
Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh

**25th Annual Progress Report
October 2004**

Chapter 7: Extensions and Improvements to DSM2

Author: Ralph Finch, Eli Ateljevich, Edward Diamond, and Tawnly Pranger

7 Extensions and Improvements to DSM2

7.1 Introduction

The DSM2 database extension project was described in the 2002 Annual Report (Ateljevich and Pranger, 2002). Since then, the project has increased in scope to include other major changes. This chapter reviews the database portion of the project, describes the other improvements, and provides a status report of the project. The new version of DSM2, referred to as DSM2-DB, includes features in addition to the database.

7.2 Major Extensions in DSM2-DB

The conversion of DSM2-DB to read fixed (non-time-varying) data from a **relational database** instead of from text files continues to be the main extension of DSM2-DB. Reasons for embarking on the Database component include:

- ❑ Providing better guarantee that different studies using the same Delta design elements will use identical parameters,
- ❑ Allowing easier implementation of a Graphical User Interface (GUI),
- ❑ Providing a mechanism for developing and testing REALM (Ateljevich and Finch, 2004), and
- ❑ Shifting to industry-standard methods and software wherever feasible, which reduces development costs and offers a much higher quality end product.

The Microsoft Access database was initially chosen to store DSM2 input data because of its low cost and ubiquity. Some initial problems with multi-user access and replication led to the decision to move to a more robust multi-user database system, with Access being retained as the public, single-user database. Firebird (the open-source version of Borland's Interbase) was chosen as the Section database until an Informix server is available later in the year. A Visual Basic script was used to move both the data structure and contents from MS Access. Another script will be constructed to move the final data structure and contents from Informix to Access (round-trip capability) to facilitate data sharing.

Firebird has worked out well as a development platform for a multi-user database. Its built-in stored procedure language provides an easy implementation of data integrity and check constraints. The security system allows individual modelers to create their own related data layers that are secure from modification by other database users. This security system has been comprehensively tested to insure that it works as intended.

7.2.1 Graphical Users Interface (GUI)

A closely related extension is the development of a **companion GUI** to the Database. This is also described by Ateljevich and Pranger (2002) and the appearance has not changed greatly from that report. In addition to facilitating manipulation of data in the Database, the GUI enables easier review of changes from an initial study.

For instance, Figure 7.1 shows the *Gate Time Series* input panel (right frame of the image in Figure 7.1) for an experimental permanent barrier planning run. To simplify the test run, the Montezuma Slough Salinity Control Structure was not supported by removing the time series input for its gate operation from the input in the *Planning Operation Gates* layer (lower left *Input Layers* panel in Figure 7.1).

The same *Gate Time Series* panel shows that, for the Old River at Head barrier, the operation of the flashboards and radial gates have been over-riden from a timed operation from a DSS file to a fixed, always-open state, as evidenced by the light-gray text of the superseded condition.

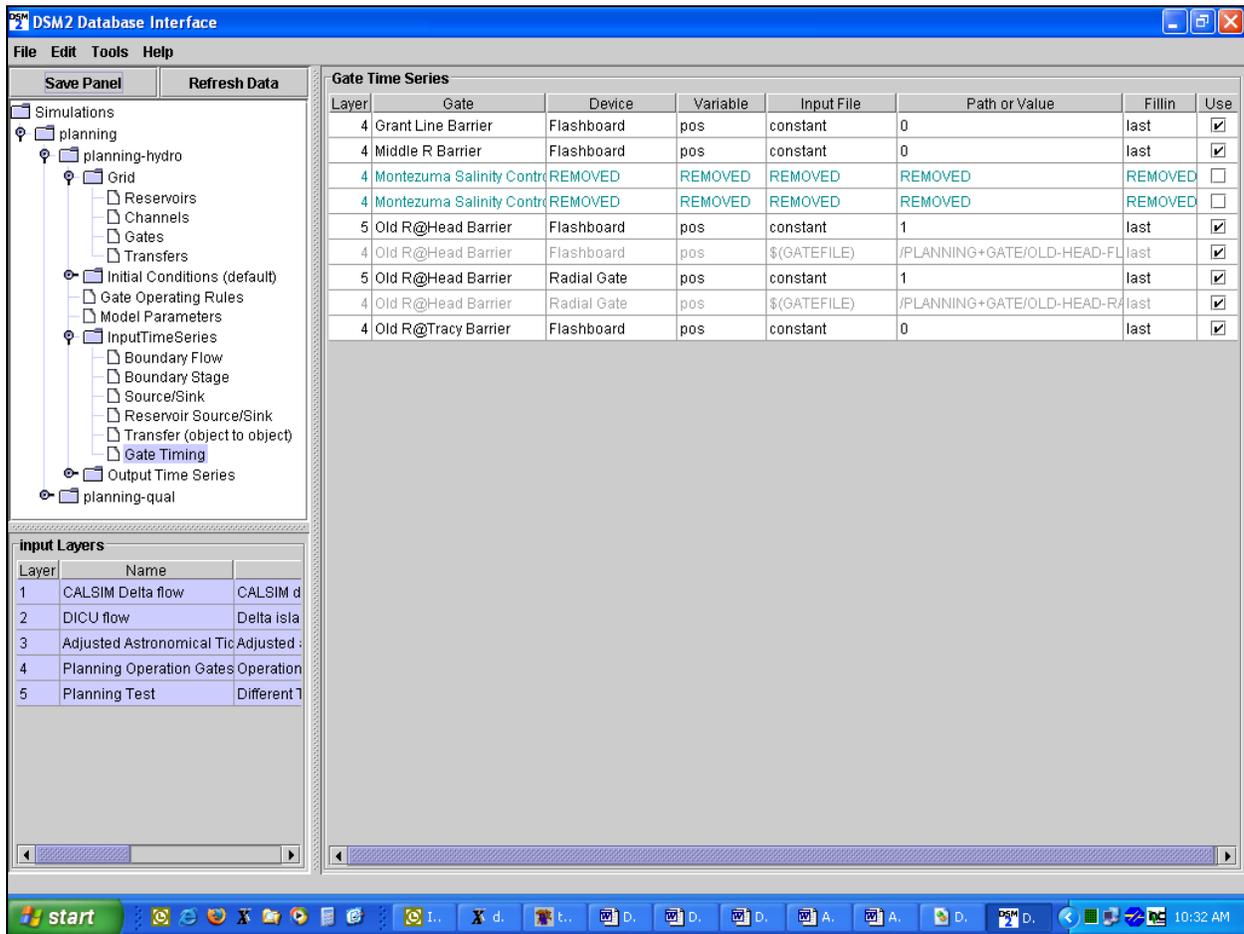


Figure 7.1: Gate Timing Panel of DSM2-DB GUI Illustrating Entries Removed and Changed with Higher Layers.

GUIs are helpful for making relatively small additions or changes to input, but are inconvenient for manipulating large amounts of data, or for repetitive tasks. Scripts have been developed that access the database, interact with its security system, and add data in bulk. The prototype application was a script that stores channel cross-section data generated by the Cross-Section Development Program (Tom, 1998).

7.2.2 Gates

Other major extensions in DSM2-DB are more flexible **gate specifications** and **reformulation of internal gate calculations**. In the current version of DSM2, gates have only one basic weir and/or pipe configuration (with multiple numbers of identical weirs and pipes). Gates can be located at channels or reservoirs, but the internal representation of channel and reservoir gates are different and reservoir gates such as the Clifton Court Forebay gates have much less functionality than channel gates.

The following features have been applied to gates:

- ❑ Gates now are collections of *devices* such as pipes and weirs.
- ❑ Devices contain the physical dimensions and parameters of what has until now been called gates.
- ❑ An arbitrary number of devices can be added to a single gate and operated independently. Physical description and properties (width, flow parameter) are applied to each device in its device description, not to the gate as a whole as in previous versions of DSM2.

Figure 7.2 illustrates a *Gate Input* panel. Higher numbered layers override specifications in lower layers. The pipe and weir devices contained in a gate are shown in the bottom half of the right panel. In this example, the Boat Lock for the Head of Old River barrier is displayed.

These modifications will satisfy a number of user requests that have emerged over the last two years. For instance, the five gates to Clifton Court Forebay can now be operated independently. The radial gates, flashboards, and boat lock of the Montezuma Salinity Control Structure can likewise be configured and operated independently. Finally, the option to treat duplicate pipe and weir devices as a single group is still available to preserve simplified gate treatment due to lack of detailed data.

Gate flow calculations have been reformulated and transformed in a manner that is mathematically equivalent and more amenable. This should improve gate flow accuracy and allow true gates to be connected to reservoirs. The older and simpler connection to channels has been retained for reservoirs with no actual gates, but the new method is used for the Clifton Court Forebay gates.

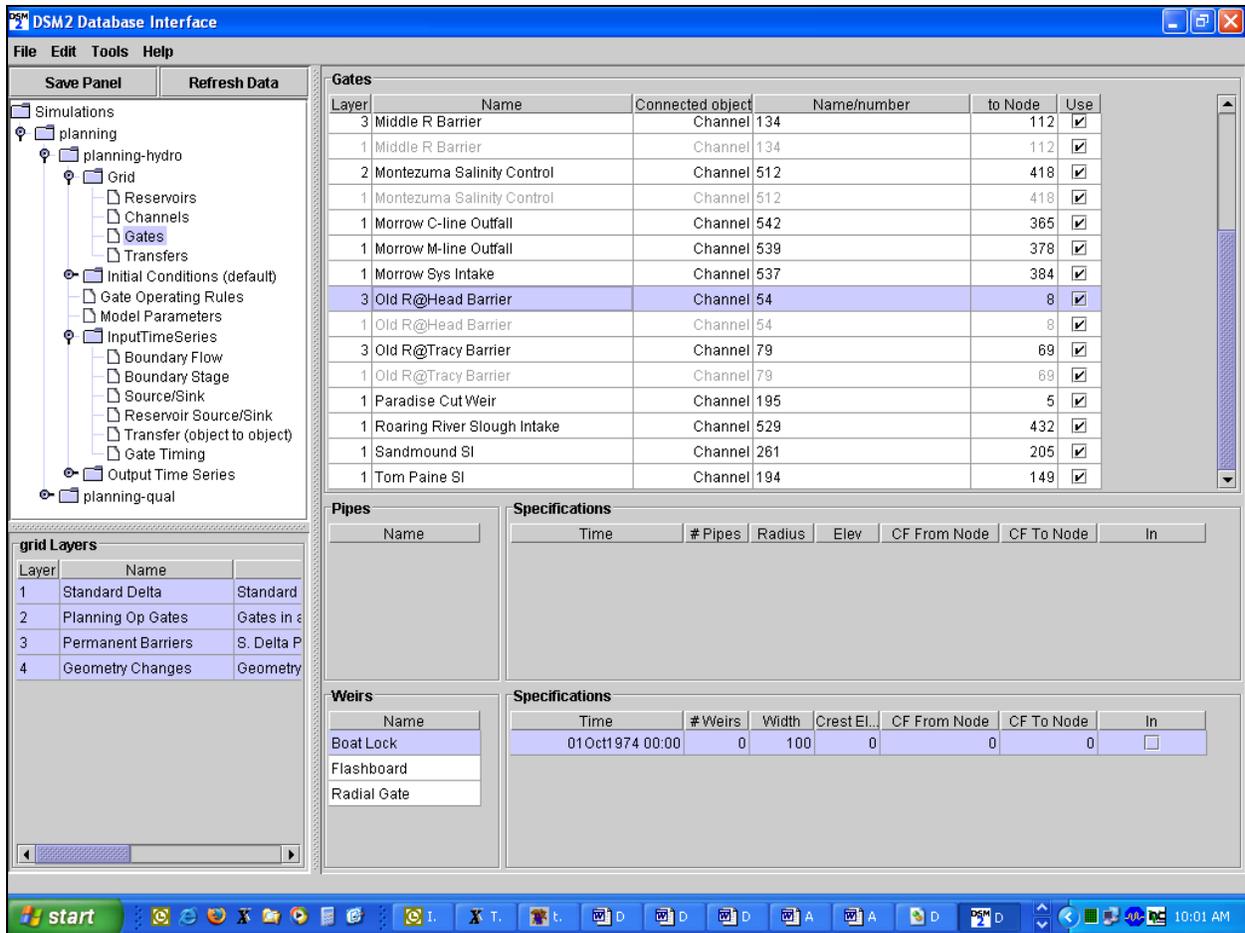


Figure 7.2: Gate Input Panel Illustrating Gate Devices and Layers.

7.2.3 Expressions and Operating Rules

Another major addition to DSM2-DB are **Operating Rules**: a *trigger* and subsequent *action*. Operating rules allow the user to model features such as gate operations and pumping changes which in the field would be managed adaptively based on conditions in the Delta. An example is to close a radial gate in response to low stage in Middle River. This is not possible in the current version of DSM2 where adaptive decisions are simulated through an iterative process.

Operating rules for gates have been available in test versions of DSM2-DB for several months, and have been applied to trial planning and historical simulations. Early versions of operating rules allowed the user to adjust the operating coefficient of a gate based on simple observations of the state of the model, such as a single flow or stage value. For the Middle River barrier, for example, it is possible to trigger the opening of a radial gate when stage in a reference channel dips below a threshold value. The trigger can optionally include a short-term anticipation (by extrapolating the current trend in the model forward in time). In DSM2-DB, triggers can only be associated with hydrodynamic parameters, such as stage, stage difference between locations or on each side of a gate, flow, or channel velocity, at user-specified locations. A trigger can also be empty, which means it is applied at the start of a run and continues indefinitely.

Recently, operating rules have been redesigned and augmented in response to feedback from trial applications. In order to increase the flexibility and expressiveness of the rules, the operating rules utilize both model information and exogenous input such as time series and information about the season. These inputs can also be combined in simple mathematical or logical expressions, examples of which are given below. The operating rule actions can manipulate boundary flows and pumping as well as gates.

The operating rules require an interface that neither limits nor overly complicates their use. DSM2-DB uses a simple interpreted language to write the rules since this is the standard for user-written rules (e.g. SQL for databases or WRESTL for CALSIM), and menu-driven control within a GUI is not flexible enough. Operating rules combine *trigger* and *action* directives, each of which is an expression based on observed model states, seasonal information and exogenous time series input, as well as other triggers and actions.

Actions are responses the model executes when its corresponding *trigger* fires. In DSM2-DB, the actions involve either gate devices or source/sink flow terms (often pumps or drains). For devices, the operating flow coefficient can be changed, as well as the maximum flow allowed through the gate device. For sources and sinks, flow may be specified by either a constant value or a time-series of values.

Expression Examples

An example of a simple numerical expression based on current DSM2-DB flow looks like this:

```
ebb := flow(channel 132, dist 1000) > 0.01
```

This example samples the current time step model flow 1,000 ft downstream of the upstream node in channel 132 and checks whether it is greater than 0.01 cfs. The expression assigns the answer the name *ebb*, so it can be reused. Note that *ebb* is a logical expression which evaluates to *true* or *false* depending on the model time step. Logical variables usually appear in triggers rather than actions.

Besides logical expressions, numeric expressions involving simple math operators can also be defined. For instance:

```
ebbmagnitude := exp(flow(channel 132, dist 1000))
```

is an expression that evaluates flow, applies the exponential function to it and then assigns it to the variable name *ebbmagnitude*.

Model time can also be used in expressions. The following expression describes the VAMP season for San Joaquin river management:

```
vamp := (month == Apr) or (month == May)
```

The definition could also include the date, day of the month, or time of day.

Finally, the following example combines a model state (stage/water surface) observation, an external time series (called `tide_level`) and simple arithmetic. The expression might be used with a slowly fluctuating tide or sea level datum to provide an idea of critical stage in the South Delta compared to ambient tide conditions.

```
critical_stage := stage(channel 132,  
                        dist 1000) < (tide_level - 1.0)
```

Operating Rule Examples

It is now straightforward to use expressions in operating rules. The following example is based on expressions that were developed above. **Bold face** words are part of the GUI; *courier* type is user input.

Name: `middle_vamp_ebb`

Expressions:

```
ebb := flow(channel 132, dist 1000) > 0.01  
vamp := (month == Apr) or (month == May)
```

Trigger: `vamp` and `ebb`

Action: `set(weir-op, gate: Middle River Barrier,
 weir: Radial Gate) to new_time_series`

The *middle_vamp_ebb* operating rule lies dormant until the first time step when *vamp* and *ebb* (a compound expression based on the expressions **vamp** and **ebb**) becomes true. At that point the action will be taken and the weir operating coefficient will start to operate according to the values in the DSS time series *new_time_series*.

Anticipation and ramping can be added to numerical expressions in triggers and actions respectively. For instance, an anticipating version of *middle_vamp_ebb* might use the definition for *ebb*:

```
ebb := predict(flow(channel 132, dist 1000), 45min) > 0.01
```

This example assigns a logical expression *true* or *false* to *ebb* based on the whether the flow 1,000 ft downstream of the upstream node in channel 132 that is predicted 45 minutes from the current time step is greater than 0.01 cfs.

Similarly, the user may wish to gradually introduce the action. This is done using the *ramp* directive, which gradually and linearly implements an action:

```
set( weir-op, gate: Middle River Barrier,  
     weir: Radial Gate) to new_time_series ramp 60min
```

Often, an operating rule is paired with a complimentary rule that will reverse its action. For instance, to complement the above rule for ebb flow the following operating rule for flood flow might be added:

Name: middle_vamp_flood

Expressions:

```
flood := flow(channel 132, dist 1000) < -0.01  
vamp := (month == Apr) or (month == May)
```

Trigger: vamp and flood

Action: set(weir-op, gate: Middle River Barrier,
weir: Radial Gate) to old_time_series

This rule effectively undoes the ebb action. This example underscores a necessary but somewhat unintuitive point about triggers: they are one-time and unidirectional. A rule whose trigger is *vamp and ebb* will activate when this expression changes from false to true but will not deactivate or even notice if *vamp and ebb* subsequently becomes false again. If the complementary behavior is desired, this intent must be specified in a second rule. Often the complementary rule is subtly different from the exact negation of the original; for instance, the trigger *vamp and flood* is not the same as *not(vamp and ebb)*. An important example of this in Delta operations is the Montezuma Salinity Control Structure, when the flood and ebb triggers are not even based on the same variable (the gate is opened based on a head difference, closed based on velocity).

The *middle_vamp_ebb* example combines *vamp*, which is the seasonal applicability of the rule with *ebb*, which is a tidal phenomenon. There are also meaningful operating rules that do not need a trigger at all. For instance, the user might want to operate SWP and CVP pumping based on a time series but bound it by some fraction of Sacramento inflow. The trigger in this case always applies, which is the default in the GUI if you leave the trigger blank. The rule will then read:

Trigger:

Action: set swp = max(swp_time_series, outflow_fraction)

7.2.4 Data Format

The last major change to DSM2-DB is the use of **HDF5 (Hierarchical Data Format, <http://hdf.ncsa.uiuc.edu/HDF5/>)** for hydrodynamic output for QUAL and the PTM.

HDF5 is a general purpose, open source library and file format designed for storing scientific data. It was designed for high performance, data intensive applications and includes compression and support for parallel systems.

The advantages in using HDF5 include generalization, flexibility, and support for large and complex datasets. It can be run on a wide variety of computing systems, ranging from desktop PCs to parallel supercomputing systems.

These aspects of HDF5 provide substantial improvements over the existing tidefile which must be written to and read from in sequence. This requires the file (currently 4 GB in size for a 16-year planning study) to be read beginning to end multiple times. In contrast, data within an HDF5 file can be randomly accessed, allowing for much more efficient retrieval and for very flexible subsets of the entire data sequence to be quickly retrieved. This flexible access allows for a more thoughtful design of how data is stored. The sequential format prevents multiple processes from accessing the same tidefile simultaneously which is not a limitation of HDF5.

Because it supports compression, there is an additional space savings using HDF5. With the use of compression, the size of the existing tidefile is reduced from 4 GB to 2.5 GB. This may enable DSM2-DB to store more data within the file and allow for the use of a single file with the Condor parallel application of the model (see section 7.2.5 for more information about Condor).

As of the date of this report, DSM2-DB is capable of writing to and reading from an HDF5 file. The current file contains most of the data stored in the existing tidefile structure (the exception being an object-to-object transfer).

7.2.5 Parallel Processing

HYDRO runs 50% or slower than QUAL, and thus has been the bottleneck in producing studies. The **Condor distributed computing system** (<http://www.cs.wisc.edu/condor/>) was implemented on the Delta Modeling Section's local area networked computers. Using Condor's Directed Acyclic Graph Manager (DAGMan) a single HYDRO run can be split into five simultaneous runs (one warm-up year and four production years) on five computers on the LAN. When each run finishes, its output is copied back to the submitting computer. QUAL is started when all runs are finished, and the separate HYDRO outputs are combined into a single DSS file (Figure 7.3). This allows a joint HYDRO and QUAL run to be finished in slightly less than 8 hours using a 3 GHz processor, and should allow reasonable running times for longer runs (e.g. 73-year run on full CALSIM results).

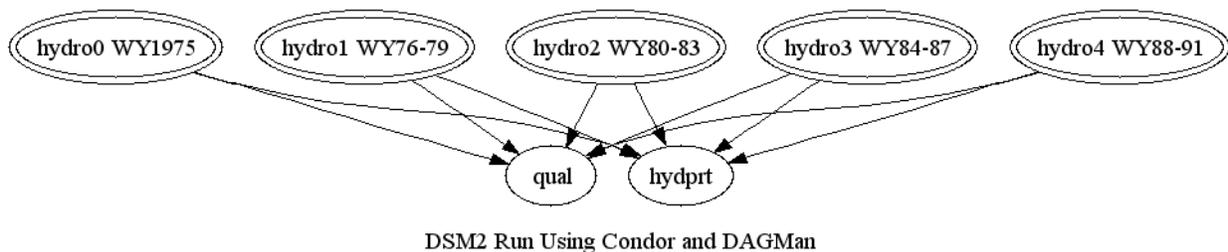


Figure 7.3: Parallelized HYDRO Runs.

7.3 Status & Future Directions

As of April 2004 DSM2-DB HYDRO and QUAL are both running. However, three types of basic runs need to be completed before this version of DSM2 can be placed into production:

- ❑ Historic, to compare to previous validation,
- ❑ Planning with South Delta agricultural temporary barriers, and
- ❑ Planning with permanent South Delta agricultural barriers operated during run.

Each run offers different challenges.

Because better gate descriptions are now available, old gate descriptions may not be adequate. The gate calculations themselves have been changed and thus a small-scale validation of the new gate configurations is needed to confirm that the model results are basically the same as before.

The timing of temporary barriers can be determined before the run starts from San Joaquin River boundary flows. This is being done now with an Excel worksheet. For DSM2-DB the logic behind temporary barrier operation will be converted to a Vscript program which will generate the needed timing and write to a DSS file.

<http://modeling.water.ca.gov/delta/models/dsm2/tools/vista/vscript/intro.html>

Finally, a planning run with permanent barriers is needed which can serve as a basis for other planning run variations. The precise permanent barrier operation is still being determined by other staff members for the South Delta Barriers Project.

7.4 References

- Ateljevich, E. and T. Pranger. (2002). "Chapter 13: DSM2 Input Database and Data Management System." *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 23rd Annual Progress Report to the State Water Resources Control Board.* California Department of Water Resources, Office of State Water Project Planning. Sacramento, CA.
- Ateljevich, E. and R. Finch. (2004). "Chapter 2: REALM Update." *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 25th Annual Progress Report to the State Water Resources Control Board.* California Department of Water Resources, Bay-Delta Office. Sacramento, CA.
- Tom, B. (1998). "Chapter 6: Cross-Section Development Program." *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 19th Annual Progress Report to the State Water Resources Control Board.* California Department of Water Resources, Office of State Water Project Planning. Sacramento, CA.