

3

WORKING WITH DESERT 1.0

3.1A GENERAL DESCRIPTION OF THE STEPS INVOLVED IN WORKING WITH DESERT

Working with DESERT requires following a series of steps from defining the input data sets to plotting the results.

- The first step involved is defining the input data files (**see Chapter 3.3**). Data files contain information such as the hydraulic description of the system, location and amount of emissions of waste load, points of measurements of water quality and quantity, etc.. Typically, such amount of heterogeneous information will be spread among several files and will have different formats. A particular text file, called the Universe file (extension **.UNV**), informs DESERT on how to retrieve the data from the appropriate files.
- Once defined the characteristics of the system, the user may desire to have a plot or to save the results of model simulations. In this case, he (or she) needs to establish a link between DESERT and a Spreadsheet (**see Chapter 3.4**).
- After these preliminary steps, DESERT can be invoked by double clicking the relative icon or by the other normal ways allowed by the Windows environment. It will appear a typical Windows application window (**see Chapter 3.5 and Chapter 4**)
- Through the opening command, the user will open the Universe file defined earlier. A representation of the systems as a binary tree will then appear. The tree will represent the reaches of the river system as well as the objects present in the system (**see figure 1.2.2**). Operating with graphical tools on the objects present on the tree, the user will define the methods to be used to carry out the hydraulic and water quality simulation (**see Chapter 3.5**).
- From the graphical representation of the Universe file, the reaches of interest will be moved to a new file, called the Rivernet file (extension **.RVR**). If a satisfactory file of this type was defined previously, then it may be open directly skipping the use of the Universe file. The Rivernet file supplies DESERT with the input data and the methods to be used to carry out both the hydraulic and the water quality simulation on the area of interest (**see Chapter 3.6**).
- Through the Terminal window (**see Chapter 3.7**), the user must define the water quality model (i.e., variables, right hand sides of the transport equations, values of the constants, etc.) and give plotting and operational instructions. These operations are possible by writing the instructions directly on the Terminal window. A more practical alternative is writing first the instruction in a text file, called Model file (**.MOD**), through any text editor and then import this file to DESERT.
- Calibration and optimization of the river model will be accomplished through the Terminal window also (**see Chapter 3.8 and 3.9**).

3.2 THE EXAMPLE USED IN THE MANUAL

In order to facilitate the understanding of the use of DESERT, the construction of a model will be reproduced step by step along Chapter 3. The model used as example is a trade off between simplicity and necessity of illustrating all the features of DESERT. For this reason, the numerical values used for the parameters are often not completely realistic. The riverine system of the example (**figure 3.2.1**) is composed by a Main river, with two second order tributaries (SecondA and SecondB). SecondA has also a tributary of the third order (Tertiary). Almost every reach presents two Cross section objects, to allow the use of the Steady state and Diffusion wave approximation. However, the Tertiary river, does not show any Cross section object. On this reach, only the Water and Matter balance methods can be applied. This situation was designed just for demonstrating the possibility of using different hydraulic and water quality equations on different reaches. A Weir is also located on the upper branch of the SecondA river. The weir has the water stage elevation-discharge relationship given in **table 3.2.1**.

WATER STAGE ELEVATION ABOVE M.S.L. (m)	WATER DISCHARGE (m ³ s ⁻¹)
185.000	0.000
186.600	0.010
186.700	0.765
186.725	1.062
186.750	1.385
186.775	1.733
186.800	2.103
186.850	2.899
186.900	3.761

Table 3.2.1 Water stage elevation-discharge relationship for WEIR0001.

The relationship has been derived using the formula for a broad-crested, well-ventilated weir, 13 m wide and 1.60 m high (White, 1986). However, to avoid stability problems during hydraulic dynamic simulation, a small discharge has been introduced for water stages lower than the crest of the weir.

The water quality simulation will regard a simple BOD-DO model. However, in order to give some background on the use of variables not subjected to the transport phenomena, an interaction with sediments has been introduced. Since the river system is artificial, the “measurement” data have been generated from the boundary conditions (**table 3.2.2**) using the model below:

$$\frac{\partial(A \cdot L)}{\partial t} + \frac{\partial(Q \cdot L)}{\partial x} = -K_r A \cdot L$$

$$\frac{\partial(A \cdot N)}{\partial t} + \frac{\partial(Q \cdot N)}{\partial x} = -K_n A \cdot N$$

$$\frac{\partial(A \cdot C)}{\partial t} + \frac{\partial(Q \cdot C)}{\partial x} = k_a B \cdot (C_s - C) - K_d A \cdot L - K_n A \cdot N - B \cdot K_{sOD}$$

where: L - carbonaceous biological oxygen demand (CBOD) in mg l⁻¹
N - nitrogenous biological oxygen demand (NBOD) in mg l⁻¹
C - dissolved oxygen concentration in mg l⁻¹
x - coordinate along the river, m
t - travel time in days
Q - streamflow in m³ d⁻¹
A - cross-section area in m²
B - stream width in m
K_r - carbonaceous BOD removal rate in d⁻¹
k_a - oxygen exchange coefficient in m d⁻¹ (see O'Connor and Dobbins, 1956)
K_d - CBOD oxygenation rate in d⁻¹,
K_n - NBOD oxygenation rate in d⁻¹,

K_{SOD} - sediment oxygen demand in $g\ m^{-2}\ d^{-1}$
 C_s - saturation concentration of dissolved oxygen in $mg\ l^{-1}$

The parameter values used in producing the “experimental data” were:

$$\begin{aligned}
 K_r &= 0.8\ d^{-1} \\
 K_n &= 0.8\ d^{-1} \\
 K_{SOD} &= 1.5\ g\ m^{-2}\ d^{-1} \\
 k_a &= k_{a0} f(T) \sqrt{\frac{U}{H}}
 \end{aligned}$$

where: $k_{a0} = 2.0\ m\ s^{-1/2}\ d^{-1}$
 U - flow velocity in $m\ d^{-1}$
 H - aeration depth, defined as the ratio A/B

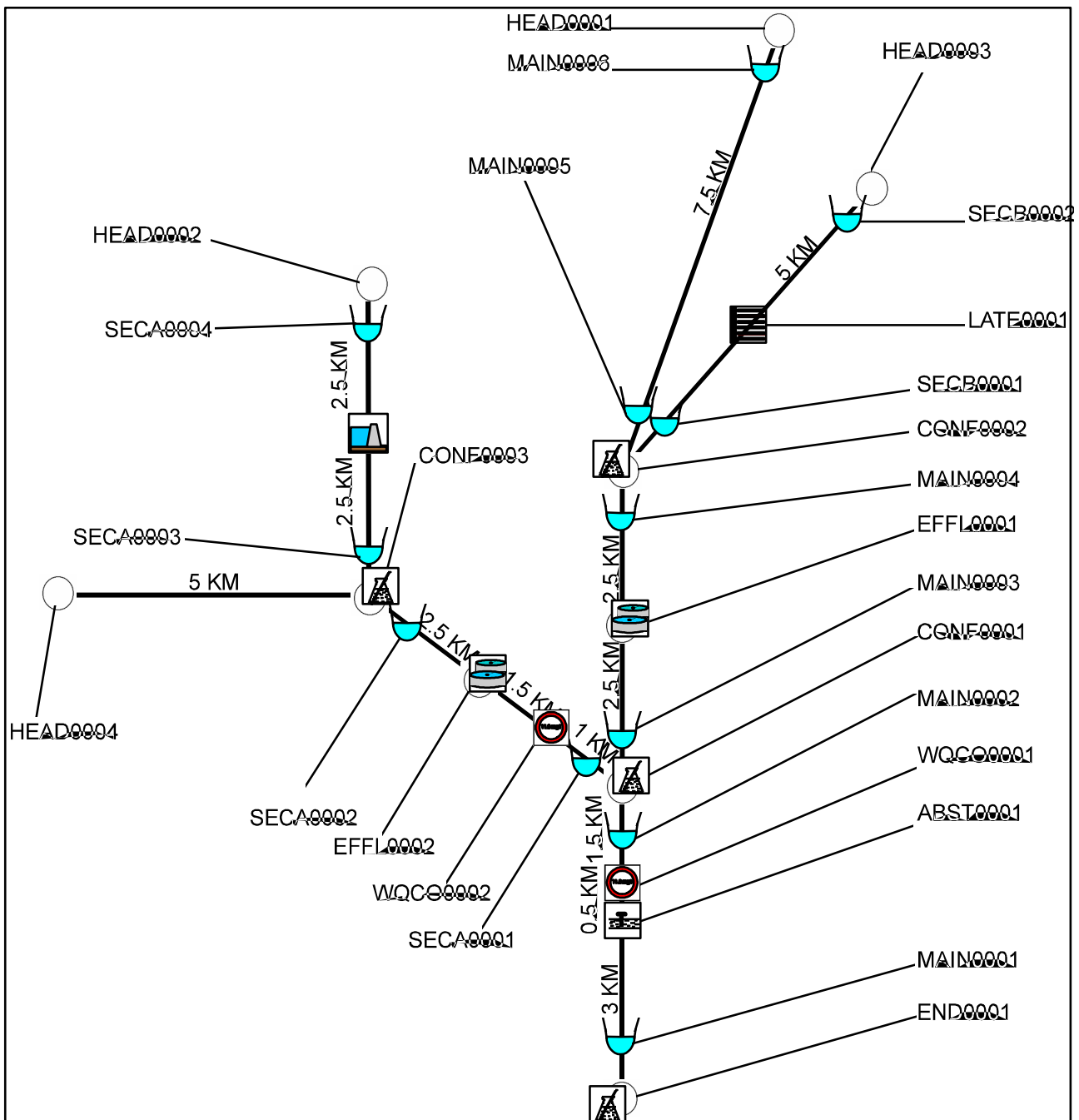


Figure 3.2.1 Description of the riverine system used as example.

The distances are intended from an object to another. However, cross-sectional profiles are not considered in determining distances (i.e., the upper part of the Main river has a total length of 7.5 km). Labels indicate the Code used for the objects.

Information on water discharges and quality (boundary conditions) are given in **table 3.2.2**.

IDCODE	FLOW (m ³ s ⁻¹)	DO (mg l ⁻¹)	CBOD (mg l ⁻¹)	NBOD (mg l ⁻¹)	TEMPERATURE (°C)
HEAD0001	3.0	9.0	12.5	2.5	18
HEAD0002	1.0	10.0	10.0	2.0	15
HEAD0003	0.7	10.0	10.0	2.0	15
HEAD0004	0.3	10.83	5.0	1.5	12
LATE0001	0.01 m ³ /s/km	7.0	15.0	20.0	15
EFFL0001	0.150	1.0	250.0	48.0	22
EFFL0002	0.100	1.0	250.0	48.0	22

Table 3.2.2 Initial boundary conditions for the example (No treatment on EFFL0001 and EFFL0002).

The results of the model simulation, obtained by using steady state approximation for both hydraulic and water quality equations and the boundary conditions presented in **table 3.2.2**, are summarized in **table 3.2.3**.

MEASUREMENT POINT	CBOD (mg l ⁻¹)	NBOD (mg l ⁻¹)	DO (mg l ⁻¹)
CONF0001	16.74	3.44	4.81
CONF0002	8.86	2.04	6.18
CONF0003	8.01	1.74	9.53
END	12.08	2.48	2.80

Table 3.2.3 Measurements generated by the model used.

The system is characterized by the presence of two sources of untreated waste water: one in the middle reach of the Main river and the other in the first reach of the SecondA river. Two water quality constraints are present in the lower part of the first reach of SecondA and just upstream of the abstraction point. This situation is the minimum requirement for optimization of Water treatment plants' allocation. Water treatment alternatives are summarized in **table 3.2.4**.

TREATMENT	CBOD (mg l ⁻¹)	NBOD (mg l ⁻¹)	UNIT OMR COST (US\$ l ⁻¹)
No treatment (NT)	250	48	0
Primary (P)	175	40	0.064
Chemically Enhanced Primary Treatment (CEPT)	113	34	0.103
Primary Precipitation (PC)	75	34	0.122
Biological (B)	25	34	0.106
CEPT+B (BC1)	25	31	0.142
PC + B (BC2)	12	31	0.165
CEPT+B+partial Denitrification (BC1DN)	12	19	0.168
PC+B+partial Denitrification (BC2DN)	12	19	0.200
PC+B+Denitrification (BCDN)	7	7	0.210

Table 3.2.4 Waste water treatment alternatives for the pollution sources EFFL0001 and EFFL0002.

The files specifying the system used in the example are contained in the directory **desert\example**. They are the following:

examplem.unv

examples.unv

masterex.dbf

infoexe1.dbf

infoexe2.dbf

profexe.dbf

alterex1.dbf

mapexe.bna

plotexe.xls

bodnsod.mod

bnsodopt.mod

3.3 THE INPUT DATA FILES

When performing the analysis of the water quality of a river basin, data involved are very heterogeneous, ranging from the hydraulic properties of the system under study, to location and magnitude of waste emissions, to location and value of water quality measurements, etc.. Often data come from different sources and have different format.

Most of these data are vital for DESERT to carry out simulation, calibration, and optimization work. Others, although not directly necessary to DESERT, can be useful to the user for a better understanding of the situation. Organizing the data in a format that could be used by the software and, at the same time, meet the variety of quantities and format of data that is typical of river basin management, is a challenging work. To achieve both these goals contemporarily, a powerful dBase style relational database engine has been introduced into DESERT.

The user has only to hierarchically organize data files in two levels. The upper level consists of a text file informing DESERT on the organization of data files (**Universe** file). The lower level groups the files actually containing information on the system:

- one .DBF file containing the list of the “objects” present in the system (**MasterTable** file);
- one or more DBF files containing the bathymetry of the reaches and weir characteristics (**ProfileTable** files);
- one .BNA file (optional) describing the geographical position of the structural objects (**MapTable** file);
- one or more .DBF files containing lateral inflows, waste water discharges, water treatment alternatives, and the water quality measurements (**InfoTable** files).
- one or more .DBF files (optional) containing the “cost”/”benefits” of the various alternatives for waste water treatment at the waste water treatment plants (**AlternativeTable** files).

When DESERT opens for the first time the .DBF files where information is stored, it creates a .NDX file for each .DBF file it opened. These index files are necessary to speed up the process of information retrieval by DESERT. Although they may occupy a not marginal spaces, they should not be canceled, otherwise a sensible slowing of the process will occur. However, if a change on the internal structure of any of the .DBF files has been made, the corresponding .NDX file must be canceled. If this precaution is not taken, a crash of the program may occur.

3.3.1 The Universe file (.UNV)

The universe file is a text file containing information on the organization of data regarding the system under study. It is subdivided in sections, each of which groups the required information on a table file. Each of these sections is uniquely identified by a header indicating the name of the table file that is described. A further section, named General, describes some features that are common to several table files. The name of sections must be enclosed within brackets. Sections do not need to be written in a particular order, but, when there is the possibility to have several tables of the same type, their enumeration must be consequent. For example, if three InfoTables are used, their headers must be [InfoTable1], [InfoTable2], and [InfoTable3]. Using a not ordered enumeration like [InfoTable1], [InfoTable2], and [InfoTable4], will induce DESERT to skip the reading of the tables out of order. Also, sections need to satisfy the particular format that is given below. In the following description, the word “field” relates to a database field of the file named at the beginning of the description of each section.

Description of the MasterTable file

[MasterTable]

Name=<path and name of the file containing the list of the objects defining the system>

RiverIndexField=<name of the field containing the River Index code>

CodeField=<name of the field containing the identification code of the objects>

OpCodeField=<name of the field containing the code indicating the type of object>

CISScale=<value with which multiplying the distances in the River Index to obtain the effective distances in meters>

CISLength=<number of digits composing the CIS system (variable by nation)>

RoughnessField=<name of the field containing the Manning’s roughness number of the river sections>

StyleField=<name of the field indicating an identification sub code of the type of object>

NameField=<name of the field containing the label to attribute to the objects>

General

[General]

Variant=<Default value for the management alternative. This name specifies the waste water treatment alternative that is used during normal simulation (i.e., when optimization is not performed)>

FlowField=<name of the field of the InfoTable and ProfileTable files that contains the measured discharge. This name field must be present for all the boundary condition and effluent specifications of all the InfoTables and for all the ProfileTables>

ElevationField=<name of the field of the InfoTable files that contains the water surface elevation of a water surface elevation-time relationship. The same name must be given to the field of the ProfileTable file that contains the elevation part of the pairs bottom elevation-width describing the reach cross-sectional profile and of the pairs water elevation-discharge relationship >

Description of the MapTable file (optional)*

[Map Table]

Name=<path and name of the file containing the Map information>

NumObjects=<the lateral size of the of the symbols in the system representation is set equal to the length of the shortest reach of the system divided by Numobjects. The default value is 3>

* When the user prefers to work with a synthetic generation of the river system representation, this section can be completely skipped.

```
[MasterTable]
name=example\Masterex.dbf
RiverIndexField=RIVERIND
CodeField=IDCODE
OpCodeField=OPCODE
CISScale=1
CISLength=6
SlopeField=SLOPE
RoughnessField=ROUGHM
StyleField=STYLE
NameField=NAME
```

```
[General]
Variant=1995
FlowField=Q
ElevationField=ELEV
```

```
[MapTable]
Name= example\Mapexe.bna
```

```
[ProfileTable1]
Name= example\Profexe.dbf
CodeField=IDCODE
ElevationField=ELEV
WidthField=WIDTH
```

```
[InfoTable1]
Name=example\Infoexe1.dbf
CodeField=IDCODE
DateField=DATE
TimeField=TIME
RadiusField=PERIOD
VariantField=VARIANT
```

```
[InfoTable2]
Name=example\Infoexe2.dbf
CodeField=IDCODE
DateField=DATE
TimeField=TIME
RadiusField=PERIOD
VariantField=VARIANT
```

```
[AlternativeTable1]
Name=example/Alterex1.dbf
CodeField=IDCODE
VariantField=VARIANT
```

Figure 3.3.1 The Universe file used in the Example (**example.unv**).

Description of the ProfileTable files

```
[ProfileTable #]      where # is the number of the Profile Table (e.g., 1, 2, etc.)
Name=<path and name of the file containing the bathymetry of the river sections>
CodeField=<name of the field containing the identification code of the sections>
WidthField=<name of the field containing the width data>
```

Description of the InfoTable files

[InfoTable #] where # is the number of the InfoTable (e.g., 1, 2, etc.)
Name=<path and name of the file containing the InfoTable>
CodeField=<name of the field containing the identification code of objects>
DateField=<name of the field containing the date of measurements>
TimeField=<name of the field containing the time of measurements>
RadiusField=<name of the field containing the frequency of measurements>
VariantField=<name of the field containing the variant label>

Description of the AlternativeTable files (optional)*

[AlternativeTable#] where # is the number of the AlternativeTable (e.g., 1, 2, etc.)
Name=<path and name of the file containing the AlternativeTable>
CodeField=<name of the field containing the identification code of the objects>
VariantField=<name of the field containing the variant label>

* When optimization is not performed, this section can be skipped completely.

3.3.2 The MasterTable file

Most of the data files are in DBF format. This format has been chosen for its wide use and availability. The Master Table File lists all the elements present in the system and some of their characteristics. It may have as many fields as needed. However, DESERT utilizes only the following items:

<CodeField>

This is a code that uniquely identifies the objects. The code does not need to follow any standard, but it must be unique for any object present in the network. This field is composed by a not restricted number of chars.

<NameField>

This name will appear on the geographical representation of the data carried out by the display unit. This field is composed by a not restricted number of chars.

<RiverIndexField>

This is the location of the object given in the River Index Code. The River Index Code is a way to describe a river system used in many countries. This method represents a river system as a binary tree. Therefore, each branch of the river system has a corresponding branch in the binary tree.

An order is associated to any branch. At every confluence of reaches in the binary tree it is possible to distinguish a main reach, that keeps the order of the downstream reach, and a secondary reach, that increases its order of 1. Thus, a position on the river system can be identified as a succession of distances composed by the number of digits indicated in the **CISLenght** specification of the Universe file.

If the object is situated on the main branch, then the first distance indicates the distance between the root of the tree and the object along this branch (of order 1) and the process terminates. If the object is not on the main branch, then the first distance is that between the root of the tree and the bifurcation where the path linking the object to the root leaves the main branch for a secondary branch (of order 2).

This process is recursively applied to the sub-tree constituted by the secondary branch and all its sub-branches. Consequently, the n^{th} distance represents the distance between the root of the order n^{th} and the object. If the object is located in a branch of order greater than n , then the n^{th} is the distance between the root of the branch of order n and the bifurcation from which the branch of order $n+1$, supporting the sub-tree where the object is located departs. The process is then reapplied for the $n+1^{\text{th}}$ distance. The final result is a series of distances that uniquely identify the "logical" position of the object. The user must remember that, on the contrary of what could be expected, the position of a confluence is always at the root of the secondary branch. This means that the River Index Code of a confluence terminates always with a value 0 expressed using as many digits as it is specified by the CISLenght specifier of the Universal file. The RiverIndexField is composed by a not restricted number of chars and the blocks of digits indicating the distances must not be separated from each other. For instance, in the example system (see figure 3.1) the objects EFFL0001, EFFL0002, HEAD0004, and CONF0002 have the following River Index codification:

EFFL0001:	003500
EFFL0002:	005000002500
HEAD0004:	005000005000005000
CONF0002:	010000000000

<OpCodeField>

These codes inform DESERT of the type of objects it is dealing with. The available object codes for the version 1.0 of DESERT are the following:

Structural objects

- 'R' indicates the end of a reach and the beginning of another reach. Reaches are segments of the river “uniform” from some viewpoint (see Chapter 1.1).
- 'E' indicates the root of the basin system (End).
- 'C' indicates the confluence of two reaches. In the synthetic representation of a river system, the secondary reach will intersect the main branch on the right.
- 'CL' indicates the confluence of two reaches. In the synthetic representation of a river system, the secondary reach will intersect the main branch on the left.
- 'H' indicates the upstream extremities of the river system (Headwater).

River objects

- 'T' indicates a treatment plant. Water quality and discharge alternatives characterizing treatment plants are given in an InfoTable, while the alternatives' costs are given in an AlternativeTable.
- 'I' indicates that along the reach where this object is located there is lateral inflow. Water quality and discharge characterizing the lateral inflow are given in an InfoTable.
- 'S' indicates a waste load point source. Water quality and discharge characterizing the source are given in an InfoTable.
- 'M' indicates a measurement point. Water quality and discharge characterizing the measurement point are given in an InfoTable.
- 'A' indicates an intake of water from a reach. Water quality and withdrawn discharge characterizing the intake are given in an InfoTable. However, DESERT will take into consideration only the discharge.
- 'Q' indicates a constrain in the water quality at any point. It is used to indicate that the water quality at a certain point needs to satisfy certain requirements. Water quality and discharge characterizing the water quality constrain are given in an InfoTable.
- 'F' indicates a cross section profile. Bathymetry regarding the cross section is given in a ProfileTable
- 'W' indicates the presence of a weir. A series of water elevation-flow pairs must be given in the ProfileTable, or a time-water elevation relationship must be described through an InfoTable.

Any of the River object can be associated to a structural object if they need to share the same location. For example a location can be at the same time a Source and a Reach point; in this case the code will be 'SR'. Other frequently used combinations are:

- Measurement and Headwater point (code 'MH');
- Measurement and End point (code 'ME');
- Source and Reach point (code 'SR');
- Confluence and measurement point (code 'MC')

Of course two structural objects can not be associated together (e.g., the code 'RC' can not be accepted). If two or more river object need to share the same position, than they need to be associated to a structural object. For example, if an abstraction point and a measurement point are located at the same position along a reach, then the first must be declared as 'AR' and the second as 'MR'.

WARNING

The Treatment plant, the Water quality constrain, and the Weir object must be associated to a structural object. Their presence changes drastically the way hydraulic simulation or optimization are performed. If the codes 'W', 'T', or 'Q' are used alone, the relative objects will not be considered. Therefore, for example, the code for a treatment plant must be 'TH' if it is associated to a headwater, 'TC' if it is associated to a confluence, 'TR' if it is associated to a point of separation of reaches, or 'TE' if it is associated to the root of the system.

<StyleField>

The general characteristic of objects are indicated by their operational code. However, additional information may be sometimes necessary. The StyleField was introduced just to permit the definition of subsets of a main class of objects.

In the present version of DESERT, the StyleField is used in connection with River cross section objects and Weir. The River Cross sections are defined by a series of pairs width-elevation of the river bottom above the mean sea level. DESERT uses this information not only to calculate the cross-sectional area corresponding to a given water height, but also it uses the elevation of the bottom in the calculation of the slope of the reach. It may happen that, while the shape of the cross-section is reliable, its elevation above the mean sea level is not. In this case it may be useful to retain the shape of the profile, but the bottom elevation should not be used for the slope calculation. A "1" in the style field informs DESERT just of this situation.

The Weir objects can belong to two subclasses: the water stage elevation-discharge weirs (where a $Q(z)$ relationship is established) and the time-water stage weirs (where a $z(t)$ relationship is formulated). The first ones are individuated by 'QZ' present on the style field. The latter ones by 'ZT' in the Style field. In base of the value in the style field, DESERT will look either in the ProfileTable or in the InfoTable for the data describing weir's behavior. This field is composed by a not restricted number of chars.

<RoughnessField>

This field contains the Manning's roughness number characterizing the river sections of the system. DESERT does not calibrate the hydraulic model of the system, but it only performs hydraulic simulation assuming as external data the value of cross sections, elevations, and roughness determined preliminarily using hydraulic models or table values. This field is a dBase float composed of n digits of which $m < n$ are decimal. The size of m and n is chosen by the user, taking into consideration the limits of the dBase language.

IDCODE	NAME	RIVER	RIVERIND	OPCODE	STYLE	ROUGHM
MAIN0001	1st section on main	MAIN	000500	F		0.03000
MAIN0002	2nd section on main	MAIN	004500	F		0.03000
MAIN0003	3rd section on main	MAIN	005500	F		0.03000
MAIN0004	4th section on main	MAIN	009500	F		0.03000
MAIN0005	5th section on main	MAIN	010500	F		0.03000
MAIN0006	6th section on main	MAIN	017000	F		0.03000
SECA0001	1st section on SecondA	SECA	005000000500	F		0.03000

SECA0002	2nd section on SecondA	SECA	005000004500	F		0.03000
SECA0003	3rd section on SecondA	SECA	005000005500	F		0.03000
SECA0004	4th section on SecondA	SECA	005000009500	F		0.03000
SECB0001	1st section on SecondB	SECB	010000000500	F		0.03000
SECB0002	2nd section on SecondB	SECB	010000004500	F		0.03000
CONF0001	Confluence of SecondA on Main	MAIN	005000000000	MC		
CONF0002	Confluence of SecondB on Main	MAIN	010000000000	MC		
CONF0003	Confluence of Tertiary on SecondA	SECA	005000005000000000	MC		
HEAD0001	Headwater of Main	MAIN	017500	MH		
HEAD0002	Headwater of SecondA	SECA	005000010000	MH		
HEAD0003	Headwater of SecondB	SECB	010000005000	MH		
HEAD0004	Headwater of Tertiary	TERT	005000005000005000	MH		
END0001	Root of the system	MAIN	000000	ME		
EFFL0002	Effluent source on lower SecondA	SECA	005000002500	TR		
EFFL0001	Effluent source on intermediate Main	MAIN	007500	TR		
LATE0001	Lateral inflow on SecondB	SECB	010000002500	I		
ABST0001	Abstraction point on lower Main	MAIN	003000	A		
WQCO0001	Water quality constrain before the abstraction	MAIN	003500	QR		
WQCO0002	Water quality constrain on lower SecondA	SECA	005000001000	QR		
WEIR0001	Weir on the upper SecondA	MAIN	005000007500	WR	QZ	

Table 3.3.1 The organization of the MasterTable file used for the example (masterex.dbf).

3.3.3 The MapTable file

The MapTable file contains the geographical coordinates of all structural objects. These data help DESERT in laying down a scheme of the system closer to the real one.

The MapTable file is in the format of commercial Map Viewer (TM) mapping software from Golden Software Inc., although it can be easily produced manually from any text editor. Each object needs two lines of description: in the upper row, the first field must contain the identification code of the object between double quotation marks, the second field can contain its name, again between double quotation marks, eventually, the third field is a 1. The three fields are separated from each other by commas. In the second row the two remaining fields, separated by a comma, will contain the geographical coordinates.

```
"END0001","Downstream end of Main",1
16.00,13.00
"CONF0001","Confluence of SecondA on Main",1
16.00,15.00
"CONF0002","Confluence of SecondB on Main",1
16.00,17.00
"CONF0003","Confluence of Tertiary on SecondA",1
14.40,16.20
"HEAD0001","Headwater of Main",1
17.00,19.80
"HEAD0002","Headwater of SecondA",1
14.40,18.20
"HEAD0003","Headwater of SecondB",1
17.60,18.80
"HEAD0004","Headwater of Tertiary",1
12.40,16.20
"EFFL0001","Treatment plant on intermediate Main",1
16.00,16.00
"EFFL0002","Treatment plant on lower SecondA",1
15.10,15.60
"WQCO0001","Water quality before the abstraction",1
16.00,14.40
"WQCO0002","Water quality constrain on lower SecondA",1
15.68,15.24
"WEIR0001","Weir on the upper SecondA",1
14.40,17.20
```

Figure 3.3.2 The mapexe.bna file.

In the case the MapTable file is not specified, DESERT will proceed to the synthetic drawing of the river scheme, according to the logical position of objects given in the MasterTable (see **Figure 1.2.1**).

In both cases, DESERT determines the ratio between height and width of the representation of the riverine system. The system is then drawn to obtain that the smallest between the ratios width of the drawing/width of the Universe

file window and height of the drawing/height of the Universe file window is equal to two (2) and at the same time preserving the ratio between the height and the width of the representation.

3.3.4 The ProfileTable files

A ProfileTable file is a .DBF file containing the bathymetry of the cross section profiles. Further, since weirs can be defined either in terms of a water stage-discharge relationship or in terms of a time-water stage relationship, the first type of relationship must be described in the ProfileTable, while the second should be described in a InfoTable. Profiles are defined by a series of pairs width (or distance from the longitudinal axis of the river)-elevation of the river bottom above mean sea level. Weirs' characteristics are here defined as a series of pairs elevation-discharge. Thus each record of this file can contain either information on a cross-sectional profile or on a weir. Anyway, this file must be composed by at least the following fields:

<CodeField>

This field is composed by a not restricted number of chars (see Section 3.3.2). However, the size of this field must equal that declared for the same field of the MasterTable file.

<WidthField>

This is the width part of the width-elevation pair. In the case the record is describing a weir's water stage-discharge relationship, this field will not be considered. This field is a dBase float composed of n digits of which m<n are decimal. The size of m and n is chosen by the user, taking into consideration the limits of the dBase language.

<ElevationField>

This field can be either the elevation part of the width -elevation pair or the water stage part of a water stage elevation-discharge pair. This field is a dBase float composed of n digits of which m<n are decimal. The size of m and n is chosen by the user, taking into consideration the limits of the dBase language.

<FlowField>

This is the discharge part of a water stage-discharge pair. If the record keeps information on a cross-section profile, then this field will not be considered. This field is a dBase float composed of n digits of which m<n are decimal. The size of m and n is chosen by the user, taking into consideration the limits of the dBase language.

To completely define the profile of one river cross section, the user must submit several width-elevation records regarding one of the halves of the cross-section profile (i.e., the space from the longitudinal axis to one of the sides). DESERT will complete the section by symmetry and linear interpolation. For defining weirs, the user needs to supply a series of water stage-discharge pairs. During the Steady state hydraulic solution, to compute the water stage DESERT will use the upstream discharge. The corresponding stage will be determined by linear interpolation of the supplied water stage-discharge pairs.

IDCODE	RIVER	DISTANCE	ELEV	WIDTH
MAIN0001	MAIN	000500	160.50000	0.00000
MAIN0001	MAIN	000500	163.50000	16.00000
MAIN0001	MAIN	000500	165.50000	22.00000
MAIN0002	MAIN	004500	164.50000	0.00000
MAIN0002	MAIN	004500	167.50000	15.00000
MAIN0002	MAIN	004500	169.50000	21.00000

MAIN0003	MAIN	005500	165.50000	0.00000
MAIN0003	MAIN	005500	168.50000	15.00000
MAIN0003	MAIN	005500	170.50000	21.00000
MAIN0004	MAIN	009500	169.50000	0.00000
MAIN0004	MAIN	009500	172.50000	14.00000
MAIN0004	MAIN	009500	174.50000	20.00000
MAIN0005	MAIN	010500	170.50000	0.00000
MAIN0005	MAIN	010500	173.50000	12.00000
MAIN0005	MAIN	010500	175.50000	18.00000
MAIN0006	MAIN	017000	177.00000	0.00000
MAIN0006	MAIN	017000	179.00000	7.00000
MAIN0006	MAIN	017000	182.00000	15.00000
SECA0001	SECA	000500	166.00000	0.00000
SECA0001	SECA	000500	168.00000	6.00000
SECA0001	SECA	000500	170.00000	12.00000
SECA0002	SECA	004500	174.00000	0.00000
SECA0002	SECA	004500	175.00000	6.00000
SECA0002	SECA	004500	178.00000	12.00000
SECA0003	SECA	005500	177.00000	0.00000
SECA0003	SECA	005500	178.00000	6.00000
SECA0003	SECA	005500	181.00000	11.00000
SECA0004	SECA	009500	193.00000	0.00000
SECA0004	SECA	009500	194.00000	5.00000
SECA0004	SECA	009500	197.00000	10.00000
SECB0001	SECB	000500	171.00000	0.00000
SECB0001	SECB	000500	172.00000	6.00000
SECB0001	SECB	000500	175.00000	11.00000
SECB0002	SECB	004500	179.00000	0.00000
SECB0002	SECB	004500	180.00000	7.00000
SECB0002	SECB	004500	183.00000	10.00000

Table 3.3.2 The ProfileTable used in the example (profexe1.dbf).

3.3.5 The InfoTable files

InfoTable files contain very heterogeneous information on the system. Typically, they are used to store data on the water quality measurements of the effluent discharges into the system. They can also contain measurement data or constraints on the water quality of some reach of the system. However, they can be used to store water discharges or water stage measurements only. These values can be used as boundary conditions for the hydraulic and water quality simulation, for the calibration, and for the optimization tasks (see **Chapter 1.2**).

Since the number of water quality parameters measured at any site will vary from country to country and from project to project, their number has not been fixed. Of course, all the records of the same InfoTable will have to report the same number and type of data, but the possibility to use several InfoTables, allow a great flexibility. However, some fields need always to be present, such as:

<CodeField>

This field contains the identification code of the object. As specified in **Sections 3.3.2** and **3.3.4**, it can contain a not restricted number of chars, but it has to be declared of the same size of the analogous field in the MasterTable file.

<DateField>

This field contains the date of the measure. The date and time of measure are used when dynamic approximation of hydraulic or water quality equation is performed, in order to change boundary conditions or calibration matching with time. This field is of date type.

<TimeField>

This field contains the time of measure. It is composed by 5 chars. It plays the same role of the DateField.

<RadiusField>

This field contains the sampling time of measures. DESERT controls that the initial time and final time of simulation fall in the period of validity of available data. The period of validity is taken as <date and time of measure \pm Sampling time/2>. In the case some parts of the simulation period are not covered by measured data, linear interpolation of the nearest available data is performed. The RadiusField is composed by one char. Available RadiusField codes are the following:

- 'H' for hourly sampling
- 'D' for daily sampling
- 'W' for weekly sampling
- 'M' for monthly sampling
- 'Y' for yearly sampling

<VariantField>

This field contains the label indicating to what version the data belongs. It is mainly used for characterizing effluents of different water treatment schemes. Through the value of this field, the water quality records and the “cost” records regarding the same management alternative of wastewater treatment can be related to each other. It can be composed by a not limited number of chars.

<FlowField>

This field contains the measured discharge in $m^3 s^{-1}$. This field is a dBase float composed of n digits of which m<n are decimal. The size of m and n is chosen by the user, taking into consideration the limits of the dBase language. This field is necessary only for characterizing the objects that acts as upstream boundaries of the system (i.e., Headwater, Lateral inflow, Point sources, Treatment plant, Abstraction, and water stage-time Weir objects). For the other objects (i.e., Measurement and Water quality constrains, this field is not necessary

The field containing the other information can be of any type available in dBase. However, DESERT will automatically discard all the non numeric fields. To indicate that a numerical datum of one record is not reliable or available, all the digits of the relative field must be set to 9. The format (i.e., the total number of digits and the number of decimal digits) of the field must be respected. In this case DESERT is not taking the datum into consideration for the simulation. Data can be conveniently organized in different InfoTable files. In the example, the data relative to the water quality resulting from the different waste water treatment alternatives have been separated from those regarding the boundary conditions and water quality measures and constraints. It is also to be noticed that the two InfoTables do not contain the same fields. Because no heat budget was simulated in the example, the effluent water temperature of the water treatment plant is not necessary.

IDCODE	DATE	TIME	PERIOD	VARIANT	Q	DO	CBOD	NBOD	T
HEAD0001	2/28/95	12:00	D	1995	3.000	9.0	12.5	2.5	18.0
HEAD0002	2/28/95	12:00	D	1995	1.000	10.0	10.0	2.0	15.0
HEAD0003	2/28/95	12:00	D	1995	0.700	10.0	10.0	2.0	15.0
HEAD0004	2/28/95	12:00	D	1995	0.400	10.8	5.0	1.5	12.0
LATE0001	2/28/95	12:00	D	1995	0.010	7.0	15.0	25.0	15.0
ABST0001	2/28/95	12:00	D	1995	0.100	0.0	0.0	0.0	15.0
CONF0001	2/28/95	12:00	D	1995	5.400	9999.9	9999.9	9999.9	17.7
CONF0002	2/28/95	12:00	D	1995	3.750	9999.9	9999.9	9999.9	17.4
CONF0003	2/28/95	12:00	D	1995	1.400	9999.9	9999.9	9999.9	14.0
WQCO0001	2/28/95	12:00	D	1995	5.400	5.5	11.3	2.3	18.0
WQCO0002	2/28/95	12:00	D	1995	1.500	5.5	11.3	2.3	15.0
END0001	2/28/95	12:00	D	1995	5.300	9999.9	9999.9	9999.9	18.0

Table 3.3.3 The InfoTable used in the example for boundary conditions and water quality constraints and measurements (Infoexe1.dbf).

IDCODE	DATE	TIME	PERIOD	VARIANT	Q	DO	CBOD	NBOD
EFFL0001	2/28/95	12:00	D	1995	0.450	1.0	250.0	48.0
EFFL0001	2/28/95	12:00	D	P	0.450	1.0	175.0	40.0
EFFL0001	2/28/95	12:00	D	CEPT	0.450	1.0	113.0	34.0
EFFL0001	2/28/95	12:00	D	PC	0.450	1.0	75.0	34.0
EFFL0001	2/28/95	12:00	D	B	0.450	1.0	25.0	34.0
EFFL0001	2/28/95	12:00	D	BC1	0.450	1.0	25.0	31.0
EFFL0001	2/28/95	12:00	D	BC2	0.450	1.0	12.0	31.0
EFFL0001	2/28/95	12:00	D	BC1DN	0.450	1.0	12.0	19.0
EFFL0001	2/28/95	12:00	D	BC2DN	0.450	1.0	12.0	19.0
EFFL0001	2/28/95	12:00	D	BCDN	0.450	1.0	7.0	7.0
EFFL0002	2/28/95	12:00	D	1995	0.300	1.0	250.0	48.0
EFFL0002	2/28/95	12:00	D	P	0.300	1.0	175.0	40.0
EFFL0002	2/28/95	12:00	D	CEPT	0.300	1.0	113.0	34.0
EFFL0002	2/28/95	12:00	D	PC	0.300	1.0	75.0	34.0
EFFL0002	2/28/95	12:00	D	B	0.300	1.0	25.0	34.0
EFFL0002	2/28/95	12:00	D	BC1	0.300	1.0	25.0	31.0
EFFL0002	2/28/95	12:00	D	BC2	0.300	1.0	12.0	31.0
EFFL0002	2/28/95	12:00	D	BC1DN	0.300	1.0	12.0	19.0
EFFL0002	2/28/95	12:00	D	BC2DN	0.300	1.0	12.0	19.0
EFFL0002	2/28/95	12:00	D	BCDN	0.300	1.0	7.0	7.0

Table 3.3.4 The InfoTable used in the example for the water quality of the different waste water treatment alternatives of the two water treatment plants (Infoexe2.dbf).

3.3.6 The AlternativeTable files

In the AlternativeTable file the “cost” of the waste water treatment alternatives needs to be specified. This file, which is also in a .DBF format, must contain the following two fields:

<CodeField>

This field contains the identification code of the object. As specified in Sections 3.3.2, 3.3.4, and 3.3.5, it can contain a not restricted number of chars, but it has to be declared of the same size of the analogous field in the MasterTable file.

<VariantField>

This field contains the label indicating to what version the data belongs. This field is mainly used for characterizing effluents of different water treatment schemes. It can be composed by a not limited number of chars.

The benefit/cost of the various alternatives must be specified in some kind of cost values that could be added. This is because at any stage of the dynamic programming method used to obtain the optimal allocation of resources, the “cost” of a feasible solution is obtained adding the cost of the feasible alternatives of the water treatment plant to the “cost” of the previous stage feasible solutions (see Chapter 1.2.7).

IDCODE	VARIANT	COSTUNIT (US\$/m ³)	COSTTOT (US\$/s)
EFFL0001	1995	0.000	0.0000
EFFL0001	P	0.064	0.0288
EFFL0001	CEPT	0.103	0.0464
EFFL0001	PC	0.122	0.0549
EFFL0001	B	0.106	0.0477
EFFL0001	BC1	0.142	0.0639
EFFL0001	BC2	0.165	0.0743
EFFL0001	BC1DN	0.168	0.0756
EFFL0001	BC2DN	0.200	0.0900
EFFL0001	BCDN	0.210	0.0945
EFFL0002	1995	0.000	0.0000
EFFL0002	P	0.064	0.0192
EFFL0002	CEPT	0.103	0.0309
EFFL0002	PC	0.122	0.0366
EFFL0002	B	0.106	0.0318
EFFL0002	BC1	0.142	0.0426
EFFL0002	BC2	0.165	0.0495
EFFL0002	BC1DN	0.168	0.0504
EFFL0002	BC2DN	0.200	0.0600
EFFL0002	BCDN	0.210	0.0630

Table 3.3.5 The AlternativeTable used in the Example (alterex1.dbf).

The measurement units for the cost have been added here for illustrative purposes and are not present in the actual file.

3.4 PREPARATION OF THE LINKING FOR EXPORTING THE DATA

One of the most important steps in using simulation is evaluating model predictions. In doing this, scientists and managers need to study a large amount of data. Often this process is facilitated by the use of graphic representations. DESERT software allows to use directly the powerful means of spreadsheets to accomplish this task. Although the OLE specifications used by DESERT are theoretically supported by many commercial spreadsheets, it turned out that, among the products tried, only Microsoft Excel for Windows version 4.0 and 5.0 have worked properly (see **Chapter 1.2.5**). Thus, it is strongly recommended the use of this popular software and all the present manual will make reference to it.

Some preliminary operations are necessary before running the system simulation if results have to be saved.

The first operation is to create an Excel worksheet (in the example this file is named **plotexe.xls**) large enough to contain the desired data. This file must be filled with some sort of data. The kind of data is not important because DESERT will automatically erase them and substitute with the data that it produces. However, the number of data is important, since it fixes the size of the worksheet. Every simulation data that will not fit inside this worksheet will be discarded. This implicates that it is recommendable to create a very large file, composed of some hundreds of rows per some tens of columns, according to the possible amount of data to be saved.

3.5 WORKING ON THE UNIVERSE FILE REPRESENTATION

After having defined the input data files and prepared the possible link to Microsoft Excel, DESERT can be invoked by double clicking its icon or by any other way guaranteed by Windows. The DESERT window and the Terminal window will then appear. The former, along with all the available commands, menus, and tools, is described in **Chapter 4**, while the latter is described along the **Chapters 3.5, 3.6, and 3.7**.

When the user wants to define a new riverine system, change the methods of hydraulic and water quality simulation of a previously defined system, or when the input data file have been changed, he (or she) must open the Universe file describing the input data files. This operation can be done through the **Open** command of the **File** menu (see **Chapter 4.2**).

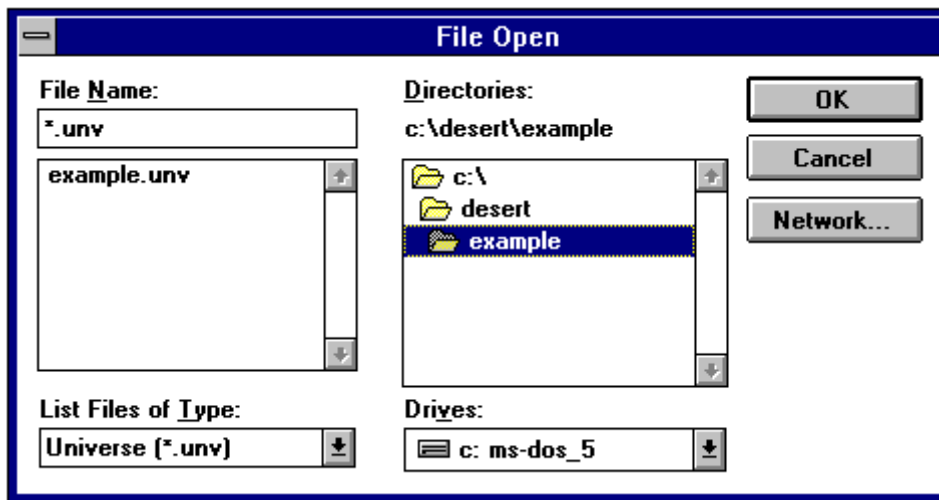


Figure 3.5.1 Opening the example.unv.

When opening a .UNV file, DESERT organizes all the information sparse in the MasterTable, InfoTables, MapTables, ProfileTables, and AlternativeTables to eventually produce a map of the system and a symbolic representation of the objects. All the operations DESERT performs to create this representation are echoed on the Terminal window. Possible errors in reading the input data file will be also indicated on this window (see **Chapter 4.2**).

To open the **example.unv**, you must select in the **Open file** dialogue box the directory **desert**, and the subdirectory **example** (figure 3.5.1). You will notice that in the representation of the **example.unv** a Reach cross section object is associated to every structural object, although no explicit declaration of these objects is present in the **profexe.dbf** file (see **Section 3.3.4**). The fact is that, in order to determine boundary conditions, a cross section is automatically generated in correspondence of each structural object by linear interpolation of the two nearest cross-sectional profiles of the same branch (see **Chapter 1.1**).

Selecting an object through the mouse and double clicking on it introduces a dialogue box representing all the information available on that object (see figure 1.2.3 for the dialogue box of a Point source object and figure 3.5.2 for that of a Reach cross section). Through these dialogue boxes the user might change the value of the data. However, changes will not affect the input data files, but only the internal representation of data created by DESERT. When the Universe file is closed, changes will vanish.

The Universe file representation has not only the passive role of showing the available information on the system, but also it is fundamental in selecting the parts of the system to be simulated and the methods by which performing the simulation. As introduced in **Sections 1.2.3 and 1.2.4**, one of the features that make DESERT an innovative software tool in environmental modeling is the possibility to use different solution methods for hydraulic and transport equations in different parts of the system. These methods are chosen by double clicking on any single reach of interest. The dialogue box of figure 3.5.3 will then appear for each reach that is double clicked.

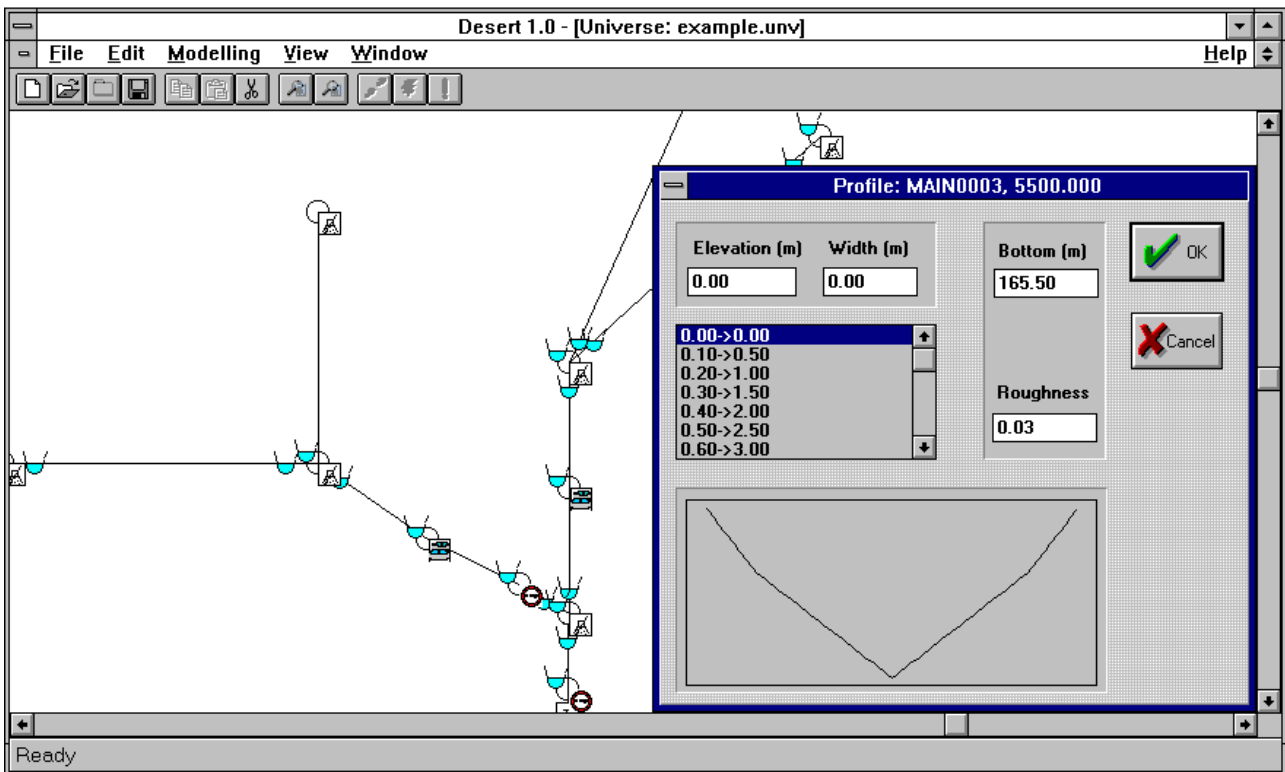


Figure 3.5.2 The dialogue box illustrating the cross-section MAIN0003 of the example.unv.

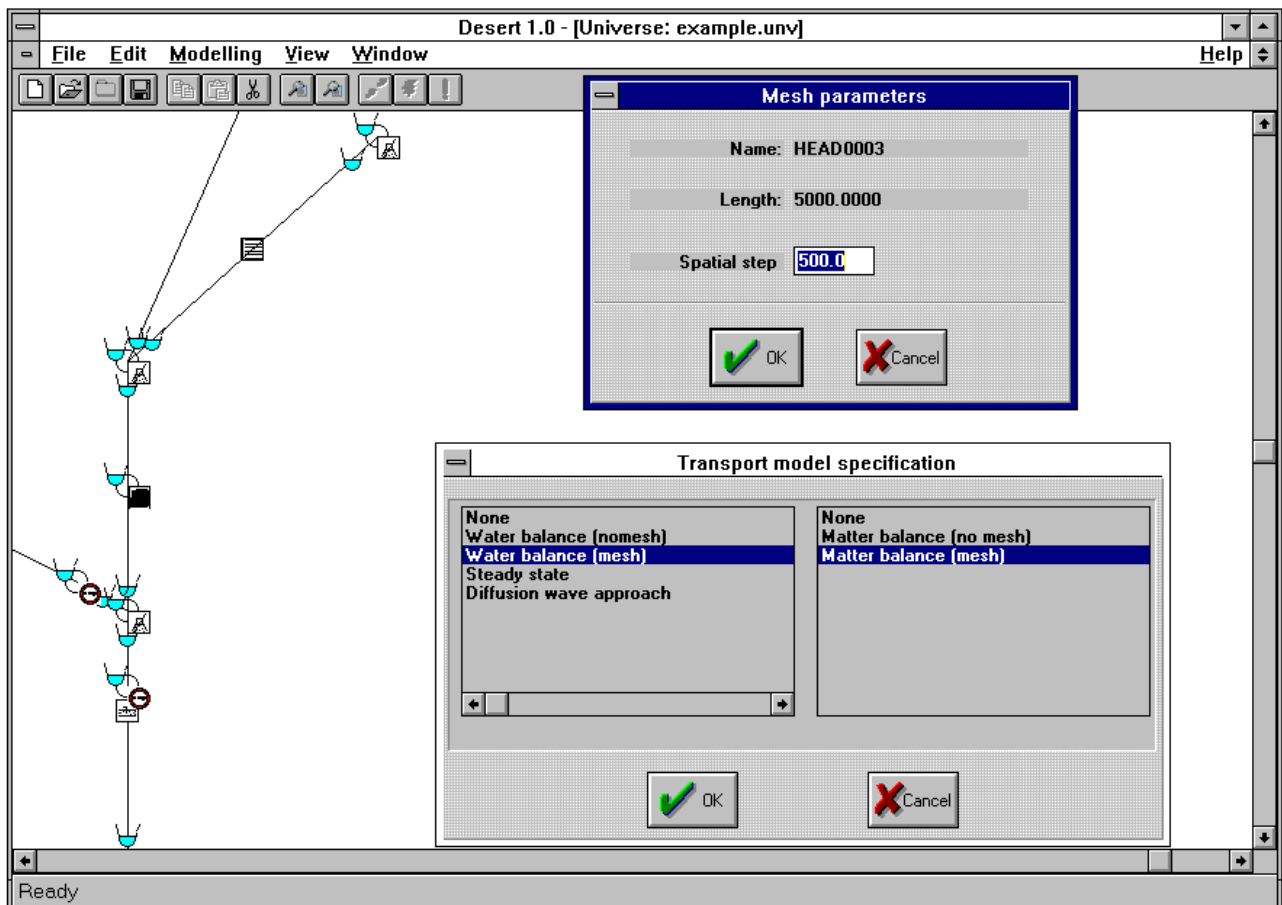


Figure 3.5.3 The dialogue box used for determining the method for carrying out the hydraulic and water quality simulation

One of the first three options is chosen for the hydraulic equations (below). If any of the chosen solution method uses a mesh, then a further dialogue box is issued requiring the size of the mesh elements (above).

Five methods for solving the hydraulic equations are available (for a theoretical explanation of the methods, see **Section 1.2.4**).

- None
- Water balance (No mesh)
- Water balance (Mesh)
- Steady state solution
- Diffusion wave solution

When any of the last three methods is chosen, a further dialog box will appear asking for the distance between consecutive mesh points (see **figure 3.5.3**). The user will then specify the desired length. If the specified distance is not an exact divisor of the total length of the reach, DESERT will use as mesh size a value of its own choice. This value (which is possible to see by reopening the dialogue box) is determined according to the following scheme:

- the number of segments desired by the user is determined by dividing the reach's length by the size expressed by the user;
- the mesh size used by DESERT is then set to the reach length divided by the number of mesh segments determined at the first step. The final size will appear on the dialogue box whenever it is opened.

Although DESERT has not been designed for this purpose, it allows the simulation of hydraulics alone. In this case, there is no need to choose any solution method for the mass transport equations.

For modeling matter transport phenomena, six different approaches are available (for a theoretical background, see **Section 1.2.4**).

- None
- Matter balance (No mesh)
- Matter balance (Mesh)
- Matter kinetics (No mesh)
- Matter transport (Steady state solution)
- Matter transport (Dynamic solution)

The choice of the method for performing hydraulic simulation influences also the choice of the method to solve the transport equations. If the None or the Water balance (with meshing or without meshing of the reach) methods are chosen, then, on the panel of water quality methods, only None and Matter balance (with meshing or without meshing of the reach) will appear (**figure 3.5.3**).

On the other hand, by choosing the Steady state or the Diffusion wave approach for the hydraulics

, the possibility to use Matter kinetic and Matter transport (either steady state or dynamic solution) will be also offered (**figure 3.5.4**).

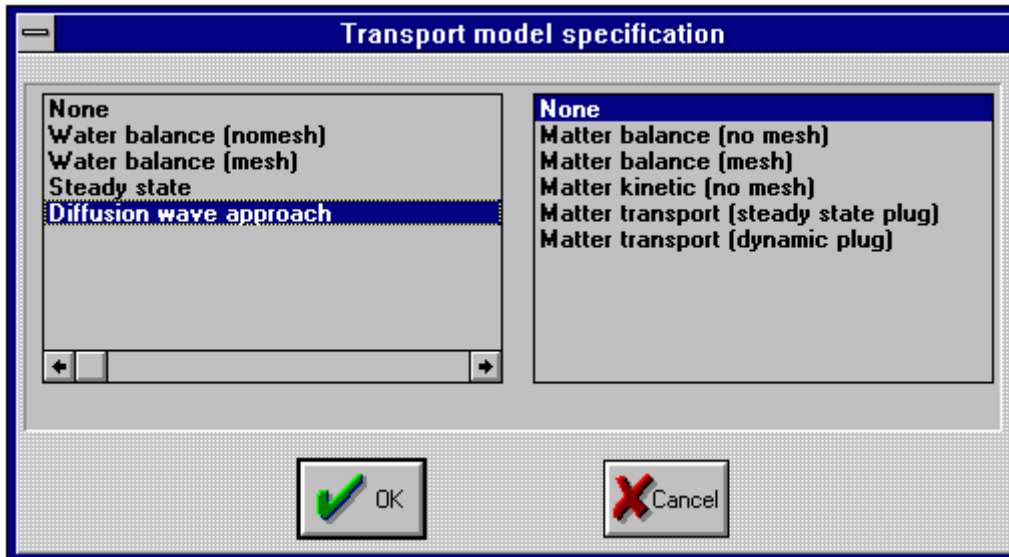


Figure 3.5.4 The Transport model specification dialogue box as it appears when Steady state or dynamic wave approach is chosen.

In defining the methods for solving hydraulic and water quality equations the user must not mix mesh and no mesh methods with each other. This means that the Water balance (No mesh) method can not be used if different hydraulic methods are used for other reaches of the network of interest. Also, if the Matter balance (no mesh) is used for a reach of the network of interest, than only this method and the Matter kinetics (No mesh) method can be used in the other reaches of the network (the symmetric rule is valid for the Matter kinetics method). On the other hand, it is possible to use meshed methods for hydraulic simulation in combination with non-meshed methods for water quality simulation (e.g., the Steady state hydraulic formulation can be combined with the Matter kinetics and Matter balance (No mesh) methods).

The definition of the hydraulic and transport solution approaches is necessary only for the reaches that are of interest in the particular moment and not for all the system.

After this operation has been made, the part of the system that is the object of the simulation needs to be indicated to DESERT. This selection is carried out through a graphical method by using the **Select Rivernet** command (see **Chapter 4.3**). Selected reaches need to be connected and form a sub-tree of the system. They can not be isolated from each other, otherwise an error in the creation of the Rivernet file will be caused. In this case, a message will appear on the Terminal window advising that DESERT supports only one Rivernet.

Coming back to the example, you should now double clicking on all the reaches and setting the hydraulic and the water quality methods to Steady state approximation. The mesh size for all the reaches should be set to 500 m.

For the Tertiary river branch (i.e., the one between the objects CONF0003 and HEAD0004) it is possible to use only the Water and Matter balance (No mesh or Mesh) method. This is due to the fact that no profile section is present in that reach. The methods for solving hydraulic equation contemporarily present on a network must be either of the “No mesh” type or of the “Mesh” type. No mixing is possible. Since in the other reaches the Steady state approach is used, you must adopt the Water Balance (Mesh) method for the Tertiary river. The same logic must be applied to the water quality methods. Therefore, you must select the Matter balance (Mesh) method on this reach. If you try to mix the two kinds of methods, you will not be able to paste the selection on the Rivernet file (see **next Chapter**) and the error “Inconsistent mesh and no mesh sections!” will be issued.

After having defined hydraulic and water quality methods for every reach, you must use the **Select Rivernet** command and select all the network (**figure 3.5.5**).

After this, the command **Copy** (see **Chapter 4.3**) must be given.

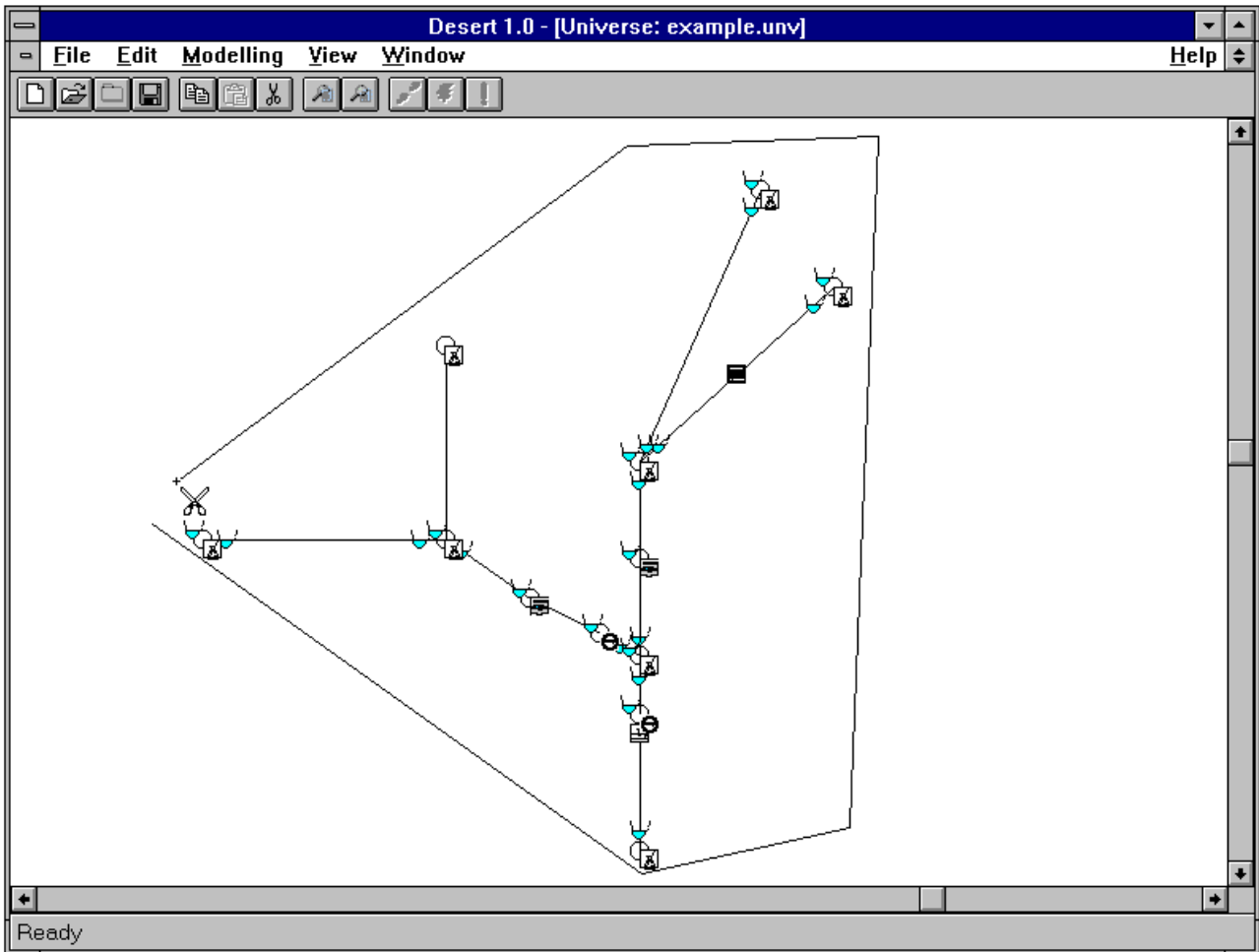


Figure 3.5.5 Selecting all the network of the example.unv file.

3.6 DEFINING A RIVERNET FILE

If the user opened the Universe file and selected a river network as explained in the previous chapter, he or she must issue a **New** command (see **Chapter 4.2**). As a result of this action, an empty window named “DESERT 1.0 - [Rivernet: river0001.RVR]” will appear. The user must now paste the previously selected network on this file by using the **Paste** command (see **Chapter 4.3**). The selected network will then appear on this window.

The Rivernet file defined in this way contains all the information that DESERT needs for carrying out the hydraulic and mass transport simulation. This file can be saved through the **Save** and **Save As** commands of the **File** menu (see **Chapter 4.2**).

When a Rivernet selection is saved as a .RVR file, the reaches and the parameters relative to the simulation procedures can be recalled directly without passing through the process of opening the Universe file, selecting the reaches, defining the simulation methods, and so on. The recall operation can be made simply by the **Open** command (see **Chapter 4.2**). Clearly, it is possible to open the Universe file, select a river network, open an existing Rivernet file, and paste the selection on this file. The previous content of the Rivernet file will be temporarily erased and substituted with the pasted data. This substitution will become permanent if the file is saved without changing its name.

When pasting a selection it may happen that the objects along a reach appear, but the link itself misses. This means that the user forgot to define the hydraulic method for that reach. In this case, the user must go back to the Universe file, define the hydraulic simulation for the missing reaches, and select, copy, and paste the area of interest again.

In the .RVR files, information stored in the objects are accessible and changeable as for the .UNV file. This is very useful in testing hypothesis regarding changes in the input data, without having the trouble to change the input data files. On the contrary of what happens with the Universe file, here changes data are saved when the Rivernet file is saved. However, changes can only affect input data. If the methods for performing hydraulic and water quality simulation need to be changed, then it is necessary to open the relative Universe file and repeat the operation described in the previous chapter. For this reason, double clicking a reach in the Rivernet file does not produce the same dialogue box that is produced in the Universe file. It will appear only a text box reporting the methods and the mesh size used in the reach (**figure 3.6.1**).

By double clicking on any point of the white background of the .RVR file, a dialogue box will appear. It reports the time settings for the simulation of the hydraulic and water quality model (**figure 3.6.2**). The Current date and the Current time refer, in this case, to the initial time and date of simulation. Finish date and time of simulation are also reported as well as the time step size in minutes. This data can be changed by the user by writing on the relative windows.

Since the example is set for Steady state simulation, you need only to save the model through the **Save as** command of the **File** menu. Since the reaches of this Rivernet file have been set for hydraulic and water quality steady state simulation, a good name for the file could be **hysswqss.RVR**

3.7 DEFINING THE WATER QUALITY MODEL AND RUNNING THE SIMULATION

In order to run both the hydraulic model and the water quality model, the user, after having obtained a .RVR file by creating it or by opening it, needs to switch to the **Terminal Window**. Up to now, this window has played a passive role, being only the place where DESERT reported messages about errors or about the state of the process. At this point of the modeling procedure, this window assumes an active role, becoming the interface between DESERT and the user for the definition of the water quality transport equations, the execution of the model, and the display of simulation results.

The language used to accomplish these tasks is called MODUS. It is conceptually similar to BASIC. **Chapter 5** explains in detail the glossary and syntax of this language. DESERT interpreters and executes the commands written by the user on the Terminal window. It also reports possible errors in the language syntax and in the execution of the simulation on the same window.

To get familiar with this language and the general way DESERT works, you should begin to test the hydraulic simulation of the **hysswqss.RVR** file.

To do this, type the word **init**.

WARNING

DESERT is very “case sensitive”. If you have typed INIT, DESERT will signal an error of “Invalid command”. All the key words (see **Chapter 5**) of the MODUS language are in lower case letters. The case sensitiveness affects any letter written in the Terminal window. For example, the variables DO, dO, Do, and do are four distinct entities.

Even if you wrote the command **init** correctly, you will not be probably happy. Several lines saying “Warning! Bottom level distinguished at point: 13000 Section: HEAD0002.” have appeared on the Terminal window.

These warnings testify that some instability has occurred during the hydraulic simulation. In this particular case, DESERT computed a negative depth during hydraulic simulation. To continue the simulation, this depth was substituted with the value obtained by using the uniform flow formulation. This error is not maybe very important in steady state simulation, but it may become a mess during dynamic simulation. Fortunately, the solution to this problem is often easy: the user has to reduce the mesh size of the reach where the warning occurred and (if the case) the time step. Playing with slope and water discharge is a further possibility if the previous options did not succeed.

Therefore, you should go back to the **example.UNV** window, double click on the two upper reaches of the SecondA river, and reduce their mesh size to 100 m. After this, you have to select the network again, copy it, and paste it to the **hysswqss.RVR** file.

Turn again to the Terminal window and retype **init**. You will see a message advising you that the simulation finished successfully.

You might relief, but probably also wonder how to see simulation results. **Chapter 5.*** lists the hydraulic variables that are supplied by DESERT. To see for example the spatial distribution of depth, you have to type:

```
variable depth
type depth
```

Spatial distribution of water depth will then be printed on the Terminal window. It is printed reach by reach in correspondence of the mesh points.

By declaring the fields of the InfoTable as variables (e.g., **variable CBOD**) and printing them you can also have a linear interpolation of the input data.

Typing directly on the Terminal window can be convenient when the user gives execution commands (like **init**, **print**, **plot**, etc.) or defines very simple water quality models. On the other hand, when models become complex, writing them every time is a really slow activity, where errors can be very frequent. Also, the models written on the Terminal window can not be saved directly. It is only possible to save the content of the Terminal window as a text

file (**Save** and **Saves as** commands of the **File** menu). Working on this file it is possible to rescue the model, but it is not a very effective way of operating.

For solving this problem, DESERT allows the user to write the models in a text file through any text editor. The text file containing the model must be saved as a Model file (extension .MOD). Then it can be imported to DESERT. Model files can be imported only when the Terminal window is active by the **Import** command of the **File** menu (see **Chapter 4.2**). The model contained in the file is read, interpreted, and echoed on the Terminal Windows line by line. After that one line of the model has been read and echoed on the Terminal window, it is interpreted by DESERT. If this finds any syntax error, it will produce error messages on the lines below.

It is time to give more taste to the example we are using. Therefore, you should go to the Terminal window and import from the directory **C:\desert\example** the file **bodnsod.MOD**. This file is the description in the MODUS language of the model described in **Chapter 3.2** (see **figure 3.7.1**).

```
#      External variables declaration

#      Hydraulic variables declaration

var depth
var flow
var width
var crossect          #Cross sectional area

#      Input Data variables

var T                #Temperature
var DO               #Dissolved Oxygen
var CBOD             #Carbonaceous Biological Oxygen Demand
var NBOD             #Nitrogenous Biological Oxygen Demand

#      Constants declaration

daysec=84600.0 #Number of seconds in one day
Kr=0.8/daysec  #CBOD removal rate (1/s)
Kn=0.8/daysec  #NBOD removal rate (1/s)
ka0=2.0/daysec #Reareation coefficient (m/s^0.5)
Ksod=1.5/daysec #Sediment Oxygen Demand (g/m^2/s)

#      Local variables declaration

subst u=flow/crossect          #Velocity (m/s)
subst ft=1.016**(T-20.0)      #Temperature dependence of reaeration
subst ka=ka0*ft*sqrt(u/depth) #Oxygen exchange at water-air interface.

subst Osat=14.652-0.41022*T+0.00799*T**2-0.000077774*T**3
                               #Saturation oxygen concentration

#      Initialization of components (setting of boundary conditions)

component L=CBOD              #Carbonaceous Biological Oxygen Demand
component OX=DO               #Dissolved Oxygen
component N=NBOD              #Nitrogenous Biological Oxygen Demand

#      Equations of DO-BOD reaction schemes

equation L=-Kr*crossect*L     #Exponential decay
equation N=-Kn*crossect*N     #Exponential decay

equation OX=ka*width*(Osat-OX)-Kr*crossect*L-Kn*crossect*N-width*Ksod
                               #Reaeration - CBOD consumption -NBOD consumption - SOD consumption
```

Figure 3.7.1 The model described used as example (**bodnsod.MOD**).

After you have done this, turn again to the Terminal window and type **init**. When you see the end of simulation message, you can start to print all the external variables and components used by the model you are interested on (for example OX, the dissolved oxygen).

From **figure 3.7.1**, it is already possible to understand some of the features of the MODUS language:

- any external variable (like flow, depth, CBOD, etc.) used in the model formulation must be explicitly declared;
- Constants must be declared with statements of the kind:

<identifier> = <expression>

- Local variables must be declared with the statement of the type:

subst <identifier> = <expression>

Local variables are not subjected to the transport equations, but have a spatial and temporal dynamics instantaneously depending on constants, external variables, other local variables, or components.

- Components (i.e., the variables subjected to the transport equations) must be declared with a statement of the type:

component <identifier> = <expression>

The component assumes as boundary values the result of **<expression>**. The expression can have as arguments constants, external variable, and local variables. However, the user must take care to avoid any reference, even indirect, to other components.

- The reaction scheme of transport equations must be declared with the statement:

equation <identifier> = <expression>

where the arguments of the expression can be constants, external variables, local variables, and components.

The Model file could also contain the execution commands. However, avoiding to write the execution commands in the Model files is probably safer until the model has not been conveniently tested.

You might probably wonder what is the convenience of having the possibility of writing in the Terminal window the model definition, given the fact that it is better to write complex models with a text editor. Or even worse, what is the convenience of having an interpreted language instead of a more traditional compiled procedure.

To understand the productivity of this approach, you could write the following statements:

```
ka=2.316*(u**0.67)/(depth**0.85)
init
```

In this way the model **bodnsod.MOD** seen before as been changed replacing the O'Connor and Dobbins's formulation for the reaeration coefficient with the formulation of Owens et al. (1964), and it has been run. If you print the component OX, you can see the difference in the results.

In BASIC, you run a program and see the results. To make a change, you need only to rewrite the lines interested by the change and run the model again. Here it is the same, to make a change you need only to rewrite (or add) the instructions you like on the Terminal window. That is all: no editing, no compiling, no linking.

3.8 PLOTTING RESULTS

As already mentioned in **Chapter 3.4**, to save results to a spreadsheet, a large Excel worksheet needs to be created. This worksheet must be linked to DESERT for transferring results from DESERT to it. This operation is accomplished by pressing the **New** command from the Rivernet file window (see **Chapter 4.2**).

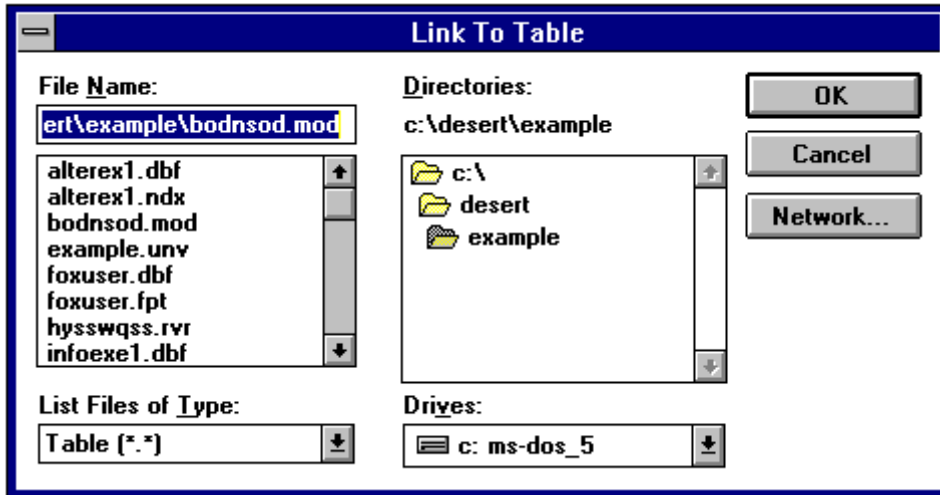


Figure 3.8.1 The Link to Table dialog box.

As a result of this operation, a typical openfile dialogue box will appear (**figure 3.8.1**). The title of the box is “Link To Table”. In the “List Files of Type:” box, the item “Table (*.*)” is present. This implicates that all the files of a directory are displayed in the box, independently from their file type. At this point the user needs to select the file created previously (see **Chapter 3.4**). Consequently, a dialogue box entitled “Insert Object” will appear (**figure 3.8.2**). A choice of object types is here available. Anyway, the user must choose the “Microsoft Excel 5.0 Worksheet” or the “Microsoft Excel Worksheet” option.

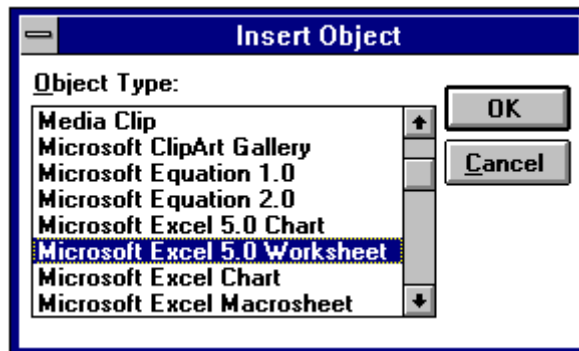


Figure 3.8.2 The Insert Object dialog box.

After some seconds, a window entitled **Table001.TAB** (the number of the table depends on how many links have been established in the session) will appear. In the window the Excel file defined earlier by the user appears in a “Clipboard” style representation. To establish the link between Excel and DESERT, the user must double click inside this window. As a result of this, the Excel window of the Excel file the user defined earlier will appear. Every time the command **Plot** will be used while running the model, the values of the argument of this command will be automatically saved in this Excel file. It is important to remember to save the data with a filename different from the original one (thus using the Excel command **Save as**). In this way, the format of the large file used for exchanging data will be saved. On the contrary, if data are saved with the original file name (using the Excel **Save** command), the format will be lost and a new large file will need to be constructed the next time saving data on a spreadsheet is required.

3.9 OPTIMIZATION PROCESS

The ultimate objective of developing DESERT was to produce a software capable of optimally allocating waste water treatment plants. The method implemented in DESERT for achieving such allocation is the dynamic programming. **Section 1.2.7** illustrates the theoretical basis of the method and its implementation in DESERT. This chapter explains the procedure to work with the optimization procedure.

First of all, it must be clear that there is no theoretical limit on the number of treatment plants to allocate, the number of alternative solutions for each plant, and the number of constraints. However, if each of the treatment plants present on the system has one alternative only, then there is only one feasible solution. In this case, DESERT will determine only if it is compatible with the given constraints.

At the other extreme of the possible situations, there could be tens of treatment plants, each having several alternative implementations. In this case, if there are few constraints or if they are very loose, then the number of feasible alternatives grows dramatically. Although theoretically DESERT should be able to deal with the situation, the probable outcome would be a crash of the system due to the consumption of all the available memory. The boundary number of treatment plants, alternative implementations, and constraints depends on the hardware (i.e., processor type, RAM and hard disk space available, etc.) and software (i.e., number of tasks running concurrently, virtual memory size, etc.) configuration of the computing system.

Another factor to take into account in designing the optimization scheme is that the hydraulics in the riverine system is computed only once during optimization. The clear advantage of this choice is that the time for performing the optimization is strongly reduced. The drawback is that plants must have the same discharge in all the alternative schemes. This implicates that it is not possible to allocate directly position and size of effluent discharges, but only their quality. It is possible to overcome the problem by optimizing system representations where the treatment plants have different positions and then comparing the optimal solutions. However, this operation requires to set several configurations of the riverine system, optimize each of them, and compare the respective optimal solutions.

For performing the optimization, the user must specify the objective function and the possible constraints. Very likely, the objective function will have as arguments some kind of “cost” parameters associated to the possible alternative configurations of the treatment plants. For each water treatment plant several alternative configurations could be implemented. For each of them, the user must specify the “cost” parameters in an AlternativeTable file and the resulting quality of the effluent discharge in an InfoTable file. These two records are linked to each other by the values present in their CodeField and VariantField fields (see **Sections 3.3.5 and 3.3.6**).

In the case more than one AlternativeTable file is used, the name of the fields containing the “cost” parameters must be the same.

In order to perform the optimization DESERT needs to know which fields of the AlternativeTable file are used as arguments of the objective function. This operation is accomplished by the statement:

property <identifier>

where <identifier> is the name of one field of the InfoTable containing one “cost” parameter. Such statement must be issued as many times as are the “cost” parameters.

When this operation is done, the objective function must be specified by the statement:

objective <expression>

where <expression> is an expression having as possible arguments the “cost” parameters specified by the **property** statements and the state variables. DESERT will allocate plants aiming to minimize the value of <expression>.

Water quality constraints must be expressed by statements of the type:

constraint <identifier> = <expression>

where <identifier> is the name given to the constraints and <expression> is an expression that will be evaluated as a C++ Boolean expression. This means that if <expression> has the value zero, then it is considered as false; it is considered as true otherwise.

To have an example of the optimization procedure, you should open the file **hysswqss.RVR**, go to the Terminal window, and import the file **bnsopt.MOD** from the directory **desert\example**.

```

#      External variables declaration

#      Hydraulic variables declaration

var depth
var flow
var width
var crossect      #Cross sectional area

#      Input Data variables

var T      #Temperature
var DO      #Dissolved Oxygen
var CBOD      #Carbonaceous Biological Oxygen Demand
var NBOD      #Nitrogenous Biological Oxygen Demand

#      Constants declaration

daysec=84600.0      #Number of seconds in one day
Kr=0.8/daysec      #CBOD removal rate (1/s)
Kn=0.8/daysec      #NBOD removal rate (1/s)
ka0=2.0/daysec      #Reaeration coefficient (m/s^0.5)
Ksod=0.5/daysec      #Sediment Oxygen Demand (g/m^2/s)

#      Local variables declaration

subst u=flow/crossect      #Velocity (m/s)
subst ft=1.016**(T-20.0)      #Temperature dependence of reaeration
subst ka=ka0*ft*sqrt(u/depth)      #Oxygen exchange at water-air interface.

subst Osat=14.652-0.41022*T+0.00799*T**2-0.000077774*T**3
      #Saturation oxygen concentration

#      Initialization of components (setting of boundary conditions)

component L=CBOD      #Carbonaceous Biological Oxygen Demand
component OX=DO      #Dissolved Oxygen
component N=NBOD      #Nitrogenous Biological Oxygen Demand

#      Equations of DO-BOD reaction schemes

equation L=-Kr*crossect*L      #Exponential decay
equation N=-Kn*crossect*N      #Exponential decay

equation OX=ka*width*(Osat-OX)-Kr*crossect*L-Kn*crossect*N-width*Ksod
      #Reaeration - CBOD consumption -NBOD consumption - SOD consumption

#      Optimization definition

constr CONST1=((OX>DO)&&(L<CBOD)&&(N<NBOD))      #Water quality constraint
property COSTTOT      #Operation cost parameter
objective COSTTOT      #Objective function

```

Figure 3.9.1 The bnsodopt.MOD file.

This file is the same water quality model presented in file **bodnsod.MOD** with the optimization specifications. Notice that the constraint asks the dynamics of the dissolved oxygen (OX component) to be greater than the value of DO and the carbonaceous and nitrogenous BOD (L and N components) to be less than the values of CBOD and NBOD. These inequalities must hold only at the two Water quality constraint objects. The values of DO, CBOD, and NBOD are the ones specified in the **infoexe1.DBF** file for the objects WQCO0001 and WQCO0002.

To run the optimization you must now digit the command

optimize

You will see on the screen the optimization proceeding until the optimal solution is reached. The optimal solution is given by implementing the BCDN alternative at the treatment plant EFFL0002, but only a B alternative at the treatment plant EFFL0001. This is likely due to the fact that the EFFL0002 is very near to the constraint point WQCO0002, while the distance between EFFL0001 and WQCO0001 is greater, thus allowing a better BOD removal.

To understand the power of the interpreter language approach and the versatility of the optimization part, digits the statements

objective L

optimize

Very easily, we have started a new optimization procedure, which tries to minimize the value of L along the system. Not surprisingly the optimal solution requires to implement the BCDN alternative for both the treatment plants.