DOCUMENT RESUME

ED 064 531	AA 001 010
TITLE	APEX (Air Pollution Exercise) Volume 2: Computer Operator's Manual.
INSTITUTION	Environmental Protection Agency, Research Triangle Park, N. C. Office of Manpower Development.
PUB DATE	[71]
NOTE	121p.
EDRS PRICE	MF-$0.65 HC-$6.58
DESCRIPTORS	City Problems; *Computer Assisted Instruction; *Computer Programs; *Environmental Education; Management Games; *Models; Pollution; Simulated Environment; *Simulation; Urban Environment

ABSTRACT

The Computer Operator's Manual is part of a set of 21 manuals (AA 001 009-001 029) used in APEX (Air Pollution Exercise), a computerized college and professional level "real world" game simulation of a community with urban and rural problems, industrial activities, and air pollution difficulties. The information in the manual is sufficiently basic to provide the reader unfamiliar with computers with the instruction necessary to get the program running. The APEX computer program (not included in manuals) is in Fortran IV and runs on an IBM 1130-2B computer or IBM 360-50 or higher series computer. The program utilizes the many "hands on" features of the computer, enabling the operator to interact freely with the system. The game simulation procedure is further described in the resume for AA 001 009. (PR)

Environmental
Protection
Agency

APEX

Air Pollution Exercise

Simulation
Exercises

VOLUME 2

COMPUTER OPERATOR'S MANUAL

ENVIRONMENTAL PROTECTION AGENCY

APEX VOLUME 2
COMPUTER OPERATOR'S MANUAL

This course is designed for professional persons in the field of air pollution control. The course manual has been prepared specifically for the trainees attending the course, and should not be included in the reading lists of periodicals as generally available.

CONDUCTED BY

The Office of Manpower Development's
Institute for Air Pollution Training

ENVIRONMENTAL PROTECTION AGENCY
Office of Air Programs
Office of Manpower Development
Institute for Air Pollution Training
Research Triangle Park
North Carolina 27711

APEX•VOLUME 2
COMPUTER OPERATOR'S MANUAL

CONTENTS

IV

LIST OF MANUALS USED IN THE APEX GAME SIMULATION

Volume 1:	Game Director's Manual
Volume 2:	Computer Operator's Manual
Volume 3:	Air Pollution Control Officer's Manual
Volume 4:	City Politician's Manual
Volume 5:	County Politician's Manual
Volume 6:	Industrialist's Manual: No. 1, Shear Power Company
Volume 7:	Industrialist's Manual: No. 2, People's Pulp Plant
Volume 8:	Industrialist's Manual: No. 3, Rusty's Iron Foundry
Volume 9:	Industrialist's Manual: No. 4, Caesar's Rendering Plant
Volume 10:	Industrialist's Manual: No. 5, Dusty Rhodes Cement Co.
Volume 11:	Developer's Manual: No. 1
Volume 12:	Developer's Manual: No. 2
Volume 13:	Developer's Manual: No. 3
Volume 14:	Developer's Manual: No. 4
Volume 15:	Developer's Manual: No. 5
Volume 16:	Developer's Manual: No. 6
Volume 17:	Developer's Manual: No. 7
Volume 18:	City Planner's Manual
Volume 19:	County Planner's Manual
Volume 20:	Reference Materials
Volume 21:	Legal References: Air Pollution Control Legislation

SECTION A

GENERAL INFORMATION

ENVIRONMENTAL PROTECTION AGENCY
Office of Air Programs • Office of Manpower Development.
Institute for Air Pollution Training

The Institute for Air Pollution Training (1) conducts training for the development and improvement of State, regional, and local governmental air pollution control programs, (2) provides consultation and other training assistance to governmental agencies, educational institutions, industrial organizations, and others engaged in air pollution training activities, and (3) promotes the development and improvement of air pollution training programs in educational institutions and State, regional, and local governmental air pollution control agencies.

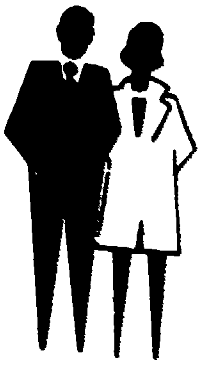
One of the principal mechanisms utilized to meet the Institute's goals is the intensive short term technical training course. A full time professional staff is responsible for the design, development and presentation of these courses. In addition the services of scientists, engineers and specialists from other EPA programs, governmental agencies, industry, and universities are used to augment and reinforce the Institute staff in the development and presentation of technical material.

Individual course objectives and desired learning outcomes are delineated to meet specific training needs. Subject matter areas covered include process evaluation and control, atmospheric sampling and analysis, field studies and air quality management. These courses are presented in the Institute's resident classroom and laboratories and at various field locations.

Harry P Kramer

Harry P. Kramer, Sc. D.
Director, Office of Manpower Development

ACKNOWLEDGEMENT



A cooperative effort of
The Comex Research Project
School of Public Administration
University of Southern California
Los Angeles, California

Dr. K. William Leffland,
Principal Investigator

Richard K. McGinty,
Project Director

Alan Kreditor,
Research Design

Ira Robinson,
Research Design

And
The Environmental Simulation Laboratory
School of Natural Resources
University of Michigan
Ann Arbor, Michigan

Dr. Richard D. Duke,
Principal Investigator

Roy I. Miller,
Project Director

Steward D. Marquis,
Research Design

Donald F. Kiel,
Systems Analyst

Developed under a Grant from
The Environmental Protection Agency
Office of Air Programs
Research Triangle Park, North Carolina

STAFF

Aubrey Boyd

Jolene Elliott

Corinne Floyd

Alan Forrest

Mark James

Frank King

Paula Magzamin

Charles Pratt

Gilbert Siegel

STAFF

Thomas Borton

Anne E. Cochran

Ferdinand Dijkstra

Katherine Fenn

David Kaueper

Marilyn Miller

Ellen Pechman

James Reeds

David Specht

IX

APEX

Air Pollution Exercise

Simulation Exercises conducted by the Institute for Air Pollution Training

The responsibility of the Federal Government's Office of Air Programs to provide leadership and assistance to State and local air pollution control agencies in the recruitment and development of qualified personnel is a major theme of the 1970 Clean Air Act. The Office of Air Programs, (OAP) in conjunction with the University of Southern California and the University of Michigan, has created and developed a simulation exercise identified as APEX (Air Pollution Exercise). This exercise establishes a dynamic atmosphere in which the trainees participate in a "real world" simulation involving a community with urban and rural problems, industrial activities, and a variety of air pollution control problems.

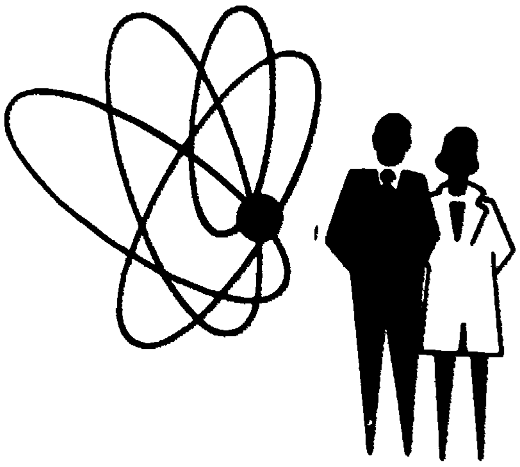
Current and projected uses of APEX have been developed through several of the University Consortia established in conjunction with OAP's Office of Manpower Development.

The use of simulation exercises for the training of air pollution control professionals offers two immediate and vital benefits:

1. A means is provided for a working application of theoretical knowledge; the learner applied information and skills to "real life"

situations. In addition, motivation directed toward additional learning results from participation in seeking solutions to the problems.

2. The focus is provided for solving problems through an interdisciplinary approach, where the interrelationship between "formal" areas of study and application becomes evident.



Students participating in APEX assume the roles of a number of decision makers: city and county politicians, city and county planners, developers, industrialists, air pollution control officers, and concerned citizens. Realistic data are supplied for each role, and the students are required to make decisions that are then analyzed by the computer. Next, the results of the decisions are presented as new situational data representing a year of "actual time." Students participating in these programs — which place special emphasis on air pollution problems — employ a wide range of skills and knowledge in a variety of areas. Additional opportunities for growth are provided through seminars, lectures, texts, and working contact with recognized authorities in a number of professions.

Within the overall format of the simulation exercise, emphasis is placed upon specific areas through the use of special situations, for example, hearings on air pollution standards or legal actions brought against a particular industry.

Additionally, preparations are underway to introduce APEX as a graduate course at OAP's new Technical Center in the fall of 1971 for students from the Triangle Universities Consortium. In addition to its use at the University of Southern California, APEX is now being conducted as a graduate course at the University of Illinois at Urbana and at Harvard University as part of an Environmental Education program for both graduate and undergraduate studies.

SECTION I

INTRODUCTION TO APEX

Section 1-1

Introduction to - APEX

APEX is one of, if not the most complex gaming-simulations of an urban area in use today. Although it was designed to supplement standard teaching methods, APEX is far more than an educational tool. It is a communication channel of a new kind -- capable of providing both the language and the forum for information transfer between persons and groups with different educational and cultural backgrounds as well as different prespectives on urban life.

APEX is composed of two essential components (1) a computerized system made up of a series of well-integrated simulation models (2) linked to a "gamed" environment encompassing a series of interactive roles. The computerized system predicts the changes that occur in several sectors of urban life in response to the decisions made by participants in the "gamed" environment, decisions made by persons outside the "gamed" environment (other actors whose behavior is simulated in the computer), and external pressures on the city (also simulated in the computer).

The county of APEX is run year by year by a set of elite decision makers performing both the mundane and extraordinary functions of their office in the "gamed" environment. Each cycle or year is condensed in time to a three to eight hour session during which the decision makers formulate their yearly policy. The decisions that emerge out of the "competitive-cooperative" environment of the gaming-simulation are used as priming inputs to the computer simulation. The change in the status of the urban area is calculated by the computer and returned to the decision makers as the primary input to the next cycle of action. Included in the change picture generated by the computer are selected social indicators measuring the magnitudes of change in assorted key areas and a newspaper serving as the focal point of local public opinion.

The key decision makers acting in the gamed environment include politicians and planners from a central city and a county, an air pollution control officer from the county, and land developers and industrialists from the private sector. The politicians are responsible for the administration of their respective jurisdictions and for the formulation and implementation of various programs to upgrade the social status of their constituents. The planners serve as aides to the politicians and represent the major long range coordinating force in the community. The air pollution control officer is charged with the task of cleaning and monitoring the air mass above APEX county. The land developers and industrialists have the responsibility of running their particular business concerns within the confines of the county. It is expected that each decision maker will find it to his advantage to coordinate and/or compete with other players in his efforts to promote his strategies. The APEX General Interaction Diagram included here (see page) indicates possible linkages among players and between players and the simulation.

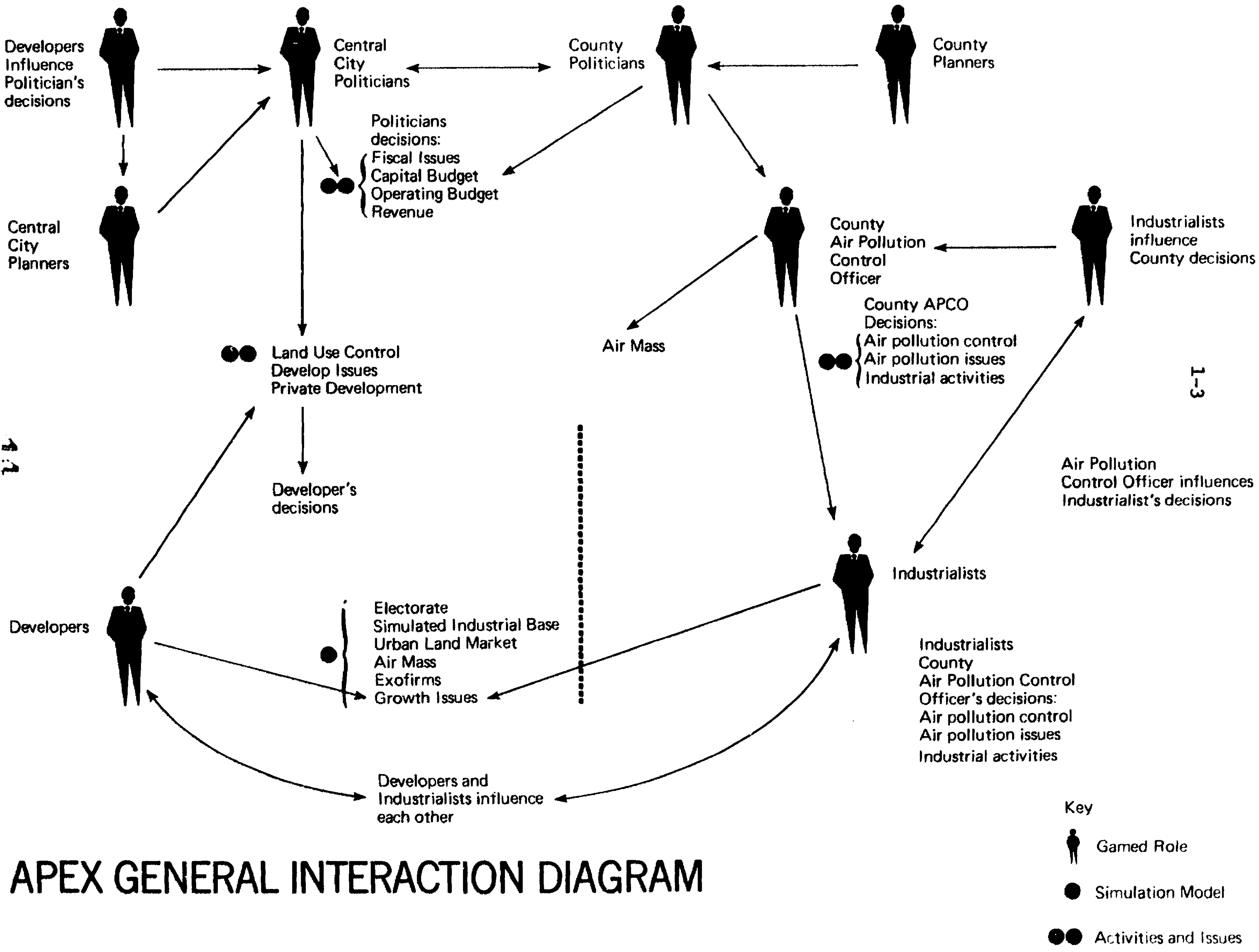
In general, people have great difficulty understanding the dynamics of a complex system through traditional means. Gaming-simulation offers participants the opportunity to study, work with, and discuss the structure of such a system and to experiment with intervention strategies designed to change that structure. When used as a teaching device, the strength of a gaming-simulation such as APEX lies in the opportunity afforded participants for involvemnt in the system. When compared with the passive observation of the system offered by traditional methods, this approach has had great success.

In theory, complex gaming-simulation of the APEX variety is more than a training device or communications facilitator. If the models were more sophisticated the data base more accurate and more complete, a complex gaming-simulation would be a policy testing device for use by practicing urban politicians, planners, APCO's and administrators. Conditional predictions (predictions based on the particular policies and/or decisions submitted to the model) of the ramifications of various decisions can be generated through the use of a complex gaming-simulation -- predictions that may forewarn the model user of unforeseen reactions to policy at several levels of the urban hierarchy ranging from that of the highest level.

The gamed environment is similar to that found in a typical midwestern industrialized town. (In fact, the prototype city is Lansing, Michigan). It has a population approaching 220,000 including several of minority groups sharing racial or ethnic ties. There is a relatively dense central city in the heart of the county, an adjacent suburb and two outlying townships. Most of the industry is located in the central city (as are the minority groups). Major firms include a large auto plant and the state government offices. The suburb houses a major university. The townships are largely agricultural, although urbanizing settlements are dotting the landscapes. There is a major river running through the city serving as the primary drainage system for the county. The climate of APEX is temperate, with summer temperatures averaging about 70 degrees and winter temperatures averaging near 25 degrees. Prevailing winds are westerly, swinging to the southwest in summer and northwest in winter.

For the purposes of the gaming exercise, APEX county is divided into 29 analysis areas (see the attached map, Section 9). Population, employment and land use will be allocated to the areas and are categorized by types established especially for APEX. These types are described in the glossary included in this manual (Section 2) a glossary designed to aid participants in learning the terminology of urban and environmental management as well as that of the gaming exercise.

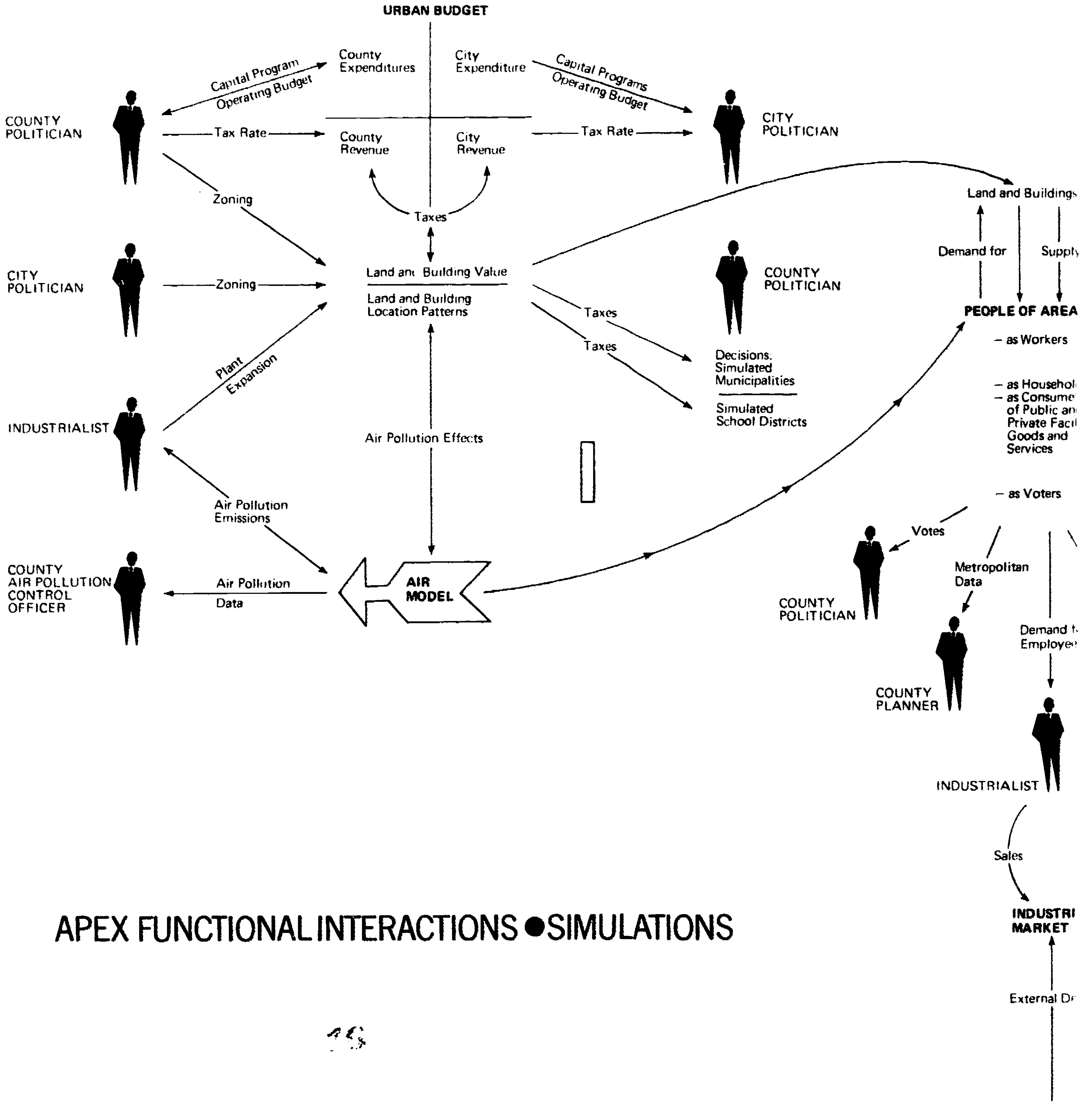
FOR ADDITIONAL INFORMATION:
Address inquiries to Chief, Institute for Air Pollution Training
Environmental Protection Agency
Research Triangle Park, North Carolina 27711



APEX GENERAL INTERACTION DIAGRAM

APEX

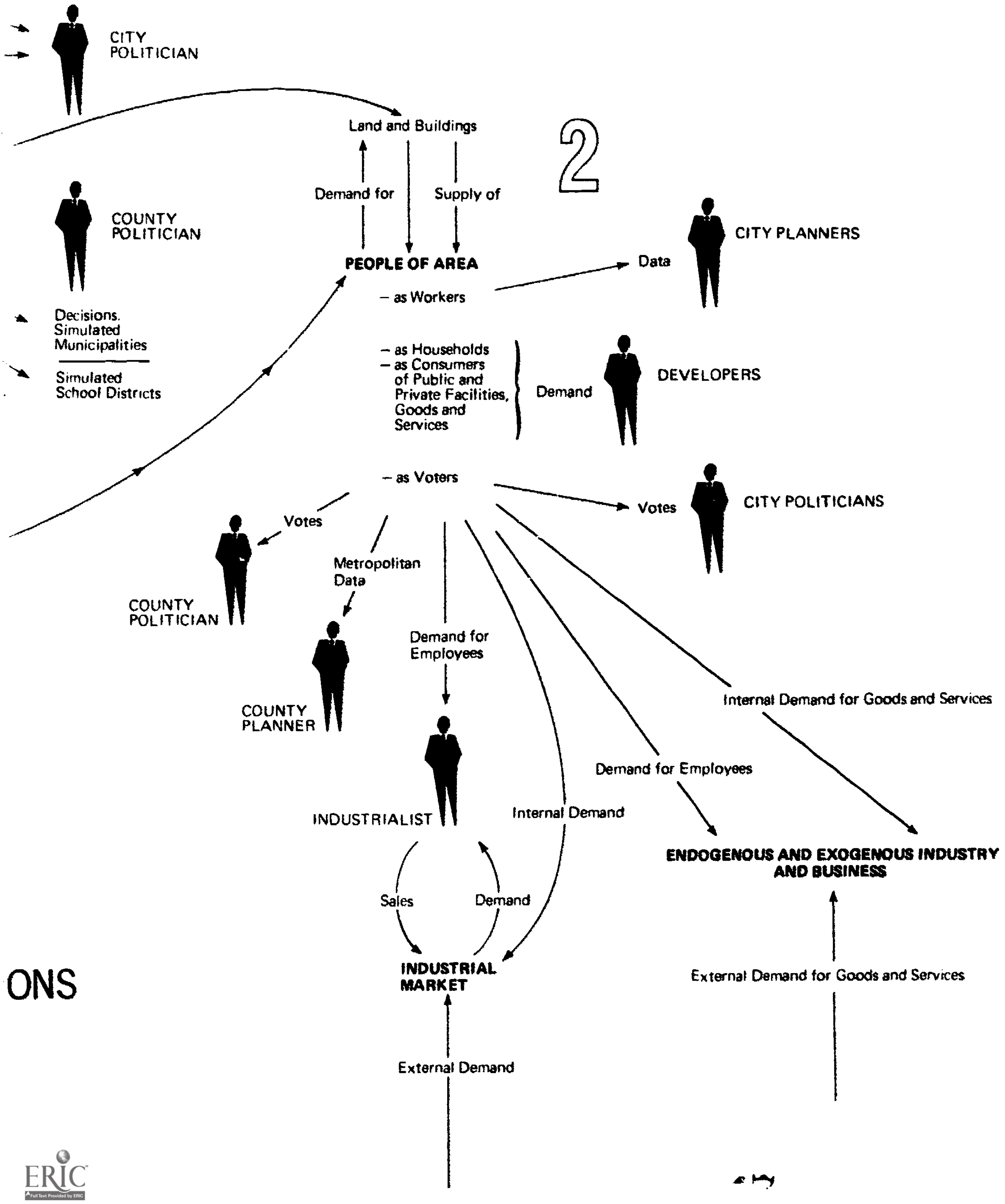
FUNCTIONAL INTERACTIONS ●SIMULATIONS



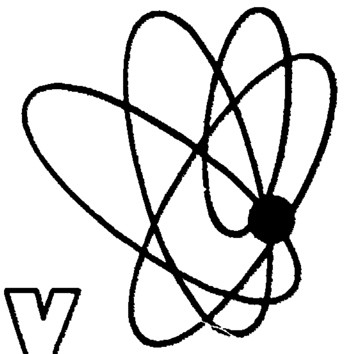
APEX FUNCTIONAL INTERACTIONS • SIMULATIONS

45

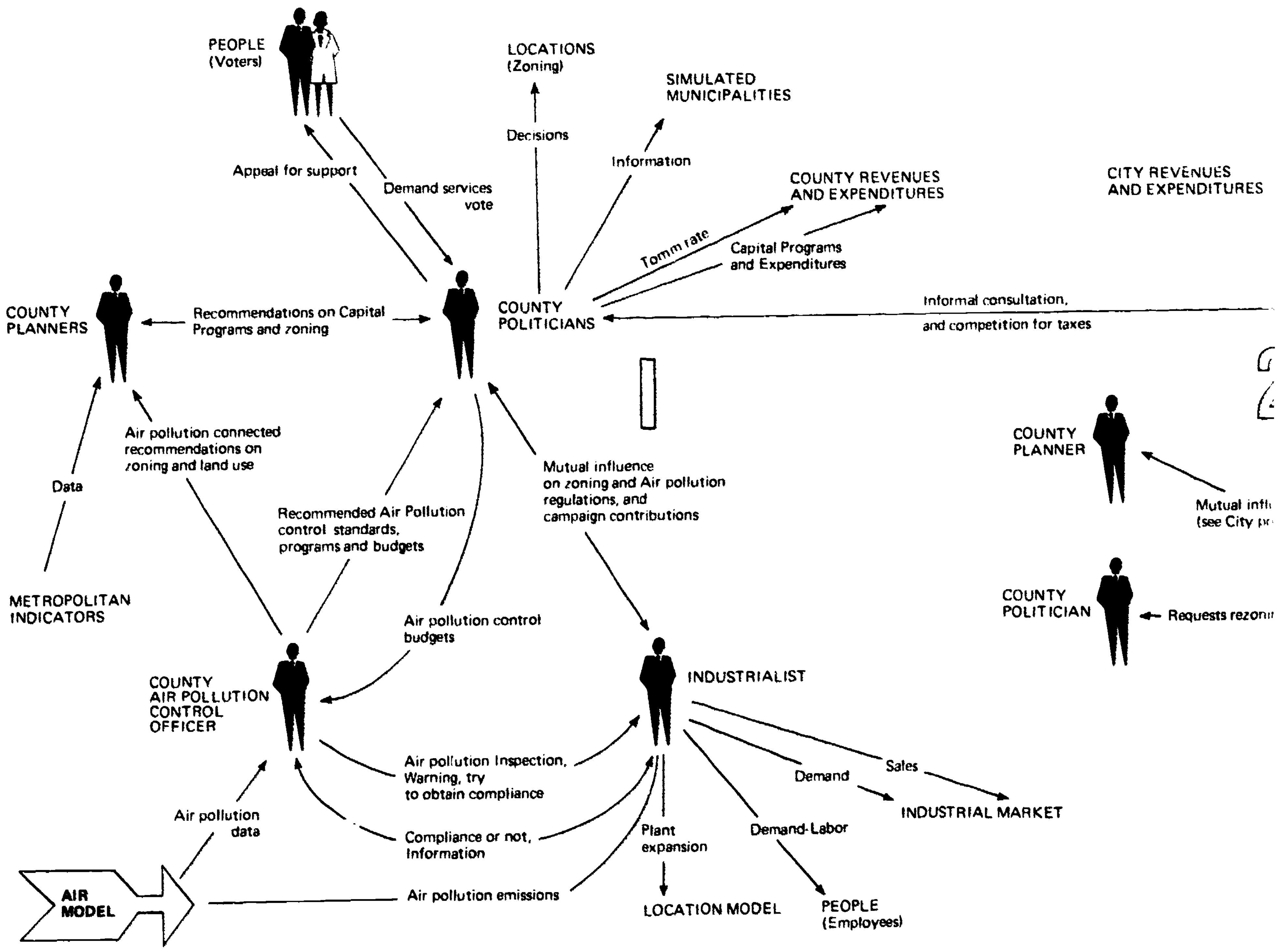
ONS


CITY POLITICIAN
COUNTY POLITICIAN
Decisions. Simulated Municipalities
Simulated School Districts
Land and Buildings
Demand for
Supply of
2
PEOPLE OF AREA
– as Workers
– as Households
– as Consumers of Public and Private Facilities, Goods and Services
Demand
– as Voters
Data
CITY PLANNERS
DEVELOPERS
Votes
CITY POLITICIANS
Votes
COUNTY POLITICIAN
Metropolitan Data
COUNTY PLANNER
Demand for Employees
INDUSTRIALIST
Internal Demand
Demand for Employees
Internal Demand for Goods and Services
ENDOGENOUS AND EXOGENOUS INDUSTRY AND BUSINESS
Sales
Demand
INDUSTRIAL MARKET
External Demand
External Demand for Goods and Services

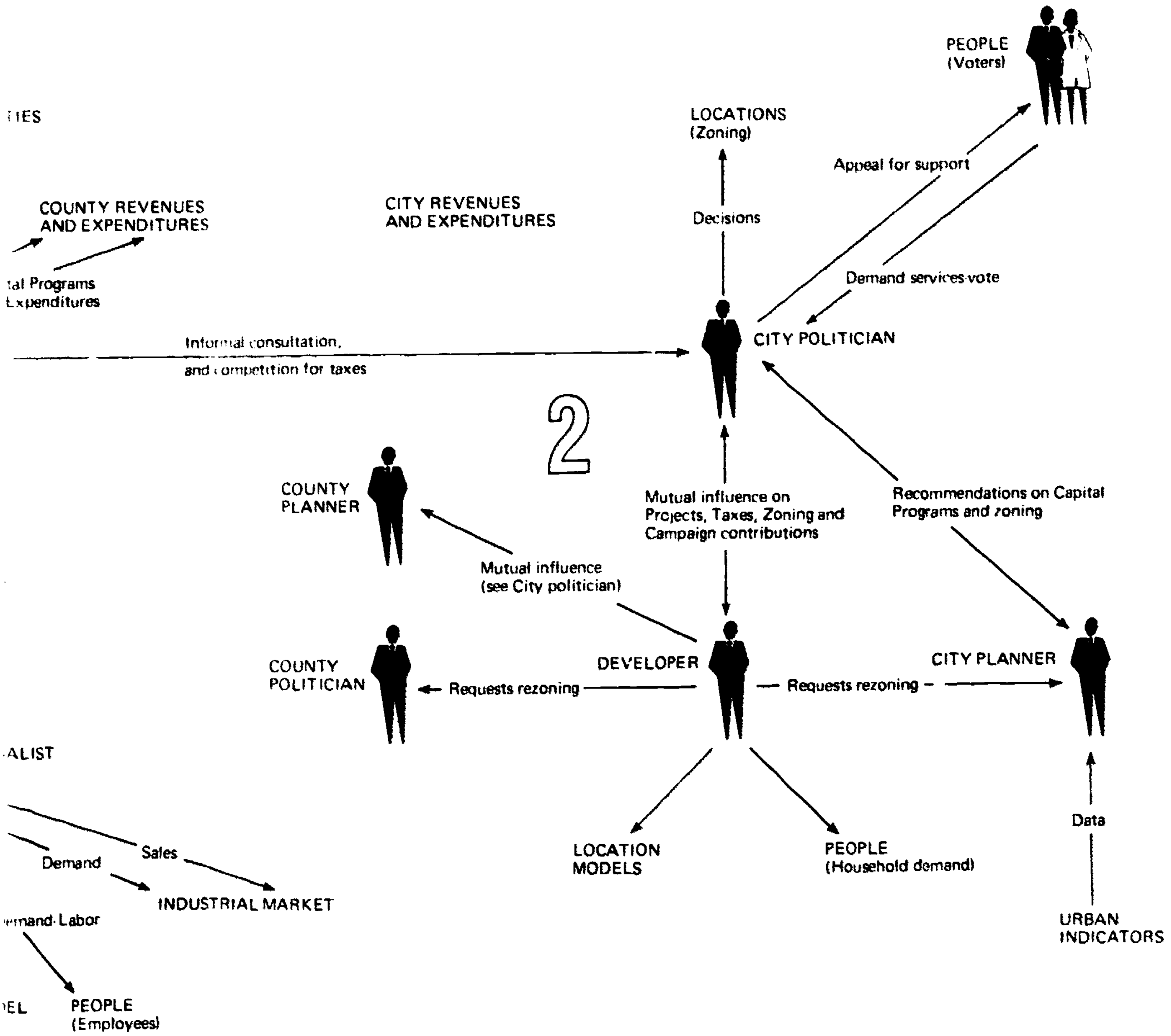
APEX



FUNCTIONAL INTERACTIONS ● ROLES



APEX FUNCTIONAL



APEX FUNCTIONAL INTERACTIONS●ROLES

SECTION B

PREFACE

SECTION B-1

Preface

This volume of the operator's kit contains the instructions and technical support material needed for processing the APEX decisions on an IBM 1130 computer. The APEX program utilizes the many "hands on" features of the IBM 1130 computer to enable the operator to interact freely with the system. Even though the system has this interactive capability, no special skills in operating the IBM 1130 are required to run the program. At the risk of boring persons with experience in using the 1130, this manual is written under the assumption that the reader is totally unfamiliar with that or any other computer--it contains all the instructions necessary to get the program running on the proper equipment.

This volume takes the computer operator from the worksheets filled out by players of the game through the computer processing procedure. The key punching operation is described first. This is followed by a description of the interactive language of the APEX system (the messages from the program and the appropriate response options are described in depth) and assorted utility and support features are described. The structure of the data as it is stored in the computer is then laid out and, finally, crude flow diagrams of the program are presented.

For the completely uninitiated, a brief description of an IBM 1130 computer is included in this introduction to set the terminology for the ensuing chapters. The heart of any computer is called the Central Processing Unit (CPU). This component is made up of magnetic on-off storage elements called bits that collectively comprise the core, and the hardware that actually performs the calculations. The bits are generally thought of in terms of groupings of 16 -- each group is called a word and is the basic storage unit for an integer number. Supporting the CPU is an additional data storage device called a disk. The disk looks like a phonograph record encased in a plastic protective cover. That record is made up of additional data in the form of groupings of magnetic bits (words), but these are reached more slowly by the machine than data stored in core. The slower disk is also much larger than the core: 500,000 words as compared to 8,000.

Peripheral equipment includes a card read/punch unit (input) and a line printer (output). The basic configuration of the IBM 1130 computer includes a fairly slow line printer (an 1132) that prints at about 80 lines per minute and an equally slow card read/punch (a 1442). Although the program is written for this basic configuration, installations with larger peripherals (or even a larger core) can make fairly simple modifications to accommodate the faster equipment. These modifications are not described in this manual. In addition, the 1130 system has a console typewriter and keyboard. All communications with the system except for the basic input streams go through the console typewriter or the switches on the face of that console (the console entry switches).

SECTION 2

FROM WORKSHEETS TO CARDS

Section 2-1

From Worksheets to Cards

The first step in the computer processing is the preparation of the input cards. There are four sets of input cards. The bulk of the player decisions are read in during the early phases of the program. Three smaller sets of input cards are read in later in the program for processing the newspaper and the elections. This chapter will take up the preparation of the main input first and cover the supplementary input streams later. A skilled operator can prepare the supplementary input cards while the program is running; however, the main input stream must be ready at the start.

Prior to the actual punching of the main input cards, the operating team must transfer and translate the decisions on the player's worksheets to coding forms. The player decisions are sometimes expressed verbally rather than in number code and appear, along with assorted extraneous entries, on the worksheet. To facilitate keypunch operations, it is advisable for the operating team to translate the verbal decisions to numeric code and enter that code in the proper location on a key punch form.[1]

In certain instances, players may find the format of the worksheets time consuming, unproductive and restrictive. In those cases, the keypunch forms themselves can be used by the role advisors to direct the players through their decisions. In the extreme case, the players can go directly to filling in the keypunch form bypassing both worksheet and role advisor.

Copies of the keypunch forms for each role are included in this chapter for reference. Additional copies of the consumable supplies, (worksheet and forms) may be reproduced as needed. The starred items on the worksheet have to be entered on the keypunch form. The coding for those items entered in verbal rather than numerical form is explained on the page following the respective coding forms.

An example of the translation process will best demonstrate the procedure for filling in the coding form (see example 1). After a team of role advisors gains experience in filling in key punch forms, the expected time duration for the transfer and translation process is about five minutes per role.

Once the decisions are entered correctly on the keypunch coding forms, the input cards must be punched. It is assumed that the reader is capable of using the key punch -- if not, a short introduction should be all that he needs. The key punch is little more than a high powered typewriter - one designed for putting holes in IBM cards rather than for printing letters on paper.

Data is entered on cards to conform with an arrangement in accordance with program specification. Two distinct arrangements, called formats, are used. The first and most common allows up to thirteen entries on a

[1] A complete set of coding forms and instructions is available.

standard 80 column card. (Each grouping of columns, or entry, is called a field). The first entry (field) takes up two columns, the next six fields take up four apiece for a total of twenty-four, and the last six fields take up nine each, filling the remaining fifty-four. The first two columns are reserved for an alphabetic code of two letters. Each card must have an alphabetic code. The next six fields are reserved for integer entries. These must be right justified; that is, the entries must appear as far to the right in each field as possible. For example, the number 12 would be entered in columns 5 and 6 on a card, rather than in columns 4 and 5, if it belongs in integer field 1. The remaining six fields are reserved for real number entries (numbers with decimal points). The entries may appear anywhere in the field but a decimal point must be punched in the appropriate location. Each card corresponds to one line on the key punch coding form. The two letter code adjacent to each table is the two letter code in the first field. The designation (in parentheses) "I-*" indicates that the entry belongs in the "*th" interger field. Similarly, the designation "F-*" indicates that the entry belongs in the "*th" real number field. (For FORTRAN buffs the input format is (A2, 6I4, 6F9.0) with integers right justified.)

The second format is used only twice, for PI and EM cards. The two letter code is handled exactly as it is above. However, the remainder of the card is partitioned into 15 four column integer fields; that is, all entries are integers and all fields are four column wide. Each entry goes in the field indicated by the I-* designation on the key punch form and each is right justified. (For FORTRAN buffs the input format is (A2, 15I4) with integers right justified).

The key punch procedure is greatly simplified by the proper use of a drum card. A drum card is a high powered tab stop for a keypunch -- it not only jumps to the right column; it sets the numeric or alphabetic (upper or lower) case as well. Thus, by using a drum card, numbers are punched in all fields but the first. Note: because the drum card sets the key punch for numeric case, use only the numeric decimal point. Also, use caution when using the skip key (the analogue to the tab). When the machine is resting in the first column of a field, depressing the skip key advances the field; i.e., the key punch shifts to the next field.

A drum card for APEX is made by placing a 1 in column one of an IBM card, followed by an A. The first column of each remaining field is blank with ampersand(s) filling the balance. (See Illustration 1).

Once all the cards are complete, an additional card with the letters XX punched in the first two columns should be added to the deck as the last card. The XX is the signal to the computer, during processing, that the last main stream input card has been read.

The supplementary input cards (supplementary in that they are needed to round out the APEX input) must also be prepared in accordance with program specifications. Each of three sets of cards must be punched: issue decision cards, candidate election cards and special headline cards. The issue decision card set is required in each cycle except (Cont. on page 2-6)

EXAMPLE 1: Worksheet

E. Exofirm Investments

	Exofirm No.*	Preferred Location	Amount*
1.	4	23	50000
2.			
3.			
4.			
5.			

Total Exofirm Investments $______________

F. Land Purchases

	Seller*	AA*	Vac/ Dev.*	Zoning Cat.*	No. of Units*	or	No. of Acres*	Price/Unit Price/Acre*	Total Price
1.	MKT	21	Vac	Res/sgl			100	15/100	151000
2.									
3.									
4.									
5.									
6.									
7.									
8.									
9.									
10.									

Total Land Purchases $______________

G. Zoning Application Fees

	AA	Vac/ Dev.	From?	To?	No. of Units	or	No. of Acres	Fee at $100 per application
1.								
2.								
3.								
4.								
5.								
6.								

Total Zoning Application Fees $______________

(Continued on the next page)

Example 1 continued.

H. Land Clearance Costs (Demolition)

	AA	Total Value of Developed Property being Rezoned	Demo. Rate	Land Clearance Cost
1.			.05	
2.			.05	
3.			.05	
4.			.05	
5.			.05	
6.			.05	

Total Land Clearance Costs $__________________

Example 1. Worksheet and Keypunch Form for a Developer.

Developer Number 7

Cycle Number 1

1. Loan Payment (LP)

Dev. No. (I-1)	Loan No. (I-2)	Amount (F-1)

2. Tax Payment (TP)

Dev. No. (I-1)	Juris. (I-2)	Amount (F-1)

3. Campaign Contribution (CC)

Dev. No. (I-1)	Pol/Bond/Mill No. (I-2)	Amount (F-1)

4. Cash Transfer (CT)

From (I-1)	To (I-2)	Amount (F-1)

5. Exofirm Investment (BI)

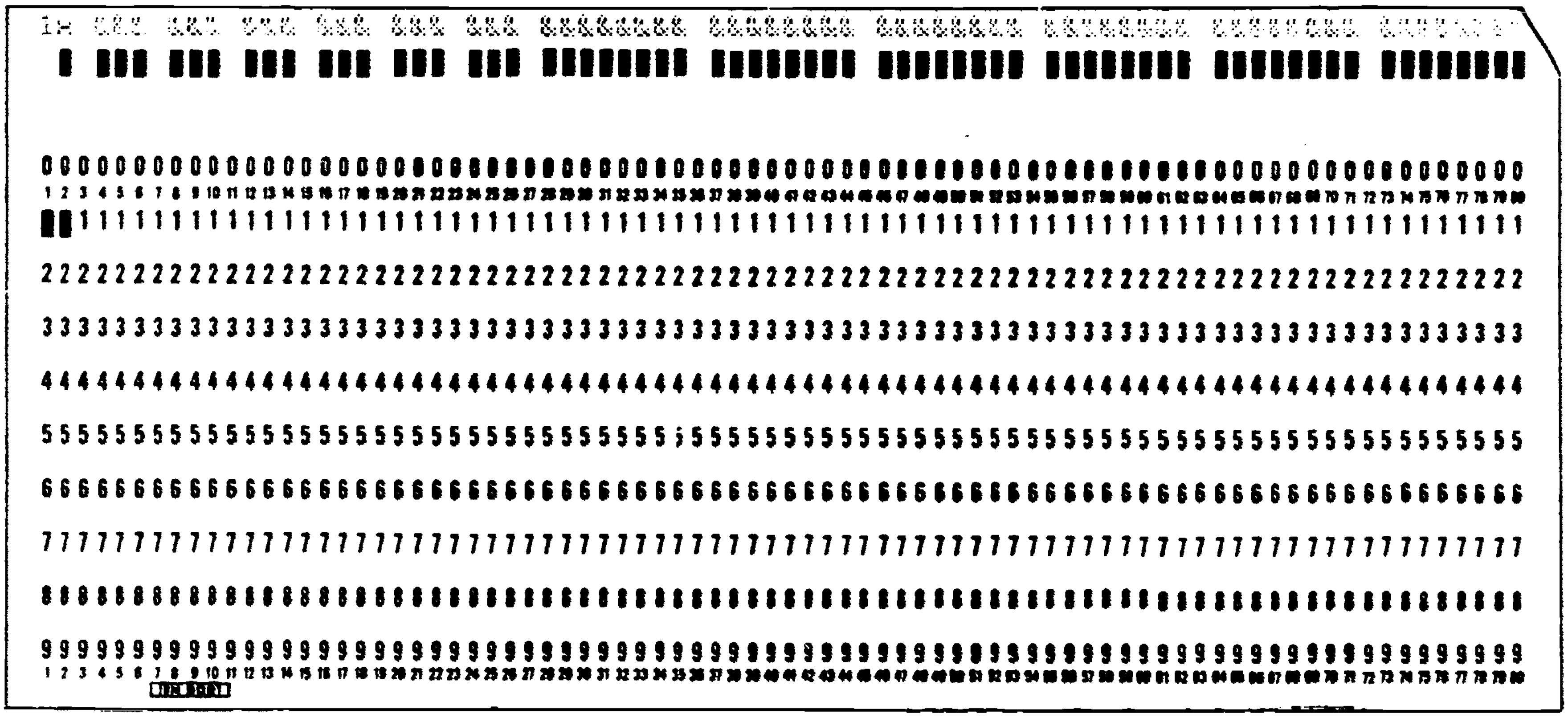
Dev. No. (I-1)	Exo.No. (I-2)	Amount (F-1)	Location
7	4	50000	23

Example 1 continued.

6. Land Transfer (LT)

Seller (I-1)	AA (I-2)	Vac/ Dev. (I-3)	Zone Cat. (I-4)	Buyer (I-5)	No. of Units (I-6)	or No. of Acres (F-1)	Price/Acre Price/Unit (F-2)
15	21	0	1	7		10.0	15100

ILLUSTRATION 1: Drum Control Card

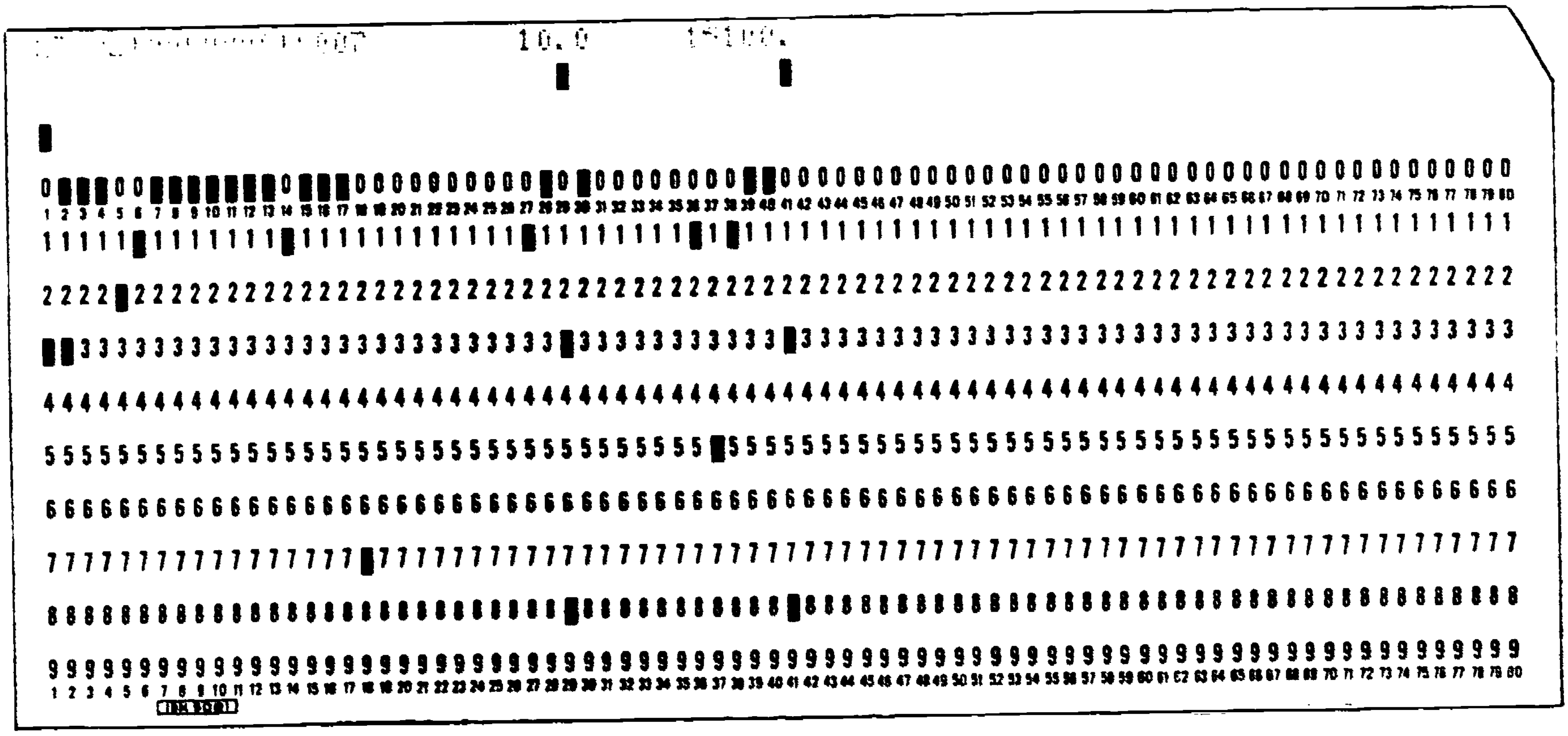


Diagram

For easy reference, the following is a chart of where each field begins on the card.

Field	begins in Column	Field	begins in Column
I-1	3	F-1	27
I-2	7	F-2	36
I-3	11	F-3	45
I-4	15	F-4	54
I-5	19	F-5	63
I-6	23	F-6	72

Example 2 is the punched card for the Land Transfer example given previously.



Example 2: Punched Card for a Land Transfer.

cycle 1 - the pre-run set of decisions and output used to start the game. The candidate election cards are needed only in those cycles designated as election years(usually every odd cycle except the first pre-run cycle). The special headline cards are optional.

An issue decision card contains one or more groups of numbers separated by commas. Each group of numbers gives the program information about one issue appearing in the newspaper that was printed at the end of the preceding cycle. The information is comprised of two required entries; the issue number and the alternative acted upon by the deciding agent (a politician or the elite opinion poll). If the alternative action required that a project be funded, the Bond or Millage number associated with that project <u>may</u> be given.

The format for issue decisions is as follows: iiia(bbb), iiia, iiia, . . . iiia*

Where: iii is a 1 to 3 digit issue number,
a is a single digit indicating the alternative chosen on that issue,
(bbb) is an optional 3 digit bond/mill number

The card is "free-format"; that is, punching may begin in any column, and blank columns are ignored and may be used for spacing to increase legibility. Each group of numbers for a single issue is separated by a comma. However, the last group must be terminated by an asterisk(*), not a comma. A maximum of 40 issue decisions may be input in any single cycle. If one card is not sufficient for all issues, decisions may be punched on any number of additional cards. The terminating asterisk appears only after the last decision on the last card of the series. The last character on any one card must be either a comma or an asterisk. Thus, a number group for any one issue must appear on only one card -- it will not be continued from one card to the next.

The following is an example of two issue decision cards:

```
1692,11,
            561(103),283*
```

This would be interpreted as follows:

Alternative 2 for issue 169, alternative 1 for issue 1, alternative 1 for issue 56 subject to the passage of bond 103 (otherwise, if Bond 103 failed, the alternative would be changed to 2 by the program), and alternative 3 for issue 28. Note that all of the above could have been punched on one card; the two cards were used only for illustration. A key punch for issue decisions (see next page) is included in the packet of forms for use by the operator.

In each cycle, the alternatives selected for every issue appearing in the preceding cycle's newspaper must be entered as a part of the issue decision card set. Use of the elite opinion poll wall chart tally sheet described in Volume 1, Chapter 3 of this kit is extremely useful in keeping tabs on issues and issue outcomes.

There are up to 16 candidate election data cards in an election year. A card must be punched for each incumbent politician running for office. (Up to eight incumbents -- three central city politicians, one county commissioner representing each outlying jurisdiction and two county commissioners representing the city -- can run each cycle.) For each incumbent running for office, a card must be punched for either his simulated or gamed opponent.

The following format is used for punching candidate election data cards.

Cols.

1-2 Candidate Number. A candidate number must be entered on each card. Incumbents should be assigned numbers 1-8. Corresponding numbers from 9-16 should be assigned to their opponents so that each matched pair are given numbers 8 apart; that is, opponent 12 runs against incumbent 4, etc. The assignment of numbers to incumbents is arbitary. Do not punch cards for opponents if there is not a card for the corresponding incumbent.

3 a) For incumbents (01-08 in columns 1-2) this column indicates whether the opponent is real or simulated:
1 = gamed opponent (real)
2 = simulated opponent

b) For gamed challengers (09-16 in columns 1-2) this column should be blank. For simulated challengers this column designates the opponent type:
1 = Conservative
2 = Moderate Republican
3 = Labor Liberal
4 = New Left Liberal
5 = Working Class Conserative
6 = Ultra-Conservative
7 = Business Candidate
8 = Black Candidate
9 = Reform Candidate

4-8 Party Support. This entry is included only for incumbents. It is the major discretionary variable under the control of the operator. Some number between 1 and 60 is entered designating the percentage of the vote, up to 60%, going automatically to the incumbent. The entry can be determined by apportioning the 60% in accordance to a hand vote of game players, operator's whim or any other suitable method. If the entry is left blank, the machine sets the appropriate variable to 30.

9-28 Candidate Name. A name up to twenty characters long is entered here to be printed with the election results in the newspaper. Names should be entered for simulated as well as gamed opponents. For example, a conservative might be called Richmond Q. Stuffshirt.

29-80 Issue Number, Candidate Stand on Issue, and Candidate Stress on Issue. The form follows that for issue decision cards -- iiia(bb) -- where iii is a 1 to 3 digit issue number, a is the alternative chosen and bb is the percent of campaign funds spent stressing the stand taken on that issue. These entries are free format and up to 40 issue decisions may be entered. Additional cards may be used as needed with the continuation card starting in column 1. An asterisk is used to designate the end of the sequence for a given candidate. Entries should be made for only those issues for which the candidate casts a vote. Note that these columns should be left blank for simulated opponents. Also, for unstressed issues (issues which would have zero in parentheses) the parentheses may be omitted.

At least one blank card must follow the last candidate card to terminate this section of the input. The following is an example of a card for an incumbent candidate:

```
01125.00ROY MILLER    1691(20) ,0012(5) ,0563,0423*
```

Incumbent 1, Roy Miller, has a real opponent. Twenty-five percent of the party allocation has been assigned to him. He has voted for alternative 1 on issue 169 and spent 20 percent of his campaign funds stressing that stand, etc.

The final input section contains special headlines written by the operator, a newspaperman or players in the game for publication in the News. These cards, like the others, must be punched to conform to program specifications. Each line of a headline requires two cards. The first column of the first card must have a code describing the type of headline. The possible codes are:

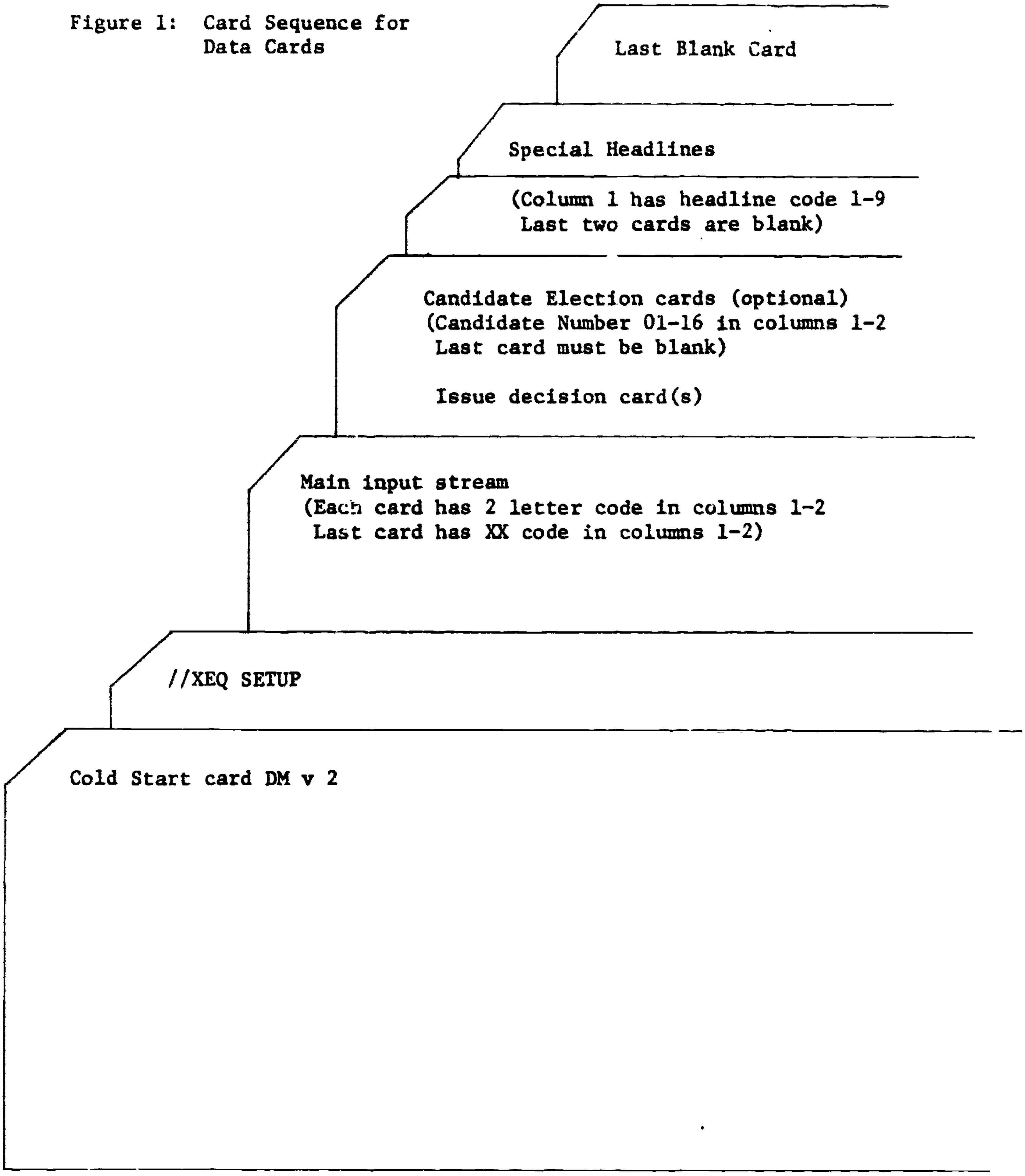
1 = Special Headline (lead article)
2 = National Headline
3 = State Headline
4 = County Headline
5 = Central City Headline
6 = Suburb Headline
7 = Township 1 Headline
8 = Township 2 Headline
9 = Business Headline
0 = Stop (last card in input)

The rest of the first card, columns 2-80 contain the first 79 characters of the line. The first 41 characters of the second card contain the last 41 characters of the same newspaper line. There can be a <u>maximum of 118 lines</u> in a given cycle.

The cards must be ordered so that the codes in column 1 are in ascending order from 1-9 followed by the mandatory headline with the 0 code (2 blank cards). To skip a line between headlines in a single section (with the same code in column 1), one of the 118 lines must be left blank; that is, the appropriate code should be entered in column 1 of a card but the remainder of the card and a second card should be left blank. The program automatically handles the placement of each section; therefore, blank cards are not needed between sections (between headline cards with different codes in column 1).

In summary, the data deck required to run a cycle from start to finish is pictured in figure 1. It is not necessary, as noted in the next chapter, to have all the cards prearranged in the sequence of figure 1 -- each stream can be submitted to the card reader at the time the program becomes ready to read them. If the latter procedure is followed (submission of sections when called for) each section should be followed by a blank card to ensure that the card reader will read the last card of the input section. Also, the reader should be cleared (NPRO button pushed and held for two seconds) prior to the submission of the next section.

Figure 1: Card Sequence for Data Cards



Last Blank Card

Special Headlines

(Column 1 has headline code 1-9
Last two cards are blank)

Candidate Election cards (optional)
(Candidate Number 01-16 in columns 1-2
Last card must be blank)

Issue decision card(s)

Main input stream
(Each card has 2 letter code in columns 1-2
Last card has XX code in columns 1-2)

//XEQ SETUP

Cold Start card DM v 2

SECTION 3

RUNNING THE PROGRAM

Section 3-1

Running the Program

The APEX computer program and data bank are stored on two IBM 1130 disk packs. The first disk contains the bulk of the program and the data. The second disk has the large data files needed to support the newspaper and several programs needed to print the newspaper or to perform calculations requiring those data files. Each disk 1 has sufficient data file space for only one run at a time. Thus each should be labeled with a team number, a number that is also recorded in a data file on the disk. Each disk 2 can accomodate up to six runs at a time because the team number is transferred from disk 1 to disk 2. Thus, if two games are run simultaneously, two disk 1s are needed, but only one disk 2 is required. The standard IBM 1130 can accomodate only one disk at a time; therefore, the process of running the program includes a change of disks at the appropriate time. That procedure is described later; for the moment, it is only necessary to appreciate that there are two separate and distinct disks.

To turn on the IBM 1130, the following steps must be followed.

1. Turn on the main power switch on the 1130 console (upper right corner).

2. Insert Disk 1 (Pull down the blue handle behind the right side cabinet d-or, insert the disk with the IBM label facing the front, close the blue handle).

3. Turn on the disk file switch in the cabinet.

4. Ready the IBM 1132 printer by pushing the green START button on the printer. (We assume that paper is in the printer. This manual does not describe the procedures for putting paper in the printer -- consult the IBM 1130 Disk Monitor System Manual usually found quite near the computer.)

5. Wait for the FILE READY light on the console to come on.

6. Set all console entry switches down.

7. Push the NPRO (non-process run out) button the 1442 reader for about two seconds to be sure the reader is clear of cards.

8. Insert a cold start (Disk Monitor System, Version 2) followed by a // XEQ SETUP card, the data cards and one blank card. (The cold start card is needed to activate the system by transferring the control system from the disk to the core, the // XEQ SETUP card loads the first part of APEX into the core, the blank card ensures that the final card will be read and not left in the read position in the reader. The // XEQ SETUP card has slashes in columns 1 and 2, XEQ in columns 4,5,6 and SETUP in columns 8-12).

9. Press the START button on the 1442 reader -- the ready light should come on.

10. Press the RESET and PROGRAM LOAD buttons on the console.

After following this procedure the program will begin executing. From this point on, the APEX system is in complete control of the computer. The program will "speak" to the operator via written messages on the console in a form of broken English called Apexese. The balance of this chapter will describe this new language.

Messages are transmitted to the operator at appropriate points during the execution of the program. The following accounting of machine messages shows all comments made on the console in capital letters. Operator responses are also in capitals but are underlined. Bracketed portions of messages are parts of statements that vary from cycle to cycle.

There are two types of operator response: 1) setting the console entry switches (pushing them up) and 2) entry of data through the console typewriter. The program has been designed so that the normal processing sequence does not require the setting of any of the console entry switches. Only three pieces of data must be entered, the date, the numbers of playing developers, the numbers of playing industrialists. All other requests may be bypassed.

There is one helpful rule to remember while processing. The computer pauses (the console lights stop flashing) only when some action is needed from the operator. When the KB SELECT light on the console is on following a typewriter pause, operator input from the keyboard is sought and the EOF key must be pressed to restart the program. For pauses in which the KB SELECT light is not on, the operator must press the green program START button on the console to continue the program. In the former case, operator intervention is being requested from the keyboard; in the latter, intervention comes through the console entry switches.

Once the program begins execution the following messages will appear:

```
***START OF APEX DISK 1 FOR TEAM [(n)]
   PRESENT IDENT IS ['SHORT COURSE -- SECOND WEEK']

ENTER NEW DATE -- MMDDYY
010270
```

#1 In case more than one game is being run simultaneously, the second line of this message allows the operator to make sure that the correct disk #1 is being used.

Usually, the date entered is the date of the next cycle.

The format required for entering the date is a two-digit month

(MM) followed by a two-digit day (DD) followed by a two-digit year (YY) with no breaks between them. Leading zeroes must be entered before a day or month having only one digit. NEVER SPACE OR USE COMMAS. Mistakes in typing can be corrected by pressing the ERASE FIELD key and entering the data anew. Press EOF to continue.

CES 10 -- TO PRINT APEX FILE CONTENTS - (FILE 6 FOR PROJECT LISTS)

#2 Any messages throughout the processing which refer to CES, SW, or SWITCH refer to the console entry switches. To print the contents of any file, set switch 10 up. If this is done, the following messages will be printed; otherwise, the program skips to message (7).

NUMERIC MODE ***

ENTER FILE NUMBER (01 - 79 OR EOF TO END LOOP)
06

#3 The most common file printed is file 6, which is the list of capital improvement projects and special programs available to the Planners and Politicians. Again, leading zeroes must be typed. Not all files between 1 and 79 are "legal" files. If the operator tries to dump a non-existent file, a message will be typed indicating this, followed by the ENTER FILE NUMBER message. When all needed files have been dumped, press the blue EOF key to end the loop. If file 6 is to be printed, the following message appears.

CES 4 -- TO SORT BY BUDGET TYPE AND SCALE
DOWN -- BY NUMERIC ORDER

#4 By setting switch 4 up at this point, the project/special program list will be printed both by type and scale. Currently, projects are numbered in BUDGET TYPE groupings, and within these by SCALE, so this option is probably not necessary.

TYPE NUMBER OF COPIES OF PROJECT LIST DESIRED
04

#5 Type in a number from 1 to 99. After the number of copies desired are printed, message (3) is repeated. After the EOF key is pressed, the following message is printed.

CES 10 -- TO ENTER VALUES INTO VARIABLE FILES

#6 Normally, operators will not enter values into the files, therefore, this option will not be used. Make sure switch 10 is down and press PROGRAM START to continue. If the operator chooses to fool with the raw data, he should read the next chapter entitled

"Other Options" to learn how.

```
IF SETUP ENTRY ONLY TO CHECK FILE CONTENTS, PUSH INTO REQ KEY ON
KEYBOARD TO EXIT FROM PROGRAM
```

#7 The program can be stopped at this point by pressing the Interrupt Request key on the console.

```
    IF THIS IS A RE-ENTRY TO SETUP (AFTER A CES 15 EXIT),
CES 15 -- TO SKIP PAST CYCLE-PRESERVATION
```

#8 This message refers to the switch 15 exit described below. This option is used only when files have already been preserved (see messages 10, 11 and 12) and the program stopped after message 12 and restarted some time later.

```
    IF THIS IS A RE-ENTRY TO SETUP (AFTER A DATA CARD FOO* ERROR)
CES 13 -- TO RESTORE HIGH COMMON AND SKIP TO INPUT SECTION
```

#9 One type of error in reading the input cards causes the program to stop. This error, an FOO*, is described in the appendix on error messages. After encountering an FOO* during input, subsequent steps can be bypassed and control passed directly to the input part of the program by setting switch 13 up at this point.

```
CES 11 -- TO RESTART FROM CYCLE 0 HISTORY
```

#10 Every time a new group is beginning a run, the initial status of the city must be restored. Set switch 11 up at this point to prepare the disk for processing the cycle 1 data deck, the standard deck that generates the first set of output given to the players. In all subsequent cycles for the same group of players, ignore this instruction. If the switch is not set, message 12 will appear next.

```
CES 12 -- TO RESTART LAST CYCLE (WITH HISTORY OF CYCLE BEFORE LAST)
  DOWN -- START A NEW CYCLE WITH HISTORY OF T-1 (OVERWRITING T-2)
```

#11 If an error was made in the most recent cycle processed and the operator wishes to re-run the cycle to correct the error, switch 12 would be set up at this point. Otherwise, ignore the instruction.

```
       AFTER PRESERVING FILE-HISTORY FROM CYCLE [(n)]
CES 15 -- TO TERMINATE PROGRAM
```

#12 This is a point at which the program can be stopped. Upon restarting a small quantity of time can be saved by exercising the switch option at message (8). Check to see that the correct cycle was saved.

```
CES 14 -- TO SUPPRESS ALL PRINT THIS CYCLE
```

#13 In the event that a mistake or change in strategy from an early cycle dictates a quick rerun of several cycles and there is no reason to generate intermediate printout all print can be suppressed by setting switch 14 at this point.

```
    GOING ON TO CYCLE [(n)] OF PLAY
PRESENT TEAM IDENT IS ['LAW SCHOOL/SNR/ARCHITECTURE']
TO CHANGE TURN ON SWITCH 8
```

Check that you are going on to the correct cycle.

#14 In processing cycle 1, the team identification is erased and a new identification may be entered. It is not mandatory that there be a team identification. If the operator wishes to specify a team ident, he may set switch 8 at this point. If switch 8 is set the following appears.

```
ENTER NEW TEAM IDENT -- UP TO 40 CHARACTERS
TEST APEX
```

#15 An alphabetic identification of up to 40 characters can be typed in at this point. Press EOF when the identification is entered.

```
ENTER NUMBERS OF PLAYING DEVELOPERS, 1...7 (7I1)
1234567
```

#16 The numbers of the playing developers are typed in at this point. If some Developers are not being played, their numbers may be omitted (e.g. 1257). In cycle 1, the initialization cycle, all Developers should be entered as playing. There is no punctuation and no spacing between Developer numbers.

```
ENTER NUMBERS OF PLAYING INDUSTRIALISTS, 1...7 (7I1)
12356
```

#17 There are currently only five Industrialists (4 and 7 do not exist). This entry is done like to one above for the Developers. The same rules apply.

```
+ START OF READA INPUT
  CARDS READY  ...
```

#18 At this point the data cards for the cycle should be in the reader and the reader readied. BS cards must be read before any CP or CC cards carrying a bond or mill number. The order of the other cards is insignificant.

```
+++ READA INPUT TERMINATED AFTER AN XX

CES 13 -- TO READ ADDITIONS TO THE INPUT
          [OR CORRECTIONS MADE UPON REJECTS IN STACKER 2]
```

At this point, corrections should be made on all incorrect cards. These cards will be rejected from the input stream and shifted to stacker 2. If no cards were rejected, the bracketed portion of the message would not be typed out. The listing of the cards on the printer gives diagnostic messages for those cards rejected. The most common messages are described in the appendix dealing with error diagnostics. Also, certain errors will cause the reader to stop midstream. These are also described in that appendix. #19

After correcting the cards, clear the card reader (press NPRO). Set switch 13 up. Corrected cards followed by an XX card and a blank are placed in the reader. The reader is readied and the program start button is pressed.

+++ READA INPUT TERMINATED AFTER AN XX

#20

CES 13 -- TO READ ADDITIONS TO THE INPUT

#21 This message appears any time the card read phase is completed and there are no rejects in the second stacker. The same procedure described above for CORRECTIONS is used here for ADDITIONS.

CES 13 -- TO COMPLETELY RESTART READA

Occasionally, the operator may choose to reread the input cards -- some decisions may have been entered incorrectly.(Most often, the operator discovers that a decimal is misplaced or omitted and, to correct the error, he must reread the entire deck. A useful practice for the operator is to scan the listing of the input cards on the printer to check for such errors -- errors that are not caught by the computer.) To completely reread the cards, set switch 13 and push PROGRAM START.

ENTERING TAXLN

#23 This message requires no response -- it is just a signal that input is complete.

SET SWITCH 10 ON IF YOU WISH TO FORCE AN EXOFIRM INTO A SPECIFIC AA ENTER
NEW US EMPLOYMENT GROWTH RATE (F5.0). WILL BE [2.50] PERCENT UNLESS CHANGED

To override the programmed entry of exofirms, the operator may set switch 10 at this point. This allows the operator to specify the analysis area of entry for any or all exofirms. Thus, he can force an exofirm into an area other than those suggested in the newspaper or into the second and third preference as suggested by the newspaper. If switch 10 is set, message (27) will be printed in either of two cases, if the exofirm enters naturally or is forced in through message (25) or (26).

#24 By typing a real number (with decimal), the operator can change the employment growth rate. For example, three percent can be entered by typing 3.0.

Because there is one pause here to both set a switch and enter something on the typewriter, the operator must press EOF to continue processing.

```
POLITICIANS FAIL TO SATISFY CONDITIONS FOR LXOFIRM 3 HAMBURG MEAT PRODUCTS
INC. WANTS TO LOCATE IN AA'S   8   0   0   0
TYPE AA YOU WANT TO FORCE IT INTO (I2) OR LOF TO BYPASS
```

#25 For each exofirm, there are a set of conditions to be met by the Politicians prior to the entry of the firm (these are printed in the newspaper handed out at the end of the last cycle). If the conditions are not met, the above message is printed. By typing in a two digit analysis area (leading zeroes included where needed) and pressing EOF, the operator can force the exofirm to enter that area. By just pressing EOF, the exofirm will not enter the city.

If the Politician's conditions are met, or if the Exofirms had been forced in response to the messages above, and if there had been insufficient investment by Developers and Industrialists in a particular firm, the following message would appear.

```
NO LAND DEVELOPER INVESTMENT IN EXOFIRM[n] [Name of firm]
TYPE 1 TO FORCE IN, EOF TO BYPASS
```

#26 If the Operator wants to have the firm locate in the region despite lack of investment, he may type a 1 and press EOF to continue. If the Operator does <u>not</u> want to override normal processing, he simply presses EOF.

```
EXOFIRM[n] WILL LOCATE IN AA [a] UNLESS CHANGED
HIT EOF IF OK OR TYPE A DIFFERENT AA (I2)
```

#27 This message is printed only if the exofirm has entered the community, either by natural processes or through operator action, and switch 10 was set in response to message (25). By typing in a two digit analysis area (leading zeroes included), the operator can change the area of entry.

After repeating the above messages for all exofirms where appropriate, message (28) is printed.

```
ENTER ADDITIONAL EXOGENOUS EMPLOYMENT.
AA (I2). CLASS --IND=1,  BUR=2 (I1), NUM. EMPL. (I4), ADDED TAX BASE (F10.0),
                                                                ACRES (F5.0)
```

```
ENTER 0 TO END LOOP
--/-/----/----------/-----/
02 1 0900 5000000.00 200.0
```

If the operator chooses to change the distribution or magnitude
of employment (see STEP 5 in Volume 1, Section 6), he can do so
at this point. If this option is pursued, the operator must
specify the location (a two-digit analysis area -- leading zeroes
where appropriate), the type of employment (1 manufacturing, 2
bureaucratic), the number of employees (right justified), the
#28 addition to the tax base (with decimal), and the number of acres
of land used (with decimal). These entries are lined up under
the printed dashes. If the option is not pursued or to signal
the end of the changes, press EOF. (The example shown indicates
that 900 manufacturing employees are being added to analysis
area 2 in a plant worth $5,000,000 on a parcel of land covering
200 acres.)

After the exofirms and employment changes are processed, the following diagnostics are typed. No response from the operator is required.

```
EXOFIRM ADDED TO AA
        32          13

CE=     2.90   CEN=    2.50    CL=    1.42   USUNE= 0.0510  PE= 93.59
  UNEMP=       0.0146  0.0357     0.0435     0.0623 0.0839
```

#29 The next relevant message is

```
SW 4 TO PRINT TOMM START PICTURE
```

The TOMM start picture displays the initial distribution of
population, employment and land use prior to the reallocation
#30 performed by the TOMM model. It is usually nothing more than
a reiteration of summary data from the last cycle and is not
printed.

```
SW 1 TO SUPPRESS TOMM PLANNERS' TABLES
SW 4 TO PRINT TOMM FINAL DIAG PICTURE
```

Throughout the processing of APEX, the operator has the option of suppressing the output for particular roles. The TOMM PLANNERS' TABLES are a critical part of the Planner's output and are not normally suppressed. Printout for various roles is most often suppressed when the cycle is being rerun to correct an error where the error does not affect the role being suppressed; that is, the new output would be identical to the old.

The TOMM FINAL DIAG PICTURE, like the START PICTURE, would be printed only for diagnostic purposes.

Press Program Start to continue.

```
SW1 TO SUPPRESS APCO OUTPUT
```

#32 In most games, the APCO role is included and, therefore, the output is printed (SW1 is left down). Press PROGRAM START to continue.

```
THESE NUMBERS REPRESENT THE PUBLIC'S SENSITIVITY TO ODOR AND SMOKE TO
CHANGE THEM, TYPE 1, TO KEEP THEM THE SAME HIT EOF
50.0  50.0  50.0  60.0  30.0
```

This message is not typed if the APCO's output is suppressed.

#33 These values are rarely changed. However, increasing them will cause a moderate increase in the number of complaints generated in the APCO's output. Typing a 1 will cause the following message to be printed.

```
TYPE IN NEW NUMBERS, 5F7.2, JUST AS ABOVE
```

#34 Now numbers are typed in directly under the corresponding numbers above.

```
SW 4 TO PRINT ALL DIFFUSION OUTPUT FOR OPERATOR
SW 5 TO ENTER 'BACKGROUND' POLLUTION LEVELS
```

#35 Switch 4 is set only when the operator wants to see more detailed pollution data. This option is rarely used and was added initially to aid in debugging the program.

Switch 5 is set when the operator wishes to exercise his option to modify the pollution levels in the region. If switch 5 is set, the following message is printed once for each of four seasons.

```
ENTER 'BACKGROUND' POLLUTION LEVELS FOR [season] (5F6.0)
WILL BE ALL ZERO UNLESS VALUES ENTERED
```

The operator can impose background pollution by adding in a fixed quantity of each contaminant throughout the region. The additions are made by typing a 6 digit number for each contaminant in the order of particulates, SO_2, CO, NO, HC. Each number must be real (with a decimal) and must fill a field of 6 spaces utilizing leading or following zeroes as appropriate. The entry for particulates is in micrograms/cubic meter. All other entries are in parts per million.

```
SET CONSOLE SWITCHES 1 THRU 7 UP TO SUPPRESS CORRESPONDING DEVELOPERS
OUTPUT SWITCHES 8-14 TO SUPPRESS INDUSTRIALISTS N+7
SWITCH 0 TO OVERRIDE AUTOMATIC DIVIDEND CALCULATION
```

#37 The switches 1-14 are set to suppress the printout pertaining to the corresponding playing developer or industrialist (SW8 corresponds to industrialist 1, etc.) If the Industrialist or Developer was not entered as a played role (messages 16 and 17), the corresponding printout is automatically suppressed despite the setting of the switches.

If switch 0 is not turned on the program will automatically calculate dividends for each of the Industrialist. (The cash increase is limited to $100,000). If an Industrialist has taken out a large loan and not spent all of the new cash, the operator should override the automatic dividend calculations to maintain the player's cash balance at an appropriate level. If the operator chooses to override the automatic calculation of dividends (for example, if he has some prior agreement with the Industrialist or knows that the Industrialist has made a loan to increase cash on hand for expansion but has not yet spent it), he must set switch 0 and respond to the following message for each Industrialist.

```
INDUSTRIALIST [3]
  CASH AT START OF CYCLE [n]        [  100000.]
  CASH CARRYOVER IF NOT CHANGED     [1136798.]
  CASH INCREASE                     [1036798.]
  ENTER DIVIDEND DOLLARS OR ZERO IF OK
1000000.
```

#38 The operator types in the dividend as a real number with a decimal point. The program responds.

```
NET WORTH                            [951798.]
DIVDS AS PERCENT OF NET WORTH      [105.06]
NET PROFIT                           [1036798.]
DIVIDS AS PERCENT OF NET PROFIT    [96.45]
CASH CARRYOVER IF NOT CHANGED        [136798.]
CASH INCREASE                         [36798.]
ENTER DIVIDEND DOLLARS OR ZERO IF OK
```

#39 If the amount of cash increase (or any other variable) is now in line with the operator's objectives, the operator presses EOF and the program continues on. If not, the operator enters a new dividend and message (39) will repeat itself with new values where appropriate. Note, the new dividend overrides the old -- it is not added to it. Also, the example shown produces the same result as the automatic dividend option.

```
CES 1 -- SUPPRESS CITY POL. OUTPUT
CES 5 -- SUPPRESS COUNTY POL. OUTPUT
CES 2-4 -- PRINT CORRESPONDING JUR.
```

Normally all switches remain down here. The Politicians' output

#40 for City and County is usually printed and the output for simulated jurisdictions 2, 3 and 4 is not usually printed. The output for simulated jurisdictions is of no use in the current version of the program.

CES 1 -- SUPPRESS CITY PLANNER OUTPUT
CES 5 -- SUPPRESS COUNTY PLANNER OUTPUT
CES 2-4 -- PRINT RECOMMENDATIONS FOR JURS 2-4 RESPECTIVELY

#41 Normally all switches will remain down here too.

CES 1 UP -- SUPPRESS PERCENTAGE DISTRIBUTION CHART
CES 2 UP -- SUPPRESS DEVELOPED AND VACANT LAND TABLES

#42 These tables are not normally suppressed.

SET SWITCH 0 UP FOR CYCLE 0 SUMMARY TABLES -- DOWN FOR NORMAL CYCLE END SUMMARY

#43 Switch 0 would <u>never</u> be set during processing of any cycle. This option is used only to check the initial data prior to running the game. It serves only the designers of the program, not the users.

SET SWITCH 1 UP TO SUPPRESS SUMMARY TABLES

#44 The summary tables include the three tables posted on the wall each cycle and portions of the Planners' output: The General Summary Information, Total Property Distribution and Capital Plant Indices by Category. These are usually printed.

SET CES 7 TO SKIP ISSUES, CEM AND NEWSPAPER

#45 If the operator does not wish to print the newspaper, he may set switch 7. Usually, however, switch 7 will be left down and message (47) will be printed. If switch 7 is set, the following message would be printed.

RUN TERMINATED BYCES 7 ON
NEWSPAPER MAY BE PRINTED LATER BY EXECUTING SUM1 ON THIS DISK

#46 Usually, the reason for not printing the newspaper is time pressure -- there may not be time to print the newspapers (a long, tedious process) in sequence with normal processing. When (or if) you wish to print the newspapers, the rerunning of the cycle can be bypassed by doing the following. Using the cold start procedure to start up with a new disk (see the opening to this chapter), the program can be restarted at this point if a // XEQ SUM1 card is used in place of the // XEQ SETUP card and the data cards are left out of the deck. If this procedure is followed, message (43) will be printed and the program will start from there. Since the summary tables have already been printed, SW1 should probably be set in response to message (44). Switch 7 should be left down to go on to the newspapers in response to message (45).

```
CHANGE DISKS NOW
PRESS PROGRAM START WHEN ISSUE DISK IS READY
```

#47 At this point, remove Disk 1 and insert Disk 2. As the message indicates, the cold start procedure is not used when disk 2 is ready. Simply press PROGRAM START. (If running on a multiple disk drive system, and disk 2 is already ready on another drive, you do not wait to change disks; simply press PROGRAM START.)

The first message on disk 2 is as follows.

```
ISSUE AND NEWSPAPER DISK, NO.2
    CONTINUATION FROM DISK 1 FOR TEAM [       ] FOR CYCLE 3
```

#48 If Cycle 1 decisions (the initialization decisions) were just processed, the following message is printed.

```
SET SWITCH 1 ON IF YOU DO NOT WANT TO PRESET CYCLE 1 DECISIONS
```

#49 As a rule, leave switch 1 down and press PROGRAM START. Preset decisions for the issues of the "cycle 0" newspaper are stored on the disk and are normally adequate. If the operator wants to alter the composition of the first newspaper given to the players, he might want to change those preset decisions. If the switch is turned on, issue decision cards are processed as always (see message (52)). If not, message 52 is bypassed and the program skips to message (56).

```
DATE TO BE PRINTED IN NEWSPAPER IS
    [WEDNESDAY, FEBRUARY 3, 1971]
IF OK HIT EOF, ELSE ENTER NEW DATE--MMDDYY
```

#50 The date is calculated from the date entered on disk 1. If it is incorrect at this point, a new date can be entered as before -- a two digit month, a two digit day and a two digit year (with leading zeroes as appropriate). If a new date is entered, message 50 is repeated. If EOF is pressed, the following is printed.

```
TYPE NUMBER OF COPIES OF NEWSPAPER DESIRED
```

#51 Type in the number of copies of the newspaper (from 1-99) desired. If multiple part paper is used, each copy asked for generates as many copies as in the multiple part paper. After typing the number, hit EOF.

```
ENTER ISSUE DECISIONS FOR CYCLE [2] -- CARDS READY....
```

#52 The issue decision card (or cards) must be placed in the 1442 card reader at this point. The reader should be cleared first by pushing NPRO -- unless the card (or cards) was placed at the tail end of the main input stream (following the XX). After the card (first card) is readied in the reader, PROGRAM START must be pushed. The cards will be listed on the printer as read. If no errors are detected, an additional listing of the decisions in "decoded" form will be printed. If errors are detected, a series of messages described in the appendix on error messages are printed.

```
FOR AN ELECTION THIS CYCLE, SET SWITCH 1
```

#53 Once the issue decision cards are stored correctly, the above message is printed. If switch 1 is left down (no election), the issue decisions read in above will be saved until an election is held based on those issues. To hold an election, set switch 1 and press PROGRAM START to get the following messages. If switch 1 is not set, message (55) is next.

```
FOLLOWING ISSUES CURRENTLY RECORDED FOR THIS ELECTION
[169] [32] [56] [25] [171] [8] [43] [96] [69] [2] [33] [58] [52] [74]
  [2] [33] [58] [74]  [17] [4]
```

#54 The issue numbers printed as a part of this message include all those issues appearing in all of the newspapers printed since the last election (up to 40). If the operator choose to read special issues instead of those on the list, he must set switch 2 and press PROGRAM START. The new issues are added precisely as those entered in response to message (52) except that there can not be bond or millage numbers associated with an issue. In fact, message (52) is reprinted if switch 2 is set.

If the issue list is satisfactory, ready the candidate election data cards in the 1442 reader. (If they were placed directly behind the issue decision cards, they will be ready; if not, push NPRO on the reader for two seconds, load the data cards and push start on the reader). Push PROGRAM START to read the candidate election data cards. If errors are found on the cards, follow the procedures in the appendix on errors. Otherwise, the following message will appear.

```
CES 8 -- CEM DIAGNOSTICS
CES 10 -- CEM RESULTS
```

#55 By setting switch 8, the operator will cause the machine to print diagnostics that are unintelligible to anyone but the program designer. Switch 10 will generate a breakout of the election results by category; that is, for each election, the percent attained by the incumbent will be broken into parts reflecting EOP performance, past performance, party support and a random part (See Vol. 1, Chapter 4). Push PROGRAM START to continue.

```
CES 11 -- BYPASS LINKED ISSUES
CES 12 -- BYPASS PRESET STATE & NAT'L ISSUES
CES 13 -- BYPASS PRESET COUNTY & LOCAL ISSUES
CES 14 -- TO ENTER ISSUE NUMBERS FROM KEYBOARD
```

#56 Normally, all of these switches would be left down. Setting switches 11, 12 or 13 will force the omission of the designated issues. (Linked issues are those triggered by the decisions made on the issues of the preceding cycle.) By setting switch 14, the operator can add to the county and local issues. If switches 13 and 14 are set simultaneously, county and local issues can be both added and deleted by first bypassing (SW 13) and then adding those to be retained plus all new ones (SW 14). To get listings of preset issues, see the description of the utility PRISS in Chapter 4. Press PROGRAM START to continue. If SW 14 is set, the following message appears, if not, message (58) is printed.

```
ENTER NUMBER 1 TO 400, EOF AFTER EACH. ENTER 0 TO END LOOP
```

#57 Issue numbers can be typed in, one at a time, in response to this message. By typing a 0 and pressing EOF, the program advances. To interpret any error messages, see the appendix.

```
SET SWITCH 4 TO INPUT SPECIAL HEADS FROM CARDS
SET SWITCH 5 TO INPUT FROM KEYBOARD
BOTH OFF IF NO SPECIAL HEADS
```

#58 If switch 4 is set and PROGRAM START pushed, the special headline cards (Ch. 2) will be read. It is possible, by setting switch 5, to enter special headlines from the keyboard. These are entered in the same format as the punched cards in response to the following message. If neither switch was set, the newspaper is printed and the program is complete.

```
NUMERIC CODE FOR LOCATION OF HEADLINE IN FIRST CHARACTER OF LINE
1=SPECIAL LEADING HEADS, 2=NAT'L, 3=STATE, 4=COUNTY, 5=CENTRAL CITY,
6=SUBURB, 7=TWP. 1, 8=TWP. 2, 9=BUSINESS PAGE
LINE BELOW IS 80 CHARS. LONG WITH/ AT 41'ST CHAR.
N-------------------------------------------------------------------------------
```

#59 Special headlines can be typed in as indicated. Whether reading from cards or keyboard, a reproduction of the headlines appears on the printer for proofreading and the following message is typed.

```
CHECK PRINTER LISTING OF HEADS AS THEY ARE NOW STORED
IF YOU WISH TO CORRECT AND RE-READ THEM, SET SWITCH 6 ON
```

#60 If the operator chooses to correct the headlines by setting switch 6, cards must be reread or headlines must be reentered. Otherwise, by pressing PROGRAM START with the switch off, the newspapers are printed and the program is complete.

SECTION 4

ERROR MESSAGES

Section 4-1

Error Messages

During the computer processing component of APEX, many opportunities exist for operator error. Frankly, it is a rare occasion when the processing of a set of decisions is free from error. The need for quick and accurate corrective action is most dramatically demonstrated during runs requiring rapid turn around because processing errors disrupt player strategies (omission of a single card might result in the non-execution of a crucial activity in connection with some major plan) and upset the morale of the audience and the timing of the game.

Errors arise from any sources. Players make errors in specifying the exact nature of their decisions. Coders of key punch forms make errors in transposition. Errors can be made by key punchers and, finally, the operator can communicate incorrectly with the program. The most difficult errors to correct are player mistakes -- unless the role advisor is sufficiently close to the players to place the logic of player decisions correctly on the worksheets, the computer operator can not correct these mistakes. (The MISMATCH OF SCALE error is the most common error of this type.) Other errors can and should be corrected; unfortunately however, the addition of a zero on a number, the minor miscoding of a single number, etc., often go by undetected. One useful activity that often minimizes the occurence of such errors is the rereading of each set of input cards prior to advancing the program beyond that input section. A second activity, if time is available, is the reading of the output for glaring inconsistencies and errors and the rerunning of the cycle to correct the mistakes. As the operator gains experience with the game, his speed and ability to catch errors before output is returned to the players should go up markedly.

This chapter takes up all the diagnostic error messages that appear on the console, printer or in the accumulator during the normal run of APEX. Each type of error is considered in a separate section of this chapter.

A. Errors in Reading the Main Input Stream

As the main input cards are read by the INPUT program stored on disk 1, incorrect cards are rejected (sent to stacker 2) and an error diagnostic is written on the printer under the line containing the image of the incorrect card. At the conclusion of the read operation, the operator must correct all incorrect cards and instruct the program to read additions to the input. If scanning the printout of card images reveals the existence of undetected errors, the errors should be corrected and the input cards reread entirely. Each error message is described briefly in the following paragraphs.

Several errors are associated with project and program input cards.

***ILLEGAL PROJECT NUMBER [n] ***

If the number entered on a project or program card is not associated with any existing project or program, the above message is printed.

***MISMATCH OF SCALE . . [n] ***

Assignment of a prnject or program to a location which is larger (more inclusive) than the scale indicated on the project list will cause this error message to appear.

***MISASSIGNMENT OF AREA [n] ***

When analysis areas are coded incorrectly, most often when projects restricted tc a particular area are assigned to another area, this message is printed.

***PRICE (F-1) OUT OF RANGE ***

When prices on capital improvement projects are outside the range specified on the project lists, this error is printed.

Another message may appear after a capital improvement or program card.

This message is not an error.

***FIELD I2 (LOC) CHANGED TO [n] ***

Assignment of a project or program to a location which is smaller (less inclusive) than the scale indicated on the project list will activate a mechanism within the program to change that location automatically and write out this message.

Errors in the balance of the input cards will trigger the following diagnostics.

***NUMERICAL VALUE [n] IMPROPER ***

An entry on a card that is clearly impossible (for example, a zero entry for a player number) will trigger this message. The number [n] indicates which of the 12 possible numeric fields is in error.

***NON-GAME LD OR IND ***

If a developer or industrialist number appears on card but is not entered as a playing developer or industrialist number in response to messages (16) and (17), this error is printed.

*** ALPHA CODE UNKNOWN ***

If the computer can not interpret the first two letters on a card, this message is printed.

*** BR/SR COMBO WRONG ***

If, on land transfer cards, the buyer and seller are the same, the preceding message is printed.

*** INCORRECT BOND/MILL NO. ***

This message is written in one of two cases. If the card requesting a bond or millage issue (BS card) is not read prior to the project or program request (CP or SP card) tied to the bond or mill, this message will appear. Rereading the CP or SP card without alteration as an addition to the input will rectify this error. If the information on linked BS, CP or SP cards is inconsistent (for example, different jurisdictions for the project on the matched cards), the message is printed. The cards must then be corrected and reread.

*** MORE THAN 20 BONDS/MILLS, ABOVE CARD IGNORED ***

If more than 20 requests for Bonds or Mills are processed in a cycle, the above message is printed. Either the card triggering the message is omitted or the input cards can be reread omitting some other less desirable request.

One more error message is printed in response to problems on cards dealing with land transactions.

*** COMBO IMPROPER ***

Illegal combinations of zoning category and development code (for example, land recorded as zoned 1 for single family residential, but developed as category 4 multi-family) or development and land measure (developed residential records) as acres not units) will trigger this message.

One last message may appear on the console as a warning of a potential error.

*** MISSING APCO COUNTY BUDGET CARD ***

This message can be bypassed by proceeding as normal.

The messages listed above are responses to errors in input found by the APEX program itself -- the program checks the cards for mistakes and inconsistencies and reports any errors. Quite often, however, errors of a nature that prevents the computer from reading the input cards are made in preparing the cards. These errors cause the computer itself to transmit diagnostics to the operator. Upon encountering an error of this kind, the computer stops and displays a diagnostic code in the accumulator -- one of the rows of flashing lights atop the IBM 1130 console A verbal description of the error can be found by checking in appendix A in the disk monitor system manual. The diagnostic codes are displayed in hexadecimal -- each grouping of four digits in the accumulator represents a single digit in base 16 arithematic. Table 1 gives the digits associated with each possible configuration of lights.

Table 1. Hexadecimal Equivalents

1 2 3 4 = 0	1 ② 3 4 = 4	① 2 3 4 = 8	①② 3 4 = C
1 2 3 ④ = 1	1 ② 3 ④ = 5	① 2 3 ④ = 9	①② 3 ④ = D
1 2 ③ 4 = 2	1 ②③ 4 = 6	① 2 ③ 4 = A	①②③ 4 = E
1 2 ③④ = 3	1 ②③④ = 7	① 2 ③④ = B	①②③④ = F

A circled number indicates that the light is on.

The most common errors encountered by the computer causing messages to be displayed in the accumulator are incorrect characters on cards and incorrect placement of decimal points (two in one number or one in an integer field). The displays for these errors normally begin with a hexadecimal ①②③④ = F. Most of these errors do not abort the program; that is, by pushing PROGRAM START, the operator can restart the program. However, the incorrect card is not read correctly (zeroes are input in place of those numbers that could not be read). Therefore, upon encountering an "F" error, the operator should note the card containing the error (it is not rejected and sent to stacker 2), push PROGRAM START, and continue reading the balance of the input. Then, the incorrect card should be corrected as should all other errors and the entire input sequence repeated (switch 13 up to completely restart READA -- message (22)).

If an accumulator displayed error is encountered that aborts the program, pressing PROGRAM START returns control of the machine to the monitor system and the APEX program must be started anew. When message (9) appears, switch 13 can be set to save a small amount of time.

As an example, suppose a parenthesis is punched on a card by mistake:

```
CP    12    1)    401
```

While trying to read the card, the computer would stop (lights would no longer flash) and the accumulator would show

/⓪①②③/ 4 5 6 7 / 8 9 10 11 / 12 13 ⑭⑮ /

This is an F003 error -- illegal character encountered in input record. By pressing PROGRAM START, the operator can restart the program for the purpose of locating other errors in the input stream. After correcting all the errors, he must reread the entire set of input cards.

B. Errors in Transfer from Disk 1 to Disk 2

At the time when control of the program is transferred from disk 1 to disk 2, two checks are made to insure that the disk transfer is implemented correctly. First, the program verifies the validity of the team number being transferred. (All information is transferred from disk 1 to disk 2 through COMMON; therefore, if COMMON is incorrect or if the second disk is run independent of the first -- something only the program designers can do -- the resulting team number may not be between 1 and 5.) If it is incorrect, the following message is printed.

```
TEAM NUMBER INVALID. ISS 1 MUST BE EXECUTED ONLY BY TRANSFER FROM DISK 1
EXECUTE 'RENEW' TO REPEAT NEWSPAPERS
```

If the team number is correct, the program checks to make sure that the cycle number transferred from disk 1 is one of the following:

a. Cycle 1 -- in which case it assumes that this run is the start of a new game

b. Cycle transferred is the same as that stored on disk 2 -- in which case it assumes that the cycle is being repeated in its entirety

c. Cycle transferred from disk 1 is one greater than that stored on disk 2 -- the normal cycling procedure

If the machine does not verify that one of these cases exists, it prints the following:

```
CYCLE NUMBER MIXUP. EITHER WRONG DISK OR NOT A TRANSFER FROM DISK 1 CYCLE
LAST RUN ON DISK 2 WAS [n]
```

If the cycle number received by disk 2 is less than 1 or less than the current cycle number on disk 2, but not equal to 1, the following will be added.

```
CYCLE ON DISK 1 SHOULD HAVE BEEN 1
```

If the cycle number received by disk 2 is more than one greater than that saved on disk 2, the following is added.

```
CYCLE FROM DISK 1 SHOULD HAVE BEEN SAME OR ONE GREATER
```

In each of the above cases, the program will terminate execution and return to monitor control. The operator must replace disk 1, execute SUMI (read in a card with a // JOB followed by one saying // XEQ SUMI and a blank), and place the proper disk 2 in position in response to the appropriate command.

C. Errors in Reading Issue Decisions

A subroutine, DECOD, is used to scan and interpret issue decisions (and CEM input cards) of the form iiia(bbb). DECOD searches for Syntax errors only. The only legal characters which may appear on an issue decision card are: blank 1 2 3 4 5 6 7 8 9 0 ( ), *
If a character other than the above is found, this message is printed.
ILLEGAL PUNCH IN COL [n]

> This message indicates the column containing the invalid character to assist the operator in making the needed corrections.

Since the maximum number of digits in a single string on a card is four and the minimum is one, a check is made for excessively long and short combinations. Strings of unacceptable length result in either of two messages.
MORE THAN 4 DIGIT NUMBER, COL [n]
LESS THAN 1 DIGIT NUMBER, COL [n]
If only one digit is found between two commas, it is assumed to be an alternative number with no preceding issue number and results in the following.
ZERO ISSUE NUMBER
Note, that no column number is given -- the operator must scan the incorrect card for the error.

Whenever a right parenthesis ")" is encountered, the program checks that there was a preceding left parenthesis. The failure to pass such a check results in the following.
WRONG PARENTHESIS, COL [n]

Each of the above errors terminates the scanning of the card. The program pauses while the operator corrects the error. This is done by removing all cards in the input hopper of the reader and clearing the reader by pressing NPRO. The incorrect card is the next to last card in the output stacker after the reader is cleared.

After all syntax errors are corrected, the program checks the content of the cards. Two warning messages may be printed.
MORE THAN 40 SETS OF NUMBERS. STOPPED AT 40.

> The maximum number of issues that can be stored at a single time is 40. Additional issues are ignored. The above message indicates that such "ignorance" is present.

WARNING -- NO TERMINATING CHAR. LAST NUMBER NOT STORED

> The last character on each issue card must be either an asterisk or a comma. Any characters appearing on a card after the last comma (but not followed by an asterisk) will not be carried over to the continuation card but will trigger the message printed above. This is only a warning in so far as the incomplete issue entry triggering the message may be restarted on the continuation card without deleting the partial entry

from the first card. If the incomplete entry is not repeated, it is, in fact, an error.

In the absence of warning and/or syntax error messages, a series of messages may be printed indicating that the entry for a given issue is invalid. If the issue number is not within the 1-400 range, does not correspond to an initialized entry in the data file, or has an incorrect alternative number associated with it, one of the following messages is printed.

```
ILLEGAL ISSUE NO. [nnn]
       or
ILLEGAL ALTERNATIVE (n) FOR ISSUE [nnn]
```

In addition, if bond/mill numbers are entered on the issue cards, the program compares the first digit of the bond or mill number to the cycle number. If they differ, the following is printed.

```
ILLEGAL BOND FOR ISSUE [nnn]
```

If any of the above messages are printed (including warnings), the following message is transmitted.

```
CORRECT ERRORS AND PUSH PROG START TO REREAD
```

At this time, the card reader should be cleared, errors corrected, and the cards reread (placed in the reader, readied and read by pressing PROGRAM START on the console). The operator might find it useful to save the listing of the issue cards (as they are read and as they are decoded) that appears on the card reader after the cards are read. The listing proves useful for rerunning the cycle at some later date and for error diagnosis.

Once the issue entries are accepted as valid, the program verifies that the issue list is consistent with the linkage list from the preceding cycle. (Each cycle, just prior to printing the first newspaper, the program prints a list of all issues that will link to other issues in the next cycle. This list is saved and the program checks that the issue decisions input for that next cycle include decisions for all those issues that required resolution to determine the nature of the linkage.) If some issue requiring resolution to direct the linkage to related issues is not included in the issue decision sequence, the following is printed.

```
**MISSING DECISION FOR ISSUE nn
CES 13 TO IGNORE MISSING DECISIONS. NO FURTHER LINKING FOR THOSE NOT
ENTERED
```

If an addition to the issue decisions is called for (or correction if an issue were entered incorrectly to cause the program to perceive an omission), the operator must correct the issue decision cards, clear the card reader, resubmit the issue cards to the reader, ready the reader and press PROGRAM START to reread the cards. To intentionally omit the linked issues, the operator may set switch 13 and advance the program in normal sequence.

One last message may appear after reading the issue decision cards. The program saves the record of the decisions from cycle to cycle for use in processing the candidate election model. A maximum of 40 issues can be saved from election cycle to election cycle. If that number is exceeded, the following is typed.

MORE THAN 40 ISSUES FOR CEM, nn IS IGNORED

This error is not disastrous (it occurs whenever the newspaper for a cycle is repeated by completely rereading the decisions). Because it is possible to input a special set of issue decisions for the Candidate Election Model, the issue list can be edited (to eliminate repetitions) when that model is executed.

D. Errors in Reading Candidate Election Model Input

The input stream recording the candidate decisions on issues is scanned by the same DECOD routine that interprets issue decision cards. Therefore, any or all of the syntax errors described in connection with the issue decision cards may appear in response to errors in the candidate decision cards. Note, references to column numbers in syntax error messages pertaining to the first decision card for a candidate are made relative to column 29, the first column in the scanned field (the first issue is entered in column 29 on a candidate decision card). Therefore, the operator must add 28 to the column number given to get the "correct incorrect" column.

In addition, a series of a straightforward error messages may be printed. ILLEGAL PLAYER NUMBER nn
This is printed if the player number is less than 1 or greater than 16.
ILLEGAL OPPONENT TYPE FOR CANDIDATE nn
This is printed if the opponent type for incumbent cards (player number 1-8) is not 1 or 2. It also appears if the simulated opponent type on the opponent cards (player 9-16) is not a digit between 1 and 9.
WRONG PARTY SUPPORT FOR CAND. nn
This message appears if the value in columns 4-8 of the incumbent card is negative or greater than 60.0.
PERCENT OVER 100 FOR ISSUE nnn CAND. nn
This is printed if the percentage appearing in parenthesis for a given issue (designating stress) is greater than 100.
ISSUE nnn FOR CANDIDATE nn NOT IN MASTER LIST
The program verifies that all issues entered for a given candidate were included on a previous issue decision card or input specially. If no record of an issue is found, the above diagnostic is printed.
DUPLICATE PLAYER NUMBER nnn
This message signals that more than one set of cards was entered for player nnn.
NO MATCHING CANDIDATE FOR CAND. nnn
For every incumbent card read (candidate number 1-8) a corresponding opponent card (candidate number 9-16 with 1 paired to 9, 2 to 10, etc.) must be read. A mismatch in pairing triggers this last message.

If any of these error messages are printed, the program signals the opportunity to correct mistakes as follows.
CORRECT ERRORS AND PUSH PROG START TO REREAD
The operator should clear the card reader, correct errors, replace the cards in the reader, press START on the reader and PROGRAM START on the

console.

One additional message is printed, if the situation is appropriate, for informational purposes.

```
PLAYER nn SPENT MORE THAN HE HAD (nnn PCT), CHANGED TO
FF.fff  FF.fff
```

The designated player entered percentages totalling more than 100% in assigning his funds to stress the various issues. The program automatically adjusts this "oversight" by cutting the percentages back proportionally. If this error arose because of an operator error in keypunching, the operator can halt execution and restart with RENEW (see the chapter entitled "Other Options").

SECTION 5

OTHER OPTIONS

Section 5-1

Other Options

This chapter deals with several nice features of APEX that can be used to enrich the game, correct errors or otherwise simplify the processing of a set of player decisions through the computer.

Routine File Changes

Assume, for the moment, that the operator chooses to risk changing the raw data in some file. In response to message 6, the computer operator sets switch 10 and presses PROGRAM START. (Note, before changing any data it is a useful precaution to print the data file to be changed prior to any modification. The file listing is then a check sheet for verification that all changes are, in fact, executed correctly.) The computer will type the format for data entry on the console typewriter as follows.

```
-- --- --- - ------ / ------------------
```

All information is recorded under the dashes according to these rules.

Col.

1-2 Two digit file number (the data set reference number of the data storage file containing the variable to be changed -- the next chapter is a complete description of data file organization). (Column 3 is skipped; therefore, there is no dash in column 3)

4-6 Three digit record number (the line or record number containing the variable to be changed).

8-10 A three-digit element number (the element number is the number of the variable within a record -- a number computed differently for different files. Further elaboration appears in the file report itself.

12 A one digit number designating whether the element is an integer or a real -- 0 for reals, 1 for integers.

14-19 A six digit integer number (if the variable to be changed is an integer, the new value is entered here -- right justified; if not, the field is skipped).

21-40 A six digit real number (if the variable to be changed is a real number, the new value is entered here -- anywhere in the field but with a decimal point).

Each component of the file change instruction must be lined up under the appropriate dashes. As each single line entry corresponding to a single change is completed, the operator presses EOF. The program goes back to message (2) to allow the operator to print the modified file thereby enabling him to check the validity of the changes.

As an example, consider the change needed to enter a new average annual wage rate for a gamed industrialist. The variable name for that rate is AVEWG. For any industry, it is the 56th element in the record corresponding to that industry in file 54. Thus, for industry 1 we would change AVEWG with the following entry.

```
-- --- --- - ------ / -----------------
54 001 056 0          9000.0
```

Adding and Removing Projects and/or Programs

A special program is available on disk 1 to facilitate the addition of capital projects and/or special programs to the project file (file 6) and to delete projects and programs when they are no longer needed. Once added, a new project or program should remain in the file for the balance of the game; however, all new projects should be deleted from the file prior to the initiation of a new game using the same disk.

This special program is entirely separate from the main APEX program; therefore, it must be executed before starting the processing of the cycle in which there is to be reference to the new project or program. The input to this special program is made from cards punched with the following format.

Project Card*

Cols.	Format	Item and Code
1-3	I3	Three digit integer project number (1-120)
4-5	I2	Location: 1-29 for Specific Analysis Area (A.A.) 30 for any A.A., 3 for any Ward 32 for any Jurisdiction
6	I1	Type: 1 = Streets, 2 = Sewers, 3 = Water, 4 = Parks and Recreation, 5 = Miscellaneous
7	I1	Capital Plant Index Impact: 0 = A.A., 1 = Ward 2 = Jurisdiction, 3 = County
8	I1	Years to Construct
9	I1	0 if Revenue Bond Financing is not allowed, 1 if it is.
10-14	F5.0	Acres required (Real number with decimal)
15-23	F9.0	Total Minimum Cost (Real number with decimal)
24-32	F9.0	Total Maximum Cost (Real number with decimal)
33-77	45A1	Project Title
78-80	I3	Issue to be printed when planner recommends project. (Usually blank for additions)

Program Card*

Cols.	Format	Item and Code
1-3	I3	Three digit integer program number (1-40)
4-5	I2	Number of years program is to run
6-7	I2	Location: 1-29 for a Specific Analysis Area (A.A.) 30 for any A.A., 31 for any Ward
(continued)		32 for any Jurisdiction and 33 for County-Wide
8-10	I3	Required project number (leave blank if none is required)
11-20	F10.0	Cost per year
21-68	48A1	48-Character Title
69-70	2X	Blank
71-73		Number of issue to be printed during the last year of the program (usually blank for additions)

All integer fields (I Formats) must be right justified.

Once the input cards are prepared, the operator can run this special utility program by submitting the data deck (if there are additions) preceded by a // XEQ MAK06 card. If no program has been run using disk 1 prior to the execution of this utility, the machine must be "cold-started" (see the chapter "Running the Program"). Otherwise, the // XEQ card must be preceded by a // JOB card.

At the beginning of execution the program will type the following message on the console typewriter.

```
CES 0 UP -- SELECTIVELY REMOVE PROJECTS AND/OR PROGRAMS
```

(A) To delete projects or programs, switch 0 is set and PROGRAM START is pushed. If not, message (6) is printed.

```
ENTER PROJECT NUM FROM 1 TO 120 OR PROGRAM NUM FROM 121 TO 160 (N+120)
EOF AS LAST ENTRY TO END LOOP
```

(B) The operator can now type the number of the project to be deleted -- from 1 - 120 -- or program numbers from 121-160. These numbers do not have to be right justified. By hitting EOF, the PROGRAM will advance.

```
CES-1 -- TO ADD PROJECTS
CES-2 -- TO ADD PROGRAMS
CES-4 -- INITIAL LOAD
```

(C) Use switch 4 option only for initializing the file (reset file contents to zero to completely destroy the old file) not for adding or deleting. Use switch 1 option for the addition of project; switch 2 for programs. A blank card signals the end of project cards or program cards, and must be used with both (i.e., two blank cards are necessary with any combination of the switch 1 and 2 options). For example, if there were no project cards and several program cards (switch 2 option), a blank card preceding the program cards signals the end of the non-existent project change cards, and a blank card following the program cards signals the end of the latter cards. After input, the following is typed.

CES-3 -- TO PRINT NEW PROJECT LISTS

(D) New lists must be distributed to the players with the new projects and/or programs. Setting switch 3 triggers the following. If the switch is not set the program terminates.

CES-4 -- TO SORT BY BUDGET TYPE AND SCALE
DOWN- IN NUMERIC ORDER

(E) By setting switch 4 up at this point, the project/special program list will be printed both by type and scale. Currently, projects are numbered in BUDGET TYPE groupings, and within these by SCALE, so this option is probably not necessary.

TYPE NUMBER OF COPIES OF PROJECT LIST DESIRED

(F) Type in any number from 1-99. The program will terminate after printing the project lists.

There are only two error messages included as a part of this program. The first is printed if the deletion option is selected at message (c)

*** WRONG NUMBER

The second is printed during addition if the number of the new project or program has a valid counterpart in the file.

DUPLICATE NUMBER, nn IS ALREADY IN THE FILE

Erroneous input cards will be sent to the alternative stacker on the 1442-6 or 7 reader. The program may detect other errors in the input stream but no messages will be printed. These cards will be sent to the alternative stacker and must be scrutinized by the operator and corrected.

Restarting Issue and Newspaper Processing

In the preceding chapters reference was made to the possibility of printing additional copies of the newspaper after stopping the program before all the required copies are printed. Also, reference was made to the possibility of rerunning a newspaper for a given cycle with different alternatives entered for the issues. These options are available through a utility program RENEW (REpeat NEWspaper), a program that allows reentry to disk 2 without reprocessing the bulk of the program stored on disk 1.

RENEW may be executed at any time after the initial transfer from disk 1 to disk 2. Once that transfer is successfully completed, the data passed from disk 1 to disk 2 is saved, making it unnecessary to execute the transfer again. RENEW is executed with the following cards:

```
a cold start - DM2
// XEQ RENEW
blank or issue decisions, etc.
```

The operator must follow the cold start procedure described for starting the game with disk 1 -- only, disk 2 must be designated as the principal drive (in the console on a one drive system or indicated in the console switches on a multi-drive system). The first message typed is as follows.

```
ISSUE AND NEWSPAPER DISK -- RESTART SEQUENCE
TYPE TEAM NUMBER (1 to 5)
```

(A) Because data for up to five teams may be stored on disk 2, the operator must enter the number of the team to be restarted. An entry out of range will trigger (B); otherwise, the program skips to (C)

(B)

```
*** TEAM NUMBER n ILLEGAL -- TRY AGAIN
```

```
TEAM IDENT WAS 'aaaaaaaa          AND CYCLE NO. WAS n
IF THIS IS TEAM YOU WANT TO RESTART HIT EOF, ELSE ENTER NEW TEAM NUMBER
```

(C) If the team number and/or identification is incorrect, the operator can type a new team number. (This method could be used to check the status of the disk). Otherwise, the operator hits EOF to advance the program.

```
CES 0 -- PRINT CYCLE 0 (EDITION 1) NEWSPAPER
```

(D) The cycle 0 newspaper is the newspaper that precedes the first preset cycle 1 decision sequence. It is used primarily to initiate linked issues in subsequent newspaper and may be printed for distribution to the players to give them the background information on issues appearing in the cycle 1 newspaper.

Caution: This option should be exercised prior to the start of a new run only. Printing the cycle 0 newspaper destroys the record of the last cycle run for the designated team and resets the disk for starting a new run. Normally, this option should <u>not</u> be used. Press PROGRAM START to continue. If switch

0 is set, the following is typed; otherwise message (G) or (H) is printed.

```
TYPED IN UP TO 40 CHARACTER TEAM IDENTIFICATION
```

(E) The operator can enter a team identification here or leave the identification blank by hitting EOF.

If switch 0 was not set in response to message (D), and the disk never used before for the team selected, the following warning is typed as well.

```
CYCLE NUMBER LESS THAN 1. ONLY CYCLE 0 NEWSPAPER CAN BE RUN
TYPE IN UP TO 40 CHARACTER TEAM IDENTIFICATION
```

(F) The operator responds as if message (E) alone was typed.

If a complete copy of the newspaper was not printed during the normal execution of the program, the following is typed.

```
CYCLE n PAPER NOT PREVIOUSLY PROCESSED
NORMAL PROCESSING SEQUENCE
```

(G) This message signals the operator to have the computer read in issue decisions as if this were a normal processing sequence. Processing proceeds from here in normal fashion -- see message (50) in chapter 3.

If the message was printed previously, the following is typed.

```
REPEAT OF CYCLE n NEWSPAPER
CES 2 TO REREAD ALL DECISIONS (MAY BE CHANGED FROM PRIOR RUN)
    DOWN -- EXACT COPY OF LAST RUN
```

(H) To enter a new set of issue decisions, the operator sets switch 2. To repeat the last newspaper printed, switch 2 is left down. The next message is typed as follows.

```
TYPE NUMBER OF COPIES OF NEWSPAPER DESIRED
```

(I) The operator should type in a number from 1-99, hit EOF and advance the program in normal sequence -- se message 50 in chapter 3.

If it is cycle 1, this message will be typed before manual processing begins.

```
CES 1 -- IF YOU DO NOT WANT TO USE PRESET CYCLE 1 DECISIONS
```

By setting switch 1, a different set of decisions can be input for cycle 1 during normal processing.

Learning the Contents of Issue Related Files

In order to check the contents of the issue, alternatives and headline file, a utility program ISLST is included on disk 2. To check the order of the present issues, a separate utility program, PRISS, is also included on disk 2. Both programs are executed in the normal fashion; that is, with disk 2 designated as the principal disk, the cold start procedure is followed with the appropriate set of cards -- for ISLST:

```
a cold start card
// XEQ ISLST
blank card
```

and for PRISS

```
a cold start card
// XEQ PRISS
blank card
```

ISLST does not require any special operator intervention. Error messages may be printed in response to mistakes in the files storing the issues, alternatives or headlines. These messages are cues to the game designer and, therefore, are not included here.

The PRISS program lists the issue numbers of the preset issues in groups corresponding to the cycle by cycle output. After printing these strings of numbers, the following message is typed.

```
TO ENTER CORRECTIONS INTO ANY OF THESE FILES, TURN ON SWITCH 1
```

If switch 1 is set, the following is typed.

```
TYPE 3 DIGIT FILE NUMBER (204 or 206) FOLLOWED BY 2 DIGIT CYCLE NUMBER
TPYE FILE = 0 TO END LOOP
```

After entering the appropriate file number (204 for regular issues, 206 for national and state issues) and cycle number, one of the following is typed.

```
TYPE UP TO 32 NEW NUMBERS, 5 DIGITS EACH, RIGHT JUSTIFIED.
                    16 NUMBERS PER LINE
```

Changes in the preset issues are entered as indicated.

```
TYPE UP TO 20 NEW NUMBERS, 5 DIGITS EACH, RIGHT JUSTIFIED
        16 NUMBERS ON FIRST LINE, 4 ON SECOND
```

Changes in the state and national headlines are entered as indicated.

SECTION 6

APEX DATA FILE ORGANIZATION

Section 6-1

APEX Disk Data File Organization

Introduction

The disk files for APEX contain and are used for three basic types of "data":

1. Actual land use, budget, population, etc., tables used to initialize the game and which are then updated from cycle to cycle as play progresses.

2. Tables of constants, conversion factors, and alphabetic character strings used for output.

3. Arrays of values saved temporarily during a cycle to make room for other data in core and then retrieved for further processing.

File Organization

The files are organized by number into three sets:

1. Files 1-29 and 50-59 are used for temporary storage, constants, and current cycle table values.

2. Files 30-39 and 60-69 contain all "cycle 0" tables and values used for initialization.

3. Files 40-49 and 70-79 contain preceding cycle tables and values.

Only 51 of these 79 numbers are presently used for defined files, allowing some room for additions within the same organizational structure.

The files within the groups 2 and 3 and some of the files in group 1 are analogous, having (with two exceptions) the same length and variable names. For example, file 31 contains initial data for file 1 and the updated values stored in file 1 during a cycle are saved in file 41 before going on to the following cycle.

As a short-cut in communicating we frequently refer to these files as the "twenty files," the "thirty files" and the "forty files," and say, for example, that the function of a certain switch is to cause the twenty files to be transferred to the forty files. Note that the "twenty files" in this example, refer only to those files from 1 to 29 which have analogues in the other two groups (the "active" files).

The table on the next page shows the correspondence between the three groups (details of individual file contents will be given later):

Initial	Preceding	Current
30	40	20
31	41	1
32	42	22
33	43	23
34	44	4
35	45	15
36	46	16
37	47	7
39	49	19
61	71	51
62	72	52
63	73	53
64	74	54
68	78	58

File Record Length

Most of the APEX data files are defined within the programs as having multiples of 320 word records. This has two advantages from a programming standpoint:

1. The IBM 1130 Disk Monitor System FORTRAN disk I/O subrountine reads or writes a 320 word disk sector every time a disk READ or WRITE is used in a FORTRAN program even though the defined record length is much smaller. This is quite inefficient and time-consuming.

We group a number of related variables and arrays in 320 word buffers and reference all of them with a single disk READ or WRITE.

2. The use of the same record length allows us to use one simple subroutine to make file corrections from the console for all files.

Some of the files have more than one number within the programs which are equivalenced to the same disk file by *FILES cards.

The following section gives detailed descriptions of each of the APEX disk files.

FILES 1, 31, and 41

These are the largest data files. They contain land-use value and price data which differs in each of the 29 analysis areas. Each of the 29 areas requires three sectors for file 1, but only two sectors for files 31 and 41. The third sector of file 1 contains arrays used only in the SELL routine in the current cycle and is not initialized or saved.

These three files are defined as follows within the rou:ines which reference them:

```
DEFINE FILE 1 (87, 320, U, KXX)
DEFINE FILE 31 (58, 320, U, KXX)
DEFINE FILE 41 (58, 320, U, KXX)
```

Brief Summary: Note -- each of these arrays is repeated for each analysis area.

File 1

Sector (record) #1

Words	Type	Names	
1-35	I	BDR (5, 7)	Number of developed units of each of the five residential housing types owned by each of the 7 game realtors. B (Game Realtors) Developed Residential Units. By rules of FORTRAN storage the first 5 values (words 1-5) are for realtor 1; the next 5 (words 6-10) for realtor 2 and so on. The 5 values represent, in order of storage, the 5 developed residential zoning combinations in the following order: R1, R2, R3, M1, M2.
36-70	I	INDR (5, 7)	Same as above for up to 7 industrialists.
71-75	I	MDR (5)	Same as above for rest of private sector (market). Market Developed Residential Units.

Words	Type	Names	
76-80	I	PDR (5)	Same for the municipality (either real or simulated), depending on the analysis area.
81-85	I	CDR (5)	Same for county.
86-90	I	QR (5)	Empty residential units in each of the five categories. This is a subset of MDR; that is MDR includes both occupied and empty dwelling units. This array contains values as a result of emigration of families from an analysis area and is used to meet demands not met by realtors in the case of immigration or for sales to realtors and the public sector.
91-95	I	EMP (5)	Emigrated Families (formerly F array in APEX.) Families leaving M (general market residential) matrix in the analysis area, by zoning type. Added to: (1) when market sells more developed residential units than are empty (QR array), i.e., occupied dev. residential; (2) when market developed residential land is rezoned, thereby causing it to become vacant. It is not initialized and is set to zero at beginning of each cycle.
96		Waste	
97-180	R	BDC (6, 7)	Realtors developed commercial industrial and agricultural acres. B (Game Developers) Developed Commercial property in acres. Developed Non-residential Zoning: 1 - CL, Local Commerce 2 - CR, -Regional Commerce 3 - IE, Endogeneous Industry 4 - IX, Exogeneous Industry 5 - OD, Exogeneous Bureaucratic 6 - AD, Developed Agricultural
181-264	R	INDC (6, 7)	Industrialists developed industrial and commercial
265-276	R	MDC (6)	Market Developed Commercial and industrial property in acres.

Words	Type	Names	
			This array does include acres devoted to developed exogeneous commerce and industry.
277-288	R	PDC (6)	Public Developed Commercial
289-300	R	CDC (6)	County Developed Commercial
301-312	R	QC (6)	Empty developed "commercial". This is a subset of MDC (1), MDC (2) and MDC (3) in the same way QR is a subset of MDR.
313-318	R	P(3)	Public developed property P (1) - Municipality P (2) - County P (3) - Right of way and streets
319-320	R	U	Base density factor for developed residential. This value is the unit per acre for the least dense zoning category (R_1) in the particular analysis area. It is stored as the reciprocal. For example, 3 units/acre is stored as .3333 (1/3). (Multiplication factors for more dense zoning categories, which are constant across all analysis areas, are stored in array UDENV (5) in File 21.) The values are used to convert from units to acres and vice versa. ACRES = UNITS x UDENV (TYPE) UNITS = ACRES/ (U x UDENV (TYPE)) For example, 9 units of R_3 houses at a multiplier of 2 (.5) in an analysis area with a base factor of .33 would require 9 x .33 x .5 = 1.5 acres.

APEX File 1, Sector 2

Words	Type	Name	
1-84	R	BVRC (6, 7)	B (Game Realtors') Vacant Residential and Commercial property holdings in acres. The six zoning Categories, in order of storage are: 1 - Residential, Single family R 2 - Residential, multiple family M 3 - Commercial C 4 - Industrial I 5 - Office O 6 - Agricultural A
85-168	R	IVRC (6, 7)	Same as BVRC for game - Industrialists
169-180	R	MVRC (6)	Same as BVRC for non-game "market". This is the balance of all vacant land not in one of the other VRC arrays.
181-192	R	PVRC (6)	Same as BVRC for the municipality (public) in the AA.
193-204	R	CVRC (6)	Same as BVRC for county in the AA
			Note: in some APEX routines - for example , where we are summing all vacant land-these five arrays are combined and dimensioned as BVRC (6, 17). This is permissible since they are all stored consecutively by columns.
205-216	R	VVRC (6)	Value of Vacant Residential and Commercial Land in dollars per acre. These are the "appraised" values printed in the large table at the end and used for all sales of vacant property and assessment.
217-226	R	VDR (5)	Value of Developed Residential in dollars per unit of the 5 zoning categories, R_1 through M_2. This is the value used for assessing developed residential property and also the minimum sales price for sales of market developed residential units to game realtors.
227-238	R	VDC (6)	Value Developed Commercial (and industrial) property in dollars per acre. Same as VDR except for the six commercial/ industrial categories. See note above.

Words	Type	Name	
239-250	R	ROWF (6)	Right-of-way (streets, sidewalks, etc.) factor for each of the six categories of vacant land. The number of acres being developed is multiplied by the appropriate ROW factor to obtain the number of acres added to or subtracted from P (3) above.
		Note:	Because right-of-way is always subtracted from vacant acres used for development, when developed land becomes vacant, the number of acres for right-of-way is taken from city and added to the vacant areas.
251-252	R	E	This is now a constant 1.0 for all AA's. EVEC (in file 21) contains cost per unit or per acre for each developed type. If at some time we conclude that improvement costs vary by AA, the factor could be changed.
253-254	R	ADDTB	Added tax base for exogeneous property.
255-276	R	PDRC (11)	Price of Developed Residential and Commercial property. These prices are only used for sales of game players' developed property. Dollars per unit for the first 5 values which correspond to the 5 price classes for single family and then multiple family residential. Dollars per acre for the last six values which correspond to the six classes of commercial, industrial and agricultural property the game players may develop. Initial 6 percent over corresponding value.
277-320			44 unused words available for expansion.

File 1, Sector 3 (not used in Files 31 and 41):

Word	Type	Name	
1-154	I	OFRC (11, 14)	Property (units for 1-5 and 10ths of acres for 6-11) cffered by the game realtor and industrialists in the AA in current cycle.
155-308	I	SORC (11,14)	Property (analogous to OFRC) actually sold by the game realtors and industrialists.
309-319	L	FOVER (11)	Corresponds to the 5 developed residential zoning categories, plus six endogenous commercial/industrial classifications. This array is used only in SELL and represents the unfilled demand for land (in 10ths of an acre) after all "cascading" or pre-empting of land by higher use types on the first four iterations.
320	Unused		

Entering values into files 1, 31 or 41 from keyboard at beginning of cycle:

The ENTFL (Enter File) subroutine, which is generalized for any 320 word record file, requires the following keyboard input:

1. FILE NUMBER - a two digit number

2. RECORD - a three digit number, representing the physical record number (relative disk sector) containing the data element we wish to change.

   To better understand the value which must be entered for physical record number, let us distinguish between physical and logical record sizes. Each of files 1, 31 and 41 contain 29 logical records (includes the variables previously described). However, it requires three physical records of 320 words each (for file 1) to contain all of the data in one logical record -- accounting for the file definition of 87 x 320. Similarly, it requires two physical records for one logical record for files 31 and 41 (58 x 320).

   Therefore, to calculate the physical record number to be entered, use the following formula:

   $$PR = (LR - 1) \times N + RR$$

   where: PR = physical record number
   N = number of physical records per logical record
   LR = logical record number
   RR = the physical record within the logical record containing the element to be changed.

   For example, suppose we wish to change the value of MDC (4) - acres of exogenous industry - in AA 24 in file 1. We find from the preceding list that MDC is in the first sector (physical record) of each logical record.
   Therefore, using the formula above, we calculate:

   $$(24 - 1) \times 3 + 1 = 070$$

   Similarly, if we wanted to change the same element in file 41, the physical record would be:

   $$(24 - 1) \times 2 + 1 = 047$$

3. ELEMENT - a three digit integer value (from 1 to 320 if the element to be changed is an integer and from 1 to 160 if the element to be changed is a real variable,) representing the relative location within the physical record of the element to be changed. For example, from the preceding list we find that MDC (4) is in the real array MDC which uses words 265-266, MDC (2) uses words 267-268, and MDC (4) uses words 271-272. Dividing the even numbered word by 2 gives us the real element number of 136.

4. TYPE -the digit 0 if a real variable or the digit 1 if an integer variable.

5. INTEGER VALUE -right justified; leave blank if Type = 0.

6. REAL VALUE -up to 20 digits including a decimal point; leave blank if Type = 1.

SUMMARY

The computer will type a line of hyphens and spaces appropriately labeled to guide in entering values. For the example used above, the following complete entry would be made:

```
-- --- --- - ------ / -------
01 070 136 0          106.
```

This would enter 106.0 acres into MDC (4) of AA 24 in file 1.

A.P.E.X. File 2 - Six Sectors

This file contains a 29 x 29 matrix of real values representing the access time from each analysis area to every other analysis area.

It is one of a kind file, having no corresponding "thirty" or "forty" files. It is used only by the TOMM model and is read only in the sense that the values are never changed within the running of the program.

The FORTRAN file definition for it is (29, 58, U, KXX). Since it has 58 word physical records, it is difficult to modify it by the ENTFL routine from the keyboard.

A.P.E.X. File 3 - one sector

This file consists of eight 40-word records which have no "thirty" or "forty" analogues, only the first six records of which are presently used. It may be defined as: DEFINE FILE 3 (8, 40, U, KXX)
or, in some cases as: DEFINE FILE 3 (1, 320, U, KXX)

If defined as 8 x 40, the STPPA could be read by the following statement:

READ (3'8) STPPA, etc.

Record	Words	Type	Name	
1-3	1-58	R	ZH(29)	Maximum allowable density in persons per acre for each analysis area. This is used in the residential loop within TOMM model as a constraint on growth in an area.
	59-116	R	ATOT(29)	Total land area in acres for each analysis area. This is used in TOMM to compute the available land for commercial land by subtracting out all other uses.
	117-120	I	IFIL	filler (not used at present).
4	121-140	R	PCTNE(10)	Percent Change in Total National Employment. CE in Table 4 of APEX I.R. #35.
5	141-160	R	PEXTC(10)	Percent Change in EXogenous employment in APEX this Cycle, both bureaucratic and industrial, for 10 cycles. This change including addition of exofirms and gamed industrial growth after national growth rate applied.
	161-180	R	GOVTG(10)	The present growth proportion of exogenous bureaucratic employment (EBUR) in AA 8 (state government) for 10 cycles. A 3% growth would be stored in this array as 1.03.
	181-200	R	UNIVG(10)	The growth proportion of university enrollment by cycle.
6 201	201-220	R	CUSLF(10)	"CL" in APEX I.R. #35. Percent change in total US labor force.
	221-240	R	CAPLF(10)	Percent change in labor force in APEX area.
7	241-250	R	UNEMP(5)	Local unemployment for each of the five household types in current cycle. Calculated from UNAPX(t+1).

Record	Words	Type	Name	
	251-272	R	UNAPX(11)	APEX area unemployment rate. The subscript is cycle number +1. The first value is starting value for cycle 0, the second value calculated for cycle 1, etc.
	273-280		Unused	
	281-282	R	STPPA	The starting, cycle 0, student population of the university presently set at 27,000.
	283-302	R	STPOP(10)	A record of the new student population for each cycle. The value for the current cycle is printed in the output for Politician 2.
	303-320	R		

A.P.E.X. Files 4, 34 and 44:

These files each require six 320 word sectors (physical records). They are best described as government and schools financial data.

Sector 1:

Word	Type	Name	
1-2	I	NKK(2)	Count of 10 word bond or millage records in sectors 4-5 (NKK(1)). NKK(2) is presently unused.
3-12	R	VTPP(5)	Total assessed value of all private property in each jurisdiction. Note that VTPP(5) is the sum of VTPP(1)...VTPP(4) and is total for county.
13-162	R	TVBK(5, 15)	Total appraised value of all private property for each of the seven land developers (1-7), the seven industrialists (8-14) and the general market (5) in each of the 5 jurisdictions.
163-192	R	BKVT(15)	Total value of vacant property for each of the 15 elements of the private sector.
193-222	R	BKDT(15)	Total value of developed property for each of the 15 elements of the private sector.
223-230	R	ASSF(4)	Assessment factor for each of the four jurisdictions.
231-248	R	TAM(9)	Total assessment millage for each of the following jurisdictions: 1-4 Municipalities 5 County 6-9 Simulated Schools This is used for computing tax due from private sector. It is the sum of AMS + SPECO + SPECD.
249-266	R	OPMIL(9)	Total operating millage. Sum of AMS + SPECO.
267-284	R	AMS(9)	Total Normal Operating Mills.
285-302	R	SPECO(9)	Total of Special Operating Mills in affect.
303-320	R	SPECD(9)	Total of all debt retirement millages.

Sector 2:

Word	Type	Name	
1-18	R	CLL(9)	Current Local Limit on operating mills.
19-36	R	SLIM(9)	State Limit on operating mills.
37-54	R	BDOTG(9)	Total General Obligation Bonds Outstanding.
55-72	R	GONEW	Total New GO Bonds this cycle.
73-90	R	BDPGO(9)	Total Minimum Payments Due Next Cycle for GO Bonds.
91-108	R	DRFGF(9)	Total dollars in debt retirement fund at beginning of cycle. Generated from debt retirement millage + surplus from last cycle. Calculated for "this" cycle in last cycle's output links.
109-126	R	DRFGF(9)	Dollars for debt retirement transferred from general fund in current cycle.
127-144	R	CPFND(9)	Total dollars in Capital Fund. At beginning of cycle the value in file 44 is that remaining from preceding cycle. This is a continuous total reflecting additions from new bonds and transfers from general fund less amount disbursed to pay for projects.
145-162	R	TNEWB(9)	Total new bonds this cycle. Sum of approved GO bonds and all revenue bonds.
163-180	R	CFTRN(9)	Funds transferred to capital budget from general fund this cycle.
181-190	R	BDOTR(5)	Revenue Bonds Outstanding (no schools.)
191-200	R	BDPRV(5)	Total payment due on revenue bonds next cycle.
201-210	R	CREDP(5)	Percentage of state equalized valuation which determines GO bond limit.
211-220	R	BLIM(5)	GO bond limit.
221-229	I	CREDS(9)	Credit Rating for each jurisdiction. Integer values 1 to 3.
230	I	IWAST	Filler.
231-236	R	RATEI(3)	Interest rate on new GO bonds for the three credit ratings.
237-238	R	REDUX	Reduction factor applied to reduce CREDP when credit rating drops.
239-278	R	SBOPM(5,4)	Schoolboard special operating millages ending in each of next 5 years. First subscript = year in the future; second subscript = jurisdiction number.
279-318	R	SBDRM(5,4)	School board special debt retirement mills; first subscript = year in the future; second subscript = jurisdiction number.
319-320			Unused.

File 4, Record 3

1-160	R	OPVEC(16,5)	Operating budget percentages of the 4 municipalities and county (#5) carried from preceding cycle in file 44 and to next cycle in file 4.
Word	Type	Name	Columns are as follows for Municipalities (1-4) and County (5): 1-Legislative, executive (Note: this item will be carried as a dollar value which cannot drop. It may only be increased by player input. 2-Planning 3-Financial 4-Judicial 5-Fire and Police (Public safety for county) 6-Parks and Recreation 7-Water and Sewage 8-Refuse collection 9-Streets 10-Employees benefits 11-Public relations 12-Library 13-Public health (county only) 14-APCO (county only) 15-Welfare and hospitalization (county only)
161-320	R	STPCV(16,5)	Standard per/capita dollar values for each budget expenditure item. Analogous to OPVEC. Zero if does not apply.

Sector 4, File 4

1-18	R	TCYRV(9)	Total General Fund Revenue.
19-36	R	SURPL(9)	Surplus (or deficit) for each jurisdiction.
37-54	R	GENFT(9)	General Fund Property Tax revenue.
55-64	R	OTHAG(5)	Revenue from other agencies, in dollars. For county is services revenue.
65-74	R	POTH(5)	Percent change in other agency revenue each cycle.
75-84	R	FFLRV(5)	Dollar revenue from licenses, fees, and fines.
85-94	R	PLFF(5)	Percent change in license, fee and fine revenue.

95-104	R	ONTR(5)	Dollar revenue from other non-tax sources.
105-114	R	PONT(5)	Percent charge in other non-tax revenue.
115-134	R	SPGRT(5,2)	Special grant revenue. i,1 = Capital project fund grants i,2 = General fund grants
135-136	R	RCREV	Road commission revenue for county.
137-138	R	PRCV	Percent charge in road commission revenue.
139-156	R	TGOPY(9)	Total payments made on General Obligation bonds current cycle.
157-174	R	SURPL(9)	Surplus in debt retirement fund after all bond payments.
175-176	R	DRINT	Interest paid on debt retirement surplus.

Word	Type	Name	
177-320			146 words unused.

Sectors 5 and 6 (640 words) contain a record of Government (politicians') bonds and special millages which are in effect. Each bond or millage requires a 10 word block, the contents of which are described below. Since the 10 word blocks contain four integer variables and three real variables, the 640 word array is dimensioned twice and equivalenced as follows:

```
DIMENSION KSMR (10, 64), SMR (5, 64)
EQUIVALENCE (KSMR (2), SMR (1))
```

Within this array new bonds or millages are added to the bottom of the list; and each cycle, as bonds or millages expire, the table is packed so that the oldest (first added) are always at the top of the list.

Sectors 5 and 6 are equivalent to M.E.T.R.O. #4 sectors 1-2. Record of bonds and special millages in effect. Total of 64 ten word subrecords.

The 10 elements of each description block are as follows:

KSMR(1,I) - Jurisdiction; municipalities are 1-4. County is 5. Schools 6-9.

KSMR(2,I) - Bond or millage number.
Must be a three-digit integer starting with the cycle number times 100. For example, 301, 302, etc. for cycle 3.

KSMR(3,I) - type (ISPEC) code:

1. General Obligation bond
2. Revenue bond
3. Special Operating millage
4. Debt Retirement millage

KSMR(4,I) - Years to run.

SMR(3,I) - Dollar amount of bond first year, unpaid balance after that.

SMR(4,I) - Interest rate (as, for example, 6.25) for bonds or number of mills.

SMR(5,I) - Minimum payment due on bond next cycle (calculated in output link and carried to next cycle) or temporary storage of bond over or under payment (calculated in BDMIL and saved here until output link).

APEX File 5

This file contains descriptions of all the preset exogenous industries in APEX. It has no analogue in the thirty or forty groups and, since it is one of the three files with a defined record length less than 320, it is difficult to modify using the ENTFL feature in SETUP. (A note on how it can be done is included at the end of this section.)

It is defined as (56, 40, U, KXX) which means that there are 56 forty word records in the file.

The first 54 records are used for exofirm descriptions. The last two records contain special arrays describing the use of and aiding in the handling of the exofirms. At present (3/70) only 45 exofirms are defined, leaving room for nine to be added in the future.

Each exofirm (40 word record) consists of the following data:

Words	Type	Item	
1-16	I	NAME(16)	Name of the exofirm. Up to 32 characters, 2 characters per word (16A2).
17-20	I	IPREF(4)	Analysis area in which the firm wishes to locate in order of preference. One to four may be given.
21	I	NEMP	Number of employees.
22	I	IWAST	Unused.

Words	Type	Item	
23	I	NISSO	Number of issues generated if the firm enters the city.
24	I	NYRS	Years over which the realtors receive a return on their investment.
25-26	I	RATE	Rate of return on realtors investments, stored as 1.05 equivalent to 5%.
27-28	R	AMT	Amount of realtor investment required in order for an exofirm to enter.
29-30	R	ACRES	Numbers of acres of I2 land needed for this exofirm.
31-32	R	TAX	Dollar addition to the tax base in the analysis area in which the firm locates.
33-34	R	COST	Cost of capital plant additions (sewer, streets) required for entry of this exofirm.
35	I	IBIT	Word containing bits set of indicate conditions. These bits are tested by function ITEST: Bit 0 -- 1 if Industrial 0 if Office (bureaucratic) Bit 1 -- On (=1) if firm should be forced in even though all politicians and/or realtors conditions are not met. Bit 2 -- On if rezoning required. Bit 3 -- On if new sewers required. Bit 4 -- On if street improvements required. Bit 5-15 -- Unused.
36	I	INDEF	If non-zero, number of industry in file 58 which must be activated if firm enters. Will be 0 if no industry in file 58.
37-40	I		Words unused in exofirm description at present.

Record 55 of file 5 consists of a single array containing a record for ten cycles of those exofirms preset to desire entry each cycle. This mechanism uses the bit testing function ITEST and works as follows:

The array dimentioned as NEXOP (4,10) constitutes a four word vector of 64 consecutive bits which may be set either <u>on</u> (1) or <u>off</u> (0). Only six exofirms may desire entry in a cycle and the corresponding bit in the appropriate vector is set on for those so preset.

<u>Record 56 of file 5</u> contains information which is carried to the newspaper about those exofirms which <u>did</u> enter in a cycle.

Word 1 -- NUMEX -- a count of the number of exofirms which did enter.

Words 2-7 NEXAA (6) -- the number of the analysis area in which the firm in the corresponding element of NEXIN located.

Words 3-13 NEXIN (6) -- the numbers of the exofirms which entered in a cycle. For example, if NEXIN (1) = 12 and NEXAA (1) = 37 it means that exofirm 12 located in analysis area 37.

Words 14-17 -- EXOUT(4) -- A 64 bit vector, cleared at the beginning of GROW, in which the corresponding bit is set on by ISET for each exofirm which <u>does not</u> enter in a cycle.

Words 18-21 -- EXTHS(4) -- A similar vector of bits containing a bit set on for each exofirm which entered in the current cycle. It is the same information as that in NEXIN bit in a different form.

Words 22-25 -- EXIN(4) -- A vector of 64 bits which is cleared only at cycle zero and has a bit set on for each exofirm entering during a game. In other words,a cumulative record of all exofirms which entered during a game.

Word 26 -- IWAST -- Unused.

Words 27-28 -- IAIRQ(2) -- Bit corresponding to analysis area set on if flagged for air pollution headline in newspaper. Set in AIR.

Notice that each of the above arrays is explicitly defined as integer. An integer array, EXBIT (12), is also defined in GROW and the above arrays are equivalenced as follows:

(EXBIT (1), EXOUT (1)), (EXBIT (5), EXTHS(1)), (EXBIT (9), EXIN (1)).

APEX FILE #6

This file contains the details of up to 120 capital plant projects and 40 special programs defined for APEX politicians. Like file 5, it has no corresponding thirty and forty files and, since it has a record length of 32 words, is more difficult to modify from SETUP (but may be in special circumstances). A program (MAK06) is stored on the disk which may be used to add projects or programs to the file.

The file is defined as (160, 32, U, KXX). (16 sectors)
The first 120 records are capital plant projects and records 121-160 are special programs.

Each 32 word record - one project - is made up in the following manner for capital projects:

Word	Type	Name	
1	X	LOC	Physical location, scale and restriction of project. Most projects will not be restricted, but a few which are expansion of existing facilities, may be restricted to a specific analysis area (there will be ward or jurisdiction restrictions). Values 1-29 indicate location restricted to the corresponding AA. 30 - May be located in any analysis area. 31 - Must be located in an entire ward. (31 on disk.) 32 - Must be located in an entire jurisdiction. (32 on disk).
2	X	ITYPE	Budget category 1 - Streets 2 - Sewers 3 - Water 4 - Parks and recreation 5 - Miscellaneous
3	X	IMPACT	CPI Impact of the project 0 - Analysis area 1 - Ward 2 - Jurisdiction 3 - County

Note: Player input will be by jurisdiction (Pols 1-4, County = 5, etc.). If location of project (item #3) is not in player's jurisdiction, then the cost - impact on CPI will apply to that player's jurisdiction, not to the location.

Word	Type	Name	
4	I	NYRS	Years required to construct project.
5	I	NBOND	0 - May not be financed by revenue bond. 1 - May be financed by revenue bond.
6	I	NISS	Issue number which will be printed in newspaper when planner recommends the project.
7-8	R	ACRES	Acres required, if any.
9-10	R	CMIN	Minimum cost (total).
11-12	R	CMAX	Maximum cost (total).
13-32	R	TITLA(10)	45 character project title, packed 5 characters/2 words.

Each 32 word record for Special Programs is formatted as follows:

1	I	LOC	Same as preceding records, except 33 = County-wide program.
2			Not used.
3			Not used.
4	I	NYRS	Number of years program is to run.
5	I	NCPRJ	Capital plant project also required, if any. To be started at minimum cost.
			Cost of each program is jurisdiction-wide.
6	I	NISS	Related issue to be printed when program has one year to go.
7-8			Not used.
9-10	R	CMIN	Cost per year.
11-12			Not used.
13-32	R	TITLE (10)	48 character description of program.

APEX Files 7, 37, and 47

These files contain records of capital plant projects in effect, recommended by planners or not carried by politicians. Files 7, 27, and 47 are all defined as (3, 320, U, KXX) -- three 320 word sectors.

The first sector each file is a record of politicians (government) projects and the second contains those projects and programs recommended by the planners during the current cycle for inclusion by the politicians' next cycle. The third record is discussed on a following page.

The array name usually used for the first sector is KAPLT (5, 63) and for the second KPLAN (5, 63). The first five words of each of the first two sectors are reserved for counters, etc., of the projects actually stored. Each project record consists of a five word sequence; that is, words 6, 7, 8, 9, and 10 of each sector are the record of a project; words 11, 12, 13, 14 and 15 are the record of another project, and so on. Only words 6-320 of each sector are used for project records per sector.

The five words in each project record are defined as follows:

Word	Example	
1	KAPLT (1, I) or KPLAN (1, I)	This word is set non-negative when a project is added to a list. It is equal to the associated bond number if it is tied to a bond or to zero if not (all planners' recommendations, of course, carry a zero code). The years the project is to run (4th word)is decremented by one in the output link, and if the years remaining are zero, this first word is set equal to -1. (For planners' recommendations, sector 2, this word is set to -1 in CPT if a recommendation matches a politician's input.) Politicians' project lists are "packed" at the beginning of CPI each cycle to move all multiyear projects - i.e., projects for which the first word is non-negative- to the head of the list before any new projects are added.
2	KAPLT (2, I) or KPLAN (2, I)	Project number * 10 + CPI Impact
3	KAPLT (3, I) or KPLAN (3, I)	Location of project * 100 + Jurisdiction inputting (to force county projects to print in county budget). 1-29 - specific analysis area 30 - Ward 1 31 - Ward 2 32 - Ward 3

33 - Ward 4
34 - Ward 5
35 - Ward 6
36 - Jurisdiction 1
37 - Jurisdiction 2
38 - Jurisdiction 3
39 - Jurisdiction 4
40 - County

4	KAPLT (4, I) or KPLAN (4, I)	Years project is to run, if new. Years remaining if multiyear project after first year.
5	KAPLT (5, I) or KPLAN (5, I)	Cost per year in <u>thousands</u> of dollars. In other words, a project cost of $325,000 would be stored as the <u>integer</u> 325. If, due to division, a cost is not an even 1000, it is rounded up to 1000. The first five words of each sector are used as follows:
1	NCP	Count of the number of projects in the list in the corresponding sector at any time.
2		Unused
3-4	PJCST	Total dollar value of the projects in effect in the list.
5		Unused.

The third sector of each of these files is used for two purposes:

(1) A record of special programs in effect.

(2) A record of those projects and programs recommended by the planners the preceding cycle but not put into effect by the politicians.

Unlike the first two sectors, the third sector is divided into <u>four</u> word subrecords (since cost is not needed).

The first four words of the sector are:

1	I	NSP	Number of special programs in the first list.
2	I	NOP	Number of project records in the second list.
3	R	PGCST	Total dollar cost of special programs.

Words 5-164 are array KPROG (4, 40). The use of the first four words is the same for each special program as the first four words of each project descriptor in sector 1. The first word may be the number of a special operating millage.

Words 165-320 are array KNOPR (4, 39). Only the first three words are pertinent and are as described for sectors 1 and 2.

APEX FILE 8, 38, 48

This is a file of two sectors used by the planner's output link during a cycle. The social indicators -- unemployment, low income families, non-white population, deteriorating buildings, etc. -- are calculated and ranked for all 29 analysis areas:

LLL Words	Type	Name	
1-29	I	KPOPL(29)	Total population, number of persons.
30-58	I	NUNEM(29)	Number of unemployed.
59-87	I	RNKEM(29)	Unemployment rank--largest percent unemployment ranked as #1.
88-116	I	RNKLO(29)	Rank from 1 (highest) to 29 (lowest) percent in low income families.
117-145	I	RDETR(29)	Rank from 1 (highest) to 29 (lowest) percent deterioration.
146-174	I	RNKNW(29)	Rank from 1 (highest) to 29 (lowest) percent non-white.
175-232	R	PUNEM(29)	Percent unemployed.
233-290	R	PLOIN(29)	Percent of families with income below $3000/year. (Household type 5)
291-348	R	DETER(29)	Percent deteriorating residential structures.
349-406	R	PNONW(29)	Percent non-white population.
407-454	R	HP(6, 4)	Percent of people in each of the six wards preferring each of 4 fuel types used by AIR model. Defined (16,40) and reads 11th record.

The necessary arrays are stored sequentially in COMMON storage and the first word of the first array (KPOPL) is equivalenced to a 454 word dummy array, LLL. Reads and writes for file 8; use LLL as the single variable list as follows:

```
DEFINE FILE 8:  (2, 320, U, KXX)
```

```
INTEGER RNKEM (29), RNKLO (29), RDETR (29), RNKNW (29)

COMMON KPOPL (29), NUNEM (29), RNKEM, RNELO, RDETR, RNKNW,
      PUNEM (29), PLOIN (29), DETER (29), PNONW (29), HP (6, 4)

EQUIVALENCY (LLL (1), KPOPL (1) )

READ (8'1) LLL

    calculations

WRITE (8'1) LLL
```

APEX File 9

An array used only in SELL for distributing overflow households to adjacent analysis areas. It is a one sector (1, 320, U, KXX) file containing the array NEXT (29, 4).

Each column of the matrix represents up to four iterations of SELL in an effort to find housing for TOMM demand. For each analysis area in a column the value is the next most accessible analysis area as determined from the accessibility matrix (file 2). For example, for analysis area 1: NEXT (1, 1) contains the most accessible area to 1 in terms of travel time, NEXT (1, 2) the second most accessible, NEXT (1,3) the third most accessible, and so on.

APEX FILE 11

This file is used to store "STUFFed" records of input decisions and special output calculations.

Each STUFF is a record 16 words long which may contain a combination of real and integer elements depending on the type of STUFF. Therefore, each 320 word sector can store up to 20 STUFF records. Since the file is defined as (50, 320, U, KXX) or 50 sectors, a total of 1000 STUFFS can be made each cycle.

The entire STUFF and search mechanism, including type codes, will be covered in a separate report.

APEX File 12

This is a single 320 word file used for temporary or working storage between links at several different points in the program.

DEFINE FILE 12 (1, 320, U, KXX) or (4, 80, U, KXX)

It should not be used for other purposes between the points listed unless care is taken to read and restore the number of words indicated at the beginning of the sector.

Record 1

A. In the READR subroutine of the INPUT link the following arrays are accumulated as BS, RM, and AF cards are input -- a total of 72 words -- and saved until after the calculation of voter turnout at the beginning of BDMIL.

Words	Name	
1-10	TTNSM(5)	Total non-school millage
11-18	TTSCM(4)	Total school millage
19-28	TTNED(5)	Total non-school dollars and bond proposals
29-36	TTSCD(4)	Total school bond proposals--dollars
37-54	RMRCY(9)	Requested millage rate, current year
55-62	RAF(4)	New assessment factors, if any
63-72	SCAMP(5)	Sum of campaign contribution, but jurisdiction.

Record 2

B. Saved from the end of BDMIL to CPI:

NWORK(39) - numbers of bond millages passed iu BDMIL
NPASS - the number of useful elements stored in NWORK

C. Records 2, 3, and 4 word 81-254 (during TOMM only).

ERRM(29, 3) - Saved during TOMM for comparisons at end.

APEX File 15

Words	Type	Name	
1-16	R	CMPDL (8)	Total campaign dollars <u>for</u> each of 8 politicians
17-32	R	OPDOL (8)	Total campaign dollars <u>opposed</u> each of 8 politicians
33-72	R	DOL (20)	Campaign expenditures (algebraic sum) for each of 20 bond or millage proposals
73	I	NBS	Number of bond or millage proposals this cycle.
74-93	I	LBS (20)	Numbers of proposals.
94		filler	
95-104	I	PROMO (5)	Total promotional dollars for each of 5 jurisdictions

Words	Type	Name	
105-114	R	PCSTD (5)	For CEM - Percent standard performance of candidates by jurisdiction
115-144	I	MVMVT (6,5)	For CEM - Numbers of voters by ward
145-164	I	ISFLG (20)	Numbers of issues to be printed in newspaper when special program has only one year to go. Stored as (issue # + 1000 x jurisdiction).

APEX Files 16, 36 and 46

These are key files in APEX. For convenience only file 16 will be described.

DEFINE FILE 16 (8, 40, U, KXX)

Only the first 3 records are currently being used. The first represents the first 40 words of COMMON storage allocation in all APEX links. Since these words contain some pointers and indicators for STUFF and for searching the STUFFed file (11) as well as other common information, they must be carefully saved and restored if this upper common area is ever to be used for other purposes, and must also be saved at various points for restarts.

The use of the forty words is as follows:

Word	Type	Name	
1	I	NCYCL	Cycle number. Set to 0 in file 36. Incremented by i in CYINJ if going on to new cycle.
2-4	I	KDATE(3)	Date of run. Set in CYIN2 for use by EJECT in output links. First word is month, second is day, third is year.
5	I	LDP	The first seven bits (0-6) are set by ISKT in CYIN2 to indicate those realtors playing in the CYCLE. Bits 7-13 are set for playing industrialists.
6	I	LRAND	Initial 5 digit random number initial value. Set in CYIN1 for RANDU.
7-10	I	MIN(4)	These sixteen words are used for STUFFing and searching. The first four words are always integer, sometimes referred to as the first four elements of MIN(16) but also referred to by specific names in some links. The next twelve words may be either integer or real,

Word	Type	Name	
			depending on the STUFF record. They are usually referred to as ZIN(6). MIN an ZIN are equivalenced as follows: EQUIVALENCE (MIN(6), ZON(1)) Other specific variable names are frequently equivalenced to specific elements of MIN or ZON. See STUFF description for more details.
23	I	PACKI	The number indicating which of the 20 sixteen word STUFF records in a sector was the last one STUFFed.
24	I	PACK2	Which sector of the 50 in file 11 last one: STUFFed.

Word	Type	Name	Description
25	I	IMPRK	Index of "found" STUFF from subroutine QERCH.
26-29	I	INFO(4)	See comments in QERCH.
30	I	LPK1	QERCH analogues of PACK1 and PACK2.
31	I	LPK2	
32	I	KC44	Player number (developer, industrialist, politician, etc.). Used by EJECT for page headings.
33	I	IROLE	Flag for EJECT to indicate output heading. 1 = Realtor 2 = Politician 3 = Educators 4 = General Summary 5 = APCO 6 = Industrialist 7 = Planners
34	I	IFUSE	Flag set in output subroutines, primarily miscellaneous output, to decide if number of lines printed requires skipping to the top of a new page. Used only in output links.
35	I	NPAGE	Used by EJECT subroutine in output links to keep running page number.
36	I	LSW	Switch set for print suppression of individual players in output links by data switch setting. = 1 Print = 2 Don't print
37	I	NOIO	Master print suppression flag. Set in INPUT link to suppress all printing and most typewriter messages.
38	I	NTEAM	Team number.
39-40			Unused.
Record #2			
1-28	R	CASH(14)	Running cash balance in account of each of the realtors (1-7) and industrialists (8-14).
29-40			Not used at present.

Word	Type	Name	Description
Record #3			
1-20	R	TNAME(10)	Team name, EBCDIC 10A4.
21-40			Unused.

APEX Files 19, 39 and 49

All files are defined as (3, 320, U, KXX). They contain capital plant values, indices, ranks, population equivalents.

The description of this file may be somewhat confusing without a programming note. In 1130 FORTRAN under the Disk Monitor System, the maximum record size which may be defined is 320 words (one disk sector). However, the system will read or write more than one record (sector) with a single disk read or write statement -- it will continue reading or writing until all elements of the "list" specified are accounted for. For example, assume the file definition above and the following FORTRAN statements:

```
COMMON CPTOT (36, 9)

READ (19'1) CPTOT
```

The above READ statement would cause all 648 words (2+ sectors) of CPTOT to be read from the file.

The following is a summary of the file 19 contents; a detailed variable description follows the summary:

CONSECUTIVE WORDS	VARIABLE TYPE	NAME & DIMENSIONS	SECTOR NUMBER	SECTOR WORD	SECTOR ELEMENTS
1-648	R	CPTOT (36, 9)	-	-	-

The 9 columns of the above array are as follows:

1-72	R	(1) Streets: CPTOT (1, 1) to CPTOT (36, 1)	1	1-72	1-36
73-144	R	(2) Sewers	1	73-144	37-72
145-216	R	(3) Water	1	145-216	73-108
217-288	R	(4) Parks/ Recreation	1	217-288	109-144
289-360	R	(5) Miscellaneous	1 2	289-320 1-40	145-160 1-20

CONSECUTIVE WORDS	VARIABLE TYPE	NAME & DIMENSIONS	SECTOR NUMBER	SECTOR WORD	SECTOR ELEMENTS
361-432	R	(6) Total Government (sum of first 5 columns)	2	41-112	21-56
433-504	R	(7) Total Elementary School	2	113-184	57-92
505-576	R	(8) Total High School	2	185-256	93-128
577-648	R	(9) Total School (sum of 7 & 8)	2 3	257-320 1-8	129-160 1-4
649-792	R	CPI(36, 2)	3	9-152	5-76
793-864	R	POPEQ(36)	3	153-224	77-112
865-866	R	CONVF	3	225-226	113
867-938	I	KRPNK(36, 2)	3	227-298	227-298
939-940	R	COCPT	3		
941-942	R	COCPI	3		
943-944	R	COPOP	3		
945-960	Unused at present.				

Variable Descriptions for Files 19, 39, and 49:

CPTOT (36, 9) Total capital plant value in each of 9 categories (see below) for each of 36 geographic areas. In actual practice the program adds cost per year from file 6 as each project is processed in the GRUPE link--SUBROUTINE CPADD. Once a project is completed no appreciation or depreciation on value takes place.

The first 29 rows correspond to the 29 APEX analysis areas.

Row 30	Ward 1	Total
Row 31	Ward 2	Total
Row 32	Ward 3	Total

Row 33	Ward 4 or Jurisdiction 2	Total
Row 34	Ward 5 or Jurisdiction 3	Total
Row 35	Ward 6 or Jurisdiction 4	Total
Row 36	Jurisdiction 1, Central City	Total

CONVF — Conversion factor for converting number of employees to population equivalents, POPEQ. It is presently (3-31-70) equal to 1.25. Population equivalents are calculated as:

Total Employees in all categories / CONVF

CPI (36, 2) — Capital Plant indices for each of the 36 areas described above. Column one is government (politicians) and is column 6 of CPTOT/POPEQ. Column two is <u>elementary school</u> capital plant index:

$$\text{CPI (I,2)} = \frac{\text{CPTOT (I, 7)}}{\text{Number of elementary pupils}}$$

KPRNK (36, 2) — The rank of the corresponding capital plant indices--the largest value is ranked 1.

COCPT — Total dollar value of county capital plant (miscellaneous).

COCPI — County capital plant index.

COPOP — County population equivalents.

<u>APEX Files 20 (30 and 40)</u>

This four sector file contains records of the seven game realtor's and 7 industrialists' loans and taxes as well as other arrays and constants necessary in link TAXLN for loan and tax processing.

In most cases these files will be defined as follows:

```
DEFINE FILE 20 (32, 40, U, KXX)
```

Each of the first 14 records in these files contains variables pertaining to one of the game realtors (records 1-7) or industrialists (records 8-14). The variables in each of the records are as follows:

Word	Type	Name	Description
1-2	R	OLDCH	Cash balance at end of preceding cycle for the player.
3-4	R	PROPV	Total property value for the player at end of preceding cycle.
5-6	R	TTXPD	Total taxes paid current cycle for the player.
7-16	R	TAX(5)	Taxes for the player for each jurisdiction computed at end of preceding cycle or total unpaid delinquent taxes during current cycle after payments are processed.
17-26	R	TDELT(5)	Total delinquent taxes paid during current cycle in each jurisdiction by the player due to land confiscation.
27-31	I	DELTX(5)	Counters of number of times taxes were not paid in full for the player, in each of the 5 jurisdictions. If unpaid for two consecutive cycles, land is confiscated for the payment.
32	I	LSF	Financial standing of the player from 1 (highest) to 3 (lowest). Presently based only on loan underpayments.
33-34	R	TLDBT	Total outstanding loan balance for player.
35-36	R	TNEWL	Total dollar value of new loans current cycle for the player.
37-38	R	TLPAY	Total loan payments made current cycle by the player.
39-40	R	FLIMT	Loan limit factor for the player--the proportion of his net worth which he can borrow.

Player No.	ENTFL Record No.
1-8	1
9-14	2

The 15th and part of the 16th record of these files contains variables not specific to any one player but which are used for calculations relating to all players:

Word	Type	Name	Description
1	I	NJ	Count of number of loans in the loan table--which follows--or pointer to last used 10 word block.
2	I	NEXT	The last loan number assigned. This is, in effect, a count of the total number of loans processed during a game since loan numbers are not re-used once a loan is paid off. It does not necessarily represent the number of loans in the table.
3-14	I	IRATE(4,3)	Interest rates at which new loans are granted. The four rows represent the interest rate based on the time required to pay back the loan. IRATE (1,I) - 1 or 2 years IRATE (2,I) - 3 to 5 years IRATE (3,I) - 6 to 10 years IRATE (4,I) - over 10 years The three columns are for the 3 possible credit standings (LSF).
15-16	R	XINT	Interest rate paid on cash savings for players. Added at beginning of cycle to cash from preceding cycle.
17-18	R	TINT	Interest rate charged and added to unpaid taxes.
19-20	R	PENFC	Penalty factor against loan underpayments.
21-22	R	REDOX	Factor for reducing FLIMT if loan underpayment, i.e., reducing loan limit.
23-26	I	LUC(14)	Loan underpayment counters for each player. If delinquent in loan payments 3 consecutive cycles, payment is forced from cash balance.
37-40 (of 15) 1-24 (of 16)	R	TNETW(14)	Net worth at end of cycle for each player. Used primarily from file 40 for comparison with current value.

The first sixteen 40 word records occupy 640 (or the first 2 sectors) of the 1280 words in the four sectors of these files.

The last 640 words of this file are the table of loans outstanding for the realtors. Each loan requires a block of ten words so that a maximum of 64 loans may be outstanding at any one time. (When the table becomes full during play the table is packed by eliminating all loans with a zero balance.)

The first four elements of each 10 word loan record are integers and the last three (6 words) are real elements. For this reason the array is identified by two names and equivalanced as follows:

```
DIMENSION XJ (5, 64)

COMMON J (10, 64)

EQUIVALANCE (J (2), XJ (1) )
```

The entire loan table may be read with the following statement:

```
READ (20'17)XJ
```

The elements of each loan record are as follows:

Word		
1	J(1,I)	Number of the land developer or industrialist receiving the loan. Integer from 1-14.
2	J(2,I)	Loan number. Assigned by the program from 1-N, printed on the output, and used to identify loan payments.
3	J(3,I)	Interest rate. Note: this is an integer value, e.g. 81 = 8.1%
4	J(4,I)	Years remaining in loan life.
5-6	XJ(3,I)	Dollar amount of loan or, after the first year, the unpaid balance.
7-8	XJ(4,I)	Dollar penalty for underpayment.
9-10	XJ(5,I)	Minimum payment next cycle.

APEX File 21

A single sector (1, 320, U, KXX) file containing constants used in land transactions in LAND and SELL. All values are preset and are not initialized or saved from preceding cycle.

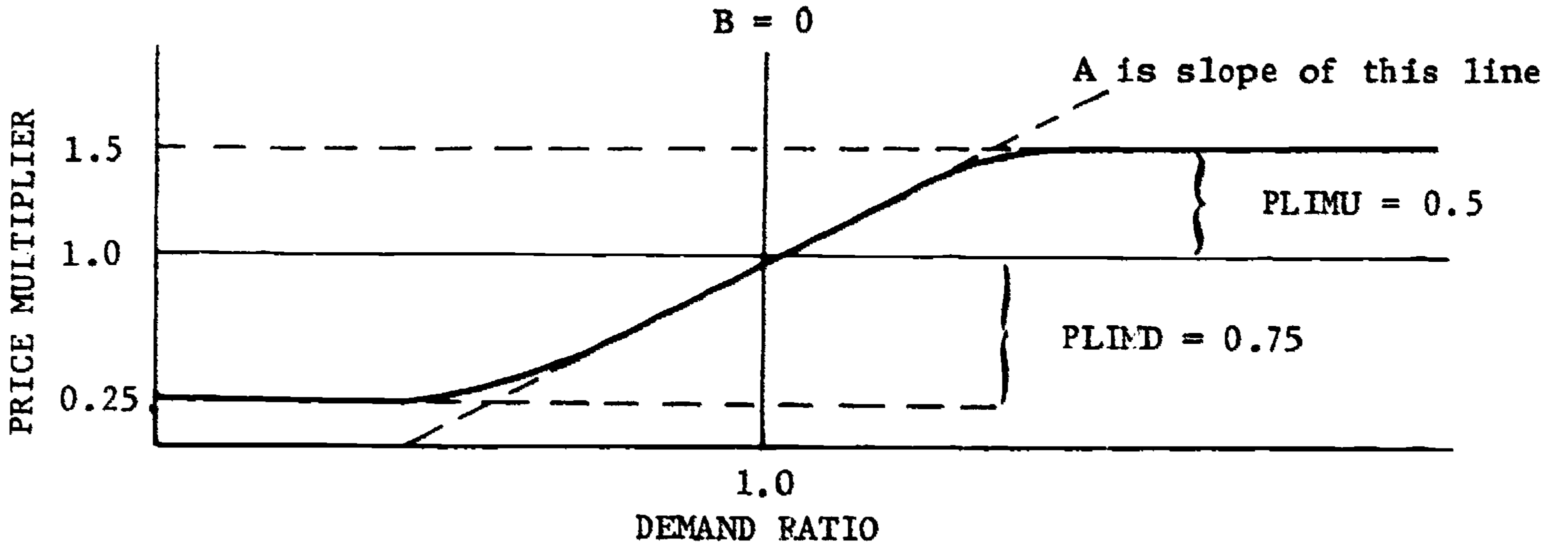
Word	Type	Name	Description
1-10	R	UDENV(5)	Spread vector used with "U" variable (see File 1 description) to get increasing land use density.
11-22	R	DTS(6)	Proportion of total property value which is demolition cost when property goes from developed to vacant. Values for the six vacant categories.
23-24	R	PNF	Public N Factor - a proportion of public land which becomes right-of-way when public land is developed.
25-26	R	PDFS	Dollar per acre cost of public land demolition.
27-48 Should be inflated each cycle.	R	RCDEV(11)	Residential-Commercial Development cost. Dollars per unit (or acre, if commercial) to construct or develop property for the 11 developed categories.
49-70	R	EVEC(11)	Raw land improvement cost (clearing, streets, curbing, sidewalks, etc.) for the same 11 categories as RCDEV. May be modified by E factor in File 1 which is presently at 1.0.
71-72	R	R2R2V	Proportion of multiple zoning type 5 (MDR(10) family or housing units which can be displaced by commercial demand in SELL. Presently set at 10%.
72-122	R	HPREF(5,5)	Percent of each household type (2nd subscript) that prefers each of the five developed residential housing types (1st subscript). Used to allocate demand after TOMM.

	HOUSING TYPE					
H.H. Type	R1	R2	R3	M1	M2	
1	50%	30%	0	20%	0	100%
2	20%	20%	20%	20%	20%	100%
3	10%	30%	20%	25%	15%	100%
4	0	20%	40%	10%	30%	100%
5	0	0	40%	0	60%	100%

Word	Type	Name	Description
123-172	R	ZPREF(5,5)	Percentage of each type of household living in each housing type. Used to allocate displaced households when units go from developed to vacant.

	Household Type					
	H_1	H_2	H_3	H_4	H_5	
R_1	62.5%	25.0%	12.5%	0	0	100
R_2	30.0	20.0	30.0	20.0	0	100
R_3	0	16.7	16.7	33.3	33.3	100
M_1	26.7	26.7	40.0	13.4	0	100
M_2	0	16.0	12.0	24.0	48.0	100

Word	Type	Name	Description
173-174	R	PLIMU	Related to price adjustment for residential property. See following discussion.
175-176	R	PLIMD	
177-178	R	PA	
179-180	R	PB	
181-182	R	CLIMU	Analogous to preceding four variables for commercial prices.
183-184	R	CLIMD	
185-186	R	CA	
187-188	R	CB	
189-320			Unused at present.



Price adjustments each cycle are computed as a function of the ratio of demand for a particular type of property within an analysis area to the demand for the same type over the entire APEX area. The function generates a multiplier which is applied to the corresponding price and which is calculated from an S curve algorithm. In other words, if demand is greater in a specific area than overall, a multiplier such as 1.03 or 3% increase would be used. If demand were lower the multiplier might be 0.96, or 4% decrease.

Mathematically the multiplier is calculated as follows:

If $T \geq 1.0$, then $X = T-1$ and $M = S * LIMU + 1$

If $T < 1.0$, then $X = \frac{(T-1)}{T}$ and $M = S * LIMD + 1$

Where T = demand ratio

M = price multiplier

$$S = \frac{2}{(e^{-AX + B} + 1)} - 1$$

In simple terms, LIMU = maximum proportion of <u>increase</u> in price, i.e., how much greater than <u>1.0</u> can the multiplier be.

LIMD = maximum proportion of <u>decrease</u> in price, i.e., how much less than <u>1.0</u> can the multiplier be.

A = maximum slope of S curve, which determines how rapidly the multiplier increases or decreases with a change in demand ratio.

B = horizontal-axis intercept. It is set at 0.0 which causes a ratio of 1.0 to produce a multiplier of of 1.0. If $B > 0$ the axis would be moved to the right, meaning that it would require a demand ratio greater than 1.0 to keep the price from going down as might be the case in a depression or, if $B < 0$ the axis would be moved to the left which would mean that a demand less than 1.0 could still cause an increase in price.

<u>APEX Files 22, 32, 42</u>

Counters of families and employment.

DEFINE File 22 (22, 29, U, KXX)

Alternately (2, 320, U, KXX)

Sector	Record	Words	Type	Name	Description
1	1-5	1-145	I	PDM(29,5)	Number of families in each of 5 types in each AA.

1	6-8	146-232	I	NJGTC(29,3)	Number of employees in each of the three endogeneous employment categories in each AA.
					Col. 1 is local commercial;
					Col. 2 is regional commercial;
					Col. 3 is local industrial.

1	9-11	233-319	I	JDELT(29,3)	(Used only in file 22, not initialized.) Change in employment from TOMM -- used in SELL.
1	320	320	Unused		
2		1-58	R	EBUR(29)	Number of exogenous bureaucratic employees in each analysis area.
2		59-116	R	EIND(29)	Exogenous industrial employees in each AA.
	(Note:	EBUR and EIND cannot be read correctly as 12th record of a 22 x 29 file.)			
2	16	117-118	R	TEXMP	Total Exogeneous Employment after GROW.
		119-128	R	HFCT(5)	Persons per household by type
		129-137	I	EXBIT(9)	Same as 16-24 of record 56 of File 5. (Not in file 32.)

APEX Files 23, 33 and 43

Simulated School Board data (more data, especially the financial data) is stored in File 4 q.v.

DEFINE FILE 23 (20, 32, U, KXX) - 2 sectors

File 23 has records of length 32 words. The first: 4 records correspond to the school boards in the central city, suburb, and the two townships, respectively:

Records 1, 2, 3 or 4 - Sector 1

Word (in record)	Type	Name	Description
1-2	R	ARB1	
3-4	R	ARB2	
5-6	R	MCH	Minimum allowable capital plant/pupil
7-8	R	SAPE	State aid/elementary pupil
9-10	R	SAPH	State aid/high school student
11-12	R	SCE	Standard capital plant/elementary pupil
13-14	R	SRCH	Standard capital plant/high school pupil
15-16	R	STE	Standard # of teachers/elementary pupil
17-18	R	STH	Standard # of teachers/high school pupil
19-20	R	STSE	Standard elementary teacher salary
21-22	R	STSH	Standard high school teacher salary
23-24	R	GOVT	Total government aid
25-26	R	OTHER	Other expenses
27-28	R	TTE	Total elementary school salaries
29-30	R	TTH	Total high school salaries
31-32	R	TOE	Total Operating expenses

Records 5 - 10 in Sector 1
11 - 20 in Sector 2

Word (in sector)	Type	Name	Description
Sector 1			

129-163	I	NEP(29)
		NHP(6)
			Number of elementary students in AA's and high school students in wards.
			(This 29 and 6 format is followed throughout).

164-233	R	CPE(29)
		CPH(6)
			Capital plant in elementary or high schools

234-303	R	DCE(29)
		DCH(6)
			Yearly depreciation in above

304-320
1-53 (Sector 2)
	R	PIE(29)
		PIH(6)
			Proposed investments this cycle in CPE, CPH

Sector 2			

54-123	R	RCE(29)
		RCH(6)
			Ratio of capital plant per pupil
			RCE = CPE/NEP

124-158	I	ICE(29)
		ICH(6)
			Integer ranking of above

159-193	I	NET(29)
		NHT(6)
			Number of elementary and high school teachers

194-263	R	RTE(29)
		RTH(6)
			Teacher-pupil ratios: RTE = NET/NEP

264-288	I	ITE(29)
		ITH(6)
			Integer rank of above

289-320	Spare space		

APEX File 25 - Control System Names - 2 sectors

There are no corresponding save and initialization files.

DEFINE FILE 25 (32, 20, U, KXX)

Word	Type	Name	Description
1-20	R	CNAME (13)	NAME OF CONTROL SYSTEM (5 to 2 words) Up to 50 characters.

APEX File 51 - Process File - 9 sectors

Files #61 to initialize and #71 to save.

Note: The record number of each process is indexed in the Industry File (#58). That is, process x for industry I is in record number IRECD + x - 1, e.g., process 2 for industry 4 would be in IRECD + 2 - 1 and if IRECD = 10, process 2 is in record 11 of the process file.

DEFINE FILE 51 (63, 45, U, KXX)

Word	Element	Type	Name	Description
1-10	1	R	REMIS(5)	Rate of contaminant emission current (in lb/hr).
11	11	I	RSMOK	Rate of smoke emitted (Ringelmann Number)
12	12	I	RODOR	Rate of odor emitted (Stinkelmann Number)
13-22	7	R	GEMIS(5)	Potential (uncontrolled) rates.
23	23	I	GSMOK	Potential Ringelmann.
24	24	I	GODOR	Ditto Stinkelmann.
25-27	25	I	ICS(3)	Control system applicable. Negative: not operating, not installed. .LT. 100: operating .GT. 100: installed, not operating.
28	28	I	ITYPE	Odor type: 1 = putrid, 2 = acrid, 3 = sulphurous, 4 = pungent.
29-30	15	R	RPROC	Process rate, in input mat./day, except for fuel, quid in lb/unit output.
31-34	16	R	MUNIT(2)	Name of units in RPROC.
35-36	18	R	OPER	Operating period, as % of day.
37-44	19	R	PNAME(4)	Process Name. Stored 5 char./2 words
45	45	I	IRATE	Conversion factor for RPROC. RPROC * IRATE = production rate in 1000 lb./hr.

APEX File 52 - Air Pollution Laws and APCO Data - one sector.

File #62 to initialize, File #72 to save.

Note: Each record 10, 11, 12 contains figures for different levels of punitive action, i.e., record 10 contains APCO recommendations, record 11, Rules with small fines, and record 12, laws with large fines.

```
DEFINE EILE 52 (1, 312, U, KXY)

        or

DEFINE FILE 52 (13, 24, U, KXY)
```

Word	Type	Name	Description
1-10	R	PRIOP(5)	Price of operation of 5 types of monitoring stations.
11-12		Unused	
13-84	R	PRICE(6,6)	Upgrade costs of monitoring stations: 0-5 x 0-5
85-86	R	COSTI	Cost of plant inspection.
87-88	R	COSTM	Cost of emission measurement.
89-100	R	CB(6)	County Budget Items.
101-102	R	CTOT	Total County Allocation.
103-108	R	FFUND(3)	Federal fund items for 3 years.
109-144		Unused	
Record 7			
145-173	I	OPMON(29)	Monitoring stations in operation.
174-202	I	MONS(29)	Monitoring stations in existence.
203-212	R	APASK(5)	Air pollution control officer's requests for contaminant level used to compute the number of days above some level.
213-216		Unused	
Record 10-12			Record 10 = Laws, 11 = Rules, 12 = Recommendations

Word	Type	Name	Description	
1-10	R	PDSHR(5)	Law concerning pounds per hour limitation. $I_3 = 1$.	
11-20	R	PDSPU(5)	Law concerning pounds per processing unit limitation. $I_3 = 2$.	
21	I	SMOKE	Law concerning smoke limitation	$I_3 = 3$
22	I	ODOR	Law concerning odor limitation	
23-24	R	FINE	Fine level for violation.	

Note: Records 10-12 use up words 217-288

289-302	R	AMT(7)	Amounts gamed industrialists have been fined.
303-312	R	COEFF(5)	Coefficients for complaint generator.

APEX File 53 - Control Systems - 5 sectors

DEFINE FILE 53 (80, 20, U, KXX)

For gamed industry, this file holds the percentages for reducing emission rates due to installation of the named system. For simulated industries this file holds the present control levels and the ideal control levels. There are up to three records for any gamed industry process and two records for each simulated industry.

Word	Type	Name	Description
1-10	R	PCT(5)	Percent reduction in emissions - particulates, SO_2, CO, NO_x, HC
11-12	I	LSMOD(2)	Reduction in smoke and odor.
13-14	R	ORIGD	Original cost of installation.
15-16	R	OPERD	Cost of annual operation.
17	I	NAME	RECORD NUMBER OR NAME IN FILE 25.

APEX FILE #54 - INDUSTRIALISTS RECORDS

File 64 to initialize, File 74 to save. Definition for it is (7, 320, U, KXX). The Industry number is the record number.

Word	Type	Name	Description
1-3	I	NEMP(3)	Number of employees last 3 cycles.
4	I	IAA	Analysis area of plant.
5-6	R	TEQBG	Total value Equipment, Building and Air Pollution Equipment.
7	I	IOP	1 if available for play. 0 if ind. not available for play (no data).
8	I	MFUEL	Fuel used this cycle.
9-10	R	CINCP	Capacity increment due to invest in equipment (Units/$)
11-12	R	CPMIN	Minimum expenditure on equipment.
13-14	R	CINCB	Potential capacity increase due to building increase (units/sq. ft.).
15-16	R	BINVM	Minimum building investment increment. Allowable.
17-18	R	CLAND	Building capacity change due to land increase (sq. ft./sq. ft.).
19-20	R	SINCB	Increment in sq. ft. building/$ expenditure on building.
21-22	R	EMPDR	Employment/Production ratio (employees/unit).
23-32	I	IPCTS(10)	% changes in sales standard MKTSL.
33-42	I	IPCTP(10)	% changes in PRMKT
43-52	I	IPCTC(10)	% changes in product-costs.
53-54	R	MCAP	Maximum capacity of firm next cycle.
55-60	R	NPL(3)	Production Level.
61-66	R	NESTS(3)	Estimated Sales (unit).

Word	Type	Name	Description
67-72	R	NACTS(3)	Actual sales (unit).
73-78	R	PRICE(3)	Price set by industrial. ($)
79-84	R	PRMKT(3)	Market price. ($)
85-90	R	ESTS(3)	Estimated sales. ($)
91-96	R	ACTS(3)	Actual sales. ($)
97-102	R	INSLD(3)	Inventory sold (units).
103-108	R	IVNT(3)	Inventory last 3 cycles. (units).
109-110	R	ICOVR	Inventory carryover. (units).
111-112	R	AVEWG	Average yearly wage/employee.
113-114	R	UMC	Unit material cost. ($/unit)
115-116	R	UAC	Unit administrative cost. ($/unit)
117-118	R	CCIVN	Total inventory carrying cost.
119-120	R	CPINV	Current investment in equipment (non-pollution). ($)
121-122	R	CPLIM	Equipment investment limit (maximum). ($)
123-124	R	BINVE	Current investment in building. ($)
125-126	R	BDLIM	Building investment limit. ($)
127-128	R	DPREX	Depreciation expenditure (deterioration).
129-130	R	SQFTB	Sq. Ft. of building.
131-133	I	ICAP(3)	% capacity used.
134	I	ISHP	Flag = 1 to indicate purchase of control system data this cycle.
135-136	R	TLC	Total labor cost.
137-138	R	TMC	Total material cost.
139-140	R	TFC	Total fuel cost.

Word	Type	Name	Description
141-142	R	TAC	Total administrative cost.
143-144	R	TPCOP	Total pollution control operating expenses paid.
145-146	R	APFPD	Air pollution fines paid.
147-148	R	EMMEX	Emission measurement expenditure this cycle.
149-150	R	CAMEX	Campaign contributions.
151-152	R	DPAL	Depreciation allowance made.
153-154	R	PROG	Gross profits.
155-156	R	FEDS	Federal-state taxes.
157-158	R	PRON	Net profit.
159-160	R	CEXEQ	Capital expenditure on equipment (non-pollution).
161-162	R	CEXLD	Capital expenditure on land.
163-164	R	CEXBG	Capital expenditure on building.
165-166	R	CEXAP	Capital expenditure on air pollution.
167-178	R	CFUEL(6)	Unit cost of each of six fuels. ($/unit)
179-180	R	TLPAY	Total loan payments.
181-182	R	TTXPD	Total tax payments (local).
183-188	I	IFUEL(6)	Fuels available for use (1 = available, 0 = not available).
189-190	R	UNIT	Name of product unit.
191-192	R	TAPCV	Total air pollution control value.
193-194	R	DVLPD	Acres developed land.
195-196	R	VACNT	Acres vacant land.
197-198	R	MKTSL	Market sales standard.
199-204	R	MALC(3)	Capacity last 3 cycles.

Word	Type	Name	Description
205-206	R	UICC	Unit inventory carrying cost.
207	I	NPROC	Number of sulprocesses.
208-222	I	NAME(15)	Industry name.
223-224	R	TOTC	Total cost of production
225-226	R	TCASH	Total cash available
227-228	R	TCAP	Total capital expenditures.
229-230	R	AVAL	Total land value.
231-232	R	ALLOW	Maximum depreciation allowance for next cycle.
233-240			Unused

APEX FILE #58 - MASTER LIST OF INDUSTRIES
FILE #68 - to initialize, File 78 to save. This File contains 4 sectors. This file is the locator file for File 51.

Note: The record number corresponds to the Industry number.

FILE 58

DEFINE FILE 58 (52, 24, U, KXX)

Words	Type	Name	Description
1	I	ICP	Operational Status of Industry
2	I	IA	Analysis Area
3-4	R	RPROD	Output of Firm
5-6	R	PRODO	Production Capacity of Firm
7	I	IRECD	Record number of the first process for gamed industries, except the gray iron foundry, IRECD is the fuel process. For simulated industries, it is the only process.
8	I	NPROC	Number of production processes used by the Firm. For simulated industries, NPROC = 0.
9-12	R	NUNIT(2)	Name of appropriate production unit per unit of time.

Word	Type	Name	Description
13-24	R	FNAME(6)	Firm name - 30 characters (5 per 2 words)

SECTION 9

29 APEX ANALYSIS AREAS

● 29 APEX ANALYSIS AREAS

TOWNSHIP 1

Areas 23, 24, 25, 26, 27 and 28

TOWNSHIP 2

Areas 14, 15, 16, 20, 21, 22 and 29

SUBURB

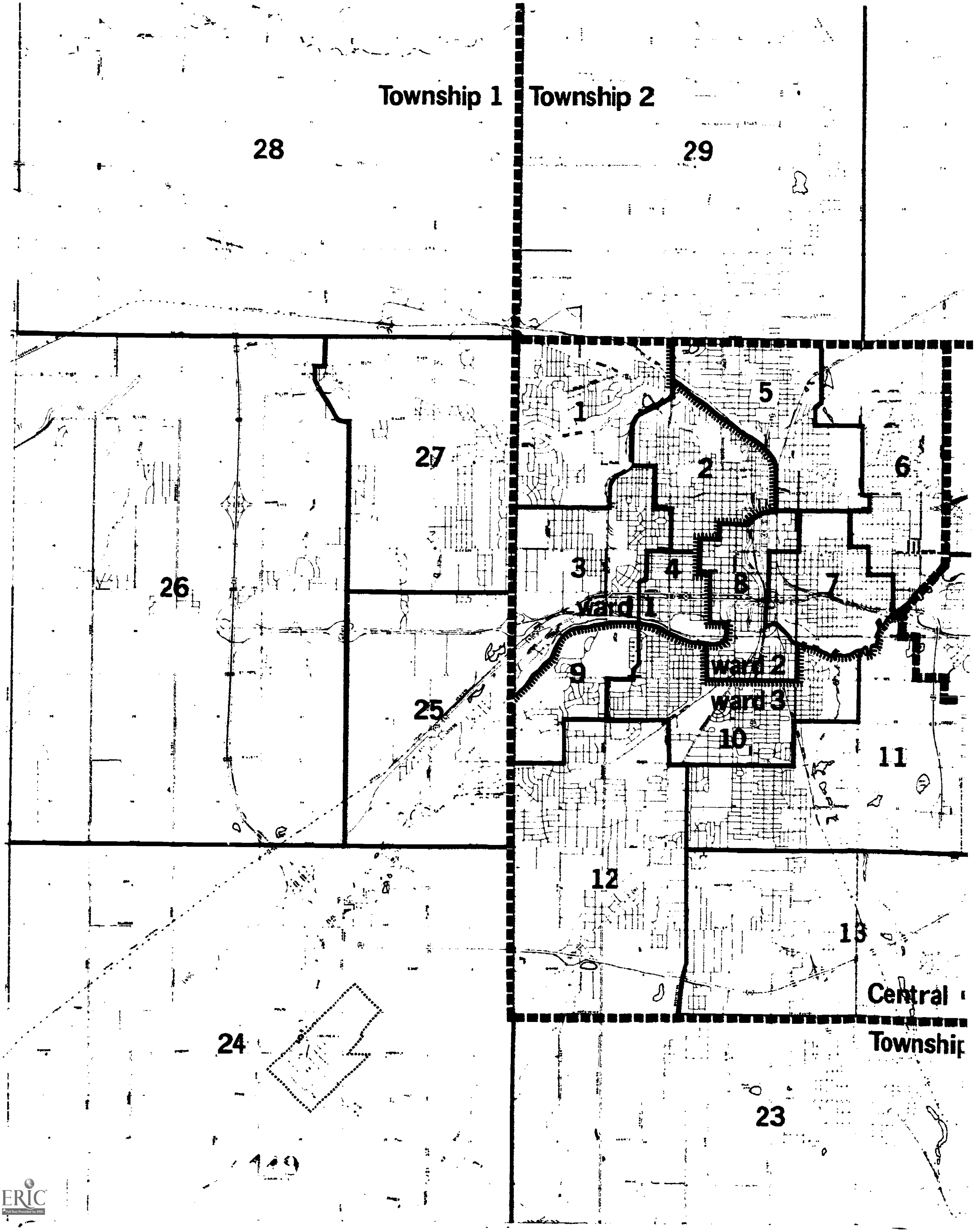
Areas 17, 18 and 19

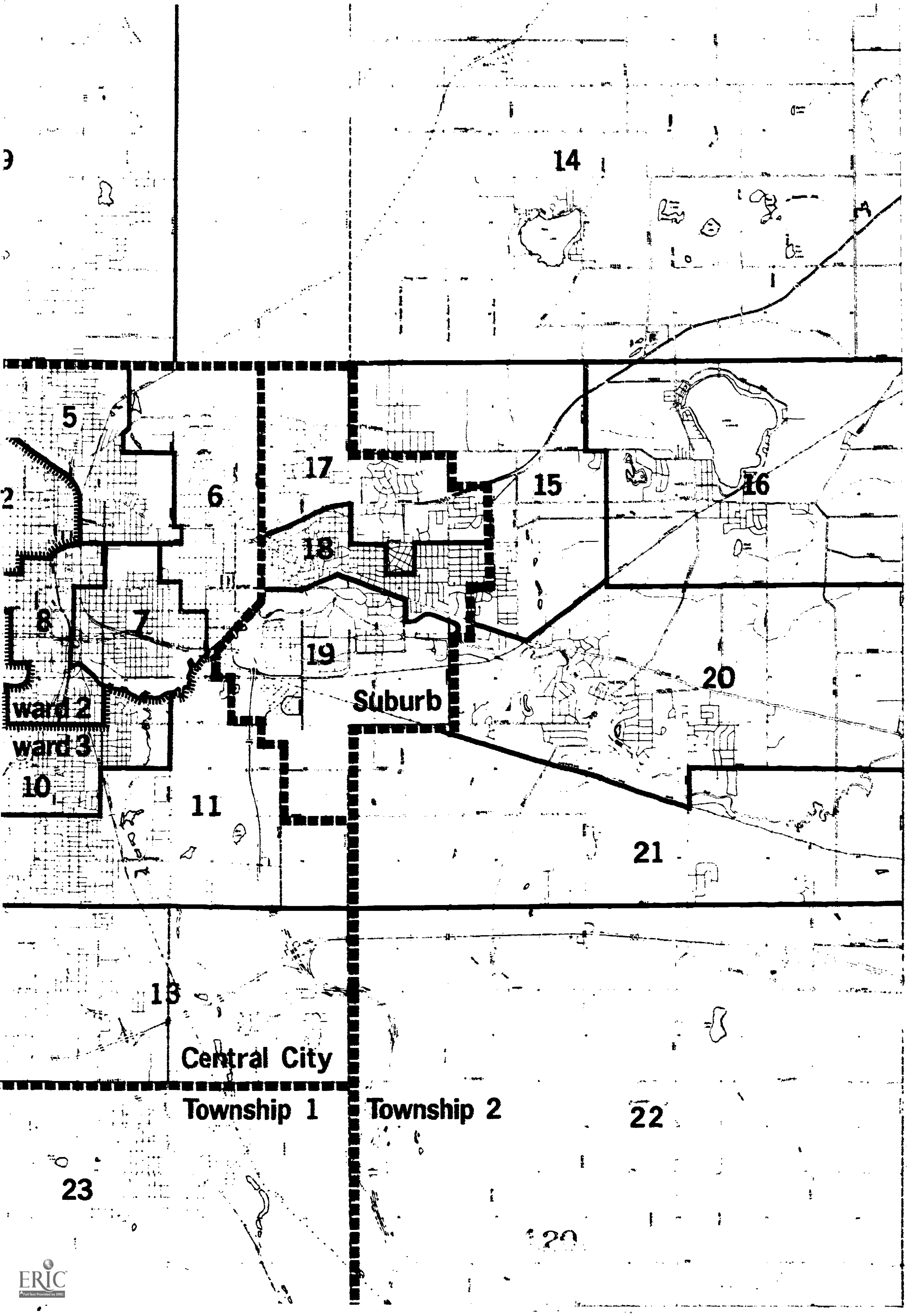
CENTRAL CITY

Ward One: Areas 1, 2, 3, and 4

Ward Two: Areas 5, 6, 7 and 8

Ward Three: Areas 9, 10, 11, 12 and 13


Township 1
Township 2
28
29
1
5
2
27
6
3
4
8
7
26
ward 1
ward 2
9
ward 3
25
10
11
12
13
Central
Townshi
24
23


9
14
5
2
6
17
15
16
18
8
7
19
20
Suburb
ward 2
ward 3
10
11
21
13
Central City
Township 1
Township 2
22
23
20

UNITED STATES · ENVIRONMENTAL PROTECTION AGENCY