

*Research and Test Department*

---

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP)

USER'S MANUAL

Report No. R-740

G. W. Stephens

---

*Washington Systems Center*



ASSOCIATION  
OF AMERICAN  
RAILROADS

**Association of American Railroads  
Research and Test Department**

**RAIL ENERGY COST ANALYSIS PACKAGE (RECAP)**

**USER'S MANUAL**

**Report No. R-740**

**G. W. Stephens**

**September 1989**

**Washington Systems Center  
Washington, D. C.**

DISCLAIMER

This report is disseminated by the AAR for informational purposes only and is given to, and accepted by, the recipient at its sole risk. The AAR makes no representations or warranties, either express or implied, with respect to the report or its contents. The AAR assumes no liability to anyone for special, collateral, exemplary, indirect, incidental, consequential or any other kind of damage resulting from the use or application of this report or its content. Any attempt to apply the information contained in this report is done at the recipient's own risk.

1. REPORT NO. R-740	2. REPORT DATE September 1989	3. PERIOD COVERED
4. TITLE AND SUBTITLE Rail Energy Cost Analysis Program (RECAP) User's Manual		
5. AUTHOR(S) G. W. Stephens, Assistant Manager, Engineering Economics, AAR		
6. PERFORMING ORGANIZATION NAME AND ADDRESS Association of American Railroads Washington Systems Center 50 F Street NW Washington, D. C. 20001	7. TYPE OF REPORT Research Report	8. CONTRACT OR GRANT NO.
9. SPONSORING AGENCY NAME AND ADDRESS Association of American Railroads Washington Systems Center 50 F Street NW Washington, D. C. 20001	10. NO. OF PAGES 103	11. NO. OF REFERENCES 1
12. SUPPLEMENTARY NOTES Engineering Economics Division		
13. ABSTRACT  <p>This report describes the capabilities of the Rail Energy Cost Analysis Program (RECAP) and serves as a User's Manual providing the detailed information necessary to set up the input files to run RECAP and interpret the results. RECAP is an engineering economics model designed to be used in combination with a physical simulator of train operations, the Train Energy Model (TEM), to estimate the cost of providing rail transportation on a specific route and service basis. The TEM/RECAP combination is designed to aid individual railroads in evaluating alternative technologies and operating strategies.</p> <p>RECAP calculates the incremental cost of running a logical set of trains; for example, a loaded and empty return movement of a unit train, an intermodal train round trip, or any combination of train movements required to provide a specific freight service.</p> <p>RECAP in combination with TEM provides rail transportation planners a state-of-the-art tool to assess a wide range of options in providing rail transportation. RECAP is unparalleled in its ability to translate detailed specification of the technical capabilities of new equipment and operational procedures into economic and managerial terms.</p>		
14. SUBJECT TERMS Rail Cost Analysis Cost Simulation Models Technology Economic Assessment Engineering Cost Analysis Route-Specific Cost Analysis	15. AVAILABILITY STATEMENT Document Distribution Center Association of American Railroads Technical Center 3140 South Federal Street Chicago, Illinois 60616	

## EXECUTIVE SUMMARY

The Rail Energy Cost Analysis Program (RECAP) is an engineering economics model designed to be used in combination with a physical simulator of train operations, the Train Energy Model (TEM), to estimate the route and service specific incremental costs of providing rail transportation. It is designed to aid individual railroads in evaluating operating strategies and/or alternative technologies.

RECAP calculates the incremental cost of running a logical set of trains; for example, a loaded and empty movement of a unit train, an intermodal train round trip, or a set of locomotive duty cycles. Any grouping of train operations required to provide a transportation service can be evaluated by RECAP.

Some examples of alternatives that can be analyzed with the combined capabilities of TEM/RECAP include the following: alternative train makeup strategies, rail lubrication, increased axle loads, reduced tare cars, equipment with improved aerodynamic characteristics, improved suspension systems, alternative crew compensation methods, or alternative operating speeds. Combinations of these and other alternatives can be evaluated accurately through the use of TEM/RECAP.

RECAP accepts a logical set of TEM outputs each of which describes the operation of a train over a route according to a specified operating plan. As many individual train operations (TEM outputs) as required to provide a complete transportation service are input to a single RECAP analysis. TEM describes each train operation in terms of the time in route, the fuel consumed, and the equipment used (train consist). The RECAP

economic model uses engineering based relationships to translate physical and financial descriptions of the activities and equipment involved in providing the transportation service into estimates of costs for crew, fuel, equipment and incremental track maintenance for each specific activity and the total service.

### The Advantage of RECAP for Transportation Planning

RECAP gives the transportation analyst on individual railroads an accurate, flexible, engineering based alternative to traditional accounting allocation approaches for determining route and service specific costs. The engineering based approach to estimation of incremental costs has two major advantages over accounting based cost analysis for transportation planning purposes: (1) operations can be evaluated over specific route - service options, and (2) the effects of new technologies and operational options not currently in service can be evaluated from engineering knowledge of the proposed operations and equipment.

An accounting allocation approach to cost analysis uses the railroad's total production process as it existed in the past as the source of information concerning the transformation of inputs into transportation outputs. This production process reflects technology and operations as they existed at that point in time. Knowledge of the way in which new technology or operations should impact the consumption of inputs can not be directly incorporated into the cost accounting estimates.

In addition to not being clearly related to a specific production technology, an accounting allocation approach to cost analysis generates results that are an average of the production process of the entire railroad (or at least a major subdivision

of it). In addition to the items which vary directly with service, such as crews, fuel, etc., the accounting allocation process includes the costs of a portion of such items as train dispatching, signal maintenance, and administrative overhead, which may not be affected by new service or technology. For many activities such as pricing services for single car movements or setting rate limits in regulatory proceedings, accounting allocation may be an appropriate approach to calculating costs, but it is not appropriate for technology assessment, detailed studies of operational alternatives, or other managerial decisions requiring accurate evaluation of specific options.

RECAP is an engineering simulation approach to incremental line haul cost analysis. RECAP permits the user to define and then vary any component of the line haul train operation and to make comparisons as to the effectiveness of any number of alternative operating scenarios and changes in technology such as the introduction of new equipment designs.

#### Uses for RECAP

RECAP is a useful tool for assisting individual railroads to rationally develop and evaluate alternative courses of action for providing a specific transportation service. Some of the applications for which RECAP is well suited include:

- o Assessing alternative technologies (alternative locomotive types, alternative car designs, suspension designs, improved rail lubrication, and improved aerodynamics).
- o Assessing train makeup strategies (horsepower or tractive effort to weight ratios).

- o Evaluating service improvements (engineered service and cost standards, evaluating cost/service trade-offs).
- o Modifying crew compensation agreements or crewing requirements.
- o Evaluating alternative train routes
- o Incremental cost analysis in support of short-term traffic assessment.

The above list of RECAP applications is not exhaustive but is representative of the type of analyses that can be performed. One of the most useful capabilities of RECAP is that options need not be analyzed in isolation. For example, you may wish to compare an alternative service concept which includes increased wheel loads, vehicle mounted rail lubricators and non-traditional crew agreements to conventional train equipment, no rail lubrication, and with a standard intra-divisional crew agreement in effect.

#### Applications for Which RECAP is NOT Well Suited

Since RECAP only calculates the incremental cost of running a set of trains through engineering simulation of the required activities and resulting resource consumptions, it does not provide information relating to costs of other factors that do not vary directly with the operation of the train. For example, RECAP can provide a very good indication of the incremental cost of running high speed, double-stack service using five platform articulated cars in a particular corridor. However, RECAP alone cannot tell you about all the costs you would need to know to

determine the long term viability of that service. In addition to the direct transportation cost of providing the service, which RECAP provides, one would need to know terminal handling costs, and the allocations of the administrative overheads and other indirect costs that must be covered if the service is to be truly profitable in the longer term.

One also must be aware that costs derived via an engineering simulation are not identical in absolute levels to the costs associated with providing a specific service unless a very precise specification of the service plan is achieved. Engineering costing is best at developing incremental cost ratios for the comparison of alternatives rather than the development of absolute cost estimates.

#### Summary

RECAP in combination with TEM provides transportation planners a state-of-the-art tool to assess a wide range of options in providing rail transportation. It is unparalleled in its ability to translate detailed specifications of the technical capabilities of new equipment and operational procedures into economic and managerial terms. The programs are sufficiently general and easy to use to allow the effective comparison of many alternatives. The principal limitation to the application of TEM/RECAP is the imagination of the analyst.



## TABLE OF CONTENTS

EXECUTIVE SUMMARY.....	vii
LIST OF EXHIBITS.....	xv
1.0 INTRODUCTION .....	1
1.1 Overview of TEM/RECAP.....	2
1.2 The Train Energy Model.....	11
1.2.1 How TEM Really Works.....	12
1.2.2 The TEM Advantage.....	13
1.2.3 Running TEM - The User's Point of View...	14
1.3 Track Maintenance Costs.....	16
1.4 Computer Requirements and Installation Procedures.....	19
1.5 Manual Organization.....	20
2.0 THE ECONOMIC MODEL.....	23
2.1 Input Files .....	23
2.2 Input File Descriptions .....	24
2.2.1 The Initialization File (.INI) .....	24
2.2.2 The Stop File (.STP) .....	25
2.2.3 The Route File (.TRK) .....	26
2.2.4 Vehicle Parameter Files (.PAR) .....	27
2.2.5 The Consist File (.CON) .....	30
2.2.6 TEM Output File (.OUT).....	30
2.2.7 Crew Data File (.CRW) .....	31
2.2.8 Crew File Rules and Strategies .....	37
2.2.9 Track Maintenance Cost File (.TMC) .....	39
2.3 Running RECAP (The Economic Model - "ECON2")....	40
2.3.1 The Multiple Simulation File (.RCP) .....	41
2.3.2 The MS-DOS Command Line (C>) .....	43

2.4	Output From The Economic Model .....	47
2.4.1	Input File List .....	47
2.4.2	Crew Costs .....	48
2.4.3	Equipment Costs .....	51
2.4.4	Fuel Costs .....	52
2.4.5	Track Maintenance Costs .....	54
2.4.6	RECAP Cost Summary .....	56
2.5	Changing Operating Strategy w/o Rerunning Simulation.....	57
2.6	Building the Track Maintenance Cost Files .....	60
2.6.1	Use of the Track Maintenance Cost Files..	62
2.6.2	What to Avoid.....	63
3.0	TKSPLICE (Track File Management System) .....	65
3.1	How to Use TKSPLICE .....	65
3.1.1	The MS-DOS Command Line .....	65
3.1.2	The TKSPLICE Command Language .....	67
3.1.3	Example TKSPLICE Command Files .....	69
4.0	TKEDIT (Track File Editor and Verifier) .....	75
4.1	Using TKEDIT .....	75
4.1.1	The MS-DOS Command Line .....	75
4.1.2	TKEDIT Commands .....	76
	4.1.2.1 Moving from Place to Place in the File.....	
	4.1.2.2 Searching the File .....	77
	4.1.2.3 Insertions, Deletions, Copy, and Editing.....	
	4.1.2.4 Exiting TKEDIT .....	79
	4.1.2.5 Other Functions .....	80
	4.1.2.6 Recap of Programmed Function Keys	82
4.1.3	Additional Features .....	83
4.1.4	Programming the Function Keys .....	84
4.1.5	Entering Repetitive Data into Blank Fields .....	85

## RAIL ENERGY COST ANALYSIS PROGRAM (RECAP)

### 1.0 INTRODUCTION

RECAP is a route and service specific engineering economics model. It is designed to aid individual railroads in developing and evaluating alternative operating strategies and/or technologies. Integral to RECAP is a simulator of train performance, the Train Energy Model (TEM). The economic analysis can not be performed by RECAP until the train operations required to provide the service have been simulated with TEM and the required output files have been created.

RECAP calculates the incremental cost of running a logical set of trains. A logical set of trains is defined any way the user wishes to define it. For example, a loaded and empty movement of a unit train, an intermodal train round trip, or a set of locomotive duty cycles. Some examples of alternatives one may wish to use RECAP to analyze include the following: alternative train makeup strategies, rail lubrication, increased axle loads, reduced tare and aerodynamic drag cars, improved suspension systems, alternative crew compensation methods, or combinations of these options.

This RECAP manual is designed to enable the user to understand; (1) the capabilities and limitations of RECAP, (2) the relationship of RECAP to the Train Energy Model (TEM) and the Track Maintenance Cost Model (TMCOST), and (3) the computer structure of RECAP (including the TEM interface) and the files that must be prepared to execute the RECAP program. To conduct a RECAP/TEM analysis, one must also have the Train Energy Model Users Manual (1) to prepare the inputs to the set of TEM runs whose outputs are input to RECAP.

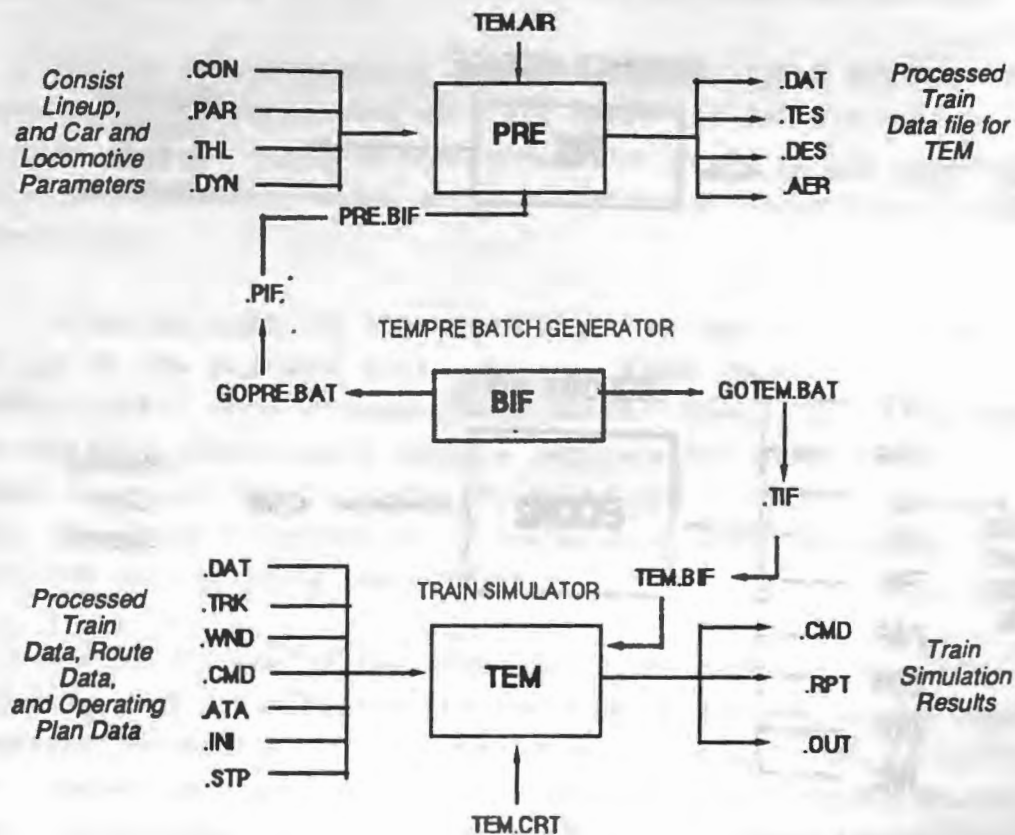
## 1.1 OVERVIEW OF TEM/RECAP

Exhibit 1 shows the basic structure of the TEM/RECAP information flows. The first step in conducting an analysis is to identify the train operations that need to be simulated with TEM. Any change in the train consist or load status of any of the cars in the train triggers the need for a separate TEM simulation. For example, simulating the operation of a single unit coal train service may involve the following steps:

- (1) Leave terminal with 115 empty cars and 5 locomotives on head end of train. Train stops once for a crew change. Arrives at mine and idle cars and locomotives for 4 hours while loading.
- (2) Leave mine with 115 loaded cars and 5 locomotives.
- (3) Stop train and add 2 helper locomotives. Change crews in lead consist. Add 1 helper crew. Stop train at the top of the hill and take off the helper locomotives.
- (4) Continue loaded train without helpers to power plant.
- (5) Run helpers light back down the hill. Helper crew goes off duty.
- (6) Unload train at power plant and return empties back to originating terminal. Idle locomotives and cars for 24 hours before making the next run.

Exhibit 1

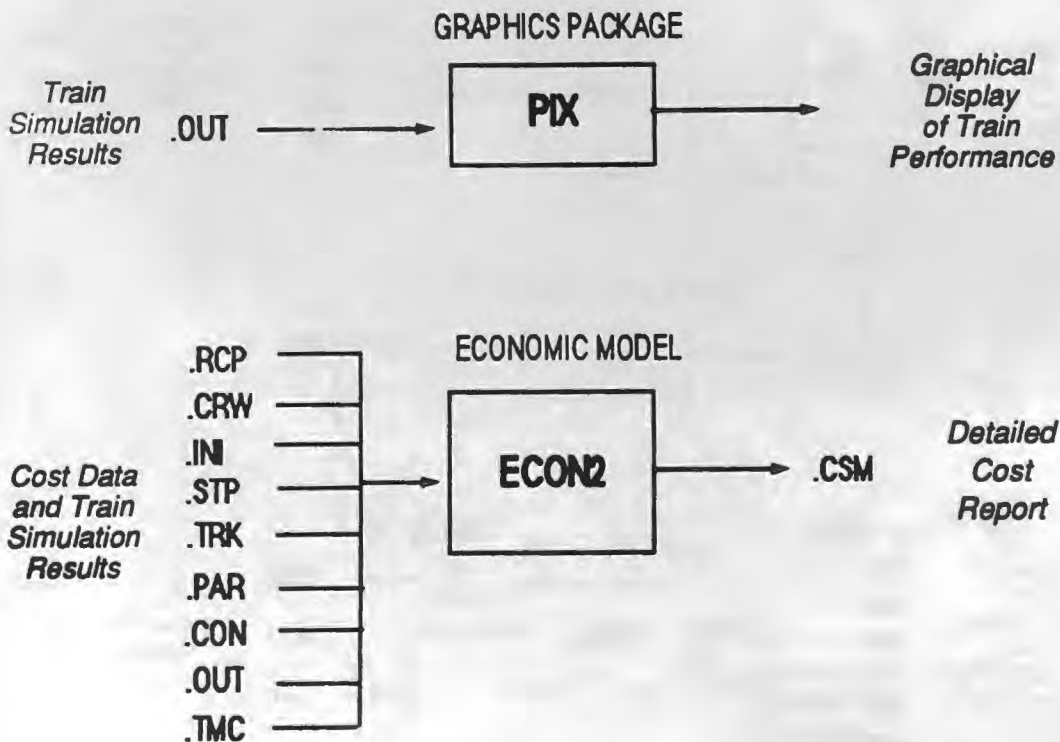
Schematic Diagram of TEM/RECAP Information Flows



File contents defined in Train Energy Model-User's Manual

Exhibit 1 (Continued)

Schematic Diagram of TEM/RECAP Information Flows



File contents defined in Section 2.2 of this manual

Six separate TEM simulations must be run in order to model this unit train operation. A TEM output file containing a detailed record of fuel consumption and running time will be generated for each of those simulations. Satisfactory TEM outputs for each of these six train operations must be completed before proceeding to the economic modeling stage using RECAP.

In the second phase of analysis, RECAP allows the user to generate costs associated with any number of TEM simulations and combine them to estimate the incremental costs of one train service. RECAP can model any combination of individual train operations.

Although separate TEM simulations are required for each change in the physical train make-up, RECAP permits very sophisticated crew-to-train assignments. Crews can be assigned to multiple trains which require separate TEM simulations. RECAP will accurately record the miles and time each crew was on duty for up to a maximum of 50 crews on as many trains as required to represent the service.

In the example of the unit coal train service, the total variable cost of providing this service is the sum of the costs incurred in each of the six simulations plus the cost of tying the locomotives and cars up for a total of twenty-eight hours at the terminal and the mine. The results of a RECAP simulation of this service are shown in Exhibit 2. These outputs include (1) a complete listing of the input files, (2) a summary of the crew assignments and compensations, (3) equipment cost summary by train (individual TEM simulation), (4) fuel consumption summary between stops and overall, (5) track maintenance costs between stations and overall, and (6) an incremental cost summary by cost category and total by train and for the entire service. Note that costs are reported by gross ton mile (GTM), per net

**EXHIBIT 2**  
**Example RECAP output for Unit Coal Train Service**

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988

Page 1

INPUT FILES FOR SIMULATION COALRUN:

07/26/88 09:09am

FILE DESCRIPTION	PATH	NAME	LAST FILE UPDATE
Recap simulation file list	C:\COAL\ (LOADED,EMPTY,HELPER,LIGHT HELPER)	COALRUN.RCP	25 Jul 1988 at 14:45
Crew data file	C:\COAL\	COALRUN.CRW	26 Jul 1988 at 09:49
TEM Initialization file	C:\COAL\ (EMPTY MOVE, TERMINAL TO MINE)	TOMINE.INI	25 Jul 1988 at 13:58
TEM Stop file	C:\COAL\	TOMINE.STP	25 Jul 1988 at 13:58
TEM simulation output file	C:\COAL\	TOMINE.OUT	25 Jul 1988 at 14:25
Consist description file	C:\COAL\	MTY100.CON	25 Jul 1988 at 10:16
Track description file	C:\COAL\	DKTOBC.TRK	25 Jul 1988 at 13:52
Track maintenance cost file	C:\COAL\	EMPTY.TMC	6 Feb 1987 at 10:32
TEM Initialization file	C:\COAL\ (LOADED MOVE FROM MINE)	FRMINE.INI	25 Jul 1988 at 10:54
TEM Stop file	C:\COAL\	FRMINE.STP	25 Jul 1988 at 10:59
TEM simulation output file	C:\COAL\	FRMINE.OUT	25 Jul 1988 at 11:57
Consist description file	C:\COAL\	HOP100.CON	25 Jul 1988 at 10:00
Track description file	C:\COAL\	BCTODK.TRK	25 Jul 1988 at 10:07
Track maintenance cost file	C:\COAL\	UNITL.TMC	7 Feb 1987 at 12:16
TEM Initialization file	C:\COAL\ (LOADED MOVE WITH HELPERS)	HELPER.INI	25 Jul 1988 at 10:55
TEM Stop file	C:\COAL\	HELPER.STP	25 Jul 1988 at 11:00
TEM simulation output file	C:\COAL\	HELPER.OUT	25 Jul 1988 at 12:38
Consist description file	C:\COAL\	HELPER.CON	25 Jul 1988 at 10:03
Track description file	C:\COAL\	BCTODK.TRK	25 Jul 1988 at 10:07
Track maintenance cost file	C:\COAL\	UNITL.TMC	7 Feb 1987 at 12:16
TEM Initialization file	C:\COAL\ (LOADED TO PLANT W/O HELPERS)	PPLANT.INI	25 Jul 1988 at 10:55
TEM Stop file	C:\COAL\	PPLANT.STP	25 Jul 1988 at 11:01
TEM simulation output file	C:\COAL\	PPLANT.OUT	25 Jul 1988 at 12:43
Consist description file	C:\COAL\	HOP100.CON	25 Jul 1988 at 10:00
Track description file	C:\COAL\	BCTODK.TRK	25 Jul 1988 at 10:07
Track maintenance cost file	C:\COAL\	UNITL.TMC	7 Feb 1987 at 12:16
TEM Initialization file	C:\COAL\ (EMPTY RETURN TO TERMINAL)	EMPTY.INI	25 Jul 1988 at 12:50
TEM Stop file	C:\COAL\	EMPTY.STP	25 Jul 1988 at 12:50
TEM simulation output file	C:\COAL\	EMPTY.OUT	25 Jul 1988 at 13:32
Consist description file	C:\COAL\	MTY100.CON	25 Jul 1988 at 10:16
Track description file	C:\COAL\	DKTOBC.TRK	25 Jul 1988 at 13:52
Track maintenance cost file	C:\COAL\	EMPTY.TMC	6 Feb 1987 at 10:32
Initialization file	C:\COAL\ (LIGHT HELPER RETURN)	LITEHP.INI	25 Jul 1988 at 13:53
TEM Stop file	C:\COAL\	LITEHP.STP	25 Jul 1988 at 11:03
TEM simulation output file	C:\COAL\	LITEHP.OUT	25 Jul 1988 at 13:55
Consist description file	C:\COAL\	LITEHP.CON	25 Jul 1988 at 10:03
Track description file	C:\COAL\	DKTOBC.TRK	25 Jul 1988 at 13:52
Track maintenance cost file	C:\COAL\	EMPTY.TMC	6 Feb 1987 at 10:32

**EXHIBIT 2 (CONTINUED)**  
**Example RECAP output for Unit Coal Train Service**

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988  
 CREW COSTS FOR SIMULATION COALRUN:

Page 2  
 07/26/88 09:09am

CREW #	PAYMENT METHOD	TOTAL TIME	CUM. TIME	TOTAL MILES	CUM. MILES	BASIC PAY \$	O-MILE PAY \$	O-TIME PAY \$	ARBIT & OTHER \$	BENEFITS \$	TOTAL COMP \$
1	1. Intra-D	01:59	01:59	90.57	90.57	475.35	0.00	0.00	225.00	252.13	952.48
2	1. Intra-D	10:21	12:21	92.76	183.33	475.35	0.00	210.49	225.00	327.90	1,238.74
3	1. Intra-D	05:25	17:46	184.93	368.26	475.35	323.10	0.00	225.00	368.44	1,391.89
4	2. Inter-D	04:34	22:20	184.45	552.70	475.35	336.48	0.00	225.00	373.26	1,410.08
5	2. Inter-D	04:34	26:54	184.45	737.15	250.35	177.21	0.00	125.00	198.92	751.48
6	1. Intra-D	03:09	30:03	109.68	846.83	475.35	7.05	0.00	225.00	254.66	962.07
7	1. Intra-D	03:09	33:12	109.68	956.51	250.35	3.69	0.00	125.00	136.46	515.50
8	1. Intra-D	07:05	40:18	113.98	1,070.49	475.35	25.13	0.00	225.00	261.17	986.66
9	1. Intra-D	02:38	42:56	100.41	1,170.91	475.35	0.00	0.00	225.00	252.13	952.48
10	1. Intra-D	03:14	46:11	108.31	1,279.22	475.35	1.31	0.00	225.00	252.60	954.25
11	1. Intra-D	03:02	49:13	110.14	1,389.35	475.35	8.98	0.00	225.00	255.36	964.69
12	1. Intra-D	02:03	51:16	76.36	1,465.71	475.35	0.00	0.00	225.00	252.13	952.48
13	1. Intra-D	02:52	54:08	108.18	1,573.89	475.35	0.75	0.00	225.00	252.40	953.50
<b>TOTAL:</b>											12,986.29

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988

Page 3

EQUIPMENT COSTS (at 8.00% discount rate) FOR SIMULATION COALRUN:

07/26/88 09:09am

EQUIP-MENT TYPE	NUMBER OF UNITS	PURCHASE PRICE (EACH)	HOURLY OWNERSHIP COST \$	TOTAL HOURS OPER.	TOTAL OWNERSHIP COST \$	REPAIR COST PER MILE \$	TOTAL MILES OPER.	TOTAL REPAIR COST \$	TOTAL COST DOLLARS
S402	5	1,117,800	15.12	8.35	631.43	0.58	183.33	531.65	1,163.08
hop100	115	38,500	0.45	8.35	429.72	0.07	183.33	1,475.79	1,905.52
caboos	1	41,300	0.48	8.35	4.01	0.28	183.33	51.33	55.34
<b>TOMINE</b>								2,058.78	3,123.93
S402	5	1,117,800	15.12	4.92	372.28	0.58	184.93	536.29	908.57
hop100	115	38,500	0.45	4.92	253.36	0.07	184.93	1,488.67	1,742.03
caboos	1	41,300	0.48	4.92	2.36	0.28	184.93	51.78	54.14
<b>FRMINE</b>								2,076.74	2,704.74
S402	7	1,117,800	15.12	7.72	816.62	0.58	294.13	1,194.16	2,010.78
hop100	115	38,500	0.45	7.72	396.97	0.07	294.13	2,367.72	2,764.69
caboos	1	41,300	0.48	7.72	3.70	0.28	294.13	82.36	86.06
<b>HELPER</b>								3,644.23	4,861.53
S402	5	1,117,800	15.12	0.47	35.28	0.58	13.87	40.23	75.51
hop100	115	38,500	0.45	0.47	26.01	0.07	13.87	111.67	135.68
caboos	1	41,300	0.48	0.47	0.22	0.28	13.87	3.88	4.11
<b>PPLANT</b>								155.79	215.29

**EXHIBIT 2 (CONTINUED)**  
**Example RECAP output for Unit Coal Train Service**

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988

Page 4

EQUIPMENT COSTS (at 8.00% discount rate) FOR SIMULATION COALRUN:

07/26/88 09:09am

EQUIP- MENT TYPE	NUMBER OF UNITS	PURCHASE PRICE (EACH)	HOURLY OWNSHP COST \$	TOTAL HOURS OPER.	TOTAL OWNSHP COST \$	REPAIR COST PER MILE \$	TOTAL MILES OPER.	TOTAL REPAIR COST \$	TOTAL COST DOLLARS
S402	5	1,117,800	15.12	24.00	1,814.15	0.58	308.83	895.62	2,709.77
hopmty	115	38,500	0.45	24.00	1,234.64	0.07	308.83	2,486.11	3,720.75
caboos	1	41,300	0.48	24.00	11.52	0.28	308.83	86.47	97.99
EMPTY	121				3,060.31			3,468.20	6,528.51

S402	2	1,117,800	15.12	24.00	725.66	0.58	294.68	341.82	1,067.49
LITEHP	2				725.66			341.82	1,067.49
TOTAL					6,755.93			11,745.57	18,501.50

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988  
 FUEL COSTS (at 0.57 per gallon) FOR SIMULATION COALRUN:

Page 5  
 07/26/88 09:09am

STOP NUMBER	SEGMENT LENGTH	ELAPSED MILES	RUNNING TIME	IDLE TIME	ELAPSED TIME	IDLE FUEL	RUNNING FUEL	FUEL COST \$	CUM. FUEL CONSUMED	CUM. FUEL EXPENSE
TOMINE-1	90.57	90.57	01:59	00:10	02:10	1	927	528.96	928	528.96
TOMINE-2	92.73	183.33	06:11	04:00	12:21	96	740	476.52	1,764	1,005.48
FRMINE-1	184.93	368.26	04:55	00:30	17:46	12	2,445	1,400.49	4,221	2,405.97
HELPER-1	184.45	552.70	04:34	00:05	22:26	1	2,224	1,268.25	6,446	3,674.22
HELPER-2	109.66	662.38	03:03	00:30	25:59	1	1,589	906.30	8,036	4,580.52
PPLANT-1	13.87	676.26	00:28	04:00	30:27	0	233	132.81	8,269	4,713.33
EMPTY-1	100.11	776.37	02:37	00:11	33:16	1	643	367.08	8,913	5,080.41
EMPTY-2	100.37	876.78	02:27	00:30	36:14	0	798	454.86	9,711	5,535.27
EMPTY-3	108.31	985.09	02:44	00:00	38:58	0	761	433.77	10,472	5,969.04
LITEHP-1	110.14	1,095.23	03:02	00:00	42:00	0	77	43.89	10,549	6,012.93
LITEHP-2	76.36	1,171.59	02:03	00:10	44:14	0	56	31.92	10,605	6,044.85
LITEHP-3	108.16	1,279.77	02:41	00:00	46:55	0	65	37.05	10,670	6,081.90
TOTAL			36:47	10:08						

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988

Page 6

TRACK MAINTENANCE COSTS FOR SIMULATION COALRUN:

07/26/88 09:09am

START STATION	END STATION	LENGTH MILES	TOTAL MGTM	CONSIST MGTM	TANGENT MILES	TANGENT COST \$	CURVED MILES	CURVED COST \$	TOTAL TRACK MAINT. COST
IAEGER	- WELCH	183.29	7,071.400	0.855	182.139	182.66	1.153	5.21	187.87
WELCH	- NORTHFORK	184.33	4,155.215	2.937	183.095	887.03	1.239	25.63	912.66
NORTHFORK	- PRINCETON	293.55	9,430.174	4.783	285.984	1,721.59	7.566	108.17	1,829.76
PRINCETON	- BLUEFIELD	13.90	479.658	0.221	11.235	68.86	2.662	34.09	102.95
BLUEFIELD	- IAEGER	308.84	11,009.498	1.441	298.619	254.12	10.216	39.04	293.17
PRINCETON	- NORTHFORK	294.70	10,520.625	0.106	287.150	18.79	7.555	2.32	21.11
TOTAL		1,278.61	42,666.571	10.342	1,248.223	3,133.06	30.391	214.46	3,347.52

**EXHIBIT 2 (CONTINUED)**  
**Example RECAP output for Unit Coal Train Service**

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988

Page 7  
07/26/88 09:09a

CATEGORY	TOTAL COST	COST PER 1000 GTM	COST PER 1000 MTM	COST PER MILLION CFM
Locomotive Ownership Costs	631.43	0.738	Empty	7.471
Locomotive Running Costs	531.65	0.622	Empty	6.291
Car Ownership Costs	433.73	0.507	Empty	5.132
Car Running Costs	1,527.13	1.786	Empty	18.069
Fuel Costs	1,005.48	1.176	Empty	11.897
Track Maintenance Costs	187.87	0.220	Empty	2.223
<b>TOMINE SUBTOTALS:</b>	<b>4,317.29</b>	<b>5.049</b>	<b>Empty</b>	<b>51.083</b>
Locomotive Ownership Costs	372.28	0.126	0.175	4.367
Locomotive Running Costs	536.29	0.182	0.252	6.291
Car Ownership Costs	255.72	0.087	0.120	3.000
Car Running Costs	1,540.45	0.523	0.724	18.069
Fuel Costs	1,400.49	0.475	0.659	16.428
Track Maintenance Costs	912.66	0.310	0.429	10.705
<b>FRMINE SUBTOTALS:</b>	<b>5,017.90</b>	<b>1.703</b>	<b>2.360</b>	<b>58.860</b>
Locomotive Ownership Costs	816.62	0.170	0.241	6.023
Locomotive Running Costs	1,194.16	0.249	0.353	8.807
Car Ownership Costs	400.67	0.084	0.118	2.955
Car Running Costs	2,450.08	0.511	0.724	18.069
Fuel Costs	2,174.55	0.454	0.643	16.037
Track Maintenance Costs	1,829.76	0.382	0.541	13.495
<b>HELPER SUBTOTALS:</b>	<b>8,865.84</b>	<b>1.850</b>	<b>2.621</b>	<b>65.386</b>
Locomotive Ownership Costs	35.28	0.160	0.221	5.516
Locomotive Running Costs	40.23	0.182	0.252	6.291
Car Ownership Costs	24.23	0.110	0.152	3.789
Car Running Costs	115.56	0.523	0.724	18.069
Fuel Costs	132.81	0.601	0.832	20.767
Track Maintenance Costs	102.95	0.466	0.645	16.097
<b>PPLANT SUBTOTALS:</b>	<b>451.05</b>	<b>2.040</b>	<b>2.827</b>	<b>70.529</b>
Locomotive Ownership Costs	1,814.15	1.259	Empty	12.742
Locomotive Running Costs	895.62	0.622	Empty	6.291
Car Ownership Costs	1,246.15	0.865	Empty	8.753
Car Running Costs	2,572.58	1.786	Empty	18.069
Fuel Costs	1,255.71	0.872	Empty	8.820
Track Maintenance Costs	293.17	0.204	Empty	2.059
<b>EMPTY SUBTOTALS:</b>	<b>8,077.38</b>	<b>5.607</b>	<b>Empty</b>	<b>56.734</b>
Locomotive Ownership Costs	725.66	6.879	Empty	0.000
Locomotive Running Costs	341.82	3.240	Empty	0.000
Car Ownership Costs	0.00	0.000	Empty	0.000
Car Running Costs	0.00	0.000	Empty	0.000
Fuel Costs	112.86	1.070	Empty	0.000
Track Maintenance Costs	21.11	0.200	Empty	0.000
<b>LITEHP SUBTOTALS:</b>	<b>1,201.46</b>	<b>11.389</b>	<b>Empty</b>	<b>0.000</b>

**EXJOBOT 2 (CONTINUED)**  
**Example RECAP output for Unit Coal Train Service**

RAIL ENERGY COST ANALYSIS PACKAGE (RECAP) Ver. 2.04 Jun 6, 1988

Page 8

RECAP COST SUMMARY FOR SIMULATION COALRUN:

07/26/88 09:09a

CATEGORY	TOTAL COST	COST PER 1000 GTM	COST PER 1000 NTM	COST PER MILLION CFM
Locomotive Ownership Costs	4,395.42	0.424	0.775	9.679
Locomotive Running Costs	3,539.77	0.342	0.624	7.795
Car Ownership Costs	2,360.51	0.228	0.416	5.198
Car Running Costs	8,205.80	0.792	1.448	18.069
Fuel Costs	6,081.90	0.587	1.073	13.393
Track Maintenance Costs	3,347.52	0.323	0.591	7.371
Crew Costs	12,986.29	1.253	2.291	28.596
<b>COALRUN TOTALS:</b>	<b>40,917.21</b>	<b>3.949</b>	<b>7.218</b>	<b>90.101</b>

ton mile (NTM) which is most relevant for weight limited commodities such as coal or grain, and by million cubic foot miles (CFM) which provides a basis for cost comparison on the movement of low density, volume limited freight such as much of the general merchandise traffic.

## 1.2 THE TRAIN ENERGY MODEL

Since TEM is a core component of RECAP and must be executed prior to running RECAP's economic model, a general discussion of the development and capabilities of TEM is appropriate in this RECAP manual. This discussion is not intended to substitute for the far more extensive material on TEM available in the Train Energy Model Users Manual (1), which should be read before attempting to conduct TEM/RECAP analyses.

TEM was developed by the AAR's Energy Program in order to facilitate the study of methods of reducing train resistance. The main TEM development objective was to overcome the known limitations of physical simulators of train motion such as the Train Performance Simulator (TPS) and the Train Performance Calculator (TPC) by making the most complete and realistic representation of the physical system possible, and by incorporating the latest results of the research being done in train resistance.

The resultant TEM more accurately (and for the user more easily) models such factors affecting train performance as: locomotive tractive effort and fuel consumption rates, dynamic and air braking systems, track geometry, and the physical characteristic of vehicles such and air resistance, curving characteristics of trucks, bearing resistance, vehicle weight, and rail lubrication.

### 1.2.1 How TEM Really Works

In 1687 Sir Issac Newton published a manuscript which explained the principles of Newtonian mechanics. Newton's Second Law states that a body will accelerate at a rate that is proportional to the force acting upon it and inversely to the mass of the body being acted upon. For example, the more powerful the locomotives the greater the acceleration of the train. However, the greater the mass of the train, the slower the acceleration given the same locomotives.

Because Newtonian mechanics accurately predicts the physical behavior of trains, the central function of TEM is to solve an equation of motion based upon Newton's second law. There exists a complex and non-linear relationship among propulsive forces, retarding forces and train handling on one hand, and running time and fuel consumption on the other. Because of the nature of these forces and relationships a non-linear differential equation of motion must be solved in order to consistently obtain an accurate estimate of fuel consumption and running time.

TEM solves the equation for acceleration and then integrates the equation to find the velocity of the train and the position at discrete points in time. This is done with a new numerical method that is "special" because of two properties. First, the approach is stable for very long simulations. In other words small inaccuracies in the solution to the equation of motion are not cumulative over time. Second, the approach permits the time increment for solving the equation to vary, permitting TEM to be relatively computation efficient.

### 1.2.2 The TEM Advantage

There are several advantages that TEM has over earlier train performance models. First is the mathematical modeling itself. As indicated earlier, moving a train involves a complex interaction of non-linear forces. These forces must be computed with a non-linear differential equation which cannot be solved directly with a mathematical formula but must be approximated with a numerical method. The train performance models developed prior to the AAR's research program in train resistance did not model the non-linear interaction of forces involved in train movement. For example, the Train Performance Calculator (TPC) assumes these forces (curve resistance, tractive effort, dynamic braking effort, etc.) and the resulting physical behavior of the train (acceleration and deceleration) can be described as linear relationships.

In addition to a basic simplification of the equations of motion, earlier models made a large number of other assumptions to simplify the modeling, that were incorrect. They performed very rudimentary modeling or no modeling of train handling, train propulsion and braking systems, the effects of wind resistance, lubrication, etc.. These models also either viewed a train as a unit mass of zero length (a particle) or a series of fixed-length blocks. TEM models the forces acting upon each individual vehicle in the consist based upon its actual position on the track.

The advantages of using TEM can be summed up as:

- (1) The approach of TEM to modeling the physics of a train is sound. Tests performed by the AAR and by individual railroads have validated the model.

- (2) TEM can provide accurate results for new, untested technologies. Since the physics and mathematics are correct, TEM does not require calibration based upon actual tests of a particular technology to get accurate results.
- (3) All of the known factors that have a measurable impact upon fuel consumption or running time are included in TEM. Given accurate descriptions of the physical characteristics of the equipment in the train, the route geometry, and the operating plan of the train, the results will be accurate.

### 1.2.3 Running TEM-The User's Point of View

Prior to simulating the physical behavior of a train running over a specific route the user must describe three fundamental things to TEM.

- (1) The physical characteristics of the train consist.
- (2) The geometry of the route over which the train will run.
- (3) The operating strategy for the particular train.

The physical characteristics of the train are described in a consist file that contains a line for each piece of equipment in the train. Each non-powered vehicle has a parameter file associated with it that describes its physical characteristics (such as weight, drag coefficient, number of axles, etc.). Each locomotive (powered vehicle) has three associated parameter files. One that includes its physical characteristics plus the fuel consumption rates in each throttle position and two files

which quantify the tractive effort and dynamic braking effort at various speeds and throttle notches. A modest library of car and locomotive files has been developed at this time, and additional car and locomotive descriptive files will be developed over time. A good library of files that describe various pieces of equipment to TEM reduces the required user effort.

The route geometry file describes the physical characteristics of the route including position, degree of curvature, percent grade, speed limit, location of trackside lubricators, weight and type of rail, etc. on a homogeneous segment-by-segment basis. The development of a good route file is the hardest physical task facing a new TEM/RECAP user, especially if one does not have anything but paper track charts as a starting point.

The final obligation of the user is to provide an operating strategy. TEM must be instructed where (in the route geometry file) to start the train, the location of interim destinations (crew change points, etc.) and the final destination. You also choose whether to run a train at a specified maximum speed below the track speed limit or to follow the speed limit profile. Additionally, you can specify if there are helper locomotives and whether the helpers run under independent or synchronous control.

Once the simulation starts, information about the status of the train is displayed on the computer screen at user specified intervals (e.g. every 30 seconds of simulated "train time"). The video display output indicates how the train is being "moved" over the route. The displayed information includes head-end and helper throttle (or dynamic brake) notch, simulated time and distance since start, current speed and acceleration,

brake pipe pressure, and draw bar force. In addition to the output to the screen, a detailed file is written to disk that provides a running record of the behavior of the train, including cumulative fuel consumption. This file provides RECAP with information it needs to analyze the costs of the simulated operations.

TEM has an option which permits the user to manually "drive" a train by issuing commands that correspond to train handling activities (changing throttle setting, making an air brake reduction, etc.). However, for the vast majority of applications one should use TEM's Automatic Train-Handling Algorithm (ATA) to run the train. ATA simulates the actions of an engineer and automatically drives the train through the simulation. This provides for consistent train handling and removes an extraneous source of variation from the comparisons of train technology or service plans.

### 1.3 TRACK MAINTENANCE COSTS

One of the significant incremental costs of providing rail transportation service is the cost of maintaining track. This cost is certainly sensitive to the characteristics of the route, the equipment, and the service plan. Unfortunately, accounting based cost allocation models have not provided a sound basis for estimating the effects of track geometry and construction standards, equipment axle loads and curving characteristics, and operating speeds.

RECAP uses the outputs from an engineering based, life cycle track maintenance cost model, TMCOST, to estimate route and service specific track maintenance costs. Since TMCOST was designed as a research tool and not for general distribution, TMCOST is very complex to use and computer running times are quite long. Rather than building TMCOST into the RECAP program,

TMCCOST was used to generate incremental track deterioration cost files to be used as input to RECAP's economic model.

There are five types of consists for which TMCCOST output files have been generated and are distributed with the RECAP software. These files contain the results of TMCCOST simulations for the following consists:

- o Loaded bulk commodity unit trains with 100 ton cars (32.5 ton axle loads).
- o Loaded bulk commodity unit trains with 125 ton cars (39 ton axle loads).
- o Empty bulk commodity unit trains.
- o Mixed manifest trains.
- o Intermodal trains.

Each of these consist types has been described to TMCCOST with appropriate values for the physical characteristics of equipment that have a significant impact upon track deterioration; wheel loading, center of gravity height, unsprung mass, and wheel diameter. In addition, the power to weight ratios, operating speeds, and train lengths have been set at typical values for these types of services. TMCCOST was then run for a matrix of track geometry and track structures to produce track maintenance cost files for RECAP.

Prior to running RECAP the user selects the TMCCOST output file that most closely resembles the consist that was simulated by TEM. Since RECAP can generate costs for multiple consists, a TMCCOST output file must be selected for each TEM simulation/consist combination.

Each of the five TMCOST files contain the cost per million gross tons for maintaining 1 mile of track under an increment of the specified traffic. These costs are sensitive to:

- o Consist Weight,
- o Rail type (8 types from light jointed to heavy, premium, CWR),
- o degree of curvature,
- o percent grade,
- o Density of Traffic (MGT per Year)
- o lubricated or non-lubricated rail,
- o Consist type.

In general, these incremental track maintenance costs estimated by TMCOST and distributed with RECAP are an adequate indicator of the relative costs of track deterioration that can be expected from most alternative operating strategies. These costs are not adequate as an absolute cost measurement tool or for detailed study of track maintenance strategies. The labor and material costs and track work productivity of maintenance gangs are based upon industry averages and may or may not be close to those experienced on a particular railroad. In addition, they only include the costs for maintaining the track from the rail down to the subgrade and do not include such factors as bridge maintenance, brush cutting, turnout and special trackwork maintenance, etc.

RECAP allows the user to build the track maintenance cost input files in any manner. If an individual road has an internal costing procedure which it wishes to use, the track

maintenance file can be constructed using that procedure. If the traffic is quite different from the five consist types distributed with RECAP, custom track maintenance files can be constructed with TMCOST. However, for the great majority of evaluations involving existing equipment, the TMCOST generated files included with RECAP should be adequate.

#### 1.4 COMPUTER REQUIREMENTS AND INSTALLATION PROCEDURES

Recap requires an IBM or compatible computer with a hard disk drive and a math co-processor. Although RECAP itself is not computational intensive, the required Train Energy Model is very computational intensive. The preferred hardware is an Intel 80386-based computer with an 80387 math chip, although TEM/RECAP will run at a slower speed on any IBM compatible.

The RECAP program is available to all AAR full member roads. To receive RECAP simply request a copy from the Engineering Economics Division of the Washington Systems Center of Research & Test. After signing and returning the standard member road license agreement, you will receive a RECAP distribution diskette. All you need do to install the RECAP software is to copy all the files on the RECAP distribution diskette to any sub-directory on your PC hard disk. Make sure that in your CONFIG.SYS file in your root directory that the "files=" parameter specifies at least 20 files, i.e., "files=20".

The files that are on the distribution diskette are: ECON2.COM (the Economic Model), TKEDIT.EXE (the track file editor), TKEDIT.DTA (a required file for the track file editor), and TKSPLICE.EXE (the track file management program). Each of these programs are documented in the following sections of this manual.

The Train Energy Model Manual and Program are distributed out of the Chicago Technical Center of AAR Research & Test. Contact the Director, Mechanical Research Programs for the TEM software and documentation.

## 1.5 MANUAL ORGANIZATION

The remainder of this RECAP Operations Manual is a detailed nuts and bolts description of how to run the RECAP economic model as well as two useful utility programs. The utility programs included in the RECAP package are: TKEDIT, a route geometry file editor, and TKSPLICE, a route geometry file management system.

The manual is divided into four major sections. A description of each section follows:

### 1.0 INTRODUCTION

This section provides an overview of the RECAP system and its potential applications, including a description of computer hardware requirements and instructions for installing the software on your computer.

### 2.0 THE ECONOMIC MODEL

Sub-sections 2.1 and 2.2 provide detailed information on the input file requirements for RECAP's economic model (named ECON2) including detailed file format and field descriptions. Most of the files required are taken directly and without modification from TEM and the contents of these files are documented in the TEM Users Manual. However, one file which provides both crew pay and

crew-to-train assignments is unique to RECAP. Two types of files, the route geometry file and the vehicle parameter files, must be modified. (Note: the modifications are ignored by TEM, thus the files are usable by both RECAP and TEM.)

Sub-section 2.3 provides detailed information on how to run the economic model. This includes how to combine several TEM simulations into a single analysis of a set of trains. Also included in these sub-sections is information about how to tell the economic model which input files are to be used. Finally, information is provided about how to specify economic data such as the price of fuel and the discount rate and time period for computing equipment ownership costs.

Sub-section 2.4 explains the 6 output reports that are generated by the RECAP economic model. It includes relevant information about the equations and processing logic that the program uses.

Sub-section 2.5 explains how certain aspects of the train operations strategy can be altered without re-running the TEM simulation. For example, if you run a TEM simulation and forget to stop the consist for a crew change, you can tell RECAP to create an artificial stop. This sub-section also provides detailed information about how the TMCOST files were created, how to interpret the results, and certain pitfalls to avoid.

### 3.0 TKSPLICE (Track File Management System)

TKSPLICE permits you to take one or more existing track files and reverse them, splice them together, or extract sub-sets of the files. This section provides detailed information on how to use this file management program.

### 4.0 TKEDIT (Track File Editor and Verifier)

TKEDIT is a full screen editor that permits the user to edit an existing route geometry file or to create one from scratch (a task to be avoided if at all possible). TKEDIT also includes a routine that checks the file for format and certain computational errors. It is TKEDIT that you may wish to use to modify the route geometry file for use by RECAP's economic model. This final section of the RECAP manual provides the information necessary to use this utility program.

## 2.0 THE ECONOMIC MODEL

The economic model "ECON2" is the core of RECAP. It is the computer program which actually computes the cost of a particular operating and/or technological scenario.

### 2.1 INPUT FILES

RECAP tries to minimize the labor intensive process of creating input files beyond those needed to run the Train Energy Model (TEM). Most of the files you will need will have been created as inputs for or generated as outputs from the run(s) of TEM which must be run prior to running the RECAP economic model.

The input files that are required for running recap are:

#### (1) Generated for or from the Train Energy Model

Initialization file - 'xxxxxx.INI'  
Stop File - 'xxxxxx.STP'  
Route Geometry File - 'xxxxxx.TRK'  
Vehicle Parameter Files - 'xxxxxx.PAR'  
Train Consist File - 'xxxxxx.CON'  
TEM detailed output file - 'xxxxxx.OUT'

#### (2) Additional Files

Track Maintenance Cost File - 'xxxxxx.TMC'  
Crew Data File - 'xxxxxx.CRW'  
RECAP batch file - 'xxxxxx.RCP'  
RECAP output file - 'xxxxxx.CSM' (not an input file)

Five track maintenance cost files are supplied as a part of the RECAP package and are included on one of the diskettes included in the package. The crew data file must be created by the user.

## 2.2 INPUT FILE DESCRIPTIONS

Of the seven input files required by RECAP only two, the crew data file and the RECAP batch file, need to be created from scratch and two files, the vehicle parameter and the route geometry file, need to be modified. The following is a description of the function of each file and for those that need to be created or modified, detailed information on how to do so.

### 2.2.1 The Initialization File (.INI)

The TEM initialization file always has the filename extension '.INI', for example, 'ATOB.INI'. In RECAP as in TEM this file identifies the name of a particular discrete simulation. The information that RECAP uses from the .INI file is the following;

- (1) The name of the consist (.CON) file used by the simulation.
- (2) The name of the route geometry (.TRK) file used by the simulation.
- (3) The position in the track file where the simulation is to start.
- (4) An indicator specifying whether vehicle mounted lubricators are functioning.
- (5) An indicator specifying whether track side lubricators are functioning.
- (6) The effective lubrication distance for track side lubricators.

This file is required by TEM and no changes or additional information is required from the user to run RECAP.

### 2.2.2 The Stop File (.STP)

The TEM Stop file tells RECAP where all interim stops and the final destination for the simulated train operation is located in the route geometry file. The stops may be for crew changes, service, train meets, or the final destination for that particular simulation. NOTE: THE .STP FILE ALWAYS HAS THE SAME FILE PREFIX AS DOES THE .INI FILE.

From the point of view of TEM the stops may occur for any reason other than to change the physical characteristics of the train consist which requires a separate TEM simulation. However, RECAP does permit the user to take multiple TEM simulations and therefore multiple consist configurations (ex., adding or deleting helpers, car setoffs, change in status from loaded to empty, etc.) and generate one set of costs for one logical set of trains. The details of executing this and other strategies are explained in section 2.3.1 titled "The RECAP Multiple Simulation File".

RECAP uses all of the information in the .STP file. This includes:

- (1) The position (miles and feet) of each train stop in the simulation.
- (2) The duration (in minutes) of the stop.
- (3) An indicator whether the locomotives are left idling or are shut down.

This file is required by TEM and no changes or additional information is required from the user to run RECAP.

### 2.2.3 The Route Geometry File (.TRK)

RECAP requires two modifications to the TEM route geometry file. Both of the entries must reside in specific columns in the file. This information can be added either by using the RECAP utility program TKEDIT (the track file editor) or by using any standard resident editor on your PC. The information required is as follows;

- (1) RAIL TYPE (COLUMN 64, INTEGER, LENGTH 1, valid values are 1 through 8 or 's')

RAIL TYPE is a classification indicator for the type of rail existing in each segment. It is one of the variables used to calculate the track maintenance costs for each segment. There must be a rail type indicator for each record in the track file. The valid rail type indicators and descriptions are:

RAIL TYPE	DESCRIPTION
1	Heavy, standard carbon, continuous welded rail (CWR)
2	Heavy, premium, CWR
3	Heavy, standard carbon, jointed
4	Heavy, premium, jointed
5	Medium, standard carbon, CWR
6	Medium, standard carbon, jointed
7	Light (less than 100 lb/yd), standard carbon, CWR
8	Light, standard carbon, jointed

There is one exception to the rail type codes listed above. The Economic Model will calculate track maintenance costs between any two track file records (route segments).

Place an 's' in column 62 instead of a rail type code at any two or more stations between which you wish to have the track maintenance costs calculated. The default (if you place no 's' indicators in the file) is that the track maintenance costs will be calculated and summed for the entire simulation.

Note you must not have an 's' in the first or last record in the track file. The economic model assumes that the rail type code for any segment with an 's' indicator is the same rail type as for the previous segment.

This field is column specific and must occur in column 64 of the route geometry file.

- (2) TRAFFIC DENSITY (COLUMN 66, LENGTH 4, REAL, VALID VALUES ' 0.1' THROUGH '999.9')

TRAFFIC DENSITY is the number of gross tons generated each year over each segment of track in the route geometry file. Although any values that will fit into the five position field will be accepted, the track maintenance costs will not change for values over 100 MGT per year; i.e., a value higher than 100 MGT will be assumed to be equal to 100 MGT. The value in the field must be a real number (i.e. include a decimal point and have at least 1 number prior to and 1 number following the decimal point).

#### 2.2.4 Vehicle Parameter Files (.PAR).

The TEM "PAR" files describe each different type of vehicle in the consist. There are two types of 'PAR' files, files for powered vehicles (locomotives) and files for non-powered vehicles (freight cars).

To create RECAP PAR files from TEM PAR files you must add 1 line of information to each vehicle PAR file in the consist. That line must be the last line in the file and will need to be inserted after the last line of the standard TEM PAR file. The information that you provide does not need to be column specific but must be in a specific order and have the correct format.

(1) CAR parameter file (ex, HOPPER.PAR)

These values are inserted on LINE 2 in the file and must be in the following order and format. Each value must be separated by at least 1 BLANK.

CUBIC CAPACITY (INTEGER, ex. 4000, must be >0)

DAILY CAR COST in dollars and cents (REAL, ex. 25.00, default is 0.00 and if the default is entered car cost is computed from the purchase price, economic life, and discount rate)

RUNNING REPAIR COST per mile in dollars and cents, (REAL, ex. 0.07)

PURCHASE PRICE in dollars (INTEGER, ex., 38500)

ECONOMIC LIFE in years (INTEGER, ex. 18, default is 20)

The DAILY CAR COST may be entered as 0.00. If this value is 0.00 then an equivalent hourly cost will be computed from the purchase price of the vehicle, the discount rate (specified on the command line, see the section titled "RUNNING THE ECONOMIC MODEL"), and the ECONOMIC LIFE of the vehicle.

The value for PURCHASE PRICE can be 0 only if you enter a non-zero value for DAILY CAR COST.

The value for ECONOMIC LIFE which is the last value in the file can omitted. If this value is not present the life of the vehicle will be assumed to be 20 years.

Therefore line 2 of a file called HOPPER.PAR may look like this if you want an equivalent hourly cost calculated from the purchase price:

```
4000 0.00 0.07 38500 18
```

If you wish to provide the daily car cost (for example a lease cost) line 2 of the file may look like:

```
4000 25.00 0.07 0 18
```

(1) LOCOMOTIVE parameter file (ex, S402.PAR)

These values are inserted on LINE 4 in the file and must be in the following order and format. Each value must be separated by at least 1 BLANK.

PURCHASE PRICE in dollars (INTEGER, ex., 1175000).

RUNNING REPAIR COST per mile (REAL, ex., 1.30).

ECONOMIC LIFE in years (INTEGER, ex., 18).

UNIT AVAILABILITY in percent (REAL, ex 0.92)

The hourly cost of a locomotive is always computed from the purchase price, discount rate, and availability of the unit. Therefore, the 4th line of a locomotive PAR file might look like:

```
1175000 1.30 18 0.92
```

### 2.2.5 The Train Consist File (.CON)

The TEM consist file is a list of the powered and non-powered vehicles in the order in which they are positioned in the consist. Each different type of vehicle in the consist file has to have a corresponding vehicle .PAR file. There are no modifications required to be made to the consist file.

If you wish to calculate the costs for an empty train you may want to check the consist file to make sure that the entry for vehicle lading (the last entry in each line) is 0.0 for each vehicle. Sometimes people will put lading on locomotives so that the weight of the fuel will be taken into account during the TEM simulation. ECON2 does not differentiate between lading on locomotives and lading on cars for computing the number of net tons (an indicator of the train's capacity to generate revenue) in the train. There is at least one technological concept for which the powered vehicles are used to carry lading.

### 2.2.6 TEM Output File (.OUT)

The TEM output file is a detailed record of the performance of a consist from the beginning to the end of the simulation. From the RECAP perspective it is a record of fuel consumption through time. In the TEM .INI file you can specify the time interval (simulated train time) for which TEM will write the detailed simulation information to disk (for example, every 30 seconds of simulated time). Although it does not matter to RECAP what that time interval is, the more frequent the interval the more accurately will the simulation's results be interpreted by RECAP. An output interval of 30 to 60 seconds is recommended to balance between accuracy and the volume of the output file.

### 2.2.7 Crew Data File (.CRW)

The Crew Data File provides information to the economic model about the type of crew pay agreement, how much each crew is paid, and upon which trains the crews are on duty. The Economic Model permits you to model very complex crew to train assignments as well as different crew compensation methods. There is only 1 crew file for each execution of RECAP regardless of the number of TEM simulations you are costing during the RECAP execution. The crew file naming convention requires that the file have a .CRW extension. The ways you can specify to RECAP which crew file to use for a particular run is discussed in section 2.3 titled "RUNNING RECAP".

The crew file is the only part of the economic model which may be difficult to understand. The great deal of flexibility that has been build into RECAP makes it understandable that some confusion may result. If you intend to model crews which are on duty across multiple TEM simulations you are advised to contact the AAR for some initial assistance.

The Crew Data File is not column specific; however, all data elements must be the correct type (real or integer) and in correct position in relation to the other fields. At least 1 space must exist between fields. The fields in the crew file are as follows;

- (1) CREW TYPE (Integer, valid values are '0', '1', or '2'). The CREW TYPE indicates the crew compensation method in effect for this particular crew.

A crew type of '0' indicates that the crew is paid on a time basis only (no mileage based pay in effect). A crew type of '0' is only useful if you wish to model non-traditional crew compensation methods.

A crew type of '1' is a standard intra-divisional crew agreement. Most railroad crews are paid on this basis. The crew is on duty only while the train is within that crew's home seniority district.

A crew type of '2' is an inter-divisional crew agreement. This type of agreement is in effect if the railroad and the union have agreed that a particular train may run a crew through more than one seniority district.

The basis for computing overmile pay is the only difference between a type '1' and a type '2' crew. For the intra-divisional crew agreement (type 1) the overmile pay rate is a fixed amount for each job class. For the inter-divisional agreement (type 2) the overmile pay rate is computed as  $\text{BASE DAY WAGES} / \text{BASE DAY MILES}$ . This formula usually results in a higher overmile rate for the inter-divisional crew agreement. However, the crew does not get an additional base day's pay upon crossing a crew district boundary.

The following list of entries for the crew file apply if the crew type is either a '1' or a '2'.

- (2) **BASE DAY WAGES.** The total base day's wages (in dollars and cents) for the entire crew. Not including arbitraries, benefits, etc.. (REAL, ex. 400.00)
- (3) **OVERMILE RATE.** The sum of the overmile pay rates for all crew members. (REAL, ex. 3.85).

- (4) FRINGE BENEFIT RATE (percent). The percentage of BASE DAY WAGES that the railroad expects to pay for fringe benefits (railroad retirement, health insurance, etc.) (REAL, ex. 0.35). The default value for fringe benefits will be computed by multiplying the fringe benefit rate by the base day wages. If you prefer you can put a minus sign (for example, -0.35) prior to the value and the fringe benefit total will be computed based upon total crew wages (base wages, arbitraries, overmile pay, and if applicable, overtime pay).
- (5) ARBITRARIES The total amount of arbitrary payment to the crew for this run in dollars and cents (REAL, ex. 200.00). At one time it was under consideration to develop a rule based arbitrary computation method. This was deemed infeasible due to the differences among various railroads in regard to this issue, i.e., arbitraries are really arbitrary.
- (6) BASE DAY MILES. The number of miles for which the crew receives 1 base day's pay (REAL or INTEGER, ex 108 or 108.00).
- (7) CREW START INI FILE NAME. The name of the TEM .INI file in which this crew goes on duty. (ALPHA, ex HOP100, Note: the extension .INI is assumed)
- (8) CREW START TRACK FILE POSITION. The position in the track file (.TRK) where the crew goes on duty. (STRING, ex. 145789 or 145789.5 in feet or 200:2345 in miles and feet)

You may specify this value as either the number of feet from the start of the track file or if you prefer

you can specify the value in miles and feet by using a colon to separate miles from feet. Because the TEM stop file (.STP) requires that all stops be specified in miles and feet and because RECAP requires that there be a stop in the .STP file for all crew changes (RECAP does not allow crews to get on or off moving trains) you may find it more convenient to use the miles and feet option.

- (9) CREW END INI FILE NAME. The name of the TEM .INI file in which this crew goes off duty. The same conventions apply from item 7 above.
- (10) CREW END TRACK FILE POSITION. The position in the TEM track file (.TRK) in which the crew goes off duty. The same conventions apply as in item 8 above.

The following represents the crew file data elements if the CREW TYPE INDICATOR is '0' (hourly crew pay).

- (2) BASE HOURLY RATE. The total hourly pay rate for the entire crew. (REAL, ex 85.65).
- (3) MINIMUM PAID HOURS. The guaranteed number of hours for which the crew will be paid even if they work fewer than that number of hours. For example if the crew were on duty only 2 hours but they get paid for a 4 hour minimum day then the entry would be '4'. (REAL, ex 4.0).
- (4) STRAIGHT TIME HOURS. The number of hours for the which the crew is paid at the BASE HOURLY RATE. Any time worked in excess of this time will be paid at the OVERTIME RATE (REAL, ex. 8.0).

- (5) OVERTIME RATE. The overtime multiplier for all work in excess of the STRAIGHT TIME HOURS (REAL, ex 1.5).
- (6) FRINGE BENEFIT RATE. The percentage of total crew wages that will be paid in fringe benefits, ex railroad retirement, health insurance, sick pay, etc. (REAL, 0.27) Note: the FRINGE BENEFIT RATE is computed on total crew wages only.
- (7) OTHER CREW COSTS. A flat amount of any additional crew costs you may want to include. This could be arbitraries, taxi costs, wages incurred while not on duty, etc. (REAL, ex. 100.00).
- (8) CREW START INI FILE NAME. The name of the TEM .INI file in which this crew goes on duty. (ALPHA, ex HOP100, Note: the extension .INI is assumed)
- (9) CREW START TRACK FILE POSITION. The position in the track file (.TRK) where the crew goes on duty. (STRING, ex. 145789 or 145789.5 in feet or 200:2345 in miles and feet)

You may specify this value as either the number of feet from the start of the track file or if you prefer you can specify the value in miles and feet by using a colon to separate miles from feet. Because the TEM stop file (.STP) requires that all stops be specified in miles and feet and because RECAP requires that there be a stop in the .STP file for all crew changes (RECAP does not allow crews to get on or off moving trains) you may find it more convenient to use the miles and feet option.

(10) CREW END INI FILE NAME. The name of the TEM .INI file in which this crew goes off duty. The same conventions apply from item 8 above.

(11) CREW END TRACK FILE POSITION. The position in the TEM track file (.TRK) in which the crew goes off duty. The same conventions apply as in item 9 above.

The following example crew file is for a run of the economic model for which the costs associated with a loaded and empty movement of a bulk commodity train are calculated. This particular example entails 2 TEM simulations having the .INI file names "HOP100" and "MTY100".

For the loaded train (HOP100) there are 4 crews. The first three are paid on the basis of an intra-divisional agreement (CREW TYPE INDICATOR = '1'). The fourth crew is is paid based upon an inter-divisional crew agreement (CREW TYPE INDICATOR = '2').

For the empty train (MTY100) there are 5 crews. All five are paid on an hourly basis (CREW TYPE INDICATOR = '0').

The example crew file would look like this:

```
1 400.00 3.85 0.35 200.00 100.00 hop100 0 hop100 100:550
1 400.00 3.85 0.35 200.00 100.00 hop100 100:550 hop100 200:3153
1 400.00 3.85 0.35 200.00 100.00 hop100 200:3153 hop100 308:4413
2 400.00 3.85 0.35 200.00 100.00 hop100 308:4413 hop100 492:667
0 85.65 4.0 8.0 1.5 0.27 0.00 mty100 0:0 mty100 105:5008
0 85.65 4.0 8.0 1.5 0.27 0.00 mty100 105:5008 mty100 183:1544
0 85.65 4.0 8.0 1.5 0.27 0.00 mty100 183:1544 mty100 291:2804
0 85.65 4.0 8.0 1.5 0.27 0.00 mty100 291:2804 mty100 392:127
0 85.65 4.0 8.0 1.5 0.27 0.00 mty100 392:127 mty100 492:667
```

### 2.2.8 Crew File Rules and Strategies

(1) There must be a stop in the TEM .STP file any time you have a crew change. This is the case for any type of crew. There will probably be cases where you will have a stop in the TEM .STP file for reasons other than a crew change (ex. setting aside or picking up cars, to pull into a siding for another train to pass, or any of a number of reasons). Therefore every crew change must have a corresponding entry in the stop file but every entry in the stop file may not entail a crew change.

(2) The train must have a crew on board at all times. For example you cannot have a crew getting off the train at position 100:550 and the next crew getting on at position 150:0. Therefore the start position of at each crew (except the first which starts at the track file position in the .INI file where the train starts running) must be the final destination of at least 1 other crew.

(3) A crew may be on duty during 1 or more TEM simulations (a single .INI file is defined as 1 simulation). If the crew is on duty in more than one simulation those simulations may occur at any time and do not need to be physically contiguous.

NOTE: READ THE SECTION TITLED "RUNNING RECAP" for a description of the concepts of contiguous and noncontiguous simulations and information on how to generate the costs for multiple trains (separate simulations viewed as a logical set of trains).

For example, in the sample crew file note that there are nine crews (each line in the file represents 1 crew) on duty

during the two simulations (two TEM .INI files corresponding to two trains). The loaded train (called HOP100) has four crews and the empty reverse route (called MTY100) has five crews. Assume that the last crew for the loaded move also unloads the train and then functions as the first crew of the empty move. In this case, one crew would operate in two simulations (two logical trains).

You must tell the economic model which simulations the crew worked in and where (in the TEM. TRK file) the crew began and terminated their duty.

Since the crew started work in simulation HOP100 and finished in simulation MTY100 the entries for the data elements CREW START INI FILE NAME, CREW START TRACK FILE POSITION, CREW END INI FILE NAME, AND CREW END TRACK FILE POSITION would be:

```
hop100 308:4413 mty100 105:5008
```

Note that in this example the crew goes on duty at 308 miles and 4413 feet into simulation HOP100 and continues through until 105 miles and 5008 feet into simulation MTY100.

In this example the two simulations that the crew operated in are physically contiguous, i.e. the simulation MTY100 occurs immediately subsequent to the simulation HOP100.

You may also have a crew which in on duty in multiple but non-contiguous simulations. For example, you may have a crew working in helper service for part of a simulation and then running the helpers light back to their origin. Assume the helper crew goes on duty at mile 200:3153 of simulation HOP100 and takes the helpers 25 miles and finally runs the helpers light back down the hill in a simulation called LITEHP.

Furthermore, assume that you want the the logical progression of trains to be such that the costs of the LITEHP move are computed after the costs of the HOP100 and MTY100 moves are computed, i.e. the LITEHP move is not contiguous to the HOP100 move even though the crew goes on duty in HOP100 and goes off duty on LITEHP.

In this example the appropriate entries in the crew file for the helper crew would be:

```
HOP100 200:3153 HOP100 225:3153 LITEHP 0:0 LITEHP 25:0
```

The above series of entries tells the economic model that the crew is to start duty in simulation HOP100 and temporarily stop duty in simulation HOP100. Since the simulation MTY100 occurs prior to LITEHP you do not want the mileage and time associated with simulation MTY100 included in the helper crews costs.

#### 2.2.9 The Track Maintenance Cost File (TMC)

An integral part of RECAP is the computer model called TMCOST. TMCOST, like TEM is a simulation model and computes the incremental (rail wear, fatigue, ties, and ballast) costs of running a specific consist over a specific route.

A version of TMCOST for general industry wide release is under development at the AAR Affiliated Lab at the Massachusetts Institute Of Technology (MIT) and will not be available until late 1989. However, output from TMCOST in the form of five TMCOST output files has been supplied with RECAP. In order to calculate the track maintenance costs associated with the

analysis you are performing one or more of the following files should be selected depending upon the particular type of consist you are running.

FILE NAME	DESCRIPTION
UNITH.TMC	Appropriate for a train consisting of primarily bulk commodity cars each weighing in excess of 267,000 pounds (including lading).
UNITL.TMC	Appropriate for a train consisting of primarily bulk commodity cars each weighing less than 267,000 pounds (including lading).
INTERM.TMC	Use this file if the consist is composed primarily of intermodal equipment.
EMPTY.TMC	Consist composed primarily of empty bulk commodity cars.
MIXED.TMC	All other consists.

Specification of the particular .TMC file you wish to use is accomplished at the same time you specify the simulation identifier (.INI file) and is discussed in the section labeled "RUNNING RECAP".

### 2.3 RUNNING RECAP (The Economic Model - "ECON2")

The economic model (ECON2) is designed to facilitate analyses of either a single consist running over a single section of track (one TEM .INI file, a single simulation) or an entire set of consists running over any number of sections of

track, i.e. a logical set of trains (any number of TEM .INI files through use of the RECAP .RCP file). The following documentation will indicate how to run the model to accomplish either of these tasks.

The name of the executable code file for the economic model is "ECON2.COM". This file name cannot be changed to any other name. Certain user specified parameters are written back to this file name upon execution of the program. If the file name is changed the executable code cannot be updated and it is possible to make the file unusable.

### 2.3.1 The RECAP Multiple Simulation File (.RCP).

If you wish to take multiple TEM simulations and generate costs for those simulations all at one time then you must use a RECAP .RCP file to specify which simulations are to be included in the analysis. This file must have the extension ".RCP". The .RCP file is nothing more than a list of those simulation .INI file names and the name of the TMCOST file you wish to have associated with each separate simulation.

You would want to use a .RCP file when, for example, you wish to cost both the loaded and empty movements of a bulk commodity train. There are no constraints on when and how to set up the .RCP file. If the analytical framework makes sense to you then use it.

The following is an example of a .RCP file that will string 3 TEM simulations together. On each line the first value is the TEM .INI file name and the second value is the Track Maintenance Cost (.TMC) file name. The data elements in this file are positional and must be listed in the proper order with at least 1 space between the entries. The example filename is COAL.RCP.

The first entry is for the the loaded move, the second for the empty move and the third for the light helper locomotive move.

Note the -T parameter on the second line. You may specify the turn time for locomotives and cars that will be used to calculate the equipment ownership costs separately for each TEM simulation. The model will use a value in the .RCP file if it exists. Otherwise it will use the value on the command line. See section 2.3.2, "The MS-DOS Command Line".

```
\TEM\HOP100    \TMCOST\UNITL
\TEM\MTY100    \TMCOST\EMPTY  -T24:00
\TEM\LITEHP    \TMCOST\EMPTY
```

The order in which the simulations were listed is not important to the model. It will generate the costs in the order you specify. One reasonable approach is to list the simulations in the chronological order that they would occur in an actual operation.

You may find it useful to place light locomotive moves last as it will make it easier to the analyst and anyone who reads the output to separate the main loaded and empty moves from the secondary helper moves. Its your choice.

Note that the MSDOS path names (\TEM\ for the .INI files and \TMCOST\ for the .TMC files) were specified. You have total flexibility as to where files are located. However, if the files are not on the current directory (where ECON2 resides) you have to tell ECON2 where the files are. All files associated with TEM either as input or output must be in the same directory.

### 2.3.2 The MS-DOS Command Line (C>).

All of the files and other user specified parameters are entered on the command line at the time of program execution. The following example command line will show you all the files and parameters that can be specified on the command line. Some of these parameters have default values and under certain circumstances can be eliminated.

```
C>ECON2 INI/RCP TMCOST CREW -I -D8% -F0.57 -T0:0 -V1000
```

#### PARAMETER

#### WHAT IT MEANS

ECON2

The filename of the economic model executable code. Typing *ECON2* with no other parameters will result in the program echoing the current default values for *discount rate, fuel price, time, and variance.*

The next three parameters are file name identifiers. These files are assumed to exist in the current MSDOS directory unless you specify otherwise. Use MSDOS drive and path conventions any time files are located on other than the current directory. You do not need to specify file extensions as these parameters are positional. They must be listed in the order described.

*INI/RCP*

The name of either an *.INI* file or an *.RCP* file. The *.INI* file is the TEM simulation initialization file that contains certain information that the economic model needs in order to run successfully. This information includes the name of the consist file, the route geometry file, etc. that the simulation used. An *.RCP* file is simply a list of *.INI* files and

TMCOST files that you wish to combine to generate costs for as a logical set of trains. This parameter is positional and must follow the program name (ECON2).

**TMCOST**

The name of the Track Maintenance Cost (.TMC) file that you wish to use. If you are generating the costs for a single .INI file and you want track maintenance costs calculated you must specify the file name. If you are using a .RCP file to list the .INI files you wish to include in the analysis the .TMC file does not need to be specified on the command line. However, you will need to place a period "." in this position if you are going to specify a crew file name. For example:

ECON2 COAL . CREW -i, etc.

**CREW**

The name of the crew data file (.CRW) that you want associated with this run. You will not need to specify a crew file unless you wish to use a crew file of a different name than either the .INI file or .RCP file. The default for the crew file (.CRW) name is the name of the .INI file or .RCP file. Therefore if you name your crew file either the same as the .INI or .RCP file the program will search for a file of that name (with the .CRW extension) on the same drive and path as the .INI or .RCP file. Furthermore, if you use the default crew file and specify a .RCP file on the command line in the first position you do not need to specify a TMCOST file as the TMCOST file can be listed in the .RCP file.

All of the remaining parameters are preceded by a keyletter. This keyletter must be preceded with a '-' and have no spaces between the '-' and the keyletter or the keyletter and the subsequent value.

- I                    This parameter (must be a "minus sign" immediately followed by an "I" or "i") tells the program that the following default values are to be used for this run and used for all subsequent runs until further modified. This parameter is not positional and can be placed anywhere on the command line. It would probably be useful to place this parameter after the filename parameters.
  
- D                    The discount rate you wish to have applied to the calculation of the equipment ownership costs being calculated from the equipment purchase price. There are two acceptable formats, either a real number or an integer followed by a percent sign. For example, -d0.08 or -d8%.
  
- F                    The price of fuel. There is only one acceptable format, a real number. For example, -f0.57.
  
- T                    The time value (in hours and minutes) that you wish to use as a basis for calculating the equipment ownership costs. The default value is the TEM .OUT file time that it took to complete the simulation. A value of -T0:0 would use the TEM .OUT file time. If you wish to override this default to account for locomotive and car turn time then specify a value such as: -T24:30 which means 24 hours and 30 minutes.

This parameter may also be specified for individual simulations in the .RCP file. If you use the .RCP file to set this parameter then set the parameter on the command line to -T0:0.

-V

Permits the user to control the economic model's printing of actual versus target TEM stopping distances. TEM permits the user to specify both a final and interim destinations (stops) for a consist. This is done in the TEM stop (.STP) file. Sometimes TEM is able to stop the train very close to its objective, sometimes TEM doesn't get that close. The economic model always finds the closest actual stop to the desired TEM stop. If no TEM stop occurs the economic model will use the closest TEM detailed output record (.OUT file) upon which to base various calculations. When a stop occurs that is in excess (either short or long) of this parameter's value a warning message will be printed on the "FUEL CONSUMPTION" report page. For example, if the value were -V1000 then any stop that occurs greater than 1000 feet from that specified in the TEM .STP file will generate a message to the user. This parameter generates messages for informational use only and has no impact upon the results.

>FILENAME

If you wish to have RECAP error messages routed to a file rather than to the screen then specify the output file name on the command line. The filename does not require a file extension but must be preceded immediately by a "greater than" sign.

## 2.4 OUTPUT FROM THE ECONOMIC MODEL

Each time the economic model is run six reports will be written to an output file. The file prefix is the name of the .INI or .RCP file specified as the first parameter on the MSDOS command line. The extension is always (.CSM). A description and sample of each report follows.

### 2.4.1 Input File List

The first report is a listing of the files that were used as input to the model. Note that in this example a .RCP file was used to facilitate the generation of costs for multiple TEM simulations. The MS-DOS command line for this simulation was:

```
>ECON2 \DRGW\COAL . \XFER\HOP100 -f0.57 -t0:0 -d8%
```

The name of the .RCP multiple simulation file list is \DRGW\COAL.RCP.

The track maintenance cost file is specified as "." because it is specified for each TEM simulation in the .RCP file.

The name of the crew data file is \XFER\HOP100.CRW.

The fuel cost is 0.57 per gallon, the discount rate is 8% and the model will compute the equipment ownership and repair costs based upon the time TEM indicated it took to run the trains.

The detail lines in this report are as follows:

<u>HEADING</u>	<u>DESCRIPTION</u>
FILE DESCRIPTION	A short description of the input file. Note: this is a good place to start looking if you find in subsequent reports that some of the information you thought you should have gotten is missing. For example if there is no crew cost information look at this report to find the crew file name.
PATH	The MS-DOS file location information.
NAME	The input file name.
LAST FILE UPDATE	The date and time of the last modification to this file.

#### 2.4.2 Crew Costs

This report provides information about each train crew that was specified in the crew file.

Since a crew can be on duty in more than one simulation crew costs are not allocated to any particular simulation. The costs are calculated by individual crew and then summed for the entire set of simulations.

<u>HEADING</u>	<u>DESCRIPTION</u>
CREW #	The position of the crew in the crew file. All crews are listed in sequential order.

**PAYMENT METHOD** Based upon the crew type indicator in the crew file. A crew may be compensated based upon an intra-divisional agreement - type (1), inter-divisional agreement - type (2), or hourly pay agreement - type (0).

**TOTAL TIME** The amount of time that each crew was on duty. The value is calculated from the TEM .OUT file based upon the time that required to run the train between the starting and ending track file positions specified in the crew file. Idle (dwell) time that occurs where the crew starts and finishes duty are not included in this value. However, idle time that occurs during the run when the crew is on duty is included in TOTAL TIME. Therefore if a stop is crew change point the time is not allocated to a particular crew. If the stop is not for a crew change (e.g., due to other traffic) then the time is allocated to any crew that is on duty at that time.

**CUM. TIME** The cumulative time for all crews up to and including the current crew. This is a running total of cumulative crew hours.

**TOTAL MILES** The number of miles each crew traveled from the start crew change point to where the crew goes off duty.

**CUM. MILES** The cumulative miles traveled for all crews up to and including the current crew.

**BASIC PAY**

The amount of compensation for 1 base day's pay if the crew type is either a '1' or a '2'. If the crew type is '0' (hourly) then the value is the crew hourly rate multiplied by the number of hours on duty (subject to minimum hours constraints, see the input file description for the crew file.)

**O-MILE PAY**

The compensation due each crew assuming the crew traveled in excess of the specified base day miles. The overmile rate is fixed if the crew type indicator is '1'. If the crew type indicator is '2' the overmile pay rate is calculated by dividing 1 base day's pay by 1 base day's miles. If the crew type is '0' (hourly) then the value is always 0.

**O-TIME PAY**

The compensation that each crew is due for overtime duty. Under the current industry labor agreement if a crew works both more than 8 hours and travels in excess of the base day miles the crew will almost always get overmile rather than overtime pay. The speed basis determination of compensation under these circumstances is programmed into the crew cost equations. If the crew type indicator is '0' (hourly) then overtime pay is computed as ((total hours worked - straight time hours) \* overtime rate).

**ARBIT & OTHER**

The value for arbitraries is taken directly from that field in the crew file.

**BENEFITS**

The fringe benefits are calculated by multiplying the fringe benefit rate field in the crew by the value for base day's pay. If the fringe benefit rate field is preceded by a '-' then the fringe benefit rate is multiplied by the total crew compensation. If the crew type is '0' (hourly) fringes are calculated based only on total crew pay.

**TOTAL COMP**

The sum of all the above fields starting with BASIC PAY.

**2.4.3 Equipment Costs**

This report computes the equipment running repair and ownership costs for each simulation (.INI file).

**HEADING**

**DESCRIPTION**

**EQUIPMENT TYPE**

The name given each vehicle in the TEM consist file (.CON).

**NUMBER UNITS**

The number of vehicles of each type.

**PURCHASE PRICE**

Taken from the modified TEM vehicle parameter (.PAR) file.

**HOURLY OWNERSHIP  
COST**

Taken from the vehicle parameter file if the value for DAILY CAR COST is > 0. Otherwise the hourly ownership cost is computed from the vehicle purchase price by computing an equivalent annual cost then converting that value to an hourly cost.

TOTAL HOURS  
OPERATED

Taken from the TEM output file (.OUT) as the time from beginning to end of the simulation. The value can be overridden by specifying a value greater than the simulation time for the "-T" parameter either on the MS-DOS command line or in the multiple simulation file (.RCP) list.

TOTAL OWNERSHIP  
COST

Number of Units \* Total Hours Operation \*  
Hourly Ownership Cost

REPAIR COST PER  
MILE

A user input value from the vehicle  
parameter file (.PAR).

TOTAL MILES  
OPERATED

The distance that the simulation covered.  
Taken from the TEM output file (.OUT).

TOTAL REPAIR COST

Number of Units \* Repair Cost Per Mile \*  
Total Miles Operated

TOTAL COST DOLLARS

Total Repair Cost + Total Ownership Cost

#### 2.4.4 Fuel Cost Report

The fuel cost report computes the amount of fuel consumed by the train in each simulation. The report disaggregates the fuel consumed between each stop specified in the TEM. stop (.STP) file.

<u>HEADING</u>	<u>DESCRIPTION</u>
STOP NUMBER	The name of the TEM .INI file (simulation name) and the number of the scheduled stop in sequential order.
SEGMENT LENGTH	The distance between the previous stop and the current stop.
ELAPSED MILES	The total cumulative miles from the specified origination of the first simulation .INI file.
RUNNING TIME	The discrete elapsed time from when the train velocity is >0 subsequent to the previous stop and the time when velocity = 0 at the current stop.
IDLE TIME	The time interval at which train speed = 0 at a selected stop.
ELAPSED TIME	The cumulative time (in hours and minutes) from the first detected beginning position of the train (taken from the first simulation's .INI file) to the last record in the .OUT file where the train speed = 0 for the current stop. If a stop did not occur in the .OUT file then RECAP will insert the dwell time specified in the .STP file and add that amount to the time calculation.

**IDLE FUEL**                    The amount of fuel consumed during defined "IDLE TIME". If no stop actually occurred RECAP will compute idle fuel as: (number of locomotives \* fuel consumption rate at idle \* .STP file specified idle time).

**RUNNING FUEL**                The amount of fuel (from the TEM .OUT file) reported to have been consumed by the consist during defined RUNNING TIME .

**FUEL COST**                    The price of fuel \* (running fuel + idle fuel). Price of fuel is entered on the MS-DOS command line through the '-F' parameter.

**CUMULATIVE FUEL CONSUMED**      The total amount of fuel consumed beginning at the starting position in the first simulation .INI file (if the fuel consumption at that point is >0 the fuel counter is set to 0) through the end of IDLE TIME at the current stop.

**CUMULATIVE FUEL EXPENSE**        Cumulative Fuel Consumed \* Price of Fuel

#### 2.4.5 Track Maintenance Costs

The track maintenance costs for any simulation are the calculated incremental costs for running the specific consist one time over a specific route geometry file. Section 2.5.2 "Track Maintenance Costs" will provide further information on how these costs were calculated.

<u>HEADING</u>	<u>DESCRIPTION</u>
START STATION	The station name (if any) from the position in the route geometry file (.TRK) the user specified as the desired beginning track segment for which to calculate track maintenance costs (either the beginning of the file or with an 's' in column 64 of the file, see section 2.2.3.
STOP STATION	The station name that the user specifies as the end of the route segment for which track maintenance costs are to be calculated. The default is the station name (if any) in the last record of the .TRK file.
LENGTH MILES	The physical length of the route segment between item 1 and 2 above.
TOTAL MGIM	The total million gross ton miles generated on this route segment. Calculated by multiplying the length of each segment by the TRAFFIC DENSITY field in the route geometry file.
CONSIST MGIM	The weight of the train consist (calculated from the .CON and .PAR files) multiplied by the length of each segment between item 1 and 2 above.
TANGENT MILES	The length of all track segments of less than 2 degrees curvature.

TANGENT COST	The cost of maintenance for track of less than 2 degrees curvature.
CURVED MILES	The length of all track segments of 2 or more degrees of curvature.
CURVED COST	The cost of maintenance for track of 2 or more degrees of curvature.
TOTAL TRACK MAINT. COST	Tangent Cost + Curved Cost

#### 2.4.6 Recap Cost Summary

This report summarizes the costs associated with each TEM simulation by cost categories. A final summary of total costs associated with all the simulations is provided. **NOTE:** since a crew can operate in more than 1 simulation crew costs are only allocated to the total cost summary report (the last report in the cost summary) and not individual simulations.

<u>HEADING</u>	<u>DESCRIPTION</u>
CATEGORY	All costs are allocated to one of 7 categories: locomotive ownership, locomotive running repair, car ownership, car running repair, fuel, track maintenance, or crew costs.
TOTAL COST	Taken from the total cost field on each of the 4 individual cost reports.

COST PER

1000 GIM

Total Cost / (Total weight of consist \*  
total train miles)

COST PER

1000 NIM

Total Cost / (Total Lading in consist \*  
total train miles)

COST PER

MILLION CFM

Primarily useful as a measure of  
productivity for intermodal equipment as  
cubic capacity is often reached prior to  
weight capacity. Calculated as (Total cubic  
capacity of all vehicles \* total train  
miles).

#### 2.5.1 Changing Operating Strategy w/o Rerunning the Simulation

ECON2 includes logic which attempts to insure that the  
output accurately reflects the operating strategy that the user  
establishes for any train.

Once the user understands this process it will be easy to  
modify certain aspects of the operating strategy without  
rerunning the TEM simulation.

As a general rule you can add interim destinations and  
change the origin location or final destination of the train.

For this discussion The TEM simulation can be viewed as  
having two components, an objective operating strategy as  
represented by the TEM stop (.STP) file and the actual operation  
as represented by the TEM simulation output (.OUT) file.

The stop file tells the simulator where (in the track file) the train is to stop, for how long, and what the throttle setting is to be during dwell time. The output file provides a detailed running record of how the simulator executed the operating strategy.

One of the aspects of TEM that enhances the accuracy of the simulation in regard to fuel consumption and running time is that the impact of train handling actions (changes in throttle position, brakes applications, etc.) are explicitly modeled. Because TEM essentially simulates the behavior of an engineer the actual execution of the operating strategy may deviate somewhat from the strategy itself. This is particularly true in regard to how close the automatic train handling algorithm is able to stop a train in relation to the objective specified in the stop (.STP) file.

Usually the actual stop is within several hundred feet of the objective. Sometimes the stop can be considerably short or long. Occasionally TEM will stop a train short of the destination and then start the train again to move it closer to the destination.

ECON2 will always find the actual stop closest to the desired stopping position as specified in the .STP file. If a stop does not occur then the .OUT file record closest to the desired stop is selected as the stop. ECON2 will print a message in the fuel consumption report if the stop does not occur.

After ECON2 has read the .STP file (the objective), a domain of potential candidate stops is established. This domain is defined as  $1/2$  the distance between any specified stop and the previous stop and  $1/2$  the distance between the specified

stop and the next stop. In the case of the train origin location and the final destination only 1/2 the distance between the second stop and the next to last stop is considered.

Therefore, any time the train speed reaches '0.00' a potential candidate for the stop exists.

When the TEM .OUT file is read, all actual stops that occur within the defined domain are extracted and evaluated. From those candidate stops the one that is closest to the desired stop is selected. The fuel consumption and running time between each selected stop and the previous selected stop is computed and the stop is matched with the specified crew change locations in the crew (.CRW) file.

As previously indicated there are three changes you can make to the operating strategy that will most likely give you reasonable results without having to rerun the TEM simulation:

- (1) Add (but not delete) interim destinations.
- (2) Advance the origin position of the train.
- (3) Set back the destination position of the train.

The first change is useful when you run a simulation but forget to add a stop for a crew change, train meet, etc. By editing the TEM .STP file and adding a destination EON2 will create an artificial stop from the closest record in the TEM output file. It will then add the appropriate dwell time and calculate the idle fuel consumption from the locomotive .PAR file. However, you cannot delete destinations as the actual stop would then become a part of another destination's domain of potential stops and may be selected as a stop even though you wanted it to be ignored.

The second and third options are useful if you want only to cost specific segments of a simulation and not necessarily including where the train actually started or the final destination. To accomplish a change in starting position change the train origin point in the TEM. INI file. To accommodate a change in final destination, edit the last record in the TEM. STP file.

ECON2 always attempts to generate costs based upon the objectives established in the .INI and .STP files. For example, assume you have a 1000 mile simulation starting at mile 0 in the .TRK file and ending at mile 1000. If you only wish to generate costs for mile 250 through mile 750 you would change the origin location in the .INI file to 250 and the final record in the .STP file to 750.

Even though the TEM output file starts recording the simulation at mile 0, ECON2 will read the file until it finds the closest record to mile 250 and set its distance and fuel consumption accumulators to 0. All of the model's calculations of fuel consumption and running times are relative to the user's objectives.

## 2.6 The Track Maintenance Cost Files (.TMC)

This intent of this section of the manual is to provide you with information about how the track maintenance costs provided with RECAP were generated. It will also provide information as to when the costs are useful and when they are not.

The five track maintenance cost files were generated using TMCOST, an engineering simulation model that computes the track deterioration generated by one run of a specific traffic mix over a specific route. Unlike typical accounting based track maintenance costing packages, TMCOST is sensitive to the primary consist specific factors (wheel loading, unsprung mass, center

of gravity height, and wheel diameter), route geometry, and track construction factors which affect track deterioration.

The cost files were generated by first establishing a base case consisting of what was considered to be an average traffic mix (certain percentages of 4 axle and 6 axle locomotives, and several different car types). This "average" traffic mix was based upon a study done at a particular U.S. railroad.

The unit costs for rail, labor, ties, ballast, etc. were based upon the R&T staff's best information of industry wide average costs. The labor costs were adjusted upward to account for lower track gang productivity in high traffic density territory.

The "average" consist was run over a route geometry file that contained twenty-four one mile long segments of track representing every combination of 0 through 2.5 percent grade and 0 through 10 degrees curve. The simulations represented each of 8 rail types (heavy, medium, and light rail, premium and standard metallurgy and, jointed and CWR) under lubricated and non-lubricated conditions in each traffic density category.

The output of the first set of simulations was the cost of running the "average" traffic mix over each 1 mile route segment taking into account rail type, lubrication and traffic density.

The five track maintenance cost files were generated by taking the engineering characteristics (locomotive and car characteristics) of five different typical consists [(1) unit train with heavy axle loadings - 39,000 pound wheel loads, 315,000 gross weight per car, (2) unit train with standard axle loadings - 32,500 pound wheel loads, 263,000 gross weight per car, (3) intermodal, (4) unit train empty and (5) mixed

locomotives and car types] and adding a traffic increment consisting of one of these consists to the base case "average" traffic mix and rerunning each of the simulations.

The differences in the costs of the base case and the costs of the base case plus one of the five additional consists is the incremental cost of running one of these additional consists over each combination of track geometry, traffic density, rail-type, and lubrication status.

The incremental costs generated were then divided by the tonnage (in millions of tons) that represented the traffic increment. Since each route segment is 1 mile long the resulting incremental costs are in dollars per million gross ton-miles.

Conceptually, you can view each track maintenance cost file as a very long list of cost values that are accessed by ECON2 on the basis of:

- (1) Consist Type
- (2) Rail Type
- (3) Lubrication Status
- (4) Consist Weight
- (5) Route Traffic Density
- (6) Gradient
- (7) Curvature

#### 2.6.1 When the Track Maintenance Cost Files are Useful

The track maintenance costing component of RECAP is most useful for comparative analysis of operating strategies and/or technologies. When evaluating alternative courses of action the analyst is really looking for gross differences in either the

cost or performance characteristics associated with the competing alternatives.

TMOOST is arguably the most comprehensive tool available for determining the impact of engineering factors upon track deterioration. Being sensitive to those engineering factors, the cost files provided will give the RECAP user a very good feel for the degree of difference in incremental costs that more traditional accounting based approaches cannot provide.

#### 2.6.2 What to Avoid

The track maintenance costs are not useful for absolute cost measurement.

TMOOST only computes costs associated with track (from the rail down to the sub-grade) deterioration. No costs for administration, bridges, brush removal, etc. are included. These costs, though real, are not directly related to either consist specific technology, operating strategy, or track characteristics.

The costs are not useful for evaluating the comparative economics of rail types.

TMOOST was extensively calibrated for 132 lb. rail. It was not so well calibrated for lighter weight rail. For example, the fatigue model calculates that fewer defects will develop in light weight rail on soft, wet subgrade than in 132 pound rail on stable, dry subgrade. A conclusion that is not in accordance with the majority opinion in the railroad engineering community. However, the cost differences associated with a light traffic mix over medium and light weight rail versus a heavy traffic mix over the same type rail are probably quite reasonable.



### 3.0 TKSPLICE (Track File Management System)

TKSPLICE is a program that enables the user to create new route geometry (.TRK) files out of files that already exist. TKSPLICE permits one to:

- (1) Extract of a specified part of an existing track file.
- (2) Reverse all or part of a track file.
- (3) Splice all or part of two or more track files together into one file.
- (4) Extract portions of two or more files, reverse them, then splice them together in one step.

TKSPLICE will allow you perform any combination of track file extraction, reversing, and splicing that you can conceive of. As a caution however, TKSPLICE has only a limited ability to determine that what you are trying to accomplish is reasonable. You can instruct the program to splice a section of track in Colorado to a section of track in Illinois and it will do so.

#### 3.1 How to Use TKSPLICE

##### 3.1.1 The MS-DOS Command Line

There are three parameters which are specified on the MSDOS command line, two of which are mandatory and one which is

optional. The format of the command line follows:

C>TKSPLICE <COMFILE >MESSAGES

Each of the parameters must be separated by at least one space. The "<" and ">" symbols are required and no spaces can exist between those symbols and the filename that follows.

PARAMETER

DESCRIPTION

TKSPLICE

The program name, must be the first entry on the command line (to state the obvious).

<COMFILE

A user specified filename that contains the instructions to TKSPLICE as to how the track files are to be manipulated. The "<" is required and instructs the program to read the named file as if it were input from the terminal. The command file name can be any legitimate MS-DOS filename (8 characters or less) and no file extension is required. The file will be created using any resident ASCII text editor.

>MESSAGES

TKSPLICE generates error messages as well as a running log of the process that is being performed. If you do not specify a messages output file, the messages will be sent to the screen. This parameter is optional.

### 3.1.2 The TKSPLICE Command Language

TKSPLICE has a very simple command language consisting of a few keywords that instruct the program which operations to perform. These keywords both must be spelled correctly and are positional (must be placed in the right order). Fortunately, the positioning of the keywords is logical and will be evident to the user from the beginning. Use either upper or lower case letters.

KEYWORD	DESCRIPTION
BEGIN	BEGIN tells TKSPLICE to create a new file. All command files must have BEGIN as the first word in the file. You must have a BEGIN every time you create a new file. However, you may create as many files as you wish in one command file. Immediately following BEGIN (separated by a space) you will specify the filename to be created. No .TRK extension is required, it is assumed. MSDOS drive and/or path specifications are required if the new file is to be written to other than the current directory.
END	END tells TKSPLICE that the end of the command file has been reached. END must be the last line of the command file and must not occur more than once in the file. END is TKSPLICE's end-of-file marker.
EXTRACT	EXTRACT tells TKSPLICE that some subset of a specified file is to be written to the new file. If EXTRACT is an operation you choose then EXTRACT must be the first command on the line. EXTRACT is always followed by an existing filename.

**REVERSE** REVERSE tells TKSPLICE that some subset of a specified file is to be extracted, reversed and written to the specified new file. REVERSE is always followed by an existing filename.

**FROM** A keyword that indicates the beginning position (in feet from the start of the file) of an existing file to be reversed or extracted. If FROM is not specified then the default start position is foot 0 in the existing file.

**TO** Specifies the position (in feet from the beginning of the file) that the extraction or reversal process is to conclude.

To illustrate the concepts surrounding these keyword the following is a sample command file. The next section will give examples of how to perform several operations that will often be useful. The sample command file is named BUILD. The file resides on the same directory as does TKSPLICE. The source input track files and the target output files reside on a sub-directory named TRACK. The TKSPLICE messages will be written to the screen.

The MSDOS command line:

```
C>TKSPLICE <BUILD
```

The file BUILD. looks like:

```
BEGIN \TRACK\ATOC [New file, .TRK is assumed]
EXTRACT \TRACK\ATOB FROM 0 TO 100000
EXTRACT \TRACK\BTOC FROM 0 TO 250000
BEGIN \TRACK\CTOA
REVERSE \TRACK\ATOC
END
```

The default for the FROM and TO positions is always respectively the first record and last record in the file. You need to specify FROM or TO only if you want a selected portion of the file extracted. You may specify a FROM or a TO by itself. If one or the other is missing the default (first or last record in the source file) will be selected.

Note also that creating the file CTOA (the reverse of ATOC) was done in two steps. The two step process works but is not necessary and wastes space if you really only wanted one output file; the product of files ATOB and BIOC spliced then reversed. Another approach would have been:

```
BEGIN \TRACK\CTOA
REVERSE \TRACK\BIOC FROM 0 TO 250000
REVERSE \TRACK\ATOB FROM 0 TO 100000
END
```

If your files BIOC and ATOB were respectively 250000 and 100000 feet long the the following would have worked just as well:

```
BEGIN \TRACK\CTOA\
REVERSE \TRACK\BIOC
REVERSE \TRACK\ATOB
END
```

Note that when you have more than one source track file (BIOC and ATOB) the process of splicing the files is implied. You do not need to tell TKSPLICE to put two files together.

### 3.1.3 Example TKSPLICE Command Files

The following examples will show you various ways of performing functions will probably need at some point in time.

Examples of using alternate sub-directories and disk drives are included:

(1) Extract a portion of an existing file from the middle to the end of the file:

```
BEGIN C:\TRACK\BTOD
EXTRACT A:ATOD FROM 54332      [TO end of file implied]
END
```

(2) Extract a portion of an existing file from the beginning to some point short of the end.

```
BEGIN F:\TRACK\ATOC
EXTRACT A:ATOD TO 54332      [FROM start of file implied]
END
```

(3) Splice all of three files together from your track file sub-directory and put them in the Train Energy Model Sub-directory.

```
BEGIN \TEM\ETOH
EXTRACT \TRACK\ETOF
EXTRACT \TRACK\FTOG
EXTRACT \TRACK\GTOH
END
```

(4) Do number 3 above but only get the first 20000 feet of the third file.

```
BEGIN \TEM\ETOH
EXTRACT \TRACK\ETOF
EXTRACT \TRACK\FTOG
EXTRACT \TRACK\GTOH TO 20000
END
```

(5) Reverse all of one file.

```
BEGIN \TEM\HTOE  
REVERSE \TEM\ETOH  
END
```

(6) Splice and reverse three complete files.

```
BEGIN \TEM\HTOE  
REVERSE \TRACK\GTOH  
REVERSE \TRACK\FTOG  
REVERSE \TRACK\ETOG  
END
```

It is important to understand the order in which TKSPICE performs functions especially when track files are being reversed. In order to avoid making some very strange mistakes, (especially if you are reversing parts of multiple files and creating one new file) you may wish to perform the functions in logical steps and create at least one interim file in the command file.

TKSPICE does everything in the exact order that you specify in the command file. In regard to the implied splice function, the splicing is always done after the extraction or reversal process. The output file is always written to disk in the order that you specify the input files.

For example, you have two input files ATOB and BTOC. Assume that BTOC starts at the same location that ATOB ends. Your objective is to get a file that starts at location C and

ends at location A. Look at the appropriate correct commands to accomplish this:

```
BEGIN CTOA  
REVERSE BTOC  
REVERSE ATOB  
END
```

In this example TKSPLICE will perform these steps in the following order:

- (1) Create empty file named CTOA.TRK
- (2) Read file BTOC
- (3) Reverse file BTOC
- (4) Write reverse of BTOC to file CTOA
- (5) Read file ATOB
- (6) Reverse file ATOB
- (7) Concatenate reverse of ATOB to file CTOA.

The result is that the starting position (0.00 feet) of new file CTOA is the same as the ending the position of file BTOC, namely location C. The end of file CTOA is the same as the beginning of file ATOB, namely A. The result is what you desired.

On the other hand if the command file had been:

```
BEGIN CTOA  
REVERSE ATOB  
REVERSE BTOC  
END
```

The result would be a file where location B was at the beginning, location A was in the middle, and location B was at the end. TKSPLICE would probably issue a warning message in this case.

The program checks to see if the last record in one file has the same elevation, grade and curvature as the first record in the next file. This should be the case if the two files are physically connected. If the elevation, grade and curve do not match, the program will force the first record of the second and all subsequent spliced files to conform to the last record of the previous file. All subsequent elevations will be adjusted so that the file will not contain vertical walls or cliffs.

A final note: BE CAREFUL WHAT YOU ASK FOR, FOR YOU WILL SURELY GET IT.

TKSPLICE takes care of all housekeeping tasks on file splices and reverses such as changing left curves to right curves, recalculating the position (in feet from the beginning of the file) all the second and all subsequent spliced files, etc.



#### 4.0 TKEDIT (Track File Editor and Verifier)

The track file editor enables the user to scan, perform format and logic checks or to modify an existing track file. TKEDIT also permits you to build a track file from scratch. This is a very labor intensive task and is recommended only as a last resort.

TKEDIT is a full screen editor similar in many ways to several commercially available editor and word processing packages. TKEDIT also includes a feature where you can program up to 40 function keys to execute a series of commands to manipulate the file.

#### 4.1 Using TKEDIT

##### 4.1.1 The MS-DOS Command Line

The program name is TKEDIT. On the command line you specify the program name and the name of the file you wish to edit. The format for the command line follows:

```
C>TKEDIT filename
```

Since you can only edit files with the extension .TRK, you need not specify the file extension. If the file you wish to edit exists on other than the current directory or drive, you need to specify it.

If the file you specify does not exist, then you will be asked if you want to create a new file. If you answer 'N' the program will terminate and you will be back to the command

line. If you answer 'Y' a blank screen with the fields labeled will be brought to your screen and you can start filling in the appropriate data.

An example MS-DOS command line would be:

```
C>TKEDIT \TRACK\SAMPLE
```

#### 4.1.2 TKEDIT Commands

On the next page you will find an example of the TKEDIT screen. Notice that each of the data elements is labeled and the format of each field conforms to TEM specifications. The last two fields, "R" (RAILTYPE) and "MGT" (TRAFFIC DENSITY) are the additions required for RECAP's economic model. The field labeled "D" (CREW DISTRICT BOUNDRY) was required for an earlier version of RECAP. You may ignore the field completely.

The following is a list of commands for performing each TKEDIT function. The commands are listed by functional category. When there is more than one command listed any of the alternatives may be used to perform the function. NOTE: the symbol '^' is the key labeled "CTRL". An "F" preceding an integer is a corresponding 'function' key.

When using the 'CTRL' key keep in mind that this key must remained pressed down while the remainder of the keystrokes are entered.

#### 4.1.2.1 MOVING FROM PLACE TO PLACE IN THE FILE

<u>KEYSTROKE(s)</u>	<u>FUNCTION</u>
LftArrow, ^S	Move cursor left one character
RgtArrow, ^D	Move cursor right one character
Enter, ^M	Move cursor to next selected field. (See ^KM in "OTHER FUNCTIONS").
UpArrow, ^E	Move cursor up one line.
DnArrow, ^X	Move cursor down one line.
PgDn, ^C	Page forward one screen (less 4 lines).
PgUp, ^R	Page back one screen (less 4 lines).
^Z	Scroll up one line.
^W	Scroll down one line.
Tab, ^F, ^I	Tab right to next field
Shift-Tab, ^A	Tab left to previous field
F3, ^Home, ^QR	Move cursor to top of file.
F4, ^End, ^QC	Move cursor to end of file.
Home, ^QE	Move cursor to top line on screen.
End, ^QX	Move cursor to last line on screen.

#### 4.1.2.2 SEARCHING THE FILE

<u>KEYSTROKE(s)</u>	<u>FUNCTION</u>
F9, ^QF	Find character string in field where cursor is positioned scanning <u>forward</u> from current line. (Prompt for target character string will be at top of screen)

F8, ^QB	Find character string in field where cursor is positioned scanning <u>backwards</u> from current line. (Prompt for target character string will be at top of screen).
^L	Find next occurrence of previously entered target character string. (Repeats either of previous "FIND" commands).
F6, ^PgDn, ^QZ	Find next record (scanning forward in file) with a non-blank or nonzero value in field where cursor is currently positioned. (useful for finding next station name in LOCATION field)
F7 ^PgUp, ^QW	Find next record (scanning backwards in the file) with a non-blank or nonzero value in field where cursor is currently positioned.

#### 4.1.2.3 INSERTIONS, DELETIONS, COPY, and EDITING.

When you wish to change any field or enter data, just start typing in that field. The default data entry mode is 'overstrike', i.e. the character you type replaces the current character. The 'INS' key toggles between overstrike and insert mode.

KEYSTROKE(s)	FUNCTION
BackSpace, ^H	Deletes character to the left of the cursor.
DEL, ^G	Deletes the character at the cursor.

^Y	Deletes the record at the current cursor position.
^N	Inserts a blank record at the current cursor position. (Pushes the current record down).
Ins, ^V	Toggle between insert and overstrike mode.
F5, ^KR	Toggle field replacement. Replaces the next field with whatever value is in the current field. (Note: a double headed arrow will appear at the top of the screen in the field label when the toggle is 'on').
^F5, ^KC	Toggle field copy. Copies into the next field the value that is in the current field <u>if the next field is blank</u> . (Note: a single headed arrow will appear at the top of the screen in the field label when the toggle is 'on').

#### 4.1.2.4 EXITING TKEDIT

KEYSTROKE(s)	FUNCTION
^KQ	Exit the editor <u>without</u> saving the changes made during current TKEDIT session.
^KX	Exit editor <u>after</u> saving changes.
F10, ^KS	Save all changes during current session but do not exit TKEDIT.

#### 4.1.2.5 OTHER FUNCTIONS

KEYSTROKE(s)	FUNCTION
<b>^KM</b>	<p>Toggle field selection for carriage return (Enter key). For any field the cursor is positioned at when this function is toggled 'on', the carriage return or tab keys will skip over the field. The field will be displayed at 1/2 intensity when toggle is 'on'. Entering the keystroke sequence again will toggle the field from 'on' to 'off'.</p>
<b>F2. ^QV</b>	<p>Validate records from current line to end of file. This function checks each field for format or column position errors. If the data in a field is an invalid type (i.e., character data in a numeric field) the validation will stop until the field is corrected.</p> <p>If you do not want to correct the field then hit the 'ESC' key and move the cursor to the next field (or line) and restart the validation process.</p> <p>The validation process will correct all GRADE values based upon the difference in elevation from the previous to the current record. Additionally, the validation process does range checks on the MGT, TRACKSIDE LUBRICATOR and RAILTYPE fields. TKEDIT also checks to determine is the next</p>

position field is greater than the previous position. TEM does not allow negative distances between records.

If for certain fields, for example GRADE, you do not want the validation process to stop if an error is encountered then toggle the verify for that field from 'on' to 'off' (see the next function). The validation routine will still correct format and computational errors where possible but will not stop for each error.

F1, ^KV

Toggle verify for field on which cursor is currently positioned. This function turns the verification process 'off' or 'on' for a specific field. In general you will want to toggle verification off for the GRADE field and let TKEDIT make the grade correspond to the elevation changes. If you have fields that are intentionally blank, especially ones used only for the economic model but not TEM (RAILTYPE and TRAFFIC DENSITY) you will want to toggle those off also. You should probably leave all other fields toggled on. To perform this function just tab to the field of interest and hit the F1 key. A blank space in the field label will exist when the verification for the field is toggled 'off'. The default is 'on'.

^P

The command to invoke the programmable function key capability of TKEDIT. Section 4.1.3 explains this feature in detail.

#### 4.1.2.6 RECAP OF PROGRAMMED FUNCTION KEYS

Although you may program up to 40 function keys to perform any logical series of keystrokes you wish a number of the most commonly used features of TKEDIT have been pre-programmed for you. By hitting the function key these commands will be automatically executed. The following is a recap of the default programmed function keys. You can change these defaults if you wish.

FUNCTION KEY	FUNCTION
F1	Toggle verify 'off' or 'on' for the field at the current cursor location.
F2	Validate records from current line to end of file.
F3	Move cursor to first line in the file.
F4	Move cursor to last record in the file.
F5	Toggle field replacement 'off' or 'on'. Remember this feature replaces what currently exists in a field with the value that exists in the field in which the cursor was located when the function was set from 'off' to 'on'.
^F5	Toggle field copy 'off' or 'on'. This feature only copies from the source field to a blank field. If the target field is non-blank the value in the target field remains unchanged.

FUNCTION KEY	FUNCTION
F6	Move cursor forward to next non-blank or nonzero field.
F7	Move cursor backwards to the next non-blank or nonzero field.
F8	Find character string in current field scanning backward through the file.
F9	Find character string in current field scanning forward through the file.
F10	Save changes to the file but do not exit the editor.

#### 4.1.3 Additional Features of TKEDIT

One of the objectives in the design of TKEDIT was to increase the productivity of the RECAP/TEM user in regard to establishing the simulation operating strategy. This is especially true in regard to building the TEM stop and RECAP crew files. TEM requires that the position of each track segment exist in terms of the number of feet from the first segment (always at 0 feet).

However, TEM also requires that the initial position of the train (given in the .INI file) and the position of all interim stops and the final destination (in the .STP) file be provided to TEM in terms of miles and feet from the beginning of the track file.

TKEDIT continually displays, at the bottom of the screen, both the record number (line number) and the position in miles plus feet that the cursor is on. Therefore, when establishing stops for crew changes, trains meets, final destination, etc., just mover your cursor (using the various FIND commands in TKEDIT) to the place in the file you want the train to stop and that position will be converted automatically into miles and feet.

#### 4.1.4 Programming the Function Keys

Any of the TKEDIT commands (except the two commands to exit the editor) available to the user can be programmed into one of 40 function keys. The purpose of this feature is to allow 1 keystroke to do the work of several.

The 40 keys available for programming are the function keys:

F1 through F10  
SHIFT F1 through F10  
ALT F1 through F10  
CTRL F1 through F10

In order to program your function keys perform the following steps:

- (1) Press the CTRL and 'P' keys at the same time. You will be prompted (at the bottom of the screen) to press the key(s) you wish to have 'learn' the keystroke sequence.

- (2) Press the function key(s) you wish to have programmed. For example, press CTRL and F9 at the same time. You will be prompted to enter the command sequence you would like to have performed by the function key.
- (3) Enter the command sequence. For example, if you want CTRL-F9 to insert a blank record in the file you would press CTRL and N at the same time.
- (4) Press the CTRL and P keys again. This tells TKEDIT that the command sequence to be programmed is over.

You may program several commands into one function key. Just be careful to insure that the commands you want performed by the function key are logical and not inconsistent. For example, programming one key to insert then delete the same record isn't a logical function.

#### 4.1.5 Entering Repetitive Data Into Blank Fields

Most user's will build route geometry files (.TRK) from existing TPC, TPS, or TDA format files. The data required for the RAILTYPE and TRAFFIC DENSITY fields in the .TRK file will not be available from these sources. If you must enter that data by hand here is a relatively quick method of doing so.

Since much of this data is repetitive (the same value will occur for many route segments) it is easy to enter the value once and then copy it to the other appropriate route segments.

For example, if the RAILTYPE for the first 40 miles of your route is '1' (132 pound, premium, CWR) perform the following steps.

(1) Enter TKEDIT specifying the existing file you wish to edit:

TKEDIT \path\filename

(2) Move the cursor (TAB key) to the RAILTYPE field (Labeled "R").

(3) Hit F1 key to turn off verification on the field.

(4) Enter the RAILTYPE code for the first segment by typing in a '1'.

(5) Make sure the cursor is on that field after you enter the data.

(6) Hit the ^F5 keys to toggle copy into blank field 'on'.

(7) Press the "DnArrow" key to copy the code through the file until the miles/feet indicator at the bottom of the screen indicates you are approaching 40 miles. The DnArrow key can be held down continuously for very fast copying.

(8) Repeat steps 2 through 7 each time the RAILTYPE changes. The same procedure can be used for the MGT (traffic density field).

(9) Periodically hit the F10 key to save the changes.

It should be noted that TKEDIT creates a backup file as soon as you enter the editor. It has the extension .BAK. This is for your safety in the event you make a major mistake, the computer fails, or there is a bug in TKEDIT that we have yet to discover.

BIBLIOGRAPHY

1. Drish, W.F., "Train Energy Model-User's Manual-Version 1.5", Association of American Railroads, Chicago, Illinois, Research Report, R-711, February, 1989.