QUALHYMO USER MANUAL AND DOCUMENTATION

Version QUALHYMO0777v1c

January, 2009

QUALHYMO0777v1c User Documentation

Foreword

This is a concise manual that documents critical elements of a development version of the QUALHYMO model. It identifies a number of features that differ from earlier versions of the model, as well as some of the basic details of model input and operation. Users making reference to this document should verify that it corresponds with the model version they are using; after February 2009 users should refer to http://qualhymo.watertoolset.com for an updated manual (and corresponding model) if available.

License

Below is a version of the license that is current as of the date this release of the manual was developed. Users are cautioned that there may be updated or different versions of the manual on the web site from which these are downloaded.

1.0 Background

1.1 An Overview of QUALHYMO Evolution

QUALHYMO was developed some 25 years ago as a research tool. It was intended to enable the rapid testing of various water quality algorithms related to BMP performance assessment. For that purpose it was decided to create a modular tool in which different sub-systems could be rapidly and easily coded, implemented, and tested on a common basis. The original QUALHYMO tool was developed with that in mind. The selected structure for this has proven to be robust and reliable, and lends itself well to use in assessing watershed level water quantity and quality problems.

Over the 20 years or so following the initial release of the model, the business activities of the original author tended to conflict with further development of the tool, and no substantive development was attempted by him during that period. Nevertheless, a number of descendant versions or variations of QUALHYMO seem to have emerged over that time. The code to the original model was available to developers and users, and it seems that an undetermined number of alternative versions of QUALHYMO have appeared as a result. No endorsement (or otherwise) of such alternatives will be attempted by the authors, and should not be interpreted from this manual. However it is noted that a casual survey suggests that these versions vary in their approaches to coding and simulation. It is not known to the authors of the present QUALHYMO effort how those multiple versions fared in practice, but discussions with individuals in the field suggested that the multiplicity of versions has led to a multiplicity of issues, not the least of which was a lack of confidence regarding which version of the tool was which.

As the practice of water resources has matured, the need for a tool like QUALHYMO for practical applications has been reinforced. A model targeted at BMP evaluation and designed for rapid simulation of long term quality/quantity behavior has emerged as a requirement in many situations. The decision was made, therefore, to update QUALHYMO into a single current version.

This re-development posed numerous challenges, not the least of which was the need to provide an upward migration path so that users accustomed to certain features could still have access to them when the new QUALHYMO was released. To the extent possible, upward compatibility has been preserved. As a part of this, variable names exposed to the user have for the most part been preserved (at least in the user manual). The underlying code is another matter. An immediate effort in developing current version was to retain no code that is not directly attributable to the identified authors. This means that the authors can release the model for general use under suitable license conditions. It also means, however, that the model may in some instances produce results that are somewhat different from those other variations of the earlier tool may generate. QA/QC procedures used in developing this tool have been substantial, and we believe that it can be relied on to produce reasonable results when applied by a knowledgeable user, but as noted in the license for the model and documentation, it is up to the user to ensure that the model is effective for the purpose for which they apply it and that the answers they generate are what they require.

2.0 Installing the Model

The model is provided in the form of a .zip file, 'installset.zip'. This can be downloaded from the http://qualhymo.watertoolset.com web site. To install QUALHYMO, place the zip file in any location that you prefer, and unzip the file. It is then ready to run. It is not necessary to install the software otherwise.

> Hint: Installed this way, the model can be run in its default configuration, and this is a good way to begin exploring the model and its use. However, it should be noted there are numerous other ways the model can be installed and run, at the discretion of the user. Appendix A provides details on how the model can be installed in ways that differ from the default case.

When the file is unzipped, you will see a folder 'InstallSet' appear in the directory you have selected as the unzip location. Depending on where you unzipped the model, that might look as follows:

Name	Size	Type	Date Modified
InstallSet		File Folder	10/14/2007 6:49 PM
InstallSet.zip	870 KB	WinZip File	10/14/2007 12:51 PM

If you look inside the InstallSet folder, you'll see the following:

Name	Size	Type	Date Modified
License		File Folder	10/14/2007 7:06 PM
Manual		File Folder	10/14/2007 11:34 AM
TestData		File Folder	10/14/2007 7:07 PM
QCONTROLFILE.TXT	1 KB	Text Document	10/14/2007 12:49 PM
QUALHYMO0777.exe	1,032 KB	Application	10/14/2007 11:27 AM

The three folders in the InstallSet folder are as follows:

- The folder 'License' contains the license under which QUALHYMO and this manual and associated documents must be accessed and used. Read it, and if you don't agree with it, don't use either the model or the manual further.
- The folder 'Manual', contains this manual or one like it.
- The folder 'TestData' contains a test data set that can be used to verify that the model is correctly installed and functioning.

Testing is described further below, and model use is described further in Chapter 2.

For now, you should test the model. There are two ways to do this:

- Double click on the QUALHYMO0777.exe file name in Windows Explorer, or else
- open a command window, navigate to the location of QUALHYMO0777.exe and enter the command 'QUALHYMO0777' followed by the return key.

If you used Windows Explorer and double clicked to run the model, you should see a command line window pop up as follows and then after a few minutes disappear:

```
C:\Data\TestLocation\InstallSet\QUALHYMO0777.exe
 Portions of this program include material copyrighted (c) by Absoft Corporation
 1988-2001
 THIS RUN WILL USE CONTROL AND LOG FILES: QCONTROLFILE.TXT AND QRUNLOG.TXT
 POND   200  CALLED
```

If you ran the model from the command line, you should see something similar to the example below, but the window will remain in place after the run has completed:

```
Command Prompt
C:\Data\TestLocation\InstallSet>QUALHYMO0777
 Portions of this program include material copyrighted (c) by Absoft Corporation
 1988-2001
 THIS RUN WILL USE CONTROL AND LOG FILES: QCONTROLFILE.TXT AND QRUNLOG.TXT
 POND   200  CALLED
0

C:\Data\TestLocation\InstallSet>
```

Either way, if you look in the directory in which the model was installed, you'll see the following files:

Name	Size	Type	Date Modified
License		File Folder	10/14/2007 7:07 PM
Manual		File Folder	10/14/2007 7:07 PM
TestData		File Folder	10/14/2007 7:22 PM
COMMAND.TBL	1 KB	TBL File	10/14/2007 7:22 PM
PONDRESULTS.BIN	14,967 KB	BIN File	10/14/2007 7:22 PM
QCONTROLFILE.TXT	1 KB	Text Document	10/14/2007 12:49 PM
QRUNLOG.TXT	2 KB	Text Document	10/14/2007 7:22 PM
QUALHYMO0777.exe	1,032 KB	Application	10/14/2007 11:27 AM

The new files that have appeared indicate that the model has been correctly installed and correctly run. If you look further, in the file QRUNLOG.TXT, you'll see a series of diagnostics that will end with the following lines:

```
=== POND COMMAND
  - RETURN FROM COMMAND INTERPRETER, NER,NCODE:  0  22
=== TIME STAMP Sun Oct 14 19:22:35 2007===
=== POND STATS COMMAND
  - RETURN FROM COMMAND INTERPRETER, NER,NCODE:  0  23
=== TIME STAMP Sun Oct 14 19:22:35 2007===
=== IDENTIFY SERIES COMMAND
  - RETURN FROM COMMAND INTERPRETER, NER,NCODE:  0  23
=== TIME STAMP Sun Oct 14 19:22:35 2007===
=== IDENTIFY SERIES COMMAND
  - RETURN FROM COMMAND INTERPRETER, NER,NCODE:  0  21
=== TIME STAMP Sun Oct 14 19:22:35 2007===
=== RUN ENDED NORMALLY Sun Oct 14 19:22:35 2007===
```

The absence of messages indicating problems confirms that the model has indeed been correctly installed and run.

3.0 Running the Model

QUALHYMO is invoked from the command line of a Microsoft Windows operating system, by typing the command 'QUALHYMO0777' in a command line window or by double clicking on the 'QUALHYMO0777.exe' file from Windows Explorer or another suitable file manager. This version has been tested on Windows XP. Appendix A provides details on model control and ways that the model can be installed in other than the default configuration. The following discussion assumes you are operating in the default configuration. Section 3.2 below provides a discussion about alternative interfaces.

3.1 Input Files

If you look in the TestData folder, you'll see the following files if the model has not been run:

Name	Size	Type	Date Modified
TestEvap.evp	10 KB	EVP File	10/4/2007 1:20 AM
TestInput.inp	9 KB	INP File	10/14/2007 11:49 AM
TestPrecip.pre	1,560 KB	SQL Server Replicat...	10/4/2007 1:20 AM
TestTemp.tmp	908 KB	TMP File	10/4/2007 1:21 AM

or the following files if you did a test run with the model:

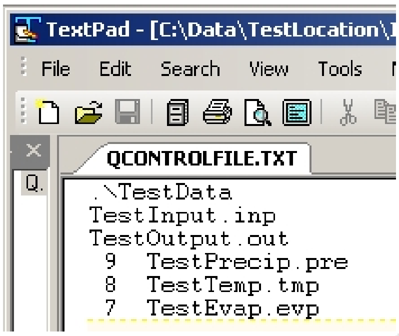
Name	Size	Type	Date Modified
QBIN1.BUF	15,619 KB	BUF File	10/14/2007 7:22 PM
QBIN2.BUF	15,619 KB	BUF File	10/14/2007 7:22 PM
TestEvap.evp	10 KB	EVP File	10/4/2007 1:20 AM
TestInput.inp	9 KB	INP File	10/14/2007 11:49 AM
TestOutput.out	77 KB	OUT File	10/14/2007 7:22 PM
TestPrecip.pre	1,560 KB	SQL Server Replicat...	10/4/2007 1:20 AM
TestTemp.tmp	908 KB	TMP File	10/4/2007 1:21 AM

You will not need to deal with the QBINx.BUF files unless you are an advanced user. The format of those files, however, is discussed in Chapter 4 of this manual if you are interested in their make-up and function. The file TestOutput.out is an output file generated by the model, and is where you'll find the results of a QUALHYMO run. The other files are:

- TestPrecip.pre – a rainfall record series,
- TestEvap.evp – an evaporation record series,
- TestTemp.tmp – a temperature record series, and
- TestInput.inp – an input deck that controls the model run.

The formats of the first three files is described elsewhere in this manual. They should be replaced with files appropriate for the model run you have decided to do. The fourth file is the file that controls what simulations are to be done, and is the way you control the model.

If you have renamed any of these four files, then you need to adjust a control file. That file is the QCONTROL.TXT file that is located in the directory where you unzipped the model. If you examine QCONTROLFILE.TXT created in the default install, you'll see the following:



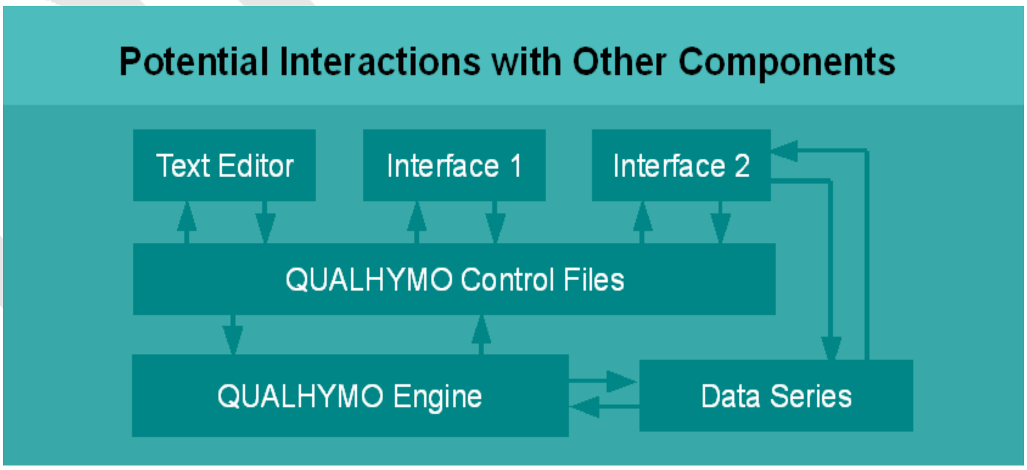
This file is how the model knows where to look to find the time series and input files that control a model run. If you have renamed any of the four files discussed above (TestInput.inp, TestPrecip.pre, TestTemp.tmp, or TestEvap.evp), you must adjust the names in the QCONTROLFILE.TXT so that it corresponds to the new file names you have chosen.

The TestOutput.out file name in this file can be changed to any valid file name, and the model will deposit its output results in a new file of that name.

More significant changes are possible by appropriate choices in the use of the QCONTROLFILE.TXT file. These are discussed in Appendix A.

3.2 Alternative Interfaces

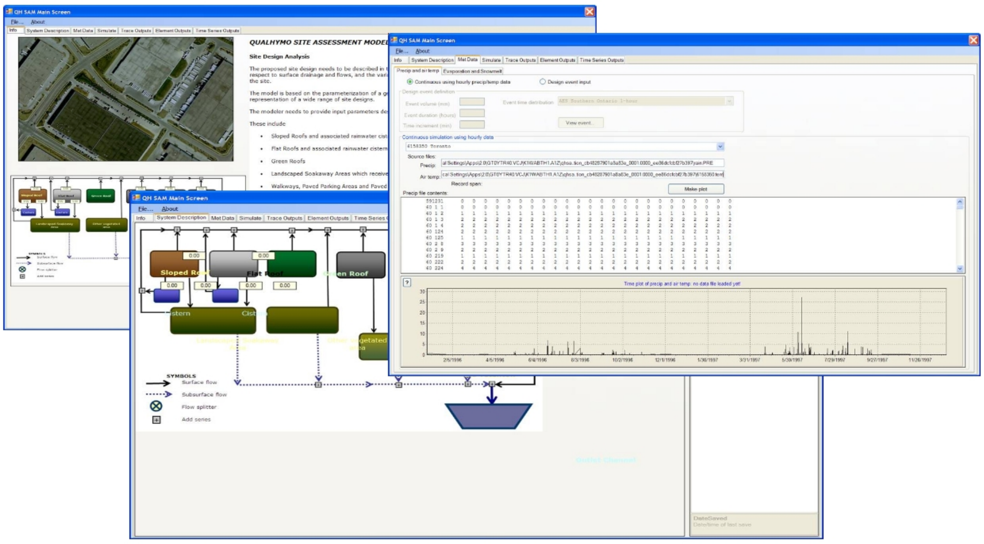
QUALHYMO is intended to function primarily as a computational engine. It is fully capable as a tool using the commands described in section 4 and input as discussed in section 3.1 above. In some communities the manipulation of text files is a preferred approach, and is held to be efficient and exact. However for many users the manipulation of text files in their raw form will seem to be awkward and a visual interface would be preferred. To support this kind of user, it is anticipated that alternative interfaces may be developed and will be available. As noted in the sketch figure below, the data set documentation makes it possible to develop interfaces that will feed off the QUALHYMO engine in several ways.



In the above example, the control files could be manipulated directly with any **text editor**, but two other possibilities also exist. In **Interface 1**, control files are edited using some preferred user interface. In **Interface 2**, the control files are manipulated, but the raw data seres themselves are also accessed. Present efforts in the development of QUALHYMO have been focused on the engine, and on making it easily possible to add new computational capabilities. In the future, there may be expanded efforts on interface. Development. In the mean time, several alternative interfaces are known to have been developed.

One, illustrated with some screen captures below, is the Water Balance Model. This is a fully developed interface made available through a web site that can be found at http://www.waterbalance.ca/waterbalance/home/wbnIndex.asp. The WBM provides a mature and capable interface that exposes many of the QUALHYMO engine capabilities. It functions much as Interface 1 in the figure above does – it writes data to and extracts data from the QUALHYMO text files, and relies on QUALHYMO to extract information from the model time series files.

Another interface that is not publicly available but that further illustrates the potential for interface development is illustrated below. This version is focused on a specific set of engine capabilities, and was developed using MIcrosoft Visual Basic as a stand alone local application. This tool not only edits the QUALHYMO control files, it operates directly on the model time series much as the **Interface 2** option described above does.



A third known interface is JAVA based, and therefore platform independent, and may be made available publicly in the near future.

Either way, the salient point is that there are numerous possibilities for interface development, and the intent that the QUALHYMO concept and documentation enable this kind of extension has been demonstrated.

4.0 Model Commands

4.1 An Overview of the Input File

QUALHYMO is command oriented. Commands and data are provided in the input file which is specified in the control file as discussed in section 3.0 and Appendix A. The model reads each command and its associated data and carries them out in the sequence in which they appear in the input file.

4.2 Input File Syntax

Users of earlier versions of QUALHYMO will probably be familiar with an input syntax in which the first 20 columns of every non-comment line is reserved for commands. This has changed. Following the old syntax rules will still work, but a new syntax has been implemented. The new syntax has more flexibility, and enables further future refinements. (It is noted for those familiar with old versions of code that the HONDO routine has been dropped in favor of an entirely new system.)

The input file can contain any printing character and blanks. The command parser extracts two kinds of information from the file:

- command names
- numbers

As well as this information, the user may if they wish add non-numeric text to the file to explain what the run is doing.

Command Name Rules:

Commands are entered by starting a new line, and entering the command name beginning in column 1. Following the command name are the parameters needed by that command. Continuation lines, if needed, are indicated by leaving a space in column 1.

Commands include a range of capabilities, including those listed below and some others, and can be entered in whatever number and order makes sense to the user to accomplish the model run. The model will run each command as it encounters it:

```
START
STORE
GENERATE
PRINT SPAN
PLOT SPAN
ADD SERIES
POND
REACH
CALIBRATE
POLLUTANT SERIES
SPLIT SERIES
DUMP PRINT
EXCEEDANCE
FULL POND SPAN
MAXMIN
PRINT SERIES
FINISH
CALC POND STATS
IDENTIFY SERIES
PRINT MODEL DETAILS
```

Numeric Value Rules:

Numbers must be entered in integer or floating point formats (including scientific notation in the form of #.###E+##).   Numbers must be entered following a space at the end of each command. Numbers not required for a command must not be entered in the file or they will be interpreted as data and the model will attempt to use them as such.

> Hint: Adding extraneous numbers in comments is a common cause of user errors, and sometimes occurs if a user adding a comment includes a number – for example a version number for the run. If this kind of input is desired, it must occur in a comment line as noted below.

Comment Rules:

There are two kinds of comments that are allowed in the model input file:

- ***Comment lines*** can be added. They can be added between commands and are identified by an asterisk (*) in the first column of the line. Comment text, including numbers, can be entered in columns 2 though 80, and can contain any content since they are ignored by the model.
- ***Comment text*** can be added. This can consist of any non-numeric text inserted after the command name and before, between or after the numeric values provided by the user.

The first few lines of a typical input file illustrate one correct use of the syntax rules:

```
*
*
*  *****************           QUALHYMO0777       **********************
*          ====     File: RunFile22.dat         ====
*          ====     Rain File:CAL-RAIN.PRE      ====
*          ====     Snowmelt FILE:CAL-TEMP.TMP  ====
***********************************************************************
START               START DATE OF SIMULATION        60  1  1
                    END DATE OF SIMULATION          95  5 31
                    RAINFALL WILL BE READ ON DEVICE  IRAIN  9
                    PRECIP IS IN AES HOURLY FORMAT   IPFORM 1
                    FLOW FILE WILL BE READ ON DEVICE IFLOW 10
```

> Hint: Users often find it is useful to develop a well commented input file and then use it as a template. For different runs or projects, all that needs to be changed in each command are the numerical values.

This is well organized and makes it easy to read the file contents without reference to the user manual. It also contains comments that document what the run used as input information. To illustrate the flexibility of the input system, it is noted that the following line is exactly equivalent to the above excerpt as far as the model is concerned. It is, however, not as user friendly.

```
START 60 1 1 95 5 31 9 1 10
```

This approach to data entry has proven to be a very efficient way to run models and will be maintained in future versions of the tool. However, because of the shift in user expectations since the model was developed, a graphical environment will be released in December 2009 at www.watertoolset.com to accommodate users with other preferences.

4.3 Commands

The following sections provide the details of model commands. ***NOTE: In this manual, not all commands contained in earlier model versions are documented. Care should be used in attempting to apply commands not documented in this manual with this version of the model.***

In the tables, blue sections must always be provided by the user. Yellow sections may or may not be required. The text explains when the yellow sections are/are not required.

4.3.1 The START Command

Purpose:

This command regulates many factors that persist over the course of a model run. It identifies the span of the simulation, provides information on file numbers, and other controlling information.

Use:

The START Command ***should*** be the first command, but can be preceded by commands that don't require information contained on the START command (e.g. IDENTIFY SERIES). It may be repeated at other locations and any new values take effect from that point on.

Numerical Values:

START Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IGY1	0 -or- a valid year	none -or- year	Controls run span	• If IGY1<>0, the simulation will run for a set period. • If IGY1=0, the simulation will run for the whole span of available input data.
IGM1 IGD1 IGY2 IGM2 IGD2	1 through 12 1 through 31 a valid year 1 through 12 1 through 31	month day year month day	n/a	• If IGY1=0 this block must be omitted • IGY1,IGM1 and IGD1 must represent a valid date that precedes a valid date represented by IGY2,IGM2 and IGD2 • The rainfall data provided by the user must encompass the dates provided here.
IRAIN	integer	none	Identifies rainfall file	• Must match value specified in control file (see section 2 of this manual) • Provide dummy value even if rain is not used
IPFORM	1 or 2	none	Specifies format of rainfall file	• If IPFORM=1 AES condensed hourly format will be expected • If IPFORM=2 HEC 'STORM' hourly format will be expected
IFLOW	integer	none	Specifies input value for flow file	• Must match value specified in control file (see section 2 of this manual) • Provide dummy value even if rain is not used
ITFORM	0, 1 or 2	none	Specifies format of temperature file	• If IPFORM=0 temperature will not be expected • If IPFORM=1 AES condensed hourly format will be expected • If IPFORM=2 HEC 'STORM' hourly format will be expected

(Continued next page)

(START continued from previous page)				
Parameter	**Value(s)**	**Units**	**Effect**	**Requirements**
ICASE	0, 1 or 2	none	Specifies catchment evaporation options	• If ICASE=0 evaporation will not be applied to catchment calculations • If ICASE=1, evaporation will be applied to catchment calculations • ***NOTE:*** ○ ***If ICASE<>0 CET, CETIMP and CETPER in GENERATE COMMAND MUST BE SET.*** ○ ***If ICASE=0, CET, CETIMP and CETPER in GENERATE COMMAND MUST BE ELIMINATED.***
CPAN EVAP (12 values)	between 0 and 1 actual	none mm/ month	Pan evaporation correction coefficient for **catchment**	• If ICASE=0 this block must be omitted • If ICASE=1 ○ CPAN and 12 EVAP values are required • If ICASE=2 ○ CPAN is required but the 12 PEVAP values in this block must be omitted ○ the evaporation file specified in the control file will be read for GENERATE calculations
EVAPCASE	0 , 1 or 2	none	Specifies pond evaporation options	• If EVAPCASE=0 evaporation will not be applied to calculations in POND command • If EVAPCASE>0, evaporation will be applied to calculations in POND command
PPAN PEVAP (12 values)	between 0 and 1 actual	none mm/ month	Pan evaporation correction coefficient for POND	• If EVAPCASE=0 this block must be omitted • If EVAPCASE=1 ○ PPAN and 12 PEVAP values are required • If EVAPCASE=2 ○ PPAN is required but the 12 PEVAP values in this block must be omitted ○ the evaporation file specified in the control file will be read for POND calculations
IFDECA	0 or 1	none	Controls pollutant simulation	• If IFDECA=1, a constituent that is removed as a first order decay process will be simulated. • IF IDECA=0, a first order constituent will not be simulated
DECAYK	actual	/hour	First order decay coefficient	• If IFDECA=0 this value must be omitted
IFSEDT	0 or 1	none	Controls sediment simulation	• If IFSEDT=1, sediments will be simulated • If IFSEDT=0, sediments will not be simulated
SEDSET (5 VALUES) SEDDIS (5 VALUES)	actual actual	m/s none	Effective sediment settling velocity Mass fraction of total sediment	• If IFSEDT=0 this block must be omitted • SEDSET(1) must match SEDDIS(1), SEDSET(2) must match SEDDIS(2), and so on to SEDSET(5) must match SEDDIS(5),

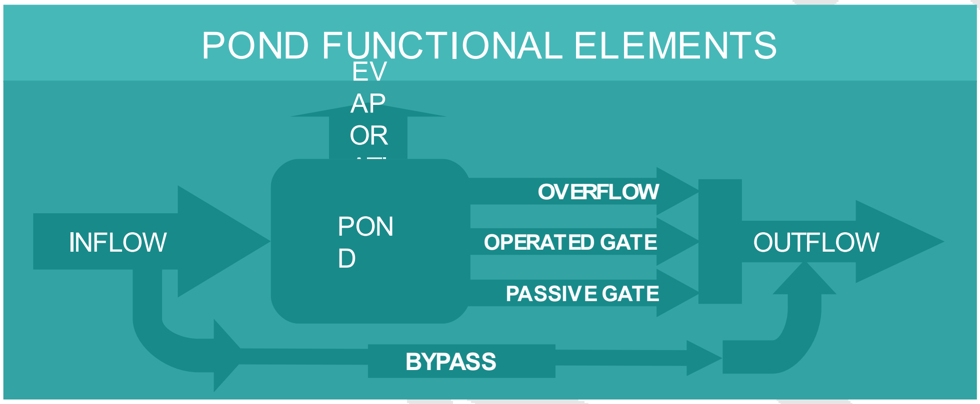
4.3.2 The POND Command

Purpose:

This command enables the simulation of control ponds, reservoirs or small lakes. It routes flows and, optionally, pollutants through the water body, simulating mixing and losses along the way.

Use:

The functions included in the POND command are illustrated below.



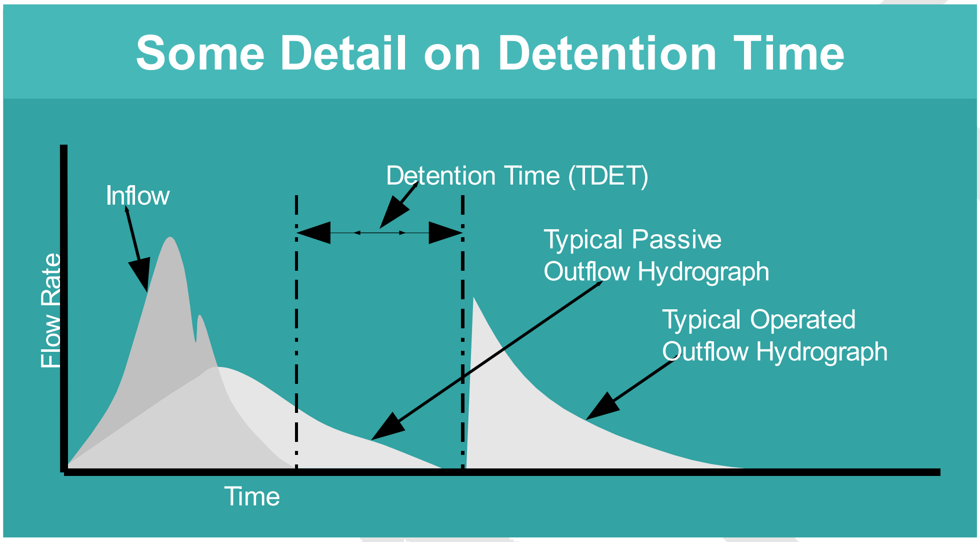
The POND command includes a substantial range of capabilities. As shown below, it can represent:

- a bypass around the whole facility, which sends a stream of inflow around the pond untreated and unrouted, to rejoin the effluent from the pond at its outlet.
- evaporation from the pond, which is a loss from the system,
- three outlet options, including
    - an overflow that only functions when the pond volume exceeds a specified minimum value,
    - a passive gate that always functions to provide an outlet as water in the pond raises and lowers and
    - an operated gate that is closed at the beginning of an inflow into the pond that
    - s a specified base flow amount and that opens after inflow ceases and a specified detention time (TDET) passes.

Experience has shown that the definition and impact of detention time as incorporated in the model is worth highlighting to avoid user misunderstanding of how this term is used.

As indicated in the figure below, the specification of a detention time not only delays the outflow, but changes the form of the outflow since the pond will tend to be fuller when release is delayed and the flows that are generated immediately shift from zero to the value that results from the full depth of water behind the opened gate. Use of outflow structures in combination can reduce this effect, since a limited outflow from a small passive structure will reduce the peak volume in the facility and therefore the rate of outflow that is developed when the operated gate finally opens.

Clearly the combination of losses, outflows and pond stage/volume relationships have a major impact on the simplistic picture just presented, but the general principles illustrate how incorporation of a detention time can affect quantity routing.



The pond volume itself can be simulated as behaving as a completely mixed reactor, a plug flow reactor, or a reactor that behaves between these extremes.

Numerical Values:

POND Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IDOUT	a valid series ID.	none	identifies output series ID.	● IDOUT<>IDH
ISER	integer	none	numerical series name, and flag for detailed outputs	● If ISER<0, then detailed pond behavior will be dumped to a binary file that can be accessed using the PULL POND SPAN and CALC POND STATS commands ● ISER is set to ABS(ISER) after the model is set to dump detailed diagnostics
IDH	a valid series ID.	none	identifies input series ID.	● The series referred to must have been developed in an earlier run or a command prior to the use of the POND command
TDET	actual	hours	specifies detention time.	● TDET must be greater than the computation time step for the pond.
NELS	1 through 99	none	specifies number of mixing elements	● If NELS = 1, the pond is completely mixed. ● If NELS = 99, the pond is approximately plug flow ● NELS between 1 and 99 are intermediate cases
RTINC	Less than 1 hour	hours	specifies computation time step for POND	● The time step for hydraulic routing (not quality routing) is set by RTINC. It should be less than one hour, and should be short enough to represent the pond hydraulics effectively.
QBAS	actual	m3/sec	specifies base flow or bypass flow	● Flows less than QBAS will not be interpreted as an inflow, so operated gate will not be shut. ● If IFQBY<>0, approach flows < QBAS will bypass the pond rather than run into is.
PPAN	actual, 0 or <0	none	pond pan evaporation correction factor	● If PPAN<0, the value of PPAN specified in the START command will be used. ● If PPAN>=0, the value provided here will be used.
IFQBY	1 or 0	none	flag specifying base flow behaviour	● If IFQBY=1, approach flows < QBAS will be diverted around the pond rather than allowed to run into it ● If IFQBY=1, then any approach flow curve provided to the POND command (i.e. set if NPTQQ>0) will be ignored
NPTQQ	0 or 2 through 25	none	specifies if bypass curve will be used	● If NPTQQ=0 the pond will not have a bypass curve ● If NPTQQ>=2, the pond will have a bypass curve
QQ(1,i),QQ(2,i) in pairs where i ranges from 1 to NPTQQ	Approach flow and Inlet flow pairs	m3/sec and m3/sec	sets values for bypass curve	● If NPTQQ=0 this block must be omitted ● All flows must be positive ● The inlet flow (QQ(1,i)) must be greater than the inlet flow (QQ(2,i)) in every pair.
NPTSQ1	0 or 2 through 25	none	specifies if passive outlet will be used	● If NPTSQ1=0 the pond will not have a passive outlet ● If NPTSQ1>=2, the pond will have a passive outlet
SQ1(1,i),SQ1(2,i) in pairs where i ranges from 1 to NPTSQ1	Pond stage and pond outflow pairs	m and m3/sec	sets values for passive outflow curve	● If NPTSQ1=0 this block must be omitted ● All flows must be positive

(Continued next page)

(POND continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
ISIGA	1 or 2	none	specifies how operated outlet will be input	
NPTSQ2 SQ1(1,i),SQ1(2,i) in pairs where i ranges from 1 to NPTSQ2	2 through 25 Pond stage and pond outflow pairs	none m and m3/sec	specifies number of points on operated outlet . sets values for operated outflow curve	• If ISIGA=2 this block must be entirely omitted, but it is otherwise required • If NPTSQ2=0, then this block must contain only NPTSQ2 • All flows must be positive
STHD BHE DIAM GRAV VISC XLENG ROUGH CD	Pipe invert Height of pond Pipe diameter Gravity Water viscosity Pipe length Roughness Discharge coefficient	m m m m/sec2 m none	sets values for calculated operated outflow curve	• If ISIGA=1 this block must be omitted, but it is otherwise required • Consistent units are required
ISIGB	1 or 2	none	specifies how overflow will be input	
NPTSQV SQV(1,i),SQV(2,i) in pairs where i ranges from 1 to NPTSQV	2 through 25 Pond stage and pond outflow pairs	none m and m3/sec	specifies number of points on overflow. sets values for overflow curve	• If ISIGB=2 this block must be entirely omitted, but it is otherwise required • If NPTSQV=0, then this block should contain only NPTSQV • All flows must be positive
STHD BHE XLENG	Pipe invert Height of pond Weir length	m m m	sets values for calculated overflow curve	• If ISIGB=1 this block must be omitted, but it is otherwise required • Consistent units are required
ISIGC	1 or 2	none	specifies how pond volume and area will be input	
NPTSV SV(1,i),SV(2,i) in pairs where i ranges from 1 to NPTSV - - - - - - - - - - - - - - NPTSA SA(1,i),SA(2,i) in pairs where i ranges from 1 to NPTSA	2 through 25 Pond stage and pond volume pairs - - - - - - - - - - - - 2 through 25 Pond stage and pond area pairs	none m and m3/sec - - - - - - - - - none m and m3/sec	specifies number of points on volume curve sets values for volume curve - - - - - - - - - - - - - - specifies number of points on area curve. sets values for area curve	• If ISIGC=2 this block must be entirely omitted, but it is otherwise required. Also, NPTSA and NPTSV cannot both be 0 (i.e. at least an area or volume curve, or both, is required) • If NPTSV=0, then the SV pairs should be omitted • If NPTSA=0, then the SA pairs should be omitted
BWIDTH BSLOPE BLEN BHEIGHT	Basin width Basin slope Basin length Basin height	m m/m m m	sets values for calculated volume and area curves	• If ISIGC=1 this block must be omitted, but it is otherwise required

(POND continued from previous page)				
Parameter	Value(s)	Units	Effect	Requirements
SBEGIN	Starting elevation in pond	m	specifies initial volume in pond	• Must be in range of pond area and volume curves
FCMULT	Multiplier for 1st order constituent concentration	none	adjusts concentrations in pond	• If FCMULT=1, has no effect
SEMULT	Multiplier for sediment concentration	none	adjusts concentrations in pond	• If SEMULT=1, has not effect

4.3.3 The PULL POND SPAN Command

Purpose:

This command allows users to pull a detailed listing of various pond characteristics out of the long time series generated in the POND command.

Use:

The PULL POND SPAN command must follow a POND command, and will develop data for only the most recent preceding POND command. Several PULL POND SPANS can be used in sequence.

The result of each invocation of this command is a listing of data, in a few lines shown as follows:

```
      DATE     TIME       INFLOW     OUTFLOW     STAGE     VOLUME    LOSSES
              (HRS)       (CMS)       (CMS)      (M)       (M**3)    (CMS)
  1970  1  1  0.00       0.000       0.000      0.042     501.854       0.004
  1970  1  1  1.00       0.000       0.000      0.041     487.089       0.004
  1970  1  1  2.00       0.000       0.000      0.040     472.331       0.004
  1970  1  1  3.00       0.000       0.000      0.038     457.581       0.004
  1970  1  1  4.00       0.000       0.000      0.037     442.838       0.004
  1970  1  1  5.00       0.000       0.000      0.036     428.102       0.004
```

Numerical Values:

PULL POND SPAN Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IFY IFM IFD ITY ITM ITD ISKIP	a valid year 1 through 12 1 through 31 a valid year 1 through 12 1 through 31 integer >1	year month day year month day n/a	retrieval span " " " " " " " " " " skip interval	● IFY,IFM and IFD must represent a valid date that precedes a valid date represented by ITY, ITM and ITD ● The date span defined on the START command must encompass the dates provided here. ● ISKIP is not needed. If not present, every record will be retrieved. If present, every ISKIPth record will be retrieved.

4.3.4 The CALC POND STATS Command

Purpose:

This command allows users to develop statistics of the long time series generated in the POND command.

Use:

The CALC POND STATS command must follow a POND command, and will develop data for only the most recent preceding POND command. Several CALC POND STATS can be used in sequence.

The result is a set of four listings of monthly maximum and minimum data and annual maximum and minimum data illustrated as follows:

```
============ MONTHLY MAXIMUMS ==========
  DATE      INFLOW      OUTFLOW      STAGE      VOLUME
             (CMS)       (CMS)        (M)       (M**3)
1960   1   0.164E+00   0.994E-02   0.550E+00   0.665E+04
1960   2   0.427E-01   0.000E+00   0.141E+00   0.168E+04
1960   3   0.174E+00   0.000E+00   0.653E+00   0.815E+04

============ MONTHLY MINIMUMS ==========
  DATE      INFLOW      OUTFLOW      STAGE      VOLUME
             (CMS)       (CMS)        (M)       (M**3)
1960   1   0.000E+00   0.000E+00   0.000E+00   0.000E+00
1960   2   0.000E+00   0.000E+00   0.000E+00   0.000E+00
1960   3   0.000E+00   0.000E+00   0.000E+00   0.000E+00

============ ANNUAL MAXIMUMS ===========
  DATE      INFLOW      OUTFLOW      STAGE      VOLUME
             (CMS)       (CMS)        (M)       (M**3)
 1960      0.916E+00   0.994E-02   0.117E+01   0.178E+05
 1961      0.827E+00   0.000E+00   0.887E+00   0.121E+05
 1962      0.654E+00   0.000E+00   0.866E+00   0.117E+05

============ ANNUAL MINIMUMS ===========
  DATE      INFLOW      OUTFLOW      STAGE      VOLUME
             (CMS)       (CMS)        (M)       (M**3)
 1960      0.000E+00   0.000E+00   0.000E+00   0.000E+00
 1961      0.000E+00   0.000E+00   0.000E+00   0.000E+00
 1962      0.000E+00   0.000E+00   0.000E+00   0.000E+00
```

Numerical Values:

CALC POND STATS Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IFY	a valid year -or- -1	year -or- none	controls statistics span	• If IFY<0, statistics will be calculated for entire run
IFM IFD ITY ITM ITD	1 through 12 1 through 31 a valid year 1 through 12 1 through 31	month day year month day	provides span information	• If IFY<0 this entire block should be omitted • IFY,IFM and IFD must represent a valid date that precedes a valid date represented by ITY, ITM and ITD • The date span defined on the START command must encompass the dates provided here.

4.3.5 The MAXMIN Command

Purpose:

This command allows users to develop maximum and minimum monthly and annual flow rate and volume statistics from any internally generated QUALHYMO time series.

Use:

The command can be issued for any properly constituted time series that remains on the disc. Therefore, the command can be issued throughout the run or collected at the end, so long as at any point the file that is of interest exists and has not been over-written. Data are generated assuming the series contains whole months and years for the period of interest. Output might look something like the following example, although it would typically tend to extend for many months and years:

```
============ MONTHLY MAXIMUMS ==========
           DATE     FLOW RATE
                     (CMS)
          1962  1   0.14110E-03
          1962  2   0.30217E-01
          1962  3   0.34496E-03
============ MONTHLY MINIMUMS ==========
           DATE     FLOW RATE
                     (CMS)
          1962  1   0.00000E+00
          1962  2   0.00000E+00
          1962  3   0.00000E+00
============ MONTHLY TOTALS ============
           DATE     FLOW VOLUME
                     (CU.M.)
          1962  1   0.28333E+02
          1962  2   0.65649E+04
          1962  3   0.99907E+02
============ ANNUAL MAXIMUMS ===========
           DATE     FLOW RATE
                     (CMS)
           1962     0.19767E+00
           1963     0.30116E+00
           1964     0.14915E+00
============ ANNUAL MINIMUMS ===========
           DATE     FLOW RATE
                     (CMS)
           1962     0.00000E+00
           1963     0.00000E+00
           1964     0.00000E+00
============ ANNUAL TOTALS =============
           DATE     FLOW VOLUME
                     (CU.M.)
           1962     0.74269E+05
           1963     0.15110E+06
           1964     0.10825E+06
```

Numerical Values:

MAXMIN Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID	A valid series ID.	none	identifies series to be processed	● a current and compatible QUALHYMO series file ● NOTE: Any properly identified file that exists in the project workspace will be processed. This need not be from a current run, it can be a legacy series left on the disc.
MONTHORYEAR	+/- 1,2 or 3	none	identifies what is to be created	● if MONTHORYEAR<0 then ○ MONTHORYEAR is reset to positive value ○ The whole series will be processed ● IFY and ITY should not be entered ● If ABS(MONTORYEAR)=1 yearly extremes printed ● If ABS(MONTORYEAR)=2 monthly extremes printed ● If ABS(MONTORYEAR)=3 yearly and monthly extremes printed
IFY ITY	A valid year A valid year	years years	starting year ending year	● IFY must predate ITY ● This block should not be input if MONTHORYEAR < 0

4.3.6 The IDENTIFY SERIES Command

Purpose:

This command allows users to access and read the metadata associated with a QUALHYMO series file.

Use:

Experience has shown that associating metadata with a series file can avoid confusion as to what the file was, how it was generated, and other key data that might be useful at some time after the work has been done. Making those data accessible can involve a huge range of technologies and options. In this case, it was decided to embed the metadata directly in the file, as discussed in section 4 below (Model Unit Files). Since the files are unformatted binary files, they cannot be readily read (although interested users can 'see' certain formatted information on the files). To make the metadata directly accessible, this command has been developed. Invoking it produces output as shown in the figure below.

```
*** SERIES HEADER INFORMATION BEGINS
SERIES INTERNAL NAME IS         :  101
SERIES TIME STEP (HOURS) IS:    :  1.00000
SERIES CREATION COMMAND WAS     : GENERATE
SERIES FILE WAS CREATED ON      ; Sat Jul 14 09:25:35 2007
INPUT FILE CREATING SERIES WAS  : REGEN.INP
MODEL VERSION WAS               : QUALHYMO0777V1AB40
COMPILER VERSION WAS            : ABSOFT 10.00.07
*** SERIES HEADER INFORMATION ENDS
```

The content is self explanatory. It is noted that the mechanism chosen for embedding this information will support substantial long term additions with only a limited risk of losing downward compatibility.

Numerical Values:

IDENTIFY SERIES Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID	A valid series ID.	none	extracts series file metadata	• a current and compatible QUALHYMO series file • NOTE: Any properly identified file that exists in the project workspace will be extracted. This need not be from a current run, it can be a legacy series left on the disc.

4.3.7 The PRINT MODEL DETAILS Command

Purpose:

This command allows users to have the model issue a set of details that document the major facts of the current model run.

Use:

Project documentation is a key element of professional practice, since the user is responsible for maintaining accurate and effective records as to the intent and outcome of their modelling work. Users will typically have practices in place that address this. As a supplement to those practices, it was recognized that an easy way to embed basic run information in a QUALHYMO output file would be useful. This command does that. It generates some essential information for the run. A typical output might be as follows:

```
*** MODEL INFORMATION BEGINS
INPUT FILE FOR THIS RUN IS      :REGEN.INP
THIS RUN STARTED AT             :Tue Jul 17 22:48:05 2007
THIS MODEL VERSION IS           :QUALHYMO0777V1AB42
THE COMPILER USED ON THIS BUILD:ABSOFT 10.00.07
FILE ASSOCIATIONS ARE           :
   ID  9  = RAIN.PRE
   ID  8  = TEMP.TMP
   ID  7  = EVAP.TXT
*** MODEL INFORMATION ENDS
```

The content is self explanatory. It is noted that the mechanism chosen for embedding this information will support substantial long term additions with only a limited risk of losing downward compatibility.

Numerical Values:

PRINT MODEL DETAILS Command Information				
Parameter	Value(s)	Units	Effect	Requirements
n/a	none	none	extracts and prints basic model run information	• none

4.3.8 The GENERATE Command

Purpose:

This command allows users to simulate runoff from a watershed.

Use:

This provides the ability to represent long term continuous flow from a watershed. It is not intended to provide an ability to simulate detailed urban networks as might be done with the SWMM model. Rather, it is intended to enable the representation of catchments at a level of detail suitable for representation of the major factors that govern long term runoff processes. If more detailed representation is needed, this can be done by splitting the watershed into more numerous smaller catchments and aggregating the result, or by incorporating the results of an external model.

Numerical Values:

GENERATE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID	A valid series ID.	none	simulates runoff from a watershed.	● None other than the ID number needs to be a valid QUALHYMO time series identifier. ● Any series already present and using the same ID number will be overwritten by this command.
ISER	integer	none	numerical series name,.	● Should be unique to the catchment being run.
DT	Integer fraction of 1 hour or multiples of 1 hour	hours	specifies computation time step for GENERATE	● The time step for computation of It should be should be short enough to represent the rainfall runoff processes effectively. ● Typically, one hour is a useful time step for watershed level simulation.
AREA	actual	Hectares	specifies watershed area.	
AB	0 or 1	none	controls printing of unit hydrographs	● If AB = 1, the unit hydrographs are printed. ● If AB = 0, the unit hydrotraphs are not printed.
FRIMP	0->1 inclusive	none	fraction of directly connected impervious area.	● If FRIMP = 0, omit next block
AA	1 or 2	none	specifies unit hydrograph type	***<u>Omit this block if FRIMP=0</u>*** ● If AA=1, a Nash hydrograph will be used ● If AA=2, a Williams hydrograph will be used ● If AA=1 XK is Nash 'n' ● If AA=2 and XK>0 XK is Williams 'K' ● If AA=2 and XK<0 XK is height 'H' ● If AA=1, TP is Nash parameter ● If AA=1, TP is Williams parameter
XK	real>0 if AA=1 real if AA=2	none	hydrograph shape parameter	
TP	real >0	none	hydrograph shape parameter	
ABSIMP	real >=0	mm	impervious initial abstraction	
RIMP	real 0-1	none	impervious runoff coefficient	
CETIMP	real 0-1 or <0	none	impervious area evaporation correction coefficient	● ***OMIT CETIMP if ICASE =0 (on START command)*** ● If CETIMP<0, CPAN from START command will be used

(Continued next page)

(GENERATE continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
AA	1 or 2	none	specifies unit hydrograph type	***Omit this block if FRIMP=1*** • If AA=1, a Nash hydrograph will be used • If AA=2, a Williams hydrograph will be used • If AA=1 XK is Nash 'n' • If AA=2 and XK>0 XK is Williams 'K' • If AA=2 and SK<0 XK is height 'H' • If AA=1, TP is the Nash parameter • If AA=1, TP is the Williams parameter
XK	real>0 if AA=1 real<> if AA=2	none	hydrograph shape parameter	
TP	real >0	none	hydrograph shape parameter	
SMIN	real>=0	mm	minimum S	
SMAX	real>0	mm	maximum S	
SK	real>0	/mm	S change	
APIK	real>0, <1	none	parameter	• Typically, APIK is near 0.9 per day
API	real>0	mm	initial API value	
ABSPER	real >=0	mm	pervious initial abstraction	
CETPER	real 0-1 or <0	none	pervious area evaporation correction coef	• ***OMIT CETPER if ICASE =0 (on START command)*** • If CETPER<0, CPAN from START command will be used
NSVOL	positive integer ***MUST BE 0 THIS MODEL VERSION***	none	number of soil reservoirs for baseflow simulation	• If NSVOL=0, a recession curve will be used • If NSVOL>0, a cascade of linear reservoirs is used • NSVOL=0, omit next block
SVOL(I),I=1,NSVOL	positive real	mm	starting reservoir volume	• Omit this whole group if NSVOL=0 • Provide one group for each of I=1,NSVOL
SK1(I),I=1,NSVOL	positive real	/s	reservoir constant	
Sk2(I),I=1,NSVOL	positive real	none	baseflow coef	
BASMIN	positive real	m3/s	minimum baseflow rate	
BFACR	real>=1.0	mm	baseflow coefficient	• Omit this whole group if NSVOL>0
SVOL	positive real	mm	starting reservoir volume	
SWILT	positive real	mm	minimum SVOL for baseflow	
SFIELD	positive real	mm	soil field capacity	
SLOSKA	positive real	/s	recession const.	
SLOSKB	positive real	none	reduction factor.	
CET	real 0-1 or <0	none	ET coefficient	• ***OMIT CET if ICASE =0 (on START command)*** • If CET<0, CPAN from START command will be used
ISNOW	1 or 2	none	sets snowmelt method	• ***Omit this whole group if temperature files are not provided*** • If SNOW=1 The model will use annual coefficients • If ISNOW=2 the model will use variable coefficient.
BASET	real	deg C	melt temperature	
SNOFAC	real	none	calibration coefficient	
PACDEP	real	mm	starting pack depth	• Applies to melt calculations
ALPHAA	real	none	calibration coefficient	
XKL	real	none	thermal conductivity ratio.	• Applies to geothermal heat flux
BCOEF	real	none	proportionality constant	
XNCOEF	real	none	insulation factor	

(Continued next page)

(GENERATE continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
KFLAG	0, 1 0R 2	none	sets removal option	● ***IN THIS VERSION SET KFLAG=0***
XN	0-1	none	fraction of snow removed	● Omit XN, DEPTH AND AREAD if KFLAG=0
DEPTH	real>0	mm	depth before removal begins	
AREAD	real>0	ha	area of donor watershed	
PSTATE	real	deg C	temp. for precip. to be snow.	● Omit this section if ISNOW=1
COEFD	real>0	none	calibration coefficient	
COEFE	real>0	none	calibration coefficient	
CFACTR	real>0	none	rain gauge correction factor	
CFACTS	real>0	none	snow gauge correction factor	
IZFLAG	1,2 OR 3	none	sets melt coefficient option	● If IZFLAG=1, user provides melt coefficients ● If IZFLAG=2, Anderson-Gray melt coefficients are used ● If IZFLAG=3 Dorset, Ontario melt coefficients are used
CMELT((I), I=1,12)	real	none	melt coefficient	● Omit this section if IZFLAG<>1

4.3.9 The ADD SERIES Command

Purpose:

This command allows users to add two series together. The shortest time step of the two will be used if the time steps are not the same. Note that the larger time step must be an integer multiple of the shorter time step if the two are not equal.

Use:

The ADD SERIES command must follow at least two commands creating output series, or must be used in a situation where two output series already reside on disc, because the command needs two input series to function. Several ADD SERIES can be used in sequence.

The result of each invocation of this command is a new series that is the sum of the two input series.

Numerical Values:

ADD SERIES Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IDOUT	A valid series ID.	integer	sets the output ID	• IDINA and IDINB must be existing series IDs • Time steps of IDINA and IDINB must be equal or one must be an integer multiple of the other
ISER	integer	integer	sets output name	
IDINA	A valid series ID.	integer	sets an input ID	
IDINB	A valid series ID.	integer	sets an input ID	

4.3.10 The PULL SERIES SPAN Command

Purpose:

This command allows users to pull a detailed listing of a subset of a time series data set.

Use:

The PULL SERIES SPAN command can be used if the specified time series exists.

The result of each invocation of this command is a listing of data, prefaced by file time step and name information. A few lines of a typical output are shown as follows:

```
  FILE CHARACTERSTICS ==========
        NAME =    208
   TIME STEP =    1.00000

 YYYYMMDD     Q          D          S1          S2          S3          S4          S5
1973 215 0.1307E-03 0.0000E+00 0.1838E-09 0.7845E-10 0.7748E-10 0.3606E-09 0.2527E-08
1973 215 0.2209E-03 0.0000E+00 0.5500E-09 0.2345E-09 0.2308E-09 0.1063E-08 0.6325E-08
1973 215 0.3345E-03 0.0000E+00 0.1299E-08 0.5531E-09 0.5424E-09 0.2471E-08 0.1272E-07
1973 215 0.4686E-03 0.0000E+00 0.2608E-08 0.1109E-08 0.1083E-08 0.4881E-08 0.2202E-07
1973 215 0.6215E-03 0.0000E+00 0.4677E-08 0.1987E-08 0.1933E-08 0.8613E-08 0.3460E-07
```

Numerical Values:

PULL SERIES SPAN Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID IFY IFM IFD ITY ITM ITD	a valid series ID. a valid year 1 through 12 1 through 31 a valid year 1 through 12 1 through 31	integer year month day year month day	specifies file controls retrieval span	• IFY,IFM and IFD must represent a valid date that precedes a valid date represented by ITY, ITM and ITD • The date span defined on the START command must encompass the dates provided here.

4.3.11 The SPLIT SERIES Command

Purpose:

This command allows users to split a time series into two using a defined flow split curve.

Use:

The SPLIT SERIES command must refer to an existing time series file, and must include identification for the two sub-series as well as the flow split curve. When used, the original file remains intact and the two new files are generated.

Numerical Values:

SPLIT SERIES Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IDIN	a valid series ID.	n/a	series to be split	● The split curve is specified in pairs where ○ the first value is the arrival flow rate ○ the second value is the split rate for series IDOUTA ○ the series IDOUTB is calculated as the difference between the arrival flow rate and the rate for IDOUTA ● The QQ curve must increase monotonically ● The split values must all be less than the corresponding arrival values
IDOUTA	a valid series ID<>IDIN	n/a	1st output ID	
ISERA	integer>0	n/a	1st output name	
IDOUTB	a valid series ID<>IDIN and <> IDOUTA	n/a	2nd output ID	
ISERB	integer>0	n/a	2nd output name	
NPTQQ	integer>0	n/a	# points on split curve	
$(QQ_{(1,I)},QQ_{(2,I)})_{I=1,NPTQQ}$	real pairs	cu.m./sec	split curve pairs	

4.3.12 The FILTERED REMOVAL Command

Purpose:

This command allows users to represent a BMP that operates as a filter medium, as might appear in an LID or sand filter system.

Use:

The FILTERED REMOVAL command must refer to an existing time series file, and must include identification for the output series as well as the parameters required for calculation. The method used is based on Urbonas, Ben R. (Stormwater Sand Filter Sizing and Design A Unit Operations Approach) found at
http://www.udfcd.org/downloads/pdf/tech_papers/Sand-flt-paper.pdf

Numerical Values:

FILTERED REMOVAL Command Information				
Parameter	Value(s)	Units	Effect	Requirements
IDIN	a valid series ID.	n/a	series to filter	• The filter throughflow is calculated as Q=FAREA*UNITFC*EXP(-DECAYC*ACCUML) where ACCUML = accumulated filter load • The filter load increases as the sum of arriving loads with capture = FREM*arrival rate • The filter load is reduced by the fraction PCTREM/100 once every MAINT times per year
IDOUT	A valid series ID.<>IDIN	n/a	output series	
ISEROUT	integer>0	n/a	output name	
FAREA	real>0	sq m	filter area	
UNITFC	real>0	none	filter constant	
DECAYC	real>0	none	decay constant	
FREM	real>=0,<1	none	fraction removed	
TSSINIT	real>0	mg	initial TSS load	
MAINT	integer>0	/year	maintenance cycle	
PCTREM	real>0,<=100	%	percent filter load removed	
BUFVOL	real>0	cu m	buffer volume	

4.3.13 The EXCEEDANCE Command

Purpose:

This command allows users to develop exceedance curves for any time series generated by the model.

Use:

The command can be applied either with specific curves provided by the user, or with the model calculating curves based on evenly spaced points over the range of the input series.

EXCEEDANCE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID	a valid input id	n/a	identifies series to develop curves for	
NINQ	integer <=20	n/a	controls points on input flow curve	• if NINQ=0, this curve will not be calculated • if NINQ<0, the model will determine the curve • if NINQ>0, the user inputs the curve
Q(i),i=1,NINQ	real	m3/s	provides span information	• If NINQ<=0 this entire block should be omitted
NIND	integer <=20	n/a	controls points on input first order curve	• if NIND=0, this curve will not be calculated • if NIND<0, the model will determine the curve • if NIND>0, the user inputs the curve
D(i),i=1,NIND	real	conc.	provides span information	• If NIND<=0 this entire block should be omitted
NINS	integer <=20	n/a	controls points on input first order curve	• if NINS=0, this curve will not be calculated • if NINS<0, the model will determine the curve • if NINS>0, the user inputs the curve
S(i),i=1,NINS	real	conc.	provides span information	• If NINS<=0 this entire block should be omitted
NUMINT	integer	n/a	# of intervals to calculate exceedance curves for	
IFY IFM IFD ITY ITM ITD	a valid year1 through 12 1 through 31 a valid year 1 through 12 1 through 31	year month day year month day	provides span information	• the date span defined on the START command must encompass the dates provided here. • user must input NUMINT sets of intervals (all six numbers)

4.3.14 The CALC SERIES STATS Command

Purpose:

This command allows users to develop some basic parametric and non-parametric statistics for any time series.

Use:

The command can be applied to any series generated by the model, and can be set to ignore or account for zero values in the series.

CALC SERIES STATS Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID	a valid input id	n/a	identifies series to develop curves for	
IFLAG	integer = 0 or 1	n/a	controls inclusion of zero values	● if IFLAG=0 zero flows will be included in calcs ● if IFLAG=1 zero flows will not be included in calcs

4.3.15 The WATERSHED Command

Purpose:

This command allows users to simulate runoff from a watershed. It is similar but not identical to the old GENERATE command.

Use:

This provides the ability to represent long term continuous flow from a watershed. It is not intended to provide an ability to simulate detailed urban networks as might be done with the SWMM model. Rather, it is intended to enable the representation of catchments at a level of detail suitable for representation of the major factors that govern long term runoff processes. If more detailed representation is needed, this can be done by splitting the watershed into more numerous smaller catchments and aggregating the result, or by incorporating the results of an external model.

Numerical Values:

WATERSHED Command Information				
Parameter	Value(s)	Units	Effect	Requirements
ID	A valid series ID.	none	simulates runoff from a watershed.	● None other than the ID number needs to be a valid QUALHYMO time series identifier. ● Any series already present and using the same ID number will be overwritten by this command.
ISER	integer	none	numerical series name,.	● Should be unique to the catchment being run.
DT	Integer fraction of 1 hour or multiples of 1 hour	hours	specifies computation time step for GENERATE	● The time step for computation of It should be should be short enough to represent the rainfall runoff processes effectively. ● Typically, one hour is a useful time step for watershed level simulation.
AREA	actual	Hectares	specifies watershed area.	
AB	0 or 1	none	controls printing of unit hydrographs	● If AB = 1, the unit hydrographs are printed. ● If AB = 0, the unit hydrotraphs are not printed.
FRIMP	0->1 inclusive	none	fraction of directly connected impervious area.	● If FRIMP = 0, omit next block
AA	1 or 2	none	specifies unit hydrograph type	***<u>Omit this block if FRIMP=0</u>*** ● If AA=1, a Nash hydrograph will be used ● If AA=2, a Williams hydrograph will be used
XK	real>0 if AA=1 real if AA=2	none	hydrograph shape parameter	● If AA=1 XK is Nash 'n' ● If AA=2 and XK>0 XK is Williams 'K' ● If AA=2 and XK<0 XK is height 'H'
TP	real >0	none	hydrograph shape parameter	● If AA=1, TP is Nash parameter ● If AA=1, TP is Williams parameter
ABSIMP	real >=0	mm	impervious initial abstraction	
RIMP	real 0-1	none	impervious runoff coefficient	
CETIMP	real 0-1 or <0	none	impervious area evaporation correction coefficient	● ***OMIT CETIMP if ICASE =0 (on START command)*** ● If CETIMP<0, CPAN from START command will be used

(Continued next page)

(WATERSHED continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
AA	1 or 2	none	specifies unit hydrograph type	***<u>Omit this block if FRIMP=1</u>*** ● If AA=1, a Nash hydrograph will be used ● If AA=2, a Williams hydrograph will be used
XK	real>0 if AA=1 real<> if AA=2	none	hydrograph shape parameter	● If AA=1 XK is Nash 'n' ● If AA=2 and XK>0 XK is Williams 'K' ● If AA=2 and SK<0 XK is height 'H'
TP	real >0	none	hydrograph shape parameter	● If AA=1, TP is the Nash parameter ● If AA=1, TP is the Williams parameter
SMIN	real>=0	mm	minimum S	
SMAX	real>0	mm	maximum S	
SK	real>0	/mm	S change	
APIK	real>0, <1	none	parameter	● Typically, APIK is near 0.9 per day
API	real>0	mm	initial API value	
ABSPER	real >=0	mm	pervious initial abstraction	
CETPER	real 0-1 or <0	none	pervious area evaporation correction coef	● ***OMIT CETPER if ICASE =0 (on START command)*** ● If CETPER<0, CPAN from START command will be used
BASMIN	positive real	m3/s	minimum baseflow rate	
SVOL	positive real	mm	starting soil reservoir volume	
SLOSKA	positive real	/s	recession const.	
SLOSKB	positive real	none	reduction factor.	
CET	real 0-1 or <0	none	ET coefficient	● ***OMIT CET if ICASE =0 (on START command)*** ● If CET<0, CPAN from START command will be used
ISNOW	1 or 2	none	sets snowmelt method	● ***Omit this whole group if temperature files are not provided*** ● If SNOW=1 The model will use annual coefficients ● If ISNOW=2 the model will use variable coefficient.
BASET	real	deg C	melt temperature	
SNOFAC	real	none	calibration coefficient	
PACDEP	real	mm	starting pack depth	● Applies to melt calculations
ALPHAA	real	none	calibration coefficient	
XKL	real	none	thermal conductivity ratio.	● Applies to geothermal heat flux
BCOEF	real	none	proportionality constant	
XNCOEF	real	none	insulation factor	

(Continued next page)

(WATERSHED continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
KFLAG	0, 1 0R 2	none	sets removal option	● ***IN THIS VERSION SET KFLAG=0***
XN DEPTH AREAD	0-1 real>0 real>0	none mm ha	fraction of snow removed depth before removal begins area of donor watershed	● Omit XN, DEPTH AND AREAD if KFLAG=0
PSTATE COEFD COEFE CFACTR CFACTS IZFLAG	real real>0 real>0 real>0 real>0 1,2 OR 3	deg C none none none none none	temp. for precip. to be snow. calibration coefficient calibration coefficient rain gauge correction factor snow gauge correction factor sets melt coefficient option	● Omit this section if ISNOW=1 ● If IZFLAG=1, user provides melt coefficients ● If IZFLAG=2, Anderson-Gray melt coefficients are used ● If IZFLAG=3 Dorset, Ontario melt coefficients are used
CMELT((I), I=1,12)	real	none	melt coefficient	● Omit this section if IZFLAG<>1

4.3.16 The PERVIOUS SURFACE Command

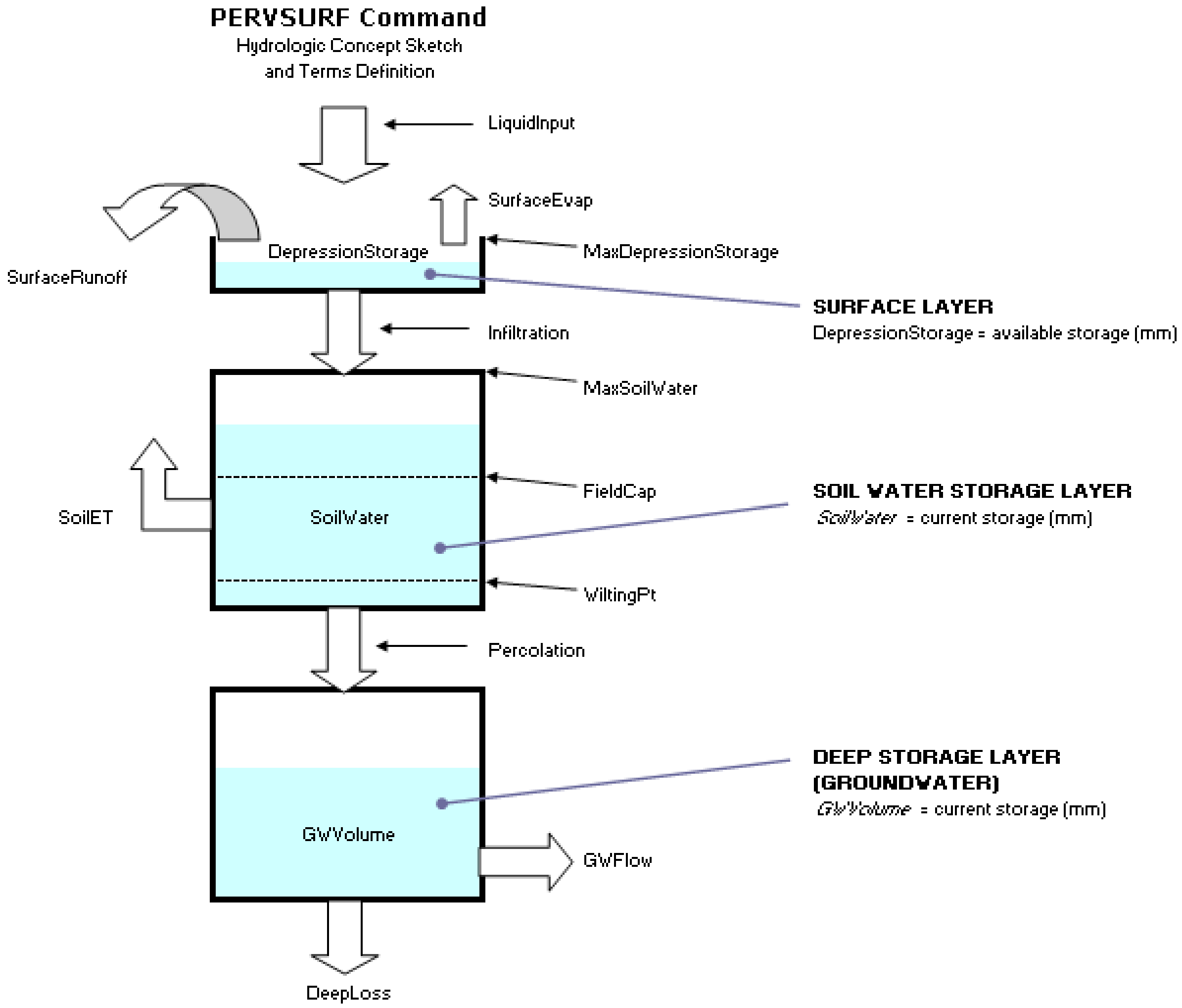
Purpose:

This command simulates hydrologic response of pervious areas (typically, vegetated or open surfaces with little development) including pollutant loads associated with surface runoff.

Use:

This command was developed specifically for the Toronto Region Conservation Authority, although it can be used elsewhere if appropriate. It can be used in place of the GENERATE command. It is one of four such commands (PERVSURF, IMPERVSURF, PERVSURFSTORAGE, IMPERVSURFSTORAGE) which were implemented to facilitate simulation of LID alternatives. As well as this section, Appendix B provides some general guidance on the use of these four commands.

The conceptual arrangement of this hydrologic element is shown below:



Outputs from the command provide several sets of information:

- Two binary outflow time series: one is surface runoff outflow (m^3/s), while the other is subsurface (groundwater) outflow (m^3/s) which conceptually could represent groundwater discharge to catchment outlet (e.g. to outlet watercourse)
- There is text output file that provides a water balance summary for each layer.
- Users can generate an optional detailed trace output: If **SurfaceID** is supplied as a negative value, then PERVSURF will generate a detailed trace output file (ASCII text format):
  - **PERVSTEPnnn.txt**: contains all water budget components (in mm), surface outflow (m^3/s) and subsurface outflow (m^3/s) for each time step (one record per time step, easy to import into spreadsheet) where **nnn = SurfaceID**

Some of the key computational factors in this command are:

- It verifies mass balance on each time step.
- Surface runoff is generated when LiquidInput exceeds current infiltration capacity, which varies with soil moisture status (see below)
- Surface runoff flow rate is computed based on the surface being represented as a sloped plane with the characteristic slope, length and surface roughness supplied as input by the modeler. The model continuously tracks the depth of water on the surface plane, and over any interval the surface outflow rate is computed using the Manning equation with hydraulic radius approximated as depth of flow.
- Soil moisture is continuously tracked. When soil moisture is above field capacity, then the excess is considered as free water that can percolate by gravity downward to the deep storage layer. The percolation rate is a constant rate supplied by the user (presumably would correspond to vertical saturated hydraulic conductivity of limiting layer).
- Soil moisture is depleted by evapotranspiration. The user supplies potential ET rates with the START command. The potential ET rates can be scaled using SoilETfactor.
- When soil moisture is at or above field capacity (wet soil condition) ET occurs at the potential rate (i.e. no limit to water supply to root systems). Actual ET will drop to zero as soil moisture drops from field capacity to wilting point.
- There is no ET from the deep storage layer. Therefore, the soil water storage layer represents the full depth of soil or overburden layers from which root systems can extract water. The input values for Field Capacity and MaxSoilWater need to be set accordingly. MaxSoilWater will generally be approximated as available porosity times the depth of the soil water layer. Similarly, field capacity as well as wilting point will in part depend on the total layer depth being represented by the soil water layer. Field capacity per unit depth will be higher in finer-textured soils (e.g. clays) than coarse-texture soils (e.g. sand).

Numerical Values:

PERVIOUS SURFACE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
SurfaceID	integer > 0	none	ID for surface outflow series	
SubsurfaceID	integer > 0	none	ID for subsurface outflow series	
ExtInflowID	integer > 0	none	ID for external surface inflow	
DT	real > 0 Integer fraction of 1 hour or multiples of 1 hour	hr	Time Step	• DT must be an integer divisor or multiple of one hour
SurfArea	real > 0	ha	Surface Area	
SurfLength	real > 0	m	Characteristic length of catchment flow path	
SurfSlope	real > 0	m/m	Characteristic slope catchment	
SurfRoughness	real > 0	none	Characteristic catchment surface roughness	
MaxDepressionStorage	real > 0	mm	Surface depression storage	
MaxSoilWater	real > 0	mm	soil layer's water storage capacity	
FieldCap	real > 0	mm	Field capacity	
WiltingPt	real > 0	mm	Wilting point	
InfiltRate1	real > 0	mm/hr	Max surface infilt rate	
InfiltRate2	real > 0	mm/hr	Min surface infilt rate	
PercRate	real > 0	mm/day	Percolation rate from soil water layer to groundwater layer	
GWOutflowCoeff	real > 0	mm/day	Subsurface outflow coeff	
DeepLoss Coeff	real > 0	mm/da	Deep loss coeff	
SoilETFactor	real > 0	none	Factor by which user-supplied potential evap raters are multiplied to get applied potential ET	
InitDepression	real > 0	mm	Initial depression storage	
InitSoilWater	real > 0	mm	Initial Soil water storage level	
InitGWVolume	real > 0	mm	Initial groundwater storage level	
SNOWMELT PARAMETERS	Same as GENERATE command	none	Same as GENERATE command	

4.3.17 The IMPERVIOUS SURFACE Command

Purpose:

This command simulates hydrologic response of impervious (e.g. paved surface) including pollutant loads associated with surface runoff.

Use:

This command was developed specifically for the Toronto Region Conservation Authority, although it can be used elsewhere if appropriate. It can be used in place of the GENERATE command. It is one of four such commands (PERVSURF, IMPERVSURF, PERVSURFSTORAGE, IMPERVSURFSTORAGE) which were implemented to facilitate simulation of LID alternatives. As well as this section, Appendix B provides some general guidance on the use of these four commands.

Users have several output types to consider:

- There are three binary outflow time series:
  - OutflowID1= FlowFraction1 * surface runoff flow and pollutant loads
  - OutflowID2 = FlowFraction2 * surface runoff flow and pollutant loads
  - OutflowID3 = FlowFraction3 * surface runoff flow and pollutant loads
- The text output file provides water balance summary.
- Users can request an optional detailed trace output: If **OutflowID1** is supplied as a negative value, then IMPERVSURF will generate a detailed trace output file (ASCII text format). This is in the form of a file, "**IMPERVSTEPnnn.txt"** where nnn = OutflowID1 that contains all water budget components (in mm) and the total surface and subsurface outflow rates (m^3/s) for each time step. This is produced with one record per time step, so that it is easy to import into a spreadsheet for further analysis.

Numerical Values:

IMPERVIOUS SURFACE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
OutflowID1	integer > 0	none	ID for surface outflow series #1. This flow series will be comprised of FlowFraction1 of the total surface runoff from the impervious surface	
OutflowID2	integer > 0	none	ID for surface outflow series #2 which will be FlowFraction2 of the total surface runoff from the impervious surface	
OutflowID3	integer > 0	none	ID for surface outflow series #3 which will be FlowFraction3 of the total surface runoff from the impervious surface	
FlowFraction1	real > 0	none	Fraction of total surface runoff and pollutant load written as series OutflowID1	
FlowFraction2	real > 0	none	Fraction of total surface runoff and pollutant load written as series OutflowID2	
FlowFraction3	real > 0	none	Fraction of total surface runoff and pollutant load written as series outlflowID3	

(Continued next page)

(IMPERVIOUS SURFACE continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
ExtInflowID	integer > 0	none	ID for external surface inflow series	
DT	real > 0	hr	Time Step	
SurfArea	real > 0	ha	Surface Area	
SurLength	real > 0	m	Characteristic length of catchment flow path	
SurfSlope	real > 0	m/m	Characteristic slope catchment	
SurfRoughness	real > 0	none	Characteristic catchment surface roughness	
MaxDepression Storage	real > 0	mm	Surface depression storage	
InitDepression	real > 0	mm	Initial depression storage	
SNOWMELT PARAMETERS		none	Same as GENERATE command	

4.3.18 The PERVIOUS W STORAGE Command

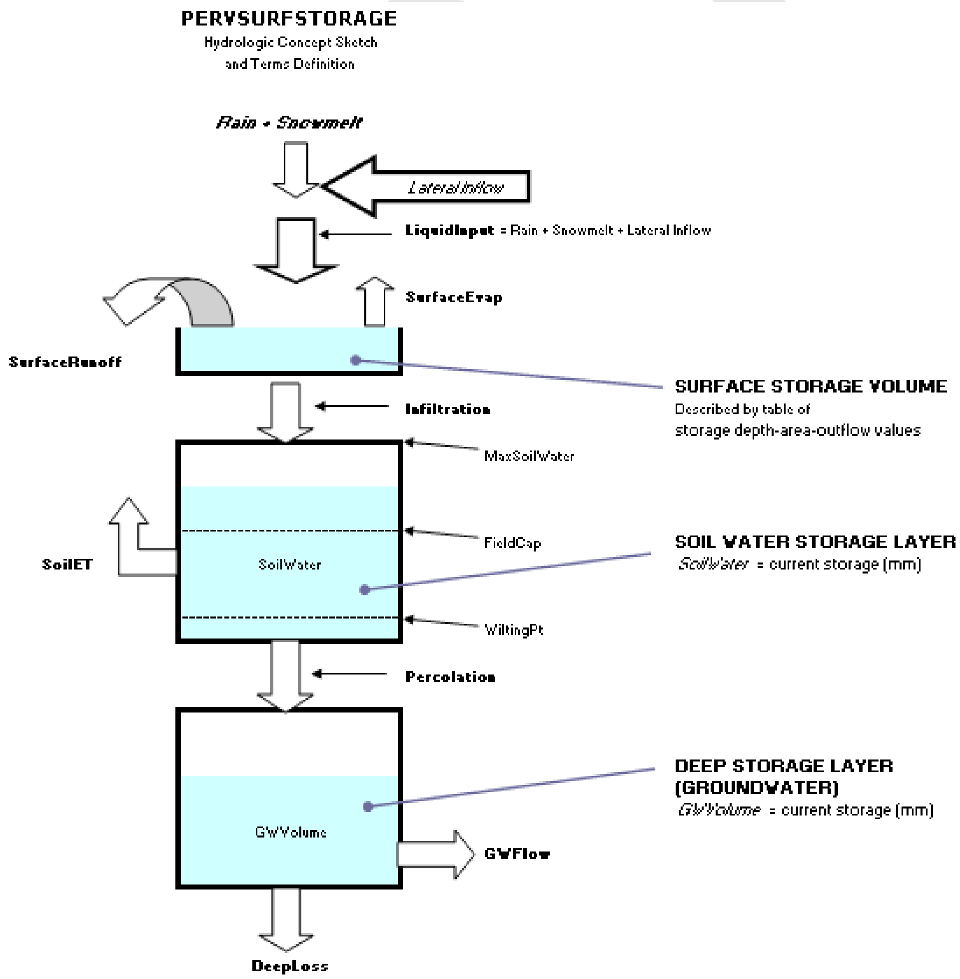
Purpose:

This command simulates hydrologic response of pervious surface that includes a surface storage volume. It is intended to allow modeling of a soakaway areas or bioretention areas that consist of vegetated area with a ponding capacity allowing both retention and infiltration into the soil profile.

Use:

This command was developed specifically for the Toronto Region Conservation Authority, although it can be used elsewhere if appropriate. It can be used in place of the GENERATE command. It is one of four such commands (PERVSURF, IMPERVSURF, PERVSURFSTORAGE, IMPERVSURFSTORAGE) which were implemented to facilitate simulation of LID alternatives. As well as this section, Appendix B provides some general guidance on the use of these four commands.

The conceptual arrangement of this hydrologic element is shown below:



The surface storage receives all surface runoff generated by the pervious surface as well as a user-specified lateral inflow series which can be any available time series. The surface storage is characterized by a table of depth-area-outflow values supplied by the user. Losses from storage include evaporation as well as exfiltration of water into the soil profile.

Outputs from the command provide several sets of information:

- There are two binary outflow time series: one is surface runoff outflow (m^3/s), the other is subsurface (groundwater) outflow (m^3/s) which conceptually can represent groundwater discharge to catchment outlet (e.g. to outlet watercourse)
- There is a text output file provides water balance summary for each layer.
- Users can specify an optional detailed trace output: If **SurfaceID** is supplied as a negative value, then PERVSURF will generate two detailed trace output files (ASCII text format):
  - **PERVSTORSTEPnnn.txt**: contains all water budget components (in mm) and the surface and subsurface outflow rates (m^3/s), for each time step (one record per time step, easy to import into spreadsheet) where **nnn = SurfaceID**
  - **PERVSURFSTORAGE.tra:** contains trace output for the surface storage volume only, including inflow, outflow, exfiltration ,evaporation and pollutant loads in and out, for each time step

Some of the computational features are:

- Soil moisture tracking and soil water storage layer simulated in same manner as described for PERVSURFACE command.
- When LiquidInput exceeds current infiltration capacity, the excess water is added to the surface storage volume.
- The surface storage volume will also receive as direct input the external inflow time series (flow and pollutant loads).
- The surface storage is defined by the user, as a table of depth-area-outflow values. The table must consist of a maximum of 20 rows, with each row containing a depth of ponding (m), a corresponding water surface area (m^2) and outflow rate (m^3/s). The values must increase from row to row. Note that the water surface area on the final row (i.e. the largest water surface area) must be less than the total surface area for the pervious surface (SurfArea)
- The surface storage volume is updated by mass balance over each time step. Water surface area is concurrently updated. Surface storage is depleted by exfiltration into the soil water storage layer. Surface storage is also depleted by surface evaporation. Exfilitration and evaporation occur only over the inundated area (i.e. water surface area). Surface outflow will occur according to the table of depth-area-outflow values.
- Pollutant routing occurs in the same manner as in the POND command. The settling velocities and first-order time decay parameter specified with the START

command are used. As with the POND command, pollutant routing is based on considering the volume to be a series of completely stirred tank reactors (nCSTRs). With nCSTRs=1, the volume is effectively a completely-mixed volume. With nCSTRs set at 5 or greater, the conditions approach those of a plug-flow reactor. With nCSTRs set between 1 and 5, the result can be considered as intermediate between completely mixed flow and plug flow.

Numerical Values:

PERVIOUS W STORAGE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
SurfaceID	integer > 0	none	ID for surface outflow series	
SubsurfaceID	integer > 0	none	ID for subsurface outflow series	
ExtInflowID	integer > 0	none	ID for external surface inflow series. Set to zero if no external inflow to be supplied	
DT	Integer fraction of 1 hour or multiples of 1 hour	hr	Time Step	
SurfArea	real > 0	ha	Total surface area	
MaxSoilWater	real > 0	mm	Soil water capacity	
FieldCap	real > 0	mm	Field capacity	
WiltingPt	real > 0	mm	Wilting point	
InfiltRate1	real > 0	mm/hr	Max surface infilt rate	
InfiltRate2	real > 0	mm/hr	Min surface infilt rate	
PercRate	real > 0	mm/day	percolation rate from soil water layer to groundwater layer	
GWOutflowCoeff	real > 0	mm/day	Subsurface outflow coeff	
DeepLossCoeff	real > 0	mm/day	Deep loss coeff	
SoilETFactor I	real > 0	none	Factor by which user-supplied potential evap raters are multiplied to get applied otential ET	
nitDepression	real >= 0	mm	Initial depression storage	
InitSoilWater	real >= 0	mm	Initial Soil water storage level	
InitGWVolume	real >= 0	mm	Initial groundwater storage level	

(Continued next page)

(PERVIOUS W STORAGE continued from previous page)				
Parameter	Value(s)	Units	Effect	Requirements
nStorageTable Entries	integer > 1, <=20	none	Number of entries (rows_ in the following depth-area-outflow table	
(Depth(i), Area(i), Outflow(i)),i=1, nStorageTableEntries	all real > 0	Depth in m, Area in m2, Outflow in M3/sec		• Depth, area and outflow must increase with each data set • Area(nStorageTableEntries)<=SurfArea
nCSTRs	integer > 0	none	the number of completely-stirred tank reactors to represent mixing conditions in the storage volume, for pollutant routing.	
SNOWMELT PARAMETERS			Same as GENERATE command	

4.3.19 The IMPERVIOUS W STORAGE Command

Purpose:

This command simulates hydrologic response of impervious (e.g. paved surface) that includes a surface storage volume. Intended to allow modeling of such features as flat roofs with surface storage and controlled drainage outlet; or paved parking area designed to provide surface ponding storage with controlled outlet.

Use:

This command was developed specifically for the Toronto Region Conservation Authority, although it can be used elsewhere if appropriate. It can be used in place of the GENERATE command. It is one of four such commands (PERVSURF, IMPERVSURF, PERVSURFSTORAGE, IMPERVSURFSTORAGE) which were implemented to facilitate simulation of LID alternatives. As well as this section, Appendix B provides some general guidance on the use of these four commands.

The surface storage receives all surface runoff generated by the impervious surface as well as a user-specified lateral inflow series which can be any available time series. The surface storage is characterized by a table of depth-area-outflow values supplied by the user. Losses from the storage include surface evaporation.

The command produces three binary outflow time series:

- OutflowID1= FlowFraction1 * surface runoff flow and pollutant load
- OutflowID2 = FlowFraction2 * surface runoff flow and pollutant loads
- OutflowID3 = FlowFraction3 * surface runoff flow and pollutant loads

There is a water balance summary in the text output file. There is also an optional detailed trace output. If **OutflowID1** is supplied as a negative value, then IMPERVSURF will generate a detailed trace output file (ASCII text format). This file is named "**IMPERVSTORSTEPnnn.txt",** where nnn = OutflowID1. It contains all water budget components (in mm) and the total surface and subsurface outflow rates (m^3/s) for each time step (one record per time step, easy to import into spreadsheet)

Numerical Values:

IMPERVIOUS W STORAGE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
OutflowID1	integer > 0	none	ID for surface outflow series #1. This flow series will be comprised of FlowFraction1 of the total surface runoff from the impervious surface.	
OutflowID2	integer > 0	none	ID for surface outflow series #2 which will be FlowFraction2 of the total surface runoff from the impervious surface	
OutflowID3	integer > 0	none	ID for surface outflow series #3 which will be Flowfraction3 of the total surface runoff from the impervious surface	
FlowFraction1	real > 0	none	Fraction of total surface runoff and pollutant load written as series OutflowID1	
FlowFraction2	real > 0	none	Fraction of total surface runoff and pollutant load written as series OutflowID2	
FlowFraction3	real > 0	none	Fraction of total surface runoff and pollutant load written as series OutflowID3	
ExtnflowID	integer > 0	none	ID for external surface inflow series	
DT	real > 0	hr	Time Step	
SurfArea	real > 0	ha	Surface Area	

(Continued next page)

(IMPERVIOUS W STORAGE continued from previous page)

Parameter	Value(s)	Units	Effect	Requirements
nStorageTableEntries	integer > 1, <=20	none	Number of entries (rows) in the following depth-area-outflow table	
(Depth(i), Area(i), Outflow(i)),i=1, nStorageTableEntries	all real > 0	Depth in m, Area in m2, Outflow in M3/sec		• Depth, area and outflow must increase with each data set • Area(nStorageTableEntries)<=SurfArea
SNOWMELT PARAMETERS			same as GENERATE command	

4.3.20 The EROSIVE Command

Purpose:

This command Computes the erosive impulse (excess boundary shear stress integrated over time) applied to a user-defined stream cross-section by a specified time series of flow rates.

Use:

This command was developed specifically for the Toronto Region Conservation Authority area, and is intended to provide a way to assess the shear stresses developed in a stream as a result of long term flows.

The stream cross-section is of trapezoidal geometry; the bottom width and side slopes of the cross-section as supplied as input to this command, along with the hydraulic roughness (Manning n value) and the longitudinal bed slope.

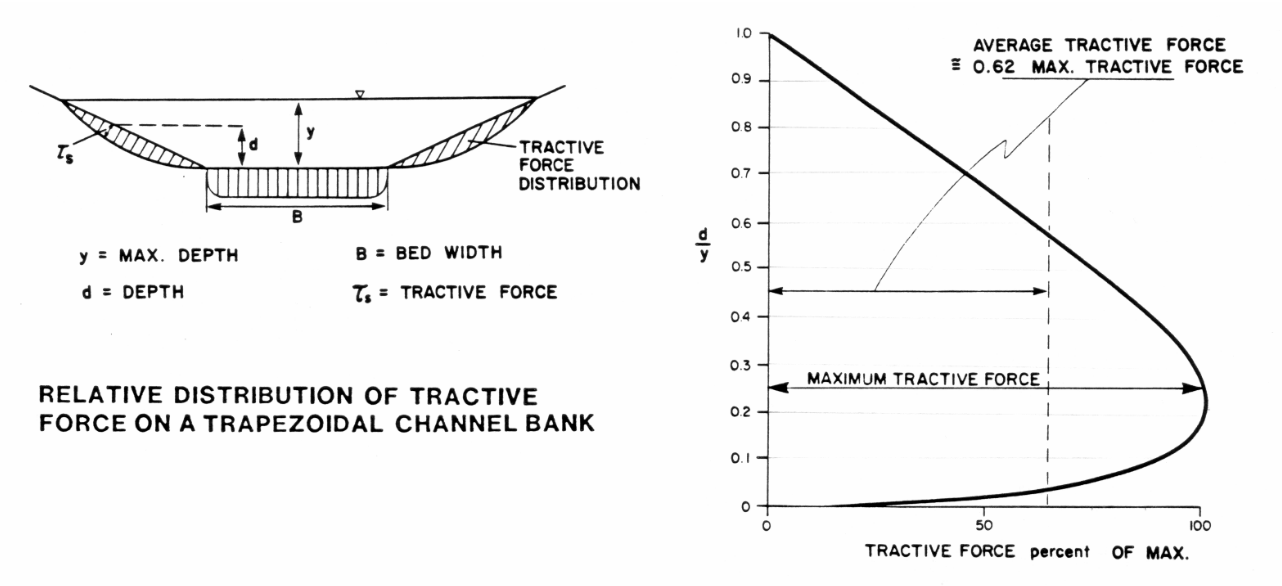
The input to this command must also include the critical tractive stress along the cross-section bed and banks. A single critical stress value must be supplied to characterize the channel bed. For the channel banks, the critical tractive stress values are supplied at equal intervals of depth above the channel bottom, per the input description below.

Generally, the critical stress values are estimated using field investigations that would include sampling and analysis of textural class of the bed and bank materials, and/or direct measurements of critical shear stress of bed and banks. Some empirical relationships between textural classification and critical stress is provided in Chow (1959).

This command functions as follows:

- In each time step, the subroutine first computes the depth of flow, the hydraulic radius and the velocity of flow, using the Manning equation with the user-supplied Manning n value and assuming that energy slope equals the user-supplied bed slope.
- The resulting tractive stresses on the channel bed along the wetted portion of channel banks are computed using empirically-derived equations documented in OMNR (1982). These equations are presented below for reference.
- In each time step, the subroutine checks if the applied bed and bank stresses are greater than the user-supplied critical values. If so, then the erosive impulse value is updated. The erosive impulse is simply the time-integral of the excess tractive stress. An impulse value is computed for the bed, and an impulse value is computed for each depth interval up the banks.

The relationships used to resolve this are well known. As shown below, in a figure taken from OMNR publication “Vulnerability of natural watercourses to erosion due different flow rates”, M.M. Dillon Limited, 1982, it is possible to develop an analytic solution of the distribution of tractive forces in this particular case.



RELATIVE DISTRIBUTION OF TRACTIVE FORCE ON A TRAPEZOIDAL CHANNEL BANK

Based on this case, the maximum applied tractive stress on channel banks is:

$$\tau_{Smax} = C_S \gamma R S_b$$

$$\tau_{Bmax} = C_B \gamma R S_b$$

τ_{Smax} = max tractive stress on channel banks (Pa); and

τ_{Bmax} = max tractive stress on bed (Pa); γ = unit weight of water (N/m^3)

R = hydraulic radius (m); S_b= channel bed slope (m/m)

$$C_S = (Z/2)^{0.36} \left[1.3 - 0.15 \ln (B/y)\right]$$

$$C_B = (Z/2)^{0.14} \left[1.42 - 0.019 \ln (B/y)^3\right]$$

Z = channel bank side slope (horizontal to vertical)
B = bed width (channel bottom width, m)
y = total depth of flow (m)

The distribution of applied tractive stress along the channel banks is:

$$\tau_S = \tau_{Smax} * \sin [\pi (d/y)^{0.45}]$$

where τ_S = tractive stress on channel bank at depth d above the channel bottom (m)

A sample input set for this command are below. The critical shear stress values are arbitrary, but the examples serve to illustrate the use of the command.

```
EROSIVE                  ID=-171   Start date 2005 05 28  End 2006 12 31
                         Number of depth increments = 50
                         Depth increment = 0.05 m
                         Bottom width    = 15 m
                         Sideslope       = 3
                         Slope           = 0.004
                         Manning N       = 0.05
                         Critical Bed Stress = 5
                         Critical Bank Stress Values
                           5  5  5  5  5  5  5  5  5  5
                           5  5  5  5  5  5  5  5  5  5
                           5  5  5  5  5  5  5  5  5  5
                           5  5  5  5  5  5  5  5  5  5
                           5  5  5  5  5  5  5  5  5  5
```

The text output file created by QUALHYMO will include the following output variables from the EROSIVE command:

- Total duration of analysis (hours)
- Total flow volume (hours)
- Maximum flow rate (m3/s)
- Maximum computed flow depth (m)
- Maximum computed bed stress and bank stress (Pa)
- Total bed impulse (cumulative excess shear stress on the channel bed, integrated over the duration of analysis), in Pa-hours
- Table listing the impulse on the channel banks at each depth increment (NDepth increments), in Pa-hours

If **IDin** is specified as a negative value, then there will also be generated a detailed trace output file (ASCII text) named EROSIVE.TRA, that presents results for each time step.

The result of the above input set would be as follows:

```
Results from EROSIVE POWER routine for series ID  171:
    -----------------------------------------------------------
    Total duration of analysis  =   13992.000 hours
    Flow volume                 =   38763.633 000s m3
    Maximum flow encountered    =     181.893 m3/sec
    Maximum computed flow depth =       3.312 m
    Max computed bed stress     =     127.285 Pa
    Max computed bank stress    =     111.987 Pa
    -----------------------------------------------------------
    Critical bed stress =   5.000 Pa
    Bed Impulse =       23241.355 Pa-hours
    -----------------------------------------------------------
     Depth   Critical     Bank Impulse
        bank stress
      (m)      (Pa)       (Pa-hours)
    -------  ----------  ---------------
     0.000      5.0000          0.000
     0.050      5.0000       9627.908
     0.100      5.0000      10281.417
     0.150      5.0000       9120.602
```

(etc, every 0.05 m)

```
     2.300      5.0000        149.373
     2.350      5.0000        138.283
     2.400      5.0000        127.900
     2.450      5.0000        117.536
```

References Cited:

- Chow, V.T., 1959. Open Channel Hydraulics. McGraw-Hill Book Company. Library of Congress Catlog Card Number 58-13860.
- Ontario Ministry of Natural Resources (OMNR), 1982. “Vulnerability of natural watercourses to erosion due to different flow rates”, report prepared by M.M. Dillon Limited, Consulting Engineers & Planners, Toronto, Ontario, dated Nov. 19, 1982, Dillon file # 9142-01, principal author F. Ivan Lorant, P.Eng.

Numerical Values:

EROSIVE Command Information				
Parameter	Value(s)	Units	Effect	Requirements
Idin	integer > 0	none	ID for time series of flowrates to be applied to the channel section.	
SYR, SMO, SDAY	integer > 0, integer >0,<12 integer >0,<32	none none	starting date of analysis	
EndYR, EndMO, EndDAY	nteger > 0, integer >0,<12 integer >0,<32	 none	Ending date of analysis	
Ndepth	integer>0		Number of depth increments at which to pompute tractive stress and impulse values	• Must be <51
dDepth	real > 0	m	Depth increment in metres.	
BW	real >= 0	m	Bottom width of trapezoidal cross-section	
SS	real > 0	m/m	Sideslopes of channel banks (horiz:vertical)	
BedSlope	real > 0	m/m	Longitudinal channel bed slope	
MN	real > 0	none	Manning n value	
CriticalBedStress	real > 0	Pa	Critical tractive stress on channel bottom (bed)	
CriticalBank Stress (n), n=1,NDepth	real > 0	Pa	Enter NDepth values of critical bed stress	

4.3.20 The CISTERN Command

Purpose:

This command simulates a simple rainwater cistern that is recharged by a specified time series, and depleted by domestic use.

Use:

This was developed specifically for the Toronto Region Conservation Authority geographic region, although it can be used anywhere. It was implemented to provide a way to simulate lot scale flow storage.

The specified cistern receives flow from a user specified inflow time series, and stores water until cistern capacity is reached. Once the cistern is full, all inflow is bypassed. Cistern storage is depleted at a user-supplied constant rate.

In the following simple example, a cistern is specified with a capacity of 1000 $m^{3.}$ It is initially empty. The daily withdrawal amounts are based on cistern water being used in summer months for irrigation.

```
CISTERN       Inflow ID = 1 Outflow ID = 91
              Cistern capacity = 1000 cu m
              Initial cistern storage = 0
              Daily withdrawal volume for each calendar month
              Jan 0.0     Feb 0.0     Mar 0.0
              Apr 0.0     May 65.0    Jun 133.0
              Jul 161.0   Aug 161.0   Sep 133.0
              Oct 0.0     Nov 0.0     Dec 0.0
```

Numerical Values:

CISTERN Command Information

Parameter	Value(s)	Units	Effect	Requirements
InflowID	integer > 0	none	ID for time series of inflow	
OutflowID1	integer > 0	none	ID for outflow time series (cistern bypass flow)	
CisternCapacity	real > 0	m^3	Cistern storage capacity	
InitCistern Storage	real > 0, <= Cistern Capcity	m^3	Initial volume in cistern	
DailyWithdrawal (nMonth), nMonth = 1, 12	real >= 0	m^3	DAILY withdrawal volume, one value for each month.	

5.0 Time Series Data

5.1 An Overview of the Time Series Data Sets

QUALHYMO is continuous simulation oriented and therefore relies heavily on time series files managed on the user's hard drive. Each time series data set is contained in a unique file. There are two categories of time series data files:

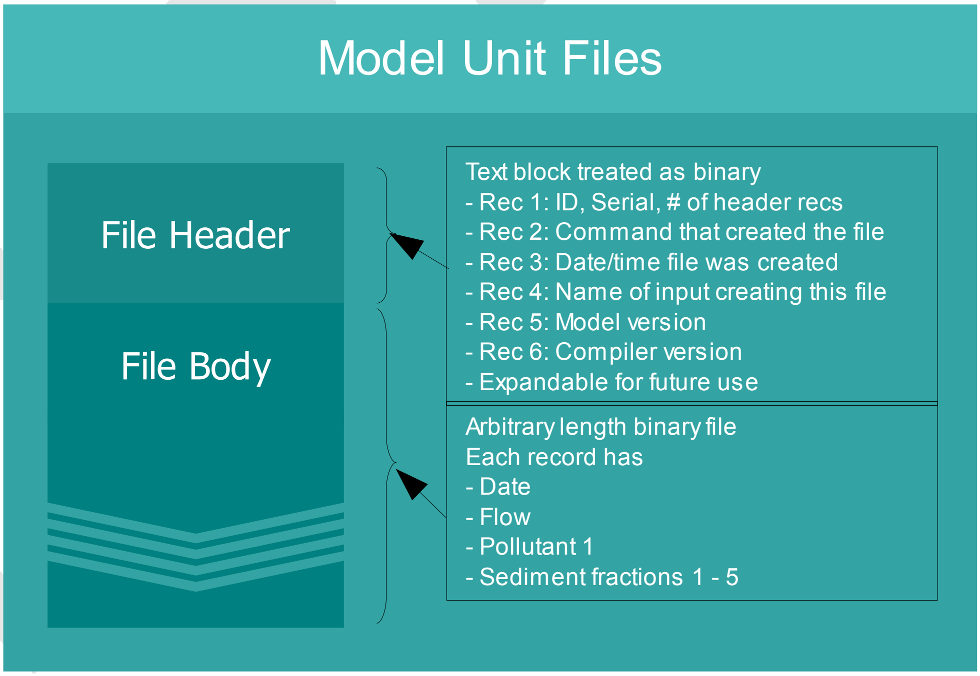
- Internal buffer files
- User input files

5.2 Internal Buffer File Structure

These are binary files that are designed to achieve high data transfer rates and small file sizes. They are structured in more than one way.

Model Unit Files

These files represent the input and output series from ADD, SPLIT, POND or REACH commands, and the output series from GENERATE commands. They are the main storage mechanism for QUALHYMO time series data. They consist of two segments, as shown below:



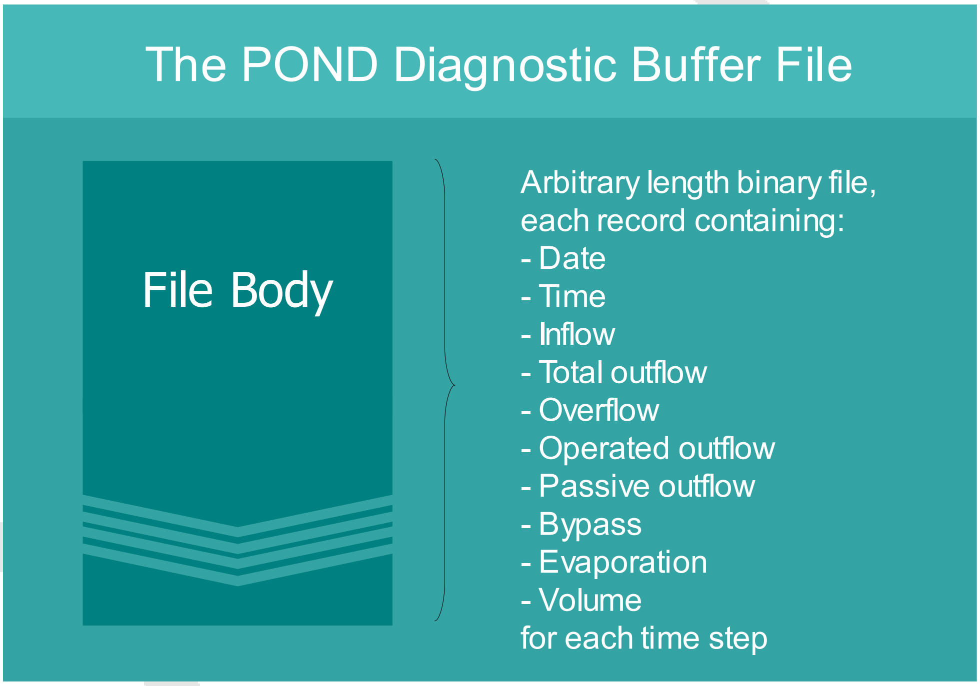
The header contains metadata that identifies file contents. The files are also named in a way that relates to the ID number associated with the file, to facilitate retrieval during a run. The body of the file contains the actual time series data, one record per time step.

> Warning: Valid ID numbers are integers from 1 to 200 inclusive. Values outside this range may give unpredictable results.

It is important to note that the model unit files remain on disc until over-written, so a user can (for example) do a GENERATE run once, and then repeatedly pass the resulting time series through a variety of POND or REACH configurations.

The POND Diagnostic Buffer File

This file is generated by the POND command if ISER<0 (see section 3.3.2), and remains in place until over-written. It is accessed by the PULL POND SPAN and CALC POND STATS commands to generate their results. It is similar to the model unit files in that it is a compact binary file, but contains no header, as shown below:



Since BMPs are a major QUALHYMO interest area, it is anticipated that this file will be accessed in a wider variety of ways in the future, but for now the PULL POND SPAN and CALC POND STATS make use of this file, and enable retrieval of data for analysis external to the QUALHYMO environment.

5.3 User Input File Structure

There are several files that fall under this category:

- rainfall
- temperature
- evaporation
- flow/pollutant series

All of these files are ASCII text files, each formatted in a common or available manner to facilitate user access.

Consistent with the intent of this draft manual, the evaporation file will be discussed further here. Other files are described in earlier manual versions, and will be described in subsequent releases of this manual.

5.3.1 Evaporation Files

These files contain evaporation information, as follows:

Year, (Evaporation(i), i=1,12)

The file is read as free formatted, so the model expects one integer followed by 12 floating point values, each separated by at least one blank (' '). An example of a few records from such a file is:

```
1911   0.00   0.00   0.00  41.50   97.70  159.00  178.00  146.50  103.60   52.00   0.00   0.00
1912   0.00   0.00   0.00  48.60  145.30  184.60  159.70  137.40   95.00   49.60   0.00   0.00
1913   0.00   0.00   0.00  56.80  117.90  128.80  175.80  140.00   93.40   46.10   0.00   0.00
1914   0.00   0.00   0.00  43.90  123.90  143.10  174.10  159.50  107.10   45.40   0.00   0.00
```

The model immediately converts this file into a time series file that is easily used by the model in subsequent calculations. For the present, this is an ASCII file that is formatted with one line per time step, as '(I4,2X,I2,2X,F8.2)', or

Year, Month, Evaporation

in a fixed format. This file is designated as a 'SCRATCH' file, and therefore only exists for the duration of a model run. Implementation of this file as a fast binary file which is persistent (exists between runs) is being considered as an extension of the model, and will be implemented if users indicate a preference for this alternative.

Appendix A

QUALHYMO Installation and Configuration Options

A.1 Ways to Run the Model

QUALHYMO can be run with or without command line parameters.

If Command line parameters are not used, then the model will run based on default control files and file locations.

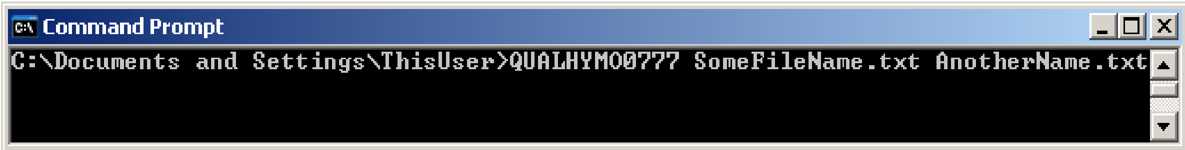
- The model command is simply its name. For example, if the model is located in a user directory 'ThisUser' in the 'Documents and Settings' directory, the model would be executed as follows:

- In this case,
  - the model will expect to find the control file 'QCONTROLFILE.TXT' in the directory the model is run from
  - the model will create a log file 'QRUNLOG.TXT' in the directory the model is run from.

If command line parameters are used then either one or two file names can be provided to the model. The model command with one parameter might be as follows:

- In this example
  - the model will expect to use the user specified name (anything the user wants but in this case 'SomeFileName.txt') as the control file.
  - the model will create a log file 'QRUNLOG.TXT' in the directory the model is run from.

The model command applied with two parameters might be as follows:

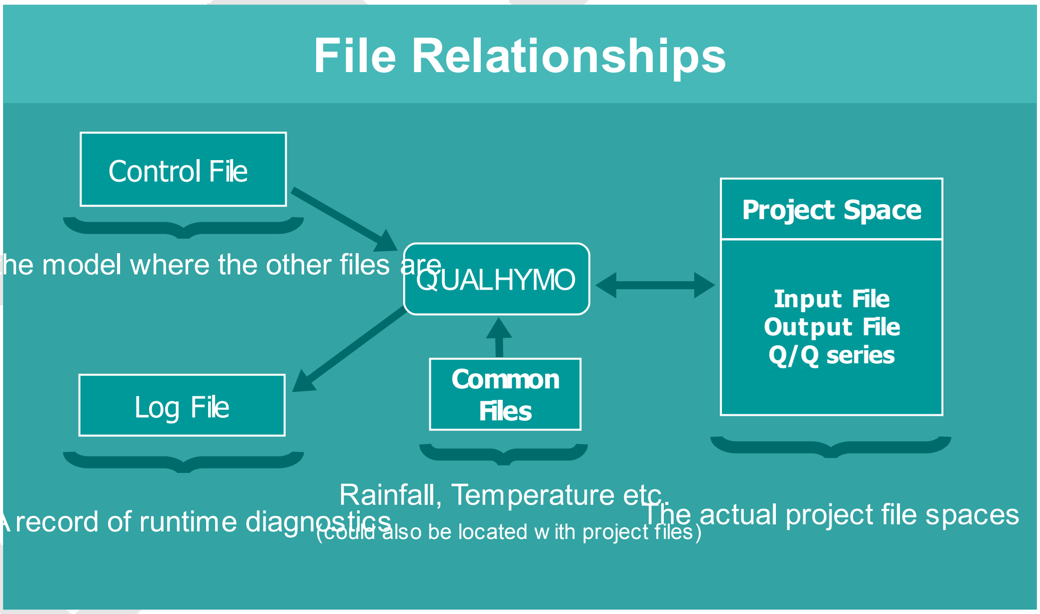


- In this example
  - the model will expect to use the first user specified name (anything the user wants but in this case 'SomeFileName.txt') as the control file.
  - the model will create a log file using the second user specified name (anything the user wants but in this case 'AnotherName.txt' in its local directory.

Note: It is not possible to specify a log file without specifying the control file.

A.2 Ways to Set up the Model Control Files

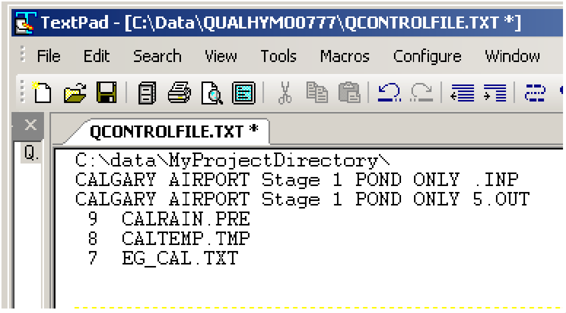
The model requires a control file which tells it where the various input and output files are located and what they are called. It also generates a log file that contains information on the events that happen during the run. The relationships between these files is illustrated below.



The Control File itself follows some basic rules:

- The file is an ASCII text file.
- The file can have any name and any extension that is legal under DOS but has a default name of 'QCONTROLFILE.TXT'.
- The content of the Control File is as follows:
  - **Path to working directory.** The working directory can be any location the user chooses. During the run, the model will look for all input files in that location, and will create its output files there as well. Two possible exceptions to this are the control file itself and the log file. The control file and log file locations are discussed in section 1.0.
  - **Name of input file.** This is the data file containing all the model commands and parameters. This name can contain embedded blanks but cannot have leading or following characters (except blanks and carriage return/line feed line terminators) that are not part of the name. This file is discussed further in section 3.0.
  - **Name of output file.** This is the file that will contain the model output generated during a run. This name can be any name legal under DOS, and can contain embedded blanks but cannot have leading or following characters (except blanks and carriage return/line feed line terminators) that are not part of the name.
  - **Up to 4 optional time series files, that can be listed in any order:**
    - **Number and name of rainfall file.** The number must always be 9 in this version of the model. The name can be anything the user wants but cannot contain embedded blanks. One or more blanks must separate the number and the name. This file is discussed further in section 3.0.
    - **Number and name of the temperature file.** The number must always be 8 in this version of the model. The name can be anything the user wants but cannot contain embedded blanks. One or more blanks must separate the number and the name. This file is discussed further in section 3.0.
    - **Number and name of the flow file.** The number must always be 10 in this version of the model. The name can be anything the user wants but cannot contain embedded blanks. One or more blanks must separate the number and the name. This file is discussed further in section 3.0.
    - **Number and name of the evaporation file.** The number must always be 7 in this version of the model. The name can be anything the user wants but cannot contain embedded blanks. One or more blanks must separate the number and the name. This file is discussed further in section 3.0.

A typical example of the control file as viewed in a popular text editor might be as follows:



The Log File is not controlled by the user except possibly its name and location as noted in section 1.0. Its characteristics are:

- The file is an ASCII text file.
- The file can have any name and any extension that is legal under DOS, but has a default name of 'QRUNLOG.TXT'.
- The content is as follows:
  - The first line is the time and date at which the model was run
  - The second line is the model version
  - The next few lines indicate successful opening of the control files and input files
  - Subsequent lines indicate events during the model run, such as commands or errors that may be encountered.
  - A final line indicates a normal run ending and the time and date at which that occurred.

An example run log might be as follows:

```
=== RUN BEGUN 28-Jun-07,02:30:35===
=== QUALHYMO VERSION 0777V1A BUILD 37 ===
=== CONTROL FILE OPENED
 - INPUT FROM: C:\data\MyProjectDirectory\CALGARY AIRPORT Stage 1 POND ONLY .INP
 - OUTPUT TO: C:\data\MyProjectDirectory\CALGARY AIRPORT Stage 1 POND ONLY 5.OUT
 - INPUT FILE OPENED
 - OUTPUT FILE OPENED
 - COMMAND TABLE GENERATED
 - READING FINAL CONTROL FILE ASSIGNMENTS
 - FILE C:\data\MyProjectDirectory\CALRAIN.PRE IDENTIFIED
 - C:\data\MyProjectDirectory\CALRAIN.PRE OPENED
 - FILE C:\data\MyProjectDirectory\CALTEMP.TMP IDENTIFIED
 - C:\data\MyProjectDirectory\CALTEMP.TMP OPENED
 - FILE C:\data\MyProjectDirectory\EG_CAL.TXT IDENTIFIED
 - C:\data\MyProjectDirectory\EG_CAL.TXT OPENED
=== CALCULATIONS BEGUN
  - RETURN FROM COMMAND INTERPRETER, NER,NCODE:  0  1
=== START COMMAND
  - RETURN FROM COMMAND INTERPRETER, NER,NCODE:  0  10
=== QUALITY PARAMETERS COMMAND
```

with further lines documenting what commands were run in what order, and finally ending as follows:

```
=== RUN ENDED NORMALLY 28-Jun-07,02:30:42 ===
```

It is not expected that most users will access the log file, since it is mainly useful in tracking input or system errors that may cause a model crash. The user specified output file will normally contain the required details of model behaviour and outputs. It is anticipated that the file will therefore be created in its default location and not routinely used. This is the reason that the optional addition of a non-default run log name and location is a command line parameter that is optional. However, users can decide for themselves if this file has material value and can manage it accordingly.

> Note: Although the internal model file handling structure is quite different from earlier versions of the model, users can still use the tools much as they have.
>
> - If they wish, users can locate all the files noted above in a single project directory, and little will have changed from earlier practices.
> - For production work, however, or to customize the model installation, users can locate files and re-use them (especially rainfall or other externally measured series) as they wish. Also, it is no longer necessary to port the executable to wherever the project files happen to be located.
>
> Flexibility has been increased, without making complexity a requirement.

A.3 A Word on Relative vs Absolute Addressing

Paths for the files used by the model can be established using standard DOS notation, which means that both relative and absolute addressing can be used. As an example, consider a case where:

- a project work space is in C:\DATA\DIR2\PROJECTLOC
- the model executable QUALHYMO0777.EXE is in C:\DATA\DIR1
- the control file is in C:\DATA\DIR1\CONTROLLOC and is called QFILE.TXT
- the user has opened a command line at C:\DATA\DIR1\RUNLOC

Since the path to the project workspace is “c:\data\dir2\projectloc\”, a suitable control file might be:

```
C:\DATA\DIR2\PROJECTLOC
REGEN.INP
TESTRESULTS.OUT
 9  RAIN.PRE
 8  TEMP.TMP
 7  EVAP.TXT
```

A suitable command line to run the model with all files as described above would be as follows:

```
Command Prompt
C:\Data\DIR1\RUNLOC>C:\DATA\DIR1\QUALHYMO0777 C:\DATA\DIR1\CONTROLLOC\QFILE.TXT
```

The example above uses absolute addressing. There is another option, which is to use relative addressing. In the case at hand, an equivalent form of the above command line would be:

```
Command Prompt
C:\Data\DIR1\RUNLOC>..\QUALHYMO0777 ..\CONTROLLOC\QFILE.TXT_
```

The usefulness of relative addressing depends very much on how the user happens to have placed the various model files on their system. ***In case of doubt, absolute addressing is recommended.***

Relative addressing is not confined to the command line. Relative addressing can also be used in the path specified in the first line of the control file. If, for example, all files are located in a single directory "C:\Documents and Settings\ThisUser\ as discussed above, then the control file might be:

```
.
REGEN.INP
TESTRESULTS.OUT
 9   RAIN.PRE
 8   TEMP.TMP
 7   EVAP.TXT
```

where the path on the first line is simply a period or decimal symbol ".", and the command line becomes:

```
Command Prompt
C:\Documents and Settings\ThisUser>QUALHYMO0777_
```

Appendix B

Representing Some Common Features with the PERVIOUS SURFACE, IMPERVIOUS SURFACE, PERVIOUS W STORAGE and IMPERVIOUS W STORAGE Commands

The QUALHYMO commands created to serve emerging requirements in the TRCA context can be employed to represent site-level measures that promote infiltration and provide physical filtration to help with pollutant load reduction. The four hydrologic commands have quite a bit in common, but differ in detail and intended use. It is up to the user to establish how to best represent a BMP in their particular case, but this section provides some insights into ways they might consider as they decide on their simulation strategy.

Command	Typical Uses
IMPERVIOUS SURFACE	Used to separately simulate different types of hard surfaces, including roadways, parking lots and roofs. These types of surface can be simulated separately by repeated use of this command so that different mean pollutant concentrations can be applied to each surface type. Changing pollutants according to land use type can be accomplished by preceding each use of IMPERVIOUS SURFACE with appropriate POLLUTANT SERIES command to apply appropriate pollutant concentrations.
PERVIOUS SURFACE	Used to simulate vegetated areas including landscaped areas. This command can be used to represent vegetated yard areas that receive roof drainage in residential areas. This command has been coded so that any existing (previously generated) flow time series can be supplied by the user as a lateral input to the surface (e.g. to roof runoff draining onto a grassed area).
IMPERVIOUS W STORAGE	Simulates impervious (e.g. paved surface) that includes a surface storage volume. Intended to allow modeling of such features as flat roofs with surface storage and controlled drainage outlet; or paved parking area designed to provide surface ponding storage with controlled outlet. The surface storage is characterized by a table of depth-area-outflow values supplied by the user. Losses from the storage include surface evaporation.
PERVIOUS W STORAGE	Used to simulate grassed swales and soakaway infiltration facilities. This command has been coded so that any existing (previously generated) flow time series can be supplied by the user as a lateral input to the surface.

These commands complement the QUALHYMO GENERATE command by providing alternate means of simulating surface runoff processes within smaller-scale situations. The GENERATE command is intended more for simulating the hydrologic response of larger watershed areas. Accordingly, GENERATE makes use of estimates of effective imperviousness over larger areas, and uses a modified SCS Curve Number approach that uses Antecedent Precipitation Index (API) to provide continuous simulation of surface runoff production at the watershed scale.

Separate representation of impervious and pervious surfaces, and allowing for surface runoff from one to be diverted onto the other, provides a more explicit means of modeling urban processes, especially when dealing within smaller-scale areas such as individual development areas or properties. This was the impetus for extending QUALHYMO's command set to include these four commands.

Notes on these commands are as follows:

1. In the case of the IMPERVIOUS SURFACE command, the command has been coded such that the surface runoff and pollutant loadings can be split into as many a three fractions. This allows, for example, one fraction to be diverted onto a pervious vegetated surface (e.g. roof drainage onto grassed area) while another fraction goes directly to storm outlet.
2. Both the PERVIOUS SURFACE and PERVIOUS W STORAGE provide continuous simulation of soil moisture status within and vertical water movement through a porous soil profile. The user must supply the following input parameters
   - Surface area in hectares
   - Maximum and minimum surface infiltration rates. The actual surface infiltration capacity at any point in time varies between these two rates depending on soil moisture status.
   - The soil moisture storage capacity, field capacity and wilting point. These are input as mm of water depth. The storage capacity is a function of the depth of the soil profile and the available porosity. Field capacity represents that amount of water that can be held against gravity drainage (*i.e.* held within the soil matrix by capillary tension). Wilting point is the amount of water held in the matrix at the point at which vegetation root systems can no longer extract moisture. Storage capacity, field capacity and wilting point will vary with soil texture. As well, when expressed as mm of water, the values will depend on the depth of the soil profile being modeled. Since all ET losses are accounted for within this layer, the depth should be at least equal to the full depth over which the vegetation's root system can extract water.
   - Percolation rate. This is the rate at which free water *(i.e*. amount of water in excess of field capacity) can move gravitationally downward through the profile and into the underlying layers, which are represented in the model as a conceptual groundwater storage reservoir. In other words, percolation is groundwater recharge. The percolation rate should be based on the limiting saturated hydraulic conductivity of the soil layers through which the water must percolate to reach the water table. The model assumes that this percolation capacity is always available.
3. In all of these commands, the total liquid input to the surface will be comprised of rain plus snowmelt, plus any lateral inflow as specified by the modeler. The lateral inflow could, for example, be the surface runoff outflow from an IMPERVIOUS

SURFACE.

4. In the PERVIOUS commands, actual evapotranspiration is computed based on potential ET rates supplied as input to the model. When soil moisture is at or above field capacity, actual ET will be equal to potential ET. Once soil moisture drops below field capacity, actual ET will be lower than the potential rate; the actual ET rate approaches zero as soil moisture approaches wilting point.
5. In the case of the PERVIOUS SURFACE command, when the total liquid input to the surface exceeds surface infiltration capacity, then surface runoff is generated. The rate of runoff (flowrate) is based on surface roughness (Manning n value), slope and characteristic length supplied by the user.
6. In the case of PERVIOUS W STORAGE, the user does not supply surface slope, length and roughness. Instead, a surface storage element is defined using a table of depth-area-outflow values that the user must compute and supply to the model. When total liquid input exceeds surface infiltration capacity, water will be stored within the surface storage element. The model continuously tracks the water depth, inundated area and volume within the surface storage element; and surface evaporation and exfiltration of stored water into the soil profile occur over only the inundated area.
7. Total outflow from the conceptual groundwater storage layer is computed as a simple linear function of storage level. The modeler can specify that a percentage of this outflow be considered as “deep losses” which do not contribute to the subsurface outflow rate.

The PERVIOUS commands do not provide any simulation of pollutant removal by filitration or other processes associated with vertical water movement downward through the soil matrix. However, the effect of this filtration process can be represented by using the POLLUTANT SERIES command to assign appropriate average pollutant concentration values to the subsurface (groundwater) outflow.

Note that in the case of the PERVIOUS W STORAGE command, pollutant removal (solids settling and first-order decay) is simulated within the volume held in the surface storage element, in the same manner as used in the POND and REACH commands. That is, the surface storage volume is represented as a number of completely-mixed tank reactors, with solids settling and first-order decay occurring based on rate parameters supplied with the START command.The PERVIOUS commands provide a conceptual representation of infiltration into, movement through and ET losses from a porous medium. These commands can therefore be used to simulate hydrologic response of vegetated areas on native soils.

As well, through appropriate values for input parameters, the PERVIOUS SURFACE and PERVIOUS W STORAGE commands can be used to represent various site-level BMPs such as infiltration trenches, vegetated swales, soakaway areas, subsurface infiltration galleries or bioretention facilities. Refer to the following table for an outline of how some commands and BMPs might be related.

Type of BMP	QUALHYMO Command	Notes on Input Parameters
Soakaway area: Designed as area for temporary surface ponding on native soils	PERVIOUS W STORAGE	• Storage depth-area-outflow table based on actual surface grading and estimate of ponding depth before spill outflow occurs. • Surface infiltration capacity and percolation capacity based on estimated saturated hydraulic conductivity of soil layers. • Lateral inflow time series is the inflow from adjacent areas
Infiltration trench or infiltration gallery: Designed using coarse granular filtration media surrounded by geotextile; allows for water to exfiltrate into surrounding native soils. May include an overflow pipe to collect water in excess of exfiltration capacity	PERVIOUS SURFACE	• Lateral inflow time series is the facility inflow • Surface area equal actual trench surface area • Surface infiltration rates set to represent high capacity for water to enter top of granular matrix. • Soil moisture holding capacity based on depth, width and effective porosity of granular matrix. • Field capacity minimal and wilting point zero (minimal water held in matrix by capillary potential). • ET factor set to zero, as no vegetative root system within granular matrix. • Percolation rate set to value representative of saturated hydraulic conductivity of surrounding native soil.
Bioretention facility: Designed to provide some surface storage ponding capacity, either on native soils or on granular fill material intended to promote infiltration	PERVIOUS W STORAGE	• Similar to soakaway area (above) • Storage depth-area-outflow table based on actual surface grading and estimate of ponding depth before spill outflow occurs. • Surface infiltration capacity based on estimated conductivity of surface soil layer or granular material • Percolation rate based on estimated conductivity of underlying soil layers
Grassed swales: Linear vegetated swales allow for infiltration of water into native soils	PERVIOUS W STORAGE	• Storage depth-area-outflow table based on average or typical swale cross-section, hydraulic roughness and bed slope to develop flow-vs-depth rating curve, which can then be used to estimate depth-area-outflow from length of swale. • Surface infiltration capacity and percolation capacity based on estimated saturated hydraulic conductivity of soil layers.

The IMPERVIOUS and PERVIOUS commands can accept lateral inflow time series as part of the total liquid water input to the surface. And, as noted above, surface outflows from IMPERVIOUS SURFACEs can be divided into three fractions. This allows for a good deal of flexibility in representing drainage connectivities that may be of significance at the urban site level.

A schematic of how this can be accomplished is shown below. This example is a model of a residential development site that includes an infiltration facility that receives all surface runoff from the site area. In this example,

Separate IMPERVIOUS SURFACEs represent roofs versus roadways and parking areas. The roof runoff is split into two equal fractions, one of which is used as lateral input to a PERVIOUS SURFACE that represents grassed yard areas that receive the roof runoff.

- A separate PERVIOUS SURFACE represents other vegetated area.
- The ADD SERIES command is then used to add up the surface runoff from the various surfaces to generate a single time series representing the total surface runoff from the area.
- This time series is then used as lateral input to a PERVIOUS WITH STORAGE that represents the infiltration facility itself. In this case, the infiltration facility is within a park area and consists of a surface ponding area over top of a granular fill matrix that allows for exfiltration into the surrounding native soils. The outflow from the surface storage element represents the net surface runoff from the site.
- The ADD SERIES command is used to add up the subsurface outflow components from each of the PERVIOUS SURFACE elements, to get the total subsurface outflow from the site.

Note that QUALHYMO generates a standard text output file that provides a water balance summary for each individual PERVIOUS or IMPERVIOUS element. These summaries include total volumes for liquid water input, surface evaporation, infiltration, ET from the soil layer and percolation to the conceptual groundwater layer, along with mass balance checks (i.e. continuity error checks). The modeler can then take the volumes from these summaries to compute the overall water balance for the site.