USER MANUAL

DANUBE EMISSIONS MANAGEMENT DECISION SUPPORT SYSTEM (DEMDESS)

JULY 1992

WATER AND SANITATION
FOR HEALTH PROJECT

USAID
EUROPE BUREAU

DANUBE EMISSIONS MANAGEMENT DECISION SUPPORT SYSTEM (DEMDESS)

USER MANUAL

Prepared for the Europe Bureau,
U.S. Agency for International Development
under WASH Task No. 271

by

Max Clark
Tim Bondelid
Tarik Pekin

September 1992

Water and Sanitation for Health Project
Contract No. DPE-5973-Z-00-8081-00, Project No. 936-5973
is sponsored by the Office of Health, Bureau for Research and Development
U.S. Agency for International Development
Washington, DC 20523

CONTENTS

APPENDICES

ACRONYMS

A.I.D.	Agency for International Development
ANNCAP	annual capital cost
BOD_5	biochemical oxygen demand (5-day, 20° C)
cmd	cubic meters per day
COD	chemical oxygen demand
CSFR	Czech and Slovak Federal Republic
DEMDESS	Danube Emissions Management Decision Support System
DO	dissolved oxygen
EBRD	European Bank for Reconstruction and Development
EC	European Community
EPA	Environmental Protection Agency (U.S.)
EUR	Bureau for Europe
leva	currency in Bulgaria; $1 = 20 leva (1992)
O&M	operation and maintenance
PAL	Paradox® Application Language
PR01	primary treatment (mechanical sedimentation)
PR02	enhanced primary treatment (mechanical sedimentation assisted by chemical coagulation)
SAS01	secondary treatment (single-stage biologically waste-activated sludge)
SAS02	enhanced secondary treatment (two-stage biologically waste-activated sludge)
TSS	total suspended solids
USAID	United States Agency for International Development
WASH	Water and Sanitation for Health Project

Chapter 1

INTRODUCTION TO DEMDESS

1.1 Background

DEMDESS (the Danube Emissions Management Decision Support System) is a computer-based system of data bases and applications developed by A.I.D.'s WASH (Water and Sanitation for Health) Project in cooperation with four Danube riparian countries: Bulgaria, the Czech and Slovak Federal Republic (CSFR), Hungary, and Romania. DEMDESS is a product of the DEMDESP (Danube Emissions Management Decision Support Project), designated as WASH Task 271, which began in September 1991. The WASH Project, which is sponsored by A.I.D.'sR&D/Office of Health, received funding from the Bureau for Europe/Development Resources/Environment mission to carry out the assignment.

The conceptual design and scope of DEMDESS were discussed and agreed upon by the four countries in a workshop held in Visegrad, Hungary, in December 1991. This was followed by a concept paper in January 1992, which outlined the data requirements and scope of a DEMDESS "target system" in accordance with the discussion at the Visegrad workshop. Subsequently, the WASH team assisted each country in transferring data into DEMDESS, and in developing applications of DEMDESS in accordance with the scope and level of detail of the available data.

This user manual describes the data bases and applications developed in each of the four countries as of June 1992, referred to as the "Initial System." It incorporates the results of applying DEMDESS to four demonstration basins: the Jantra basin in Bulgaria, the Nitra basin in CSFR, the Altalar basin in Hungary, and the Arges basin in Romania.

Application of DEMDESS to the Jantra basin was presented at a wrap-up workshop in Dubravka, Czechoslovakia, in May 1992, hosted by the Slovak Commission on the Environment and USAID. The May wrap-up workshop was attended by the four original participating countries; by representatives from Austria, Croatia, Slovenia, and Yugoslavia; and by representatives of potential donors (USAID, World Bank, the European Bank for Reconstruction and Development [EBRD], United States Environmental Protection Agency [EPA], and the European Community [EC], which is providing a program coordination unit for the multidonor-sponsored Environmental Program for the Danube River basin). Discussion at the wrap-up workshop included possible future extensions of DEMDESS in terms of additional river basins, additional types of users, and additional types of applications.

The conceptual framework of DEMDESS, the strategic and policy options it can address, and the future possibilities for DEMDESS identified in the wrap-up workshop are presented in Chapter 3 of the DEMDES Project WASH Team Assignment Report No. 374. This chapters is reprinted in Appendix A of this manual.

In brief, the focus of DEMDESS is on controlling emissions from industrial and municipal point sources through evaluating the economic, financial, institutional, and stream quality impacts of various policy options (such as construction of treatment facilities, selection of effluent criteria and stream quality standards, provision of federal subsidies for construction, and imposition of fines and taxes on wastewater emissions).

1.2 Computer Requirements

DEMDESS has been implemented using Paradox® * (version 3.5), a commercially available relational data base software package. A quick tour of the capabilities of Paradox® is given in Appendix B, which is an excerpt from the "Introduction" volume of the Paradox® documentation.

USAID has purchased copies of Paradox® and distributed them to the four countries. Paradox® runs on PC-compatible computers having 286, 386, or 486 processors. For efficient use of DEMDESS, the minimum requirements are a 386SX/12 MHz processor, 2 MB of memory, and 100 MB of hard disk for a countrywide data base. At a small additional cost, the recommended configuration is a 486 50 MHz processor, 4 MB memory, and 200 MB hard disk. An ink-jet or laser-jet printer for graphic output is also recommended.

In transferring data from country data bases into DEMDESS, other software programs have been written in BASIC, but the computer requirements for these programs are less restrictive than those for Paradox®.

> **Note**: Before starting any work with DEMDESS be sure to back up the entire system. Ideally, it should be backed up to floppy disks, so that even if a "disaster" were to occur on the hard disk, DEMDESS would not be lost. Backing up the system provides another important operational advantage as well: it allows you to feel free to experiment with queries and edits without having to worry that you will "break" something. DEMDESS is not a "canned" system in which the user is protected from mistakes; it is an open system that allows the user great freedom to test many different options. Backing up the system, therefore, is critical. If you do not know how to back up using DOS (or other backup utilities), find someone to teach you how.

---

* Paradox® is a registered trademark of Borland International, Inc.

1.3 Installation and Initial Use of DEMDESS

Four steps are involved in installing DEMDESS:

1. Installing Paradox®,
2. Customizing Paradox® to the available printer,
3. Installing the DEMDESS files, and
4. Executing Paradox® to use the DEMDESS Initial System.

Installing Paradox® on the hard disk of a PC is described in the Paradox® documentation, and requires only that the user have the five floppy disks containing Paradox®, and that the user execute the Install program found on the first installation floppy disk. Thereafter, the Install program prompts the user as to when to install the appropriate floppy disks in the disk drive. In an early step of the installation, the user is also asked to provide the registration number for the copy of Paradox® being installed.

By convention, Paradox® should be installed in the subdirectory C:\PDOX35, and this name should be added to the PATH list in the AUTOEXEC.BAT file in the C:\ directory of the hard disk.

Customizing Paradox® is accomplished by typing the following command:

C:\PDOX35> Paradox CUSTOM

This starts the Paradox® Custom program, which provides a series of menus. On the Main menu, the Report and Graph options are used to define the type of printer separately for tabular and graphic output, respectively. None of the other options provided by the Custom program is essential in using DEMDESS.

The DEMDESS files for the Initial System are specific to each country, and have been (or will be) distributed on floppy disks. These files should be copied into a subdirectory, for example,

C:\PDOX35\BULGARIA

To access the DEMDESS files from Paradox®, use the DOS command "CD" (change directory) to enter the appropriate subdirectory, and type the program name Paradox. In the example for Bulgaria, the command line would be:

C:\PDOX35\BULGARIA> Paradox

When Paradox® starts up, it shows the Main menu across the top of the screen, and the display area in the lower part of the screen is blank. Menu options can be selected using the cursor keys to move left or right to the appropriate option, and hitting the <ENTER> key. A beginner in using DEMDESS should learn the following menu options first:

"View" to display existing tables on the screen,

"Ask" to make queries concerning existing tables, "Modify Edit" to change data values in a table, and

"Modify Sort" to sort the records in a table.

In using Paradox® to develop DEMDESS, data and procedures can be divided into three items: data bases, scripts, and output.

Data bases: These make up the Paradox® tables, which have been obtained by transferring data from country data bases into Paradox®, and which represent the input data for subsequent analysis. An objective of the DEMDES Project is to standardize the contents and definitions of data items in the data bases among the four Danube countries.

Scripts: Scripts are Paradox® queries or Paradox® Application Language (PAL) programs written to retrieve, join, or reorganize data from the data bases; to perform computations; and/or to produce output tables.

Output: Paradox® tables or graphs represent the output from scripts or queries of the data bases.

1.4 A Quick Tour: The DEMDESS Initial System for Bulgaria

For application to the Jantra River basin in Bulgaria, a series of DEMDESS tables and scripts have been developed. They are described in detail in subsequent sections of this manual. For the purpose of introducing the user to DEMDESS, the types of analyses and output that one can obtain by using the system have been incorporated into 11 plots or graphs (see Figures 1 through 11 at the end of this chapter). Many additional types of analyses can be carried out, and the level of detail in the analysis can be increased; these figures merely illustrate some of the calculations DEMDESS can perform.

The user can examine the information in Figures 1 through 11 by playing the script "Jantra," which carries out the required computations and ends by playing the script "Plotout" to show the results. To play out "Jantra," the user makes the following series of menu selections:

Script	accesses a stored procedure
Play	executes a stored procedure
Jantra	the name given to the stored procedure

To obtain only the precomputed output, the user would make the following menu selections:

Script Play Plotout

Figures 1 through 11 are concerned with the following considerations.

(1) Problem identification

In Figure 1, the actual measured concentrations of BOD_5 along the main stem of the Jantra River are compared with the BOD_5 standards or limits for the three stream quality classifications along the river. It can be seen that measured values of 40 to 90 mg/L greatly exceed the standards, which range from 5 to 25 mg/L.

In Figure 2, the data for 38 towns in the Jantra basin are summarized; none of the towns selected is served by a wastewater treatment plant. The plot in Figure 2 has been obtained by (1) sorting the towns into order by ascending population; (2) computing the associated cumulative population, BOD_5 load, and wastewater flow; and (3) expressing the cumulative values as percentages of the totals for the basin (407,000 people, 13,000 kg/day of BOD_5, and 63,000 cubic meters/day [cmd] total flow). Industrial point sources and two towns served by existing treatment plants are not included in this demonstration application, but adding these sources is within the capability of DEMDESS.

Figure 2 illustrates the types of variations between communities that may be typical of many river basins. Of the 38 towns, the smallest 28 towns represent 39 percent of total population, but contribute little to the total BOD load (25 percent) and total wastewater flow (17 percent). Towns containing major industries contribute more significantly to pollution than would be indicated by population alone; four of the towns (ranked 29, 30, 36, and 37) account for 71 percent of the municipal BOD_5 load within the basin, and 81 percent of the wastewater flow.

(2) Cost comparison of wastewater treatment options

Figures 3 and 4 compare the unit costs (leva/cu m of flow and leva/kg of BOD_5 removed, respectively) of four uniform treatment options designated as follows:

PR01	*Primary treatment (mechanical sedimentation)*
PR02	Enhanced primary treatment (mechanical sedimentation assisted by chemical coagulation)
SAS01	Secondary treatment (single-stage biologically waste-activated sludge)
SAS02	Enhanced secondary treatment (two-stage biologically waste-activated sludge)

The unit costs have been computed by applying the four treatment options to all 38 towns, and computing the sums of the flows, BOD removals, and costs. The costs include amortized annual costs of construction (over 20 years at a 10 percent discount rate), and the annual operation and maintenance costs for labor, electricity, and materials (assuming chlorination of the effluent).

Figure 3 shows that the cost per cubic meter of flow increases significantly with the level of treatment provided, from about 3.5 leva/cu m for primary treatment to about 11 leva/cu m for enhanced two-stage secondary treatment. However, for the objective of BOD_5 removal, the cost per kg of BOD_5 removed (Figure 4) falls into a narrower range of 48 to 56 leva/kg, and the most efficient treatment levels are enhanced primary treatment and single-stage activated sludge.

In developing a basinwide plan, other types of site-specific costs should be included, in particular costs for sewage collection, interceptors, and pump stations within the service area of each town. Estimates of site-specific costs can be inserted into the DEMDESS data base, but have not been included in the Initial System.

(3) Federal taxes, fines, and subsidies

Adopting a uniform treatment requirement for all towns is an option that might be considered; for example, the U.S. adopted secondary treatment as the required minimum treatment level. Uniform treatment does not ensure that stream quality standards are met, however, nor does it minimize the treatment costs of achieving a given set of stream quality standards. It does achieve equity of treatment among communities, although small villages and towns will pay higher per capita costs than large cities, as a result of the economies of scale for large treatment plants.

Figure 5 shows the implications at the national level of considering five uniform treatment options for the 38 towns in the Jantra basin: No treatment, and the four levels of treatment considered previously. For the results shown, three types of federal income or expenditure have been considered:

Taxes — A charge of 0.5 leva per cu m of wastewater flow, paid by any town that does not provide wastewater treatment.

Fines — A charge of 25 leva per kg of effluent BOD_5 that is in excess of the allowable effluent limit. This limit varies by stream and by reach within the Jantra, from 5 to 25 mg/L BOD_5 concentration in the effluent, and equal to the allowable BOD_5 concentration in the stream itself. The effluent limits are quite low, and make no allowance for dilution of effluent in the receiving stream waters. Charges are also made for excessive TSS (total suspended solids), but the allowable limits are very high (1,000 to 1,500 mg/L) and do not generate revenue from fines for the Jantra basin. A total of 26 parameters (including 12 heavy metals) are included in the existing system of fines in Bulgaria, but lack of data has precluded consideration of these in the analysis.

The fine of 25 leva/kg BOD_5 used in this example was selected to present a suitable pattern of income from taxes and fines for a hypothetical environmental fund under existing conditions of no treatment, which could later be spent on subsidies to pay a portion of the construction costs of a treatment plant. The existing fine on BOD_5 is actually 2 leva/kg.

Subsidies — A federal subsidy of 75 percent of capital costs has been assumed, patterned after U.S. experience. Capital costs include construction, and an allowance for engineering, contingencies, and legal and other administrative costs. In Figure 5, the subsidy has been expressed as an annual cost amortized over the planning period (20 years, at a 10 percent discount rate). The remaining

25 percent of the capital cost would be paid by the municipalities. The amortized value of the municipal share is shown in Figure 5 as ANNCAP, the annual capital cost.

The results shown in Figure 5 indicate that the initial income in an environmental fund would be about 120 million leva/year in taxes and fines under existing conditions of no municipal wastewater treatment. For uniform primary treatment (treatment option PR01), the annual income from fines would be sufficient to pay the subsidy on capital cost, without any expenditure from an environmental fund. For enhanced primary treatment (PR02), subsidies would exceed fines by about 25 million leva/year over the planning period, but this is small compared with the initial income in the environmental fund. For single-stage activated sludge (SAS01), income from fines would virtually cease, and the subsidies on construction would be 90 million leva/year; in the absence of government equity financing, it would take many years of continuing pollution to accumulate sufficient money in an environmental fund. Two-stage activated sludge (SAS02) would require subsidies of about 140 million leva/year, which clearly cannot be supported by the trial values for taxes and fines used in this example.

(4) Options at the municipal level

A major purpose of taxes, fines, and subsidies at the federal level is to promote the construction of wastewater treatment plants by municipalities. A basic assumption in this example is that each municipality knows the costs of each treatment option (including the effects of taxes, fines, and subsidies), and will select the most cost-efficient one.

The municipal-level costs for the uniform treatment options are shown in Figure 6. These costs include taxes, fines, remaining annual capital costs (labeled ANNCAP), and annual operation and maintenance costs (O&M). In aggregate, the 38 towns would minimize their costs by selecting single-stage activated sludge (SAS01), followed closely by enhanced primary treatment (PR02). The total annual costs for these two options, in the vicinity of 80 to 85 million leva/year each, would provide a substantial savings over the cost (in fines and taxes) of about 120 million leva/year when no wastewater treatment is provided.

The annual per capita costs at the municipal level are shown in Figure 7. For the five treatment options, these costs follow the same proportions as in Figure 6, and represent only a change in the scale of the y axis. The per capita costs range from almost 200 to nearly 300 leva/year, which would seem to reflect the financial feasibility of user charges to recoup the municipal share of the costs, particularly when the charges to domestic users would be lower than shown here, after industry pays its share of the total cost.

For the trial values of fines, taxes, and subsidies, the minimum-cost solutions in the individual towns are summarized in the table below, in which the numerical values have been generated by a Paradox® query.

TREATLEV	Towns	Population	Kg/d BOD_5 Removed	Cost[a]
NONE	12	32069	0	0
PR02	19	290992	2163	56
SAS01	7	83767	7965	78

[a]Costs (million leva/year) include total annualized capital costs and annual O&M costs.

The cumulative populations, BOD_5 removals, and total annual costs for the above example are plotted in Figure 8, which has been developed in the same manner as Figure 2. The 11 smallest towns would choose to pay taxes and fines rather than build treatment plants, and the majority of the costs (71 percent) and BOD_5 removals (73 percent) are associated with the four largest municipal sources of BOD_5 (the towns ranked 29, 30, 36, and 37 in ascending population).

(5) Impact on ambient stream quality

Modeling of stream water quality has traditionally been concerned with estimating the dissolved oxygen (DO) profile along a network of rivers using the Streeter-Phelps equations, which DEMDESS can do. A large variety of Fortran computer programs has been developed, of which the EPA's QUAL2 model is an example. However, the same basic equations and methodology can be employed in Paradox® by embedding the computations within a PAL script (PAL is the Paradox® Application Language provided in the Paradox® software package).

For the Jantra basin, data are insufficient to calibrate a DO model, so a very coarse set of assumptions has been made: a uniform velocity of 0.5 m/s in all streams, and a BOD_5 decay coefficient (similar to the k1 coefficient applied to ultimate BOD) of 2/day. The results under existing conditions are shown in Figure 9 for the main stem of the Jantra River, in which the cumulative BOD_5 load discharged into the river is compared with the modeled values of decayed BOD_5 load along the river. At river kilometer 100, it can be seen that a large BOD_5 load has been discharged into a tributary of the Jantra, but the BOD_5 in the tributary has decayed away to a small level by the time the flow enters the main stem.

Under the minimum-cost option, the total BOD_5 load discharged within the basin would decline from 13,000 kg/day to 2,800 kg/day. The resulting improvements in BOD_5 load along the Jantra are shown in Figure 10.

Monitoring of ambient water quality is a major task of the environmental ministries and agencies in the four countries. The results of water quality sampling and testing have been incorporated into DEMDESS, as well as the results of the international sampling program at 11 stations along the Danube, which have been transferred from InfoDanube into DEMDESS. The data can be used in tracking the changes in water quality over time, and comparing the results with water quality standards, as illustrated in Figure 11 for one station in the Jantra basin. This same type of analysis is necessary in selecting representative values of parameters such as BOD_5 and COD to be used in calibration of water quality models.

Figure 1

Jantra Basin, Bulgaria

BOD Concentration Measured vs WQ Std

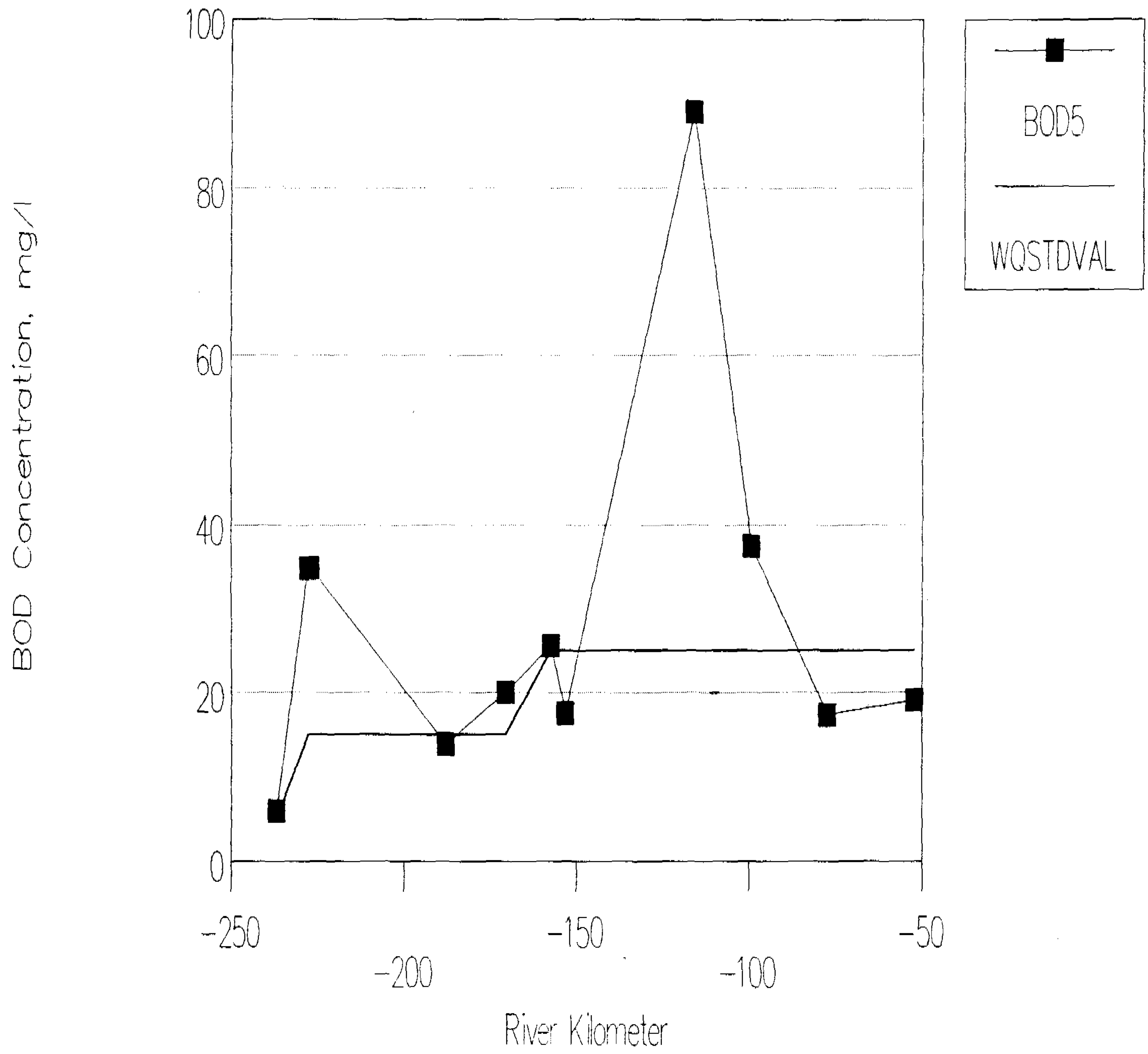


Figure 2

Jantra Basin, Bulgaria

Cum. Population, BOD Load and Flow

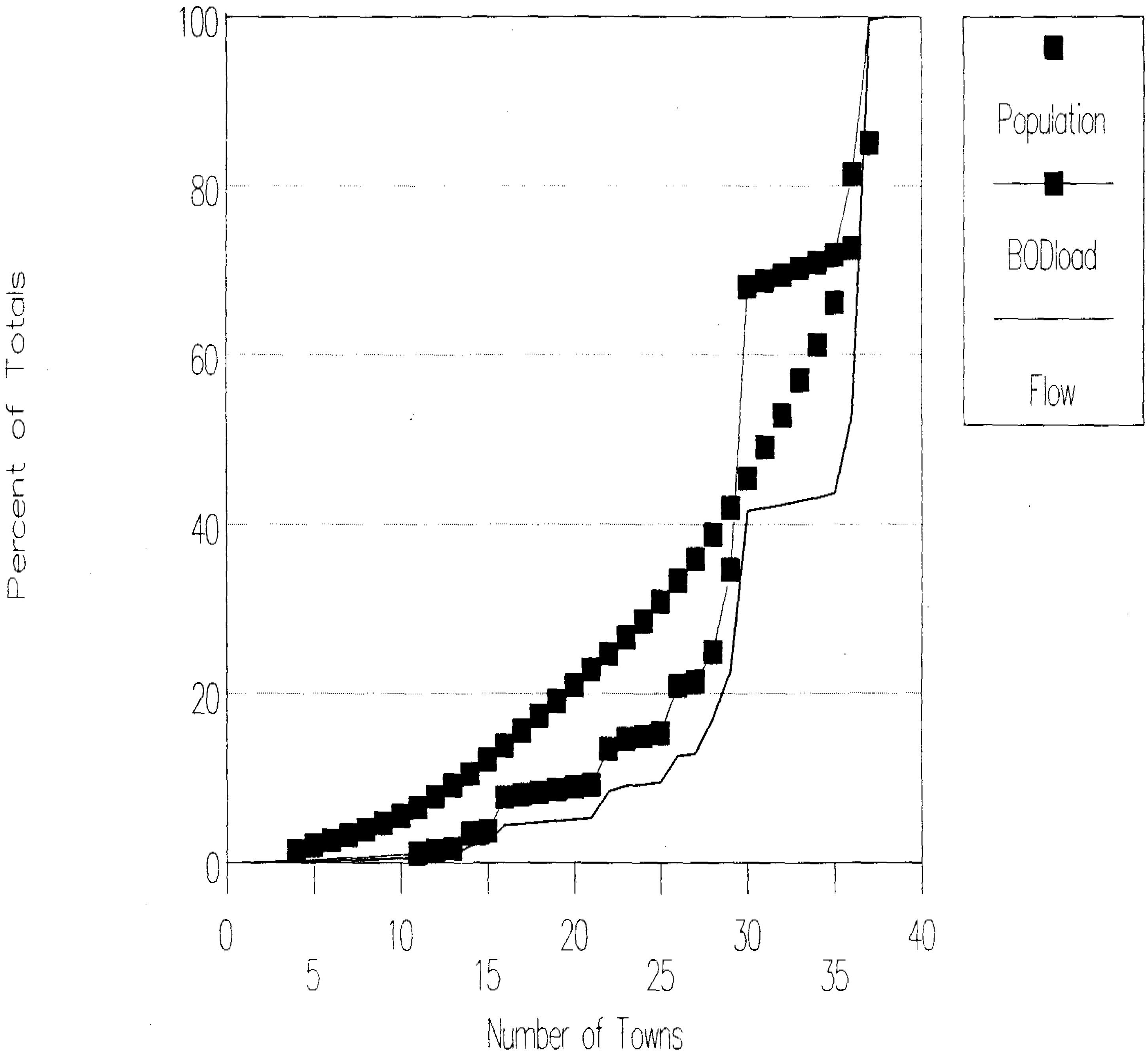


Figure 3

Jantra Basin, Bulgaria

Cost per Water Volume Treated

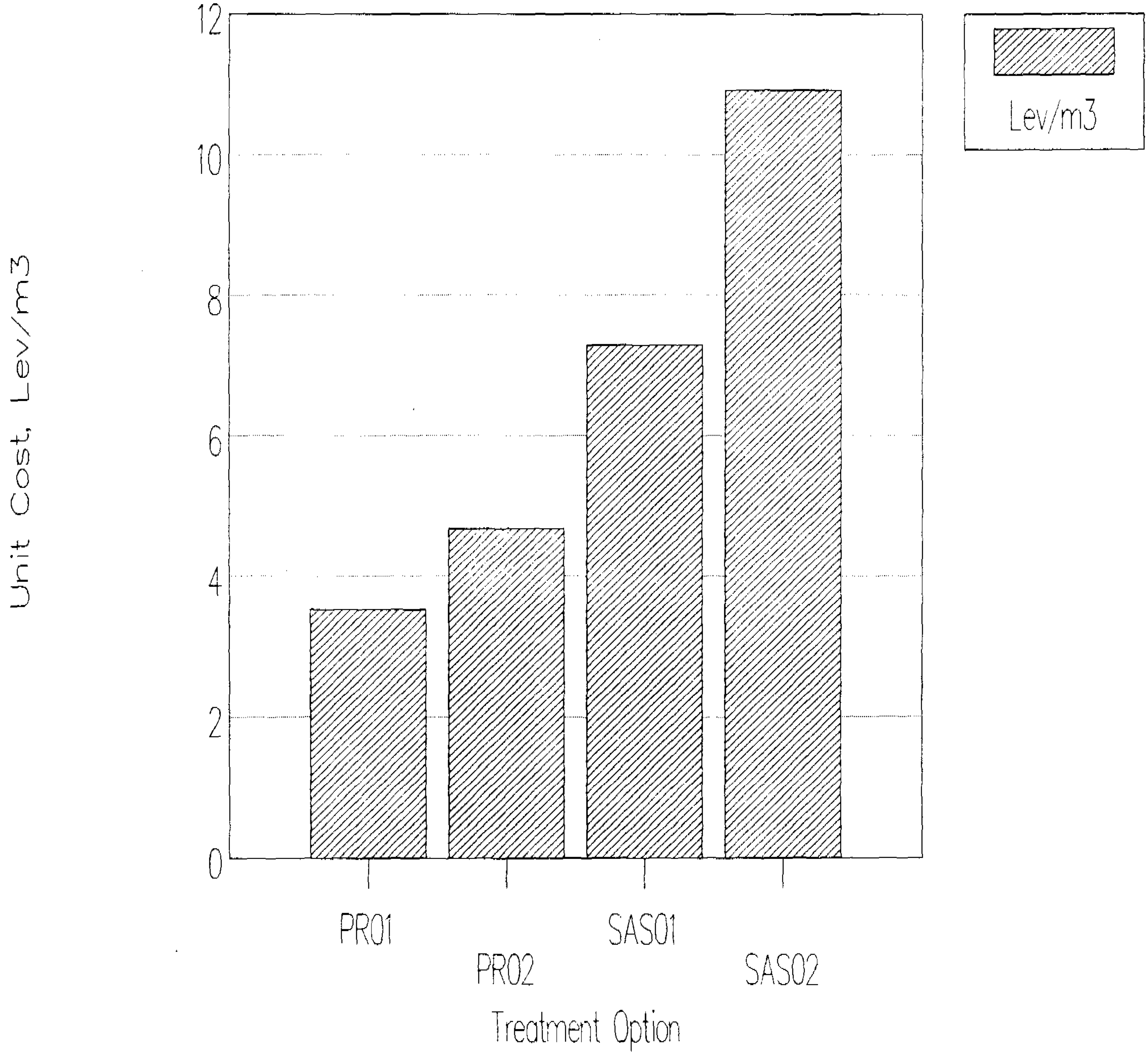


Figure 4

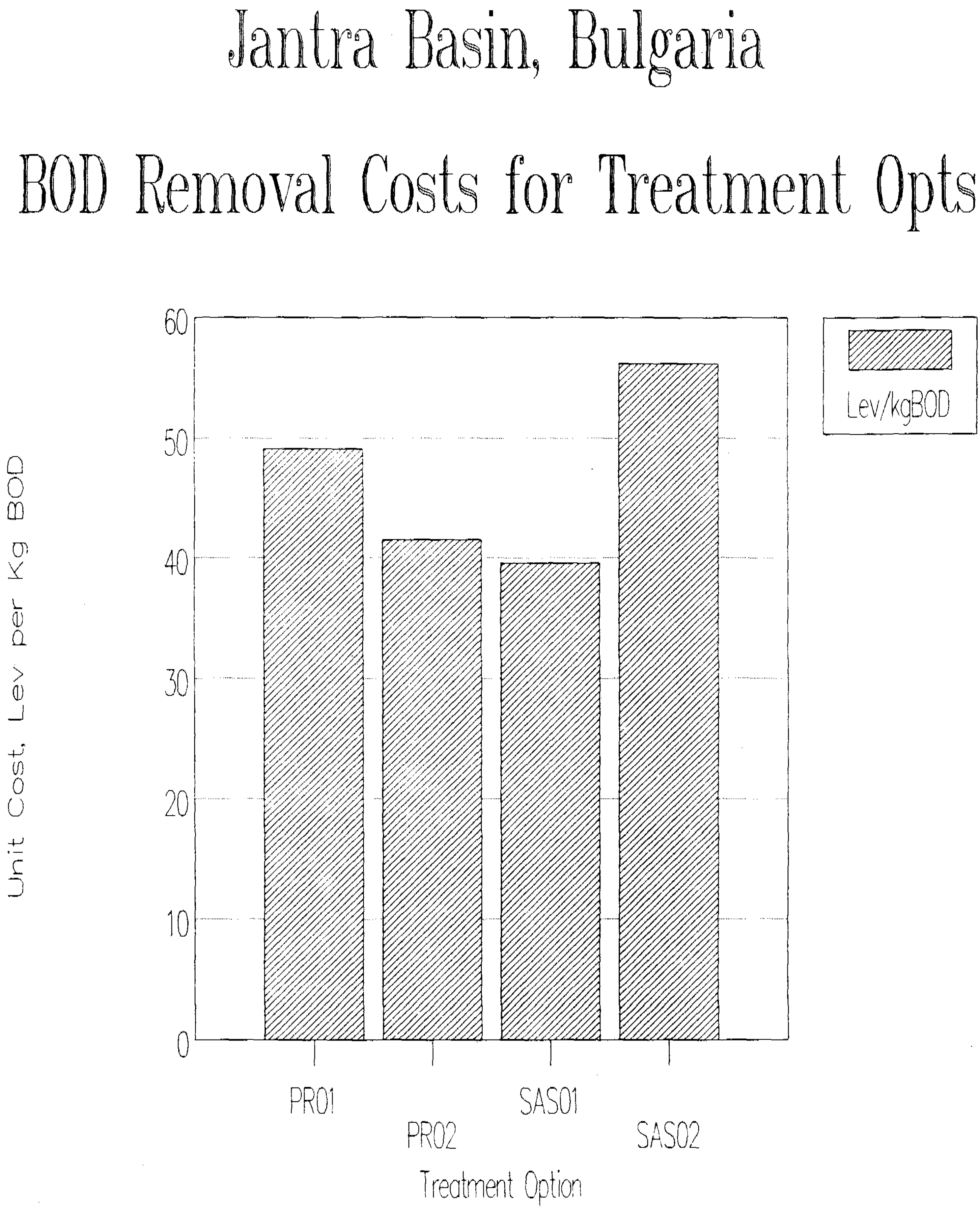

Jantra Basin, Bulgaria
BOD Removal Costs for Treatment Opts
Unit Cost, Lev per Kg BOD
60
50
40
30
20
10
0
Lev/kgBOD
PR01
PR02
SAS01
SAS02
Treatment Option

Figure 5.

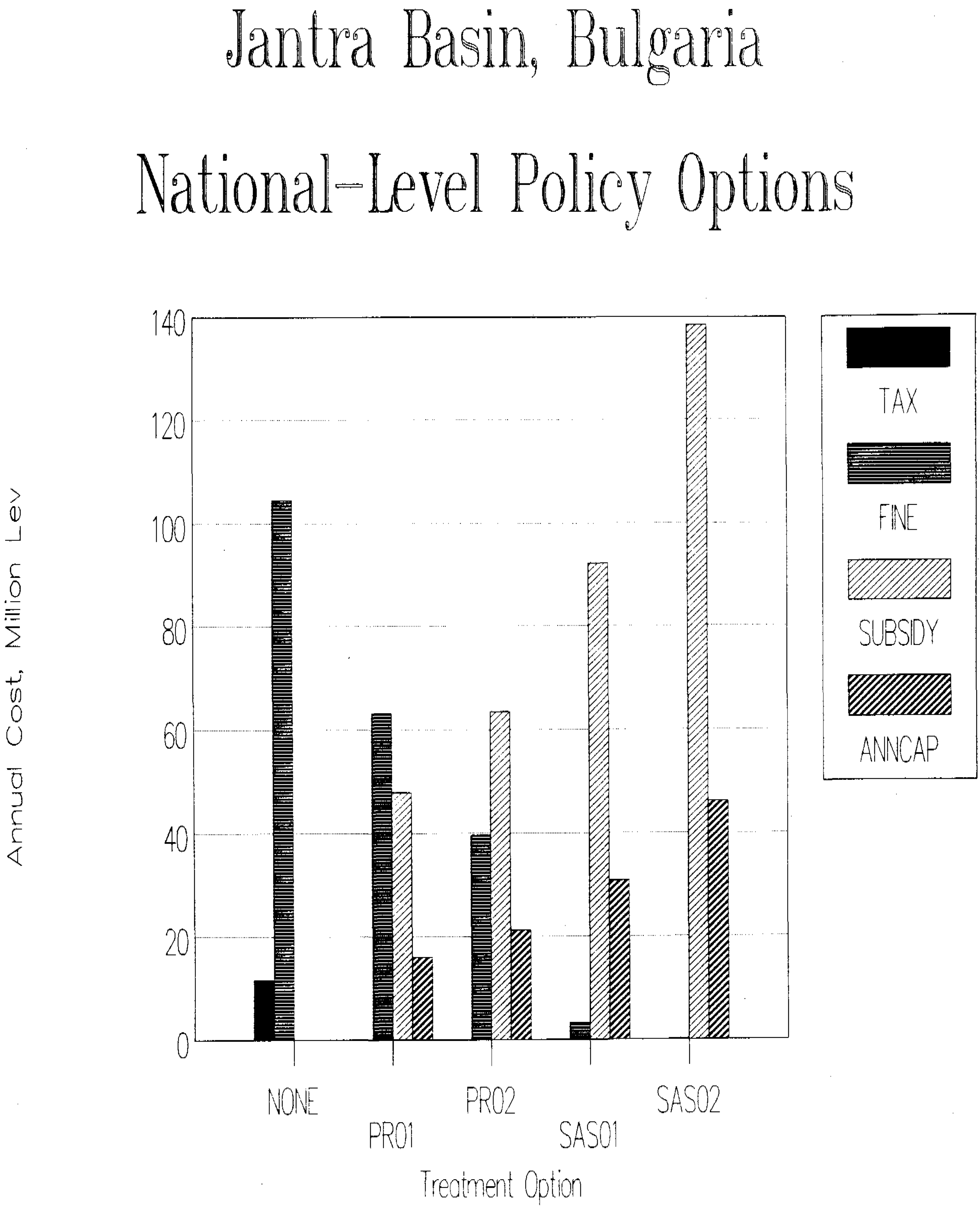

Jantra Basin, Bulgaria
National-Level Policy Options
140
120
100
80
60
40
20
0
Annual Cost, Million Lev
TAX
FINE
SUBSIDY
ANNCAP
NONE
PR01
PR02
SAS01
SAS02
Treatment Option

Figure 6

Jantra Basin, Bulgaria

Municipal-Level Cost Options

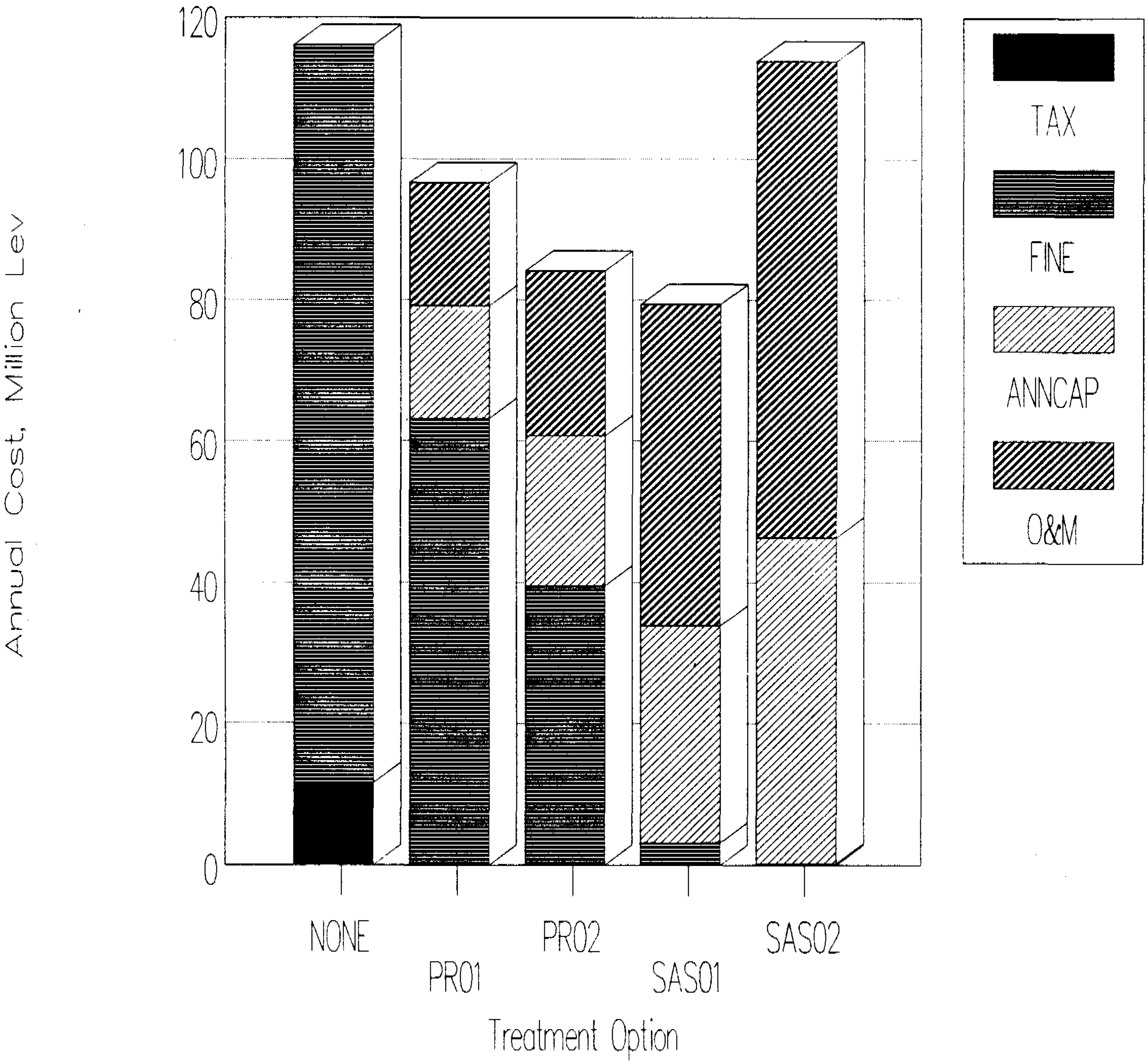


Figure 7

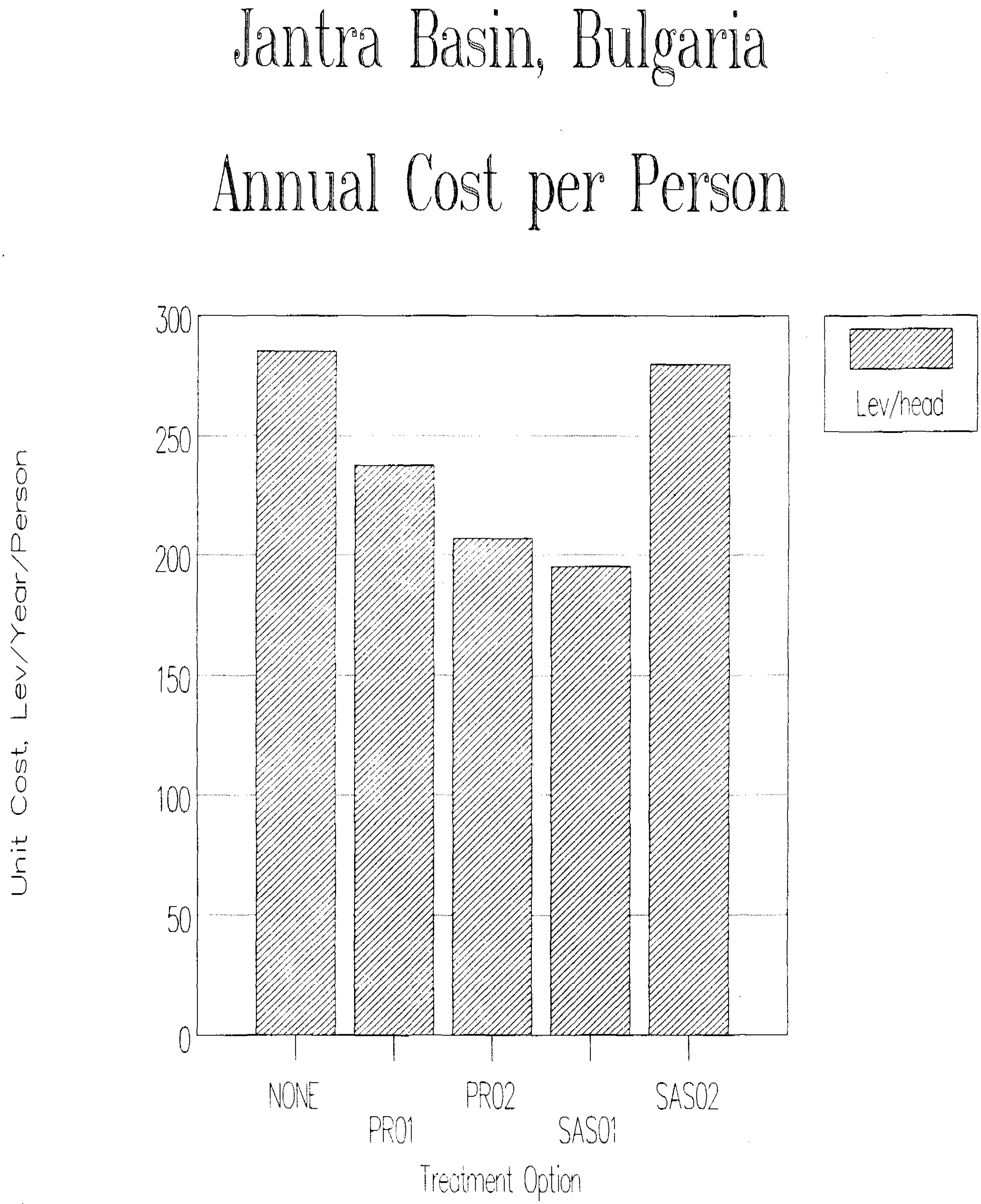

Jantra Basin, Bulgaria
Annual Cost per Person
Unit Cost, Lev/Year/Person
300
250
200
150
100
50
0
Lev/head
NONE
PR01
PR02
SAS01
SAS02
Treatment Option

Figure 8

Jantra Basin, Bulgaria

Cum. Population, BOD rem'vd, Cost

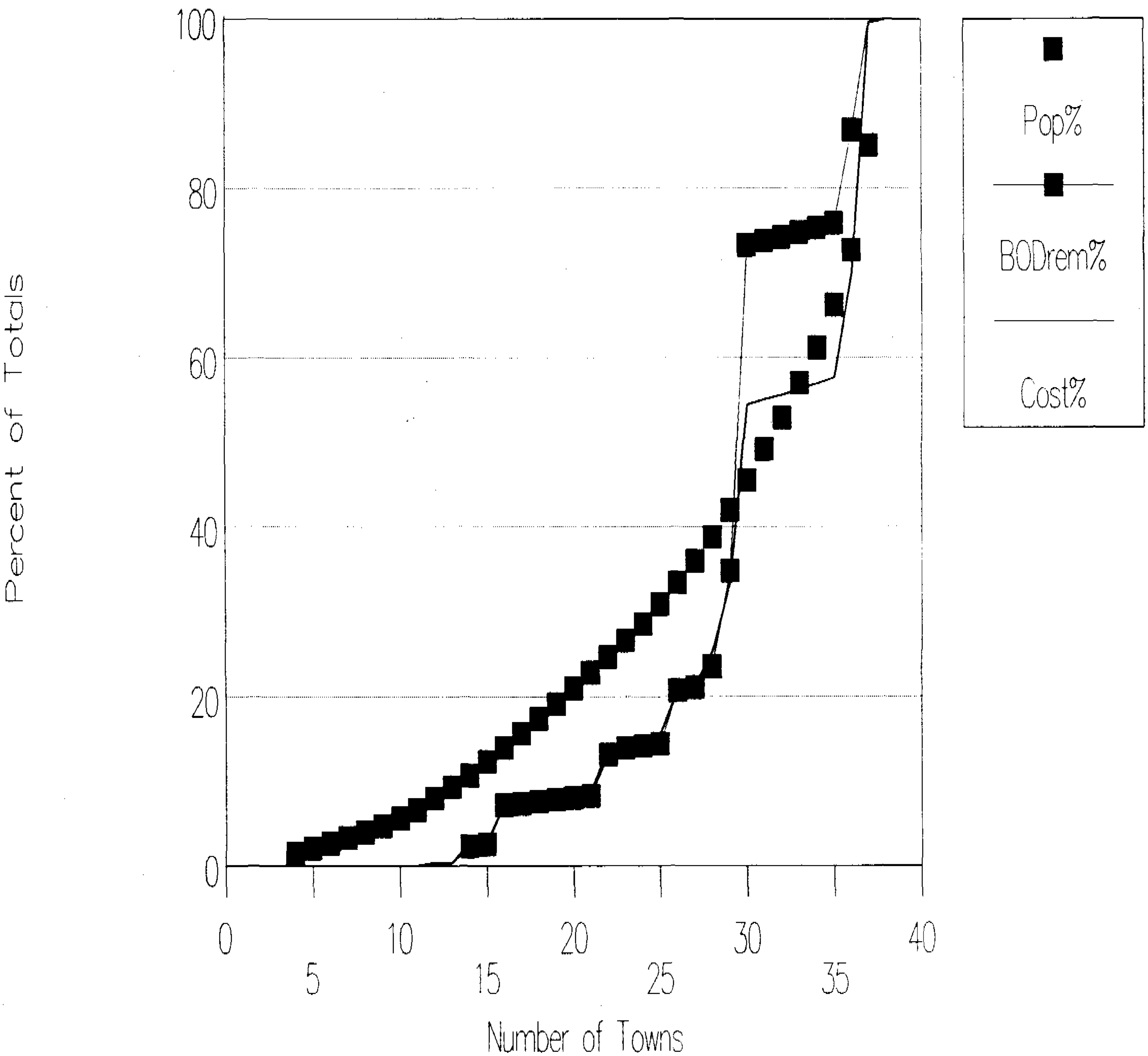


Figure 9

Jantra Basin, Bulgaria

Existing BOD Load (Cum. & Decayed)

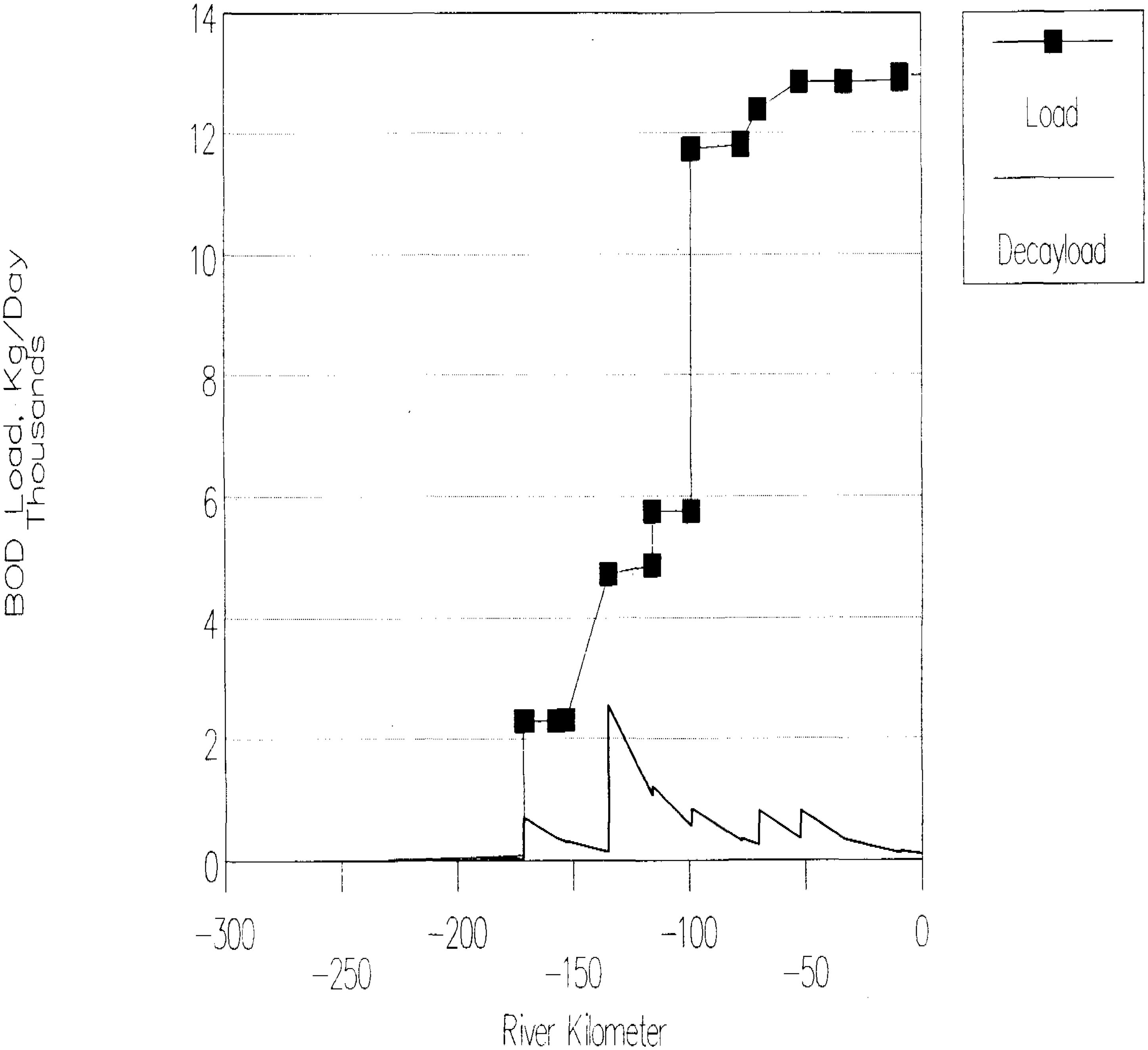


Figure 10

Jantra Basin, Bulgaria

Min.Cost BOD Profile (Cum. & Decayed)

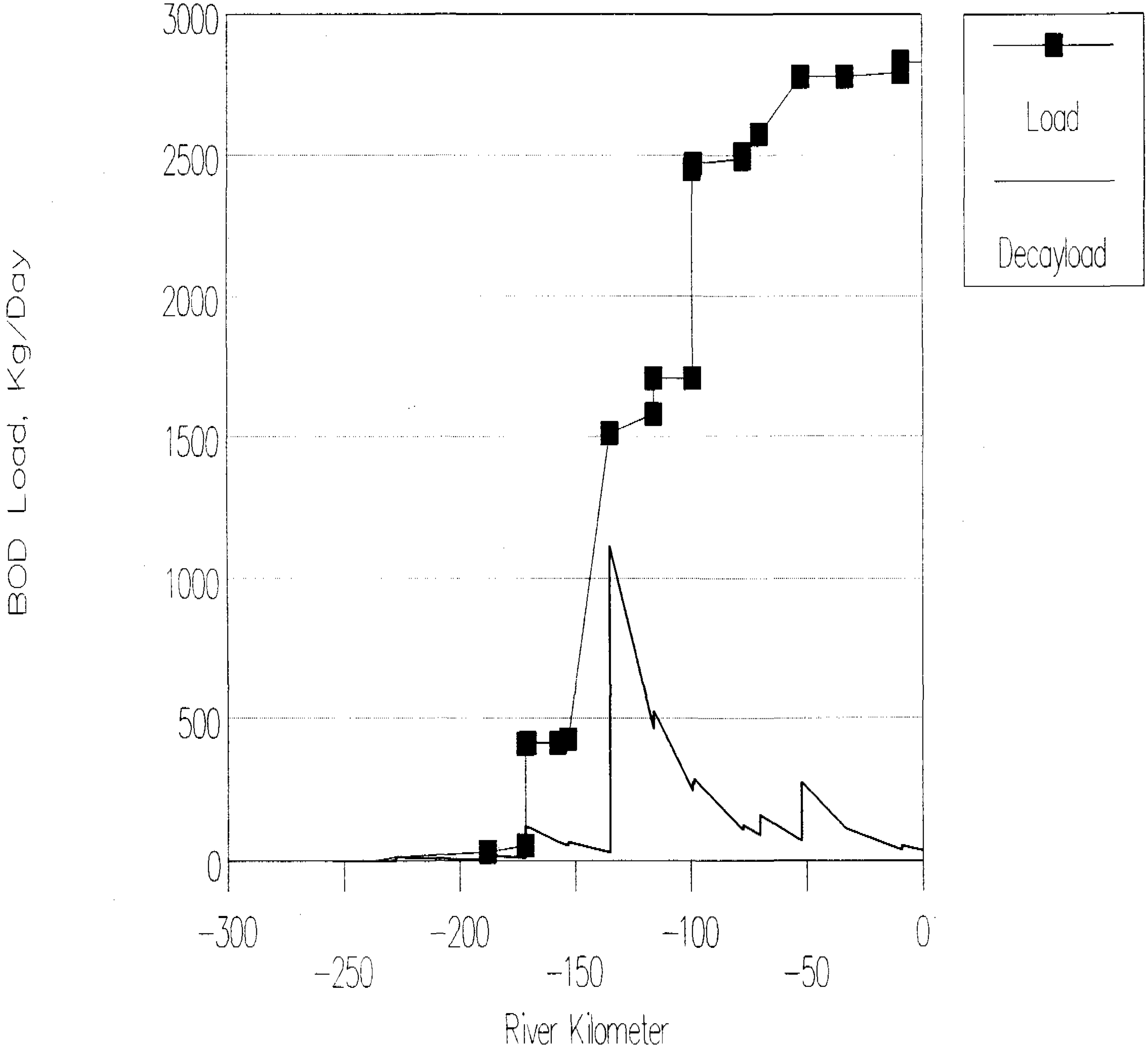
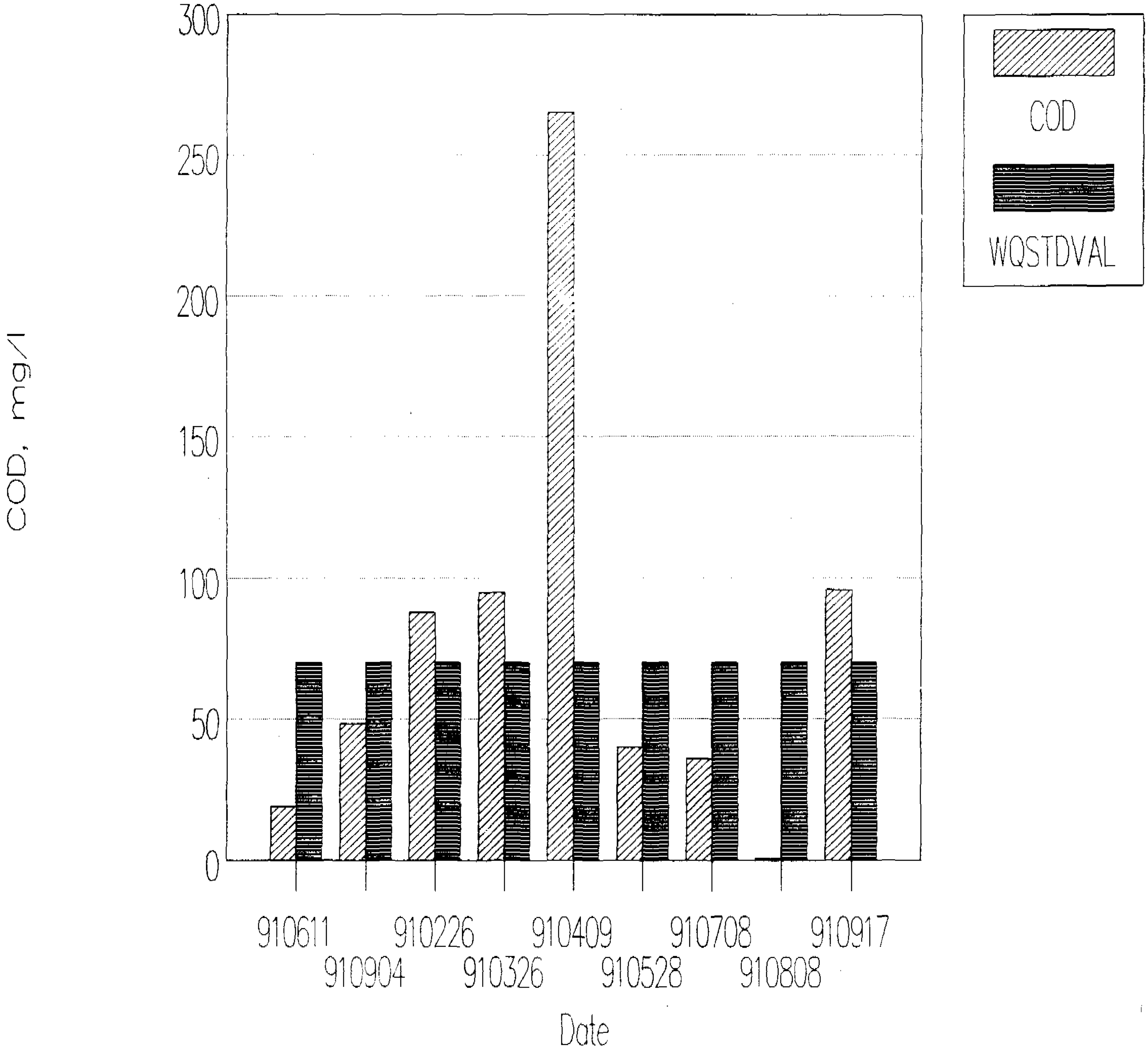


Figure 11

Jantra Basin, Bulgaria

COD Measurements at 1 Station



Chapter 2

DATA BASE DESIGN

2.1 Data Base Structure and Organization

In this chapter, the organization and contents of the 21 DEMDESS data bases are described. The use of Paradox® scripts to obtain output also is described briefly. These scripts are those developed by the WASH team; users of DEMDESS will find it convenient to create their own scripts for commonly used queries and computations.

The DEMDESS data base consists of a set of tables, data base relationships, and data elements. This design is expressed formally in the DEMDESS data base dictionary, which contains definitions of all data elements in each DEMDESS table. Throughout this chapter the data base design and dictionary are discussed. Because of the central importance of the dictionary for almost all DEMDESS operations, however, the dictionary is consolidated and repeated in Appendix D. As users become familiar with DEMDESS, they will likely find the consolidated versions more convenient.

Figure 12 shows the organization of the 21 data bases into 6 subsystems. The grouping of tables into these 6 subsystems is for descriptive purposes only; a subsystem is not a defined entity within Paradox®, but is a convenience adopted herein for describing DEMDESS. The 6 subsystems are as follows.

1. *Regulatory Subsystem*: The fines, taxes, and water quality standards for both treated effluent and ambient stream quality.

2. *Institutional Subsystem*: The public agencies or authorities responsible for emissions management.

3. *Emissions Subsystem*: The industrial and municipal dischargers, their wastewater emissions, and associated water supply data.

4. *Treatment Subsystem*: The costs, effluent characteristics, and parameters for economic and financial analysis of wastewater treatment plants.

5. *Reach Subsystem*: The description of river networks, including the connectivity between reaches, the sequence in which reaches should be analyzed for water quality modeling, stream flow, and cross-references to the hydrologic sequencing data for the participating countries.

6. *Water Quality Subsystem*: The monitoring data on ambient stream quality and coefficients for stream quality modeling.

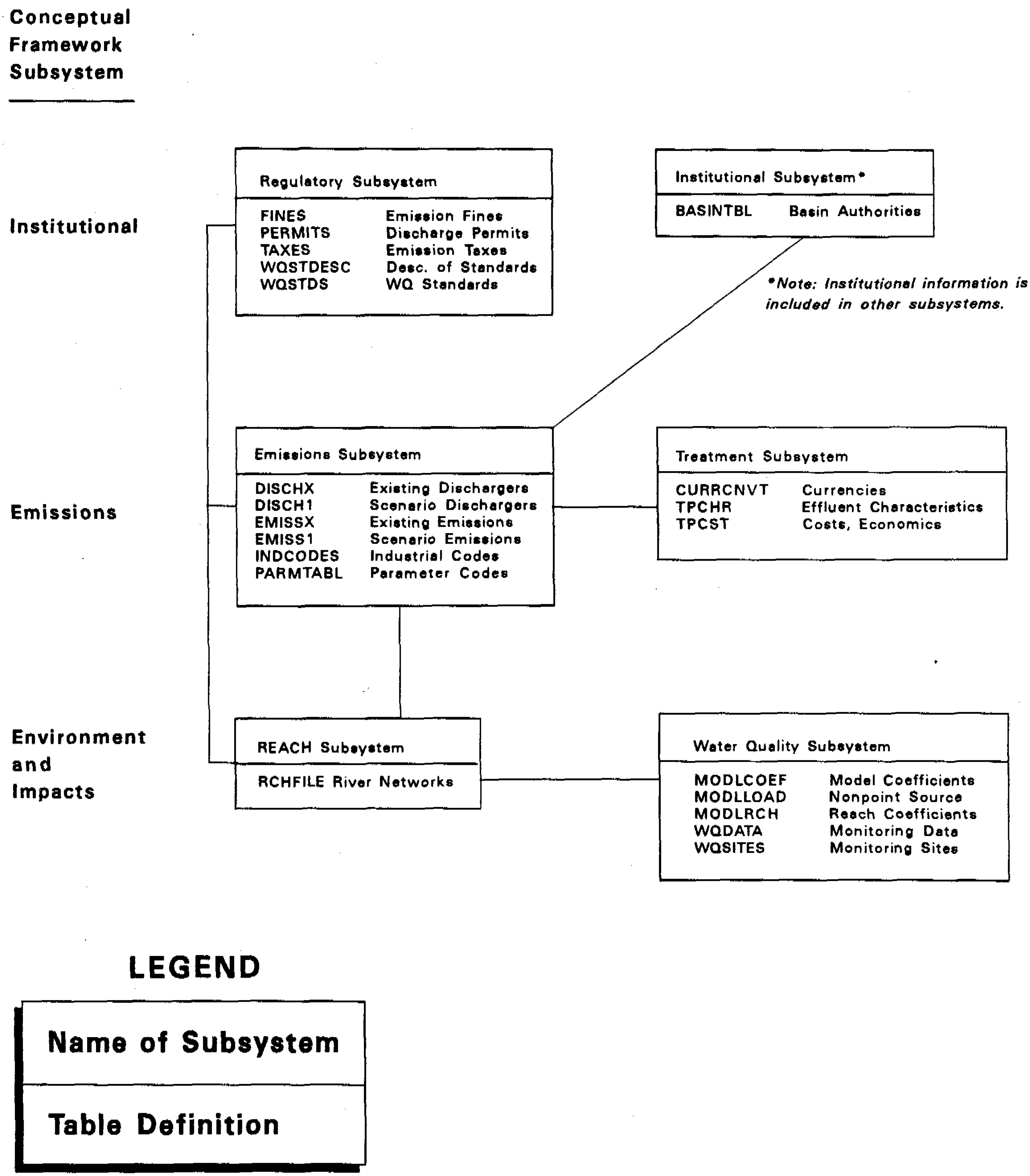


Figure 12. Organization of DEMDESS Databases

Each data base is a Paradox® table, in which the rows of the table are called records, and the columns are called fields. In a relational data base, the fields can be of two types: key fields, which link data bases, and data fields, which generally do not appear in more than one data base, and are used to store data or describe the contents of a data record.

The DEMDESS data bases are summarized in Table 1, including an indication of the number of data records in a typical Initial System (usually, for the Jantra basin in Bulgaria). The listing of key fields shown in Table 1 has been abbreviated in the following respects.

REACH	Refers to three fields defining the river location of a discharger, water supply intake, or water quality monitoring station: the BASIN, the SEGMENT or numbered river segment within a basin, and the RIVER KM, which is the distance from the downstream end of the river segment to the point of discharge or other point of interest (for example, a monitoring station, a location where a modeling coefficient changes, and so on).
INST'NS	Refers to a list of institutions having a role or responsibility with respect to a discharger; these include a basin authority, a regional authority, and the authority responsible for the discharge permit.
WTR SPPLY	Refers to both the institutional and quantitative data on water supply for a discharger. Due to lack of data, analyses of water supply data have not been implemented in the Initial Systems for the four countries.
PARAMETR	Refers to the field label Parmcode, for which each water quality parameter is assigned a numeric code ($1=BOD_5$, $4=TSS$, etc.). For the sake of generality, the quantity of flow is also considered a parameter (Parmcode=1,000).
INDCODE	Refers to the Industrial Code, a numeric value assigned to the type of industry or discharger. In particular, INDCODE=22 refers to municipal dischargers.

The output tables obtained for the Initial Systems are computed and retained, and represent joined subsets, modifications, or rearrangements of the input data bases. For the Bulgarian Initial System, the results of computations are stored in Tables TRTPLCY1,..,TRTPLCY6, and the tables that save data for plots (Figures 1 through 11 in Chapter 1) are saved in Tables PLOT1,..., PLOT7ba4,...,PLOT11.

Table 1

ORGANIZATION OF DEMDESS DATA BASES

Subsystem:	EMISSIONS **Table: DISCHX** Records: 174
Description:	Existing dischargers
Key fields:	DISCHID TREATCLASS INDCODE REACH TAXID FINEID WQSTD TREATLEV PROVINCE INST'NS WTR SPPLY COUNTRY TREATID
Data fields:	Population, flow (cmd), name, treatment sites, reach connectivity, capacity, peak flow, days & hours of operation, residential/non-res./indust'l flows, sewer/street length, % sewered, wtr. supply

Subsystem:	EMISSIONS **Table: DISCH1** Records: 174
Description:	Scenario dischargers
Key fields:	DISCHID TREATCLASS INDCODE REACH TAXID FINEID WQSTD TREATLEV PROVINCE INST'NS WTR SPPLY COUNTRY TREATID
Data fields:	Same as table DISCHX

Subsystem:	EMISSIONS **Table: EMISSX** Records: 565
Description:	Existing emissions
Key fields:	DISCHID PARAMETR TREATCLASS
Data fields:	Data source, quality-control code, data value, time & date, discharge type

Subsystem:	EMISSIONS **Table: EMISS1** Records: 573
Description:	Scenario emissions
Key fields:	DISCHID PARAMETR TREATCLASS
Data fields:	Same as table EMISSX

Table 1 (cont'd)

Subsystem:	EMISSIONS **Table: INDCODES** Records: 25
Description:	Industrial sector codes
Key fields:	INDCODE
Data fields:	Code description, cross-reference to country codes

Subsystem:	EMISSIONS **Table: PARMTABL** Records: 46
Description:	Water-quality parameters
Key fields:	PARAMETR
Data fields:	Parameter name, cross-reference to country codes

Subsystem:	REACH **Table: RCHFILE** Records: 44
Description:	River reach networks
Key fields:	REACH WQSTD COUNTRY PROVINCE
Data fields:	Basin/segment/km of current reach, 3 upstream reaches, 2 D/S reaches, level, D/S reach level, sequence no., seg length, Pathkm, reach name, flow, velocity, k2 coeff., parmcode, pt. source flow & load, ka, cum. & decayed load

Subsystem:	TREATMENT **Table: CURRCNVT** Records: 8
Description:	Currencies, unit prices
Key fields:	CURRENCY COUNTRY
Data fields:	Exchange rate, year, country, local unit costs for labor, energy and material costs

Subsystem:	TREATMENT **Table: TPCHR** Records: 52
Description:	Treatment characteristics
Key fields:	TREATLEV PARAMETR TREATID
Data fields:	Treated-effluent concentration, % remaining in effluent, data source

Table 1 (cont'd)

Subsystem:	TREATMENT	**Table: TPCST**	Records: 15
Description:	Treatment costs, economics		
Key fields:	TREATLEV CURRENCY TREATID		
Data fields:	Treatment description, data source, currency, cost coefficients, eng. & cont. allowance, % subsidy, discount rate, years for amortization, sludge production rate and percent dry solids		

Subsystem:	REGULATORY	**Table: FINES**	Records: 116
Description:	Fines on emissions		
Key fields:	FINEID WQSTD PARAMETR CURRENCY		
Data fields:	Amount of fine, units		

Subsystem:	REGULATORY	**Table: TAXES**	Records: 3
Description:	Taxes on wastewater flows		
Key fields:	TAXID WQSTD PARAMETR CURRENCY		
Data fields:	Amount of tax, units		

Subsystem:	REGULATORY	**Table: WQSTDESC**	Records: 29
Description:	Water-quality standards		
Key fields:	WQSTD		
Data fields:	Description of water-quality standard		

Subsystem:	REGULATORY	**Table: WQSTDS**	Records: 205
Description:	Stream/effluent standards		
Key fields:	WQSTD PARAMETR		
Data fields:	Value(s) of water-quality standard, units		

Table 1 (cont'd)

Subsystem:	WATER QUALITY **Table: MODLCOEF** Records: 0
Description:	Modeling coefficients
Key fields:	REACH PARAMETR
Data fields:	Modeling coefficients ka, kb, kc, kd

Subsystem:	WATER QUALITY **Table: MODLRCH** Records: 0
Description:	Modeling coefficients
Key fields:	REACH PARAMETR
Data fields:	Condition code, flow, velocity, k2 rearation

Subsystem:	WATER QUALITY **Table: WQDATA** Records: 5393
Description:	WQ monitoring data
Key fields:	WQSITE PARAMETR
Data fields:	Date, time, location code, quality-control code, value of parameter, remarks

Subsystem:	WATER QUALITY **Table: WQSITES** Records: 58
Description:	WQ monitoring sites
Key fields:	WQSITE REACH PROVINCE REGION COUNTRY AGENCY
Data fields:	Data source, type of station, start/end dates, name or description of WQ station site

Subsystem:	INITIAL OUTPUT **Table: PLOTn**
Description:	Tables for plots 1 - 11

Table 1 (cont'd)

Subsystem:	INITIAL OUTPUT **Table: TRTPLCYn**
Description:	Tables for Jantra basin computations

Subsystem:	INITIAL OUTPUT **Table: RCHROUT**
Description:	Jantra BOD decay model

Subsystem:	INITIAL OUTPUT **Table: RCHGRAF**
Description:	Jantra existing BOD profile

Subsystem:	INITIAL OUTPUT **Table: RCHGRAF2**
Description:	Jantra min. cost BOD profile

2.2 Institutional Subsystem

Data for the Institutional Subsystem are derived from input data from the Emissions, Regulatory, and Water Quality subsystems. Those data are contained thus far in the following tables, which are described in more detail in the remainder of this chapter: DISCHX, DISCH1, EMISSX, and EMISS1 (from the Emissions Subsystem); PERMITS (from the Regulatory Subsystem); and WQSITES (from the Water Quality Subsystem).

2.3 Emissions Subsystem

2.3.1 Data Bases

Six data bases are included in the Emissions Subsystem, for which the Paradox® table names are DISCHX, DISCH1, EMISSX, EMISS1, INDCODES, and PARMTABL.

The tables INDCODES and PARMTABL contain definitions of the numeric code numbers assigned to industrial sectors and water quality parameters, respectively.

For example, in the table INDCODES,

INDCODE=1 designates the coal mining industry,

INDCODE=12 represents the pulp and paper industry, and

INDCODE=22 signifies municipal wastewater.

In the table PARMTABL, examples include

PARMCODE=1 is BOD_5,

PARMCODE=2 is COD,

PARMCODE=4 is TSS (total suspended solids), and

PARMCODE=1000 is wastewater flow rate (cmd).

These two tables are used in many of the other subsystems in addition to Emissions. Sample listings are shown in Table 2 for INDCODES and Table 3 for PARMTABL.

The tables DISCHX and EMISSX, in a countrywide or Danube-wide DEMDESS, would contain the current and historic data on all dischargers and their emissions. The two tables are linked by several key fields, which include the DISCHID (a user-defined unique identifier for each discharger). For example, all the historic BOD_5 measurements for a given discharger described in DISCHX would appear in EMISSX under the same DISCHID with PARMCODE=1 (the parameter code for BOD_5).

In general, the DEMDESS user will be interested in examining estimates for future conditions or the data for some subset of all the dischargers in a country or in the Danube basin. The subsets or data modifications derived from DISCHX and EMISSX should be selected or computed and stored in the two associated "scenario" tables DISCH1 and EMISS1. The tables DISCH1 and EMISS1 are used in the DEMDESS scripts developed by the WASH team. These may be either direct copies of DISCHX and EMISSX, or they may be subsets based on selections of the key fields provided in DISCHX, such as COUNTRY, INDCODE, REACH, and PROVINCE. In the more general case, the user would also modify such variables as population, flow, and emission characteristics to reflect future conditions, such that the data in *DISCH1* and *EMISS1 would differ from* the existing conditions.

The fields in the tables DISCHX and DISCH1 are identical and are defined in Table 4. Similarly, the fields in the tables EMISSX and EMISS1 are defined in Table 5.

Table 2

SAMPLE LISTING OF TABLE **INDCODES**—INDUSTRIAL CODES

INDCODE	**INDCODEDESC**
1	Coal Mining
2	Ore Mining and Dressing
3	Mineral Mining and Processing

(cont'd)

Table 2 (cont'd)

4	Petroleum and Gas Extraction
5	Chemical Industry
6	Machinery Manufacturing
7	Electrical and Electronic Manufacturing
8	Energy Production
9	Transportation
10	Construction
11	Lumber Products
12	Pulp and Paper
13	Agriculture
14	Miscellaneous Food and Beverages
15	Sugar
16	Textiles
17	Animal Farms
18	Meat Processing
19	Dairy Products
20	Leather Tanning and Manufacturing
21	Slaughter House
22	Municipal WWTP
23	Fuels
24	Logging
25	Other - Not Classified

Table 3

SAMPLE LISTING OF TABLE
PARMTABL—WATER QUALITY PARAMETER CODES

PARMCODE	PARMNAME	S-COE	B-MOE	Popozov
1	BOD-5	3	1	
2	COD-Mn (Manganese)	4	2	20
3	TDS	8	3	8
4	TSS		4	
5	Ammonia NH3	11	5	14
6	Nitrates NO3	12	6	16
7	Chloride CL	14	7	10
8	Sulphate SO4	15	8	11
9	Hydrogen Sulfide H2S	6	9	
10	Oil		10	
11	Phenols	19	11	33
12	Iron Fe	9	12	12
13	Manganese Mn	10	13	13
14	Phosphates		14	18
15	Chromium (6+) Cr-6	29	15	26
16	Arsenic As	26	16	
17	Lead Pb	25	17	24
18	Copper Cu	27	18	34
19	Zinc Zn	31	19	29
20	Detergents-anionic	20	20	32
21	Temp-air C			2
22	Temp-water C			3
23	DO mg/l	1		4
24	pH	7		5

(cont'd)

Table 3 (cont'd)

PARMCODE	PARMNAME	S-COE	B-MOE	Popozov
25	Total Solids mg/l			6
26	Evaporative Solids			7
27	TDS Evaporate			9
28	Nitrites			15
29	Organic N			17
30	Total Phosphorus	13		19
31	BOD-21			21
32	Extracted Matters	21		22
33	Cadmium	24		23
34	Chromium (3+) Cr-3	28		25
35	Cobalt			27
36	Nickle	30		28
37	Cyanide, Dissolved (CN)			30
38	Cyanide, Total (CN)	22		31
39	TKN			
40	Total Nitrogen			
41	Animal Fats			
42	Mercury (Hg)	23		
43	Formaldehyde			
44	DO, % Saturation	2		
45	COD-Cr	5		
46	Calcium, Ca	16		
47	Magnesium, Mg	17		
48	Fluorine, F	18		
49	Vanadium, V	32		
50	Silver, Ag	33		

(cont'd)

Table 3 (cont'd)

PARMCODE	PARMNAME	S-COE	B-MOE	Popozov
51	Radioactivity, Alpha	34		
52	Radioactivity, Beta	35		
53	Radium, Ra-226	36		
54	Uranium, U	37		
55	Tritium, H	38		
56	Strontium, Yttrium Sr, Y-90	39		
57	Cesium	40		
58	Coliform Bacteria	41		
59	Psychrophage Bacteria	42		
60	Faecal Bacteria	43		
61	Enterococci	44		
1000	Flow m3/day	1000		
1001	Flow m3/sec	1		
1002	Flow m3			

Table 4

FIELDS IN TABLE **DISCHX**—EXISTING DISCHARGERS

Field	Format	Units	Description
DISCHID	A10	-	User-defined unique identifier of a discharger.
TREATCLASS	N	-	Flow stream path number, linked to the emissions record in Table EMISSX.
COUNTRY	A1	-	B=Bulgaria, H=Hungary, R=Romania, S=Slovakia.
INDCODE	N	-	Industrial classification code; see Table INDCODES.
DISCHID-TO	A10	-	For an industrial discharger, the DISCHID of the municipal treatment plant; see definition sketch in Figure 13.

(cont'd)

Table 4 (cont'd)

Field	Format	Units	Description
BASIN	N	-	River basin ID number; see table RCHFILE.
SEG	N	-	River segment number within a river basin.
KM	N	-	River kilometer, from downstream end of segment.
POP	N	-	Population.
QTOT	N	cmd	Average daily flow; for municipalities, includes industrial, residential, infiltration/inflow, etc.
IALTID	N	-	Discharger ID in country's originating data base.
NAME	A55	-	Name and description of discharger.
TAXID	A10	-	Identifier of tax record in Table TAXES.
FINEID	A10	-	Identifier of fines record in Table FINES.
WQSTD	A5	-	Identifier of effluent quality standard record in Table WQSTDS.
TPID	A10	-	Identifier of wastewater treatment plant location; see definition sketch in Figure 13.
TPLEV	A5	-	Identifier of treatment level record in Table TPCST.
PTID	A10	-	Identifier of pretreatment plant location; see definition sketch in Figure 13.
PTLEV	A5	-	Identifier of pretreatment level in Table TPCST.
KM1	N	-	Cross-reference to river location using hydrologic sequence logic used within a particular country.
KM2	N	-	In Slovakia: the Mahovlic system in which KM1 is river km on the Danube from the downstream end,
KM3	N	-	and KM2,...,KM7 are river km on lower-order tributaries of the Danube.
KM4	N	-	
KM5	N	-	

(cont'd)

Table 4 (cont'd)

Field	Format	Units	Description
KM6	N	-	
KM7	N	-	
PROVINCE	A20	-	Province code or name for querying the data base.
BASINAUTH	A20	-	River basin authority.
RGNAUTH	A20	-	Regional authority.
WTRAUTH	A20	-	Water supply or water resources authority.
MUNAUTH	A20	-	Municipal authority.
PERMIT	A10	-	Identification code of the discharge permit.
PERMITAUTH	A20	-	Authority issuing the discharge permit.
QCAP	N	cmd	Existing capacity of wastewater treatment plant, either pretreatment or treatment, depending on INDCODE (=22 for municipalities).
QPEAK	N	cmd	Peak hydraulic flow into the wastewater treatment plant.
DAYSOP	N	days	Days per year of operation.
HRSOP	N	hours	Average hours per day of operation.
QRES	N	cmd	Residential flow.
QNONRES	N	cmd	Nonresidential flow; includes commercial, institutional, and infiltration/inflow.
QIND	N	cmd	Total industrial flow entering a municipal plant.
COLL-LENGTH	N	km	Length of sewers in the collection system.
STREET-LENGTH	N	km	Length of streets in a municipality.
PCTSEWERED	N	%	Percentage of population served by a municipal collection system.
WTRSPPLYID	A10	-	Identification code for the water supply authority.
WTRSPPLYID-TO	A10	-	Discharger ID that receives effluent as its (discharger's) supply.

(cont'd)

Table 4 (cont'd)

Field	Format	Units	Description
WTRSPPLYTYPE	A1	-	User identification of type of supply (surface supply, groundwater supply, treated, etc.).
WTRSPPLYBASIN	N	-	River basin identifier in RCHFILE.
WTRSPPLYSEG	N	-	River segment identifier in RCHFILE.
WTRSPPLYKM	N	-	River kilometer, from downstream end of reach.

Table 5

FIELDS IN TABLE **EMISSX**—EXISTING EMISSIONS

Field	Format	Units	Description
DISCHID	A10	-	Identification code of the discharger.
TREATCLASS	N	-	Flow path number of the discharger.
PARMCODE	N	-	Parameter code number from Table PARMTABL.
DATASRC	A6	-	Data source identification code.
QCCODE	A6	-	Quality-control code (user defined).
VALUE	N	[1]*	Value of the water quality parameter.
DATE	A6	-	Date (yymmdd) when the sample was taken.
TIME	A4	-	Time (hhmm) when the sample was taken.
DISCHTYPE	A2	-	Flow-routing code defined in Figure 13.

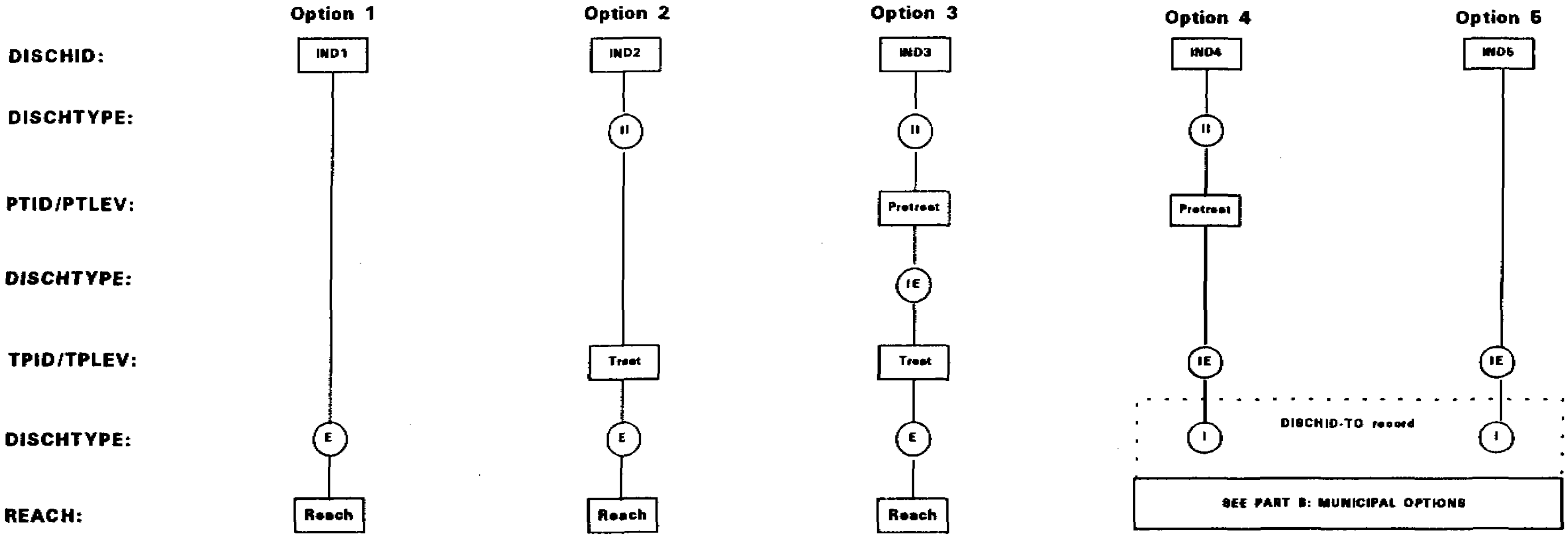
* The units (such as mg/L) for given parameters are defined in the DEMDESS table WQSTDS.

2.3.2 Routing of Industrial and Municipal Flows

The data records in the tables DISCH1 and EMISS1 contain several data fields that are used to route wastewater flows through industrial pretreatment, municipal wastewater treatment, and disposal. In general, a municipal wastewater collection system may serve many industries, and each industry may or may not pretreat its own wastewater. Industries may discharge treated or untreated wastewater into a river, they may practice land disposal of the effluent, or they may discharge into a municipal collection system for subsequent treatment in a municipal wastewater treatment plant. For these and other flow-routing options, the fields or variables in the tables DISCH1 and EMISS1 that determine flow routing are shown in the definition sketch in Figure 13.

Figure 13. Definition Sketch:
Flow-Routing Options

A. INDUSTRIAL EMISSIONS OPTIONS

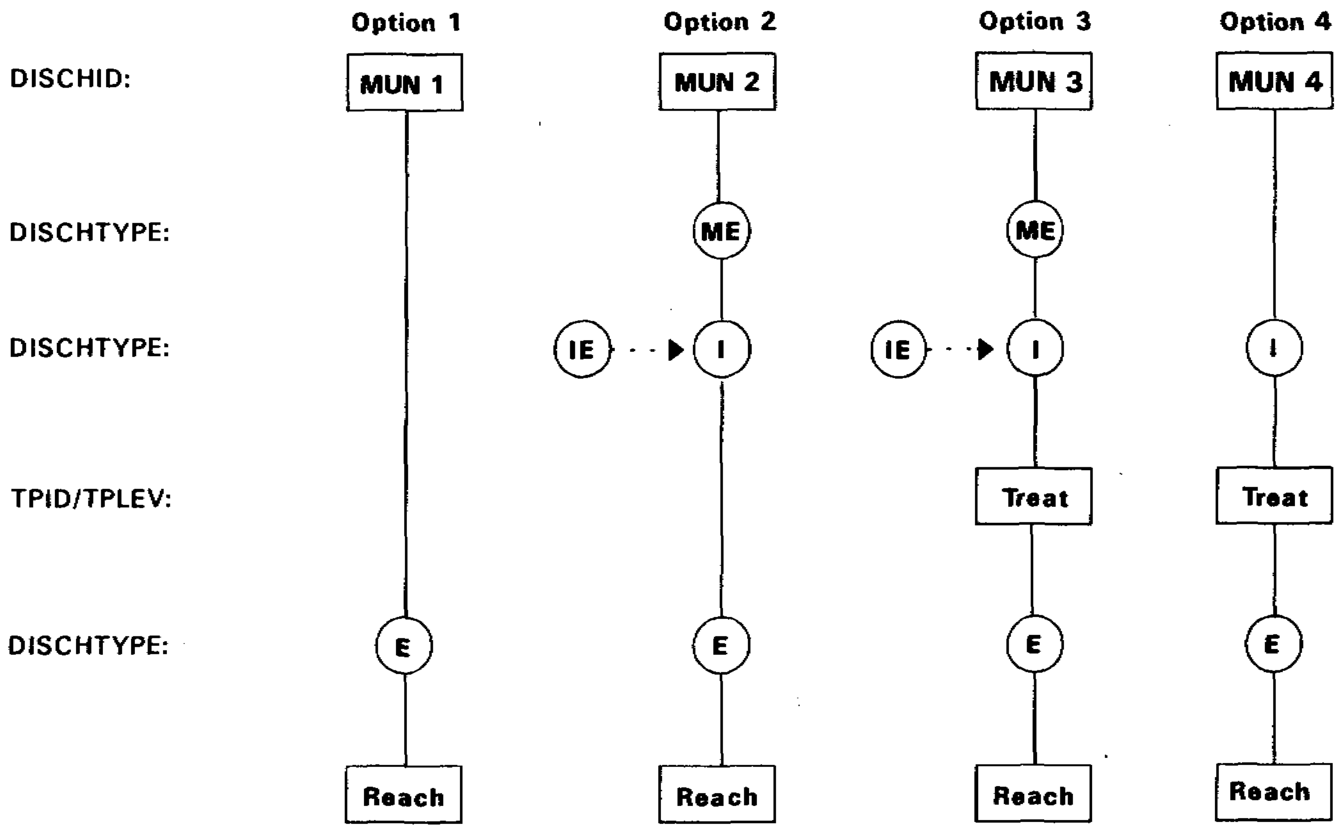


Option	DISCHTYPE in EMIS61 table: Given	Changed to	Computed	DISCHID-TO in DISCH1 table
1. No Treatment	E	–	–	Blank
2. Treatment	E	II	E	Blank
3. Pretreat/treat	E	II	IE, E	Blank
4. Pretreat/to Mun. STP	IE	II	IE	MUN 2,3 (See Part B)
5. To Municipal STP	IE	–	–	MUN 2,3 (See Part B)

Option	Pretreatment Comps: PTID/PTLEV	Cost, Comp.	Treatment Comps: TPID/TPLEV	Cost Comp.	Reach
1. No Treatment	Blank	No	Blank	No	Given
2. Treatment	Blank	No	Given	Yes	Given
3. Pretreat/treat	Given	Yes (Sum)	Given	Yes (Sum)	Given
4. Pretreat/to Mun. STP	Given	Yes	Blank	No	Blank
5. To Municipal STP	Blank	No	Blank	No	Blank

Figure 13. Definition Sketch:
Flow-Routing Options (continued)

B. MUNICIPAL EMISSIONS OPTIONS



Option	DISCHTYPE in EMIS61 Table: Given	Changed to	Computed	Treatment Compo: TPID/TPLEV	Cost Comp
1. No Treatment	E	–	–	Blank	No
2. Sum, No Treatment	M E	–	I, E	Blank	No
3. Sum, Treat	M E	–	I, E	Given	Yes
4. Treat	E	I	E	Given	Yes

In Figure 13, in the data base table DISCH1:

DISCHID and TREATCLASS
The discharger ID and the flow path number TREATCLASS together specify a single flow stream to be routed. Thus, a single industrial discharger may have more than one wastewater stream to be treated; for example, a stream requiring chemical treatment and a stream susceptible to biological treatment. Similarly, a municipality may be divided into existing or potential service areas for more than one wastewater treatment plant.

PTID and PTLEV
For an industrial pretreatment facility, PTID and PTLEV link to the fields TREATID and TREATLEV in the tables TPCHR and TPCST, to specify the cost curves and removal efficiencies for an industrial pretreatment plant. PTID is used to identify a plant-specific set of cost/efficiency data (with PTID = DISCHID), while TREATLEV is used to identify a generalized set of cost/efficiency data. Either one (PTID or PTLEV) or both of the fields can be given for a discharger, provided that a corresponding matching pair (TREATID and/or TREATLEV) is given in the records for TPCHR and TPCST.

TPID and TPLEV
For a municipal wastewater treatment plant, TPID and TPLEV link to the fields TREATID and TREATLEV in the tables TPCHR and TPCST, in the same manner as for the fields PTID and PTLEV. For an industrial discharger, TPID and TPLEV describe treatment given before discharge to a river, and may be preceded by pretreatment.

DISCHID-TO
The discharger ID for a municipal treatment plant. For a municipal discharger record, DISCHID = DISCHID-TO. For an industrial discharger record, DISCHID-TO links to the appropriate value of DISCHID for the municipal treatment plant.

BASIN, SEG, KM
The river reach into which the treated or untreated wastewater is discharged. The values for the river basin code, reach segment number, and river kilometers from the downstream end of the segment should be given for all dischargers for use in data retrievals on total emissions by river basin or river reach.

WTRSPPLYID-TO	For certain industrial dischargers (such as cooling-water emissions from thermal power plants), the effluent may be the water supply for another industry or water user. WTRSPPLYID-TO is the discharger ID of the recipient of the effluent.
DISPOSAL	This field allows the user to specify that the effluent is not a point source emission discharged into the river, but instead is converted into a nonpoint source (for example, by land application of the effluent).

In the data base table EMISS1:

DISCHID, TREATCLASS, and PARMCODE — A set of records is required for each discharger (DISCHID) and flow stream (TREATCLASS), with each record giving the data value for a given water quality parameter (PARMCODE).

DISCHTYPE — Sets of records by DISCHID, TREATCLASS, and PARMCODE may also be given to describe the location in the flow routing where the parameter values apply. Values for DISCHTYPE include:

- II Industrial influent, before treatment
- IE Industrial effluent, to municipal sewers
- ME Municipal effluent, not including industrial effluent
- I Combined industrial and municipal influent to a municipal wastewater treatment plant
- E Emission to a river, with or without treatment

As shown in the definition sketch in Figure 13, DISCHTYPE is the primary indicator for flow-routing options that the user wants to consider. These include five options for routing industrial emissions, and four options for routing municipal emissions. The purpose of the software design is to permit for each option computation of the appropriate flow routings, influent and effluent quantities, characteristics, treatment costs, emissions to rivers, and the loads removed by treatment. The options shown correspond to the following requirements:

1. Each discharger described in DISCH1 has one set of emissions records in EMISS1, corresponding to the scenario and flow-routing option selected by the user. It is assumed that this emissions record represents a copy, an extrapolation, or an averaging of records

representing historical or periodic emissions, obtained from the EMISSX table. In all of these cases, the records must have a value of DISCHTYPE that is either "E" for emissions to a river, "IE" for industrial effluent, or "ME" for municipal effluent.

2. A script (programmed procedure) is used to add industrial and municipal emissions ("IE" and "ME") to calculate "I," the combined influent to a municipal wastewater treatment plant. Similarly, a script is used to calculate the costs for pretreatment and treatment facilities, and to create new treated-effluent records (DISCHTYPE = "E").

3. The fields QTOT and QCAP in the discharger record in DISCH1 contain the average daily flow and required treatment capacity associated with the scenario and flow-routing option. For existing facilities to be expanded or upgraded under the given scenario, the capacity QCAP and treatment level (PTLEV and/or TPLEV) are stored in the same discharger record in the table DISCHX.

2.4 Reach Subsystem

The Reach File is the standardized river network used in DEMDESS. The design is based on the United States Environmental Protection Agency (EPA) River Reach File. This design has been implemented in the U.S. on a national level in three versions, with the latest version containing more than 3 million river segments. In the last 12 years, the design has been proven to be a very effective and efficient tool for identifying river reaches, linking diverse data bases to the river network, and performing hydrologic retrievals and analyses.

The Reach File contains one record for each reach, with a reach generally defined as the stretch of river going from confluence to confluence. The fields in Table RCHFILE are defined in Table 6.

Table 6

FIELDS IN TABLE **RCHFILE**—RIVER REACH NETWORK

Field	**Format**	**Units**	**Description**
BASIN	N	-	The numerical code designation for the major river basin.
SEG	N	-	The segment number of the river. All reaches of a river have the same SEG value.
KM	N	Km	The river kilometer of the downstream point of the reach.

(cont'd)

Table 6 (cont'd)

Field	Format	Units	Description
TYPE	A1	-	The reach type code. S=start T=terminal R=regular
USBASIN1	N	-	The BASIN code of the mainstem upstream reach.
USSEG1	N	-	The SEG code of the mainstem upstream reach.
USKM1	N	Km	The KM value of the mainstem upstream reach.
USBASIN2	N	-	The BASIN code of the tributary upstream reach.
USSEG2	N	-	The SEG code of the tributary upstream reach.
USKM2	N	Km	The KM value of the tributary upstream reach.
USBASIN3	N	-	The BASIN code of a second tributary reach; usually not used.
USSEG3	N	-	The SEG code of a second tributary reach; usually not used.
USKM3	N	Km	The KM value of a second tributary reach; usually not used.
DSBASIN	N	-	The BASIN code of the reach downstream.
DSSEG	N	_	The SEG code of the reach downstream.
DSKM	N	Km	The KM value of the reach downstream.
CBASIN	N	-	The BASIN code of the downstream complement reach, i.e., the other reach entering the d.s. junction.
CSEG	N	-	The SEG code of the downstream complement reach.
CKM	N	Km	The KM value of the downstream complement reach.
LEV	N	-	The stream level of the reach.
J	N	-	The LEV value of the reach downstream.

(cont'd)

Table 6 (cont'd)

Field	Format	Units	Description
SEQNO	N	-	The hydrologic sequence number; important for sorting in hydrologic order.
SEGL	N	Km	The length of the reach in kilometers.
WQSTATUS	A3	-	The current water quality status; links to WQSTDS.
WQSTD	A3	-	The current water quality standard for the reach; links to WQSTDS.
LOWQ	N	m^3	The low flow value for the reach.
COUNTRY	A1	-	The country code for the reach (the country the reach is in).
PROVINCE	N	-	The province code for the reach.
PATHMILE	N	Km	The same as KM, except in Bulgaria; PATHMILE will be eliminated in future DEMDESS versions.
RCHNAME	A30	-	The name of the river.
RCHJUNCDESC	A52	-	A description of the junction at the downstream end of the reach.
BPATTERN	A20	-	Used for display of schematic diagrams of the river network; sort by SEQNO to use BPATTERN.
KM1	N	Km	KM1-KM7 are the host country river kilometer index values. A first step in building the Reach File
KM2	N	Km	is to load the KM1-KM7 fields and then run the DEMDESS Reach File maker software to build the full
KM3	N	Km	Reach File. After the Reach File is built, the KM1-KM7 field values are used to assign other tables
KM4	N	Km	to the Reach File (such as the DISCHX and WQSITES tables).
KM5	N	Km	
KM6	N	Km	
KM7	N	Km	

(cont'd)

Table 6 (cont'd)

Field	Format	Units	Description
CODUL1	S	-	CODUL1-CODUL7 are used for the Romanian values that define the river network. These values
CODUL2	S	-	are used in a similar manner as the KM1-KM7 fields.
CODUL3	S	-	
CODUL4	S	-	
CODUL5	S	-	
CODUL6	S	-	
CODUL7	S	-	
USKM	N	Km	The upstream river kilometer value from the Romanian river network.

In Table 6, the reaches are defined by a three-level key consisting of the following:

1. BASIN—a numerical field that is common to all reaches in the major watershed delineation.
2. SEG—a segment number that is unique for each river in the basin.
3. KM—the kilometer point along the river, measured from the downstream end. Each reach is identified by the KM value of the downstream end of the reach. A linked table, such as DISCHX, will contain the actual KM point of the discharger.

The three-level reach identifier key permits unique identification of any point along any river in the Reach File. A point, such as an emission location or water quality station, is linked to the Reach File by using the BASIN-SEG-KM. Any data base that contains the proper BASIN-SEG-KM of the site can be linked to the Reach File and integrated with any other data bases that are also linked.

Each reach record contains many other data elements that are needed for identification, indexing, reporting, and routing. The identification data elements include the reach name, description of the downstream junction, a hydrologic branching pattern, the country, and the province. Data for indexing other data bases to the Reach File include the river kilometer values used by the host country data bases.

Also included in each reach record are the reach numbers (BASIN-SEG-KM) of the connecting reaches at the upstream and downstream ends of the reach. The connectivity reaches mean that, starting at any point on a reach, the Reach File can be traversed upstream or downstream at will. By incorporating reach linkages to other data bases, requests such as "find all dischargers upstream of this drinking water intake" can be met. Fulfilling this type of request has been one of the major uses of the Reach File in the United States.

The Reach File connectivity has been carefully designed so that only one reach record at a time need be in computer memory. This is very important for DEMDESS because it means that the limitation on the size of the river network being analyzed depends on disk space rather than internal computer memory ("RAM").

For hydrologic routing, there are some additional variables that permit implementation of very fast, compact routing routines. These routing routines also require that only one reach record at a time be in memory. The specialized routing variables are as follows:

SEQNO—the hydrologic sequence number. When the Reach File is processed in SEQNO order, the reaches are in the order needed to route and model going from upstream to downstream. A good way to see the impact of the SEQNO is to sort the Reach File SEQNO and display the branching pattern (BPATTERN) along with the NAME and RCHJUNCDESC.

LEV—the stream level, with the Danube being level 1, a tributary to the Danube being level 2, and so on. For example, the Váh is a level 2 stream and the Nitra, which is a tributary of the Váh, is level 3.

J—the level of the reach downstream. For instance, the most downstream reach of the Nitra River has a LEV=3 and J=2. The intermediate reaches of the Nitra River have LEV=3 and J=3.

TYPE—a one-character code that distinguishes between a start reach (TYPE=S), a terminal reach (TYPE=T), and a regular reach (TYPE=R). A start reach has no reaches above it. A terminal reach has no reaches below it, for example, the last reach on the Danube. A regular reach has reaches upstream and downstream of it.

SEQNO, LEV, and J permit a full pollutant routing using one reach at a time. The LEV and J values are used as array indexes to accumulate loadings and store them for use when the proper level is reached. The accumulating arrays need to be dimensioned at the maximum level in the Reach File; the maximum level encountered in the national river network data bases is 7. An example of a Reach File pollutant routing algorithm is in the example script RCHROUT.

For load routing and modeling purposes, it is useful to determine and add the reach SEQNO value for each discharger loading. At the same time, the reach LEV value should be added, and a J value equal to the LEV should be set. This enhanced loadings data base can then be interleaved with the Reach File. This interleaved data file can then be sorted by SEQNO-descending KM and will, in effect, create a routable Reach File with records between confluences as well as between discharge points. This file will permit pollutant accumulation and decay at and between all points of change.

In the initial DEMDESS, the basic one-dimensional river network is represented. The full Reach File design includes capabilities for lakes, reservoirs, and divergent junctions. These extended capabilities are not implemented in the initial DEMDESS Reach File.

2.5 Treatment Subsystem

The Treatment Subsystem of DEMDESS consists of three Paradox® tables named TPCHR, TPCST, and CURRCNVT. Sample listings are shown in Tables 7, 8, and 9. These data bases define various treatment levels (technologies) applicable to the treatment of emissions. The pollutant removal efficiencies or residual pollutant concentrations for each treatment level are specified and used to compute treated effluent concentrations. Coefficients in generalized cost equations and unit costs are also defined, for use in computing capital and annual costs.

Table TPCHR defines the treatment characteristics (efficiencies) for various treatment levels (technologies), and Table TPCST defines the constants and variables for cost functions for the same treatment levels. Table CURRCNVT contains the data for conversion of costs expressed in different country currencies. These three tables and their use are discussed below.

2.5.1 Treatment Characteristics

The fields in TPCHR (Treatment Process Characteristics, Table 7) are described below.

Table 7

SAMPLE LISTING OF TABLE TPCHR—TREATMENT CHARACTERISTICS

TREATID	TREATLEV	PARMCODE	EFFLCONC	PCTEFFL	DATASRC
	NONE	1		1.00000	
	NONE	4		1.00000	
	NONE	30		1.00000	
	NONE	39		1.00000	
	NONE	40		1.00000	
	SAS02	1		.10000	
	SAS02	4		.10000	
	SAS02	30		.80000	
	SAS02	39		1.00000	

(cont'd)

Table 7 (cont'd)

TREATID	TREATLEV	PARMCODE	EFFLCONC	PCTEFFL	DATASRC
	SAS02	40		1.00000	
	SAS02	1		.05000	
	SAS02	4		.10000	
	SAS02	30		.80000	
	SAS02	39		.20000	
	SAS02	40		1.00000	
	PR01	1		.65000	
	PR01	4		.50000	
	PR01	30		.90000	
	PR01	39		1.00000	
	PR01	40		1.00000	
	PR02	1		.45000	
	PR02	4		.50000	
	PR02	30		.25000	
	PR02	39		1.00000	
	PR02	40		1.00000	

TREATID

The user-defined code for TREATID is optional, and this field would normally be blank when the user wants to use the generalized data on removal efficiencies. For wastewater treatment plants for which the removal efficiencies are known, TREATID is used to link the associated record in the table TPCHR with the corresponding discharger record in the table DISCH1.

TREATLEV

The designations in TREATLEV (Treatment Level) represent various treatment levels or technologies both for industrial and municipal applications. This field is also used in the TPCST table (described later) as the key field. The treatment level of NONE represents base conditions without treatment and may represent a scenario alternative for certain sources. The other levels are treatment such as Single Stage Conventional Activated Sludge (SAS01) or Cyanide Destruction (IS01). Treatment level designations applicable to industrial wastewater are preceded with the letter "I." The user can add other treatment levels and technologies.

PARMCODE

PARMCODE (Parameter Code) contains the codes for BOD_5 (1) and others, such as Suspended Solids (4). PARMCODE is defined under the Emissions Subsystem.

EFFLCONC

EFFLCONC (Effluent Concentration) gives the absolute value of the pollutant in mg/L that remains in the effluent. This method of describing treatment efficiency is generally applicable to industrial wastewater treatment systems. (For example, effluent concentration of heavy metals primarily depends on treatment process pH rather than on influent concentration). If the user wants to specify EFFLCONC, he or she should leave the PCTEFFL field blank.

PCTEFFL

PCTEFFL (Percent Effluent) gives the ratio of the parameter remaining in the effluent compared with the influent level. For parameters that are not affected by treatment, the ratio is 1.0. This type of indicator is common for municipal wastewater treatment plants.

DATASRC

DATASRC (Data Source) gives the reference from where the efficiency data are taken.

2.5.2 Treatment Costs and Other Outputs

The Treatment Subsystem includes the coefficients for a set of functions for each treatment level with the flow parameters QTOT (Average Daily Flow) or QCAP (Plant Capacity) in the DISCH1 table as the (input) variable. The outputs of the equations are

- estimated construction cost;
- annual material and supplies cost;
- annual energy use; and
- annual manpower.

The functions reside in PAL scripts but the constants and variables for the functions are given in Table TPCST (Treatment Process Cost, Table 8). TPCST also contains data for financial analysis and sludge production functions, which also reside in PAL scripts. The fields used in TPCST and the related functions are described below. A more detailed explanation of cost functions and unit prices is provided in Appendix C.

Table 8

SAMPLE LISTING OF TABLE **TPCST**—TREATMENT COSTS

```
TREATID:          TREATLEV: NONE   DATASRC:
TREATDESC: None
CURRENCY:               CSTDATE: 6/1992
A:   0.0  B: 0.000   Construction  (1000     )=A*(Cap.)^B
I:  0.0  J: 0.000   Materials (1000    /yr)=I*(Flow)^J
K:   0.0  L: 0.000   Energy    (1000  kwh/yr)=K*(Flow)^L
H:   0.0  M: 0.000   Labor(1000  man-hours/yr)=H*(Flow)^M
E&CPCT:   0.0     SUBSIDYPCT:  0.0
INTRATE: 0.0 %    YRSAMORT:      0
SLUDGEP: 0.000     SLUDGEQ:   0.000
```

```
TREATID:          TREATLEV: SAS01   DATASRC:
TREATDESC: Conventional  Activated Sludge - Single
        Stage
CURRENCY: USD           CSTDATE: 6/1992
A: 1035.0   B: .800   Construction  (1000  USD)=A*(Cap.)^B
I:  20.0   J:  .850   Materials (1000  USD/yr)=I*(Flow)^J
K:  60.0   L: .900   Energy    (1000  kwh/yr)=K*(Flow)^L
H:  18.0   M: .500   Labor(1000  man-hours/yr)=H*(Flow)^M
E&CPCT:  20.0     SUBSIDYPCT: 75.0
INTRATE: 10.0 %    YRSAMORT:     20
SLUDGEP: .190     SLUDGEQ:    .480
```


(cont'd)

Table 8 (cont'd)

TREATID:	TREATLEV: SAS02 DATASRC:
TREATDESC: Conventional Activated Sludge - Two Stage	
CURRENCY: USD	CSTDATE: 6/1992
A: 1500.0 B: .820	Construction (1000 USD)=A*(Cap.)^B
I: 30.0 J: .840	Materials (1000 USD/yr)=I*(Flow)^J
K: 100.0 L: .900	Energy (1000 kwh/yr)=K*(Flow)^L
H: 25.0 M: .550	Labor(1000 man-hours/yr)=H*(Flow)^M
E&CPCT: 20.0	SUBSIDYPCT: 75.0
INTRATE: 10.0 %	YRSAMORT: 20
SLUDGEP: .190	SLUDGEQ: .480

TREATID:	TREATLEV: PR01 DATASRC:
TREATDESC: Primary Sedimentation	
CURRENCY: USD	CSTDATE: 6/1992
A: 500.0 B: .840	Construction (1000 USD)=A*(Cap.)^B
I: 8.0 J: .830	Materials (1000 USD/yr)=I*(Flow)^J
K: 12.0 L: .790	Energy (1000 kwh/yr)=K*(Flow)^L
H: 7.0 M: .650	Labor(1000 man-hours/yr)=H*(Flow)^M
E&CPCT: 20.0	SUBSIDYPCT: 75.0
INTRATE: 10.0 %	YRSAMORT: 20

(cont'd)

Table 8 (cont'd)

TREATID:	TREATLEV: PR02 DATASRC:
TREATDESC: Primary Sedimentation with Chemical Coagulation	
CURRENCY: USD	CSTDATE: 6/1992
A: 650.0 B: .850	Construction (1000 USD)=A*(Cap.)^B
I: 10.0 J: .820	Materials (1000 USD/yr)=I*(Flow)^J
K: 20.0 L: .780	Energy (1000 kwh/yr)=K*(Flow)^L
H: 10.0 M: .630	Labor(1000 man-hours/yr)=H*(Flow)^M
E&CPCT: 20.0	SUBSIDYPCT: 75.0
INTRATE: 10.0 %	YRSAMORT: 20

TREATLEV

Treatment Level, as in the DISCH1 table.

TREATDESC

TREATDESC (Treatment Description) briefly describes treatment levels or technologies.

DATASRC

Data Source.

CURRENCY

CURRENCY defines the type of currency that applies to the costs. USD under the CURRENCY field defines that cost output is in United States dollars.

CSTDATE

CSTDATE (Cost Date) refers to the date for which the costs are valid. All costs in one table should be valid for one date, although the date itself can be approximate, such as 1991 or 6/1991 (mid-1991). In the example table, TPCST, the cost data is valid for mid-1992; therefore, 6/1992 is entered.

A, B

A is the coefficient of proportionality and B is the exponent of economy of scale in the Construction Cost Formula $C = A Q^{\wedge B}$, where C represents cost in 1,000 currency units and Q represents flow in 1,000 cmd. (Q represents average daily flow or plant capacity and has the same units in all formulas).

I, J

I is the coefficient of proportionality and J is the exponent of economy of scale in the Annual Material & Supplies Cost Formula $S = I (QA)^{\wedge J}$, where S represents the cost in 1,000 currency units per year and QA represents the average flow over the planning period.

K, L

K is the coefficient of proportionality and L is the exponent of economy of scale in the Annual Energy Consumption Formula $KWH = K(QA)^{\wedge L}$, where KWH represents 1,000 kilowatt-hours per year. The cost is computed by multiplying the consumption with the local energy cost that resides in the CURRCNVT table, which is described later.

H, M

H is the coefficient of proportionality and M is the exponent of economy of scale in the Annual Labor Formula $L = H(QA)^{\wedge M}$, where L represents man-hours per year. The annual labor cost is computed by multiplying man-hours with the local labor cost per hour that resides in the CURRCNVT table.

E&CPCT

E&CPCT (Engineering and Contingency) represents costs such as engineering, a contingency allowance, and any other costs associated with construction that are not normally included in the construction cost bid by a contractor. This cost is represented as a percentage of the construction cost. The sum of construction cost C and E&CPCT represents the capital construction cost (or investment). Practices in each country may result in a different percentage for E&CPCT. For example, turnkey design/construct contracts versus phased contracts would contain some of the engineering cost within the construction cost. Therefore, the percentage should be determined by the user to reflect local conditions. In any case, capital construction cost should include all contractors' costs of construction, including overhead and profit; costs of land, relocation, and right-of-way and easement acquisition; design engineering, field exploration, and engineering services during construction; administrative and legal services; start-up costs such as operator training; and interest during construction. Contingency allowances consistent with the level of complexity and detail of the cost estimates should be included.

SUBSIDYPCT

SUBSIDYPCT (Subsidy Percentage) gives the percentage of the capital cost that may be paid by subsidies from the national government.

INTRATE

INTRATE (Interest Rate) represents the discount rate or opportunity value of money as used in standard engineering economics. INTRATE is used in PAL functions that capitalize annual costs or annualize capital costs.

YRSAMORT

YRSAMORT (Years Amortization) represents the planning period in years. YRSAMORT is used in PAL functions that capitalize annual costs or annualize capital costs over the planning period. In the examples, the service lives of facilities are assumed to be equal to the planning period, and therefore salvage value or replacement cost has been ignored to simplify the examples. More detailed economic and financial analysis should be used where justified.

SLUDGEP

SLUDGEP is the coefficient of proportionality in the formula SLD = (SLUDGEP)(QA), where SLD represents tons of dry solids per year. The formula is applicable to activated sludge processes, and the coefficient depends on the type of activated sludge process. This field is included in Table TPCST because the ultimate disposal of sludges (transportation, landfilling, land application, and so on) is a secondary but major cost item.

SLUDGEQ

SLUDGEQ is the coefficient of proportionality in the formula SLC = (SLUDGEQ)(QA), where SLC represents tons of sludge cake or liquid per year from the facility. The coefficient depends on the degree of sludge dewatering achieved by the facility and addition of sludge conditioning chemicals, which also contribute to the solids. SLUDGEQ and SLUDGEP are related so that (SLUDGEQ)(% SOLIDS) = (SLUDGEP).

2.5.3 Local Cost Computation and Conversions

In those cases where costs are computed in one currency (as in examples where U.S. cost functions are used), conversion of these costs to local currency is done by using the data in Table CURRCNVT. Annual cost items of labor and energy are first computed in man-hours and kilowatt-hours, respectively. The local costs for these items are computed by using local rates rather than converting costs.

It should be noted that when cost functions from other countries are used, they should be adjusted by using appropriate indexes prior to inputting data into Table TPCST. In the example given in Table 9, the U.S. costs are assumed to be applicable in Bulgaria as "world prices." These costs are default values and their use should be eventually replaced by local costs. The recommended approach is development of local coefficients and exponents for the cost functions indexed to a recent date. (The definition of cost functions requires at least two cost data points, but more are desirable). The fields in the CURRCNVT table (Table 9) are described below.

Table 9

SAMPLE LISTING OF TABLE **CURRCNVT**—CURRENCY CONVERSIONS AND LOCAL UNIT PRICES

CURRENCY	VALUE	YEAR	COUNTRY	COUNTRY-NAME
USD	1	1992	U	United States
Ft	75	1992	H	Hungary
LEV	20	1992	B	Bulgaria
LEI	358	1992	R	Romania
DM	2	1992	G	Germany
Kr	27	1992	S	Slovakia
Sch	11	1992	A	Austria
ECU	1	1992	E	European Curr. Unit

CURRENCY	COUNTRY	LABORCOST	ENERGYCOST	MATLCOST
USD	U			
Ft	H			
LEV	B	40	2	20
LEI	R			
DM	G			
Kr	S			
Sch	A			
ECU	E			

CURRENCY

CURRENCY lists the name of currency for each country (e.g., U.S. dollars, leva, etc.).

VALUE

VALUE gives the ratio of the country currency to the U.S. dollar. This should be based on free market values.

YEAR

YEAR gives the date for which the VALUE is valid.

COUNTRY

COUNTRY gives a country letter code (U, B, etc.).

COUNTRY-NAME

COUNTRY-NAME spells out the country name.

LABOR COST

LABOR COST gives the local labor cost in local currency per hour. This cost should reflect actual market value, and subsidies to labor, such as housing, should be added to compute actual cost in an economic analysis.

ENERGY COST

ENERGY COST gives the local cost per 1,000 kilowatt-hours. The cost should be actual cost without subsidies. (Actual costs are necessary in a true economic analysis, although subsidized costs can be used in a cash flow analysis if the subsidies are expected to be long-term.)

MATLCOST

MATLCOST (Materials Cost) converts the annual cost of supplies and materials in the base currency into local currency units, and thus combines the effect of the exchange rate with an index of local cost of materials.

2.6 Regulatory Subsystem

The Regulatory Subsystem contains five Paradox® tables: FINES, PERMITS, TAXES, WQSTDESC, and WQSTDS. Sample listings and/or definitions of fields for these are shown in Tables 10 through 14. The fields in the FINES table (Table 10) are country specific. For example, the prices charged to polluters in Bulgaria depend on the concentration levels of 26 water quality parameters. There is no charge when the concentration level for a given parameter is less than that dictated by the stream quality standard, which in turn varies by river reach. Above the level allowed by the standard, a different price is charged for each parameter, in leva/kg of pollutant. In the sample application to the Jantra basin, the only prevalent measurements of emissions were BOD_5 measurements and TSS measurements; even for these two parameters, it was necessary to assume concentrations of BOD_5 and TSS for several of the smaller municipalities in the basin. For TSS the allowable standard is very high, generally greater than 1,000 mg/L. Given the general lack of data on the remaining 24 parameters, the fine for BOD_5 emissions was raised from 2 leva/kg to a trial value of 25 leva/kg in the Jantra basin sample application.

Table 10

SAMPLE LISTING OF TABLE **FINES**—CHARGES BASED ON EMISSIONS

FINEID	WQSTD	PARMCODE	FINE	FINEUNIT	CURRENCY
B1992		1	25	Kg	LEV
B1992		2	2	Kg	LEV
B1992		3	1	Kg	LEV
B1992		4	4	Kg	LEV
B1992		5	4	Kg	LEV
B1992		6	1	Kg	LEV
B1992		9	120	Kg	LEV
B1992		10	28	Kg	LEV
B1992		11	160	Kg	LEV
B1992		12	8	Kg	LEV
B1992		13	40	Kg	LEV
B1992		14	8	Kg	LEV
B1992		15	240	Kg	LEV
B1992		16	60	Kg	LEV
B1992		17	240	Kg	LEV
B1992		18	24	Kg	LEV
B1992		19	2	Kg	LEV
B1992		20	8	Kg	LEV
B1992		24	2	Kg	LEV
B1992		28	200	Kg	LEV
B1992		33	1200	Kg	LEV
B1992		34	24	Kg	LEV

(cont'd)

Table 10 (cont'd)

FINEID	WQSTD	PARMCODE	FINE	FINEUNIT	CURRENCY
B1992		36	60	Kg	LEV
B1992		37	160	Kg	LEV
B1992		41	3	Kg	LEV
B1992		42	1200	Kg	LEV
H1992	HC1	2	1	Kg	Ft
H1992	HC2	2	1	Kg	Ft
H1992	HC3	2	1	Kg	Ft
H1992	HC4	2	1	Kg	Ft
H1992	HC5	2	1	Kg	Ft
H1992	HC6	2	1	Kg	Ft
H1992	HE1	2	1	Kg	Ft
H1992	HE2	2	1	Kg	Ft
H1992	HE3	2	1	Kg	Ft
H1992	HE4	2	1	Kg	Ft
H1992	HE5	2	1	Kg	Ft
H1992	HE6	2	1	Kg	Ft
H1992	HE1	4	1	Kg	Ft
H1992	HE2	4	1	Kg	Ft
H1992	HE3	4	1	Kg	Ft
H1992	HE4	4	1	Kg	Ft
H1992	HE5	4	1	Kg	Ft
H1992	HE6	4	1	Kg	Ft
H1992	HC1	5	5	Kg	Ft
H1992	HC2	5	5	Kg	Ft
H1992	HC3	5	5	Kg	Ft
H1992	HC4	5	5	Kg	Ft

(cont'd)

Table 10 (cont'd)

FINEID	WQSTD	PARMCODE	FINE	FINEUNIT	CURRENCY
H1992	HC5	5	5	Kg	Ft
H1992	HC6	5	5	Kg	Ft
H1992	HE1	5	5	Kg	Ft
H1992	HE2	5	5	Kg	Ft
H1992	HE3	5	5	Kg	Ft
H1992	HE4	5	5	Kg	Ft
H1992	HE5	5	5	Kg	Ft
H1992	HE6	5	5	Kg	Ft
H1992	HE1	6	1	Kg	Ft
H1992	HE2	6	1	Kg	Ft
H1992	HE3	6	1	Kg	Ft
H1992	HE4	6	1	Kg	Ft
H1992	HE5	6	1	Kg	Ft
H1992	HE6	6	1	Kg	Ft
H1992	HC1	8	1	Kg	Ft
H1992	HC2	8	1	Kg	Ft
H1992	HC3	8	1	Kg	Ft
H1992	HC4	8	1	Kg	Ft
H1992	HC5	8	1	Kg	Ft
H1992	HC6	8	1	Kg	Ft
H1992	HC1	11	50	Kg	Ft
H1992	HC2	11	50	Kg	Ft
H1992	HC3	11	50	Kg	Ft
H1992	HC4	11	50	Kg	Ft
H1992	HC5	11	50	Kg	Ft

(cont'd)

Table 10 (cont'd)

FINEID	WQSTD	PARMCODE	FINE	FINEUNIT	CURRENCY
H1992	HC6	11	50	Kg	Ft
H1992	HE1	11	50	Kg	Ft
H1992	HE2	11	50	Kg	Ft
H1992	HE3	11	50	Kg	Ft
H1992	HE4	11	50	Kg	Ft
H1992	HE5	11	50	Kg	Ft
H1992	HE6	11	50	Kg	Ft
H1992	HC1	12	1	Kg	Ft
H1992	HC2	12	1	Kg	Ft
H1992	HC3	12	1	Kg	Ft
H1992	HC4	12	1	Kg	Ft
H1992	HC5	12	1	Kg	Ft
H1992	HC6	12	1	Kg	Ft
H1992	HE1	12	2	Kg	Ft
H1992	HE2	12	2	Kg	Ft
H1992	HE3	12	2	Kg	Ft
H1992	HE4	12	2	Kg	Ft
H1992	HE5	12	2	Kg	Ft
H1992	HE6	12	2	Kg	Ft
H1992	HC1	20	60	Kg	Ft
H1992	HC2	20	60	Kg	Ft
H1992	HC3	20	60	Kg	Ft
H1992	HC4	20	60	Kg	Ft

(cont'd)

Table 10 (cont'd)

FINEID	WQSTD	PARMCODE	FINE	FINEUNIT	CURRENCY
H1992	HC5	20	60	Kg	Ft
H1992	HC6	20	60	Kg	Ft
H1992	HC1	24		??	Ft
H1992	HC2	24		??	Ft
H1992	HC3	24		??	Ft
H1992	HC4	24		??	Ft
H1992	HC5	24		??	Ft
H1992	HC6	24		??	Ft
H1992	HE1	24	5	??	Ft
H1992	HE2	24	5	??	Ft
H1992	HE3	24	5	??	Ft
H1992	HE4	24	5	??	Ft
H1992	HE5	24	5	??	Ft
H1992	HE6	24	5	??	Ft
H1992	HE1	30	40	Kg	Ft
H1992	HE2	30	40	Kg	Ft
H1992	HE3	30	40	Kg	Ft
H1992	HE4	30	40	Kg	Ft
H1992	HE5	30	40	Kg	Ft
H1992	HE6	30	40	Kg	Ft

In Hungary, the sample listing in Table 10 shows that the level of fines depends upon both the water quality standard applicable to a discharger, and upon the values of 10 water quality parameters (PARMCODE=2, 4, 5, 6, 8, 11, 12, 20, 24, and 30).

Data on discharge permits can be stored in the PERMITS table (Table 11), and tied to the data on dischargers by the key fields PERMIT, PERMITAUTH, and TREATCLASS, which also appear in the records of the DISCHX table. Although not implemented in the Initial System due to a lack of data, the PERMITS table allows for the future possibility that data on negotiated discharge permits for industries could form the basis for one scenario option on industrial emissions.

Table 11

FIELDS IN TABLE **PERMITS**—DATA ON DISCHARGE PERMITS

Field Name	**Format**	**Units**	**Description**
PERMIT	A10	-	Identification code of the discharge permit.
TREATCLASS	N	-	Treatment class or flow path, as in DISCH1 table.
PERMITAUTH	A20	-	Authority responsible for the discharge permit.
PARMCODE	N	-	Parameter code, as in PARMTABL.
PERMITLIMIT	N	-	Maximum allowable limit, or lower limit of range.
PERMITLIMIT2	N	-	Upper limit of range (such as pH).
UNITS	A4	-	Units for the parameter, as in the discharge permit.

The TAXES table (Table 12) has been developed thus far only for Bulgaria, where polluters are required to pay for the quantity of wastewater flow that is discharged untreated, with one price for municipalities (0.5 leva/cu m) and another for industries (0.8 leva/cu m), as shown in Table 12.

Table 12

SAMPLE LISTING OF TABLE **TAXES**—CHARGES BASED ON WASTEWATER FLOWS

TAXID	WQSTD	PARMCODE	TAX	TAXUNIT	CURRENCY
BM1		1002	.5	cu m	LEV
BI1		1002	.8	cu m	LEV
BT1		1002	0.0	cu m	LEV

For Bulgaria and Hungary, the descriptions of the water quality standards are given in Table 13, and the limits for the various parameters are given in Table 14. In both cases, some of the standards refer to limits on emissions and other standards refer to limits on ambient stream quality; these data bases have been designed intentionally with both purposes in mind.

Table 13

SAMPLE LISTING OF TABLE **WQSTDESC**—DESCRIPTION OF WATER QUALITY STANDARDS

WQSTD	WQSTDDESC
B1	Bulgarian Instream Standard I - Drinking
B2	Bulgarian Instream Standard II - Irrigation
B3	Bulgarian Instream Standard III - Polluted
BCN	Bulgarian Collector Standard - No Treatment Plant Exists
BCT	Bulgarian Collector Standard - Treatment Plant Exists
H1A	Hungarian Instream Std I - Excellent Biological Integrity
H1B	Hungarian Instream Std I - Acceptable Biological Integrity
H2A	Hungarian Instream Std II - Excellent Drinking
H2B	Hungarian Instream Std II - Acceptable Drinking
H3A	Hungarian Instream Std III - Excellent Industrial

(cont'd)

Table 13 (cont'd)

H3B	Hungarian Instream Std III - Acceptable Industrial
H4A	Hungarian Instream Std IV - Excellent Irrigation
H4B	Hungarian Instream Std IV - Acceptable Irrigation
H5A	Hungarian Instream Std V - Excellent Fishery
H5B	Hungarian Instream Std V - Acceptable Fishery
H6A	Hungarian Instream Std VI - Excellent Multi-Purpose
H6B	Hungarian Instream Std VI - Acceptable Multi-Purpose
HE1	Hungarian Emission Std I for Instream Std H1
HE2	Hungarian Emission Std II for Instream Std H2
HE3	Hungarian Emission Std III for Instream Std H3
HE4	Hungarian Emission Std IV for Instream Std H4
HE5	Hungarian Emission Std V for Instream Std H5
HE6	Hungarian Emission Std VI for Instream Std H6
HC1	Hungarian Collector Std I for Instream Std H1
HC2	Hungarian Collector Std II for Instream Std H2
HC3	Hungarian Collector Std III for Instream Std H3
HC4	Hungarian Collector Std IV for Instream Std H4
HC5	Hungarian Collector Std V for Instream Std H5
HC6	Hungarian Collector Std VI for Instream Std H6

Table 14

SAMPLE LISTING OF TABLE **WQSTDS**—WATER QUALITY STANDARDS BY PARAMETER

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
B1	1	5		mg/L
B1	2	25		mg/L
B1	3	30		mg/L
B1	4	700		mg/L
B1	24	7	9	pH
B1	14	0		mg/L
B2	14	1		mg/L
B3	14	2		mg/L
B1	5	0		mg/L
B2	5	2		mg/L
B3	5	5		mg/L
B2	1	15		mg/L
B2	2	70		mg/L
B2	3	50		mg/L
B2	4	1000		mg/L
B2	24	6	9	pH
B3	1	25		mg/L
B3	2	100		mg/L
B3	3	100		mg/L
B3	4	1500		mg/L
B3	24	6	9	pH
BCN	1	400		mg/L
BCN	2	700		mg/L

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
BCN	3	200		mg/L
BCN	24	7	9	pH
BCT	1	600		mg/L
BCT	2	1000		mg/L
BCT	3	400		mg/L
BCT	24	7	9	pH
H1A	1	5		mg/L
H1B	1	10		mg/L
H1A	2	25		mg/L
H1B	2	40		mg/L
H1A	5	1		mg/L
H1B	5	3		mg/L
H1A	24	7	8	pH
H1B	24	7	9	pH
H1A	28	0		mg/L
H1B	28	0		mg/L
H1A	6	20		mg/L
H1B	6	40		mg/L
H1A	11	0		mg/L
H1B	11	0		mg/L
H1A	23	6		mg/L
H1B	23	4		mg/L
H1A	20	0		mg/L
H1B	20	1		mg/L
H2A	24	7	8	pH
H2B	24	7	9	pH

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
H2A	23	8		mg/L
H2B	23	15		mg/L
H2A	5	1		mg/L
H2B	5	3		mg/L
H2A	28	0		mg/L
H2B	28	0		mg/L
H2A	6	20		mg/L
H2B	6	40		mg/L
H2A	7	100		mg/L
H2B	7	200		mg/L
H2A	8	200		mg/L
H2B	8	300		mg/L
H2A	11	0		mg/L
H2B	11	0		mg/L
H2A	20	0		mg/L
H2B	20	1		mg/L
H3A	24	7	8	pH
H3B	24	7	9	pH
H3A	2	25		mg/L
H3B	2	40		mg/L
H3A	7	100		mg/L
H3B	7	200		mg/L
H3A	8	100		mg/L
H3B	8	250		mg/L
H4A	24	7		pH
H4B	24	8		pH

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
H4A	7	100		mg/L
H4B	7	200		mg/L
H4A	8	100		mg/L
H4B	8	250		mg/L
H5A	24	7	8	pH
H5B	24	7	9	pH
H5A	23	6		mg/L
H5B	23	4		mg/L
H5A	5	1		mg/L
H5B	5	3		mg/L
H5A	1	8		mg/L
H5B	1	15		mg/L
H5A	28	0		mg/L
H5B	28	0		mg/L
H5A	7	20		mg/L
H5B	7	40		mg/L
H5A	11	0		mg/L
H5B	11	0		mg/L
H5A	20	0		mg/L
H5B	20	1		mg/L
H6A	24	7	8	pH
H6B	24	7	9	pH
H6A	23	6		mg/L
H6B	23	4		mg/L
H6A	2	25		mg/L
H6B	2	40		mg/L

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
H6A	1	5		mg/L
H6B	1	10		mg/L
H6A	5	1		mg/L
H6B	5	3		mg/L
H6A	28	0		mg/L
H6B	28	0		mg/L
H6A	6	20		mg/L
H6B	6	40		mg/L
H6A	7	100		mg/L
H6B	7	200		mg/L
H6A	8	100		mg/L
H6B	8	250		mg/L
H6A	11	0		mg/L
H6B	11	0		mg/L
H6A	20	0		mg/L
H6B	20	1		mg/L
HE1	2	50		mg/L
HE2	2	75		mg/L
HE3	2	100		mg/L
HE4	2	100		mg/L
HE5	2	150		mg/L
HE6	2	75		mg/L
HE1	24	7	9	pH
HE2	24	7	9	pH
HE3	24	5	9	pH
HE4	24	6	9	pH

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
HE5	24	5	10	pH
HE6	24	6	9	pH
HE1	11	0		mg/L
HE2	11	0		mg/L
HE3	11	3		mg/L
HE4	11	3		mg/L
HE5	11	3		mg/L
HE6	11	3		mg/L
HE1	4	100		mg/L
HE2	4	100		mg/L
HE3	4	200		mg/L
HE4	4	200		mg/L
HE5	4	500		mg/L
HE6	4	200		mg/L
HE1	5	2		mg/L
HE2	5	5		mg/L
HE3	5	30		mg/L
HE4	5	10		mg/L
HE5	5	30		mg/L
HE6	5	10		mg/L
HE1	12	10		mg/L
HE2	12	10		mg/L
HE3	12	20		mg/L
HE4	12	20		mg/L
HE5	12	20		mg/L
HE6	12	20		mg/L

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
HE1	30	2		mg/L
HE2	30	2		mg/L
HE3	30	2		mg/L
HE4	30	2		mg/L
HE6	30	2		mg/L
HE1	6	40		mg/L
HE2	6	50		mg/L
HE3	6	80		mg/L
HE4	6	80		mg/L
HE6	6	80		mg/L
HC1	2	1000		mg/L
HC2	2	1000		mg/L
HC3	2	1200		mg/L
HC4	2	1200		mg/L
HC5	2	1500		mg/L
HC6	2	1000		mg/L
HC1	11	5		mg/L
HC2	11	5		mg/L
HC3	11	10		mg/L
HC4	11	10		mg/L
HC5	11	10		mg/L
HC6	11	10		mg/L
HC1	20	20		mg/L
HC2	20	20		mg/L
HC3	20	50		mg/L
HC4	20	50		mg/L

(cont'd)

Table 14 (cont'd)

WQSTD	PARMCODE	WQSTDVAL	WQSTDVAL2	UNITS
HC5	20	80		mg/L
HC6	20	50		mg/L
HC1	24	7	10	pH
HC2	24	7	10	pH
HC3	24	7	10	pH
HC4	24	7	10	pH
HC5	24	7	10	pH
HC6	24	7	10	pH
HC1	8	400		mg/L
HC2	8	400		mg/L
HC3	8	400		mg/L
HC4	8	400		mg/L
HC5	8	400		mg/L
HC6	8	400		mg/L
HC1	5	100		mg/L
HC2	5	100		mg/L
HC3	5	150		mg/L
HC4	5	150		mg/L
HC5	5	150		mg/L
HC6	5	150		mg/L
HC1	12	10		mg/L
HC2	12	10		mg/L
HC3	12	20		mg/L
HC4	12	20		mg/L
HC5	12	20		mg/L
HC6	12	20		mg/L

2.7 Water Quality Subsystem

Two types of data bases are included in the Water Quality Subsystem: data for modeling of receiving water quality in rivers; and monitoring data at stations on river networks. The modeling data are contained in three Paradox® tables named MODLCOEF, MODLLOAD, and MODLRCH (defined in Tables 15, 16, and 17, respectively). The monitoring data are contained in two Paradox® tables named WQDATA and WQSITES (defined in Tables 18 and 19). These five tables are shown on the following pages.

Table 15

FIELDS IN TABLE **MODLCOEF**—COEFFICIENTS
TO COMPUTE DECAY RATE K1 FOR STREAM QUALITY MODELING

Field	Format	Units	Description
PARMCODE	N	-	Parameter code, as defined in Table PARMTABL.
Ka	N	/day	Coefficient Ka in the formula K1 [decay/day] = Ka +
Kb	N	/day	+ Kb* exp(...
Kc	N	/C	exp (Kc T)+
Kd	N	/d-C	+Kd*T, where T is temperature in degrees Celsius.

Table 16

FIELDS IN TABLE **MODLLOAD**—NONPOINT SOURCE DATA

Field	Format	Units	Description
BASIN	N	-	River basin code, as defined in Table RCHFILE.
SEG	N	-	River segment code, as defined in Table RCHFILE.
KM	N	km	River km from downstream end of river SEG.
CPMDCPDE	A6	-	User code for physical scenario (wet season, etc.).
PARMCODE	N	-	Water-quality parameter code, as in Table PARMTABL.
BKCONC	N	mg/L	Background concentration of the parameter.

(cont'd)

Table 16 (cont'd)

NPSLOAD	N	kg/d	Nonpoint source loading in lateral inflows.
PSLOAD	N	kg/d	Point source loading at the given river location.

Table 17

FIELDS IN TABLE **MODLRCH**—FLOW, VELOCITY, REAERATION K2 FOR STREAM QUALITY MODELING

Field	**Format**	**Units**	**Description**
BASIN	N	-	River basin code in Table RCHFILE.
SEG	N	-	River segment code in Table RCHFILE.
KMDS	N	km	Upstream end of reach to which the parameters and coefficients apply.
KMUS	N	km	Downstream end of the river reach (measured in km from mouth of river SEG).
CONDCODE	A6	-	User code defining the physical scenario (low flow, etc.).
Q	N	cmd	River stream flow.
VEL	N	m/s	Flow velocity.
K2	N	/day	Reaeration coefficient in the Streeter-Phelps equation for DO profile.

Table 18

FIELDS IN TABLE WQDATA—AMBIENT
WATER QUALITY MONITORING DATA

Field	Format	Units	Description
WQSITEID	A10	-	Identification code for the water quality monitoring site, as in WQSITES.
PARMCODE	N	-	Water quality parameter code defined in PARMTABL.
SAMPDATE	A6	yymmdd	Date on which sample was taken.
SAMPTIME	A4	hhmm	Time at which sample was taken.
LOCCODE	A1	-	Location code (left bank, surface, bottom, etc.).
QCCODE	A6	-	Quality control code.
VALUE	N	[1]*	Measured value of the parameter.
VAL-RMK	A1	-	Remarks code.

**Note*: [1] Units as defined in Table WQSTDESC.

Table 19

FIELDS IN TABLE WQSITES—DESCRIPTION
OF WATER QUALITY MONITORING STATIONS

Field	Format	Units	Description
WQSITEID	A10	-	Site identification code for a monitoring station.

(cont'd)

Table 19 (cont'd)

DATASRC	A6	-	Identification code for source of data.
BASIN	N	-	River basin code defined in the table RCHFILE.
SEG	N	-	River basin code defined in the table RCHFILE.
KM	N	km	River km from mouth of river segment.
PROVINCE	A10	-	Province name.
REGION	A10	-	Region name.
COUNTRY	A1	-	Country code.
STATYPE	A6	-	Code for station type.
AGENCY	A10	-	Operating agency.
STARTDATE	A6	yymmdd	Beginning date of monitoring record.
ENDDATE	A6	yymmdd	End date of monitoring record.
WQSTADESC	A80	-	Description of the water quality monitoring station.

2.8 DEMDESS Software and Output

In using the Paradox® software package, the user develops software applications by using scripts. User-developed software can perform many types of functions: customizing the user interface for input to a data base; querying and selecting records from a data base to view them on the screen; producing output reports; and producing graphs. Computations and stream quality modeling can also be carried out by scripts. Scripts are easy to develop and can be made in a combination of two ways:

- by recording keystrokes to capture the choices made from Paradox® menus; and
- by writing programs in Paradox®'s PAL (Paradox® Application Language), which is a simple structured language similar to Pascal.

Chapter 3 provides suggestions on how to learn to use Paradox® and the Initial DEMDESS. The software developed for the Initial System is listed in Appendix F, and is provided as a learning tool. It should be noted that the listings in Appendix F may appear to be long, but that software development in Paradox® is a very easy process. The actual software development time for the Initial System was about five working days, of which about three working days were associated with learning to use Paradox® to produce the output plots (Figures 1 through 11 in Chapter 1).

Chapter 3

USE OF THE INITIAL DEMDESS

3.1 Introduction

DEMDESS is built using the "prototyping" approach in which design, implementation, and analysis is an iterative process. Use of DEMDESS should be approached in a step-wise fashion, starting with simple queries and reports and working up to complex analytical "packages." This chapter is not meant to be a comprehensive guide for all types of DEMDESS applications. Rather, it presents some rules to follow and guidelines for mastering the system.

DEMDESS uses Paradox® as its data base manager. This chapter assumes that the reader has access to the Paradox® manuals while working through DEMDESS. Paradox® has many components and several manuals; it can be somewhat daunting at first trying to decide where to start. The best place to start is to read through the *Introduction to Paradox®*. After reading that manual, the *Paradox® Users Guide* will be the primary reference book in working with DEMDESS. For working with DEMDESS, it is important to read through the "Data Base Design" chapter, and have the data base dictionary handy.

Note: Before starting any work with DEMESS be sure to back up the entire system. Ideally, it should be backed up to floppy disks, so that even if a "disaster" were to occur on the hard disk, DEMDESS would not be lost. Backing up the system provides another important operational advantage as well: it allows you to feel free to experiment with queries and edits without having to worry that you will "break" something. DEMDESS is not a "canned" system in which the user is protected from mistakes; it is an open system that allows the user great freedom to test many different options. Backing up the system, therefore, is critical. If you do not know how to back up using DOS (or other backup utilities), find someone to teach you how.

The rest of this chapter will step through suggested techniques, starting with simple queries and going through complex analyses using the full power of the DEMDESS design.

3.2 Simple Queries, Reports, and Graphs

Start up Paradox® in the DEMDESS directory. The first Paradox® command is VIEW. Browse through the DEMDESS tables using this command. Try out various VIEW keys, such as [HOME], [END], and [F7]. [F7] alternates from table view to form view; this is your first introduction to Paradox® forms. Forms are a powerful way to present data. Paradox® will automatically create a default form when you hit [F7].

After becoming familiar with the basic contents of DEMDESS, try the next Paradox® command, ASK. ASK is a very powerful "query by example" system. This command can be used to produce summaries of individual tables as well as perform complex multitable connections with computations. Learning the ASK system will provide you with many of the basic capabilities of DEMDESS.

Start with a simple query of one table, such as DISCH1. Some ideas to try:

1. Produce a simple list of all DISCHIDs and INDCODES by checkmarking with [F6] the DISCHID and INDCODE fields. Hit [F2] to produce the ANSWER table.

2. Produce a count of the number of dischargers by INDCODE by removing the checkmark from DISCHID and replacing it with the command CALC COUNT.

3. Produce a summary of the total flow by INDCODE by removing the CALC COUNT from DISCHID and putting CALC SUM in the QTOT field.

4. Produce a list of all municipal dischargers by checking the DISCHID field (using [F6]) and entering 22 in the INDCODE field (the INDCODE value for municipal is 22). Be sure to remove the CALC SUM from the QTOT field.

Many summaries can be produced for any given table in this way.

3.3 Basic Analyses: Merging Tables

Most summaries and reports of existing conditions can be performed with the ASK command using table connections, or "joins." The table join capability is extensive; in some cases in DEMDESS, up to six tables are joined in one operation, with many computations and "what if" conditions.

Two tables are joined by doing two ASK commands in a row so that the two tables to join are in the working space at the same time. The joins are performed by putting the same "example element" value in the common field(s) of the two tables. Example elements are entered by first hitting [F5] and then typing the example element name.

Again, start with a simple example, such as the following one:

Join DISCH1 to INDCODES using the INDCODE variable as the link. Start with example 1 above, in which DISCH1 is placed in the work space. Add the INDCODES table in the work space by hitting [F10], select ASK, and enter INDCODES as the table name. In the

INDCODE field in both tables, enter IND as the example element. [F3] and [F4] willmove you back and forth between the two tables. Check the INDCODE field in DISCH1 and check the INDCODEDESC field in the INDCODES table. Hit [F2]. If it is set up correctly, you should get a list that includes the text name of the industry group along with the DISCHID.

Example elements are very powerful for joining tables and also for performing calculations. The sample scripts TAXCOMP.SC and FINECOMP.SC provide examples of joining tables and performing calculations. These fairly simple queries perform operations such as determining which dischargers exceed emission standards and computing total kilograms of pollutant that exceed the standard. Much of the power of DEMDESS is in table relationships; becoming comfortable with the query system will provide you with quick answers to even complex questions.

3.4 Reports and Graphs

The REPORTS and IMAGE-GRAPHS operations provide the capabilities to generate ad hoc output quickly as well as design and produce finished products. As with the other operations, the best way to proceed is to start out simple and work up to finished products. Paradox® provides capabilities for saving report and graph specifications; these specifications can be included as a component of a completed analysis package. The TRTPLCY, PLOTS, and PLOTOUT scripts in Appendix F provide several examples of using predefined graphics specifications.

3.5 Defining Scenarios

Scenarios are built using a variety of techniques, depending upon the specifications of the scenario. Changes in regulations and standards, and so on are made by simply editing or adding the values in the appropriate tables. For instance, to evaluate changes in revenue that would occur with a new fine schedule, you can either edit the existing FINES table values or enter a new set of FINEID values in the FINES table. The latter method will readily permit comparisons between the two fine schedules. If a new FINEID schedule is entered, then evaluation may require global changes to the FINEID values in the DISCH1 table. The ASK command CHANGETO is a powerful global change facility.

Many scenarios, and combinations of scenarios, can be developed by adding or editing the DEMDESS tables and then building the queries to summarize the results. More complex scenarios, such as the TRTPLCY example, require a multistep process. To build these more complex scenarios, work in relatively small steps, verifying and refining as you go.

As the operations get more complex, you will need to get into the Paradox® Application Language (PAL). PAL is a powerful programming language that combines all of the power of the Paradox® ASK features with full programming logic. Start with simple PAL programs (perhaps use the TRTPLCY programs as examples) and gradually work your way up.

It is beyond the scope of this manual to provide a full tutorial on Paradox® and PAL. Further, because DEMDESS is still very new, a full suite of applications has as yet not been developed. As all of the users develop their applications, it is expected that a standardized, user-friendly package will become available.

Chapter 4

ENHANCING DEMDESS

4.1 Adding Data to DEMDESS

As with the original loading of DEMDESS (see Appendix E), data can be added to the system in three ways:

1. Manual entry directly into DEMDESS tables;
2. Building a data entry system and then converting and loading into DEMDESS; and
3. Converting from other, existing data bases and files.

The first method is obviously the simplest and fastest when there is a limited amount of data to enter and doing so will not be a regular operation. Examples of such data include water quality standards, fines, and tax schedules.

The second method is most useful when a large amount of data, usually on standard paper forms, needs to be entered. Experience has shown that the best way to enter this kind of data is to create on-screen forms (using the Paradox® FORMS generator) that mimic the look of the paper forms. It is common that the data base and on-screen forms for data entry will need to be somewhat different than the DEMDESS data structures, so that a conversion from the data entry system to DEMDESS will also have to be written. Remember, the forms were not developed for DEMDESS, but for some other purpose, so it will only be coincidental if the two match perfectly. Two examples of this method are Bulgaria's DENTR system and Romania's data entry system.

The third method is required in all countries and is critical to DEMDESS. The existing or newly emerging administrative procedures will be the best and richest sources of data for DEMDESS. Programs have been written for converting the major existing data systems in the four countries. Maintenance and use of these conversion programs are essential for the continuing viability of DEMDESS.

4.2 Building Stand-Alone Applications

The Paradox® Personal Programmer is a powerful and relatively simple way to convert analyses built in Paradox® into stand-alone applications. The Paradox® Runtime Package can be used to distribute these applications. It is anticipated that such stand-alone applications will become available in the future as data are verified and users standardize operations.

4.3 Guidelines for Enhancing DEMDESS

As noted in Chapter 3, DEMDESS is designed as an open system that can be expanded to include additional components. There are two primary ways to enhance the existing DEMDESS system:

1. Add new data elements to the existing tables. For instance, the DISCHX table contains many of the anticipated data elements for expected DEMDESS uses, but it is likely that additional data elements will be added in the future as analytical and reporting needs become more clearly defined.

2. Adding new tables that link to existing DEMDESS tables.

The most important consideration in adding new data elements or tables to DEMDESS is retaining the system's essential data base design. This means that special care would need to be exercised to maintain existing data table links and analytical capabilities. For example, changing the way dischargers link to emissions could have serious consequences for DEMDESS's capabilities. The ability to model changes in treatment levels could be damaged, for instance. Additionally, the impact changes could have may not be immediately obvious.

Trade-offs are involved in adding new data elements to existing tables versus adding new tables with associated data. Some considerations include overall system performance, simplicity, compatibility with existing applications, and retention of the essential DEMDESS design. There are no hard and fast rules; this is where data base management experience and judgment come into play.

4.4 Expanding the Scope of DEMDESS

Some specific areas have been identified as candidates for expanding DEMDESS's scope. These areas were, as much as possible, considered in the initial DEMDESS design so that the system's scope could be expanded with minimal impact to its other elements.

4.4.1 Evaluating National-Level Economic Impacts

A new table, preliminarily named "NATECON" (for NATional ECONomics), could be added to examine national-level economic impact analyses. This table would contain data on output, employment, energy, and profit by economic sector. As part of DEMDESS's development, the WASH team has performed preliminary work on the design of NATECON. It is hoped that follow-up efforts can fully implement this new table.

NATECON could also include capabilities to evaluate trade balance factors. This would require reliable data and projections for imports and exports by industrial sector, as well as reliable estimates of currency conversion rates.

4.4.2 Enhancing Abilities to Evaluate Toxins

Reliable data on many toxins and micropollutants are not now widely available in many Danube basins, but DEMDESS is fully capable of working with these data as they become available. It may be many years before significant and reliable toxins data are available, however. In their absence, some techniques are available for evaluating toxins on an approximate basis. The U.S. has developed data bases on toxins using various treatment levels; these data bases could be incorporated into DEMDESS's treatment data base to provide approximate calculations on toxins emissions, given different treatment scenarios. Several potential data bases and techniques were identified during work on the initial DEMDESS, and these may be feasible for addition in follow-up activities.

4.4.3 Evaluating Health Effects Impacts

Human health effects analysis will require data on ambient water quality changes and on the potential impact of changes in ambient water quality on morbidity and mortality. The first requirement will necessitate enhancing the system's water quality modeling components as well as expanding data and modeling for toxins. The second requirement will necessitate building dose-response data and models as well as building population exposure capabilities.

4.4.4 Enhancing Institutional Data Analysis

Institutional relationships were built into the initial DEMDESS. Additional tables describing the institutions themselves would greatly expand the system's analytical capabilities. Institutional data such as current and projected staffing, staff training requirements, and institutional responsibility could be added. These data would permit valuable results, such as training needs analyses, projections of staff shortages, evaluations of the impact of reorganization, and so on. Full reporting on who is responsible for what could prove invaluable in building and maintaining effective institutional structures.

4.4.5 Enhancing Collector Costs Analysis

Collector costs are a major factor in overall municipal treatment analysis. Currently the system can manage collector data to a degree, but its capability should be enhanced to include cost impact analysis. The DEMDESS design can be readily expanded to include collector cost tables as data become available.

4.4.6 Evaluating Laboratory Demands

Laboratory demand projections could be an important element in determining costs and the ability to meet requlatory requirements. Sampling and analysis costs are not straightforward to predict. The evaluation of laboratory demands depends on factors such as the number of samples, the factors being analyzed, quality control requirements, and detection limits. Data relating parameters to analysis trains and detection limits willbe needed, along with policy data on numbers of samples required per year per location. Analysis costs, staffing, and equipment information will also need to be incorporated into DEMDESS.

The effort needed to incorporate laboratory demands would be justified by the ability such an effort would provide to maximize available laboratory analysis capacity. Laboratory demand analysis would also offer a rationale for funding improvements in national and international laboratory capabilities.

4.4.7 Enhancing Water Supply Tables

Water supply components have been designed into DEMDESS, but not fully, due to insufficient time and data. Water supply components are an important aspect of treatment policy and health effects analysis. Expanding and exercising them in DEMDESS would enhance the initial model significantly.

Appendix A

DEMDESS CONCEPTUAL FRAMEWORK

This appendix reviews the conceptual framework of DEMDESS as described in Chapter 3 of the DEMDESP WASH Team Assignment Report ("Point Source Pollution in the Danube Basin," Volume I). This chapter summarizes the structure and status of DEMDESS, its potential uses as a decision support tool, and its status regarding implementation and the actions necessary to make DEMDESS fully operational.

Chapter 3

THE DECISION SUPPORT SYSTEM (DEMDESS)

This chapter provides an overview of the structure and status of DEMDESS and its potential uses as a decision support tool. The following topics are discussed: a general water quality management framework for large river basins; the design of DEMDESS and the information that can be stored and analyzed using the system; the principal clients and users of DEMDESS; and the status of DEMDESS in each of the four study countries. The *User Manual* provides more detailed information on the structure of DEMDESS; Chapter 5 provides more detailed information on the current status and future development of DEMDESS.

3.1 The Need for Emissions Management

The need for emissions management can be best described by defining the relationship of emissions management to the overall water quality management framework. Figure 3.1 presents in graphic form the three subsystems within a water quality management framework, which is applicable to the Danube River basin or any other large river basin. The three subsystems, in summary, are as follows:

- **The Emissions Subsystem:** This subsystem includes all point and nonpoint sources of emissions, such as municipal, agricultural, and industrial wastewater, treated effluent, stormwater, and groundwater.
- **The Environment and Impact Subsystem:** This subsystem includes the environment and other entities that are affected by the emissions. Water quality and quantity and water uses are the major concerns in this subsystem.
- **The Institutional Subsystem:** This subsystem includes all the organizations and individuals that together define the rules and physical interventions by which the emissions subsystem is controlled.

The three subsystems are highly interrelated, and the interrelationships occur at a large number of physical points and other interfaces. For example, thousands of diverse upstream pollutant sources impact the water quality at a given river section, and the regulations and organizations that control either the sources or the water use are many. Formulating regulatory policies, planning technical interventions, and organizing related institutions in such a large and complex system require a decision support system.

A *decision support system* is an information system that is designed to assist decision makers. It does so by presenting to them the best available and most relevant information on which to

This Chapter is from "Point Source Pollution in the Danube Basin," Volume I.

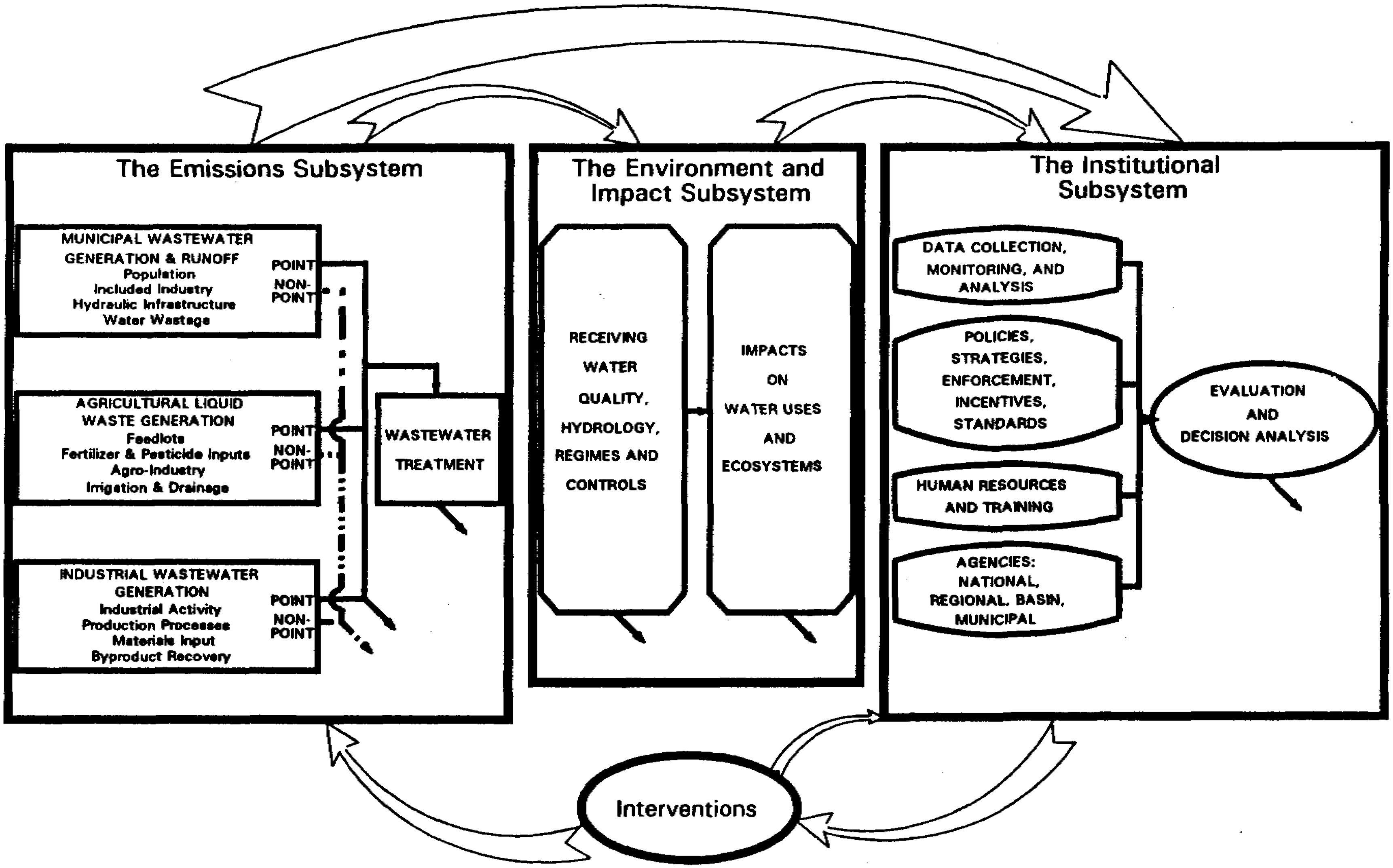


FIGURE 3.1 - WATER QUALITY MANAGEMENT FRAMEWORK

base their decisions. It does not make decisions, and it will inevitably omit certain judgmental aspects that cannot easily be made explicit or reduced to formal analysis. However, it combines large amounts of basic data, technical procedures for analyzing those data, the processing power of modern computers, and efficient techniques for presenting information in easily understandable form.

A decision support system enhances the effectiveness of decision makers in three ways. First, it relieves them of time-consuming and technically demanding data reduction tasks, thus allowing them to focus directly on the key aspects of the decisions they are called on to make. Second, it employs sophisticated analytic methods to explore the implications of information, implications that might otherwise be missed. Third, it organizes and presents findings in forms that are easily and readily understood.

3.2 The Design of DEMDESS

The Danube Emissions Management Decision Support System (DEMDESS) is designed to assist decision makers who are concerned at some level with affecting the discharge of pollutants into the Danube River or its tributaries. It is intended to be a part of a larger decision support system that covers the entire field of water quality management in the Danube River basin (perhaps eventually all water management, or even all environmental management). Although it is closely related in concept to these broader concerns, its immediate focus is the direct or indirect discharge of pollutants, a topic that has received less attention in some countries than ambient water quality conditions, health effects, and other major aspects of water quality management.

Policies designed to achieve an appropriate level of water pollution control for the Danube River and the Black Sea must focus primarily on limiting discharges or emissions of pollutants into the Danube and its tributaries. Nonconservative pollutants can be assimilated and conservative and nonconservative pollutants can be diluted, but assimilation and dilution cannot be the cornerstones of pollution control policy because they have only a limited capability to reduce pollution damage. Reduction of emissions must be the main instrument of pollution control policy, and emissions themselves are the strategic variables in DEMDESS.

Although its focus is emissions, DEMDESS must address all the subsystems of the water quality management framework (Figure 3.1) to be an effective decision support system. The framework reflects the ways in which emissions are generated, the effects they produce, and the institutional control subsystem through which water quality can be managed. Full implementation of a decision support system encompassing all of the subsystems of this conceptual framework will present to decision makers the full implications of the pollution control policy and investment options among which they may choose. Interventions such as constructing wastewater treatment plants are initiated according to a certain regulatory policy. Such policies and their application must consider pollution effects and costs. For example, regulations to require a uniform standard of biological treatment for all municipal wastewaters in a country would have immense cost implications and would not necessarily achieve desired water quality in all waters at minimum cost. Alternatives to such a policy must be evaluated, and DEMDESS is a useful tool for doing this.

A decision support system can help in the formulation of cost-effective policies by estimating water quality and the economic and financial impacts of various options. Another significant policy issue is phasing of interventions. Interventions that provide the highest pollution reduction for investment should be identified by a decision support system and given high priority. The interventions need not always be structural or "end-of-pipe," however. Nonstructural interventions are often the most effective solution. For example, effluent standards that address key pollutants, backed up by monitoring and enforcement, will affect in-factory procedures for waste management, as well as treatment decisions, and may also induce changes in processes. Taxing (or removing subsidies on) certain raw materials, such as metals, would encourage their recovery in industrial processes, and they would then be less likely to end up in wastewater emissions.

DEMDESS is designed to be "data driven." It can store and analyze a large data base to represent existing conditions. Using the existing data base as the departure point, DEMDESS can simulate and analyze various regulatory and technical emissions control scenarios in terms of water quality and economic and financial impacts under today's conditions and expected future conditions.

DEMDESS was developed using the commercially available relational data base application development program, Paradox®, which was chosen because of its compatibility with other data base software (e.g., dBASE® and SQL®), its analytic power, its economy in use of computer memory, and its reporting capabilities. The program consists of data tables, queries (which express the questions asked of the data), and scripts (which define queries and computations on the data with a simple programming language).

The design of DEMDESS is described in block diagram form in Figure 3.2. The *User Manual* should be referred to for a more detailed and technical overview. DEMDESS consists of six subsystems: Reach, Emissions, Treatment, Water Quality, Regulatory, and Institutional. The six subsystems are more detailed and specific breakdowns of the subsystems of the framework presented in Figure 3.1, except for the Reach File, which defines the river network.

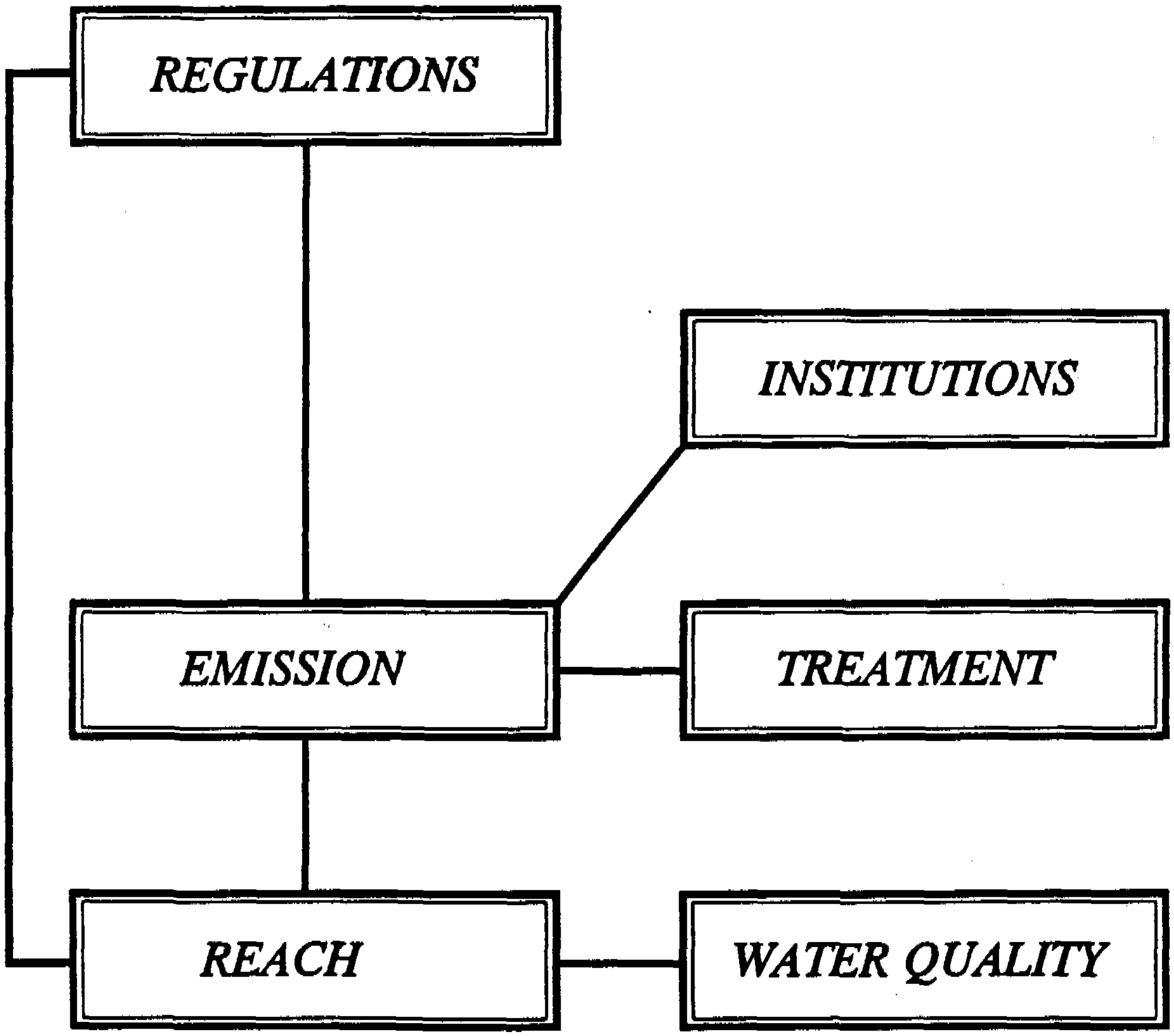


FIGURE 3.2 - ORGANIZATION OF DEMDESS

The Reach Subsystem is central to the design and operation of DEMDESS. It alphanumerically defines every river segment in terms of its basin (Danube) and sub-basins (tributaries to the Danube and their tributaries), and it also defines stations on the river segments, in kilometers. The Reach Subsystem is the "conduit" that connects various subsystems, for example, by bringing together for comparison the cumulative emissions load at a given river reach and station, the resultant water quality, the relevant in-stream water quality regulation, and the regulatory authority. The Reach Subsystem can also be used to connect DEMDESS to other data systems, such as in-stream hydrologic and water quality models and data bases.

The Emissions Subsystem contains data on emissions, such as type, pollution parameter, quantity, and sampling date, and emission sources, such as industrial sector, location, and size (population, sales). Scenarios can be developed by selecting or adding to existing emissions. The Water Quality Subsystem contains in-stream monitoring data, monitoring sites, and water quality modeling coefficients. The Treatment Subsystem contains data on treatment efficiency and coefficients for cost and economic and financial functions. The Regulatory Subsystem contains water quality and discharge standards and data on related fines and taxes. The Institutional Subsystem contains data on control authorities, such as riverbasin authorities, and is connected to the Emissions Subsystem. Institutional data are also included in other subsystems.

DEMDESS provides the results of a regulatory and technical intervention scenario in terms of modified emissions quality, in-stream quality, and financial costs. Costs can be given individually for each pollution source or all sources, or they can be aggregated by sector. Per capita costs and sectoral costs can be provided to evaluate the regional or national economic impacts of scenarios. These analytic features are built with the query and scripting (programming) features of Paradox®.

As noted, DEMDESS is a data-driven system, and the effort that goes into programming analytic procedures is only a fraction of the effort needed for data collection and development. The analytic questions that are answered by DEMDESS currently are limited in number and constitute a demonstration of the kinds of questions that can be handled. The flexibility, adaptability, and simplicity of the query and scripting features are important to the enhancement and future utilization of the system. Also, DEMDESS currently focuses on limited areas of the water quality management framework, but with the above-described Reach Subsystem and scripting capabilities, it has the inherent capability to link easily to many other areas of water quality management.

3.3 The Decision Makers and the Users

The term *clients* is used to denote those decision makers who are likely to benefit from the information provided by DEMDESS. Potential clients exist at many levels in the water quality control hierarchy. Among them are policymakers, planners, and regulators. Each of these potential client types has specific information needs, which are often quite different from those of other potential clients. (A fourth client type, facility and system managers, is likely to require information that is more site specific and operationally oriented than is that provided by DEMDESS.) Potential clients and their information requirements may be characterized as follows:

- **Policymakers** in any of the four participating countries within which DEMDESS has been implemented: These policymakers include legislators and the top officials of the pollution control ministries in the four countries. The effectiveness of such policymakers will be greatly enhanced if they can request information on and be shown the effects of various alternative policies that are available to them. Also included in this category are international organizations, such as the Danube Commission and EPDRB.

  Policymakers, whether in the ministries or the legislature, will be assisted by a decision support system that identifies and compares the widest possible range of practically available policy options. One of the significant shortcomings of most policymaking institutions is that they have too narrow a focus or definition of options. DEMDESS can assist policymakers in considering a broad range of policy options. Examples of such policy options include standard setting and enforcement by fines, effluent charges, subsidies, and compliance scheduling.

  Policymakers are primarily interested in data with a high degree of aggregation, particularly for formulating policies at the national level. DEMDESS can provide information with a high degree of aggregation.

- **Planners** in any of the four participating countries: In this context, planners are those individuals whose role is to achieve established policy goals to the maximum extent possible, given available resources. Quite often, planning in the water quality management sector is organized on a river basin level, reflecting the substantial interdependencies within a basin and the lesser water quality interdependencies between a basin and the areas outside it. Thus, planners will most often be found within river basin or regional organizations in the four countries, although they may also be found at the subregional (subbasin) level. International donor and lending agencies are also included in this category of potential DEMDESS clients because they, too, are interested in the evaluation of specific potential development projects.

  Planners examine investment and other strategic options below the policy level. They implement policies in a national, regional, local, or basin context. Planners require greater disaggregation and greater detail than do policymakers, but less detail than is required by regulators. Unlike policymakers, they seek information about site-specific dischargers and facilities. Unlike regulators, they need not see detailed information about the operation of existing dischargers and facilities.

- **Regulators** in any of the four participating countries: The role of a regulator is to attain compliance with established rules, whether ambient, performance (discharge), or design standards, or other rules. Regulators will most often be found at the local level, usually as agents of regional or national regulatory agencies. Regulation necessarily occurs on a site-specific, thus local, basis.

In order to serve a diverse mix of decision makers who call on DEMDESS intermittently as they need information, an entity must exist to maintain DEMDESS on a continuous basis. The term *user* denotes this entity. At least one system user must exist in each country. The responsibilities of the people in such a group include assembling and assessing data, maintaining the data base, interacting with decision makers, formulating new queries, and expanding and improving the system. In each country, there should be a primary user, which would also be the focal point for both national and international coordination of the system. This would be an office within the ministry of the environment or its equivalent. While it may perform some of the functions through a subcontract with an institute or other technical entity, the ministry must maintain positive control of DEMDESS. This primary user may have multiple clients at the national and lower levels.

As agreed at the Institutionalization Workshop (see Appendix A), the functions of the primary user should include the following:

- responding to policy changes by the parent ministry
- establishing topics and procedures for regular reporting by other users in the country
- providing public information
- responding to requests for special reports
- providing technical assistance and training for other users in the country
- establishing default data
- conducting ongoing system development
- protecting legal ownership and sensitivity of information

Other users (an office within, or serving, a regional or basin authority, for example) are likely to be closely associated with a single client, but they may provide services for other clients on request.

3.4 General and Country-Specific Status of DEMDESS

The structure of DEMDESS is now initially specified and incorporated into a framework of Paradox* data tables, queries, and scripts. However, before DEMDESS can be used in a specific application (whether at the river basin, country, or international level), the application-specific data that are to be analyzed must be added to the basic structure. In other words, the basic DEMDESS structure is similar to a set of mathematical equations that contain variable names and relationships but no values for those variables. The user must add all of the empirical information needed to apply DEMDESS to a specific situation or purpose.

The DEMDESS structure will no doubt be expanded and linked to other decision support systems in the future, as DEMDESS is called on by decision makers at all levels to answer additional questions. It will also be applied to an increasing number of geographic areas, in its current form and as it is expanded. At present, implementation of DEMDESS has been initiated in one demonstration river basin in each of four Danubian countries (Bulgaria, the CSFR, Hungary, and Romania). The degree of implementation in each of the basins differs, largely due to differences in data availability among the four sites. In each case, the river basin chosen for initial demonstration purposes was one that had a number of water quality problems, so that the the application would be valuable in itself. However, the river basins were chosen as well because better data were available for them than for some other basins in each country. The demonstration applications can thus be useful not only for setting emissions control priorities within the subject basins but also for setting data collection priorities elsewhere in each country and for exploring policy options more completely than would be possible on a countrywide basis.

3.4.1 Bulgaria

The Jantra River basin was chosen to be the initial demonstration basin in Bulgaria. It was also the initial application within which the DEMDESS structure evolved, as WASH team members and their Bulgarian colleagues created the concrete implementation of the DEMDESS conceptual framework developed earlier in the project. Work began in November 1991, and the results of an intensive, special data collection effort in the basin were incorporated. Thus, the data used in this application are unusually complete. Consequently, the Jantra application is the most developed and the most completely tested of the four demonstrations.

The pollution parameters currently included in the Jantra application are BOD and total suspended solids (TSS). Current ambient and emissions standards are included. A limited policy analysis comparing alternative levels of taxes (effluent charges) and fines (for exceeding standards) was conducted at the request of the Bulgarian Ministry of the Environment, which must soon recommend new levels to the legislature.

Bulgaria has made a major commitment to use DEMDESS as a primary tool for water quality management. Existing water quality data systems are to be converted to DEMDESS, which will now be the official water quality management data base for the country. All Bulgarian streams and basin boundaries were digitized as part of the DEMDESP in order to facilitate this changeover.

3.4.2 CSFR (Slovak Republic)

The Slovak Commission on Environment chose the Nitra River basin as its demonstration basin. Work to implement this decision began in February 1992. The Slovak water quality data base is a high-quality one, and software routines to convert that information into Paradox* files have been written and the conversion accomplished. Work on the incorporation of the Slovak institutional parameters (fines and standards) is now in process. The system has been demonstrated for about 30 technical experts from Slovak water agencies. Although it is not yet ready to support client requests, the Slovakian application is well advanced.

3.4.3 Hungary

The Altalar River basin was chosen as the demonstration basin in Hungary. This choice was made in April 1992. Consequently, DEMDESS implementation in this basin has just begun. Fines and water quality standards information have been entered into the data base, but other pertinent data have yet to be entered. Obviously, use of the DEMDESS system by Hungarian clients lies in the future.

3.4.4 Romania

The Arges River basin was chosen as the demonstration basin in Romania. Work began in February 1992 on this application. Water quality data collection in Romania was spotty until 1992. However, a new system for data collection and maintenance was designed and implemented in 1992, and future prospects are encouraging. Software for converting the Romanian water quality data, as they accumulate, into Paradox* format has been written. Although only limited sample data have yet been entered in this application, the system will be ready to receive and utilize data as the new data collection system makes them available. No client analyses have been performed in Romania, and none will be performed soon, due to the lack of reliable water quality and emissions data.

Appendix B

A QUICK TOUR OF PARADOX

What is Paradox?

To begin with, what is a paradox? Something that appears to be a contradiction, but which is, in fact, true. For instance, powerful database programs have historically been difficult and slow to learn and use. So it is a paradox to say that sophisticated database programs are easy and quick to learn and use.

Hence, Paradox. On the one hand, it is an information management program of powerful capabilities and complex functions. On the other hand, it is easy to learn, swift to act, and simple to use.

- Paradox is a full-featured relational database management program that can be used either as a standalone system on a single computer or as a multiuser system on a network.
- Paradox Multi-Pack is a cost-effective way to increase the number of authorized users on a network. To use the Multi-Pack, you must have one or more copies of Paradox installed on your network. See the Paradox *Network Administrator's Guide* for details.

This manual applies both to Paradox and to Paradox Multi-Pack™. There are two other software products that extend the capabilities of Paradox: Paradox SQL Link and the Paradox Engine.

Paradox SQL Link

Paradox SQL Link works with Paradox 3.5 to provide access to data stored on SQL database servers.

Structured Query Language (abbreviated SQL and commonly pronounced *sequel*) is the standard language for querying and manipulating data in relational databases. Using the familiar Paradox interface, you can access data on one or more SQL database servers.

With Paradox SQL Link, you have the power of SQL at your fingertips, without needing to learn SQL programming. To access SQL data, you must install Paradox SQL Link with Paradox—you can use it only from within Paradox 3.5.

The Paradox Engine

Another member of the Paradox family is the Paradox Engine, a comprehensive library of C functions and Pascal procedures and functions that can be called from programs written in C or Pascal. These functions let you manipulate Paradox tables in both single-user and multiuser environments.

Extracted from the core of Paradox's data handling and multiuser concurrency technology, the Paradox Engine provides the programmer with an Applications Programming Interface (API).

What Paradox includes

Paradox is designed for computer users with all levels of experience—from beginners to the most sophisticated. No programming is required to use Paradox—you tell the program what you want it to do by making choices from menus.

You can also extend the power of Paradox by using the Paradox Personal Programmer,™ a powerful application generator that enables you to develop custom single-user database applications without programming, and by using PAL™ (the Paradox Application Language™), Paradox's structured programming language. Both of these tools are included in your Paradox package.

This book was written for those without extensive experience with computers or database programs. It includes an overview of Paradox and an extensive tutorial with step-by-step examples. These tutorials use sample tables installed automatically during Paradox installation.

This chapter gives you an overview of Paradox. In it, you'll find

- a quick tour of Paradox
- a description of its capabilities and system requirements
- a guide to the Paradox manuals

Welcome to Paradox! We think you'll find it both extremely useful and exceptionally usable.

A quick tour

Paradox is a relational database management program for your personal computer—whether it is a standalone system or part of a network. If you sometimes feel overwhelmed by the vast quantity or variety of information you work with, this program is for you. It allows you to control the expanding volumes of information you work with daily.

Before telling you what Paradox can do, however, let us show you by taking you on a quick tour of the program.

Tables

All information in Paradox is arranged in *tables*. In tables, categories of information are arranged in vertical *columns*, while individual records of people or things are arranged in horizontal *rows*. This makes it easy to examine or change the information. You can have as many Paradox tables as you want.

In Figure 1-1, a sample table of customers (included on your Installation/Sample Tables Disk) shows how information is arranged.

Figure 1-1. The sample *Customer* table

Categories of information in columns

CUSTOMER	Cust ID	Last Name	Init	Street	City	State	Zip	Country	Credit
1	1386	Aberdeen	F	45 Utah Street	Washington	DC	20032		50,000.00
2	1388	Svenvald	I	Gouvernment House	Reykjavik			Iceland	1,250,000.00
3	1784	McDougal	L	4950 Pullman Ave NE	Seattle	WA	98105		150,000.00
4	2177	Bonnefemme	S	120 University Drive	Stanford	CA	94323		75,000.00
5	2579	Chavez	L	Cypress Drive	Palm Springs	FL	32938		250,000.00
6	2779	Fahd	S	The Palace	Riyadh			Saudi Arabia	5,000,000.00
7	3128	Elspeth, III	R	1 Hanover Square	London			England	1,000,000.00
8	3266	Hanover	A	15 State Street	Dallas	TX	75043		750,000.00
9	3271	Massey	C	29 Aragona Drive	Oxon Hill	MD	29902		1,000,000.00
10	3771	Montaigne	L	30 Tauton Drive	Bellevue	WA	98004		450,000.00
11	4277	Matthews	A	P. O. Box 20336	Albuquerque	NM	87234		50,000.00
12	4335	Farouk	K	Hotel Cairo	Cairo			Egypt	250,000.00
13	4400	Samuelson	F	Bull Run Ranch	Aurora	CO	80022		600,000.00
14	4485	Fischer	R	14 Willow Lane	Birmingham	MI	48011		1,000,000.00
15	4589	*Leonardo*	D	198 *Via Canales*	***Rome***			*Italy*	1,500,000.00
16	4700	Harris	J	Old Country Road	Atherton	CA	94322		750,000.00
17	4884	Anders	B	Jaktstigen 42	Lidingo			Sweden	300,000.00
18	5341	Chevalier	R	392 Boulevard Raspil	Montpelier			France	1,750,000.00
19	5728	Helms	D	52 Brattle Street	Cambridge	MA	02138		85,000.00
20	5855	Chin	F	Hotel Orient	Jurong			Singapore	150,000.00
21	6125	Hawes-Anderson	D	Waves Cottage	Palm Springs	FL	32302		600,000.00
22	6666	Matthews	J	1050 12th Street	San Francisco	CA	94232		1,250,000.00
23	6954	Mayor	K	40 Winding Way	Salt Lake City	UT	84100		100,000.00
24	7000	Simms	R	Box 13, RFD 2	Topeka	KS	66104		1,500,000.00
25	7550	Yee	L	2930 42nd Street	New York	NY	10032		1,500,000.00
26	7640	Raymond	S	390 Centre Street	Winnipeg	MB		Canada	500,000.00
27	7700	Connors	S	27 Portfolio Drive	Belair	CA	90026		900,000.00
28	8585	Sampson	L	29 Buena Vista Drive	Tiburon	CA	97992		150,000.00
29	8776	Weidner	R	56 Santa Ysbel	San Francisco	CA	94332		120,000.00
30	8996	Smith	J	Hotel Americain	Paris			France	1,000,000.00
31	9004	Ranier	T	8947 San Andreas	Klamath Falls	OR	97603		125,000.00
32	9226	Simpson	H	3 Pooks Hill	Dallas	TX	75211	Texas	75,000.00
33	9650	O'Hare	C	1 Airport Drive	Chicago	IL	60542	Chicago	2,500,000.00

Records for individual customers in rows

Rows and columns make it easy to examine your data

Forms

You might sometimes find it convenient to work with information about one person or thing at a time, rather than with a table full of information. With a single keystroke, you can display data from a table on a *form*, like the customer information form shown below. The information is the same as in the table, but arranged differently. If you change the information on the form, the information in the table changes as well.

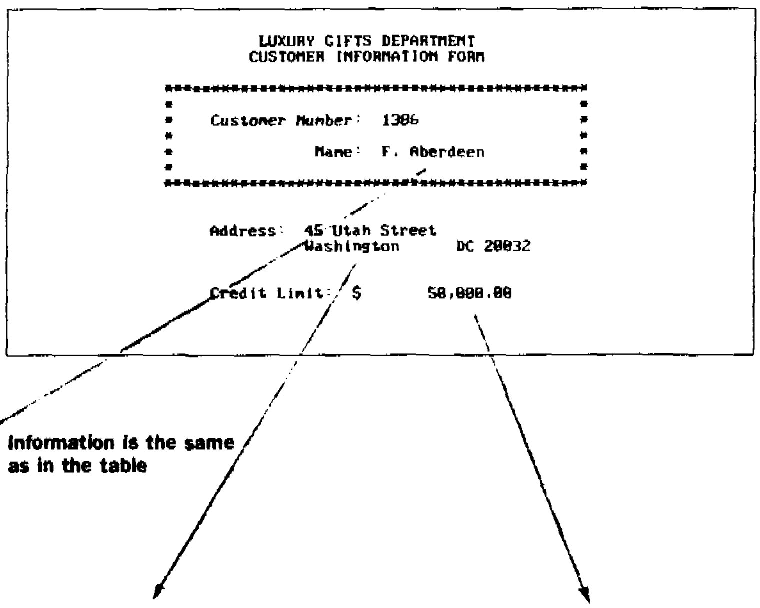
When a form is active, you can display each record from the table on it, or switch back and forth instantly between the *table view* and the *form view*. Paradox automatically creates one form for each table (you can design your own custom forms as well).

In addition to forms that display a single record from one table, you can create forms that display several records from a table or even records from several tables at once.

Figure 1-2. Table view and form view

F7 **Form Toggle**

One keystroke lets you switch between the table and the form



CUSTOMER	Cust ID	Last Name	Init	Street	City	State	Zip	Country	Credit
1	1306	Aberdeen	F	45 Utah Street	Washington	DC	20032		50,000.00
2	1300	Svenvald	I	Gouvernment House	Reykjavik			Iceland	1,250,000.00
3	1704	McDougal	L	4950 Pullman Ave NE	Seattle	WA	98105		150,000.00
4	2177	Bonnefemme	S	120 University Drive	Stanford	CA	94323		75,000.00
5	2579	Chavez	L	Cypress Drive	Palm Springs	FL	32930		250,000.00
6	2779	Fahd	S	The Palace	Riyadh			Saudi Arabia	5,000,000.00
7	3120	Elspeth, III	R	1 Hanover Square	London			England	1,000,000.00
8	3266	Hanover	A	15 State Street	Dallas	TX	75043		750,000.00
9	3271	Massey	C	29 Aragona Drive	Oxon Hill	MD	29902		1,000,000.00
10	3771	Montaigne	L	30 Tauton Drive	Bellevue	WA	98004		450,000.00
11	4277	Matthews	R	P. O. Box 20336	Albuquerque	NM	07234		50,000.00
12	4335	Farouk	K	Hotel Cairo	Cairo			Egypt	250,000.00
13	4400	Samuelson	F	Bull Run Ranch	Aurora	CO	80022		600,000.00
14	4405	Fischer	R	14 Willow Lane	Birmingham	MI	48011		1,000,000.00
15	4509	Leonardo	D	190 Via Canales	Rome			Italy	1,500,000.00
16	4700	Harris	J	Old Country Road	Atherton	CA	94322		750,000.00
17	4004	Anders	B	Jaktstigen 42	Lidingo			Sweden	300,000.00
18	5341	Chevalier	R	392 Boulevard Raspil	Montpelier			France	1,750,000.00
19	5720	Helms	D	52 Brattle Street	Cambridge	MA	02138		85,000.00
20	5855	Chin	F	Hotel Orient	Jurong			Singapore	150,000.00
21	6125	Hawes-Anderson	D	Waves Cottage	Palm Springs	FL	32302		600,000.00
22	6666	Matthews		12th Street	San Francisco	CA	94232		1,250,000.00

Queries

You *query* Paradox whenever you want to really get your hands on your data. Paradox's query ability is the heart and soul of the program. You can use queries to

- find or select information from a table
- combine information from more than one table
- perform calculations on the data in a table
- insert new data in a table or delete old data
- change selected values in a table
- define groups and sets of information on which to perform calculations and comparisons

To use this powerful feature, you fill out *query forms* onscreen. These forms specify exactly how you want to use your information in the query.

The query in Figure 1-3 tells Paradox to show the names, cities, and states of all customers who live on the West Coast (zip codes 90000 or greater) and have a credit limit of more than $100,000.

The result is displayed in a table named *Answer* because it is the answer to your query. Although this query retrieves information from only one table, you can query up to 24 concurrently open tables. Chapter 4 in the *Paradox User's Guide* tells how to query multiple tables.

Figure 1-3. A query

The question: **Which West Coast customers have a credit limit of more than $100,000?**

CUSTOMER	Cust ID	Last Name	Init	Street	City	State	Zip	Country	Credit
		√	√		√	√	>=90000		√ >100000

Each check in the query form produces a column in the Answer table.

The answer:

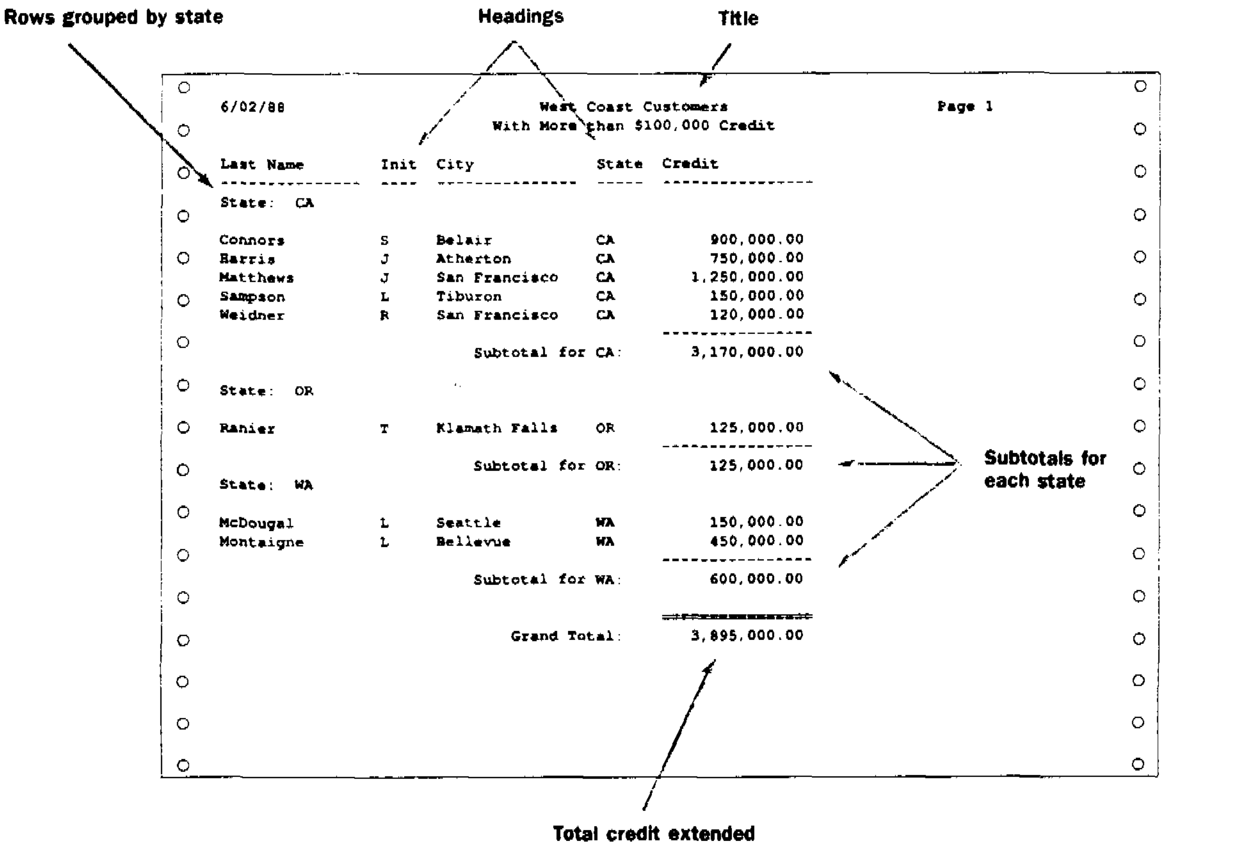
ANSWER	Last Name	Init	City	State	Credit
1	Connors	S	Belair	CA	900,000.00
2	Harris	J	Atherton	CA	750,000.00
3	Matthews	J	San Francisco	CA	1,250,000.00
4	McDougal	L	Seattle	WA	150,000.00
5	Montaigne	L	Bellevue	WA	450,000.00
6	Ranier	T	Klamath Falls	OR	125,000.00
7	Sampson	L	Tiburon	CA	150,000.00
8	Weidner	R	San Francisco	CA	120,000.00

Reports

Many people need to work with their information in printed *reports*. Paradox reports are flexible and powerful. You can easily sort and group rows, calculate fields and totals, enter titles and headings, and arrange your data in an almost infinite variety of formats, including mailing labels and form letters. You can also place fields from one table on a report for another table.

The report in Figure 1-4 presents the information from our query, sorted by state, with totals and subtotals for credit limit.

Figure 1-4. A sample report



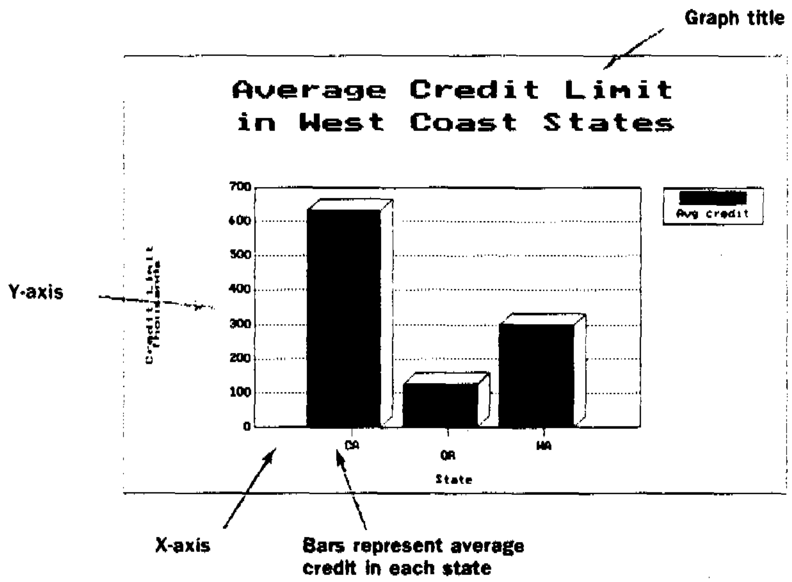
You can produce complex reports without programming

Graphs

Sometimes you can most effectively analyze your data and present it to others when you see it in its most visually powerful form: a graph. With the touch of the [Ctrl][F7] keys, Paradox draws an instant graph of your data. Other commands easily let you modify the graph type, layout, and colors to produce exactly the graph you want. You can create a graphic "slide show" for an onscreen presentation, or if you have a graphics printer or plotter, you can print your graphs. Paradox automatically updates a graph when users update the source table, in both network and standalone environments.

The graph in Figure 1-5 shows the average credit limit of customers in the West Coast states shown in our query.

Figure 1-5. A sample graph



Paradox automatically updates a graph when the data changes.

The Main menu

Moving from one Paradox operation to another is a simple matter of pressing function keys or making choices from menus. The program's *Main menu* contains the major Paradox operations.

You can make a menu selection by pressing the first letter of the command on your keyboard or by highlighting the command and pressing [Enter].

View	Ask	Report	Create	Modify	Image	Forms	Tools	Scripts	Help	Exit
		Output	Borrow	Sort	TableSize	Design	Rename	Play		No
		Design	Help	Edit	ColumnSize	Field	Query-	BeginRecord		Yes
		Tabular	DO-IT	Coedit	Format	Area	Speed	QuerySave		
		Free-form	Cancel	DataEntry	Zoom	Border	Export-	ShowPlay		
		Change		MultiEntry	Move	Page	Import	RepeatPlay		
		RangeOutput		Restructure	PickForm	Style	Copy	Editor		
		SetPrinter			KeepSet	Multi	Delete			
					Graph	Help	Info			
					Modify	DO-IT	Net			
					Load	Cancel	More			
					Save	Change	Add			
					Reset		MultiAdd			
					CrossTab		FormAdd			
					ViewGraph		Subtract			
							Empty			
							Protect			
							Directory			
							ToDOS			

Appendix C

DEVELOPMENT OF GENERALIZED COSTS FUNCTIONS AND UNIT PRICES

C.1 Introduction

The Treatment Subsystem of DEMDESS contains mathematical functions that represent generalized capital and annual facility costs for various levels of wastewater treatment. These functions were defined in detail in Section 2.5.2 of this manual. In this appendix some background information and recommendations are provided regarding the use and further development of generalized cost functions and related methodology.

The cost functions have the following general relationship:

$$C = (COEFF) \times Q^{exp}$$

Cost (C) is equal to wastewater flow (Q) raised to an exponent and then multiplied by a coefficient. The coefficient is called the coefficient of proportionality and reflects the magnitude of the cost for a given currency. The exponent is called the exponent of economy of scale. For exp=1.00, the relationship would be linear or there would not be any economy of scale. In other words, cost per unit flow would be the same at all facilities, regardless of size. For exponents of less than 1.00, cost per unit flow decreases as Q is increased and, therefore, economy of scale is indicated.

In order to define the coefficient and exponent in the relationship, two data points are required. (The relationship plots as a straight line on log-log paper.) However, it should be noted that this cost relationship is strictly empirical and as the range of flow is extended, the accuracy may drop and it may be necessary to have more than one function to define different ranges with greater accuracy. There are two basic methods to determine coefficients and exponents:

1. **Best fit plotting of existing cost data.** This method involves collecting cost data for existing facilities and possibly final estimates for planned facilities. It is necessary to reduce the costs to the same basis. Costs from different time periods need to be adjusted to a single month or year by using cost indexes. One practical problem with this approach is that the facilities are seldom the same and a breakdown of cost data is difficult to obtain. Additionally, since the design concepts may vary from one design to another even for the same effluent limit, the data are not always consistent.

2. **Design-based cost estimate.** This method involves estimating costs based on the actual design of unit processes. Materials are estimated and multiplied by unit costs. A unit cost schedule is essential. Unit processes and their costs can be combined to form different process trains and related costs. The advantage of this approach is that the cost functions have the same basis, and inconsistencies are reduced.

Comparing design-based estimates with recent existing data would provide the most reliable information.

C.2 Cost Data for Examples

As described in more detail in Section 2.5.2, cost functions are given for capital costs and annual costs.

C.2.1 Capital Costs

The cost function coefficients and exponents for Capital Construction Cost are based on generalized U.S. cost data. Construction cost represents the costs bid by U.S. contractors according to general U.S. practice. To arrive at a facility's total capital cost, a percentage allowance for contingency, engineering, fiscal, legal, and administrative costs must be added, as these additional costs are not included in the construction cost. The percentage used in the examples for this allowance is 20 percent, although a high contingency allowance may be justified in certain cases, and can be as high as 35 percent.

C.2.2 Annual Costs

The annual operation and maintenance costs are broken down into the following items:

Material and supplies: This item includes annual equipment replacement costs that can vary between 1 percent and 3 percent of the construction cost, depending on the unit process. Chemicals and other material and supplies are also included. The cost coefficients and exponents are based on U.S. data.

Energy consumption: This cost item is represented in terms of annual kilowatt-hours and is converted to currency by multiplying by the average cost of energy. As a result, local costs can be directly estimated by using local energy costs.

Manpower: This cost item is given in terms of annual labor hours and is converted to currency by multiplying by the average hourly labor cost. Local costs can be estimated by using local labor costs, but it should be noted that the labor hour data is based on U.S. practice, and it appears that U.S. treatment plants use less personnel than those in Eastern Europe. This may be due to a higher level of automation in U.S. facilities as well as higher labor costs.

C.3 Treatment Technologies

Data have been provided in the tables TPCHR and TPCST for the treatment process trains described below.

C.3.1 Domestic Wastewater Treatment

The costs for all process trains for domestic wastewater treatment include land, process and administration building, site development, and utilities. Facility sizes are based on standard sanitary engineering practice. An outfall is not included in any process train. Sludges are assumed to be landfilled adjacent to the treatment facility and, therefore, off-site sludge transport and off-site landfill costs are not included.

Primary sedimentation (PR01): This process train includes raw wastewater pumping, preliminary treatment (screening and grit removal), anaerobic sludge digestion, and sludge beds and chlorination.

Enhanced primary sedimentation (PR02): This process train is the same as PR01 except for the addition of a coagulant for enhanced sedimentation.

Conventional activated sludge (SAS01): This process train contains the same unit processes as PR01 plus activated sludge, gravity thickening of primary sludge, flotation thickening of secondary sludge, and mechanical dewatering of digested sludge.

Two-stage activated sludge (AS21): This process train is the same as SAS01 except for the addition of a second aeration and sedimentation stage. This process train may be applicable for high strength wastewaters or where nitrification is desired.

Conventional activated sludge with phosphorus removal (SAS11): This process train is the same as SAS01 except for the addition of a coagulant to the primary stage for phosphorus removal.

Conventional activated sludge with phosphorus removal and nitrification (SAS12): This process train is the same as SAS11 except for use of a high rate activated sludge stage plus a second aeration and sedimentation stage for nitrification.

Conventional activated sludge with phosphorus and denitrification (SAS13): This process train is the same as SAS12 except for the addition of multimedia filters with chemically assisted denitrification. (Biological denitrification is a feasible alternative to chemically assisted denitrification.)

C.3.2 Industrial Wastewater Treatment

Use of generalized cost functions for industrial wastewater treatment is often inappropriate because of the wide variance in manufacturing processes and other conditions from one industrial plant to another, even in the same industrial sector. Generalization of costs could be done at a unit process level such as cyanide destruction, filtration, or filter press dewatering, but the cost correlation usually needs to be done using more parameters than flow. For example, a generalized cost function for cyanide destruction would be usable only if the function were expressed in terms of flow plus the cyanide concentration. A cost function based on flow parameter alone can be given for a fixed cyanide concentration or range of concentration but will have limited usefulness beyond that range. Below are two such limited cost functions for the purpose of describing DEMDESS only. More detailed costs functions need to be developed for actual use. For industrial wastewater treatment (or process level measures), actual cost estimates on a case-by-case basis would be the most reliable and should be preferred to cost functions. However, for scenario analyses in which a lot of processes may be screened for a large number of sites prior to design, general cost functions would be more practical.

Chemical clarification (ITP01): This process train includes screening, wastewater pumping, chemically assisted clarification, sludge thickening, and mechanical dewatering. Ultimate sludge disposal is not included. The costs are valid for influent suspended solids concentration of 150 to 300 mg/L (measured after pH neutralization).

Metals removal with chrome reduction and cyanide destruction (ITP02): This process train is given as an example of the relative complexity of industrial wastewater treatment and is based on an actual case. The process train includes treatment facilities to handle three wastewater streams that have different treatment requirements. The waste streams are cyanide-bearing wastewaters, hexavalent-chrome-bearing wastewaters, and acid/alkali wastewaters, and in this case occur at about equal volumes. The treatment includes a two-stage cyanide destruction (by chlorination) train and a chrome reduction train that both discharge to a neutralization and precipitation train that also receives the acid/alkali wastewaters. Average cyanide load is 15 mg/L, hexavalent chrome 10 to 35 mg/L, and other metals 33 mg/L. Segregated wastewater piping and pumping systems, equalization tanks for the three wastewater streams, treatment facility building, and all auxiliary equipment are included. Sludge is mechanically dewatered but ultimate sludge disposal is not included.

It is, of course, unlikely one would encounter a similar treatment case during application of DEMDESS and, as noted earlier, actual cost estimates or unit process based cost functions would be needed.

C.4 Recommendations

The recommendations given below are aimed at initial demonstration and also at further development of the cost estimating capabilities of DEMDESS.

C.4.1 Domestic Wastewater Treatment Cost Functions

The process trains and related cost coefficients and exponents given here should be used for demonstration of DEMDESS methodology only. The U.S.-based costs could be considered as world prices and may be acceptable in a straight economic comparison of scenarios. In a financial analysis that involves local costs such as taxes, user charges, and penalties, the U.S.-based costs and particularly construction and material and supplies costs would most likely introduce significant errors.

Local cost data should be developed by all users. In all countries there is a considerable amount of raw data on existing facilities and many completed treatment plant designs. The most significant problem in compiling the data may be the indexing of the costs so that they can be put on the same basis and also be used to estimate costs at different dates. The rapid inflation in the countries seems to have made many cost data unreliable.

To update construction cost data, it may be necessary to go back to the materials estimate level. One approach would be to take one or preferably several sizes of a standard activated sludge treatment plant and estimate costs periodically based on actual unit prices. This method would give a treatment plant construction cost index. Other cost indexes, such as a general construction cost index, can also be used once a correlation is established with the treatment plant construction cost index. Data for annual costs should be accurately defined. For example, care should be taken to define material unit costs such as chemical costs as free market costs or subsidized costs. If a chemical is imported but subsidized, two separate unit costs are involved. There may also be differences in free market labor costs versus those for labor provided by government workers. Also recommended is that a separate chemical cost function be developed rather than one materials and supplies cost function.

C.4.2 Industrial Wastewater Treatment Cost Functions

The cost data given here for two process trains are for demonstration purposes only. As stated earlier, there is great variance in industrial wastewaters. Cost estimates on a case-by-case basis should be used, or cost functions should be developed at the unit process level (such as equalization, pH control, chemical addition, clarification, and so on) and should be based on actual design and material estimates. Again, the need to index costs is emphasized.

Appendix D

DEMDESS DATA BASE DICTIONARY

The DEMDESS data base dictionary consists of descriptions of all data elements in each DEMDESS table. DEMDESS users should reference this dictionary to make sure they retain the DEMDESS data base design while using the system, particularly when adding or updating data and enhancing DEMDESS. Developers of DEMDESS applications willfind this dictionary useful as a reference when designing, implementing, and testing software. More detailed descriptions of the DEMDESS data base dictionary can be found in Chapter 2.

Fields in Table DISCHX

Field	Format	Units	Description
DISCHID	A10	-	User-defined unique identifier of a discharger.
TREATCLASS	N	-	Flow-stream path number, linked to the emissions record in Table EMISSX.
COUNTRY	A1	-	B=Bulgaria, H=Hungary, R=Romania, S=Slovakia.
INDCODE	N	-	Industrial classification code; see Table INDCODES.
DISCHID-TO	A10	-	For an industrial discharger, the DISCHID of the municipal treatment plant; see definition sketch on Figure 13.
BASIN	N	-	River basin ID number; see Table RCHFILE.
SEG	N	-	River segment number within a river basin.
KM	N	-	River kilometer, from downstream end of segment.
POP	N	-	Population.
QTOT	N	cmd	Average daily flow; for municipalities, includes industrial, residential, infiltration/inflow, etc.
IALTID	N	-	Discharger ID in country's originating data base.
NAME	A55	-	Name and description of discharger.
TAXID	A10	-	Identifier of tax record in Table TAXES.
FINEID	A10	-	Identifier of fines record in Table FINES.
WQSTD	A5	-	Identifier of effluent quality standard record in WQSTDS.
TPID	A10	-	Identifier of wastewater treatment plant location; see definition sketch in Figure 13.
TPLEV	A5	-	Identifier of treatment level record in Table TPCST.
PTID	A10	-	Identifier of pretreatment plant location; see Definition Sketch on Figure 13.

Fields in Table DISCHX (cont'd)

Field	Format	Units	Description
PTLEV	A5	-	Identifier of pretreatment level in Table TPCST.
KM1	N	-	Cross-reference to river location using hydrologic sequence logic used within a particular country.
KM2	N	-	In Slovakia: the Mahovlic system in which KM1 is River km on the Danube from the downstream end,
KM3	N	-	and KM2,...,KM7 are River km on lower-order tributaries of the Danube.
KM4	N	-	
KM5	N	-	
KM6	N	-	
PROVINCE	A20	-	Province code or name for querying the data base.
BASINAUTH	A20	-	River basin authority.
RGNAUTH	A20	-	Regional authority.
WTRAUTH	A20	-	Water supply or water resources authority.
MUNAUTH	A20	-	Municipal authority.
PERMIT	A10	-	Identification code of the discharge permit.
PERMITAUTH	A20	-	Authority issuing the discharge permit.
QCAP	N	cmd	Existing capacity of wastewater treatment plant, either pretreatment or treatment, depending on INDCODE (=22 for municipalities).
QPEAK	N	cmd	Peak hydraulic flow into the wastewater treatment plant.
DAYSOP	N	days	Days per year of operation.
HRSOP	N	hours	Average hours per day of operation.
QRES	N	cmd	Residential flow.
QNONRES	N	cmd	Nonresidential flow; includes commercial, institutional, and infiltration/inflow.
QIND	N	cmd	Total industrial flow entering a municipal plant.
COLL-LENGTH	N	km	Length of sewers in the collection system.

Fields in Table DISCHX (cont'd)

Field	Format	Units	Description
STREET-LENGTH	N	km	Length of streets in a municipality.
PCTSEWERED	N	%	Percentage of population served by a municipal collection system.
WTRSPPLYID	A10	-	Identification code for the water supply authority.
WTRSPPLYID-TO	A10	-	Discharger ID who receives effluent as its (discharger's) supply.
WTRSPPLYTYPE	A1	-	User identification of type of supply (surface supply, groundwater supply, treated, etc.).
WTRSPPLYBASIN	N	-	River basin identifier in RCHFILE.
WTRSPPLYSEG	N	-	River segment identifier in RCHFILE.
WTRSPPLYKM	N	-	River kilometer, from downstream end of reach.

****

Fields in Table EMISSX

Field	Format	Units	Description
DISCHID	A10	-	Identification code of the discharger.
TREATCLASS	N	-	Flow path number of the discharger.
PARMCODE	N	-	Parameter code number from Table PARMTABL.
DATASRC	A6	-	Data source identification code.
QCCODE	A6	-	Quality control code (userdefined).
VALUE	N	Var's	Value of the water-quality parameter.

DATE	A6	-	Date (yymmdd) when the sample was taken.
TIME	A4	-	Time (hhmm) when the sample was taken.
DISCHTYPE	A2	-	Flow-routing code defined in Figure 13.

Fields in Table MODLCOEF

Field	Format	Units	Description
PARMCODE	N	-	Parameter code, as defined in Table PARMTABL.
Ka	N	/day	Coefficient Ka in the formula K1 [decay/day] = Ka +
Kb	N	/day	+ Kb * exp(...
Kc	N	/C	exp (Kc T) +
Kd	N	/d-C	+ Kd T, where T is temperature in degrees Celsius.

****

Fields in Table MODLLOAD

Field	Format	Units	Description
BASIN	N	-	River basin code, as defined in Table RCHFILE.
SEG	N	-	River segment code, as defined in Table RCHFILE.
KM	N	km	River km from downstream end of river SEG.
CONDCODE	A6	-	User code for physical scenario (wet season, etc.).
PARMCODE	N	-	Water-quality parameter code, as in Table PARMTABL.
BKCONC	N	mg/L	Background concentration of the parameter.
NPSLOAD	N	kg/d	Nonpoint source loading in lateral inflows.
PSLOAD	N	kg/d	Pointsource loading at the given river location.

****

Fields in Table MODLRCH

Field	Format	Units	Description
BASIN	N	-	River basin code in Table RCHFILE.
SEG	N	-	River segment code in Table RCHFILE.
KMDS	N	km	Upstream end of reach to which the parameters and.

			coefficients apply.
KMUS	N	km	Downstream end of the river reach (measured in km from mouth of river SEG).
CONDCODE	A6	-	User code defining the physical scenario (low flow, etc.).
Q	N	cmd	River stream flow
VEL	N	m/s	Flow velocity.
K2	N	/day	Reaeration coefficient in the Streeter-Phelps equation for DO profile.

****

Fields in Table PERMITS

Field Name	Format	Units	Description
PERMIT	A10	-	Identification code of the discharge permit.
TREATCLASS	N	-	Treatment class or flow path, as in DISCH1 table.
PERMITAUTH	A20	-	Authority responsible for the discharge permit.
PARMCODE	N	[1]	Parameter code, as in PARMTABL.
PERMITLIMIT	N	[1]	Maximum allowable limit, or lower limit of range.
PERMITLIMIT2	N	[1]	Upper limit of range (such as pH).
UNITS	A4	-	Units for the parameter, as in the discharge permit.

****

Fields in Table RCHFILE

Field	Format	Units	Description
BASIN	N	-	The numerical code designation for the major river basin.
SEG	N	-	The segment number of the river. All reaches of a river have the same SEG value.
KM	N	Km	The river kilometer of the downstream point of the reach.
TYPE	A1	-	The reach type code. S=start T=terminal R=regular
USBASIN1	N	-	The BASIN code of the mainstem upstream reach.
USSEG1	N	-	The SEG code of the mainstem upstream reach.
USKM1	N	Km	The KM value of the mainstem upstream reach.
USBASIN2	N	-	The BASIN code of the tributary upstream reach.
USSEG2	N	-	The SEG code of the tributary upstream reach.
USKM2	N	Km	The KM value of the tributary upstream reach.
USBASIN3	N	-	The BASIN code of a second tributary reach; usually not used.
USSEG3	N	-	The SEG code of a second tributary reach; usually not used.

Fields in Table RCHFILE (cont'd)

Field	Format	Units	Description
USKM3	N	Km	The KM value of a second tributary reach; usually not used.
DSBASIN	N	-	The BASIN code of the reach downstream.
DSSEG	N	-	The SEG code of the reach downstream.
DSKM	N	Km	The KM value of the reach downstream.
CBASIN	N	-	The BASIN code of the downstream complement reach, i.e., the other reach entering the d.s. junction.
CSEG	N	-	The SEG code of the downstream complement reach.
CKM	N	Km	The KM value of the downstream complement reach.
LEV	N	-	The stream level of the reach.
J	N	-	The LEV value of the reach downstream.
SEQNO	N	-	The hydrologic sequence number; important for sorting in hydrologic order.
SEGL	N	Km	The length of the reach in kilometers.
WQSTATUS	A3	-	The current water quality status; links to WQSTDS.
WQSTD	A3	-	The current water quality standard for the reach; links to WQSTDS.
LOWQ	N	m3/s	The low flow value for the reach.

Fields in Table RCHFILE (cont'd)

Field	Format	Units	Description
COUNTRY	A1	-	The country code for the reach (the country th reach is in).
PROVINCE	N	-	The province code for the reach.
PATHMILE	N	Km	The same as KM, except in Bulagria; PATHMILE will be eliminated in future DEMDESS versions.
RCHNAME	A30	-	The name of the river.
Rchjuncdesc	A52	-	A description of the junction at the downstream end of the reach.
Bpattern	A20	-	Used for display of schematic diagrams of the river network; sort by SEQNO to use.

Fields in Table RCHFILE (cont'd)

Field	Format	Units	Description
KM1	N	Km	KM1-KM7 are the host country river kilometer index values. A first step in building the Reach File
KM2	N	Km	is to load the KM1-KM7 fields and then run the DEMDESS Reach File maker software to build the full
KM3	N	Km	Reach File. After the Reach File is built, the KM1-KM7 field values are used to assign other tables
KM4	N	Km	to the Reach File (such as the DISCHX and WQSITES Tables).
KM5	N	Km	
KM6	N	Km	
KM7	N	Km	
CODUL1	S	-	CODUL1-CODUL7 are used for the Romanian values that define the river network. These values
CODUL2	S	-	are used in a similar manner as the KM1-KM7 fields.
CODUL3	S	-	
CODUL4	S	-	
CODUL5	S	-	
CODUL6	S	-	
CODUL7	S	-	
USKM	N	Km	The upstream river kilometer value from the Romanian river network.

Fields in Table WQDATA

Field	Format	Units	Description
WQSITEID	A10	-	Identification code for the water quality monitoring site, as in WQSITES.
PARMCODE	N	-	Water quality parameter code defined in PARMTABL.
SAMPDATE	A6	yymmdd	Date on which sample was taken.
SAMPTIME	A4	hhmm	Time at which sample was taken.
LOCCODE	A1	-	Location code (left bank, surface, bottom, etc.).
QCCODE	A6	-	Quality control code.
VALUE	N	[1]	Measured value of the parameter.
VAL-RMK	A1	-	Remarks code.

****

Fields in Table WQSITES

Field	Format	Units	Description
WQSITEID	A10	-	Site identification code for a monitoring station.
DATASRC	A6	-	Identification code for source of data.
BASIN	N	-	River basin code defined in the Table RCHFILE.
SEG	N	-	River segment code defined in the Table RCHFILE.
KM	N	km	River km from mouth of river segment.
PROVINCE	A10	-	Province name.
REGION	A10	-	Region name.
COUNTRY	A1	-	Country code.
STATYPE	A6	-	Code for station type.
AGENCY	A10	-	Operating agency.
STARTDATE	A6	yymmdd	Beginning date of monitoring record.
ENDDATE	A6	yymmdd	Ending date of monitoring record.
WQSTADESC	A80	-	Description of the water quality monitoring station.

Appendix E

DEMDESS LOADING ROUTINES

Three types of procedures are used to load data into DEMDESS:

1. Software conversion of existing country data bases into the DEMDESS data bases;
2. Data entry input of data from country-specific forms to data files that can then be loaded into DEMDESS; and
3. Manual data entry directly into DEMDESS tables.

The following is a description of the procedures and methods used to load the various data tables into DEMDESS. Country-specific software modules are also described.

1. General tables that are commonly used by all countries:

   - INDCODES
   - PARMTABL
   - CURRCNVT
   - BASINTBL

   Loading method: Manual data entry directly into DEMDESS.

2. Tables currently loaded with general data that can contain country-specific data:

   - TPCST
   - TPCHR

   Loading method: Manual data entry directly into DEMDESS.

3. Tables containing country-specific data based on regulations:

   - WQSTDS
   - WQSTDESC
   - TAXES
   - FINES

   Loading method: Manual data entry directly into DEMDESS.

4. Tables loaded from existing administrative procedures:

   - DISCHX

- EMISSX
- RCHFILE
- WQSITES
- WQDATA

Loading methods: Conversion software and custom data entry procedures.

5. Country-specific DEMDESS loading software. *Note*: Programs with the extension ".SC" denote Paradox PAL scripts; programs with the extension ".BAS" denote Basic programs. The Basic programs produce ASCII data files that can be directly imported into the DEMDESS tables. In all cases the input files are derived directly from the existing administrative routines used in the specific country involved.

1. Bulgaria's Software

   a. BULGLOAD.BAS—Prepares the Bulgaria DISCHX and EMISSX data for loading into DEMDESS.

   b. BULGRCHL.BAS—Converts the Bulgarian river network into the DEMDESS Reach File.

   c. DENTR.SC—A fully developed stand-alone data entry system that inputs the Bulgarian "special study" data into Paradox tables that are then converted and loaded into DEMDESS.

   d. DXFLLCNV.BAS—Converts the stream and basin boundary data digitized using AUTOCADD into PC Reach File-compatible trace files. This program also converts from AUTOCADD x-y coordinates to latitude/longitude.

2. Slovakia's Software

   a. SLOVWQ.BAS—Prepares the Slovak Hydrometeorological Institute water quality file for loading into the DEMDESS WQSITES and WQDATA tables. *Note*: Reach File assignments in WQSITES were performed manually.

   b. DASSIGNS.SC—Assigns Reach numbers to DISCHX records using the Slovak river kilometer values.

   c. DISCHCNV.SC—Converts the Slovak discharge file to the DEMDESS DISCHX and EMISSX tables.

   d. RFMAKES.SC—Converts the Slovak Water Research Institute river kilometer network into the DEMDESS RCHFILE.

3. Romania's Software

   a. RFMAKER.SC—Builds the DEMDESS RCHFILE table using the Romanian river kilometer network. Data entry for the Romanian river network is performed using a specially designed Paradox form and procedure. *Note*: Manual data entry was performed by WASH on the Arges basin for conversion to the RCHFILE.

b. DASSIGNR.SC—Assigns dischargers to the Romanian Reach File using the Romanian river kilometer values.

c. TXTCODUL.SC—Makes the Romanian "Codul" values compatible between the RCHFILE and DISCHX for use in DASSIGNR.SC.

d. DLOADR.SC—Converts the Romanian discharger data to DISCHX and EMISSX. *Note*: This program is not finished because the Romanian administrative procedures are new and not yet loaded with data. The Romanian staff performed direct manual data entry into the DEMDESS DISCHX table for the existing dischargers in the Arges basin.

4. Hungary's Software

a. RFMAKEH.SC—Makes the Hungarian RCHFILE table using the Hungarian river kilometer network.

b. DASSIGNH.SC—Assigns dischargers to the Hungarian RCHFILE.

c. HWQDATA.SC—Converts the VITUKI water quality data to the DEMDESS WQSITES and WQDATA tables.

d. DLOADH.SC—Converts the VITUKIdischarger data to the DEMDESS DISCHX and EMISSX tables. *Note*: This program is not finished because the Vituki procedures are new and data are not yet available for final development and testing.

Listings of the DEMDESS Loading Software

1.a BULGLOAD.BAS

```
REM $DYNAMIC
DEFINT I-N
'Program BULGLOAD.BAS
'program to read and process Bulgaria emissions data
'written by Tim Bondelid for WASH Task 271 - Danube
'November 1991
DIM q(3), c(20), bulbasin$(15), b#(15)
CLS
year$ = "1990"
country$ = "B" 'Bulgaria country code
'list of Bulgarian basin names (6 char max for filenames)
bulbasin$(1) = "DANUBE"
bulbasin$(2) = ""
bulbasin$(3) = "VOJNSH"
bulbasin$(4) = "VIDBOL"
bulbasin$(5) = "ARSHAR"
bulbasin$(6) = "SKOMLA"
bulbasin$(7) = "LOM   "
bulbasin$(8) = "ZIBRIZ"
bulbasin$(9) = "OGOSTA"
bulbasin$(10) = "ISKAR "
bulbasin$(11) = "VIT   "
bulbasin$(12) = "OCAM  "
bulbasin$(13) = "JANTRA"
bulbasin$(14) = "RUSENS"
bulbasin$(15) = ""
'list of Bulgarian basin codes
b#(1) = 30001000 '"DANUBE"
b#(2) = 0        '""
b#(3) = 30003000 'VOINISKA
b#(4) = 30004000 '"VIDBOL"
b#(5) = 30005000 '"ARSHAR"
b#(6) = 30006000 '"SKOMLA"
b#(7) = 30007000 '"LOM   "
b#(8) = 30008000 '"ZIBRIZ"
b#(9) = 30009000 '"OGOSTA"
b#(10) = 30010000 '"ISKAR "
b#(11) = 30011000 '"VIT   "
b#(12) = 30012000 '"OCAM  "
b#(13) = 12 '30017000 '"JANTRA"
b#(14) = 30018000 '"RUSENS"
```

```
b#(15) = 0          '""

'remove tabs in disch name file
FOR nbasin = 13 TO 13 '1 TO 15 'loop on basins
IF bulbasin$(nbasin) > " " THEN

basin$ = bulbasin$(nbasin)
indname$ = "PI" + basin$ + ".txt"
OPEN "i", #4, indname$
indoutname$ = "PI" + basin$ + ".tx"
OPEN "O", #5, indoutname$
PRINT indoutname$
ndischarger = 0 'discharger number
DO UNTIL EOF(4)
      'read disch name
      a$ = INPUT$(1, 4)
      IF a$ = CHR$(9) THEN a$ = "        "
      PRINT #5, a$;
LOOP

CLOSE #5 'close output file
CLOSE #4 'close input disch name file

END IF 'if bulbasin$>""

NEXT nbasin

'read and process the emissions files
exfile$ = "exemiss.txt"
dfile$ = "disch.txt"
OPEN "o", #2, exfile$
OPEN "o", #3, dfile$


FOR nbasin = 13 TO 13 '1 TO 15 'loop on basins
IF bulbasin$(nbasin) > " " THEN

basin$ = bulbasin$(nbasin)
infile$ = "PR" + basin$ + ".txt"
OPEN "i", #1, infile$
indname$ = "PI" + basin$ + ".tx"
OPEN "i", #4, indname$
PRINT infile$
DATASRC$ = "INSPCT"
QCCODE$ = " "
ndischarger = 0 'discharger number
DO UNTIL EOF(1)
```

```
  ndischarger = ndischarger + 1
  ndisch$ = STR$(ndischarger)
  IF ndischarger < 10 THEN
    ndisch$ = "00" + MID$(ndisch$, 2, 1)
  ELSEIF ndischarger < 100 THEN
    ndisch$ = "0" + MID$(ndisch$, 2, 2)
  ELSEIF ndischarger < 1000 THEN
    ndisch$ = MID$(ndisch$, 2, 3)
  ELSE
    ndisch$ = "XXX"
  END IF
  dischid$ = basin$ + ndisch$

  FOR i = 1 TO 3 'read 3 lines for each discharger
    INPUT #1, q(i), c(1), c(2), c(3), c(4), c(5), c(6), c(7), c(8),
c(9), c(10), c(11), c(12), c(13), c(14), c(15), c(16), c(17),
c(18), c(19), c(20), iseg, indcode
    IF i = 1 THEN 'write disch record
      'read disch name
      LINE INPUT #4, dischname$
      dischnme$ = MID$(dischname$, 1, 28)
      dischnmc$ = MID$(dischname$, 71, 31)
      dischnme$ = RTRIM$(dischnme$)
      dischnmc$ = LTRIM$(dischnmc$)
      'extract date
      SDATE$ = MID$(dischname$, 63, 2) + MID$(dischname$, 57, 2) +
MID$(dischname$, 60, 2)
      'PRINT dischname$
      'PRINT dischnme$
      'PRINT dischnmc$
    END IF
    IF i = 3 THEN 'write disch record
      qtot = q(1) + q(2) + q(3)
      rchkm = 0
      WRITE #3, dischid$, country$, qtot, indcode, b#(nbasin),
iseg, rchkm, dischnme$, dischnmc$
    END IF
    IF q(i) > 0 THEN
      itreat = i
      iparm = 1000 'flow is parm= 1000
      WRITE #2, dischid$, itreat, iparm, DATASRC$, QCCODE$, q(i),
SDATE$'print flow to emission file
      FOR j = 1 TO 20
        IF c(j) > 0 THEN WRITE #2, dischid$, i, j, DATASRC$,
QCCODE$, c(j), SDATE$
      NEXT j
    END IF
```

```
  NEXT i
LOOP

CLOSE #1 'close input file
CLOSE #4 'close input disch name file

END IF 'if bulbasin$>""
NEXT nbasin
CLOSE #2, #3 'close output files
END
```

1.b BULGRCHL.BAS

```
REM $DYNAMIC
DEFINT I-N
'Program BULGRCHL.BAS
'program to read and process Bulgaria emissions data
'written by Tim Bondelid for WASH Task 271 - Danube
'November 1991
DIM q(3), c(20), bulbasin$(15), b#(15), nsegs(15)
DIM iusnode(50, 3), lowq#(50), segl(50), iwqclass(50), iprovince(50)
DIM idsnode(50), lev(50), jlev(50), iseqno(50), istatype(50), idanconn(15), iusbasin(50, 3)
DIM pmile(50), pmlev(10), namelev$(10), rchname$(50), juncdesc$(50), hydroschem$(50)
CLS
year$ = "1990"
country$ = "B" 'Bulgaria country code
'list of Bulgarian basin names (6 char max for filenames)
bulbasin$(1) = "DANUBE"
bulbasin$(2) = ""
bulbasin$(3) = "VOJNSH"
bulbasin$(4) = "VIDBOL"
bulbasin$(5) = "ARSHAR"
bulbasin$(6) = "SKOMLA"
bulbasin$(7) = "LOM   "
bulbasin$(8) = "ZIBRIZ"
bulbasin$(9) = "OGOSTA"
bulbasin$(10) = "ISKAR "
bulbasin$(11) = "VIT   "
bulbasin$(12) = "OCAM  "
bulbasin$(13) = "JANTRA"
bulbasin$(14) = "RUSENS"
bulbasin$(15) = ""
'list of Bulgarian basin codes
```

```
b#(1)  = 30001000 '"DANUBE"
b#(2)  = 0        '""
b#(3)  = 30003000 'VOINISKA
b#(4)  = 30004000 '"VIDBOL"
b#(5)  = 30005000 '"ARSHAR"
b#(6)  = 30006000 '"SKOMLA"
b#(7)  = 30007000 '"LOM   "
b#(8)  = 30008000 '"ZIBRIZ"
b#(9)  = 30009000 '"OGOSTA"
b#(10)  = 30010000 '"ISKAR "
b#(11)  = 30011000 '"VIT   "
b#(12)  = 30012000 '"OCAM  "
b#(13)  = 12  '30017000 '"JANTRA"
b#(14)  = 30018000 '"RUSENS"
b#(15)  = 0        '""
'set Danube mainstem downstream connecting reaches
idanconn(1)  = 0
idanconn(2)  = 0
idanconn(3)  = 5
idanconn(4)  = 6
idanconn(5)  = 7
idanconn(6)  = 8
idanconn(7)  = 10
idanconn(8)  = 11
idanconn(9)  = 13
idanconn(10)  = 15
idanconn(11)  = 16
idanconn(12)  = 17
idanconn(13)  = 21
idanconn(14)  = 22
idanconn(15)  = 0

rchfile$ = "RCHFILE.TXT"
OPEN "O", #2, rchfile$

FOR nbasin = 13 TO 13 '1 TO 15 'loop on basins to get nsegs by
basin
IF bulbasin$(nbasin) > " " THEN
basin$ = bulbasin$(nbasin)
infile$ = "PU" + basin$ + ".txt"
OPEN "i", #1, infile$
iseg = 0 'segment number
DO
  iseg = iseg + 1
  LINE INPUT #1, a$
LOOP UNTIL EOF(1)'until eof(1)
nseg = iseg - 1 'number of segments
```

```
nsegs(nbasin) = nseg
CLOSE #1 'close input file
END IF 'if bulbasin$>""
NEXT nbasin


FOR nbasin = 13 TO 13 '1 TO 15 'loop on basins
IF bulbasin$(nbasin) > " " THEN

basin$ = bulbasin$(nbasin)
infile$ = "PU" + basin$ + ".txt"
OPEN "i", #1, infile$

'nbasin is the rchbasin number
iseg = 0 'segment number
DO

  iseg = iseg + 1
  PRINT iseg
  LINE INPUT #1, a$
  istatype(iseg) = VAL(MID$(a$, 54, 1))
  iusnode(iseg, 1) = VAL(MID$(a$, 56, 5))
  iusnode(iseg, 2) = VAL(MID$(a$, 61, 5))
  iusnode(iseg, 3) = VAL(MID$(a$, 66, 5))
  lowq#(iseg) = VAL(MID$(a$, 72, 12))
  segl(iseg) = VAL(MID$(a$, 88, 5))
  iwqclass(iseg) = VAL(MID$(a$, 95, 1))
  iprovince(iseg) = VAL(MID$(a$, 100, 2))
  IF iprovince(iseg) = 9 THEN iprovince(iseg) = 5
  rchname$(iseg) = MID$(a$, 103, 20)
  juncdesc$(iseg) = MID$(a$, 1, 53)
  IF nbasin <> 1 THEN
    iseqno(iseg) = (idanconn(nbasin) - 2) * 100 + iseg
  ELSE
    iseqno(iseg) = (iseg - 1) * 100'check this
    IF iseg = 1 THEN iseqno(iseg) = 1
  END IF

LOOP UNTIL EOF(1)'until eof(1)
nseg = iseg - 1 'number of segments
CLOSE #1 'close input file

'compute other reach data
IF nbasin <> 1 THEN 'trib to Danube
  iclev = 1 'current level
  FOR iseg = 1 TO nseg
    IF iusnode(iseg, 2) = iusnode(iseg, 1) THEN iusnode(iseg, 2) =0
```

```
    IF iusnode(iseg, 3) = iusnode(iseg, 1) THEN iusnode(iseg, 3) =0
    IF iusnode(iseg, 3) = iusnode(iseg, 2) THEN iusnode(iseg, 3) =0
    FOR i = 1 TO 3
      IF iusnode(iseg, i) > 0 THEN idsnode(iusnode(iseg, i)) = iseg
    NEXT i
    IF istatype(iseg) = 1 THEN
      iclev = iclev + 1
    ELSEIF istatype(iseg) = 3 THEN 'criteria to step down a level
      iclev = iclev - 1
    ELSEIF istatype(iseg) = 4 THEN 'criteria to step down a level
      iclev = iclev - 2
    END IF
    lev(iseg) = iclev
  NEXT iseg
  'assign jlev - level of rch downstream
  FOR iseg = 1 TO nseg
    IF idsnode(iseg) > 0 THEN
      jlev(iseg) = lev(idsnode(iseg))
    ELSEIF iseg = nseg THEN
      jlev(iseg) = 1
    ELSE
      jlev(iseg) = 0
      PRINT "jlev(iseg)=0 "; iseg
      INPUT "hit <cr>"; a$
    END IF
    IF iseg = nseg THEN jlev(iseg) = 1
  NEXT iseg

ELSEIF nbasin = 1 THEN 'Danube mainstem
  FOR iseg = 1 TO nseg
    IF iusnode(iseg, 2) = iusnode(iseg, 1) THEN iusnode(iseg, 2) =0
    IF iusnode(iseg, 3) = iusnode(iseg, 1) THEN iusnode(iseg, 3) =0
    IF iusnode(iseg, 3) = iusnode(iseg, 2) THEN iusnode(iseg, 3) =0
    FOR i = 1 TO 3
      IF iusnode(iseg, i) > 0 THEN idsnode(iusnode(iseg, i)) = iseg
    NEXT i
    lev(iseg) = 1
    jlev(iseg) = 1
  NEXT iseg
  jlev(nseg) = 0'end of Danube at Black Sea
END IF
'rchfile$ = "RF" + basin$ + ".txt"
'OPEN "O", #2, rchfile$
  'compute path mile
  FOR i = 1 TO 10
    pmlev(i) = 0
    namelev$(i) = ""
```

```
  NEXT i
  FOR iseg = 1 TO nseg
    jseg = nseg - iseg + 1 'work from bottom to top
    IF jlev(jseg) < lev(jseg) THEN 'new path
      pmlev(lev(jseg)) = 0
      namelev$(lev(jseg)) = rchname$(jseg)
    END IF
    IF namelev$(lev(jseg)) <= " " AND rchname$(jseg) > " " THEN
namelev$(lev(jseg)) = rchname$(jseg) 'for mainstems or name changes
in middle of path
    pmile(jseg) = pmlev(lev(jseg))
    pmlev(lev(jseg)) = pmlev(lev(jseg)) + segl(jseg) 'accumulate
path miles
    IF   rchname$(jseg)   <=   "   "   THEN   rchname$(jseg)   =
namelev$(lev(jseg))
  NEXT iseg
rchkm = 0
FOR iseg = 1 TO nseg
  rchtype$ = "R"
  IF iusnode(iseg, 1) = 0 THEN rchtype$ = "S"
  FOR i = 1 TO 3
    iusbasin(iseg, i) = 0
    IF iusnode(iseg, i) > 0 THEN iusbasin(iseg, i) = nbasin
  NEXT i
  IF nbasin = 1 THEN
    IF iseg = nseg THEN rchtype$ = "T" 'last rch of Danube
    'assign tributary usnode of Danube
    FOR j = 1 TO 15
      IF idanconn(j) = iseg THEN
        iusbasin(iseg, 2) = j
        iusnode(iseg, 2) = nsegs(j)
      END IF
    NEXT j
  END IF

  'set connectivity esp. for Danube connections
  idsbasin = nbasin
  IF iseg = nseg THEN
    idsbasin = 1
    'need to set Danube dsnode number here
    IF nbasin <> 1 THEN
      idsnode(iseg) = idanconn(nbasin)
    ELSE 'Danube mainstem
      idsnode(iseg) = 0
    END IF
  END IF
  'make branching pattern
```

```
  bpattern$ = "                    " '20 chars for now
  levi = lev(iseg)
  FOR i = 1 TO levi
    lev2 = (i - 1) * 2 + 1
    MID$(bpattern$, lev2, 1) = "|"
  NEXT i
  IF rchtype$ = "S" THEN
    lev2 = (lev(iseg) - 1) * 2 + 1
    MID$(bpattern$, lev2, 1) = "+"
  END IF
  IF lev(iseg) > 1 AND jlev(iseg) < lev(iseg) THEN
    lev2 = (lev(iseg) - 1) * 2
    MID$(bpattern$, lev2, 1) = "_"
    lev2 = (jlev(iseg)) * 2
    MID$(bpattern$, lev2, 1) = "_"
    'next is for reach at 3-way junction
    'IF iseg < nseg AND jlev(iseg + 1) = (lev(iseg + 1) - 2) THEN
    '  lev2 = (jlev(iseg)) * 2
    '  MID$(bpattern$, lev2, 1) = " "
    'END IF
    IF iseg < nseg AND jlev(iseg) = (lev(iseg) - 1) AND jlev(iseg)
<> (lev(iseg + 1)) THEN
      lev2 = (jlev(iseg)) * 2
      MID$(bpattern$, lev2, 1) = " "
    END IF
  END IF
  'WQSTATUS AND WQSTD
  IF iwqclass(iseg) = 1 THEN
    wqstd$ = "B1"
    wqstatus$ = "B1"
   ELSEIF iwqclass(iseg) = 2 THEN
    wqstd$ = "B2"
    wqstatus$ = "B2"
   ELSEIF iwqclass(iseg) = 3 THEN
    wqstd$ = "B3"
    wqstatus$ = "B3"
  END IF
  'write reach structure record
  WRITE #2, b#(nbasin), iseg, rchkm, rchtype$, b#(iusbasin(iseg,
1)), iusnode(iseg, 1), rchkm, b#(iusbasin(iseg, 2)), iusnode(iseg,
2), rchkm, b#(iusbasin(iseg, 3)), iusnode(iseg, 3), rchkm,
b#(idsbasin), idsnode(iseg), rchkm, cbasin, cseg, ckm, _
lev(iseg), jlev(iseg), iseqno(iseg), segl(iseg), wqstatus$, wqstd$,
lowq#(iseg), country$, iprovince(iseg), pmile(iseg),
rchname$(iseg), juncdesc$(iseg), bpattern$
NEXT iseg
'CLOSE #2  'close output file
```

```
END IF 'if bulbasin$>""
NEXT nbasin
CLOSE #2  'close output file

END
```

1.c DENTR.SC

Not included; see DEMDESS distribution disk.

1.d DXFLLCNV.BAS

```
'dxfllcnv.bas
'extract lines from DXF files, convert to lat/lon
   DIM ax#(4), ay#(4), c#(4), b#(4, 4), XP#(4, 4), YCOEF#(4),
XCOEF#(4)
   DIM LX#(4), LY#(4)
   'prep for lat/lon conversion
   CLS
    xmax# = 0!
    ymax# = 0!
    xmin# = 10000000000#
    ymin# = 10000000000#
    LX#(1) = 210848
    LY#(1) = 314418
    LX#(2) = 224231
    LY#(2) = 680193
    LX#(3) = 1290206
    LY#(3) = 684410
    LX#(4) = 1297946
    LY#(4) = 314731
    ax#(1) = 23
    ay#(1) = 42
    ax#(2) = 23
    ay#(2) = 43
    ax#(3) = 27
    ay#(3) = 43
    ax#(4) = 27
    ay#(4) = 42
    rivon = 0
    basinon = 0
    sleerror = 0
    GOSUB 3500
    IF sleerror = 1 THEN STOP
g1% = 0
```

```
BULG$ = "BULGARIA"
OPEN "o", #2, BULG$ + ".POL"
OPEN "o", #3, BULG$ + ".DLG"
OPEN "I", #4, "dxfiles.dat"
WHILE NOT EOF(4)
LINE INPUT #4, a$
PRINT "DXF File Name:"; a$
OPEN "i", #1, a$ + ".DXF"
nriver = 0
'OPEN "o", #2, a$ + ".POL"
'OPEN "o", #3, a$ + ".DLG"
'ignore until section start encountered
1090 : GOSUB 2000
IF g% <> 0 THEN 1090
IF s$ <> "SECTION" THEN 1090
GOSUB 2000
'skip until entities section
IF s$ <> "ENTITIES" THEN 1090
'scan until end of section
1200 : GOSUB 2000
1210 : IF g% = 0 AND s$ = "ENDSEC" THEN 2200
IF g% = 0 AND s$ = "LINE" THEN GOSUB 1400: GOTO 1210
GOTO 1200
1400 : 'accumulate line entity groups
1430 : GOSUB 2000
IF g% = 10 THEN x1 = X: y1 = Y: z1 = z
IF g% = 11 THEN x2 = X: y2 = Y: z2 = z
IF g% = 0 THEN PRINT "Line from ("; x1; ","; y1; ") to ("; x2; ",";
y2; ")": RETURN
GOTO 1430
2000 : 'read group code and following value
       'for x, y and possibly z coords
IF g1% < 0 THEN g% = -g1%: g1% = 0 ELSE INPUT #1, g%
IF g% < 10 OR g% = 999 THEN LINE INPUT #1, s$: RETURN
IF g% >= 38 AND g% <= 49 THEN INPUT #1, V: RETURN
IF g% >= 50 AND g% <= 59 THEN INPUT #1, a: RETURN
IF g% >= 60 AND g% <= 69 THEN INPUT #1, P%: RETURN
IF g% >= 70 AND g% <= 79 THEN INPUT #1, F: RETURN
IF g% >= 210 AND g% <= 219 THEN 2130
IF g% >= 20 THEN PRINT "Invalid group code:"; g%: STOP
2130 : INPUT #1, X
INPUT #1, g1%
IF g1% <> (g% + 10) THEN PRINT "Invalid Y coord code:"; g1%: STOP
INPUT #1, Y
INPUT #1, g1%
IF g1% <> (g% + 20) THEN g1% = -g1% ELSE INPUT #1, z
 IF X = 0 AND Y = 0 AND INSTR(s$, "RIVER") > 0 THEN
```

```
    rivon = 1
    basinon = 0
    nriver = nriver + 1
    PRINT #3, a$;
    PRINT #3, USING "####"; nriver
   ELSEIF X = 0 AND Y = 0 AND INSTR(s$, "BASIN") > 0 THEN
    rivon = 0
    basinon = 1
    PRINT #2, s$
   ELSEIF X = 0 AND Y = 0 THEN
    rivon = 0
    basinon = 0
   ELSEIF rivon = 1 AND X > 0 AND Y > 0 THEN
    NX# = XCOEF#(1) + XCOEF#(2) * X + XCOEF#(3) * Y + XCOEF#(4) *
X * Y
    NY# = YCOEF#(1) + YCOEF#(2) * X + YCOEF#(3) * Y + YCOEF#(4) *
X * Y
    IF NX# > xmax# THEN xmax# = NX#
    IF NX# < xmin# THEN xmin# = NX#
    IF NY# > ymax# THEN ymax# = NY#
    IF NY# < ymin# THEN ymin# = NY#
    PRINT #3, USING "###.####"; NY#; NX#
   ELSEIF basinon = 1 AND X > 0 AND Y > 0 THEN
    NX# = XCOEF#(1) + XCOEF#(2) * X + XCOEF#(3) * Y + XCOEF#(4) *
X * Y
    NY# = YCOEF#(1) + YCOEF#(2) * X + YCOEF#(3) * Y + YCOEF#(4) *
X * Y
    IF NX# > xmax# THEN xmax# = NX#
    IF NX# < xmin# THEN xmin# = NX#
    IF NY# > ymax# THEN ymax# = NY#
    IF NY# < ymin# THEN ymin# = NY#
    PRINT #2, USING "###.####"; NY#; NX#
 END IF
'PRINT #2, s$, x, y
RETURN
2200 CLOSE #1
WEND 'while not eof(4)
CLOSE #2, #3, #4
'pause until key hit
INPUT "Hit any key..."; a$
END

     REM MULTIPLE REGRESSION ROUTINE FOR LAT/LON CONVERSION
3500 n = 4
     M = 4
     FOR i% = 1 TO M
     c#(i%) = 0
```

```
FOR j% = 1 TO M
b#(i%, j%) = 0
NEXT j%
NEXT i%
FOR i% = 1 TO n
XP#(i%, 1) = ay#(i%)
XP#(i%, 2) = LX#(i%)
XP#(i%, 3) = LY#(i%)
XP#(i%, 4) = LX#(i%) * LY#(i%)
NEXT i%
FOR i% = 1 TO n
Y# = XP#(i%, 1)
XP#(i%, 1) = 1
FOR j% = 1 TO M
c#(j%) = c#(j%) + XP#(i%, j%) * Y#
FOR k% = 1 TO M
b#(j%, k%) = b#(j%, k%) + XP#(i%, j%) * XP#(i%, k%)
NEXT k%
NEXT j%
XP#(i%, 1) = Y#
NEXT i%
GOSUB 4000
FOR i% = 1 TO n
YCOEF#(i%) = c#(i%)
NEXT i%
FOR i% = 1 TO n
XP#(i%, 1) = ax#(i%)
NEXT i%
FOR i% = 1 TO M
c#(i%) = 0
FOR j% = 1 TO M
b#(i%, j%) = 0
NEXT j%
NEXT i%
FOR i% = 1 TO n
Y# = XP#(i%, 1)
XP#(i%, 1) = 1
FOR j% = 1 TO M
c#(j%) = c#(j%) + XP#(i%, j%) * Y#
FOR k% = 1 TO M
b#(j%, k%) = b#(j%, k%) + XP#(i%, j%) * XP#(i%, k%)
NEXT k%
NEXT j%
XP#(i%, 1) = Y#
NEXT i%
GOSUB 4000
FOR i% = 1 TO n
```

```
     XCOEF#(i%) = c#(i%)
     NEXT i%
     RETURN
4000 REM SUBROUTINE SLE
     FOR i% = 1 TO n
     DIV# = b#(i%, i%)
     IF ABS(DIV#) < .000001 THEN GOTO 4170
     FOR j% = 1 TO n
     b#(i%, j%) = b#(i%, j%) / DIV#
     NEXT j%
     c#(i%) = c#(i%) / DIV#
     FOR j% = 1 TO n
     IF j% = i% THEN GOTO 4140
     RATIO# = b#(j%, i%)
     FOR k% = 1 TO n
     b#(j%, k%) = b#(j%, k%) - RATIO# * b#(i%, k%)
     NEXT k%
     c#(j%) = c#(j%) - RATIO# * c#(i%)
4140 NEXT j%
     NEXT i%
     RETURN
4170
    LOCATE , 50
     PRINT "SUBROUTINE SLE ENDED"
    LOCATE , 50
     PRINT "BECAUSE DIV#=0"
     sleerror = 1
     INPUT "Hit any key..."; a$
     RETURN
```

2.a SLOVWQ.BAS

```
'slovwq.bas
'reformat slovak water quality data files
CLS
DIM PARMCODE$(11) 'currently loads 11 parameters
DIM PARMCODE(11)
DIM A$(5)
'Hard-coded lists to find and convert parameter values
PARMCODE$(1) = "A1" 'parameter identifier
PARMCODE(1) = 23 'DEMDESS parameter code
PARMCODE$(2) = "A3"
PARMCODE(2) = 1
```

```
PARMCODE$(3) = "A4"
PARMCODE(3) = 2
PARMCODE$(4) = "C3"
PARMCODE(4) = 24
PARMCODE$(5) = "C9"
PARMCODE(5) = 22
PARMCODE$(6) = "B6"
PARMCODE(6) = 3
PARMCODE$(7) = "B7"
PARMCODE(7) = 4
PARMCODE$(8) = "C4"
PARMCODE(8) = 12
PARMCODE$(9) = "G1"
PARMCODE(9) = 5
PARMCODE$(10) = "G2"
PARMCODE(10) = 28
PARMCODE$(11) = "G3"
PARMCODE(11) = 6

OPEN "i", #1, "nitra90.dat" 'input data from WRI
OPEN "o", #2, "nitra90.txt" 'output file of WQ data to be exported
                            'into DEMDESS Table WQDATA using
append-ascii
                            'delimited
OPEN "o", #3, "wqsite90.txt" 'output file of WQ sites to be
exported into
                             'DEMDESS Table WQSITES using append
ascii
                             'delimited
SAMPTIME$ = "" 'Sample time not in file so left blank
LOCCODE$ = "A" '"A"=ambient water quality station
QCCODE$ = "" 'No Quality Control code
VALRMK$ = "" 'No value remark in data base


DO WHILE NOT EOF(1)
  A$(1) = INPUT$(110, 1)
  A$(2) = INPUT$(96, 1)
  A$(3) = INPUT$(96, 1)
  A$(4) = INPUT$(112, 1)
  A$(5) = INPUT$(96, 1)
  LINE INPUT #1, x$ 'read to end of line
  LINE INPUT #1, x$ 'skip 2nd line
  siteid$ = MID$(A$(1), 1, 8)
  SAMPDATE$ = MID$(A$(1), 9, 6)
  WRITE #3, siteid$
  FOR k = 1 TO 5
```

```
   FOR I = 1 TO 11
    IF INSTR(A$(k), PARMCODE$(I)) > 0 THEN
      J = INSTR(A$(k), PARMCODE$(I)) + 2
      VALUE = VAL(MID$(A$(k), J, 14))
      WRITE #2, siteid$, PARMCODE(I), SAMPDATE$, SAMPTIME$,
LOCCODE$, QCCODE$, VALUE, VALRMK$
      PRINT siteid$, PARMCODE(I), VALUE#
    END IF
   NEXT I
  NEXT k

LOOP

CLOSE #1, #2, #3

END
```

2.b DASSIGNS.SC

```
;DASSIGNS.SC
;PROCEDURE TO ASSIGN REACH NUMBERS FROM KM INDEX FILE IN SLOVAKIA
;WRITTEN BY TIM BONDELID FOR WASH/DEMDESP
;           APRIL-MAY 1992
;NOTE: NEED TO CHANGE DBASIN REFERENCES PER RECENT Data base
CHANGES!
@ 7,12
?? "RUNNING THE DEMDESP DISCHARGER ASSIGNMENT TO REACH FILE FOR
SLOVAKIA"
@ 9,12
?? "COPYING AND SORTING DISCH TO TABLE DT"
ARRAY SEGHOLD[6]
ARRAY KMHOLD[6]
SORT "DISCH1" ON "BASIN","DISCHID" TO "DT"
VIEW "SRT"
EDIT "DT"
NRECS = NRECORDS("DT")
HOME
FOR I FROM 1 TO NRECS         ;ASSIGN EACH SITE
  MOVETO RECORD I
  @ 10,12
  ?? "On Record Number ", I
  DLEV = 1
  IF [KM2] > 0
```

```
    THEN
      DLEV = 2
      HOLDKM = [KM2]
  ENDIF
  IF [KM3] > 0
    THEN
      DLEV = 3
      HOLDKM = [KM3]
  ENDIF
  IF [KM4] > 0
    THEN
      DLEV = 4
      HOLDKM = [KM4]
  ENDIF
  IF [KM5]> 0
    THEN
      DLEV = 5
      HOLDKM = [KM5]
  ENDIF
  IF [KM6] > 0
    THEN
      DLEV = 6
      HOLDKM = [KM6]
  ENDIF
  IF DLEV = 1                         ;ON DANUBE MAINSTEM - ALL DONE
    THEN
    [BASIN] = [DBASIN] * 100000
    [SEG] = 1
    [KM] = [KM1]
   ELSE                               ;ON A TRIB - CONTINUE
    [BASIN] = [DBASIN] * 100000 + [KM1]
    IF DLEV = 2                      ;ON SEGMENT 1
      THEN
      [SEG] = 1
      [KM] = [KM2]
     ELSE                             ;ON A TRIB IN BASIN
      SRCHBASIN = [BASIN]
      SRCHSEG = 1
      SRCHKM = [KM2]
      DOWNIMAGE                         ;GO TO REACH FILE
      SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND [KM] =
SRCHKM)
        SRCHSEG = [CSEG]
        SRCHKM = [CKM]
        QUITLOOP
      ENDSCAN
      UPIMAGE                      ;GO TO DISCH TABLE
```

```
IF DLEV = 3
  THEN                                  ;ON A LEV 3 TRIB - DONE
  [DT->SEG] = SRCHSEG
  [DT->KM] = [KM3]
 ELSE                                  ;ON A LEV 4 OR GREATER - CONTINUE
  SRCHKM = [KM3]
  DOWNIMAGE                                 ;GO TO REACH FILE
  SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND [KM]
= SRCHKM)
    SRCHSEG = [CSEG]
    SRCHKM = [CKM]
    QUITLOOP
  ENDSCAN
  UPIMAGE                              ;GO TO DISCH TABLE
  IF DLEV = 4
    THEN                                ;ON A LEV 4 TRIB - DONE
    [DT->SEG] = SRCHSEG
    [DT->KM] = [KM4]
   ELSE                                       ;ON A LEV 5 OR GREATER -
CONTINUE
    SRCHKM = [KM4]
    DOWNIMAGE                               ;GO TO REACH FILE
    SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND
[KM] = SRCHKM)
      SRCHSEG = [CSEG]
      SRCHKM = [CKM]
      QUITLOOP
    ENDSCAN
    UPIMAGE                            ;GO TO DISCH TABLE
    IF DLEV = 5
      THEN                              ;ON A LEV 5 TRIB - DONE
      [DT->SEG] = SRCHSEG
      [DT->KM] = [KM5]
     ELSE                                    ;ON A LEV 6 OR GREATER -
CONTINUE
      SRCHKM = [KM5]
      DOWNIMAGE                             ;GO TO REACH FILE
      SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND
[KM] = SRCHKM)
        SRCHSEG = [CSEG]
        SRCHKM = [CKM]
        QUITLOOP
      ENDSCAN
      UPIMAGE                          ;GO TO DISCH TABLE
      IF DLEV = 6
        THEN                            ;ON A LEV 6 TRIB - DONE
        [DT->SEG] = SRCHSEG
```

```
              [DT->KM] = [KM6]
             ELSE                            ;ON A LEV 7 OR GREATER -
CONTINUE
              SRCHKM = [KM6]
              DOWNIMAGE                          ;GO TO REACH FILE
              SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND
[KM] = SRCHKM)
                SRCHSEG = [CSEG]
                SRCHKM = [CKM]
                QUITLOOP
              ENDSCAN
              UPIMAGE                       ;GO TO DISCH TABLE
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDFOR

DO_IT!
```

2.c DISCHCNV.SC

```
;dischcnv.sc
;converts slovak discharge file to disch1 table and emiss1 table
;written by Tim Bondelid for AID/WASH
;May 1992
;cleanup
clearall
empty "disch1"

view "vstup"
edit "disch1" ;demdess discharger table
upimage ;goto vstup
scan

  downimage ;goto disch1
  [dischid] = [vstup->nec]
  @12,12
  ?? [dischid]
  indchar = substr([dischid],6,1)
  switch
    case indchar="D" : [indcode] = 22
```

```
    case indchar="P" : [indcode] = 100
    case indchar="R" : [indcode] = 100
    case indchar="S" : [indcode] = 100
    case indchar="T" : [indcode] = 100
    case indchar="M" : [indcode] = 3
    case indchar="N" : [indcode] = 3
    case indchar="O" : [indcode] = 3
  endswitch
  pzt1 = [vstup->pzt1] * 1000 / [vstup->pvd] ;BOD from tonnes/yr to
kg/day
  if [indcode] = 22 ;assign municipal emission wqstd
    then
    if pzt1 < 30
      then
      [WQSTD] = "SM1"
    endif
    if pzt1>=30 and pzt1 < 300
      then
      [WQSTD] = "SM2"
    endif
    if pzt1>=300 and pzt1 < 1500
      then
      [WQSTD] = "SM3"
    endif
    if pzt1>=1500 and pzt1 < 6000
      then
      [WQSTD] = "SM4"
    endif
    if pzt1 >=6000
      then
      [WQSTD] = "SM5"
    endif
  endif
  [COUNTRY] = "S"
  [qtot] = [vstup->mvo] * 1000 / [vstup->pvd]
  [daysop] = [vstup->pvd]
  [hrsop] = [vstup->pvh]
  [name] = [vstup->naz_org]
  [tpid] = [dischid]
  [tplev] = [vstup->druh_cist]
  cishyd = [vstup->cis_hyd] ;mahachek reach number
  bas1 = substr(cishyd,1,4)
  ;assign reach numbers
  [km1] = 0
  [km2] = 0
  [km3] = 0
  [km4] = 0
```

```
  [km5] = 0
  [km6] = 0
  [basin] = 100000 + numval(bas1)/100
  [km] = [vstup->rie_km] / 100
  lenhyd = len(cishyd)
  if lenhyd > 20
    then
    hydtext = substr(cishyd,15,5)
    [km4] = numval(hydtext) / 100
  endif
  if lenhyd > 15
    then
    hydtext = substr(cishyd,10,5)
    [km3] = numval(hydtext) / 100
  endif
  if lenhyd > 10
    then
    hydtext = substr(cishyd,6,4)
    [km2] = numval(hydtext) / 100
  endif
  if lenhyd > 5
    then
    hydtext = substr(cishyd,1,4)
    [km1] = numval(hydtext) / 100
  endif

  down ;add a record
  upimage ;goto vstup

 endscan

  do_it!

;add emiss1 records
;cleanup
clearall
empty "emiss1"

view "vstup"
edit "emiss1" ;demdess emisssion table
upimage ;goto vstup

scan ;loop through records
  downimage ;goto emiss1
  [dischid] = [vstup->nec]
  @13,12
  ?? [dischid]
```

```
[parmcode] = 1 ;BOD
[datasrc] = "HYDROM"
[value] = [vstup->pzmg1]
[dischtype] = "I" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] = 1 ;BOD
[datasrc] = "HYDROM"
[value] = [vstup->vzmg1]
[dischtype] = "E" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] =2 ;COD
[datasrc] = "HYDROM"
[value] = [vstup->pzmg2]
[dischtype] = "I" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] = 2 ;COD
[datasrc] = "HYDROM"
[value] = [vstup->vzmg2]
[dischtype] = "E" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] = 3 ;TDS
[datasrc] = "HYDROM"
[value] = [vstup->pzmg3]
[dischtype] = "I" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] = 3 ;TDS
[datasrc] = "HYDROM"
[value] = [vstup->vzmg3]
[dischtype] = "E" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] = 4 ;TSS
[datasrc] = "HYDROM"
[value] = [vstup->pzmg5]
[dischtype] = "I" ;influent
down ;add a record
[dischid] = [vstup->nec]
[parmcode] = 4 ;TSS
[datasrc] = "HYDROM"
[value] = [vstup->vzmg5]
[dischtype] = "E" ;influent
down ;add a record
```

```
  [dischid] = [vstup->nec]
  [parmcode] = 5 ;NH4
  [datasrc] = "HYDROM"
  [value] = [vstup->pzmg6]
  [dischtype] = "I" ;influent
  down ;add a record
  [dischid] = [vstup->nec]
  [parmcode] = 5 ;NH4
  [datasrc] = "HYDROM"
  [value] = [vstup->vzmg6]
  [dischtype] = "E" ;influent
  down ;add a record
  [dischid] = [vstup->nec]
  [parmcode] = 6 ;NO3
  [datasrc] = "HYDROM"
  [value] = [vstup->pzmg7]
  [dischtype] = "I" ;influent
  down ;add a record
  [dischid] = [vstup->nec]
  [parmcode] = 6 ;NO3
  [datasrc] = "HYDROM"
  [value] = [vstup->vzmg7]
  [dischtype] = "E" ;influent
  down ;add a record
  [dischid] = [vstup->nec]
  [parmcode] = 10 ;OIL
  [datasrc] = "HYDROM"
  [value] = [vstup->pzmg8]
  [dischtype] = "I" ;influent
  down ;add a record
  [dischid] = [vstup->nec]
  [parmcode] = 10 ;OIL
  [datasrc] = "HYDROM"
  [value] = [vstup->vzmg8]
  [dischtype] = "E" ;influent
  down ;add a record
  upimage ;goto vstup
endscan
do_it!

;now merge with rchfile to fill in seg field
clearall
play "segmrg"
do_it!
clearall
view "answer"
edit "disch1"
```

```
scan
  @14,12
  ?? [dischid]
  upimage ;goto answer
  scan for [dischid] = [disch1->dischid]
    [disch1->seg] = [seg]
  endscan
  downimage ;goto disch1
endscan

do_it!
```

2.d RFMAKES.SC

```
;RFMAKES.SC
;PROCEDURE TO MAKE REACH FILE FROM KM INDEX FILE IN SLOVAKIA
;WRITTEN BY TIM BONDELID FOR WASH/DEMDESP
;            APRIL 1992
;NOTE: NEED TO TEST WITH RECENT DATA BASE CHANGES!
ARRAY SEGHOLD[7]
ARRAY KMHOLD[7]
@ 3,12
?? "RUNNING THE DEMDESP REACH FILE MAKER FOR SLOVAKIA"
;INPUT FILE IS HSNITRA.TXT - READ IN AS A TABLE
@ 4,12
?? "READING HSNITRA.TXT INTO PARADOX"
{Tools}
{ExportImport}
{Import}
{Ascii}
{Text}
{HSNITRA}
{HSNITRA}
DO_IT!
CLEARIMAGE
@ 5,12
?? "LOADING KM INDEX DATA INTO RCHFILE"
EMPTY "RCHFILE"
DO_IT!
CLEARIMAGE
VIEW "HSNITRA"
EDIT "RCHFILE"
NRECS = NRECORDS("HSNITRA")
FOR I FROM 3 TO NRECS   ;LOOP ON REACHES
 UPIMAGE          ;GO TO HSNITRA TABLE
```

```
MOVETO RECORD I
HQUEST = SUBSTR([TEXT],1,1)
IF HQUEST = "?"           ;IS RECORD TAGGED AS PROBLEM?
 THEN
 @6,12
 ?? "RECORD SKIPPED = ", I
 DOWNIMAGE                ;TO RCHFILE
 ELSE
 HNAME = SUBSTR([TEXT],7,20)
 HTEXT = SUBSTR([TEXT],27,1)
 HLEV = NUMVAL(HTEXT)
 HTEXT = SUBSTR([TEXT],31,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM1 = NUMVAL(HTEXT) / 10
 HTEXT = SUBSTR([TEXT],36,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM2 = NUMVAL(HTEXT) / 10
 HTEXT = SUBSTR([TEXT],41,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM3 = NUMVAL(HTEXT) / 10
 HTEXT = SUBSTR([TEXT],46,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM4 = NUMVAL(HTEXT) / 10
 HTEXT = SUBSTR([TEXT],51,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM5 = NUMVAL(HTEXT) / 10
 HTEXT = SUBSTR([TEXT],56,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM6 = NUMVAL(HTEXT) / 10
 HTEXT = SUBSTR([TEXT],61,4)
 IF HTEXT = "    "
   THEN HTEXT = "0000"
 ENDIF
 HKM7 = NUMVAL(HTEXT) / 10
 IF HLEV = 1
```

```
      THEN
      HSEGL = HKM2
      HKM2 = 0
    ENDIF
    IF HLEV = 2
      THEN
      HSEGL = HKM3
      HKM3 = 0
    ENDIF
    IF HLEV = 3
      THEN
      HSEGL = HKM4
      HKM4 = 0
    ENDIF
    IF HLEV = 4
      THEN
      HSEGL = HKM5
      HKM5 = 0
    ENDIF
    IF HLEV = 5
      THEN
      HSEGL = HKM6
      HKM6 = 0
    ENDIF
    IF HLEV = 6
      THEN
      HSEGL = HKM7
      HKM7 = 0
    ENDIF
    DOWNIMAGE              ;GO TO RCHFILE
     [RCHNAME] = HNAME
     [SEGL] = HSEGL
     [KM1] = HKM1
     [KM2] = HKM2
     [KM3] = HKM3
     [KM4] = HKM4
     [KM5] = HKM5
     [KM6] = HKM6
     [KM7] = HKM7
    DOWN                   ;ADD A NEW RECORD
   ENDIF  ;RECORD SKIP WITH ?
  ENDFOR
  DO_IT!
  CLEARIMAGE
  @ 9,12
  ?? "SORTING RCHFILE"
  SORT "Rchfile" ON "BASIN","KM1","KM2","KM3","KM4","KM5","KM6"
```

```
EDIT "RCHFILE"
NRECS = NRECORDS("RCHFILE")
HOME
FOR I FROM 1 TO NRECS          ;FILL IN VALUES IN EACH ENTERED RECORD
  MOVETO RECORD I
  @ 10,12
  ?? "On Record Number ", I
  [LEV] = 2
  [BASIN] = 1
  IF [KM2] > 0
    THEN
      [LEV] = 3
      HOLDKM = [KM2]
  ENDIF
  IF [KM3] > 0
    THEN
      [LEV] = 4
      HOLDKM = [KM3]
  ENDIF
  IF [KM4] > 0
    THEN
      [LEV] = 5
      HOLDKM = [KM4]
  ENDIF
  IF [KM5]> 0
    THEN
      [LEV] = 6
      HOLDKM = [KM5]
  ENDIF
  IF [KM6] > 0
    THEN
      [LEV] = 7
      HOLDKM = [KM6]
  ENDIF
  HOLDLEV = [LEV]
  [SEG] = I
  SEGHOLD[HOLDLEV] = [SEG]
  [J] = [LEV] - 1
  [KM] = 0
  [BASIN] = [BASIN] * 100000 + [KM1]
  [TYPE] = "S"
  [COUNTRY] = "S"                    ;SLOVAKIA
  SWITCH
    CASE [LEV] = 1 : [BPATTERN] = "|"
    CASE [LEV] = 2 : [BPATTERN] = "|_|"
    CASE [LEV] = 3 : [BPATTERN] = "|¯|_|"
    CASE [LEV] = 4 : [BPATTERN] = "| |¯|_|"
```

```
  CASE [LEV] = 5 : [BPATTERN] = "| | | |_|"
  CASE [LEV] = 6 : [BPATTERN] = "| | | | |_|"
  CASE [LEV] = 7 : [BPATTERN] = "| | | | | |_|"
ENDSWITCH
IF I = 1 THEN
  [DSBASIN] = [BASIN] - [KM1]
  [CBASIN] = [DSBASIN]
  [CSEG] = 1
  [CKM] = [KM1]
  [RCHJUNCDESC] = "DANUBE"
 ELSE
  HOLDSEG = [SEG]
  FINDLEV = [LEV] - 1
  HOLDJDESC1 = [RCHNAME]
  ; GO TO RECORD FOR BREAKIN REACH
  END
  WHILE [SEG] <> SEGHOLD[FINDLEV]
    UP
  ENDWHILE
  [USBASIN2] = [BASIN] ;FILL DATA FOR DOWNSTREAM RCH
  [USSEG2] = HOLDSEG
  [USKM2] = 0
  [USBASIN1] = [BASIN]
  [USSEG1] = [SEG]
  [USKM1] = HOLDKM
  [TYPE] = "R"
  HOLDDSEG = [SEG]
  HOLDDKM = [KM]
  HOLDJDESC2 = [RCHNAME]
  COPYTOARRAY RCHREC   ;MAKE NEW REACH RECORD
  RCHREC["J"] = [LEV]
  RCHREC["CBASIN"] = [BASIN]
  RCHREC["CSEG"] = HOLDSEG
  RCHREC["CKM"] = 0
  RCHREC["KM"] = HOLDKM
  RCHREC["DSBASIN"] = [BASIN]
  RCHREC["DSSEG"] = HOLDDSEG
  RCHREC["DSKM"] = HOLDDKM
  RCHREC["USBASIN1"] = 0
  RCHREC["USSEG1"] = 0
  RCHREC["USKM1"] = 0
  RCHREC["USBASIN2"] = 0
  RCHREC["USSEG2"] = 0
  RCHREC["USKM2"] = 0
  RCHREC["USBASIN3"] = 0
  RCHREC["USSEG3"] = 0
  RCHREC["USKM3"] = 0
```

```
    RCHREC["RCHJUNCDESC"] = HOLDJDESC1
    RCHREC["TYPE"] = "S"
    SWITCH
      CASE RCHREC["LEV"] = 1 : RCHREC["BPATTERN"] = "|"
      CASE RCHREC["LEV"] = 2 : RCHREC["BPATTERN"] = "| |"
      CASE RCHREC["LEV"] = 3 : RCHREC["BPATTERN"] = "| | |"
      CASE RCHREC["LEV"] = 4 : RCHREC["BPATTERN"] = "| | | |"
      CASE RCHREC["LEV"] = 5 : RCHREC["BPATTERN"] = "| | | | |"
      CASE RCHREC["LEV"] = 6 : RCHREC["BPATTERN"] = "| | | | | |"
      CASE RCHREC["LEV"] = 7 : RCHREC["BPATTERN"] = "| | | | | | |"
    ENDSWITCH
    END                          ;ADD NEW REACH RECORD
    DOWN
    COPYFROMARRAY RCHREC

    MOVETO RECORD I              ;GO BACK TO INPUT REACH
    [CBASIN] = [BASIN]
    [CSEG] = HOLDDSEG
    [CKM] = HOLDKM
    [DSBASIN] = [BASIN]
    [DSSEG] = HOLDDSEG
    [DSKM] = HOLDDKM
    [RCHJUNCDESC] = HOLDJDESC2

  ENDIF

ENDFOR

  @ 12,12
  ?? "COMPUTING REACH LENGTHS"
SCAN                             ;UPDATE SEGMENT LENGTHS
  [RCHNO] = [SEG] + [KM]/1000
  IF [USSEG1] = [SEG]
    THEN
      [SEGL] = [USKM1] - [KM]
    ELSE
      [SEGL] = [SEGL] - [KM]
  ENDIF
ENDSCAN

  @ 13,12
  ?? "COMPUTING HYDROLOGIC SEQUENCE NUMBERS"
DO_IT!
CLEARIMAGE
INDEX "RCHFILE" ON "RCHNO"
EDIT "RCHFILE"
MOVETO RECORD 1              ;GO BACK TO FIRST REACH FOR SEQNO
```

```
SEQDONE = “N”
NHOLD = 0
STARTLEV = [LEV]
SEQHOLD = 0
WHILE SEQDONE = “N”
  SEQHOLD = SEQHOLD + 1
  [SEQNO] = SEQHOLD
  @ 14,12
  ?? “SEQNO = ”, SEQHOLD
  IF [USSEG1] > 0
    THEN
    SEGHOLD2 = [USSEG1]
    KMHOLD2 = [USKM1]
    IF [USSEG2] > 0
      THEN
      NHOLD = NHOLD + 1
      SEGHOLD[NHOLD] = [USSEG1]
      KMHOLD[NHOLD] = [USKM1]
      SEGHOLD2 = [USSEG2]
      KMHOLD2 = [USKM2]
    ENDIF
    MOVETO [RCHNO]
    SRCHRCH = SEGHOLD2 + KMHOLD2 / 1000
    LOCATE SRCHRCH
    IF RETVAL                    ;CHECK FOR VALID REACH
      THEN
       ;IT IS OK
      ELSE
      @15,12
      ?? “ERROR IN SRCHRCH ”, SRCHRCH
    ENDIF
    ;SCAN FOR ([SEG] = SEGHOLD2 AND [KM] = KMHOLD2)
    ;  QUITLOOP
    ;ENDSCAN
   ELSE   ;START REACH
     IF NHOLD > 0
       THEN  ;MORE REACHES?
       MOVETO [RCHNO]
       SRCHRCH = SEGHOLD[NHOLD] + KMHOLD[NHOLD] / 1000
       LOCATE SRCHRCH
       IF RETVAL    ;CHECK FOR VALID REACH
         THEN
           ;IT IS OK
         ELSE
         @15,12
         ?? “ERROR IN SRCHRCH ”, SRCHRCH
       ENDIF
```

```
        ;SCAN FOR ([SEG] = SEGHOLD[NHOLD] AND [KM] = KMHOLD[NHOLD])
        ;  QUITLOOP
        ;ENDSCAN
      NHOLD = NHOLD - 1
       ELSE                 ;FINISHED
        SEQDONE = "Y"
      ENDIF
   ENDIF
ENDWHILE
   SCAN                                    ;INVERT SEQNOS
     [SEQNO] = SEQHOLD - [SEQNO] + 1
     IF [TYPE] = "S"                       ;FILL UPSTREAM KM VALUES ON S
RCHS
       THEN
        [USKM1] = [KM] + [SEGL]
     ENDIF
   ENDSCAN

DO_IT!
```

3.a RFMAKER.SC

```
;RFMAKER.SC
;PROCEDURE TO MAKE REACH FILE FROM KM INDEX FILE IN ROMANIA
;WRITTEN BY TIM BONDELID FOR WASH/DEMDESP
;            APRIL 1992
@ 3,12
?? "RUNNING THE DEMDESP REACH FILE MAKER FOR ROMANIA"
ARRAY SEGHOLD[7]     ;SEGMENT HOLD ARRAY
ARRAY KMHOLD[7]      ;KM HOLD ARRAY
ARRAY USKMHOLD[7]    ;UPSTREAM KM HOLD ARRAY FOR ROMANIA

@ 7,12
?? "COPYING VALUES FROM ROMRCHIN TO RCHFILE..."
EMPTY "RCHFILE"           ;START WITH EMPTY RCHFILE
DO_IT!

VIEW "ROMRCHIN"
EDIT "RCHFILE"
UPIMAGE
SCAN                      ;SCAN ROMRCHIN
  DOWNIMAGE               ;GO TO RCHFILE
   [CODUL1] = [ROMRCHIN->INCODUL1]
   [CODUL2] = [ROMRCHIN->INCODUL2]
   [CODUL3] = [ROMRCHIN->INCODUL3]
   [CODUL4] = [ROMRCHIN->INCODUL4]
```

```
    [CODUL5] = [ROMRCHIN->INCODUL5]
    [CODUL6] = [ROMRCHIN->INCODUL6]
    [CODUL7] = [ROMRCHIN->INCODUL7]
    [RCHNAME] = [ROMRCHIN->INRCHNAME]
    [USKM] = [ROMRCHIN->INUSKM]
    [SEGL] = [ROMRCHIN->INSEGL]
    [TOTSEGL] = [ROMRCHIN->INSEGL]
   DOWN                        ;ADD A RECORD
   UPIMAGE                     ;GO TO ROMRCHIN
 ENDSCAN
 DO_IT!
 ?? "DONE."
 CLEARIMAGE

 ;FILL IN CODUL VALUES FOR EACH REACH
 @ 8,12
 ?? "FILLING IN CODUL VALUES..."
 EDIT "RCHFILE"
 NRECS = NRECORDS("RCHFILE")
 FOR I FROM 1 TO 7
   KMHOLD[I] = 0
 ENDFOR
 HOME
 HOLDBASIN = 10                  ;ARGES BASIN IS 10
 SCAN
   [BASIN] = HOLDBASIN
   IF [CODUL6] > 0
     THEN
     DLEV = 7
     ELSE
     IF [CODUL5]> 0
       THEN
       DLEV = 6
       ELSE
       IF [CODUL4] > 0
         THEN
         DLEV = 5
         ELSE
         IF [CODUL3] > 0
           THEN
           DLEV = 4
           ELSE
           IF [CODUL2] > 0
             THEN
             DLEV = 3
             ELSE
             IF [CODUL1] > 0
```

```
            THEN
            DLEV = 2
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
FOR I FROM DLEV TO 7
  KMHOLD[I] = 0
ENDFOR
IF [CODUL1] > 0
  THEN
  KMHOLD[1] = [CODUL1]
  ELSE
  [CODUL1] = KMHOLD[1]
ENDIF
IF [CODUL2] > 0
  THEN
  KMHOLD[2] = [CODUL2]
  ELSE
  [CODUL2] = KMHOLD[2]
ENDIF
IF [CODUL3] > 0
  THEN
  KMHOLD[3] = [CODUL3]
  ELSE
  [CODUL3] = KMHOLD[3]
ENDIF
IF [CODUL4] > 0
  THEN
  KMHOLD[4] = [CODUL4]
  ELSE
  [CODUL4] = KMHOLD[4]
ENDIF
IF [CODUL5] > 0
  THEN
  KMHOLD[5] = [CODUL5]
  ELSE
  [CODUL5] = KMHOLD[5]
ENDIF
IF [CODUL6] > 0
  THEN
  KMHOLD[6] = [CODUL6]
  ELSE
  [CODUL6] = KMHOLD[6]
ENDIF
```

```
  IF [CODUL7] > 0
    THEN
    KMHOLD[7] = [CODUL7]
    ELSE
     [CODUL7] = KMHOLD[7]
  ENDIF
ENDSCAN    ;END SCAN TO FILL CODUL VALUES
;CONVERT UPSTREAM CODULS AND KMs TO DOWNSTREAM KM VALUES
USKMHOLD[1] = 100   ;TEMP VALUE FOR KM POINT ON DANUBE
SCAN
  IF [CODUL7] > 0
    THEN
     [KM1] = KMHOLD[1]
     [KM2] = KMHOLD[2]
     [KM3] = KMHOLD[3]
     [KM4] = KMHOLD[4]
     [KM5] = KMHOLD[5]
     [KM6] = KMHOLD[6]
     [KM7] = USKMHOLD[6] - [USKM]
    ELSE
    IF [CODUL6] > 0
      THEN
       [KM1] = KMHOLD[1]
       [KM2] = KMHOLD[2]
       [KM3] = KMHOLD[3]
       [KM4] = KMHOLD[4]
       [KM5] = KMHOLD[5]
       [KM6] = USKMHOLD[5] - [USKM]
      KMHOLD[6] = [KM6]
      USKMHOLD[6] = [SEGL]
       [KM7] = 0
      ELSE
      IF [CODUL5] > 0
        THEN
         [KM1] = KMHOLD[1]
         [KM2] = KMHOLD[2]
         [KM3] = KMHOLD[3]
         [KM4] = KMHOLD[4]
         [KM5] = USKMHOLD[4] - [USKM]
        KMHOLD[5] = [KM5]
        USKMHOLD[5] = [SEGL]
         [KM6] = 0
         [KM7] = 0
        ELSE
        IF [CODUL4] > 0
          THEN
           [KM1] = KMHOLD[1]
```

```
[KM2] = KMHOLD[2]
[KM3] = KMHOLD[3]
[KM4] = USKMHOLD[3] - [USKM]
KMHOLD[4] = [KM4]
USKMHOLD[4] = [SEGL]
[KM5] = 0
[KM6] = 0
[KM7] = 0
ELSE
IF [CODUL3] > 0
  THEN
  [KM1] = KMHOLD[1]
  [KM2] = KMHOLD[2]
  [KM3] = USKMHOLD[2] - [USKM]
  KMHOLD[3] = [KM3]
  USKMHOLD[3] = [SEGL]
  [KM4] = 0
  [KM5] = 0
  [KM6] = 0
  [KM7] = 0
  [KM6] = 0
  [KM7] = 0
  ELSE
  IF [CODUL2] > 0
    THEN
    [KM1] = KMHOLD[1]
    [KM2] = USKMHOLD[1] - [USKM]
    KMHOLD[2] = [KM2]
    USKMHOLD[2] = [SEGL]
    [KM3] = 0
    [KM4] = 0
    [KM5] = 0
    [KM6] = 0
    [KM7] = 0
    ELSE
    IF [CODUL1] > 0
      THEN
      [KM1] = USKMHOLD[1] - [USKM]
      KMHOLD[1] = [KM1]
      USKMHOLD[1] = [SEGL]
      [KM2] = 0
      [KM3] = 0
      [KM4] = 0
      [KM5] = 0
      [KM6] = 0
      [KM7] = 0
    ENDIF
```

```
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF

ENDSCAN     ;END SCAN TO FILL KM VALUES
DO_IT!
?? "DONE."
CLEARIMAGE
@ 9,12
?? "SORTING RCHFILE..."
SORT "Rchfile" ON "BASIN","KM1","KM2","KM3","KM4","KM5","KM6"
DO_IT!
?? "DONE."
CLEARIMAGE
EDIT "RCHFILE"
NRECS = NRECORDS("RCHFILE")
                            ;BEGIN  MAKING  REACH  FILE  USING
DOWNSTREAM
HOME                        ;KM VALUES
FOR I FROM 1 TO NRECS      ;FILL IN VALUES IN EACH ENTERED RECORD
  MOVETO RECORD I
  @ 10,12
  ?? "On Record Number ", I
  [LEV] = 2
  IF [KM2] > 0
    THEN
      [LEV] = 3
      HOLDKM = [KM2]
  ENDIF
  IF [KM3] > 0
    THEN
      [LEV] = 4
      HOLDKM = [KM3]
  ENDIF
  IF [KM4] > 0
    THEN
      [LEV] = 5
      HOLDKM = [KM4]
  ENDIF
  IF [KM5]> 0
    THEN
      [LEV] = 6
      HOLDKM = [KM5]
  ENDIF
```

```
IF [KM6] > 0
  THEN
    [LEV] = 7
    HOLDKM = [KM6]
ENDIF
HOLDLEV = [LEV]
[SEG] = I
SEGHOLD[HOLDLEV] = [SEG]
[J] = [LEV] - 1
[KM] = 0
[BASIN] = [BASIN] * 100000 + [KM1]
[TYPE] = “S”
[COUNTRY] = “R”
SWITCH
  CASE [LEV] = 1 : [BPATTERN] = “|”
  CASE [LEV] = 2 : [BPATTERN] = “|_|”
  CASE [LEV] = 3 : [BPATTERN] = “|¯|_|”
  CASE [LEV] = 4 : [BPATTERN] = “| |¯|_|”
  CASE [LEV] = 5 : [BPATTERN] = “| | |¯|_|”
  CASE [LEV] = 6 : [BPATTERN] = “| | | |¯|_|”
ENDSWITCH
IF I = 1 THEN
  [DSBASIN] = [BASIN] - [KM1]
  [CBASIN] = [DSBASIN]
  [CSEG] = 1
  [CKM] = [KM1]
  [RCHJUNCDESC] = “DANUBE”
 ELSE
  HOLDSEG = [SEG]
  FINDLEV = [LEV] - 1
  HOLDJDESC1 = [RCHNAME]
  ; GO TO RECORD FOR BREAKIN REACH
  END
  WHILE [SEG] <> SEGHOLD[FINDLEV]
    UP
  ENDWHILE
  [USBASIN2] = [BASIN] ;FILL DATA FOR DOWNSTREAM RCH
  [USSEG2] = HOLDSEG
  [USKM2] = 0
  [USBASIN1] = [BASIN]
  [USSEG1] = [SEG]
  [USKM1] = HOLDKM
  [TYPE] = “R”
  HOLDDSEG = [SEG]
  HOLDDKM = [KM]
  HOLDJDESC2 = [RCHNAME]
  COPYTOARRAY RCHREC   ;MAKE NEW REACH RECORD
```

```
      RCHREC["J"] = [LEV]
      RCHREC["CBASIN"] = [BASIN]
      RCHREC["CSEG"] = HOLDSEG
      RCHREC["CKM"] = 0
      RCHREC["KM"] = HOLDKM
      RCHREC["DSBASIN"] = [BASIN]
      RCHREC["DSSEG"] = HOLDDSEG
      RCHREC["DSKM"] = HOLDDKM
      RCHREC["USBASIN1"] = 0
      RCHREC["USSEG1"] = 0
      RCHREC["USKM1"] = 0
      RCHREC["USBASIN2"] = 0
      RCHREC["USSEG2"] = 0
      RCHREC["USKM2"] = 0
      RCHREC["USBASIN3"] = 0
      RCHREC["USSEG3"] = 0
      RCHREC["USKM3"] = 0
      RCHREC["RCHJUNCDESC"] = HOLDJDESC1
      RCHREC["TYPE"] = "S"
      SWITCH
        CASE RCHREC["LEV"] = 1 : RCHREC["BPATTERN"] = "|"
        CASE RCHREC["LEV"] = 2 : RCHREC["BPATTERN"] = "|  |"
        CASE RCHREC["LEV"] = 3 : RCHREC["BPATTERN"] = "|  |  |"
        CASE RCHREC["LEV"] = 4 : RCHREC["BPATTERN"] = "|  |  |  |"
        CASE RCHREC["LEV"] = 5 : RCHREC["BPATTERN"] = "|  |  |  |  |"
        CASE RCHREC["LEV"] = 6 : RCHREC["BPATTERN"] = "|  |  |  |  |  |"
      ENDSWITCH
      END                            ;ADD NEW REACH RECORD
      DOWN
      COPYFROMARRAY RCHREC

      MOVETO RECORD I                ;GO BACK TO INPUT REACH
      [CBASIN] = [BASIN]
      [CSEG] = HOLDDSEG
      [CKM] = HOLDKM
      [DSBASIN] = [BASIN]
      [DSSEG] = HOLDDSEG
      [DSKM] = HOLDDKM
      [RCHJUNCDESC] = HOLDJDESC2

    ENDIF

  ENDFOR

    @ 12,12
    ?? "COMPUTING REACH LENGTHS..."
  SCAN                               ;UPDATE SEGMENT LENGTHS
```

```
  IF [USSEG1] = [SEG]
    THEN
      [SEGL] = [USKM1] - [KM]
    ELSE
      [SEGL] = [SEGL] - [KM]
  ENDIF
ENDSCAN
?? "DONE."

  @ 13,12
  ?? "COMPUTING HYDROLOGIC SEQUENCE NUMBERS..."
MOVETO RECORD 1                ;GO BACK TO FIRST REACH FOR SEQNO
SEQDONE = "N"
NHOLD = 0
STARTLEV = [LEV]
SEQHOLD = 0
WHILE SEQDONE = "N"
  @ 14,12
  ?? SEQHOLD
  SEQHOLD = SEQHOLD + 1
  [SEQNO] = SEQHOLD
  IF [USSEG1] > 0
    THEN
    SEGHOLD2 = [USSEG1]
    KMHOLD2 = [USKM1]
    IF [USSEG2] > 0
      THEN
      NHOLD = NHOLD + 1
      SEGHOLD[NHOLD] = [USSEG1]
      KMHOLD[NHOLD] = [USKM1]
      SEGHOLD2 = [USSEG2]
      KMHOLD2 = [USKM2]
    ENDIF
    SCAN FOR ([SEG] = SEGHOLD2 AND [KM] = KMHOLD2)
      QUITLOOP
    ENDSCAN
   ELSE   ;START REACH
     IF NHOLD > 0
       THEN  ;MORE REACHES?
       SCAN FOR ([SEG] = SEGHOLD[NHOLD] AND [KM] = KMHOLD[NHOLD])
         QUITLOOP
       ENDSCAN
     NHOLD = NHOLD - 1
      ELSE             ;FINISHED
       SEQDONE = "Y"
     ENDIF
  ENDIF
```

```
ENDWHILE
  SCAN                              ;INVERT SEQNOS
    [SEQNO] = SEQHOLD - [SEQNO] + 1
    IF [TYPE] = "S"                 ;FILL UPSTREAM KM VALUES ON S
RCHS
      THEN
      [USKM1] = [KM] + [SEGL]
    ENDIF
  ENDSCAN
@15,12
?? "DONE."


DO_IT!
```

3.b DASSIGNR.SC

```
;DASSIGNR.SC
;ASSIGN REACH NUMBERS TO DISCH SITES USING RCHFILE AND SRCHCODUL
;WRITTEN BY TIM BONDELID FOR AID/WASH DEMDESP
;           APRIL 1992
;
;MAKE TEXT VERSION OF ROMANIAN CODUL NUMBERS IN RCHFILE
;WRITTEN BY TIM BONDELID FOR AID/WASH DEMDESP PROJECT
;           APRIL 1992
CLEARALL
EDIT "RCHFILE"
SCAN
  IF [CODUL7] > 0
    THEN
    [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " + STRVAL([CODUL2])
+ " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4]) + " " +
STRVAL([CODUL5]) + " " + STRVAL([CODUL6]) + " " + STRVAL([CODUL7])
    ELSE
    IF [CODUL6] > 0
      THEN
      [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4])
+ " " + STRVAL([CODUL5]) + " " + STRVAL([CODUL6])
      ELSE
      IF [CODUL5] > 0
        THEN
        [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4])
+ " " + STRVAL([CODUL5])
        ELSE
```

```
        IF [CODUL4] > 0
          THEN
          [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4])
          ELSE
          IF [CODUL3] > 0
            THEN
            [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3])
            ELSE
            IF [CODUL2] > 0
              THEN
              [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2])
              ELSE
              IF [CODUL1] > 0
                THEN
                [SRCHCODUL] = "A " + STRVAL([CODUL1])
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDSCAN

DO_IT!
;ASSIGN REACH NUMBERS TO DISCH SITES USING RCHFILE AND SRCHCODUL
CLEARALL
VIEW "RCHFILE"
EDIT "ARGES"
SCAN           ;GO THROUGH ARGES TABLE ONE RECORD AT A TIME
  SRCHTXT = [ROM-BIBLE-NO]
  UPIMAGE    ;GO TO RCHFILE
  MOVETO [SRCHCODUL]
  LOCATE SRCHTXT
  [ARGES->RCHBASIN] = [BASIN]
  [ARGES->RCHSEG] = [SEG]
  [ARGES->RCHKM] = [TOTSEGL] - [ARGES->ROM-BIBLE-KM] /10
  DOWNIMAGE  ;GO TO ARGES TABLE

ENDSCAN

DO_IT!
```

3.c TXTCODUL.SC

```
;TXTCODL.SC
;MAKE TEXT VERSION OF ROMANIAN CODUL NUMBERS IN RCHFILE
;WRITTEN BY TIM BONDELID FOR AID/WASH DEMDESP PROJECT
;             APRIL 1992
EDIT "RCHFILE"
SCAN
  IF [CODUL7] > 0
    THEN
    [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " + STRVAL([CODUL2])
+ " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4]) + " " +
STRVAL([CODUL5]) + " " + STRVAL([CODUL6]) + " " + STRVAL([CODUL7])
    ELSE
    IF [CODUL6] > 0
      THEN
      [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4])
+ " " + STRVAL([CODUL5]) + " " + STRVAL([CODUL6])
      ELSE
      IF [CODUL5] > 0
        THEN
        [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4])
+ " " + STRVAL([CODUL5])
        ELSE
        IF [CODUL4] > 0
          THEN
          [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3]) + " " + STRVAL([CODUL4])
          ELSE
          IF [CODUL3] > 0
            THEN
            [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2]) + " " + STRVAL([CODUL3])
            ELSE
            IF [CODUL2] > 0
              THEN
              [SRCHCODUL] = "A " + STRVAL([CODUL1]) + " " +
STRVAL([CODUL2])
              ELSE
              IF [CODUL1] > 0
                THEN
                [SRCHCODUL] = "A " + STRVAL([CODUL1])
              ENDIF
            ENDIF
          ENDIF
```

```
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDSCAN

DO_IT!
```

4.a RFMAKEH.SC

```
;RFMAKEH.SC
;PROCEDURE TO MAKE REACH FILE FROM KM INDEX FILE IN HUNGARY
;WRITTEN BY TIM BONDELID FOR WASH/DEMDESP
;           APRIL 1992
@ 7,12
?? "RUNNING THE DEMDESP REACH FILE MAKER FOR HUNGARY"
@ 9,12
?? "COPYING AND SORTING RCHFILE TO TABLE SRT"
ARRAY SEGHOLD[6]
ARRAY KMHOLD[6]
SORT "Rchfile" ON "BASIN","KM1","KM2","KM3","KM4","KM5","KM6" TO
"SRT"
EDIT "SRT"
NRECS = NRECORDS("SRT")
HOME
FOR I FROM 1 TO NRECS        ;FILL IN VALUES IN EACH ENTERED RECORD
  MOVETO RECORD I
  @ 10,12
  ?? "On Record Number ", I
  [LEV] = 2
  IF [KM2] > 0
    THEN
      [LEV] = 3
      HOLDKM = [KM2]
  ENDIF
  IF [KM3] > 0
    THEN
      [LEV] = 4
      HOLDKM = [KM3]
  ENDIF
  IF [KM4] > 0
    THEN
      [LEV] = 5
      HOLDKM = [KM4]
  ENDIF
  IF [KM5]> 0
```

```
  THEN
    [LEV] = 6
    HOLDKM = [KM5]
ENDIF
IF [KM6] > 0
  THEN
    [LEV] = 7
    HOLDKM = [KM6]
ENDIF
HOLDLEV = [LEV]
[SEG] = I
SEGHOLD[HOLDLEV] = [SEG]
[J] = [LEV] - 1
[KM] = 0
[BASIN] = [BASIN] * 100000 + [KM1]
[TYPE] = "S"
[COUNTRY] = "H"
SWITCH
  CASE [LEV] = 1 : [BPATTERN] = "|"
  CASE [LEV] = 2 : [BPATTERN] = "|_|"
  CASE [LEV] = 3 : [BPATTERN] = "| |_|"
  CASE [LEV] = 4 : [BPATTERN] = "| | |_|"
  CASE [LEV] = 5 : [BPATTERN] = "| | | |_|"
  CASE [LEV] = 6 : [BPATTERN] = "| | | | |_|"
ENDSWITCH
IF I = 1 THEN
   [DSBASIN] = [BASIN] - [KM1]
   [CBASIN] = [DSBASIN]
   [CSEG] = 1
   [CKM] = [KM1]
   [RCHJUNCDESC] = "DANUBE"
 ELSE
  HOLDSEG = [SEG]
  FINDLEV = [LEV] - 1
  HOLDJDESC1 = [RCHNAME]
  ; GO TO RECORD FOR BREAKIN REACH
  END
  WHILE [SEG] <> SEGHOLD[FINDLEV]
    UP
  ENDWHILE
   [USBASIN2] = [BASIN] ;FILL DATA FOR DOWNSTREAM RCH
   [USSEG2] = HOLDSEG
   [USKM2] = 0
   [USBASIN1] = [BASIN]
   [USSEG1] = [SEG]
   [USKM1] = HOLDKM
   [TYPE] = "R"
```

```
    HOLDDSEG = [SEG]
    HOLDDKM = [KM]
    HOLDJDESC2 = [RCHNAME]
    COPYTOARRAY RCHREC    ;MAKE NEW REACH RECORD
    RCHREC["J"] = [LEV]
    RCHREC["CBASIN"] = [BASIN]
    RCHREC["CSEG"] = HOLDSEG
    RCHREC["CKM"] = 0
    RCHREC["KM"] = HOLDKM
    RCHREC["DSBASIN"] = [BASIN]
    RCHREC["DSSEG"] = HOLDDSEG
    RCHREC["DSKM"] = HOLDDKM
    RCHREC["USBASIN1"] = 0
    RCHREC["USSEG1"] = 0
    RCHREC["USKM1"] = 0
    RCHREC["USBASIN2"] = 0
    RCHREC["USSEG2"] = 0
    RCHREC["USKM2"] = 0
    RCHREC["USBASIN3"] = 0
    RCHREC["USSEG3"] = 0
    RCHREC["USKM3"] = 0
    RCHREC["RCHJUNCDESC"] = HOLDJDESC1
    RCHREC["TYPE"] = "S"
    SWITCH
      CASE RCHREC["LEV"] = 1 : RCHREC["BPATTERN"] = "|"
      CASE RCHREC["LEV"] = 2 : RCHREC["BPATTERN"] = "|   |"
      CASE RCHREC["LEV"] = 3 : RCHREC["BPATTERN"] = "|   |   |"
      CASE RCHREC["LEV"] = 4 : RCHREC["BPATTERN"] = "|   |   |   |"
      CASE RCHREC["LEV"] = 5 : RCHREC["BPATTERN"] = "|   |   |   |   |"
      CASE RCHREC["LEV"] = 6 : RCHREC["BPATTERN"] = "|   |   |   |   |   |"
    ENDSWITCH
    END                           ;ADD NEW REACH RECORD
    DOWN
    COPYFROMARRAY RCHREC

    MOVETO RECORD I               ;GO BACK TO INPUT REACH
    [CBASIN] = [BASIN]
    [CSEG] = HOLDDSEG
    [CKM] = HOLDKM
    [DSBASIN] = [BASIN]
    [DSSEG] = HOLDDSEG
    [DSKM] = HOLDDKM
    [RCHJUNCDESC] = HOLDJDESC2

  ENDIF

ENDFOR
```

```
  @ 12,12
  ?? "COMPUTING REACH LENGTHS"
SCAN                            ;UPDATE SEGMENT LENGTHS
  IF [USSEG1] = [SEG]
    THEN
      [SEGL] = [USKM1] - [KM]
    ELSE
      [SEGL] = [SEGL] - [KM]
  ENDIF
ENDSCAN

  @ 13,12
  ?? "COMPUTING HYDROLOGIC SEQUENCE NUMBERS"
MOVETO RECORD 1              ;GO BACK TO FIRST REACH FOR SEQNO
SEQDONE = "N"
NHOLD = 0
STARTLEV = [LEV]
SEQHOLD = 0
WHILE SEQDONE = "N"
  SEQHOLD = SEQHOLD + 1
  [SEQNO] = SEQHOLD
  IF [USSEG1] > 0
    THEN
    SEGHOLD2 = [USSEG1]
    KMHOLD2 = [USKM1]
    IF [USSEG2] > 0
      THEN
      NHOLD = NHOLD + 1
      SEGHOLD[NHOLD] = [USSEG1]
      KMHOLD[NHOLD] = [USKM1]
      SEGHOLD2 = [USSEG2]
      KMHOLD2 = [USKM2]
    ENDIF
    SCAN FOR ([SEG] = SEGHOLD2 AND [KM] = KMHOLD2)
      QUITLOOP
    ENDSCAN
   ELSE   ;START REACH
     IF NHOLD > 0
       THEN  ;MORE REACHES?
       SCAN FOR ([SEG] = SEGHOLD[NHOLD] AND [KM] = KMHOLD[NHOLD])
         QUITLOOP
       ENDSCAN
     NHOLD = NHOLD - 1
      ELSE               ;FINISHED
       SEQDONE = "Y"
     ENDIF
  ENDIF
```

```
ENDWHILE
  SCAN                                          ;INVERT SEQNOS
    [SEQNO] = SEQHOLD - [SEQNO] + 1
    IF [TYPE] = "S"                             ;FILL UPSTREAM KM VALUES ON S
RCHS
      THEN
       [USKM1] = [KM] + [SEGL]
    ENDIF
  ENDSCAN


DO_IT!
```

4.b DASSIGNH.SC

```
;DASSIGNH.SC
;PROCEDURE TO ASSIGN REACH NUMBERS FROM KM INDEX FILE IN HUNGARY
;WRITTEN BY TIM BONDELID FOR WASH/DEMDESP
;           APRIL 1992
@ 7,12
?? "RUNNING THE DEMDESP DISCHARGER ASSIGNMENT TO REACH FILE FOR
HUNGARY"
@ 9,12
?? "COPYING AND SORTING DISCH TO TABLE DT"
ARRAY SEGHOLD[6]
ARRAY KMHOLD[6]
SORT "DISCH" ON "DBASIN","DISCHID" TO "DT"
VIEW "SRT"
EDIT "DT"
NRECS = NRECORDS("DT")
HOME
FOR I FROM 1 TO NRECS            ;ASSIGN EACH SITE
  MOVETO RECORD I
  @ 10,12
  ?? "On Record Number ", I
  DLEV = 1
  IF [KM2] > 0
    THEN
      DLEV = 2
      HOLDKM = [KM2]
  ENDIF
  IF [KM3] > 0
    THEN
      DLEV = 3
      HOLDKM = [KM3]
  ENDIF
```

```
  IF [KM4] > 0
    THEN
      DLEV = 4
      HOLDKM = [KM4]
  ENDIF
  IF [KM5]> 0
    THEN
      DLEV = 5
      HOLDKM = [KM5]
  ENDIF
  IF [KM6] > 0
    THEN
      DLEV = 6
      HOLDKM = [KM6]
  ENDIF
  IF DLEV = 1                       ;ON DANUBE MAINSTEM - ALL DONE
    THEN
    [BASIN] = [DBASIN] * 100000
    [SEG] = 1
    [KM] = [KM1]
   ELSE                             ;ON A TRIB - CONTINUE
    [BASIN] = [DBASIN] * 100000 + [KM1]
    IF DLEV = 2                    ;ON SEGMENT 1
      THEN
       [SEG] = 1
       [KM] = [KM2]
     ELSE                           ;ON A TRIB IN BASIN
      SRCHBASIN = [BASIN]
      SRCHSEG = 1
      SRCHKM = [KM2]
      DOWNIMAGE                       ;GO TO REACH FILE
      SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND [KM] =
SRCHKM)
        SRCHSEG = [CSEG]
        SRCHKM = [CKM]
        QUITLOOP
      ENDSCAN
      UPIMAGE                    ;GO TO DISCH TABLE
      IF DLEV = 3
        THEN                        ;ON A LEV 3 TRIB - DONE
         [DT->SEG] = SRCHSEG
         [DT->KM] = [KM3]
       ELSE                        ;ON A LEV 4 OR GREATER - CONTINUE
        SRCHKM = [KM3]
        DOWNIMAGE                       ;GO TO REACH FILE
        SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND [KM]
= SRCHKM)
```

```
          SRCHSEG = [CSEG]
          SRCHKM = [CKM]
          QUITLOOP
        ENDSCAN
        UPIMAGE                    ;GO TO DISCH TABLE
        IF DLEV = 4
          THEN                         ;ON A LEV 4 TRIB - DONE
          [DT->SEG] = SRCHSEG
          [DT->KM] = [KM4]
         ELSE                              ;ON A LEV 5 OR GREATER -
CONTINUE
          SRCHKM = [KM4]
          DOWNIMAGE                        ;GO TO REACH FILE
          SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND
[KM] = SRCHKM)
            SRCHSEG = [CSEG]
            SRCHKM = [CKM]
            QUITLOOP
          ENDSCAN
          UPIMAGE                    ;GO TO DISCH TABLE
          IF DLEV = 5
            THEN                         ;ON A LEV 5 TRIB - DONE
            [DT->SEG] = SRCHSEG
            [DT->KM] = [KM5]
           ELSE                            ;ON A LEV 6 OR GREATER -
CONTINUE
            SRCHKM = [KM5]
            DOWNIMAGE                      ;GO TO REACH FILE
            SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND
[KM] = SRCHKM)
              SRCHSEG = [CSEG]
              SRCHKM = [CKM]
              QUITLOOP
            ENDSCAN
            UPIMAGE                  ;GO TO DISCH TABLE
            IF DLEV = 6
              THEN                       ;ON A LEV 6 TRIB - DONE
              [DT->SEG] = SRCHSEG
              [DT->KM] = [KM6]
             ELSE                          ;ON A LEV 7 OR GREATER -
CONTINUE
              SRCHKM = [KM6]
              DOWNIMAGE                    ;GO TO REACH FILE
              SCAN FOR ([BASIN] = SRCHBASIN AND [SEG] = SRCHSEG AND
[KM] = SRCHKM)
                SRCHSEG = [CSEG]
                SRCHKM = [CKM]
```

```
            QUITLOOP
          ENDSCAN
          UPIMAGE                 ;GO TO DISCH TABLE
        ENDIF
      ENDIF
    ENDIF
  ENDIF
 ENDIF
ENDIF
ENDFOR

DO_IT!
```

4.c HWQDATA.SC

```
;HWQDATA.SC
;PARADOX PROGRAM TO CONVERT VITUKI WATER QUALITY DATA
;TO DEMDESP DATA STRUCTURE
;WRITTEN BY TIM BONDELID FOR AID/WASH  APRIL 1992
@5,12
?? "DEMDESP CONVERT FOR HUNGARY WATER QUALITY DATA"
@8,12
?? "Converting 01FF61 to Paradox..."
{TOOLS}
{ExportImport}
{Import}
{Dbase}
{01FF61}
{01FF61}
DO_IT!
VIEW "01FF61"   ;OPEN WQ DATA TABLE CONVERTED FROM DBASE
STDATE = SUBSTR([Md],1,6)   ;START DATE FOR WQSITES TABLE
STAID  = [Tr]               ;STA ID FOR WQSITES TABLE
EDIT "WQDATA"   ; OPEN DEMDESP WQ DATA TABLE
IF NRECORDS("WQDATA") > 1
  THEN
  END              ;GO TO BOTTOM OF WQDATA
  DOWN             ;ADD A RECORD
ENDIF
@10,12
?? "CONVERTING TO DEMDESP WQDATA TABLE..."
UPIMAGE          ;GO TO 01FF61
NRECS = 0        ;COUNTER OF RECORDS IN 01FF61
NADDED = 0       ;COUNTER OF ADDED RECORDS TO WQDATA
```

```
@11,12
?? NRECS
@12,12
?? NADDED
SCAN FOR [Md] > "900000"        ;LOOP THROUGH 01FF61 RECORDS ONE AT
A TIME
                                ;LOAD ONLY 1990 AND LATER
  NRECS = NRECS + 1
  @11,12
  ?? NRECS
  SITEID = [Tr]
  SDATE = SUBSTR([Md],1,6)
  STIME = SUBSTR([Md],7,4)
  IF [A25] < 9999
    THEN
    NADDED = NADDED + 1
    @12,12
    ?? NADDED
    VAL = [A25] / 100
    DOWNIMAGE    ;GO TO WQDATA
     [WQSITEID] = SITEID
     [PARMCODE] = 24                  ;pH
     [SAMPDATE] = SDATE
     [SAMPTIME] = STIME
     [LOCCODE] = "A"
     [QCCODE] = " "
     [VALUE] = VAL
     [VAL-RMK] = " "
    DOWN         ;ADD A NEW RECORD
    UPIMAGE
  ENDIF
  IF [A22] < 9999
    THEN
    NADDED = NADDED + 1
    @12,12
    ?? NADDED
    VAL = [A22] / 100
    DOWNIMAGE    ;GO TO WQDATA
     [WQSITEID] = SITEID
     [PARMCODE] = 1              ;BOD-5
     [SAMPDATE] = SDATE
     [SAMPTIME] = STIME
     [LOCCODE] = "A"
     [QCCODE] = " "
     [VALUE] = VAL
     [VAL-RMK] = " "
    DOWN         ;ADD A NEW RECORD
```

```
    UPIMAGE
  ENDIF
  IF [A38] < 9999
    THEN
    NADDED = NADDED + 1
    @12,12
    ?? NADDED
    VAL = [A38] / 100
    DOWNIMAGE   ;GO TO WQDATA
     [WQSITEID] = SITEID
     [PARMCODE] = 5              ;AMMONIA
     [SAMPDATE] = SDATE
     [SAMPTIME] = STIME
     [LOCCODE] = "A"
     [QCCODE] = " "
     [VALUE] = VAL
     [VAL-RMK] = " "
    DOWN        ;ADD A NEW RECORD
    UPIMAGE
  ENDIF
ENDSCAN

EDATE = SUBSTR([Md],1,6)  ;END DATE FOR WQSITES TABLE

DO_IT!
CLEARIMAGE
DO_IT!
DELETE "01FF61"    ;DELETE PARADOX VERSION OF TABLE
DO_IT!
EDIT "WQSITES"
SCAN FOR [WQSITEID] = STAID  ;ADD START AND END DATES TO WQSITES
  [STARTDATE] = STDATE
  [ENDDATE] = EDATE
  QUITLOOP
ENDSCAN
CLEARIMAGE

DO_IT!
```

Appendix F

DOCUMENTATION OF DEMDESS INITIAL SYSTEM

F.1 Introduction

This appendix lists the software developed for the DEMDESS initial system. Its purpose is to serve as a reference tool for users as they learn to use DEMDESS.

F.2 Software Scripts and Output Files

A script developed for a Paradox® application is generally small, and can be used within Paradox® either as a stand-alone procedure or nested within other procedures. Scripts can be viewed on the screen using the SCRIPT EDIT command under Paradox®. For scripts that form queries, the SCRIPT PLAY command is more useful, both for viewing and for editing the selection of records and the specification of any computations to be performed.

In Table F-1 the scripts that carry out the computations and produce the output tables for the Initial System are described briefly. The scripts are listed in the order in which they are used. The asterisks (*) preceding the name of each script indicate the level of nesting: "Jantra" is the master script that calls or "plays" the first-level scripts "Trtplcy," "Plots," and "Plotout," which are marked with one asterisk. Second-level and third-level scripts are marked with two and three asterisks, respectively.

Table F-2 contains a brief description of the output tables and files that make up the Initial System. All Paradox® application files follow the normal MS-DOS conventions for file names, in the form:

Filename.ext

The file extension indicates the contents of the file:

Filename.DB is a Paradox® table or data base.

Filename.G is a Paradox® graph settings file.

Filename.R, Filename.R1, Filename.R2, or Filename.R3 contain the format for a Paradox® report.

Note that other file extensions are also used by Paradox®, but not in Table F-2.

Included in table F-2 are various data base files and reports used in preparing this user manual. The names of these files generally end in the letters "DF" for "data definition"; for example, "Wqsitedf" describes the fields in the data base "Wqsites." These files are intended to form the basis of a future on-line documentation system, and are a starting point for translating the definitions of the data into the languages of the four participating countries.

F.3 Listings of Software Scripts

Table F-3 lists the software scripts in the same sequence in which they are used for computations in the Initial System, which is also the same sequence as given in Table F-1.

Table F-1 Software Scripts

Script Name	Description
Jantra	Master script; plays three stand-alone scripts (Trtplcy, Plots, and Plotout) in sequence.
* Trtplcy	Controls the computations to produce output tables TRTPLCY1,..,TRTPLCY5.
** Trtplcy1	Joins tables EMISS1, DISCH1, TPCHR, WQSTDS & FINES and computes fines+emissions for treatment options.
** Trtplcy1B	Creates table TRTPLCY1, eliminates negative fines, changes TAXID for options that include municipal wastewater treatment.
** Trtplcy2	Joins tables TRTPLCY1, TPCST, DISHC1, TAXES, CURRCNVT; computes effluent load, load removed, and tax for treatment options.
** Trtplcy3	Fills table TRTPLCY3 from TRTPLCY2, computes costs, amortization, subsidies; marks minimum-cost option for each municipality.
** Trtplcy3A	Joins tables TRTPLCY3, TRTPLCY1 to compute Leva/kgBOD$_5$ removed; saves result in TRTPLCY3.
** Trtplcy4	Queries table TRTPLCY3 to find sums of fines, taxes, and costs (capital, annual capital, subsidy, O&M, net to municipality), and population served.
** Trtplcy5	Queries tables TRTPLCY1 & DISCH1, forms table TRTPLCY5 containing cumulative population, BOD load and flow.
* Plots	Creates tables PLOTn for plots 1,...,11 that can be displayed later using the script Plotout.
** Plot1A	Queries table TRTPLCY4 to form the output table PLOT1, ready for plotting National-policy Options.
** Plot1B	Rearranges the fields in PLOT1 table, as required by Paradox® to produce the graph called Plot1.
** Plot2A	Queries table TRTPLCY4 to form the output table PLOT2 (municipal-level costs of treatment options).
** Plot2BX	Prepares PLOT2 table for use in a Paradox® graph.
** Plots345	Queries TRTPLCY4 table to abstract data for Plots 3, 4, and 5.

Table F-1. Software Scripts (cont'd)

Script Name	Description
** Plts345A	Forms the PLOTS345 table from the query in Script Plots345.
** Plot3A	Abstracts the table PLOT3 from PLOTS345, to get unit costs of BOD removal for the treatment options.
** Plot4A	Abstracts the table PLOT4 from PLOTS345, to get annual cost per person at the municipal level.
** Plot5A	Abstracts the table PLOT5 from PLOTS345, to get cost/cu m for treatment options.
** Plot6	Queries TRTPLCY5 table to form PLOT6 (cumulative population, raw BOD load, and wastewater flow.
** Plot7A	Queries TRTPLCY3 table to abstract the minimum-cost option for each municipality.
** Plot7B	Forms table PLOT7BAS to include "Town" field, sorts records on Population field.
** Plot7C	Queries PLOT7BAS table, computes total amortized costs for treatment options.
** Plot7D	Forms table PLOT7BA4 rearranging the sequence of the fields, ready for plotting cumulative values.
** Plot7E	Computes cumulative values (population, BOD removed, cost) for minimum-cost municipal treatment options.
** Rchrout	Master script for computing existing BOD profiles (cumulative BOD emissions + decayed BOD load in the river). Forms tables MODLLOD1, RCHROUT, RCHGRAF.
*** Fillmlod	Queries RCHFILE, DISCH1, and TRTPLCY1 to abstract dischargers, loads, and flows by river reach.
*** Fillmlo2	Queries MODLLOD1 to change blank PARMCODE to PARMCODE=1 (BOD_5).
*** Fillmrou	Queries RCHFILE, MODLRCH, MODLLOD1, and MODLCOEF to form initialized table RCHROUT ready for BOD modeling.
** Plot8A	Queries RCHGRAF to form the table PLOT8 (Existing profiles of cum.BOD_5 emissions and decayed BOD_5 load along the main stem of the river).
** Rchrout2	Master script for computing "Minimum-Cost" BOD profiles; forms tables MODLLOD2, RCHROUT2, and RCHGRAF2.

Table F-1. Software Scripts (cont'd)

Script Name	Description
*** Fillmllc	Queries RCHFILE, DISCH1, TRTPLCY1, TRTPLCY3 to get data for the minimum-cost option at each municipality.
*** Fillml12	Queries MODLLOD2 to change blank PARMCODEs to PARMCODE=1 (BOD5).
*** Fillmrlc	Queries RCHFILE, MODLRCH, MODLLOD2, and MODLCOEF to obtain initialized Table RCHROUT2 ready for BOD routing in all rivers of the basin.
** Plot9A	Queries RCHGRAF2 to obtain data for PLOT9 (Min.-Cost BOD profiles - cumulative emissions + decayed load - along the main stem of the river).
** Plot10A	Queries WQDATA, WQSITES, RCHFILE, and WQSTDS to form PLOT10 (river BOD profile from measurements at monitoring sites, + BOD standards along the river).
** Plot10B	Prepares PLOT10 table for producing a Paradox® graph.
** Plot11A	Queries WQDATA, WQSITES, RCHFILE, and WQSTDS to obtain a time series of COD measurements at one monitoring station, stored in PLOT11 table.
** Plot11BX	Prepares PLOT11 table for producing a Paradox® graph.
* Plotout	Master script for a user interface to review the analysis of treatment options (Plots 1 - 11 and associated tables) and output to screen or printer.

Table F-2. DEMDESS Output Tables and Files

Table	.Ext	Description
Dischxdf	DB	Data for TABLE 4.
Emissxdf	DB	Data for TABLE 5.
Mdlcofdf	DB	Data for TABLE 15.
Mdlloddf	DB	Data for TABLE 16.
Mdlrchdf	DB	Data for TABLE 17.
Orgdess	DB	Data for TABLE 1.
Outtabls	DB	Data for this table, TABLE F-2.
Permitdf	DB	Data for TABLE 11.
Plot1	DB	Data for Figure 5.
Plot10	DB	Data for Figure 1.
Plot11	DB	Data for Figure 11.
Plot2	DB	Data for Figure 6.
Plot3	DB	Data for Figure 4.
Plot4	DB	Data for Figure 7.
Plot5	DB	Data for Figure 3.
Plot6	DB	Data for Figure 2.
Plot7BA4	DB	Data for Figure 8.
Plot7BAS	DB	Intermediate results.
Plot8	DB	Data for Figure 9.
Plot9	DB	Data for Figure 10.
Plots345	DB	Intermediate results.
Plts345A	DB	Intermediate results.
Scripts	DB	Data for Table F-1.
Trtplcy1	DB	Detailed information on treatment options
Trtplcy2	DB	"
Trtplcy3	DB	"
Trtplcy4	DB	"
Trtplcy5	DB	"
Trtplcy6	DB	"
Wqdatadf	DB	Data for TABLE 18.
Wqsitedf	DB	Data for TABLE 19.
Plot1	G	Graph settings for Plot 1.
Plot10	G	Graph settings for Plot 10.
Plot11	G	Graph settings for Plot 11.
Plot2	G	Graph settings for Plot 2.
Plot3	G	Graph settings for Plot 3.
Plot4	G	Graph settings for Plot 4.
Plot5	G	Graph settings for Plot 5.
Plot6	G	Graph settings for Plot 6.
Plot7	G	Graph settings for Plot 7.
Plot8	G	Graph settings for Plot 8.
Plot9	G	Graph settings for Plot 9.
Currcnvt	R	Default report format for Table CURRCNVT

Disch1	R	Default report format for Table DISCH1
Emiss1	R	Default report format for Table EMISS1
Fines	R	Default report format for TABLE 10.
Indcodes	R	Default report format for TABLE 2.
Parmtabl	R	Default report format for TABLE 3.
Permits	R	Default report format for Table PERMITS.
Tpchr	R	Default report format for TABLE 7.

Table F-2. DEMDESS Output Tables and Files (cont'd)

Table	.Ext	Description
Wqstdesc	R	Default report format for TABLE 13.
Wqstds	R	Default report format for TABLE 14.
Currcnvt	R1	Report format for Table CURRCNVT.
Dischxdf	R1	Report format for TABLE 4.
Emissxdf	R1	Report format for TABLE 5.
Orgdess	R1	Report format for TABLE 1.
Outtabls	R1	Report format for this table, F-2.
Permitdf	R1	Report format for TABLE 11.
Scripts	R1	Report format for TABLE F-1.
Tpchr	R1	Report format for TABLE 7.
Tpcst	R1	Report format for TABLE 8.
Currcnvt	R2	Report format for TABLE 9, first part.
Currcnvt	R3	Report format for TABLE 9, second part.

Table F-3. Listings of Software Scripts

```
;Script "Jantra"
;
;Developed by WASH team, May 1992 - Tim Bondelid and Max Clark
;
play "trtplcy"
play "plots"
play "plotout"
;
;end of Script "Jantra"
;***********************************************************
;Script "TRTPLCY"
;MASTER SCRIPT TO RUN THE 4-STEP TREATMENT POLICY ANALYSIS
;SET UP FOR BULGARIA
;WRITTEN BY TIM BONDELID    WASH/AID - DEMDESS   MAY 1992
;
@5,12
?? "RUNNING THE TREATMENT POLICY ANALYSIS"
;CLEANUP FOR RUN
@6,12
?? "PERFORMING CLEANUP..."
CLEARALL
RELEASE PROCS ALL
RELEASE VARS ALL
IF ISTABLE("TRTPLCY1")
  THEN
  DELETE "TRTPLCY1"
  ENDIF
IF ISTABLE("TRTPLCY2")
  THEN
  DELETE "TRTPLCY2"
ENDIF
IF ISTABLE("TRTPLCY4")
  THEN
  DELETE "TRTPLCY4"
ENDIF
EMPTY "TRTPLCY3"
?? "DONE."
@7,12
?? "PERFORMING STEP 1 ..."
;COMPUTE PARAMETER-LEVEL FINES AND EMISSION LEVELS FOR EACH
;DISCHARGER AND TREATMENT LEVEL BEING ANALYZED
PLAY "TRTPLCY1"
DO_IT!
?? "DONE."
@8,12
```

```
?? "PERFORMING STEP 1B ..."
RENAME "ANSWER" "TRTPLCY1"
;CLEANUP TRTPLCY1 - NO NEGATIVE FINES AND CHANGE TAXID PER
;MUNICIPAL TREATMENT PLANT
CLEARALL
PLAY "TRTPLC1B"
DO_IT!
?? "DONE."
;SUM FINES AND BRING IN TREATMENT PLANT COST TABLES
CLEARALL
@9,12
?? "PERFORMING STEP 2 ..."
PLAY "TRTPLCY2"
DO_IT!
?? "DONE."
@10,12
?? "PERFORMING STEP 3 ..."
RENAME "ANSWER" "TRTPLCY2"
CLEARALL
;COMPUTE COSTS FOR EACH DISCH AND TREATMENT LEVEL
PLAY "TRTPLCY3"
CLEARALL
;MERGE IN LEV/TONNE BOD5
PLAY "TRTPLC3A"
DO_IT!
CLEARALL
DELETE "TRTPLCY3"
RENAME "ANSWER" "TRTPLCY3"
@11,12
?? "PERFORMING STEP 4 ..."
;SUMMARIZE COST DATA BY TREATMENT LEVEL FOR THE ENTIRE BASIN
PLAY "TRTPLCY4"
DO_IT!
RENAME "ANSWER" "TRTPLCY4"
?? "DONE."
@12,12
?? "PERFORMING STEP 5 ..."
PLAY "TRTPLCY5"
DO_IT!
?? "DONE."
; end of Script "TRTPLCY"
;***********************************************************
;Script "TRTPLCY1"
;
Query

 Emiss1 |    DISCHID     | TREATCLASS |  PARMCODE     |  VALUE     |
```

```
        | Check _disch | Check _tc   | Check _parm | Check _v |
        |              |             |             |          |
        |              |             |             |          |

Emiss1  | DISCHTYPE |
        | E         |
        |           |
        |           |

Disch1  | DISCHID | TREATCLASS | INDCODE | POP |    QTOT    |
        | _disch  | _tc        | 22      | >0  | Check _q   |
        |         |            |         |     |            |
        |         |            |         |     |            |

Disch1  | TAXID |     FINEID     |     WQSTD      | TPID  | TPLEV |
        | Check | Check _fineid  | Check _wqstd   | BLANK | BLANK |
        |       |                |                |       |       |
        |       |                |                |       |       |

Disch1  |  QCAP |
        | Check |
        |       |
        |       |

Disch1  |
        |
        |
        |

Tpchr  |                     TREATLEV                            | PARMCODE |
       | Check NONE OR PR01 OR PR02 OR SAS01 OR SAS02            | _parm    |
       |                                                         |          |
       |                                                         |          |

Tpchr |
PCTEFFL                                                               |
      |    _pcteffl,calc    _v*_pcteffl    as    neweffl,    CALC
_v*_pcteffl*_q/1000  as  load,  CALC  _v*(1-_pcteffl)*_q/1000  as
kgremoved |
      |
                                                                 |
      |
                                                                 |

Tpchr |
      |
      |
```

```
      |

 Wqstds | WQSTD   | PARMCODE |  WQSTDVAL     |
        | _wqstd  | _parm    | Check _wqsval |
        |         |          |               |
        |         |          |               |

 Fines | FINEID  | PARMCODE |                                  FINE
                             |
       |  _fineid  |   _parm           |  Check   _fine,   calc
_q*(_v*_pcteffl-_wqsval)*_FINE*.365 AS FINECOMP |
       |         |          |
                             |
       |         |          |
                             |

 Fines | CURRENCY |
       | Check    |
       |          |
       |          |

Endquery
;end of Script "TRTPLCY1"
;************************************************************
;Script "TRTPLC1B"
;CHANGE TAXID IF TREATMENT IS SELECTED
CLEARALL
EDIT "TRTPLCY1"
SCAN  ;TRTPLCY1
  IF [FINECOMP] < 0
    THEN
    [FINECOMP] = 0
  ENDIF
  IF [TREATLEV] <> "NONE"
    THEN
    [TAXID] = "BT1"
  ENDIF

ENDSCAN

DO_IT!


;End of Script "TRTPLC1B"
;************************************************************
;Script "TRTPLCY2"
Query
```

Trtplcy1	DISCHID	TREATCLASS	QTOT	TAXID
	Check _disch	Check _tc	Check _qtot	Check _TAXID

Trtplcy1	QCAP	TREATLEV	CURRENCY
	Check _qcap	Check _tl	_curr

Trtplcy1	FINECOMP
	calc sum as finetot

Tpcst	TREATLEV	A	B	I	J
	_tl	Check	Check	Check	Check

Tpcst	K	L	H	M	E&CPCT	SUBSIDYPCT
	Check	Check	Check	Check	Check	Check

Tpcst	INTRATE	YRSAMORT
	Check	Check

Disch1	DISCHID	TREATCLASS	POP
	_disch	_tc	Check

Disch1	

Disch1	

```
        |
Taxes | TAXID  |              TAX                       |
      | _TAXID | _tax, calc _qtot*_tax*365 as taxcomp   |
      |        |                                        |
      |        |                                        |

Taxes |
      |
      |
      |

Currcnvt | CURRENCY |            VALUE                  |
         | _curr    | _currv, calc _currv as currcnvt   |
         |          |                                   |
         |          |                                   |

Currcnvt | LABORCOST | ENERGYCOST | MATLCOST |
         | Check     | Check      | Check    |
         |           |            |          |
         |           |            |          |

Endquery
;end of Script "TRTPLC2"
;*******************************************************
;Script "trtplcy3"
;computes treatment costs, subsidies, etc for policy analysis
;written by Tim Bondelid for WASH/DEMDESP
;    May, 1992
clearall
EMPTY "trtplcy3"
view "trtplcy2"
edit "trtplcy3"
upimage ;go to trtplcy2

scan ;scan trtplcy2
  @12,12
  ?? "DISCHID = ", [DISCHID]

  ;ASSIGN LOCAL VARIABLES FROM TRTPLCY2
  DISCHID = [DISCHID]
  TREATCLASS = [TREATCLASS]
  TREATLEV = [TREATLEV]
  CURRCNVT = [CURRCNVT]
  LABORCOST = [LABORCOST] ;LOCAL CURRENCY/HR
  ENERGYCOST = [ENERGYCOST] ;LOCAL CURRENCY/KWH
  MATLCOST = [MATLCOST] ;LOCAL CURRENCY CONVERSION FROM USD
```

```
QTOT = [QTOT]
QCAP = [QCAP]
POP = [POP]
TAXTOT = [TAXCOMP]
A = [A] ;* 1000 ;TABLE IS IN $1000?
B = [B]
I = [I] ;* 1000 ;TABLE IS IN $1000?
J = [J]
K = [K] ;* 1000 ;TABLE IS IN $1000?
L = [L]
H = [H] ;* 1000 ;TABLE IS IN $1000?
M = [M]
SUBSIDYPCT = [SUBSIDYPCT] / 100
INTRATE = [INTRATE] / 100
YRSAMORT = [YRSAMORT]
FINETOT = [FINETOT]
ECPCT = [E&CPCT]

downimage ; GO TO trtplcy3 AND ASSIGN VALUES
[DISCHID] = DISCHID
[TREATCLASS] = TREATCLASS
[TREATLEV] = TREATLEV
[FINETOT] = FINETOT
[POP] = POP
[TAXTOT] = TAXTOT
[QTOT] = QTOT
QTOT1000 = QTOT/1000
[CAPCOST] = A * POW(QTOT1000, B) * CURRCNVT * 1000
[CAPCOST] = [CAPCOST] * (1 + ECPCT/100)
[SUBSIDYAMT] = [CAPCOST] * SUBSIDYPCT
[NETCAPCOST] = [CAPCOST] - [SUBSIDYAMT]
[ANNCAPCOST] = PMT([NETCAPCOST],INTRATE,YRSAMORT)
LABOR = H * POW(QTOT1000,M) * LABORCOST * 1000
MAT = I * POW(QTOT1000,J) * MATLCOST * 1000
ENERGY = K * POW(QTOT1000,L) * ENERGYCOST * 1000
[OMCOST] = LABOR + MAT + ENERGY
[DIRCOST] = [ANNCAPCOST] + [OMCOST]
[NETCOST] = [DIRCOST] + FINETOT + TAXTOT
[ANNSUBSIDY] = PMT([SUBSIDYAMT],INTRATE,YRSAMORT)

;CONVERT ALL COSTS IN TABLE TO MILLIONS OF CURRENCY
[FINETOT] = [FINETOT] / 1000000
[TAXTOT] = [TAXTOT] / 1000000
[CAPCOST] = [CAPCOST] / 1000000
[SUBSIDYAMT] = [SUBSIDYAMT] / 1000000
[NETCAPCOST] = [NETCAPCOST] / 1000000
[ANNCAPCOST] = [ANNCAPCOST] / 1000000
```

```
  [OMCOST] = [OMCOST] / 1000000
  [DIRCOST] = [DIRCOST] / 1000000
  [NETCOST] = [NETCOST] / 1000000
  [ANNSUBSIDY] = [ANNSUBSIDY] / 1000000
  DOWN ;ADD A RECORD
  UPIMAGE ;GO TO TRTPLCY2

endscan
DO_IT!
;FIND MINCOST OPTION FOR EACH DISCHARGER
CLEARALL
EDIT "TRTPLCY3"
NRECS = NRECORDS("TRTPLCY3")
HOLDDISCH = "          "

FOR I FROM 1 TO NRECS ;LOOP ON ALL RECORDS
  MOVETO RECORD I
  IF I = 1
    THEN
    HOLDDISCH = [DISCHID]
    MINVAL = [NETCOST]
    MINREC = I
  ENDIF
  IF HOLDDISCH <> [DISCHID] ;ON FIRST RECORD OF DISCHID
    THEN
    MOVETO RECORD MINREC ;MARK MIN COST RECORD
    [MINCOST] = "*"
    MOVETO RECORD I
    HOLDDISCH = [DISCHID]
    MINVAL = [NETCOST]
    MINREC = I
   ELSE ;NOT ON FIRST RECORD
    IF [NETCOST] < MINVAL
      THEN
      MINVAL = [NETCOST]
      MINREC = I
    ENDIF
  ENDIF
ENDFOR

;DO LAST DISCHID
MOVETO RECORD MINREC ;MARK MIN COST RECORD
[MINCOST] = "*"
DO_IT!
;end of Script "TRTPLCY3"
;*****************************************************
;Script "TRTPLC3A"
```

Query

Trtplcy3	DISCHID	TREATCLASS	TREATLEV	FINETOT	TAXTOT
	Check _disch	Check _tc	Check _tl	Check	Check

Trtplcy3	CAPCOST	SUBSIDYAMT	NETCAPCOST	ANNCAPCOST	ANNSUBSIDY
	Check	Check	Check	Check _cap	Check _sub

Trtplcy3	OMCOST	DIRCOST	NETCOST	POP	QTOT
	Check _om	Check _dircost	Check	Check	Check

Trtplcy3	MINCOST
	Check

Trtplcy1	DISCHID	TREATCLASS	PARMCODE	TREATLEV
	_disch	_tc	1	_tl

Trtplcy1	Kgremoved
	Check _kgr, calc (_cap+_sub+_om)*1000000/(_kgr*365) as Lev/KgBOD5

```
 Trtplcy1 |
          |
          |
          |

Endquery
;end of Script "TRTPLC3A"
;**************************************************************
;Script "TRTPLCY4"
Query

 Trtplcy3 | TREATLEV |     FINETOT       |     TAXTOT       |
          | Check    | CALC SUM AS FINE  | CALC SUM AS TAX  |
          |          |                   |                  |
          |          |                   |                  |

 Trtplcy3 |         CAPCOST         |        ANNCAPCOST        |
ANNSUBSIDY      |
          | CALC SUM AS CAPTOT | CALC SUM AS ANNCAP | CALC SUM AS
SUBSIDY |
          |                    |                    |
       |
          |                    |                    |
       |

 Trtplcy3 |      OMCOST     |      NETCOST      |        POP
|
          | CALC SUM AS O&M | CALC SUM AS NET | CALC SUM AS POPTOT
|
          |                 |                 |
|
          |                 |                 |
|

 Trtplcy3 |        QTOT          |     Kgremoved       |
          | CALC SUM AS FLOWTOT  | calc sum as KGBOD5  |
          |                      |                     |
          |                      |                     |

Endquery
;End of Script "TRTPLCY4"
;**************************************************************
;
;Script "trtplcy5"
;compute cumulative distribution values for pop, load, and flow
;cleanup
clearall
```

```
release procs all
release vars all
empty("trtplcy5")
Query

 Trtplcy1 |   DISCHID     | PARMCODE |  QTOT  |
          | Check _disch  | 1        | Check  |
          |               |          |        |
          |               |          |        |

 Trtplcy1 | TREATLEV |  Load  |
          | NONE     | Check  |
          |          |        |
          |          |        |

 Disch1 | DISCHID |  POP   |
        | _disch  | Check  |
        |         |        |
        |         |        |

 Disch1 |
        |
        |
        |

 Disch1 |
        |
        |
        |

Endquery
do_it!
sort "answer" on "pop"
clearall
popsum = csum("answer","pop")
loadsum = csum("answer","load")
qsum = csum("answer","qtot")
popcum = 0
loadcum = 0
qcum = 0
rank = 0
view "answer"
edit "trtplcy5"
upimage ;go to answer
scan
  rank = rank + 1
  popcum = popcum + [pop]
```

```
  loadcum = loadcum + [load]
  qcum = qcum + [qtot]
  downimage ;go to trtplcy5
  [DISCHID] = [ANSWER->DISCHID]
  [POP] = [ANSWER->POP]
  [QTOT] = [ANSWER->QTOT]
  [LOAD] = [ANSWER->LOAD]
  [POPCUM] = POPCUM
  [LOADCUM] = LOADCUM
  [QCUM] = QCUM
  [rank] = rank
  [poppct] = popcum/popsum
  [loadpct] = loadcum/loadsum
  [qpct] = qcum/qsum
  down ;add a record
  upimage ;go to answer
endscan
do_it!
;end of Script "TRTPLCY5"
;****************************************************
;Script "plots"
; creates plots of data for Jantra Basin, Bulgaria
; Max Clark, WASH Team, Danube Study, DEMDESS, 26 May 1992
clearall
release procs all
release vars all
;play "plotemp"; empties plot tables PLOTn.DB
play "plot1a"
play "plot1b"
; play "plot1c"
clearall
play "plot2a"
     ;play "plot2b"
play "plot2bx"
clearall
play "plots345"
play "plts345a"
clearall
play "plot3a"
DO_IT!         ;play "plot3b"
Rename "Answer" "Plot3"
clearall
play "plot4a"
DO_IT!         ;play "plot4b"
Rename "Answer" "Plot4"
clearall
play "plot5a"
```

```
DO_IT!              ;play "plot5b"
Rename "Answer" "Plot5"
clearall
play "plot6"
DO_IT!              ;play "plot6a"
Rename "Answer" "Plot6"
clearall
play "plot7a"
play "plot7b"
play "plot7c"
play "plot7d"
play "plot7e"
                ;play "plot7f"
clearall
play "rchrout"
clearall
play "plot8a"
DO_IT!          ;play "plot8b"
Rename "Answer" "Plot8"
clearall
play "rchrout2"
clearall
play "plot9a"
DO_IT!         ;play "plot9b"
Rename "Answer" "Plot9"
clearall
play "plot10a"
play "plot10b"
           ;play "plot10c"
sort "plot10" on "dist"
clearall
play "plot11a"
DO_IT!         ;play "plot11b"
play "plot11bx"
rename "answer" "Plot11"
clearall
;End of Script "PLOTS"
;**************************************************
;Script "Plot1A"
Query

 Trtplcy4 | TREATLEV  |  FINE    |   TAX    | ANNCAP   | SUBSIDY  |
          | Check     | Check    | Check    | Check    | Check    |
          |           |          |          |          |          |
          |           |          |          |          |          |

 Trtplcy4 |
```

```
         |
         |
         |
         |
Endquery
;End of Script "Plot1A"
;*********************************************************
;Script "Plot1B"
Do_It! Menu {Tools} {Rename} {Table} {answer} {plot1} {Replace}
Menu {Image} {Move} {TAX} Right Right Enter Menu {Image} {Move}
{SUBSIDY} Right Right Enter Menu {Image} {KeepSet} Up Left Left
UpImage ClearImage Menu {Scripts} {End-Record}
;End of Script "Plot1B"
;************************************************************
;Script "Plot2A"
Query

 Trtplcy4 | TREATLEV |  FINE  |  TAX   | ANNCAP |  O&M   |
          | Check    | Check  | Check  | Check  | Check  |
          |          |        |        |        |        |
          |          |        |        |        |        |

 Trtplcy4 |
          |
          |
          |

Endquery
;End of Script "Plot2A"
;*************************************************************
;Script "Plot2BX"
Do_It!
Menu {Image} {Move} {TAX} Right Right Enter UpImage ClearImage
Menu {Image} {KeepSet}
Menu {Tools} {Rename} {Table} {answer} {plot2} {Replace}
Menu {Scripts} {End-Record}
;End of Script "Plot2BX"
;**************************************************************
;Script "Plots345"
Query

 Trtplcy4 | TREATLEV | ANNCAP | SUBSIDY | O&M | NET  |
  POPTOT               |
          | Check     | _ann   | _sub    | _om | _net | _p,calc
_net*1000000/_p as Lev/head |
          |          |        |         |     |      |
                       |
```

```
           |          |        |         |     |      |
                         |

 Trtplcy4 |                  FLOWTOT                   |
          | _q,calc (_ann+_sub+_om)*1000000/(_q*365) as Lev/m3 |
          |                                            |
          |                                            |

 Trtplcy4 |                  KGBOD5
|
          | _k,calc (_ann+_sub+_om)*1000000/(_k*365) as Lev/kgBOD
|
          |
|
          |
|

Endquery
;End of Script "Plots345"
;*****************************************************************
;Script "Plts345A"
Do_It! Menu {Tools} {Rename} {Table} {ANSWER} {PLOTS345} {Replace}
ClearImage ClearImage {Scripts} {End-Record}
;End of Script "Plts345A"
;*****************************************************************
;Script "Plot3A"
Query

 Plots345 | TREATLEV | Lev/kgBOD |
          | Check    | Check >0  |
          |          |           |
          |          |           |

Endquery
;End of Script "Plot3A"
;*****************************************************************
;Script "Plot4A"
Query

 Plots345 | TREATLEV | Lev/head |
          | Check    | Check >0 |
          |          |          |
          |          |          |

Endquery
;End of Script "Plot4A"
;**************************************************************
```

```
;Script "Plot5A"
Query

 Plots345 | TREATLEV | Lev/m3   |
          | Check    | Check >0 |
          |          |          |
          |          |          |

Endquery
;End of Script "Plot5A""
;*************************************************************
;Script "Plot6"
Query

 Trtplcy5 |  RANK   |              POPPCT               |
LOADPCT          |
          | Check   | _p, calc _p*100 as Population | _l, CALC
_L*100 as BODload |
          |        |                                |
             |
          |        |                                |
             |

 Trtplcy5 |          QPCT             |
          | _Q, CALC _Q*100 as Flow |
          |                           |
          |                           |

Endquery
;End of Script "Plot6"
;***********************************************************
;Script "Plot7A"
Query

 Trtplcy3 | DISCHID | TREATLEV | FINETOT | TAXTOT | CAPCOST |
          | Check   | Check    | Check   | Check  | Check   |
          |         |          |         |        |         |
          |         |          |         |        |         |

 Trtplcy3 | ANNCAPCOST | ANNSUBSIDY | OMCOST | NETCOST |
          | Check      | Check      | Check  | Check   |
          |            |            |        |         |
          |            |            |        |         |

 Trtplcy3 |  POP  |  QTOT | MINCOST | Kgremoved | Lev/KgBOD5 |
          | Check | Check | ="*"    | Check     | Check      |
          |       |       |         |           |            |
```

```
     |        |        |          |           |            |

Endquery
;End of Script "Plot7A"
;***********************************************************
;Script "Plot7B"
Do_It! Menu {Modify} {Restructure} {answer} Ins
Right  "Town" Right  "n" Do_It!
Menu {Tools} {Rename} {Table} {answer}
{plot7bas} {Replace} Menu {Scripts} {End-Record}
sort "plot7bas" on "pop"
;End of Script "Plot7B"
;***********************************************************
;Script "Plot7C"
Query

 Plot7bas |  Town   | DISCHID | TREATLEV | ANNCAPCOST | ANNSUBSIDY
|
          | Check   | Check   | Check    | _cap       | _sub
|
          |         |         |          |            |
|
          |         |         |          |            |
|

 Plot7bas | OMCOST |          NETCOST                  |  POP   |
Kgremoved |
          | _om    | calc (_cap+_sub+_om) as cost | Check  | Check
   |
          |        |                              |        |
    |
          |        |                              |        |
    |

 Plot7bas |
          |
          |
          |

Endquery
;End of Script "Plot7C"
;***********************************************************
;Script "Plot7D"
Do_It! Menu {Modify} {Restructure} {answer} Down Down Down Down
Down Down  "Pop%" Right  "n" Enter  "BODrem%" Right  "n" Enter
 "Cost%" Right  "n" Do_It! Menu {Tools} {Rename} {Table} {answer}
{plot7ba4} {Replace} Menu {Scripts} {End-Record}
```

```
;End of Script "Plot7D"
;************************************************************
;Script "Plot7E"
;plot7e.sc
;compute cumulative distribution values for pop, bod removal and
cost
;cleanup
clearall
release procs all
release vars all
sort "plot7ba4" on "pop"
popsum = csum("plot7ba4","pop")
loadsum = csum("plot7ba4","Kgremoved")
qsum = csum("plot7ba4","cost")
town=0
popcum = 0
loadcum = 0
qcum = 0
edit "plot7ba4"

scan
  town=town+1
  popcum = popcum + [pop]
  loadcum = loadcum + [Kgremoved]
  qcum = qcum + [cost]
  [town]=town
  [POP%] = POPCUM*100/popsum
  [BODrem%] = LOADCUM*100/loadsum
  [Cost%] = QCUM*100/qsum
endscan
do_it!
;End of Script "Plot7E"
;*********************************************************
;Script "Rchrout"
;rout loads on reaches in rchfile
;cleanup
array loadrout[10] ;accumulater array by level for undecayed loads
array loadroutd[10] ;accumulater array by level for decayed loads
clearall
if istable("modllod1")
  then
  delete("modllod1")
endif
play "fillmlod"
do_it!
rename "answer" "modllod1"
clearall
```

```
play "fillmlo2"
do_it!
clearall
if istable("rchrout")
  then
  delete("rchrout")
endif
play "fillmrou" ;merge loads, etc into rchfile
do_it!
rename "answer" "rchrout"
sort "rchrout" on "seqno"
clearall
edit "rchrout"
for i from 1 to 10
  loadrout[i] = 0
  loadroutd[i] = 0
endfor
scan ;loop on reaches and accumulate loads
  @12,12
  ?? "On Seg ", [seg]
  if isblank([psload])
    then
    [psload] = 0
    [psflow] = 0
  endif
  lev = [lev]
  j = [j]
  if [type] = "S" ;new level path zero out accumulaters
    then
    loadrout[lev] = 0
    loadroutd[lev] = 0
  endif
  ;calc accumulated loads
  [tloadcum] = loadrout[lev]
  [bloadcum] = [tloadcum] + [psload]
  loadrout[lev] = [bloadcum]
  ;calc decayed loads
  [tloaddecayed] = loadroutd[lev]
  ;convert segl in km to m, vel in m/s to m/day
  trav = [segl]*1000 / ([vel]*86400) ;travel time in days
  decay = -[ka] * trav
  @13,12
  ?? trav, decay
  [bloaddecayed] = [tloaddecayed] * exp(decay)
  [bloaddecay2] = [bloaddecayed] + [psload]
  loadroutd[lev] = [bloaddecay2]
  if j < lev and j > 0 ;rout loads down to next level
```

```
    then
    loadrout[j] = loadrout[j] + [bloadcum]
    loadroutd[j] = loadroutd[j] + [bloaddecayed]
  endif
endscan
do_it!
clearall
;make rchgraf table for plotting
empty "rchgraf"
view "rchrout"
edit "rchgraf"
upimage ;rchrout
scan ;rchrout
  dist = -[pmile]
  downimage ;rchgraf
  [dist] = dist - [rchrout->segl]
  if [rchrout->type] = "S"
    then
    [seq] = 1
   else
    [seq] = 3
  endif
  [load] = [rchrout->tloadcum]
  [decayload] = [rchrout->tloaddecayed]
  [rchname] = [rchrout->rchname]
  down ;add a record
  [dist] = dist
  [seq] = 2
  [load] = [rchrout->bloadcum]
  [decayload] = [rchrout->bloaddecayed]
  [rchname] = [rchrout->rchname]
  down ;add a record
;  [dist] = dist
;  [seq] = 3
;  [load] = [rchrout->bloadcum]
;  [decayload] = [rchrout->bloaddecay2]
;  [rchname] = [rchrout->rchname]
;  down ;add a record
  upimage ;rchrout
endscan
do_it!
sort "rchgraf" on "dist" "seq"
;end of Script "RCHROUT"
;*********************************************************
;Script "Fillmlod"
;fillmlod.sc
;makes the mlodlod1 table for use in rchfile routing and modeling
```

```
;query below uses BOD and treatlev=NONE
;cleanup
;clearall
;if istable("modllod1")
;  then
;  delete("modllod1")
;endif
Query

 Rchfile |    BASIN       |    SEG       |    KM      |
         | Check _basin!  | Check _seg!  | Check _km! |
         |                |              |            |
         |                |              |            |

 Rchfile |
         |
         |
         |

 Rchfile |
         |
         |
         |

 Rchfile |
         |
         |
         |

 Disch1 | DISCHID | TREATCLASS | BASIN  | SEG  | KM  |
        | _disch  | _tc        | _basin | _seg | _km |
        |         |            |        |      |     |
        |         |            |        |      |     |

 Disch1 |
        |
        |
        |

 Disch1 |
        |
        |
        |

 Disch1 |
        |
        |
```

```
       |

Trtplcy1 | DISCHID | TREATCLASS | PARMCODE |        QTOT         |
         | _disch  | _tc        | Check 1  | calc sum as psflow  |
         |         |            |          |                     |
         |         |            |          |                     |

Trtplcy1 |  TREATLEV  |        Load         |
         | Check NONE | calc sum as psload  |
         |            |                     |
         |            |                     |

Trtplcy1 |
         |
         |
         |

Endquery
;do_it!
;rename "answer" "modllod1"
;clearall
;play "fillmlo2" ;fill in parmcode of 1 where blank
;do_it!
;clearall
;end of Script "FILLMLOD"
;*********************************************************
;Script "FILLMLO2"
Query

Modllod1 |      PARMCODE      |
         | blank, changeto 1  |
         |                    |
         |                    |

Endquery
;End of Script "FILLMLO2"
;********************************************************
;Script "FILLMROU"
Query

Rchfile |     BASIN     |     SEG     |     KM     |  TYPE  |
        | Check _basin  | Check _seg  | Check _km  | Check  |
        |               |             |            |        |
        |               |             |            |        |

Rchfile |  LEV   |   J    |
        | Check  | Check  |
```

```
          |        |        |
          |        |        |

Rchfile  | SEQNO  |  SEGL  | PMILE  | RCHNAME |
         | Check  | Check  | Check  | Check   |
         |        |        |        |         |
         |        |        |        |         |

Rchfile  |
         |
         |
         |

Modlrch  | BASIN  | SEG  | KMDS | CONDCODE |   Q   |  VEL  |
         | _basin | _seg | _km  | LOWFLO   | Check | Check |
         |        |      |      |          |       |       |
         |        |      |      |          |       |       |

Modlrch  |  K2   |
         | Check |
         |       |
         |       |

Modllod1 | BASIN  | SEG  | KM  |  PARMCODE    | TREATLEV | Psflow
|
         | _basin | _seg | _km | Check _parm  |          | Check
|
         |        |      |     |              |          |
|
         |        |      |     |              |          |
|

Modllod1 | Psload |
         | Check  |
         |        |
         |        |

Modlcoef | PARMCODE |                                        Ka

|
         | _parm     | Check calc 0 as tloadcum,calc 0 as
tloaddecayed,calc 0 as bloadcum,calc 0 as bloaddecayed,calc 0 as
bloaddecay2 |
         |           |

 |
```

```
          |          |

 |

 Modlcoef |
          |
          |
          |

Endquery
;End of Script "FILLMROU"
;*********************************************************
;Script "Plot8A"
Query

 Rchgraf | Dist  | Seq   | Load  | Decayload |   Rchname    |
         | Check | Check | Check | Check     | Check Jantra |
         |       |       |       |           |              |
         |       |       |       |           |              |

Endquery

Do_It!
;End of Script "Plot8A"
;*********************************************************
;Script "RCHROUT2"
;rout loads on reaches in rchfile
;this one uses the municipal-level least cost options for treatment
;cleanup
array loadrout[10] ;accumulater array by level for undecayed loads
array loadroutd[10] ;accumulater array by level for decayed loads
clearall
if istable("modllod2")
  then
  delete("modllod2")
endif
play "fillmllc"
do_it!
rename "answer" "modllod2"
clearall
play "fillml12"
do_it!
clearall
if istable("rchrout2")
  then
  delete("rchrout2")
endif
```

```
play “fillmrlc” ;merge loads, etc into rchfile
do_it!
rename “answer” “rchrout2”
sort “rchrout2” on “seqno”
clearall
edit “rchrout2”
for i from 1 to 10
  loadrout[i] = 0
  loadroutd[i] = 0
endfor
scan ;loop on reaches and accumulate loads
  @12,12
  ?? “On Seg ”, [seg]
  if isblank([psload])
    then
    [psload] = 0
    [psflow] = 0
  endif
  lev = [lev]
  j = [j]
  if [type] = “S” ;new level path zero out accumulaters
    then
    loadrout[lev] = 0
    loadroutd[lev] = 0
  endif
  ;calc accumulated loads
  [tloadcum] = loadrout[lev]
  [bloadcum] = [tloadcum] + [psload]
  loadrout[lev] = [bloadcum]
  ;calc decayed loads
  [tloaddecayed] = loadroutd[lev]
  ;convert segl in km to m, vel in m/s to m/day
  trav = [segl]*1000 / ([vel]*86400) ;travel time in days
  decay = -[ka] * trav
  @13,12
  ?? trav, decay
  [bloaddecayed] = [tloaddecayed] * exp(decay)
  [bloaddecay2] = [bloaddecayed] + [psload]
  loadroutd[lev] = [bloaddecay2]
  if j < lev and j > 0 ;rout loads down to next level
    then
    loadrout[j] = loadrout[j] + [bloadcum]
    loadroutd[j] = loadroutd[j] + [bloaddecayed]
  endif
endscan
do_it!
clearall
```

```
;make rchgraf table for plotting
if istable("rchgraf2")
  then
  empty "rchgraf2"
 else
  copy "rchgraf1" "rchgraf2"
  empty "rchgraf2"
endif
view "rchrout2"
edit "rchgraf2"
upimage ;rchrout2
scan ;rchrout2
  dist = -[pmile]
  downimage ;rchgraf2
  [dist] = dist - [rchrout2->segl]
  if [rchrout2->type] = "S"
    then
    [seq] = 1
   else
    [seq] = 3
  endif
  [load] = [rchrout2->tloadcum]
  [decayload] = [rchrout2->tloaddecayed]
  [rchname] = [rchrout2->rchname]
  down ;add a record
  [dist] = dist
  [seq] = 2
  [load] = [rchrout2->bloadcum]
  [decayload] = [rchrout2->bloaddecayed]
  [rchname] = [rchrout2->rchname]
  down ;add a record
  upimage ;rchrout2
endscan
do_it!
sort "rchgraf2" on "dist" "seq"
;end of Script "RCHROUT2"
;*****************************************************
;Script "FILLMLLC"
Query

 Rchfile |     BASIN      |     SEG      |     KM     |
         | Check _basin!  | Check _seg!  | Check _km! |
         |                |              |            |
         |                |              |            |


 Rchfile |
         |
```

Rchfile |

Rchfile |

Disch1	DISCHID _disch	TREATCLASS _tc	BASIN _basin	SEG _seg	KM _km

Disch1 |

Disch1 |

Disch1 |

Trtplcy1	DISCHID _disch	TREATCLASS _tc	PARMCODE Check 1	QTOT calc sum as psflow

Trtplcy1	TREATLEV _tl	Load calc sum as psload

Trtplcy1 |

```
 Trtplcy3 | DISCHID  | TREATCLASS | TREATLEV |
          | _disch   | _tc        | _tl      |
          |          |            |          |
          |          |            |          |

 Trtplcy3 | MINCOST |
          | "*"     |
          |         |
          |         |

Endquery
;End of Script "FILLMLLC"
;*********************************************************
;Script "FILLMLL2"
Query

 Modllod2 |      PARMCODE       |
          | blank, changeto 1   |
          |                     |
          |                     |

Endquery
;End of Script "FILLMLL2"
;*********************************************************
;Script "FILLMRLC"
Query

 Rchfile |     BASIN      |     SEG      |     KM      |  TYPE  |
         | Check _basin   | Check _seg   | Check _km   | Check  |
         |                |              |             |        |
         |                |              |             |        |

 Rchfile |  LEV   |   J    |
         | Check  | Check  |
         |        |        |
         |        |        |

 Rchfile | SEQNO  |  SEGL  | PMILE  | RCHNAME |
         | Check  | Check  | Check  | Check   |
         |        |        |        |         |
         |        |        |        |         |

 Rchfile |
         |
         |
         |
```

```
 Modlrch | BASIN   | SEG  | KMDS | CONDCODE |   Q   |  VEL  |
         | _basin  | _seg | _km  | LOWFLO   | Check | Check |
         |         |      |      |          |       |       |
         |         |      |      |          |       |       |

 Modlrch |   K2    |
         | Check   |
         |         |
         |         |

 Modllod2 | BASIN  | SEG  | KM  |  PARMCODE    | Psflow |
          | _basin | _seg | _km | Check _parm  | Check  |
          |        |      |     |              |        |
          |        |      |     |              |        |

 Modllod2 | Psload |
          | Check  |
          |        |
          |        |

 Modlcoef | PARMCODE |                                           Ka
|
          | _parm      | Check calc 0 as tloadcum,calc 0 as
tloadecayed,calc 0 as bloadcum,calc 0 as bloaddecayed,calc 0 as
bloaddecay2 |
          |          |
 |
          |          |
 |

 Modlcoef |
          |
          |
          |

Endquery
;End of Script "FILLMRLC"
;*******************************************************************
;Script "Plot9A"
Query

 Rchgraf2 | Dist  | Seq   | Load  | Decayload |    Rchname    |
          | Check | Check | Check | Check     | Check Jantra  |
          |       |       |       |           |               |
```

```
            |          |          |          |             |                |

Endquery
;End of Script "Plot9A"
;*****************************************************************
;Script "Plot10A"
Query

 Wqdata  | WQSITEID |  PARMCODE    |       VALUE            |
         | Check _w | Check _pc,1  | calc average as BOD5   |
         |          |              |                        |
         |          |              |                        |

 Wqdata  |
         |
         |
         |

 Wqsites  | WQSITEID | DATASRC | BASIN | SEG | KM |
          | _w       | Check   | _b    | _s  | _k |
          |          |         |       |     |    |
          |          |         |       |     |    |

 Wqsites  | WQSTADESC |
          | Check     |
          |           |
          |           |

 Rchfile  | BASIN | SEG | KM |
          | _b    | _s  | _k |
          |       |     |    |
          |       |     |    |

 Rchfile  |  WQSTD   |  LOWQ   |           PMILE                |
          | Check _d | Check   | _p,calc _p * -1 as dist        |
          |          |         |                                |
          |          |         |                                |

 Rchfile  | RCHNAME |
          | Jantra  |
          |         |
          |         |

 Rchfile  |
          |
          |
          |
```

```
Wqstds | WQSTD | PARMCODE | WQSTDVAL |
       | _d    | _pc      | Check    |
       |       |          |          |
       |       |          |          |

Endquery
;End of Script "Plot10A"
;*****************************************************************
;Script "Plot10B"
 Do_It! Menu {Image} {Move} {Dist} Right Enter Menu {Image} {Move}
{BOD5} Right Enter Menu {Image} {Move} {WQSTDVAL} Right Enter
Menu {Image} {KeepSet} Menu {Tools} {Rename} {Table}
{answer} {plot10} {Replace} Menu {Scripts} {End-Record}
;End of Script "Plot10B"
;****************************************************************
;Script "Plot11A"
Query

 Wqdata | WQSITEID | PARMCODE       | SAMPDATE | VALUE           |
        | Check _w | Check _pc,2    | Check    | Check as COD    |
        |          |                |          |                 |
        |          |                |          |                 |

 Wqdata |
        |
        |
        |

 Wqsites | WQSITEID | DATASRC | BASIN | SEG | KM |
         | _w       | Check   | _b    | _s  | _k |
         |          |         |       |     |    |
         |          |         |       |     |    |

 Wqsites | WQSTADESC |
         | Check     |
         |           |
         |           |

 Rchfile | BASIN | SEG | KM |
         | _b    | _s  | _k |
         |       |     |    |
         |       |     |    |

 Rchfile | WQSTD    | LOWQ  | PMILE                               |
         | Check _d | Check | _p,calc _p * -1 as dist,227.1       |
         |          |       |                                     |
         |          |       |                                     |
```

```
Rchfile | RCHNAME |
        | Jantra  |
        |         |
        |         |

Rchfile |
        |
        |
        |

Wqstds | WQSTD | PARMCODE | WQSTDVAL |
       | _d    | _pc      | Check    |
       |       |          |          |
       |       |          |          |

Endquery
;End of Script "Plot11A"
;*************************************************************
;Script "Plot11BX"
Menu {Image} {Move} {SAMPDATE} Right Enter
Menu {Image} {Move} {WQSTDVAL} Right Right Right Right Enter Left
Menu {Image} {KeepSet}
Menu {Scripts} {End-Record}
;End of Script "Plot11BX"
;*************************************************************
;Script "Plotout"
; Produces 11 plots, stored for Plot "n" in
;     files Plotn.db, Plotn.sc and Plotn.g
;       for   Tables, Scripts  and Graph settings [titles, type
;                                of bar chart or X-Y plot, etc.]
;     [except for the file Plot7ba4.db, used for Plot 7]
;
; Max Clark, WASH Danube Study, DEMDESS, 25 May 92
;
;  The 11 plots can be run individually using the files BIGn.SC
;
;  Plot 1: National-policy options (tax,fine,AnnCap,Subsidy) vs.
Treatlev
;   Plot  2: Municipal-level  options  (tax,fine,AnnCap,  O&M)  vs.
Treatlev
;  Plot 3: BOD removal costs,Lev/kg vs Treatlev
;  Plot 4: Annual Cost per Person, Lev/head vs. Treatlev
;  Plot 5: Cost per WW volume treated, Lev/m3, vs. Treatlev
;  Plot 6: Cum.% Population, BOD, Flow vs. Number of towns
;  Plot 7: Cum.% Population, BOD Removed, Cost vs. Number of towns
;  Plot 8: Existing BOD (Cum.+Decayed) vs. River Km
;  Plot 9: Min.Cost BOD (Cum.+Decayed) vs. River Km
```

```
;  Plot 10: Measured + WQ Standard BOD vs. River Km
;   Plot 11: Measured + WQ Standard COD vs. Date at one WQ
monitoring stn.
;
release procs all
release vars all
;
;   OUTPUT TO SCREEN, PRINTER, OR TABLE
printmode=0
while printmode<4
clearall
paintcanvas fill " "
  attribute 62
  2, 0, 24, 79
@5,5
?? "DEMDES Output for Treatment Policy Analysis Example"
@2,2
?? "Select output mode."
setmargin 12
@10,5
Text
CHOICE       OUTPUT

  1          Plots shown on screen
  2          Plots sent to printer
  3          Tables shown on screen
Endtext

showmenu
  "1" : "Screen",
  "2" : "Printer",
  "3" : "Tables",
  "Quit" : "Quit"
TO printtyp
@12,12
?? "printmode = ", printtyp
If printtyp="Quit"
then
printtyp="4"
endif
printmode = NUMVAL(printtyp)
if printmode=4
then
QUIT
else
printmode=printmode-1
endif
```

```
;
; PLOT(S) TO BE OUTPUT:
pnum=0
while pnum<12
clearall
paintcanvas fill " "
  attribute 62
  2, 0, 24, 79
@2,2
?? "Select type of plot or table:"
@5,5
SETMARGIN 6
Text
PLOT #:    CHOICES

     0: ALL Plots or Tables

Plot  1:  National-policy  options  (tax,fine,AnnCap,Subsidy)  vs.
Treatlev
Plot 2: Municipal-level options (tax,fine,AnnCap, O&M) vs. Treatlev

Plot 3: BOD removal costs,Lev/kg vs Treatlev

Plot 4: Annual Cost per Person, Lev/head vs. Treatlev

Plot 5: Cost per WW volume treated, Lev/m3, vs. Treatlev

Plot 6: Cum.% Population, BOD, Flow vs. Number of towns

Plot 7: Cum.% Population, BOD Removed, Cost vs. Number of towns

Plot 8: Existing BOD (Cum.+Decayed) vs. River Km

Plot 9: Min.Cost BOD (Cum.+Decayed) vs. River Km

Plot 10: Measured + WQ Standard BOD vs. River Km

Plot 11: Measured + WQ Standard COD vs. Date at one WQ monitoring
stn.
Endtext
showmenu
  "0" : "All",
  "1" : "National-level Options: Taxes, Fines, Subsidies, Mun. Ann.
Cap. Costs",
  "2" : "Municipal-level Options: Taxes, Fines, Ann. Cap. Costs,
O&M Costs",
  "3" : "Total Cost (Ann.Cap. + Subsidy + O&M) per kgBOD",
```

```
  "4" : "Net Annual Cost (Taxes + Fines + Ann.Cap. + O&M) per
Head",
  "5" : "Total Cost (Ann.Cap. + Subsidy + Taxes) per m3",
  "6" : "Cum. Population, BOD load (kg/day), Flow (cmd) by No. of
Towns",
  "7" : "Cum. Population, BOD removed, Cost by Towns [Min.Cost
Soln]",
  "8" : "River profile, existing conditions: Mun.BOD5 vs. River
Km",
  "9" : "River profile, min.cost solution: Mun.BOD5 vs. River Km",
  "10" : "River profile, Measurements: BOD5 vs River Km",
  "11" : "Measurements at 1 WQ Station: COD by date",
  "Quit" : "Quit"
TO printtyp
If printtyp="Quit"
then
printtyp="12"
endif
pnum = NUMVAL(printtyp)
If pnum = 0
then
inc = 1
else
inc = 0
endif
;
;  *****************  PLOT 1  ****************
;
pnum = pnum+inc
if pnum=1
then
if printmode<2
then
Menu {View} {plot1} Right Right
Menu {Image} {Graph} {Load} {plot1}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
```

```
          View "Plot1"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=2
then
;
;  *****************  PLOT 2  ****************
;
Clearall
if printmode<2
then
Menu {View} {plot2} Right Right
Menu {Image} {Graph} {Load} {plot2}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot2"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
endif
pnum=pnum+inc
if pnum=3
then
;
;  *****************  PLOT 3  ****************
;
Clearall
if printmode<2
then

Menu {View} {plot3} Right Right
Menu {Image} {Graph} {Load} {plot3}
```

```
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot3"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
endif
pnum=pnum+inc
if pnum=4
then
;
;  *****************  PLOT 4  ****************
;
Clearall

if printmode<2
then
Menu {View} {plot4} Right Right
Menu {Image} {Graph} {Load} {plot4}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot4"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
```

```
pnum=pnum+inc
endif
if pnum=5
then
;
;  *****************  PLOT 5  ****************
;
Clearall

if printmode<2
then
Menu {View} {plot5} Right Right
Menu {Image} {Graph} {Load} {plot5}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot5"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=6
then
;
;  *****************  PLOT 6  ****************
;
Clearall
if printmode<2
then
Menu {View} {plot6} Right Right
Menu {Image} {Graph} {Load} {plot6}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
```

```
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot6"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=7
then
;
;  *****************  PLOT 7  ****************
;
Clearall
if printmode<2
then
Menu {View} {plot7ba4} Right Right Right Right Right Right Right
Menu {Image} {Graph} {Load} {plot7}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot7ba4"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=8
then
;
;  *****************  PLOT 8  ****************
```

```
;
Clearall
if printmode<2
then
Menu {View} {plot8} Right Right Right
Menu {Image} {Graph} {Load} {plot8}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot8"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=9
then
;
;  *****************  PLOT 9  ****************
;
Clearall
if printmode<2
then
Menu {View} {plot9} Right Right Right
Menu {Image} {Graph} {Load} {plot9}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
```

```
          View "Plot9"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=10
then
;
;  *****************  PLOT 10  ****************
;
Clearall
if printmode<2
then
Menu {View} {plot10} Right Right
Menu {Image} {Graph} {Load} {plot10}
endif
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot10"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
pnum=pnum+inc
endif
if pnum=11
then
;
;  *****************  PLOT 11  ****************
;
Clearall
if printmode<2
then
Menu {View} {plot11} Right Right Right Right
Menu {Image} {Graph} {Load} {plot11}
endif
```

```
if printmode = 0
  then
  Menu {Image} {Graph} {ViewGraph} {Screen}
 else
if printmode = 1
  then
  Menu {Image} {Graph} {ViewGraph} {Printer}
endif
endif
       if printmode=2
          then
          View "Plot11"
                 wait table
                    prompt "Press F2 when finished viewing"
                 until "f2"
          endif
endif
pnum=pnum+inc
Clearall
endwhile; on plot number
endwhile; on type of output device
;end of Script "Plotout"
;*********************************************************
```