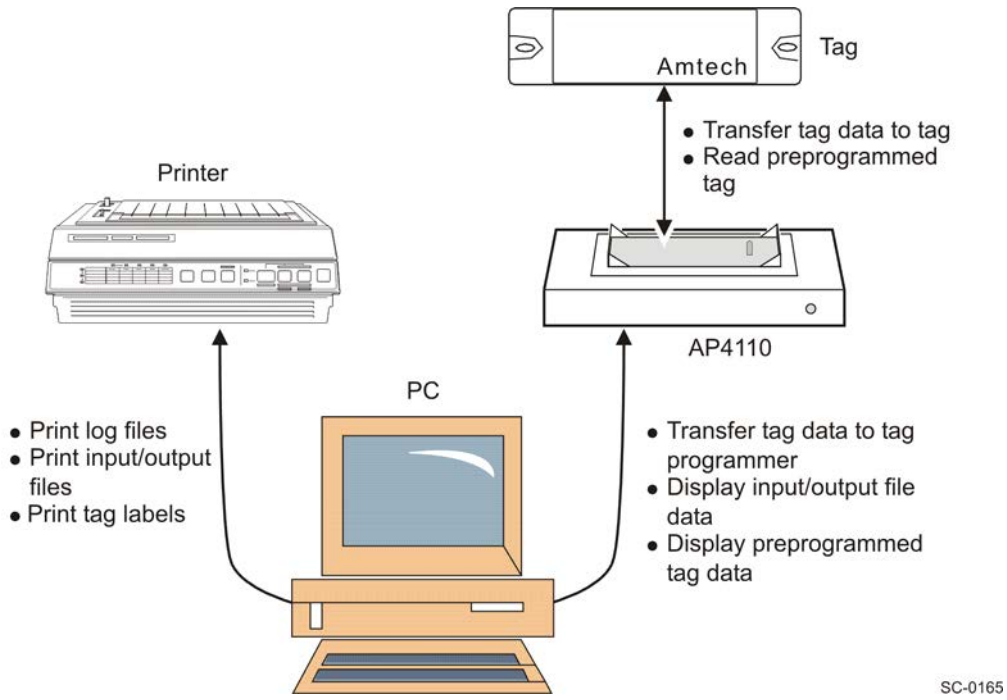

Figure 1-1 Components of the TP System

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

Entities

- Association of American Railroads (AAR) Standard for AEI
- International Standards Organization (ISO) for Freight Containers – Automatic Identification 10374.2
- American Trucking Associations (ATA) Standard for AEI
- American National Standards Institute (ANSI) Standard for Freight Containers – Automatic Identification (ANSI MH5.1.901990)
- International Air Transportation Association (IATA) Resolution 1640

This guide instructs the user about the following tag programming functions:

- Understanding the data entry screen
- Entering tag data
- Transferring data to tags
- Storing tag data to output file
- Modifying and deleting tag records

- Retrieving tag data from input files
- Using log files of the programming session
- Using output files
- Using input files
- Printing log, output, and input files
- Creating format files to customize the data entry interface
- Using table files to enter significant numbers of equipment having the same physical attributes
- Reading preprogrammed tag data
- Transferring tag data to tag programmer
- Displaying input/output file data
- Displaying preprogrammed tag data
- Printing labels for placement on each newly programmed tag
- Creating log files journaling TP 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
- Sealing tag programming port
- Using AS8010 Plug Insertion Tool

Before You Use This Guide

Before you install and use the TP software, you should already know how to use a PC and be familiar with the following operations:

- Starting the PC
- Using Microsoft Windows
- Accessing the hard disk
- Using the CD-ROM drive
- Copying files from CD-ROM

This guide includes software and hardware requirements.

Software Requirements

The TP software requires that the PC be running Windows 95, 98, 2000, or NT. The TP software is available on CD.

The tag programmer must be equipped with firmware version 2.10 or higher for the TP 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 TP software should operate on a PC that has the following hardware specifications:

- Intel Pentium (or equivalent) processor, 90MHz or higher
- Minimum of 32 MB of RAM
- VGA or higher resolution monitor
- CD-ROM drive
- Mouse or other pointing device
- Available serial port for configuring the serial RS-232 PC communications port as COMl:
- Null modem cable with a DB25 for programmer model 4101 or DB9 for model 4110 plug connector on one end and a PC-compatible plug connector on the other end

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

- Printer
- Printer port
- Printer cable
- Standard 15/16 in. x 3 1/2 in. single-column label forms

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

Guide Topics

This user guide contains the following chapters and appendixes.

Chapter 1 – Getting Started	Describes the purpose, intended audience, software and hardware requirements, guide topics, and document conventions
Chapter 2 – Setting Up the Tag Programmer and Starting the Software	Provides connectivity information for communicating with a PC and instructions for installing and starting the TP software
Chapter 3 – Navigating the Data Entry Screen	Provides information on how to use the data entry window to enter information about the vehicle or object being tagged
Chapter 4 – Programming and Reading Pre-programmed Tags	Details the steps involved in programming a tag and explains how to read pre-programmed tags
Chapter 5 – Using Files	Explains how to create, replace, append, and print a log file; how to use data from output files; how to program tags using input files, and how to print input files
Appendix A – Record Formats for Tag Programming Input/Output Files	Explains the TP input data record formats required by the AAR, ATA, ISO, ANSI, and the IATA standards for each supported equipment type
Appendix B – Creating Tag Formats	Describes the function of format files
Appendix C – Creating and Using Table Files	Describes how to use and create table files for tag programming

Typographical Conventions

These conventions shown in Table 1-1 are used in this manual.

Table 1-1 Typographical Conventions

Convention	Indication
	This procedure might cause harm to the equipment and/or the user.
	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.

Table 1-1 Typographical Conventions (continued)

Convention	Indication
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.
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.

2

Setting Up the Tag Programmer and Starting the Tag Programming Software

Chapter 2

Setting Up the Tag Programmer and Starting the TP Software

This chapter provides connectivity information for communicating with a PC and instructions for installing and starting the TP software.

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 (Figure 2-1) has an EIA RS-232 serial interface requiring a null modem cable with a DB-25 plug connector on one end and a PC-compatible plug connector on the other end. On the PC using this cable, connect the tag programmer to the COM1: port, which is the first serial communications port.

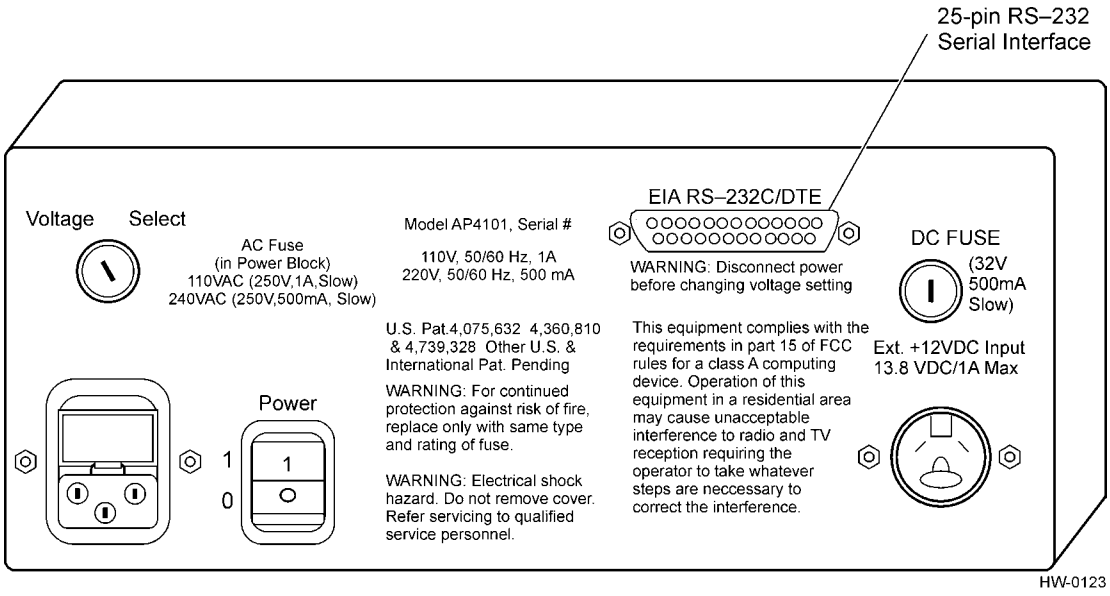

Figure 2-1 Sample View of Programmer Model 4101 Back Panel

Connecting Programmer Model 4110

Programmer model 4110 (Figure 2-2) has an EIA RS-232 serial interface requiring a null modem cable with a DB9 plug connector on one end and a PC-compatible plug connector on the other end. On the PC using this cable, connect the tag programmer to the COM1: port, which is the first serial communications port.

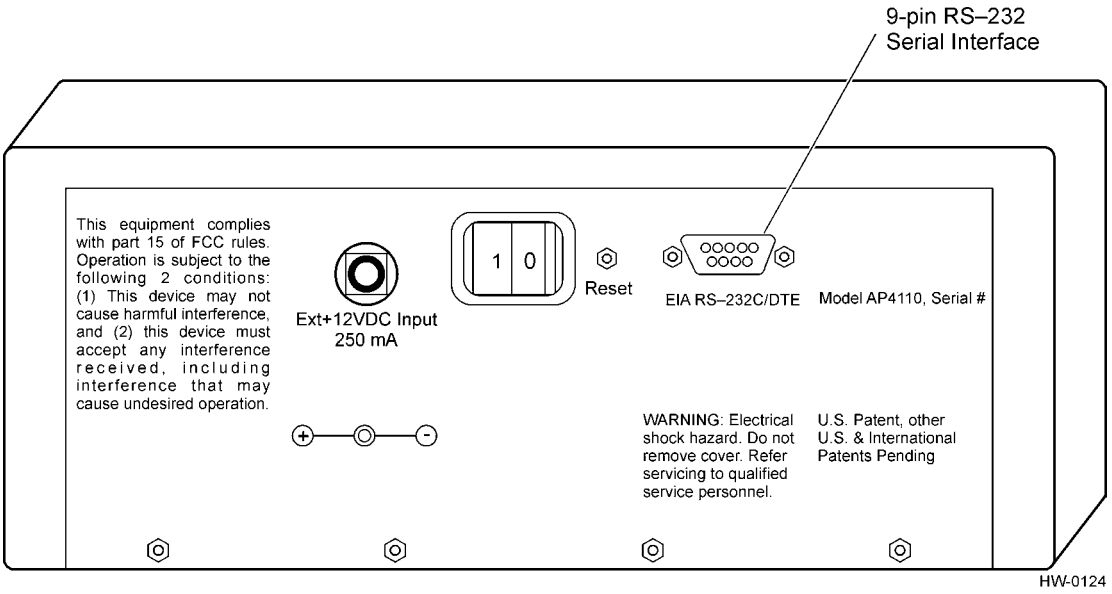

Figure 2-2 Sample View of the Programmer Model 4110 Back Panel

Installing TP Software on Your PC

Install the TP software on your PC after you have properly connected the hardware. You can install the software on PCs that have both a hard disk and CD-ROM drive.

To copy the TP software and associated files directly to the hard disk

1. Switch on the PC.
2. Insert the TP software installation CD into the CD-ROM drive.
3. Run **Setup** from the CD. Follow the instructions to create a folder on your PC hard drive.
4. The installation creates an **AP4110** shortcut that appears on your PC desktop.
5. Remove the CD from the drive.

Starting the TP Software

Once the TP software is installed on the PC, double click on the **AP4110** shortcut icon to start the TP software and display the **Data Entry** window (Figure 2-3).

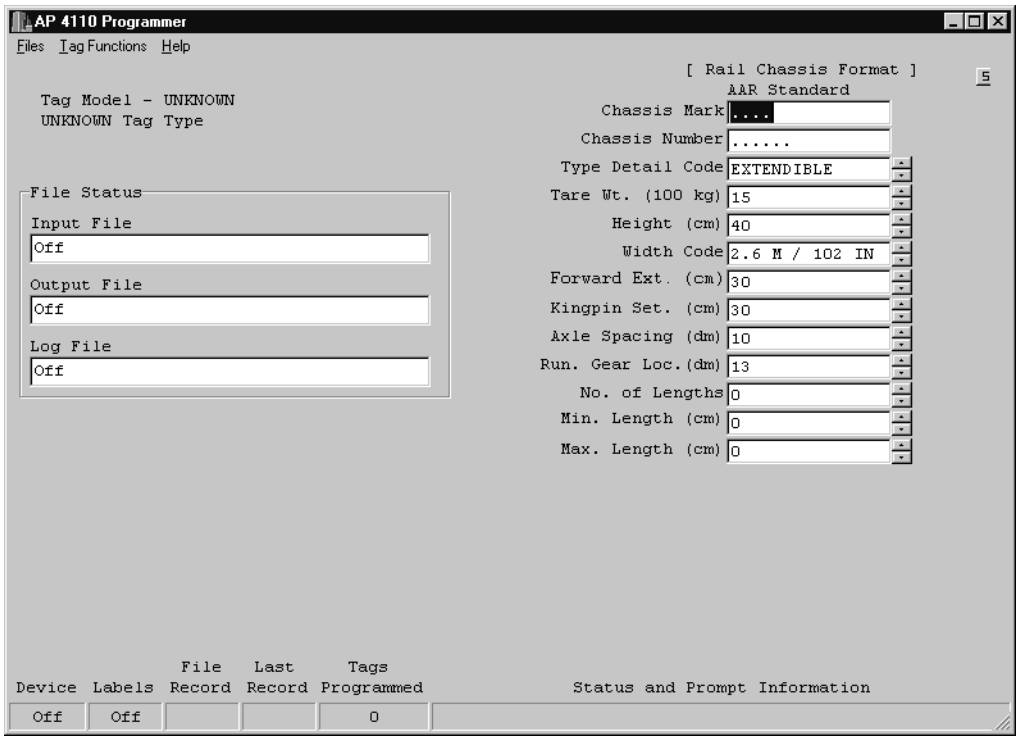

Figure 2-3 Sample TP Software Data Entry Window

3

Navigating the Data Entry Window

Chapter 3

Navigating the Data Entry Window

This chapter explains the Data Entry window and its use.

Understanding the TP Software Data Entry Window

You use the TP software **Data Entry** window to access drop-down menu lists or enter information about the vehicle or object being tagged. The **Data Entry** window also displays tag data from input files or from previously programmed tags (Figure 3-1).

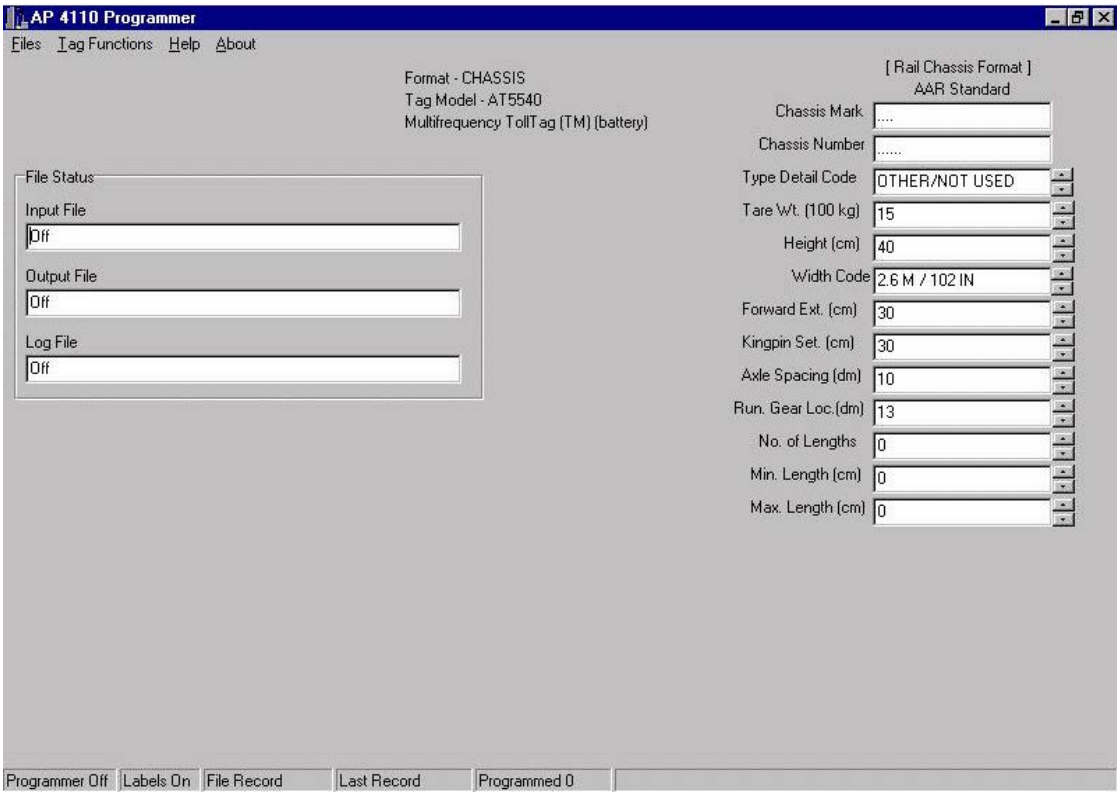

Figure 3-1 Sample TP Software Data Entry Window

The TP software **Data Entry** window has one primary menu bar (Figure 3-2).

Figure 3-2 AP4110 TP Software Main Menu Bar

The menu bar provides access to the following drop-down menus:

- **Files**
- **Tag Functions**
- **Help**

The menus are described in the following sections.

Using the Files Menu

The **Files** menu provides access to several functions. Figure 3-3 shows the **Files** drop-down menu items.

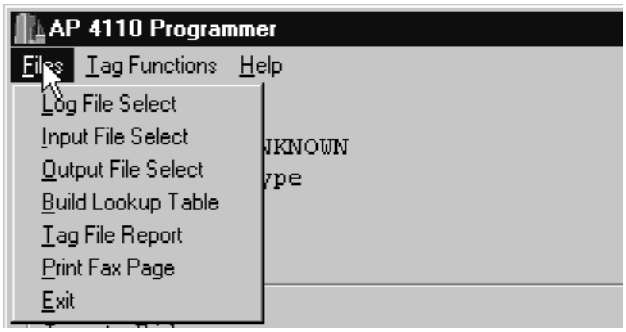

Figure 3-3 Files Drop-down Menu

Table 3-1 lists the **Files** menu functions.

Table 3-1 Files Menu Functions

Menu Function	Description
Log File Select	Prompts the user for a file (*.log) in which to record the date, format type, and all data field information for each successfully programmed tag.
Input File Select	Prompts the user for the file (*.tag) containing previously entered tag data, and selects the file to program tags. Input files are created using the **Output File Select** function.
Output File Select	Prompts the user for the file (*.tag) in which tag data will be stored. Output files store tag data, allowing tags to be programmed at a later time using the **Input File Select** function.

Table 3-1 Files Menu Functions (continued)

Menu Function	Description
Build Lookup Table	Prompts the user for a file (*.tag) in which tag programming information will be stored.
Tag File Report	Prompts the user for a tag file (*.tag), then prints a report containing the filename, date, item, record number, tag model, tag format, and tag contents.
Print Fax Page	Displays the user's local or network printer options. You use the fax page to report problems with the tag programmer or TP software to the TransCore Customer Service Center.
Exit	Exits the TP software program.

Figure 3-4 shows a Software Performance Report Fax form that is created when you choose the **Print Fax Page** feature.

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 3-4 Sample Software Performance Report Fax Form

Using the Tag Functions Menu

The **Tag Functions** menu provides access to the tag functions (Figure 3-5).

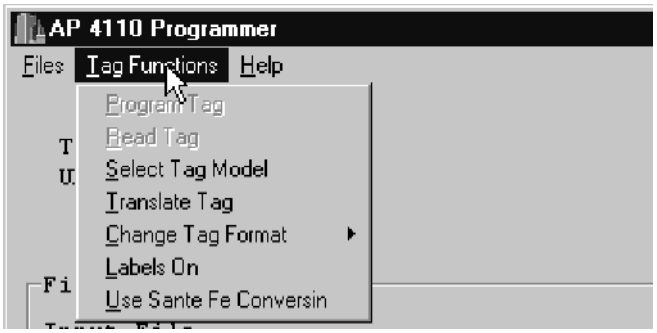

Figure 3-5 The Tag Functions Menu

Table 3-2 lists the tag functions.

Table 3-2 Tag Functions Menu

Menu Function	Description
Program Tag	Transfers tag data from the PC, via the tag programmer, to the tag positioned on the programmer. Alternately, if an output file has been designated, this command sends the completed tag data to the output file so the tags can be programmed at a later time.
Read Tag	Reads information from the tag positioned on the programmer and displays the tag data on the fields in the **Tag Format** section.
Select Tag Model	Selects tag model to be programmed.
Translate Tag	Converts six-bit or hexadecimal tag ID.
Change Tag Format	Changes the tag format type.
Labels On	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.*
Use Santa Fe Conversion	File function that had been used by Amtech® to convert customer-supplied tag files to a format usable for factory programming of tags.

Using Help

The **Help** menu lists special keys and their functions. To view the keyboard Help screen (Figure 3-6), click on the **Help** button.

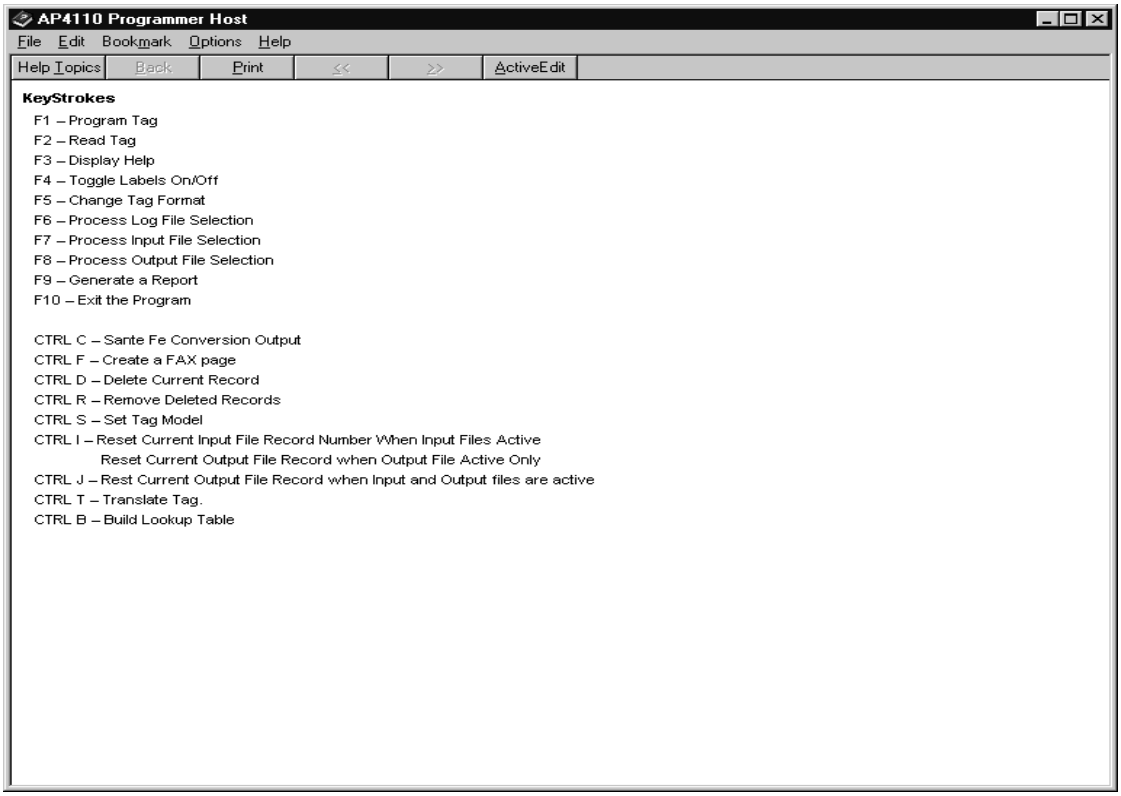

Figure 3-6 Keyboard Help Screen

Using the Status Information Boxes

The status information boxes (Figure 3-7) display the values or settings in effect for the current tag programming/data entry session.

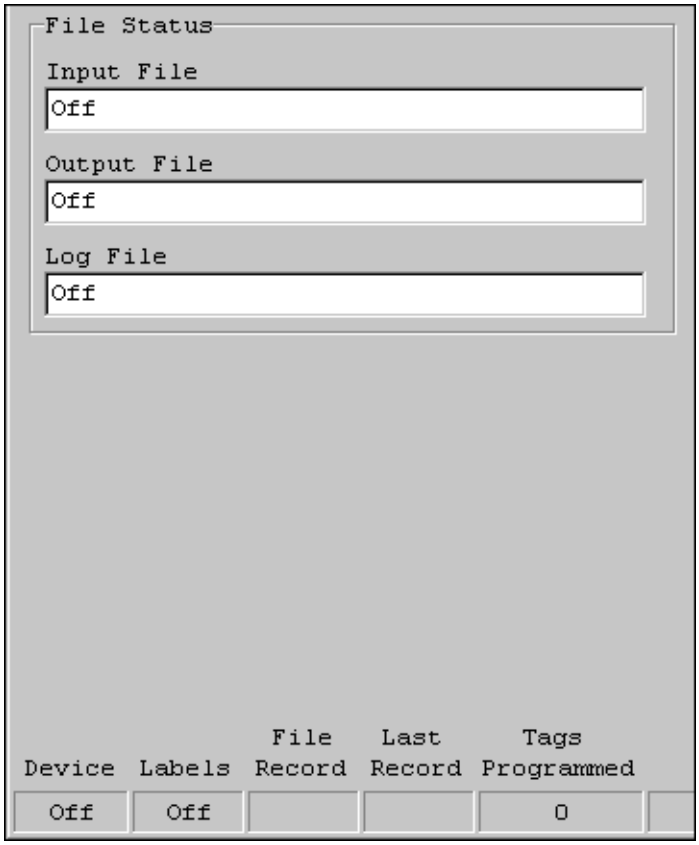

Figure 3-7 Status Information Boxes

The status information fields and their settings are discussed in Table 3-3.

Table 3-3 Status Information Field Descriptions

Status Field	Description
Input File	This field displays the name of the file being used to program tags using previously entered tag data. Figure 3-3 shows the menu selectable option used to select an input file.
Output File	This field displays the name of the file receiving tag data to be stored. Figure 3-3 shows the menu selectable option used to select an output file.
Log File	This field displays the name of the log file receiving tag programming records. Figure 3-3 shows the menu selectable option used to select a log file.
Device	This field indicates whether a tag programmer is attached and active (ON).
Labels	This field indicates whether or not to print the tag data on labels as tags are successfully programmed and verified. Figure 3-5 shows the menu selectable option that switches label printing off or on.

Table 3-3 Status Information Field Descriptions (continued)

Status Field	Description
File Record	This field 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	This field displays the number of the last record in an active input or output file.
Tags Programmed	This field displays the total number of tags programmed during the current session. **Exit** from the program to reset this number to zero.

Understanding the Tag Model Information

The Tag Model section of the **Data Entry** window displays information about the tag model that is being programmed, that is being stored to an output file, or that has been retrieved from an input file (Figure 3-8).

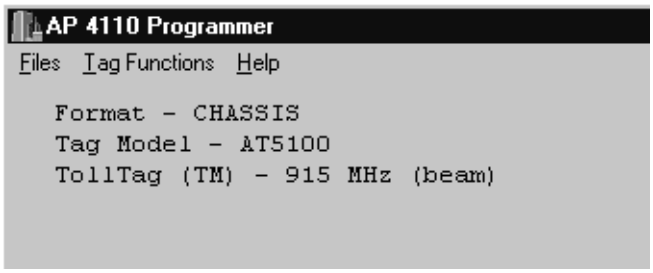

Figure 3-8 Tag Model Information

Choose the appropriate tag model from the list 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. Figure 3-8 shows that an AT5100 TollTag™, 915 MHz (beam) style, has been selected. If you do not enter a tag model, the tag window displays `Tag Model - UNKNOWN, UNKNOWN Tag Type` (Figure 3-9).

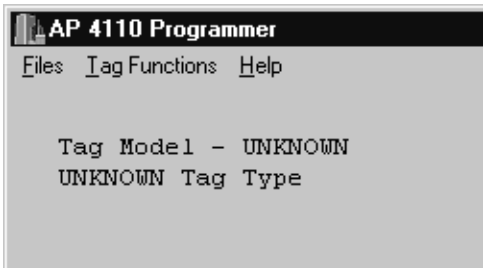

Figure 3-9 Default Tag Model Information

If you use an input file to program tags, the Tag Model section 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.

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

Using the Tag Format

The Tag Format section of the **Data Entry** window, shown in Figure 3-10, 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 section contains four main components:

- Tag format type
- Tag format standard status
- Data entry field description
- Data entry field

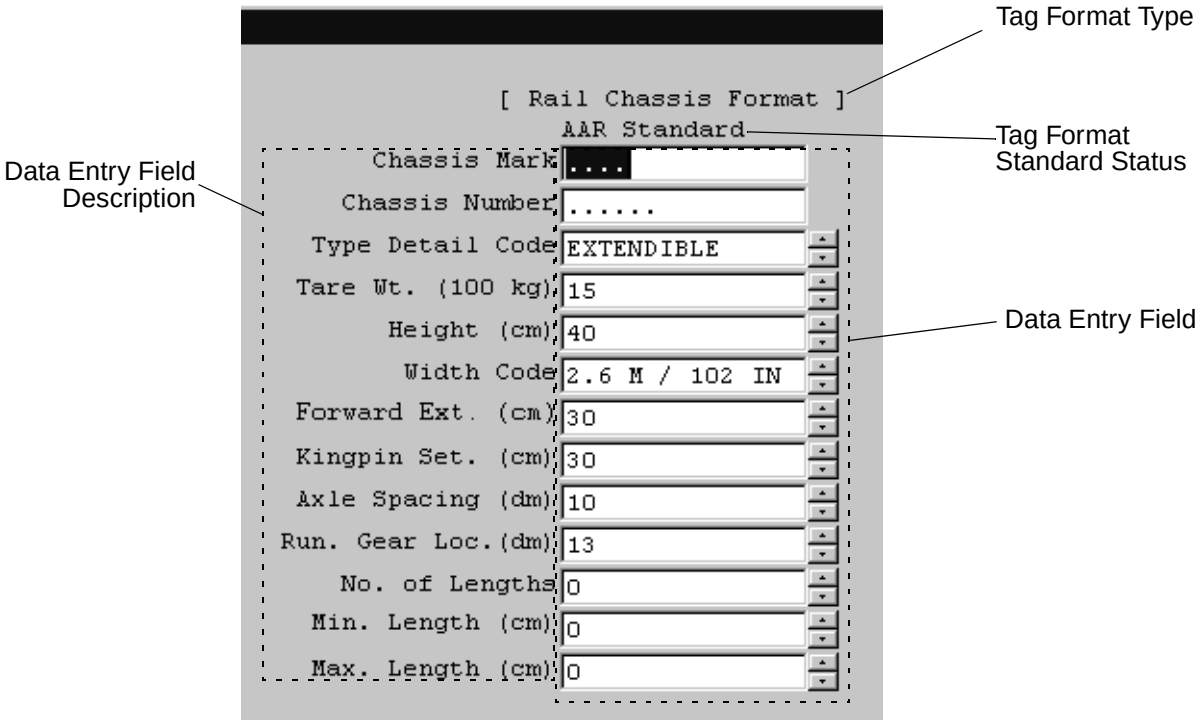

Figure 3-10 The Tag Format Section

Table 3-4 lists the tag format fields and their descriptions.

Table 3-4 Tag Format Fields

Tag Format Fields	Explanation
Tag Format Type	Presents the type of vehicle or object to be tagged. The data entry field descriptions change based on the format selected. To choose the desired tag format, select **Change Tag Format**.
Tag Format Standard 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 is 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	Shows the location where data is entered for each field item. The number of dots represents the maximum number of characters that may be entered. The dots are replaced with characters as they are being typed.

Status and Prompt Information

The Status and Prompt Information section (Figure 3-11) is located in the lower right corner of the **Data Entry** window. Unlike the other areas of the window, the message area is not displayed continuously. It is displayed only when there is a Help message, field value range message, activity status message, or error message.

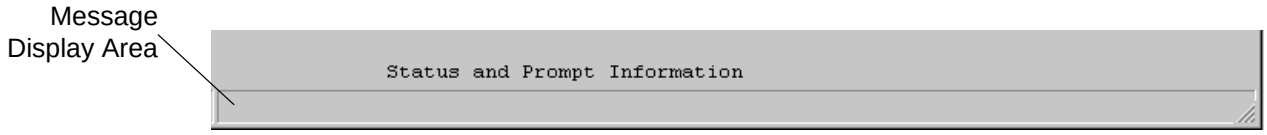

Figure 3-11 Status and Prompt Information Section

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 TP software contains numeric, alphabetic, alphanumeric, and table fields. If you enter a field entry that is inconsistent with the field type or outside the range of allowable values for the field, an error tone signals. An appropriate Help message for that field appears in the message area. Figure 3-6 lists control characters that can be used to navigate the data entry tasks. In addition, some keys on the keyboard (e.g., Page Up, Page Down, Delete) 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.

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 TP software uses certain keys on the number keypad for cursor control purposes. On PC models without a 101-key extended keyboard, you must use the number keys at the top of the PC keyboard to enter numerical characters.

4

Programming Tags and Reading Pre-programmed Tags

Chapter 4

Programming Tags and Reading Pre-programmed Tags

This chapter details the steps involved in programming tags and reading pre-programmed tags.

Overview

This chapter covers the following tasks involved in programming a tag:

- Selecting the label print option to switch label printing 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
- Reading pre-programmed tags

Selecting the Label Print Option

If you want to print a label for each tag, access the **Tag Functions** menu from the **Data Entry** window and click **Labels On** (Figure 4-1)**.** Insert standard 15/16-inch, single-column label forms in the printer. Selecting **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 select **Labels 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 six lines per inch.*

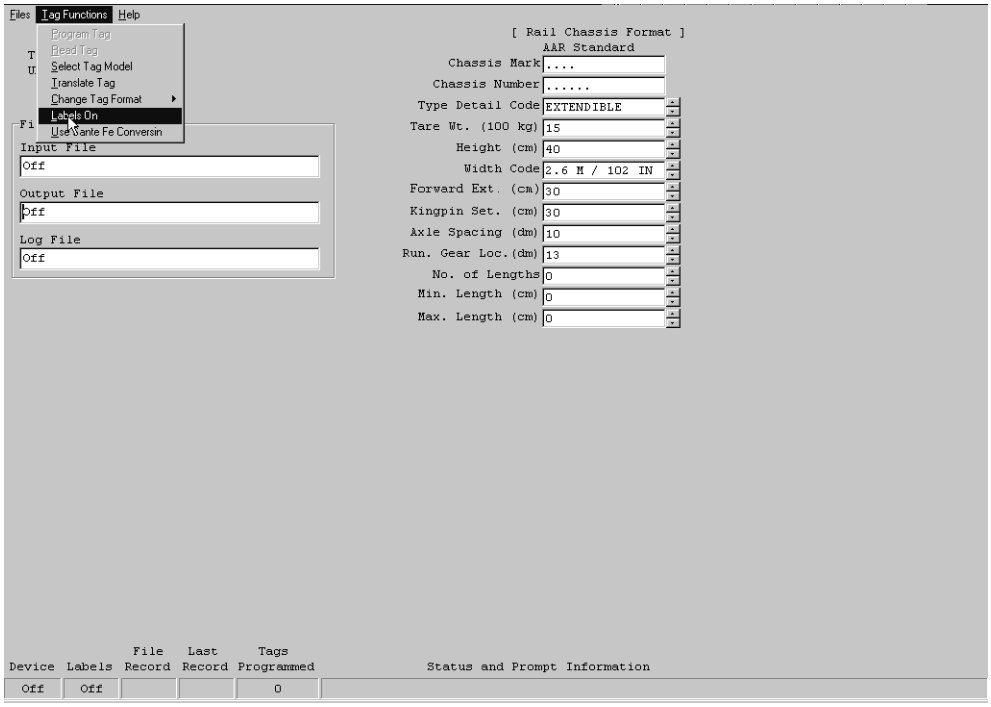

Figure 4-1 Labels On Menu Item

Selecting a Tag Format

The first task in programming a tag is to select a tag format. The TP 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 “Using the Tag Format” on page 3-10.

To select the tag format

1. Access **Tag Functions** drop-down menu, then select **Change Tag Format**.
2. Use your mouse to click on a tag format (Figure 4-2).

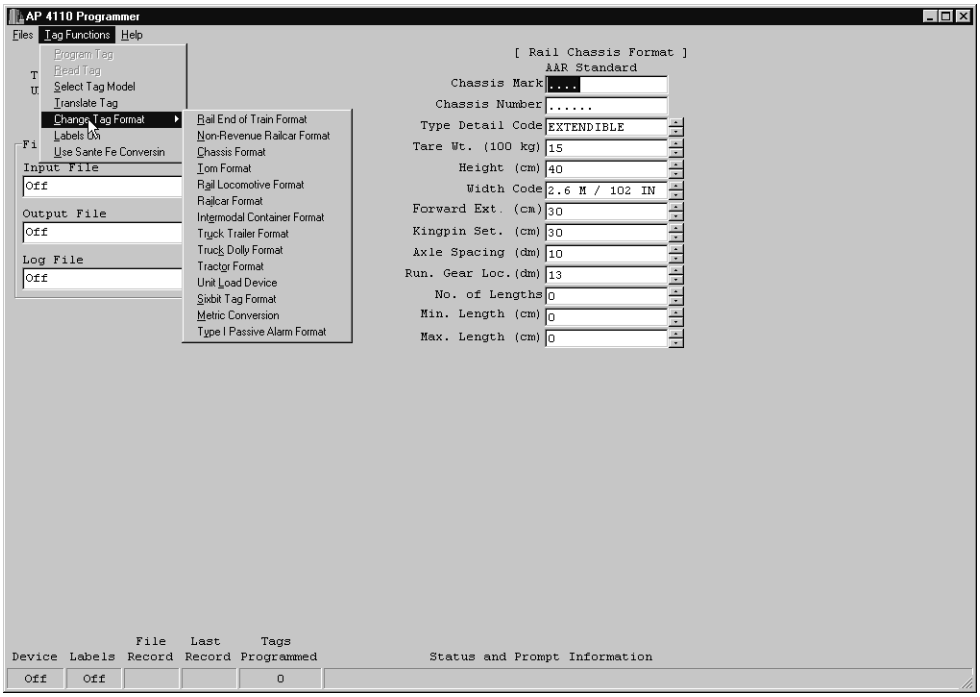

Figure 4-2 Tag Format Selection List

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 “Understanding the Tag Model Information” on page 3-9.

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. Select the **Tag Functions** pull-down menu, then click on **Select Tag Model**, which displays the tag model selection list.
2. Use the slide scale on the side to view all tag models. Highlight the tag model on the list (Figure 4-3) and click **Select**.

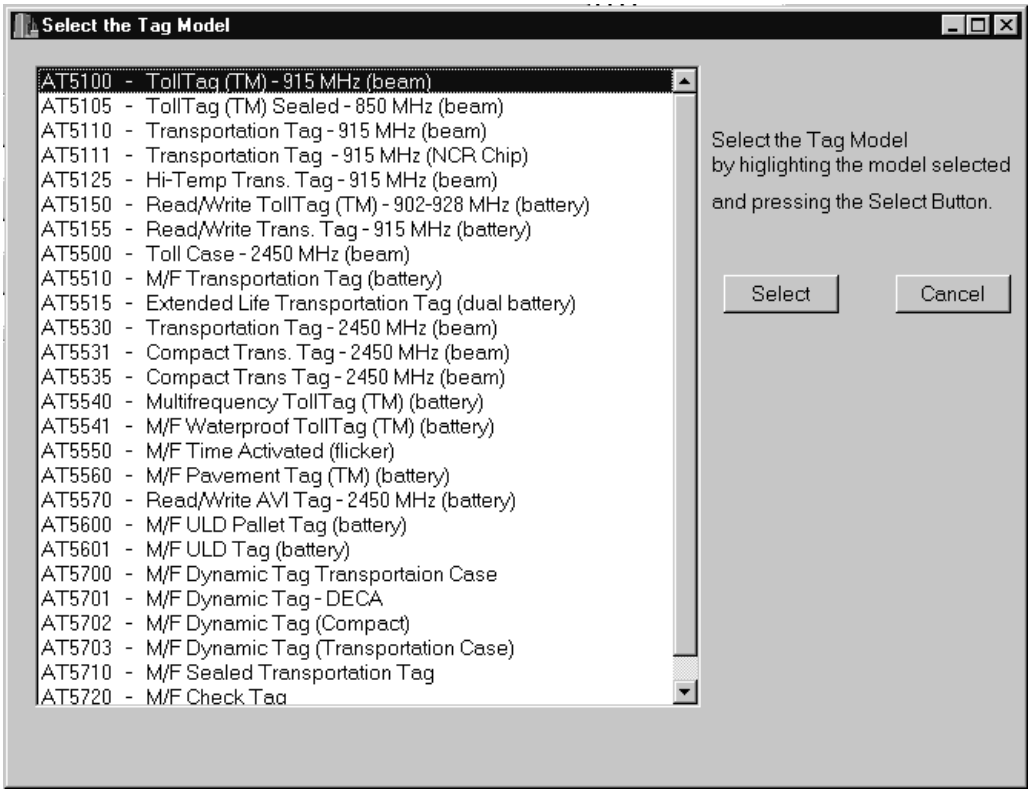

Figure 4-3 Tag Model List

Entering Tag Data

Once you have selected a tag format and selected the tag model, 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 section. 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 3, “Navigating the Data Entry Window”, for data entry techniques.

End-of-Train Device Format Entries

Figure 4-4 illustrates a sample **Data Entry** window for entering end-of-train (EOT) device tag information.

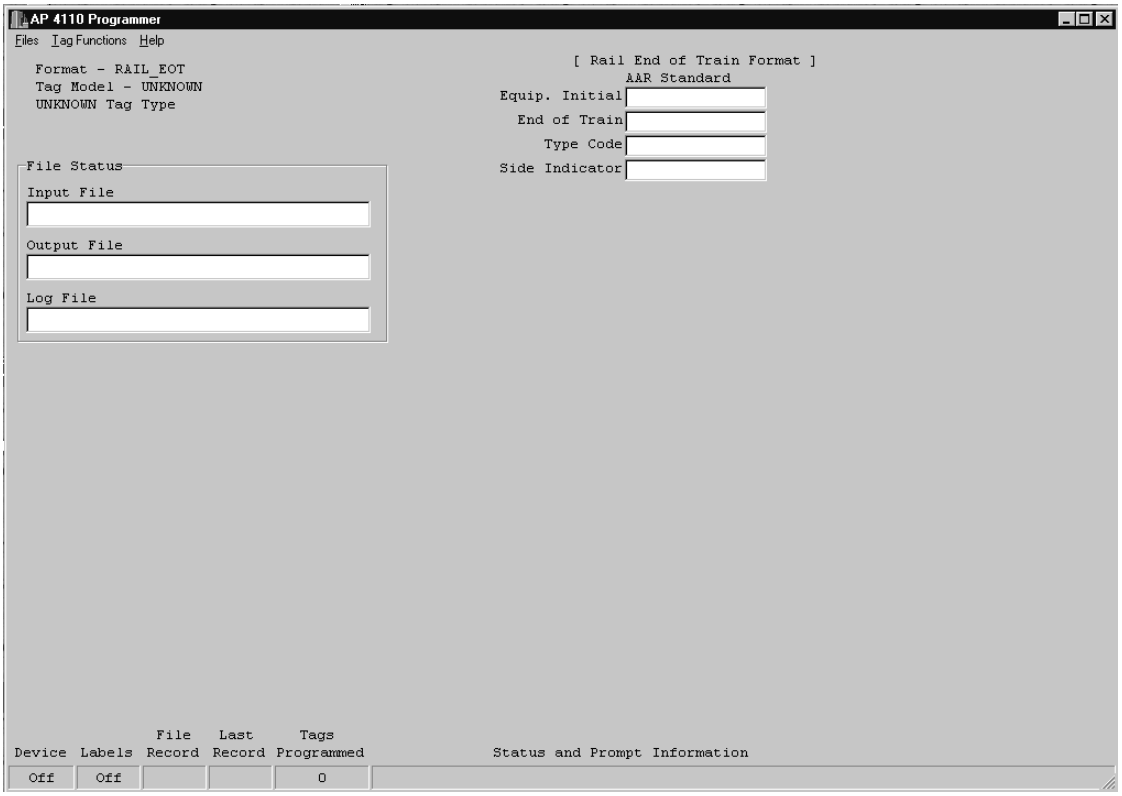

Figure 4-4 EOT Device Data Entry Window

The EOT device format field entries are described in Table 4-1.

Table 4-1 EOT Device Format Field Entries

Field Entry	Description
Equip. Initial	Enter up to four alphanumeric characters from A to ZZZZ representing the car initial
End of Train	Enter up to six numbers from 0 to 999999 representing the EOT device number.
Type Code	Enter one of the numbers listed below to provide a generic description of the EOT device: 0 Active EOT, AAR-compatible 1 Active EOT, not AAR-compatible 2 Passive EOT 3 Not used/unknown

Table 4-1 EOT Device Format Field Entries (continued)

Field Entry	Description
Side Indicator	Enter a one-digit code indicating whether the tag is to be installed on the left or right side of the EOT device. The right or left side is in reference to a person standing to the rear of and facing an EOT device properly mounted. 0 Left side 1 Right side

Locomotive Format Entries

This section provides examples of the locomotive format entries.

Figure 4-5 illustrates a sample locomotive format **Data Entry** window for entering locomotive tag information.

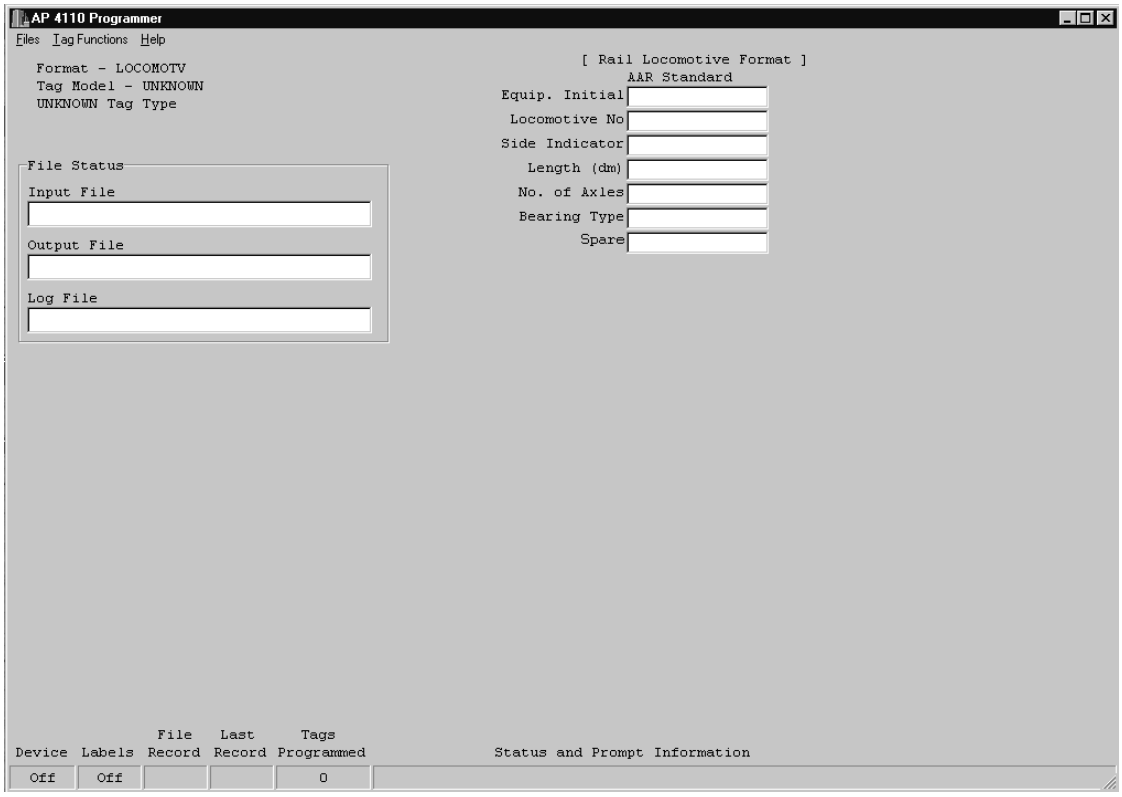

Figure 4-5 Locomotive Data Entry Window

Table 4-2 describes the locomotive format field entries.

Table 4-2 Locomotive Format Field Entries

Field Entry	Description
Equip. Initial	Enter up to four alphabetic characters from A to ZZZZ representing the car initial.
Locomotive No.	Enter up to six numbers from 0 to 999999 representing the locomotive number.
Side Indicator	Enter a one-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 a 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 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 (dm).
No. of Axles	Enter a number from 1 to 32 indicating the number of axles on the locomotive.
Bearing Type	Enter a 1 for roller bearings, which will be correct for 99% of cases. Other possible entries include: 0 Plain bearings 2 Roller bearings, inboard 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 five characters may be entered in this field.

Figure 4-6 shows the **Feet Inches Entry** pop-up window that you use to convert feet and inch measurements to dm. You can type in the measurement or use the slide scale to the right of each entry box to enter your data.

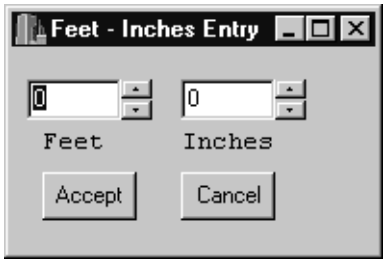

Figure 4-6 Feet Inches Entry Pop-up Window

Rail Car Format Entries

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

Figure 4-7 illustrates a sample **Data Entry** window for entering rail car tag information.

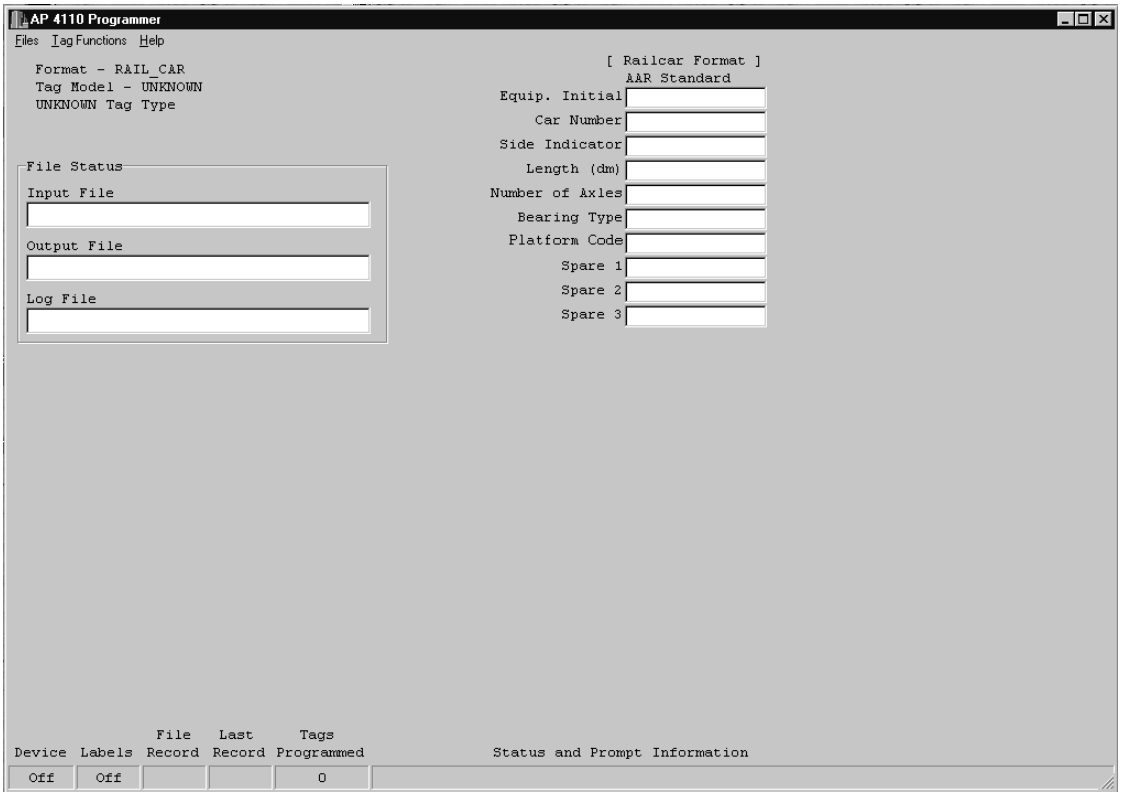

Figure 4-7 Rail Car Data Entry Window

Table 4-3 describes the rail car format field entries.

Table 4-3 Rail Car Format Field Entries

Field Entry	Description
Equip. Initial	Enter up to four alphabetic characters from A to ZZZZ representing the car initial.
Car Number	Enter up to six numbers from 0 to 999999 representing the car number.

Table 4-3 Rail Car Format Field Entries (continued)

Field Entry	Description
Side Indicator	Enter a one-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
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 dm. Refer to Figure 4-6 for information on the **Feet Inches Entry** pop-up window
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, which is correct for 99% of the cases. Other possibilities include: 0 Plain bearings 2 Roller bearings, inboard 3 Roller bearings, 3-axle truck, 1-axle obstructed (*Buckeye Design*, an old-style friction bearing) 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 time. Refer to Chapter 5, “Using 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. Select **Program Tag** from the **Tag Functions** menu to transfer the tag data from the PC to the tag programmer, which displays 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 (Figure 4-8) with the pins on the contact block (Figure 4-9).

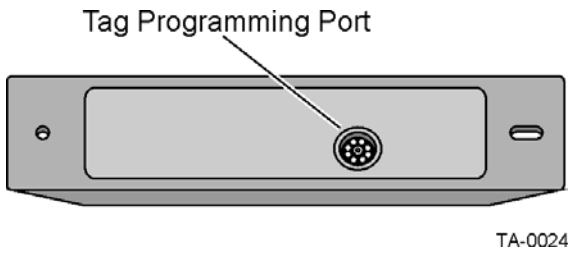

Figure 4-8 Tag To Be Programmed

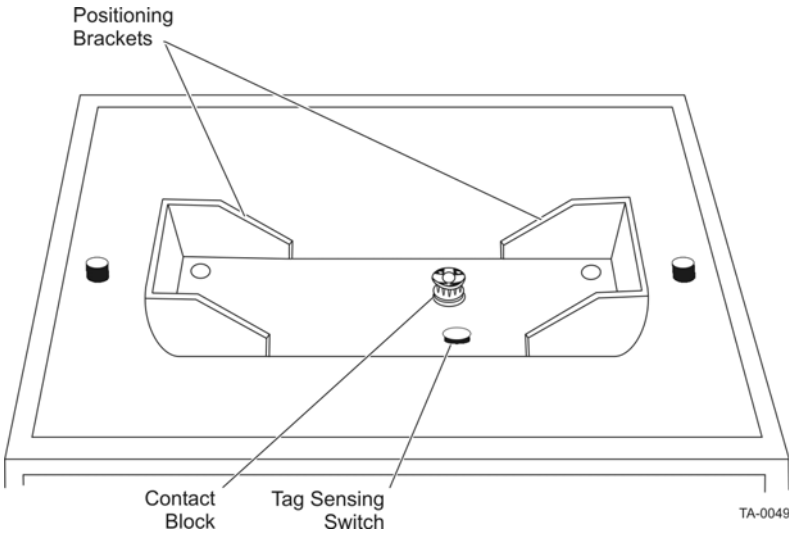

Figure 4-9 Tag Programming Head of the Tag Programmer

4. Place the tag between the positioning brackets (Figure 4-10) press down firmly and evenly to activate the tag sensing switch. Select **Files** and **Program Tag** or press **F1** to program the tag.

   The tag programmer waits 10 seconds while you position the tag before issuing a time-out error to the TP software. The tag programmer sounds a tone every second while waiting for the tag to be positioned. If the message `Error, Try Again` displays in the message window followed by a long tone, the tag was not placed on the tag programming head before the programmer timed out. Return to Step 2 and program the tag again.

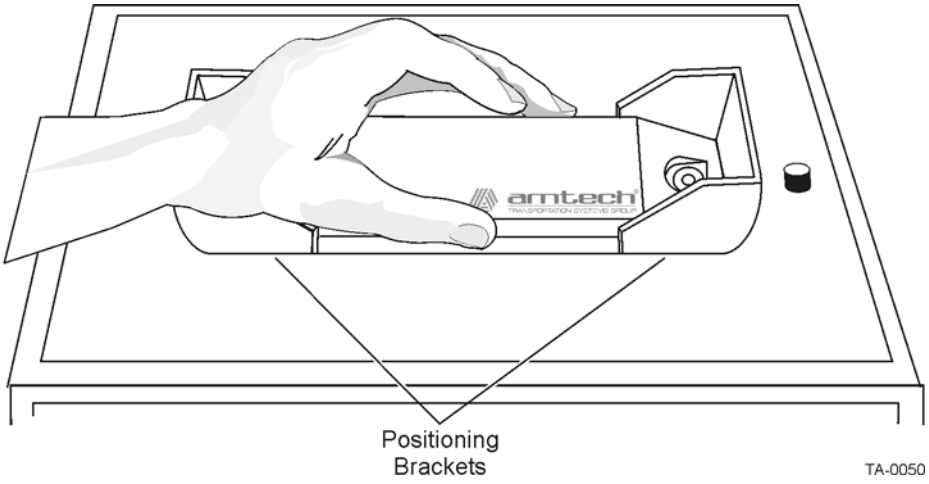

Figure 4-10 Tag in Proper Position

5. A series of quick tones signaling successful data transfer to the tag accompanied by the message `Tag Code Verified` indicate that the tag data displayed on the screen has been successfully programmed into the tag.
6. Remove the tag from the programming head. If the **Labels On** menu item 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, select **Exit** from the **Files** drop-down menu.

Sealing the Tag Programming Port

Ensure that the tag programming port is sealed before mounting the tag on any equipment. Seal the tag programming port using the AS8010 Plug Insertion Tool.

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 (Figure 4-8) 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 any 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 corners. Handle with care.

3. Fit an O-ring on the plug (Figure 4-11).

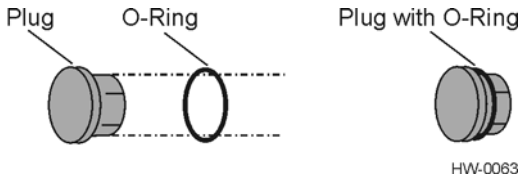

Figure 4-11 Plug and O-Ring

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 4-12).

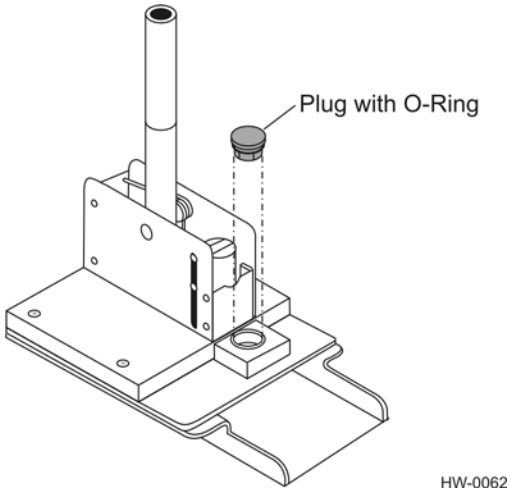

Figure 4-12 Plug in Positioning Plate

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 4-13).

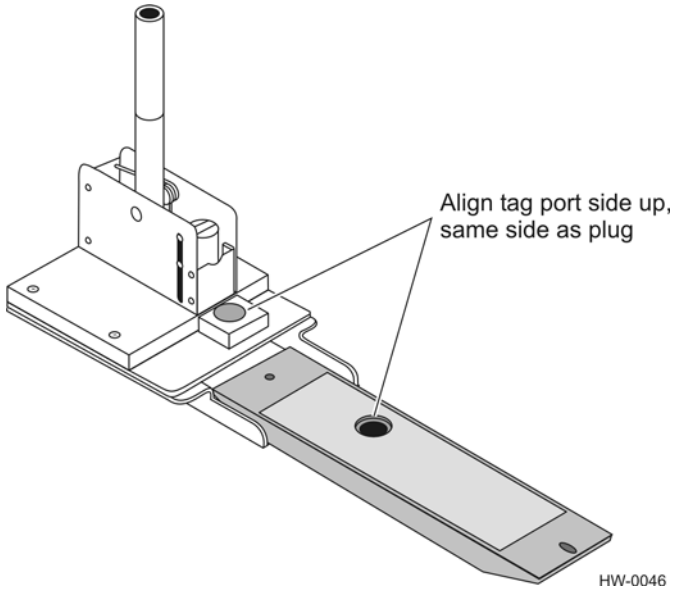

Figure 4-13 Tag in Positioning Tray

6. Slide the tag all the way into the tag positioning tray (Figure 4-14).

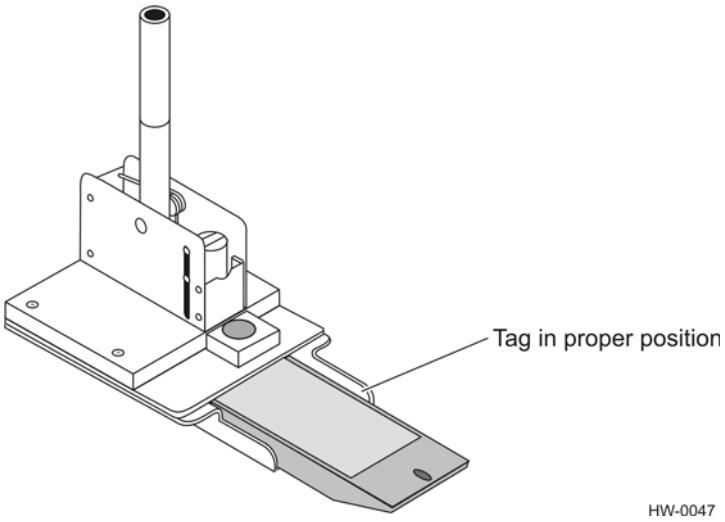

Figure 4-14 Tag Correctly Positioned in Plug Insertion Tool

7. Pull the handle down gently but firmly to insert the plug into the programming port (Figure 4-15).

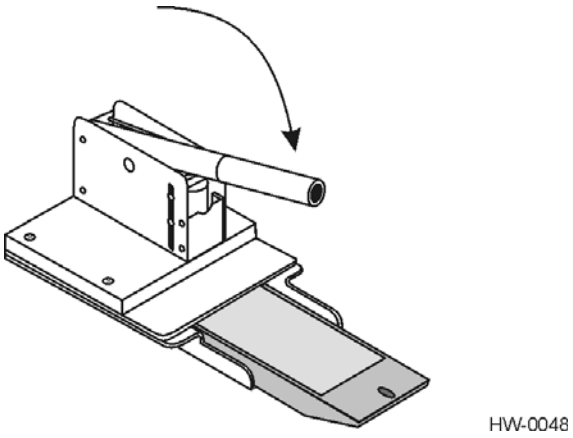

Figure 4-15 Plug Insertion

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

Attaching Tag Labels

If you have set the **Labels On** menu item 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 tone continues 10 times until the tag programming sequence timer has expired.

The following is a list of problems that can cause tag programming errors.

- The tag-to-programming head connection is not secured.
- The data to be transferred is invalid.
- You attempted to program security characters through a programmer without enabling security settings.
- The programmer timed out before the tag was placed on the programming head.

Reading Pre-programmed Tags

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 (Figure 4-8).

Note: *TransCore recommends that you use either the TransCore AS8001 Plug Removal Tool or AS8002 Plug Removal Tool to remove the plug and O-ring.*

2. Select **Read Tag** from the **Tag Functions** drop-down menu. A series of tones indicate 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 4-9).
4. Place the tag between the positioning brackets, and press down firmly and evenly to activate the tag sensing switch (Figure 4-10). If the tag and programming head are not correctly connected, the tone continues to sound 10 times until the timer has expired. 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. The tag data contents are displayed on the TP screen.

5

Using Files

Chapter 5

Using Files

This chapter explains how to create, replace, append, and print the log, output, and input files.

Using Log Files

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

To create a log file

1. Choose **Log File Select** from the **Files** menu to access an existing `.log` file or to create a new file. This action opens the **Select Log File** window (Figure 5-1).

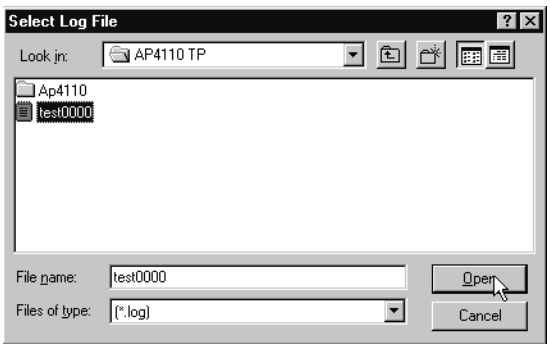

Figure 5-1 Select Log File Window

2. Type a filename in the **File name:** window and select **Open**. To cancel, press **Esc**.
3. See “Transferring Data to Tags” on page 4-11 for complete tag programming instructions.
4. When tag programming is complete, choose **Select Log File** to close the log file and to 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 cannot be altered using the TP software; you may, however, edit the file using standard DOS editing techniques for ASCII files on the PC.

To append or replace log files

5. If you have already created a log file, choose that filename and select **Open**, which displays the log file name in the **Log File** field in the window. To cancel, press **Esc**. Figure 5-2 shows the **Existing Log File Selections** Window.
6. Select **Append Logged Data to Existing File**, **Replace All Data in the Existing File**, or **Cancel Logging Data**. Click **Exit** to close this window.

Figure 5-2 Existing Log File Selection Window

7. Place the tag between the positioning brackets (Figure 5-3) and press down firmly and evenly to activate the tag sensing switch. Select **Program Tag** from the **Tag Functions** drop-down menu or press **F1** to program the tag.

   The tag programmer waits 10 seconds while you position the tag before issuing a time-out error to the TP software. The tag programmer sounds a tone every second while waiting for the tag to be positioned. If the message `Error, Try Again` displays in the message window followed by a long tone, the tag was not placed on the tag programming head before the programmer timed out. Return to Step 7 and program the tag again.

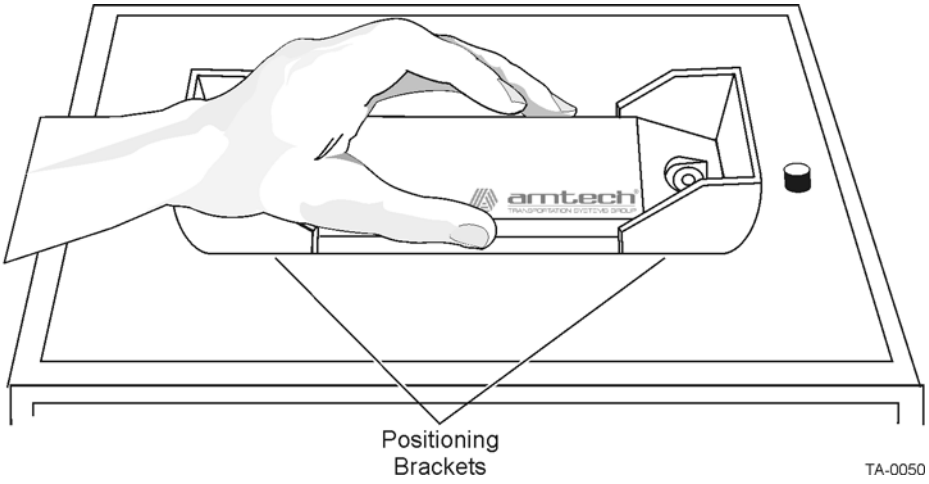

Figure 5-3 Tag in Proper Position

8. A series of quick tones signaling successful data transfer to the tag accompanied by the message `Tag Code Verified` indicate that the tag data displayed on the screen has been successfully programmed into the tag.
9. Remove the tag from the programming head. If the **Labels On** menu item is set to **on**, the tag data prints on the label. Attach the label to the tag.
10. Enter all required data for the next tag and repeat steps 7 through 9.
11. When all tags have been programmed, select **Exit** from the **Files** drop-down menu.When tag programming is complete, choose **Log File Select** to close the log file and to save the contents to the TP directory.

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

Printing Log Files

1. Select **Tag File for Report** from the **Files** menu This action opens the **Select .tag File For Report** window (Figure 5-4).

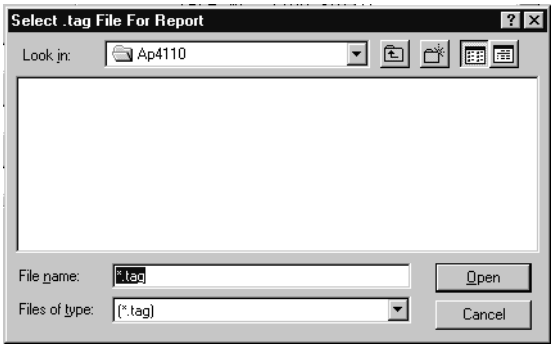

Figure 5-4 The .tag File For Report Window

2. Select the file to be printed from your directory.
3. Select **Open** and the **Print** window displays (Figure 5-5).
4. Select **OK** to print your log file report.

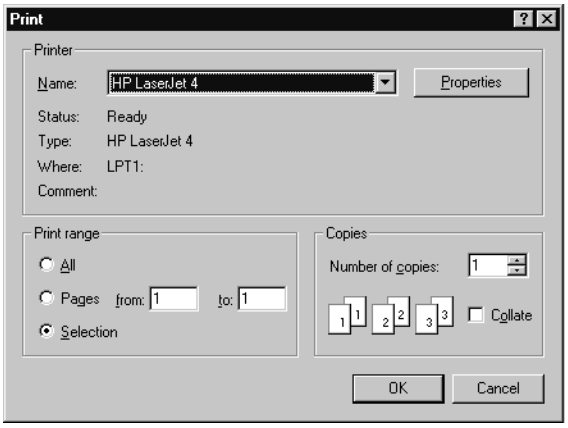

Figure 5-5 Sample Print Window

Using Output Files

This section explains how to perform the following functions:

- Creating an output file
- Modifying output file records
- Deleting output file records
- Restoring deleted output file record
- Purging deleted output file records
- Printing 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 option 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. Output files have a `.tag` extension.

To create an output file

1. Choose **Output File Select** from the **Files** menu to access a `.tag` file. This action opens the **Select Output File** window (Figure 5-6).

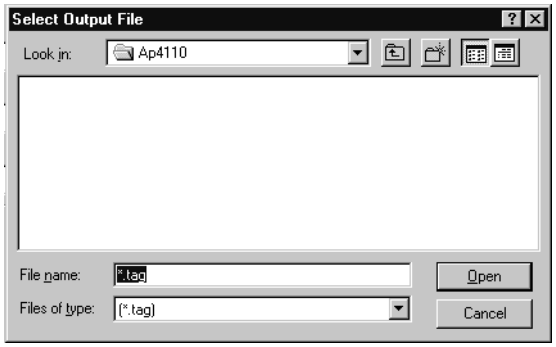

Figure 5-6 Select Output File Window

2. Type the output file name and press **Enter**, which displays the output file name in the **Output File** section (Figure 5-7). To cancel, press **Esc**.

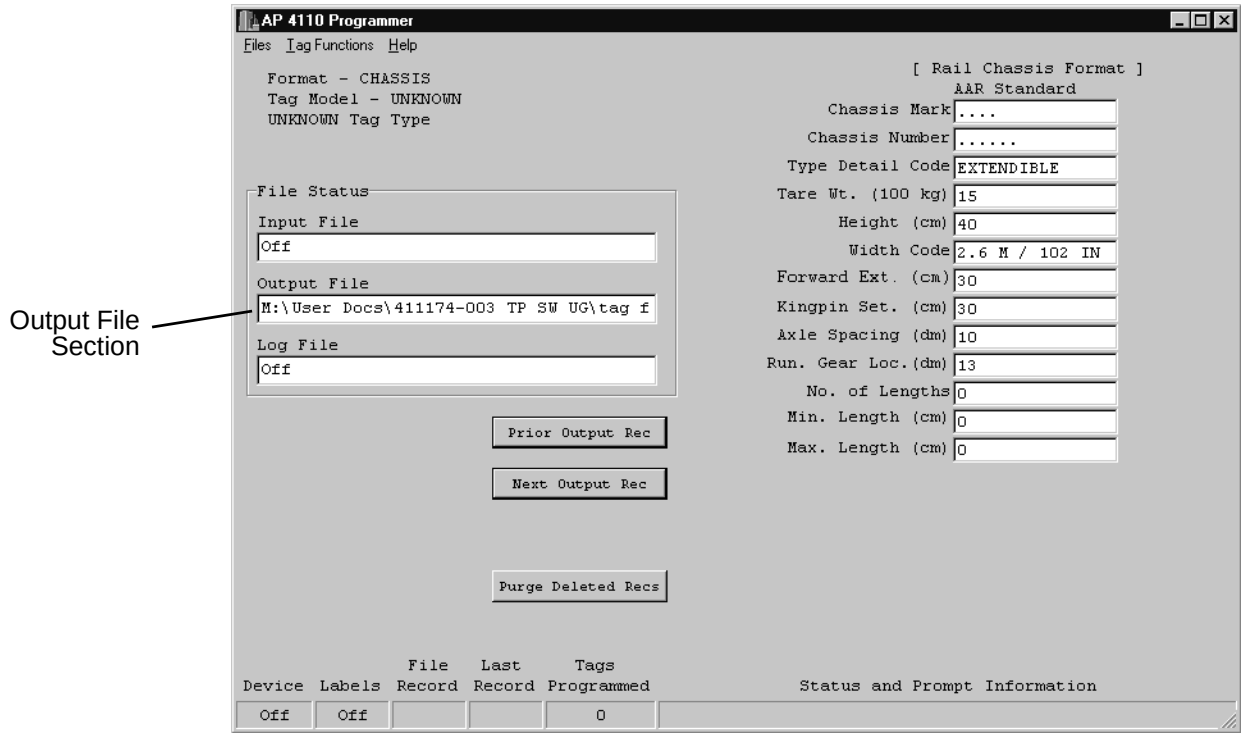

Figure 5-7 Output File Section

For steps 3 through 5 refer to tag format, tag model number, and tag data entry instructions outlined in Chapter 4, pages 4-4 through 4-6.

3. Select the tag format.
4. Enter the tag model number.
5. Enter the tag data in each field.
6. When field data entry is complete, select **Program Tag** from the **Tag Functions** menu to write the current tag record into the output file.
7. When all tag records have been entered, choose **Select Output File** 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. Choose **Output File Select** from the **Files** menu to access a `.tag` file (Figure 5-6).
2. Choose the output file and select **Open**, which displays the output file name in the **Output File** section (Figure 5-7). To cancel press **Esc**.

3. The **Selection of Output File Mode** window appears (Figure 5-8). Choose from the four mode selections listed in Table 5-1.

Table 5-1 Output File Mode Descriptions

Mode	Description
Append Output to Existing Output	New tag data is added to the end of the current output record file.
Replace Entire Output File	All output records are replaced with new ones.
Selectively Replace Output File Records	Individual records only are replaced.
Abort the Selection of the Output File	Cancels the output file modification action and returns you to the **Data Entry** window.

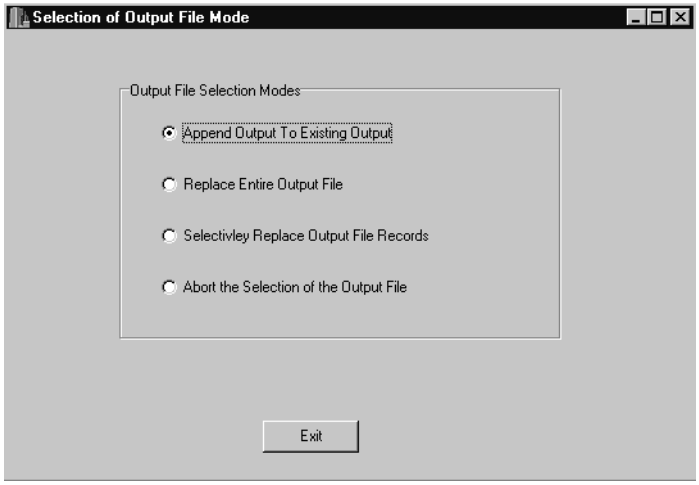

Figure 5-8 Output File Modification Selections

4. Make any desired changes in the record fields.
5. When you have finished making record field changes, you can select another record and make modifications or choose **Select Output File** 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 removal. The deleted record remains in the file until purged using record removal procedures. Refer to “Purging Deleted Output File Records” on page 5-11.

To delete an output file record

1. Choose **Output File Select** from the **Files** menu to access a `.tag` file. This action opens the **Select Output File** window (Figure 5-6).

2. Choose the output file, which displays the output file name in the **Output File** section. Select **Open** to access the file. To cancel press **Esc**.
3. The **Selection of Output File Mode** window appears (Figure 5-8). Choose **Selectively Replace Output File Records** and click on **Exit**.
4. The **Starting Output File Record Number** pop-up window appears (Figure 5-9). You can select the record number you want by typing in the number or using the slide scale arrows to the right of the number box. Click **Accept** when you are ready to proceed. Click **Cancel** to halt the process.

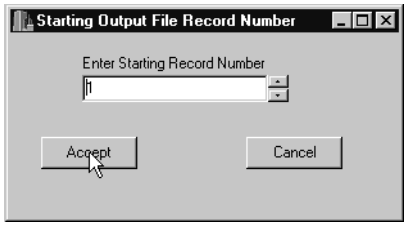

Figure 5-9 Starting Output File Record for Deletion

5. You can then select **Prior Output Rec** to select the previous file record or select **Next Output Rec** to select the next file record (Figure 5-10).

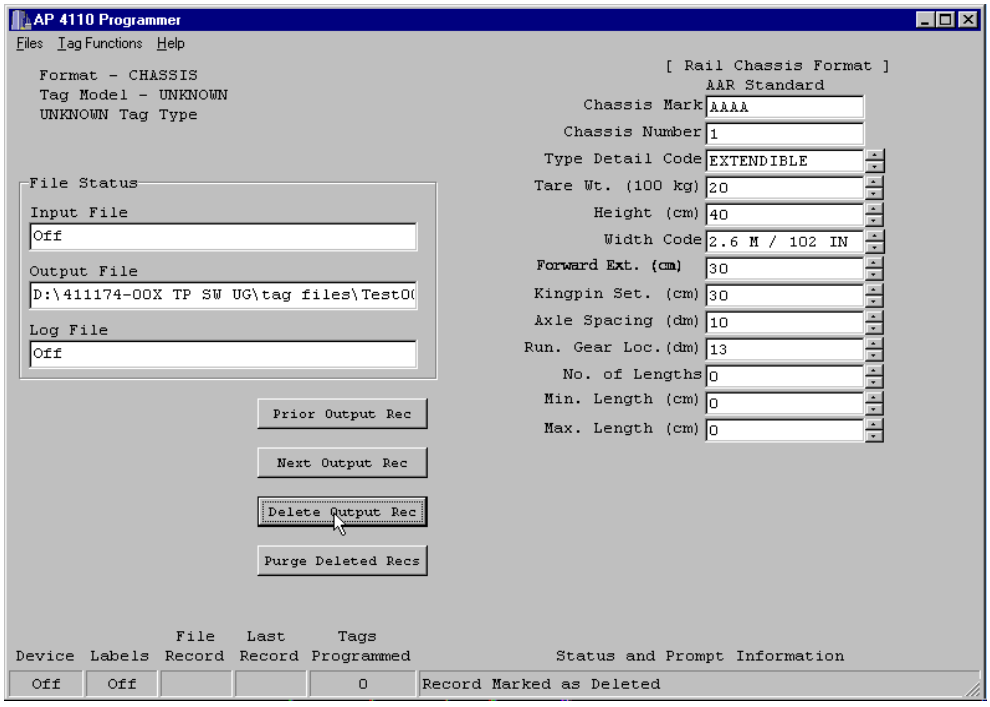

Figure 5-10 Delete Output Record Button

You may select another record for deletion or choose **Output File Select** 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 “Purging Deleted Output File Records” on page 5-11. You may restore deleted output file records before they have been purged or removed.

1. Choose **Output File Select** from the **Files** menu to access a `.tag` file. This action opens the **Select Output File** window (Figure 5-6).
2. Choose the output file and select **Open**, which displays the output file name in the **Output File** section. To cancel, press **Esc**.
3. To display the first record, press **Enter**. You can then select **Prior Output Rec** to select the previous file record or choose **Next Output Rec** to select the next file record (Figure 5-10).
4. You may select another record for restoration or choose **Output File Select** from the **Files** menu to close the output file and save the contents. The output file status is changed to **Off**.
5. You can cancel the record delete action by canceling without saving the output file. Select the **Files** menu and then select **Exit**.

Purging 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 purge deleted output file records

1. Choose **Output File Select** from the **Files** menu to access a `.tag` file. This action opens the **Select Output File** window (Figure 5-6).
2. Choose the output file, which displays the output file name in the **Output File** section. Select **Open** to access the file. To cancel press **Esc**.
3. The **Selection of Output File Mode** window appears (Figure 5-8). You can choose **Append Output to Existing Output, Replace Entire Output File,** or **Selectively Replace Output File Records** to access the **Select Output File** window.
4. Once you have accessed the **Select Output File** window, click on **Purge Deleted Recs** to remove the deleted records (Figure 5-11).

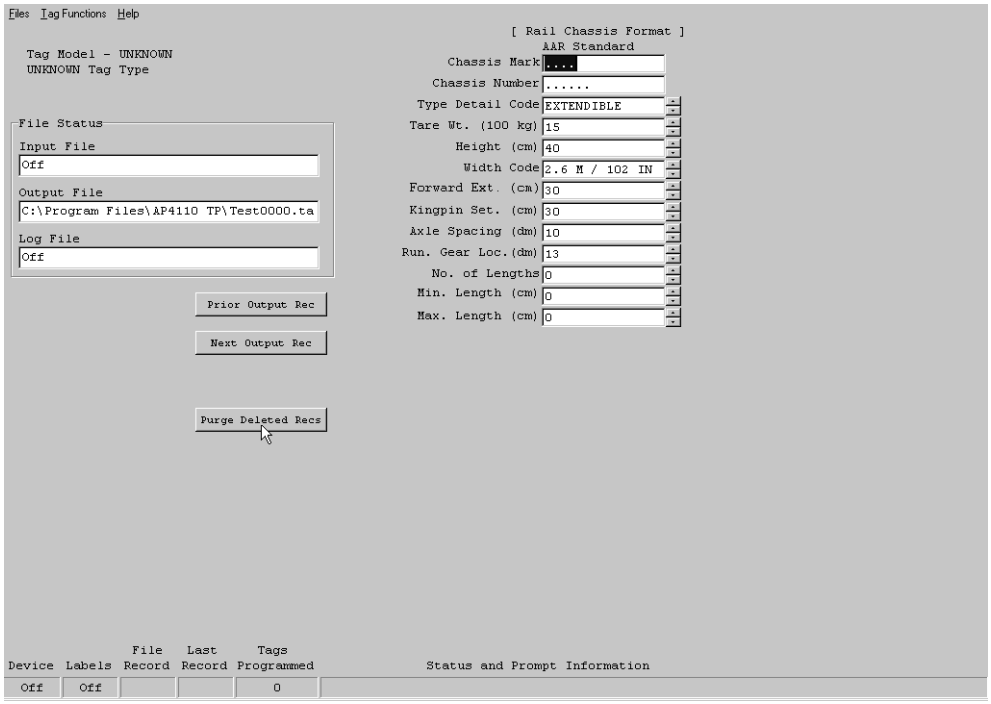

Figure 5-11 Purge Deleted Records Button

5. To display the first record, press **Enter**. You can then select **Prior Output Rec** to select the previous file record, or select **Next Output Rec** to select the next file record (Figure 5-10).
6. You may select another record for deletion or choose **Output File Select** from the **Files** menu to close the output file and save the contents. The output file status is changed to **Off**.
7. Select **Purge Deleted Recs** to remove the deleted records.
8. Choose **Output File Select** from the **Files** menu to close the output file and purge the deleted records. 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. Choose **Select .tag File For Report** from the **Files** menu (Figure 5-12).

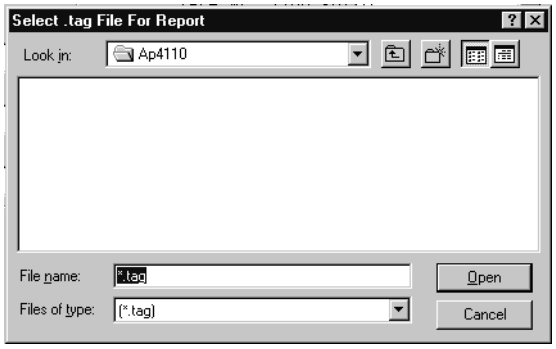

Figure 5-12 Select .tag File For Report Window

2. Select file to be printed from your directory.
3. Select **Open**.
4. Select **OK** to print your tag file report (Figure 5-5).

Using Input Files

You use input files, which also serve as output files, to enter previously recorded tag data.

Programming Tags Using Input Files

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

To program tags using input files

1. Choose **Input File Select** from the **Files** menu to access a `.tag` file. This action opens the **Select Input File** window (Figure 5-13).

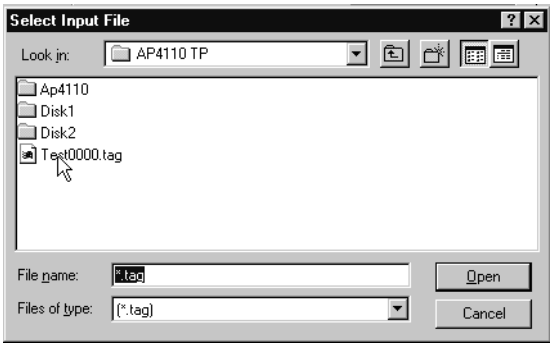

Figure 5-13 Select Input File Window

2. Select your input file and press **Open**.
3. To select the first record in the file, press **Enter**. If you are resuming the programming of tags from a previous session and need to select a tag record other than the first, select **Next Input Rec** to navigate through the tag records.
4. Select **Program Tag** from the **Tag Functions** drop-down menu 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.
5. Place a tag on the programming head, aligning the tag's programming port (Figure 5-14) with the pins on the contact block (Figure 5-15).

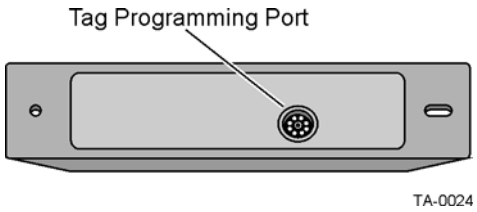

Figure 5-14 Tag Showing Programming Port

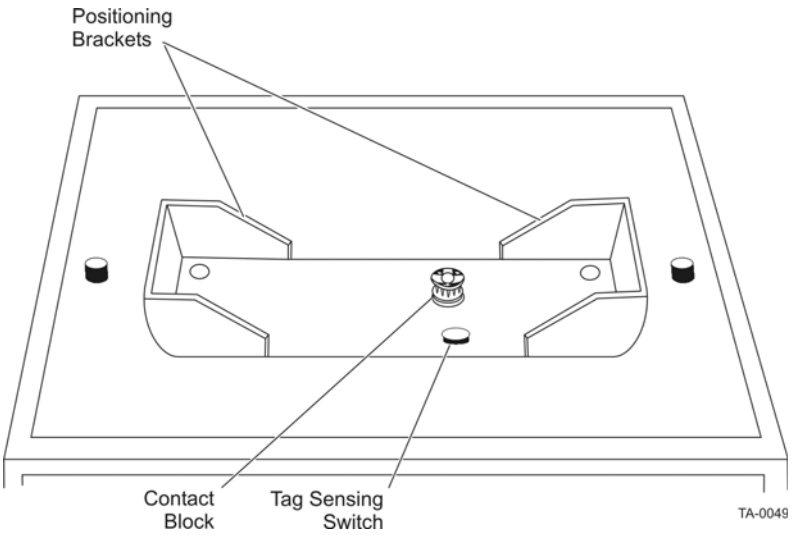

Figure 5-15 Tag Programming Head

6. Place the tag between the positioning brackets, and press down firmly and evenly to activate the tag sensing switch (Figure 5-16).

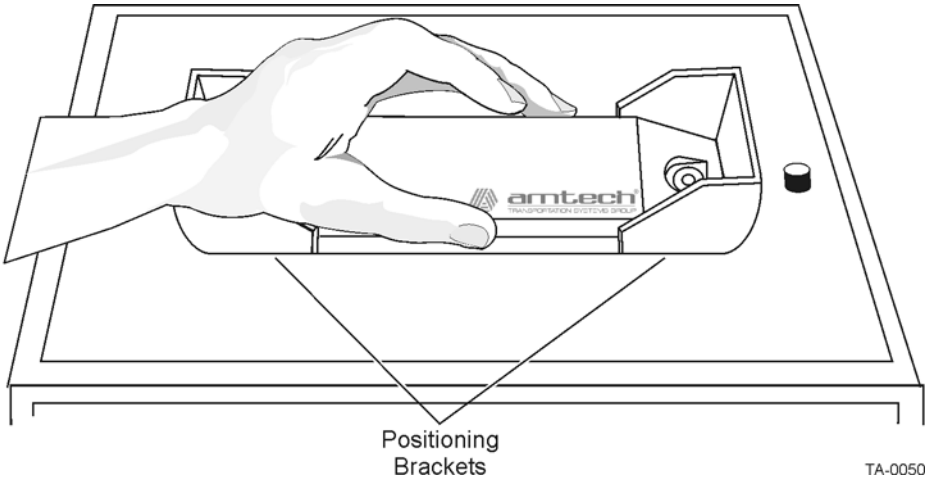

Figure 5-16 Tag in Proper Position

A series of quick tones signal successful data transfer to the tag and displays the message `Tag code verified`, which 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 of the change.

7. Remove the tag from the programming head.

   You may seal the tag programming port using the plug insertion tool or insert the plug manually. TransCore highly recommends 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 4-13.

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

9. Repeat steps 5 through 8 to program any remaining tags.

10. When tag programming is completed, choose **Input File Select** from the **Files** menu 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 field 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. Select **Tag File Report** from the **Files** menu (Figure 5-12).
2. Select the file to be printed from your directory.
3. Select **Open**.
4. Select **OK** to print your tag file report (Figure 5-5).

Figure 5-17 shows a sample tag file report.

Tag File Report Date: 3/26/02

File: C:\AP4110\Test0000.tag Time: 10:17

Page: 1

Rec #	Tag Model	Tag Format	Tag Contents
000001		CHASSIS	27 2 AAAA 1 Extendible 20 40 2.6 m / 102 in 30 30 10 13 0
000002		CHASSIS	27 2 BBBB 2 Extendible 15 40 2.6 m / 102 in 30 30 10 13 0
000003		CHASSIS	27 2 CCCC 3 Extendible 15 40 2.6 m / 102 in 30 30 10 13 0
000004		CHASSIS	27 2 DDDD 4 Extendible 15 40 2.6 m / 102 in 30 30 10 13 0
000005		CHASSIS	27 2 EEEE 5 Extendible 15 40 2.6 m / 102 in 30 30 10 13 0
000006		RAIL_EOT	6 2 FFFF 6 0 Left 0 0 NA S
000007		RAIL_EOT	6 2 GGGG 7 0 Left 0 0 NA S
000008		RAIL_EOT	6 2 HHHH 8 0 Left 0 0 NA S
000009		RAIL_EOT	6 2 IIII 9 0 Left 0 0 NA S
000010		RAIL_EOT	6 2 JJJJ 10 0 Left 0 0 NA S
000011		LOCOMOTV	5 2 KKKK 11 Left 0 1 0 0 NA S
000012		LOCOMOTV	5 2 LLLL 12 Left 0 1 0 0 NA S
000013		LOCOMOTV	5 2 MMMM 13 Left 0 1 0 0 NA S
000014		LOCOMOTV	5 2 NNNN 14 Left 0 1 0 0 NA S
000015		LOCOMOTV	5 2 OOOO 15 Left 0 1 0 0 NA S
000016		RAIL_CAR	19 2 PPPP 16 Left 0 1 0 0 0 0 0 0 NA S
000017		RAIL_CAR	27 2 QQQQ 1 Left 0 1 0 0 0 0 0 0 NA S
000018		RAIL_CAR	27 2 RRRR 1 Left 0 1 0 0 0 0 0 0 NA S
000019		RAIL_CAR	27 2 SSSS 1 Left 0 1 0 0 0 0 0 0 NA S
000020		RAIL_CAR	27 2 TTTT 1 Left 0 1 0 0 0 0 0 0 NA S
000021		CHASSIS	27 2 AAAA 1 Extendible 20 40 2.6 m / 102 in 30 30 10 13 0
000022		CHASSIS	27 2 AAAA 1 Extendible 20 40 2.6 m / 102 in 30 30 10 13 0
000023		CHASSIS	27 2 AAAA 1 Extendible 25 40 2.6 m / 102 in 30 30 10 13 0

Figure 5-17 Sample Tag File Report

A

Record Formats for TP Input/Output Files

Appendix A

Record Formats for TP Input/Output Files

This appendix explains the TP input/output data record formats for each supported equipment type required by the AAR, ATA, ISO, ANSI, and IATA standards.

Preparing Record Input Files

The TP software accepts data input from files as well as from the menu selections and keyboard entries. Using input data files can improve the tag programming procedure by reducing potential data entry errors.

The input files can be created either as a TP output file or as files generated by a software program run independently of TP. 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 data files supplied with each TP software distribution include the descriptions of the formats. By storing these variable record descriptions in data files, TP software can support new formats without program modification.

The input file name must consist of at least one, but no more than eight characters. The file name extension must 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, which 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 that 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	Minimum	Maximum	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
Filler	8	Spaces	Spaces	19-26
Tag Type	1	2	2	27

Table A-2 Rail Car Record Layout (continued)

Field Name	Length	Minimum	Maximum	Position
Filler	8	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
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
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
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
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	Minimum	Maximum	Position
Tag Model Number	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
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
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
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
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	Minimum	Maximum	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
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	Minimum	Maximum	Position
Tag Model Number	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
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	Minimum	Maximum	Position
Filler	8	Spaces	Spaces	59-66
Tare Weight (100 kg)	2	00 or 15	77	67-68
Filler	8	Spaces	Spaces	69-76
Height (cm)	3	000 or 040	166	77-79
Filler	7	Spaces	Spaces	80-86
Tandem Width Code	1	0	3	87
Filler	9	Spaces	Spaces	88-96
Forward Extension (cm)	3	000 or 030	154	97-99
Filler	7	Spaces	Spaces	100-106
Kingpin Setting (cm)	3	000 or 030	154	107-109
Filler	7	Spaces	Spaces	110-116
Axle Spacing (dm)	2	00 or 10	40	117-118
Filler	8	Spaces	Spaces	119-126
Running Gear Locat. (dm)	2	00 or 13	43	127-128
Filler	8	Spaces	Spaces	129-136
Number of lengths	1	0	7	137
Filler	9	Spaces	Spaces	138-146
Minimum Length (cm)	4	0000	2046	147-150
Filler	6	Spaces	Spaces	151-156
Spare	1	0	3	157
Filler	9	Spaces	Spaces	158-166
Maximum Length (cm)	4	0000	2046	167-170
Filler	6	Spaces	Spaces	171-176
Industry Code	1	S	S	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	Minimum	Maximum	Position
Tag Model Number	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
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
Filler	9	Spaces	Spaces	58-66
Length	3	000	510	67-69
Filler	7	Spaces	Spaces	70-76
Number of Axles	2	01	32	77-78
Filler	8	Spaces	Spaces	79-86
Bearing Type	1	0	7	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 (continued)

Field Name	Length	Minimum	Maximum	Position
Industry Code	1	S	S	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	Minimum	Maximum	Position
Tag Model Number	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
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
Filler	9	Spaces	Spaces	58-66
Length (cm)	4	0000	2000	67-70
Filler	6	Spaces	Spaces	71-76
Height (cm)	3	000	500	77-79
Filler	7	Spaces	Spaces	80-86
Width (cm)	3	200	300	87-89
Filler	7	Spaces	Spaces	90-96

Table A-6 Intermodal Container Record Layout (continued)

Field Name	Length	Minimum	Maximum	Position
Container Type	3	000	127	97-99
Filler	7	Spaces	Spaces	100-106
Max. Weight (100 kg)	3	045	455	107-109
Filler	7	Spaces	Spaces	110-116
Tare Weight (100 kg)	2	00	91	117-118
Filler	8	Spaces	Spaces	119-126
Spare	1	SPACE	3	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
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	Minimum	Maximum	Position
Tag Model Number	6	6 spaces	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
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	Minimum	Maximum	Position
Identification Number	8	00000001	ZZZZZZZZ	47-54
Filler	2	Spaces	Spaces	55-56
Length (cm)	4	0000	2047	57-60
Filler	6	Spaces	Spaces	61-66
Width Code	1	0	3	67
Filler	9	Spaces	Spaces	68-76
Tandem Width Code	1	0	3	77
Filler	9	Spaces	Spaces	78-86
Type Detail Code	2	00	15	87-88
Filler	8	Spaces	Spaces	89-96
Forward Extension (cm)	3	000 or 030	284	97-99
Filler	7	Spaces	Spaces	100-106
Tare Weight (100 kg)	3	000 or 015	141	107-109
Filler	7	Spaces	Spaces	110-116
Height (cm)	3	000	511	117-119
Filler	7	Spaces	Spaces	120-126
Industry Code	1	S	S	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	Minimum	Maximum	Position
Tag Model Number	6	6 spaces	AT9999	1-6
Format Name	7	TRACTOR	TRACTOR	7-13
Filler	3	Spaces	Spaces	14-16

Table A-8 Tractor Record Layout (continued)

Field Name	Length	Minimum	Maximum	Position
Equipment Code	2	17	17	17-18
Filler	8	Spaces	Spaces	19-26
Tag Type	1	2	2	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 Axles	1	0	7	57
Filler	9	Spaces	Spaces	58-66
Tare Weight (100 kg)	3	000	255	67-69
Filler	7	Spaces	Spaces	70-76
Wheelbase (dm)	2	00 or 26	64	77-78
Filler	8	Spaces	Spaces	79-86
5th Wheel Offset (dm)	1	0	8	87
Filler	9	Spaces	Spaces	88-96
Tare Weight on Steering Axle (100 kg)	2	0 or 20	50	97-98
Filler	8	Spaces	Spaces	99-106
Drive Axle Spread (dm)	2	00	26	107-108
Filler	8	Spaces	Spaces	109-116
Reserved	1	0	0	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
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 (EOT) record layout.

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

Field Name	Length	Minimum	Maximum	Position
Tag Model Number	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
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
Filler	9	Spaces	Spaces	58-66
Side Indicator	1	0 (left)	1 (right)	67
Filler	9	Spaces	Spaces	68-76
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
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	Minimum	Maximum	Position
Tag Model Number	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
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
Filler	9	Spaces	Spaces	48-56
ID Number Code	4	0000	9999	57-60
Filler	6	Spaces	Spaces	61-66
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
Filler	9	Spaces	Spaces	98-106
Filler	210	Spaces	Spaces	107-316

B

Creating Tag Formats

Appendix B

Creating Tag Formats

This appendix explains the function of format files.

Overview

The core of the TP software is a collection of files called format files, which define the organization of tag data as defined by standards-setting bodies. The format files included with the TP software are current with the various standards set by such agencies as the AAR, ATA, ISO, ANSI, and IATA. These formats are stringently tested and verified for compliance 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:
Use caution when creating or modifying TP format files. When creating format files for TP, it is extremely important that you make no mistakes. If you make a mistake in a format file, the tags will be encoded incorrectly by TP. If these incorrectly coded tags are installed in the field, a considerable expense will be incurred in uninstalling the tags and returning them for new encoding.

TransCore recommends that any newly created or modified format is tested thoroughly. Each bit must be accounted for correctly. Creating or modifying any format file is performed at your own risk.

Any screen editor that handles simple character data may be used to create or modify format files. TransCore recommends that you do not use word processing software to create or modify format files. If you do choose 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 one to eight 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. TransCore strongly discourages using a word processor to edit the TP.INI file.

TP.INI File Layout

The TP.INI file has a fixed layout. You should adhere to this layout exactly. The TP.INI file is a comma-delimited ASCII file, which means that each field is separated by a comma. Terminate each record with a carriage return (CR character, ASCII 13). Most text editors append a carriage return at the end of every record. Figure B-1 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
```

Figure B-1 Standard TP.INI File

TP.INI Field Descriptions

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

Table B-1 TP.INI File Fields and Descriptions

Field	Description
Format Name	This field must contain the name of the specific format file to which the record pertains, and 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 uppercase characters.
Format Description	This field contains a brief description of the format for which the entry stands. This field may be entered in both uppercase and lowercase characters.
Industry Code Indicator	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 uppercase.
Equipment Identification Code	This field 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. A text editor such as Microsoft Notepad will supply this EOF mark automatically.*

Creating a New Format File

Format files are used to add and modify tag encoding schemes without necessitating software modification to TP. You should exercise care when 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 that cannot be decoded.

You can create or modify format files using any screen editor that handles simple character data. TransCore recommends that you do not use word processing software to add or modify table files. If you do choose to use a 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 one to eight 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 identifies the attributes of the different fields required for the particular tag encoding scheme. The field default values section addresses the non-modifiable values of certain fields, as well as the processing of tables and special fields. Both sections are contained in the same file. Each record in both sections must be terminated by a carriage return (CR, ASCII 13). Most text editors will automatically place a CR at the end of each record. Each record is a comma-delimited record, which means that a comma separates each field value.

Figure B-2 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-2 Sample Tag Format File for Locomotives

Field and Screen Definition

The field and screen definition section (Figure B-2) 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 processes each field.

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

The *DONE* record that is located approximately halfway down the file designates the end of the first section. This important record indicates to the TP software that all fields have been defined.

Field and Screen Definition Entries

All records between the format name record (record 1) and the first *DONE* record have the same record layout. This layout consists of 12 entry fields, which are defined as follows:

Field Name consists of one to eight characters, all uppercase. The name must be unique for each field within the same format file.

Field Type consists of one to five characters that 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 uppercase (only six bits)
- Z Alphanumeric uppercase (only six bits and seldom used)
- P Protected field, no entry allowed
- X Alphanumeric, uppercase and lowercase (seldom used)
- C Check digit
- D For display only to screen

The second through fifth characters designate how the field is 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 to centimeters

K Convert pounds or ounces to kilograms

S Alphanumeric uppercase (only six bits)

! Numeric entry only

C Check digit

1 – 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:

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 character has a format-specific function requiring special table file construction. Refer to “Appendix C - Constructing and Using Table Files.”

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

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

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

C2 Calculate check digit based on two previous fields.

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

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

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

Number of Bits	lists the total number of bits this field occupies in the tag. Value must be greater than 0 and less than 61. The only exception is a field that is display only, such as the *Format* and *Format2* fields. These fields can contain 0 since they are not stored in the tag.
Minimum Value	contains the minimum value allowed. The applicable standard document establishes the minimum value.
Maximum Value	contains the maximum value allowed. The applicable standard document establishes the maximum value.
Field Offset	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 two 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	contains the value of the row on the screen where the input prompt is displayed and the input value is accepted.
Input Column	contains the value of the column on the display row as defined above where the input isaccepted.
Index	is reserved for future use. The *Format* and *Format2* fields should contain a 0. All other fields should contain a 1.
Field Length	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	contains the value for the column on the display row, as described above, where the input prompt will display.
Input Prompt Text	contains the actual input prompt text, enclosed in double quotation marks, which 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 has 128 bits, but only 120 bits are user-programmable. The various standards documents reference the following non-programmable fields:

First Checksum	Bits 60 - 61
Reserved Frame Marker	Bits 62 - 63
Second Checksum	Bits 124 - 125
Frame Marker	Bits 126 - 127

Amtech® tag programmers and readers process these fields. 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 use all 120 bits, spare fields should be used for the remaining bits. If the format does not use all 120 bits, TP displays 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

The records in the format file that follow the first *DONE* record comprise the field default values. At least one record must be in the field default values to match each record discussed in “Field and Screen Definition” on page B-8. No more than one record can be in the field default values for each record in the field and screen definition if the field is intended to be a scrollable table. A scrollable table is a field with a fixed number of values. Examples are the side indicator (SID) in the locomotive format file example (Figure B-2) or the type detail code in the chassis format file. If a field is set up as a scrollable table, TP loads values into the field by the user pressing the right or left arrow keys.

Field Default Values Fields

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

Field Name — 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 SID example above.

Initial Display Value — 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 — 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 — 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-2, the initial display value field for the *Format* and *Format2* records contains a blank. Any field that is for display

only will have a *D* in the field type in the field and screen definition record and should contain only a blank in the initial display value field.

Any field that is not to be displayed will have an *I* as one of the characters in the field type in the field and screen definition record and must contain the value in the initial display value field that 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-2. In this example, *0* is to be stored for the Reserved field, *NA* for Security, and *S* for Tformat. The Tformat value in the initial display value field 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

Constructing and Using Table Files

Appendix C

Constructing and Using Table Files

This appendix explains the construction and usefulness of table files to tag programming.

Overview

The TP software 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 while saving time and reducing data entry errors when the tags are actually programmed. There are two types of input disk files for the TP software: (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 the TP software 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.

Constructing Table Files

You may create or modify table files using any screen editor that uses simple character data. TransCore recommends that you do not use word processing software to add or modify table files. If you do use word processing software as an editor, be careful that it does not insert control characters such as tabs, page breaks, margin settings, and printer controls. The TP software 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 four-character owner identification or equipment initial code, (e.g., AMTC [Amtech]), 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 Locomotives
AMTC06.TBL	Amtech Rail End-of-Train Devices
AMTC10.TBL	Amtech Intermodal Containers
AMTC14.TBL	Amtech Non-revenue Rail Cars
AMTC17.TBL	Amtech Tractors
AMTC19.TBL	Amtech Rail Cars
AMTC20.TBL	Amtech Dollies
AMTC21.TBL	Amtech Trailers
AMTC27.TBL	Amtech Chassis

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, EOT 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 by 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 depend upon the format being used as described in Appendix A, “Record Formats for TP Input/Output Files." There can be as many of these records as needed to cover the different ranges of equipment attributes.

Caution

You cannot carry over the beginning and ending identification numbers from one record to another, and you cannot duplicate ranges.

Creating a Sample Table File

To create a valid rail car table file for equipment initial code *TEST,* perform the following tasks:

1. Use a text editor to create a file named TEST19.TBL.
2. Enter the first record (header record). Table C-1 shows the record.

Table C-1 Rail Car Table File Example

Field Description	Field Value
The standard equipment initial code	TEST
Comma	,
Actual format file name without the `.FMT` extension	RAIL_CAR
Comma	,
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 shows a sample rail car table record.

Table C-2 Rail Car Table Record Example

Field Description	Field Value
Starting range identification number	000001
Comma	,
Ending range identification number	000099
Comma	,
Side indicator	0
Comma	,
Length in decimeters	1025
Comma	,
Number of axles	06
Comma	,
Bearing type	1
Comma	,

Table C-2 Rail Car Table Record Example (continued)

Field Description	Field Value
Platform Code	01
Comma	,
Spare 1 value	00
Comma	,
Spare 2 value	0000
Comma	,
Spare 3 value	000
Comma	,
Reserved value	000
Comma	,
Security (usually *NA*)	NA
Comma	,
Data format code or industry code	S

4. Repeat this record entry until you have entered all of the table file records.

Each record must be terminated with a carriage return (CR, ASCII 13). A text editor will place the CR at the end of the record automatically. An end-of-file (EOF) mark must be placed after the last record in the table file. A 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 Entries" section of Appendix B, "Creating Tag Formats."

Once you have modified the format file, rerun the TP software. When you enter the four-character owner identification or equipment initial code and identification number in the **Tag Format** window you cause the TP software to search for the matching

table file. If this table file is found, the TP software searches the file for the range of identification numbers that include the entered number. If the range is found, the TP software 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 occur within any of the ranges in the table file records, the message

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

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

Using Table Files

The following rules govern the use of table files:

- Table files can only be used for formats that contain a four-character alphanumeric owner identification or equipment initial code and a numeric identification number, such as chassis, intermodal containers, locomotives, rail cars, rail EOT devices, and non-revenue rail cars. Formats for the various types of equipment that mandate alphanumeric identification numbers, such as follies, tractors, trailers, and unit load devices, cannot 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, which 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 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 uppercase letters delimited by commas.
- Each record within the table file must be terminated with a carriage return (CR, ASCII 13).
- An EOF mark must terminate the table file.

Caution

You can use table files to save time and prevent errors during tag information data entry; however, if you construct the table file with erroneous information, the tags will encode incorrectly. You should exercise caution when developing and using 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 uninstallation, re-encoding, and reinstallation of the tag. Valuable data can also be lost if tags are encoded incorrectly and installed in the field. Developing and using table files are to be performed at your own risk.

Index

Index