

APPENDIX E

POST PROCESSOR OBJECT-ORIENTED DESIGN

Marion W. Jenkins

March 16, 2000

INTRODUCTION

This appendix provides an overview of the design of a post-processing software tool to examine and analyze the output from CALVIN alternatives (model runs). The design is object-oriented (Coad 1990) and applies generally to time-series and paired data from water resource system models. Its development in subsequent phases of this research project could be of value to water resources agencies and institutions in California.

POST-PROCESSING FUNCTIONALITY

The functions of the post-processor are to display (in plots or tables), distill, combine, analyze, compare, and present output from CALVIN. The two main objectives of post-processing functions are internal checking (calibration and validation) of a model and presentation of policy and management results. In its raw form, CALVIN output consists of time-series of flows (e.g., diversions, deliveries, releases, and pumping), volumes (e.g., storage and evaporation), and economic values (dual-costs or shadow-values) of water at link and node locations throughout the network model. In addition, CALVIN produces a total minimized cost (“penalty”) of each model run, reflecting the integrated value of all the water allocation decisions represented in the model.

In a post-processor, input data are also often required to examine, interpret, analyze, or otherwise translate raw output data. Input data to CALVIN consist of time-series (inflows of hydrology and time-varying constraints), scalar values (fixed constraints, fixed costs, and fractional gains or losses), and paired data (functional relationships between such things as storage, elevation, and area in a reservoir, or between quantity delivered and economic value or cost of shortage for urban and agricultural water users).

Tables E-1 and E-2 give some examples of the kinds of analyses one might want to perform on raw model output such as that produced by CALVIN or that of other water resource system models, to internally check model results or provide policy and management information. All the results in Tables E-1 and E-2 can be represented in graphical or tabular form as either plots or tables. Hence, the two fundamental result “objects” that are produced by the post-processor are plots and tables. These are the basic building blocks of the design.

Table E-1. Analyses and Presentation of Results for Output from a Single Model Run

Raw output from one model run	Examples of analyses, operations, and presentations with model output
time series of a decision variable at one location (a node or link in the network)	<ol style="list-style-type: none"> 1. Display raw values in a plot or table 2. Compute descriptive statistics (mean, median, minimum, maximum, percentiles, cumulative distribution, exceedence, etc.) 3. Aggregate to larger time unit 4. Filter by time period (i.e., by month, hour, day) 5. Compare values to input target, constraint, or reference value(s) in a plot or statistically 6. Compute time series of derived values (elevation, costs, energy, quality, willingness-to-pay, stream losses, etc.) using the appropriate input functional relationship for the location (paired data in CALVIN's case) 7. Display computed, aggregated, filtered, statistical or derived values in a plot or table 8. Save post-processed data in DSS format as derived paired or time series data back in model run output file
time series of a decision variable for a group of locations	<ol style="list-style-type: none"> 1. Display raw values for the group of locations together in one plot or table 2. Display raw values for the group of locations in separate plots for each location as in item 1 above 3. Compute time series of derived values for each location in the group of locations as in 5 above 4. Combine raw or derived values for the group of locations into an aggregated time series of values (e.g., time series of all surface storage in a sub-region, of all urban groundwater pumping, of all State Water Project deliveries, of export/inflow ratio for Bay-Delta, or mass balance on a node or in a local region, etc.); in this case, the group of locations and combination calculation become a "derived data location" object as shown in Figures E-1 and E-2 5. Save post-processed data in DSS format as derived paired or time series data in model run output file
paired data in CALVIN (a functional relationship) for a single location	<ol style="list-style-type: none"> 1. Display a paired data relationship in a plot or table (or set of plots or tables for more than 2 dimensions or more than one time-period) 2. Compute target value(s) from paired "penalty" (economic or priority) data for deliveries 3. Compute break points and slopes from piece-wise linear paired data 4. Report computed values in a table 5. Save post-processed data in DSS format as derived paired or time series derived data back in model run output file
paired data for a group of locations	<ol style="list-style-type: none"> 1. Display paired data relationships for the group of locations in one plot/table 2. Display paired data relationships for the group of locations on separate plots/tables 3. Compute target value(s) and break points as in 2 and 3 above for each location in the group 4. Report computed values in a table 5. Aggregate targets for the group of locations 6. Save post-processed data in DSS format as derived paired or time series derived data back in model run output file

Table E-2. Analyses and Presentation of Results for Output from Several Model Runs (Alternatives)

Raw output from several model runs	Examples of analyses, operations, and presentations with model output
time series of a decision variable for a single location in <u>two</u> model runs	<ol style="list-style-type: none"> 1. Display raw, statistical, aggregated, filtered or derived values at location for each run together in one plot or table 2. Compare raw or derived value to input target, constraint, or reference value(s) for each run together in one plot or statistically 3. Compute change or difference in raw or derived values between two runs 4. Compute descriptive statistics of change/difference between two runs 5. Aggregate change/difference to larger time unit 6. Filter change/difference by time period 7. Display computed, aggregated, filtered or statistical change/difference values in a plot or table 8. Save post-processed data in DSS format as derived paired or time series data back in (appropriate) model run(s) output file
time series of a decision variable for a single location in <u>more than two</u> model runs	<ol style="list-style-type: none"> 1. Display raw, statistical, aggregated, filtered or derived values at location for each run together in one plot or table 2. Compare raw or derived values to input target, constraint, or reference value(s) for each run together in one plot or statistically 3. Save post-processed data in DSS format as derived paired or time series data back in (appropriate) model run(s) output file
time series of a decision variable for a group of locations in <u>two</u> model runs	<p><i>Comparison of model runs at each location:</i></p> <ol style="list-style-type: none"> 1. Display raw, statistical, aggregated, filtered or derived values for both runs using a single plot or table for each location 2. Compare raw or derived value to input target, constraint, or reference value(s) for both runs using a single plot for each location or statistically <p><i>Change/difference in value at each location between two runs:</i></p> <ol style="list-style-type: none"> 3. Display computed, aggregated, filtered or statistical change/difference value between two runs at each location, using one plot or table for each location <p><i>Treat group of locations as a “derived data location” (see Figure E-1):</i></p> <ol style="list-style-type: none"> 4. Display raw, statistical, aggregated, filtered or derived values at the “derived data location” (constructed from the group of locations) for both runs together in one plot or table 5. Compute and display change/difference between the two runs at the “derived data location” (aggregated, filtered, or statistical raw or derived change/difference values) in one plot or table 6. Save post-processed data in DSS format as derived paired or time series data back in (appropriate) model run(s) output file
time series of a decision variable for a group of locations in <u>more than two</u> model runs	<p><i>Comparison of model runs at each location:</i></p> <ol style="list-style-type: none"> 1. Display raw, statistical, aggregated, filtered or derived values at each location for all runs, using a single plot or table for each location 2. Compare raw or derived value to input target, constraint, or reference value(s) for all runs, using a single plot for each location or statistically <p><i>Treat group of locations as an aggregated location (“derived data location” in Figure E-1):</i></p> <ol style="list-style-type: none"> 3. Display raw, statistical, aggregated, filtered or derived values at the “derived data location” (constructed from the group of locations) for all runs together in one plot or table 4. Save post-processed data in DSS format as derived paired or time series derived data back in (appropriate) model run(s) output file

OBJECT-ORIENTED DESIGN OBJECTS AND HIERARCHY

The objectives, functions, and user actions of a post-processing software tool were analyzed to develop the preliminary object elements and hierarchy shown in Figure E-1. There are five main branches descending from the “Post Processor Application” in Figure E-1. The first branch consists of two separate “DLLs” or libraries of code for defining functions for calculating and filtering data. The second branch consists of instructional or definitional objects stored in the “Post Processing Database” which are used to construct a “portfolio”. A portfolio is a set of results produced by the post processor on the raw output from one or more specified model runs according to encoded instructions or requests from the user. There are seven instructional/definitional objects shown in Figure E-1 on this branch. Some of these, as described below, are comprised of unique combinations of lower level instructional objects along with additional definitional information. The third branch is the “Portfolio Generator”, an interface component that generates the portfolio using a selected “portfolio template” (the encoding of the user’s instructions and requests) and a selected “alternative” or “alternatives” (the input or output from a model run or runs). Once generated, portfolios are stored on the hard drive, as presently envisioned, in a sub-directory as a set of files (e.g., a set of EXCEL workbooks). The fourth branch displays the objects comprising a portfolio and their relationship to each other. The fifth branch consists of the interface editors where instructional objects and requests are created, deleted, saved, selected, and edited by the user to produce a desired “portfolio template” of instructions for processing a set of results. Further elaboration of these objects follows in the next sections. An initial sketch of some of the details on the proposed properties and methods of objects are given in Figure E-2.

INSTRUCTIONAL/DEFINITIONAL OBJECTS STORED IN A DATABASE

The Post Processor Database would most likely have a separate table for storing each of the following types of instructional or definitional objects:

1. Data Locations Groups: each “Group” record would contain a list of data locations
2. Data Locations Derived Definitions: each “definition” record would contain a reference to a Data Locations Group stored in the first table and a calculation to apply to the locations in this group
3. Portfolio Templates: each “template” record would contain a list of presentation templates
4. Presentation Templates: each “template” record would contain a list of “presentation components” and a time window for the analysis
5. Presentation Components: each “component” record is a definition of a set of tables or plots comprised of information on actions 9a through 9h of the Presentation Template Editor User Actions presented below
6. Plot Templates: each template contains a generic definition type and excel chart template
7. Table Templates: each template contains a generic definition type and formatting attributes

Figure E-1. CALVIN Generalized Post-Processing Design Object Hierarchy

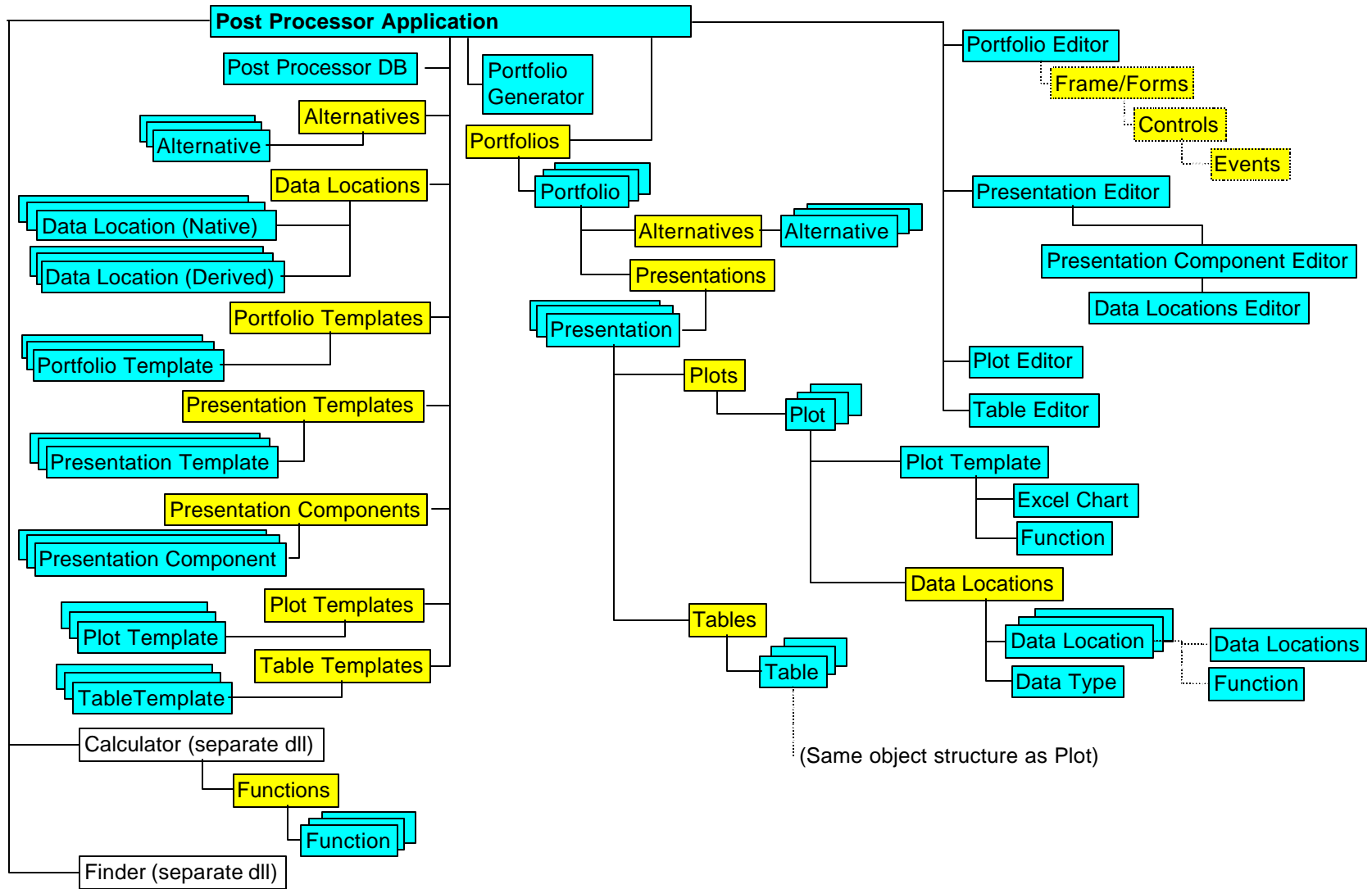


Figure E-2. Proposed Objects, Properties, and Methods



Figure E-2. Proposed Objects, Properties, and Methods (Continued)



Alternatives would not be stored in this database because they would be created and managed by the “Model Data Manager” Application in its own database of modeling runs (see Appendix D). All objects in the Post Processor database are created, deleted, saved, and edited in the editors shown in Figure E-1 according to the user actions described in the next section.

A portfolio template consists of a set of “presentation templates,” which in turn consists of “presentation components”. Defined in the presentation template, and therefore general across all presentation components in that template, is the window of time covered by the presentation’s results. A presentation component holds the definition for each plot or table in its analysis set of plots or tables. A presentation component definition emerges by combining a plot or table template from the list of available plot template objects in the database, with a data locations group, a specified data type, and a time step, and choosing between “single plot/table of multiple locations” or “multiple plots of single location”. The data locations group can either be selected from the list of existing groups or created as a new group and saved in the database of data locations groups. For example, you may want to plot the time-series of monthly storage in all groundwater basins on a single plot for comparative purposes. This particular analysis becomes a “component” of a larger presentation and its definition consists of a user-formatted or pre-defined plot template, the list of all the groundwater basin elements in the network, the decision variable of storage, and the time step of month. You may also want to see the storage time series for these groundwater basins each on separate plots. This would be a different presentation component with all the same information except a different choice of multiple versus single plot option.

POST PROCESSOR USER INTERFACE EDITORS AND ACTIONS

The following lists of actions indicate how the editors on the fifth branch of the design in Figure E-1 would be used to build the instructional objects (the templates described above and stored in the Post Processor Database) and how the Portfolio Generator would use these instructional objects to generate a portfolio. The Portfolio Generator and all editors except the Data Locations Editor would be on the main menu as each can be operated independently of the others. The Data Locations Editor would be a sub-menu of the Presentation Template Editor.

Portfolio Generator User Actions

- 1 Select (specify) alternative(s) <from Model Data Manager>
- 2 Select (specify) portfolio template
- 3 Generate and save a portfolio (post process data according to portfolio template for selected alternative(s))
- 4 Select a portfolio (an existing portfolio processed collection of EXCEL files)
- 5 Review selected portfolio (= read-only look at portfolio collection of EXCEL files)
- 6 Edit portfolio template
- 7 Specify name for portfolio (= collection of EXCEL files, one for each presentation)
- 8 Specify batch process for generating multiple portfolios

Portfolio Template Editor User Actions

- 1 Select presentation template(s) from list
- 2 Add presentation template(s)
- 3 Delete presentation template(s)
- 4 Create portfolio template(s)
- 5 Copy portfolio template(s) (=save as)
- 6 Select portfolio template
- 7 Delete portfolio template
- 8 Save portfolio template

Presentation Template Editor User Actions

- 1 Select and open presentation template
- 2 Choose time window (attribute of presentation template)
- 3 Copy presentation template (=save as)
- 4 Save presentation template
- 5 Delete presentation template
- 6 Create a presentation template
- 7 Display list of items ("presentation components") in presentation template
- 8 Remove item(s) ("presentation component(s)") from list
- 9 Add item(s) (a "presentation component" set of plots or tables) to presentation template list as follows:
 - a. choose between (two basic types of "components"):
 - multiple plots of single location (analogous for tables)
 - single plot of multiple location
 - b. select and specify data locations (from list, add, remove)
 - c. select and specify data type for each location ((e.g., storage, evap., flow, penalty)
 - d. select functional operator (min, max, cumulative, exceedence, etc)
 - e. select time step
 - f. specify alternative interaction:
 - none (comparison for more than 1)
 - functional operator (subtract across alts, % change across alternatives for 2 alternatives)
 - g. select and specify plot template to apply (e.g., TS, percentiles, paired data, exceedence, etc.) from a list of pre-defined plot templates
 - h. filter ? (e.g., exceedence for a specific month)

Plot Template Editor User Actions

(Produces/saves a general plot format not specific to location choice, to alternative choice, to time window choice, or to time step choice)

Analogous for Table Template Editor Actions

- 1 Select a plot template from list of templates
- 2 Open plot template definition
- 3 Display template sample
- 4 Close template
- 5 Save modified template to list
- 6 Delete template (from list)
- 7 Copy (=save as) a new plot template to list
- 8 Edit template (embedded EXCEL Chart):
 - a. use EXCEL chart tools to allow formatting of desired chart and then store the chart properties to use in building chart when generating portfolio
 - b. select display (Bar, Line, etc.)
 - c. set number of data series per alternative?

Data Locations Editor User Actions

- 1 Select and open a group of data locations from a list of data location groups or collections (groups of locations based on some user criteria, selection, etc., could be filtered using attributes stored with each network element, based on geographic location, type of water use, project, etc.)
- 2 Edit by adding/removing a location from the open group list
- 3 Specify a name for the group
- 4 Copy ("save as") a new group to the list
- 5 Close an open group
- 6 Delete a group from the list
- 7 Create a new group of locations:
 - a. open list of all native and derived data locations
 - b. select locations by adding them to a "desired" list box
 - c. specify the name, description of the new group
 - d. save the new group to the list of groups
- 8 Create a new derived location:
 - a. open list of all native and derived data locations
 - b. select locations by adding them to a "desired" list box
 - c. choose a function for combining locations in the list
 - d. specify the name, description of the new derived location
 - e. save the new derived location to the list of derived locations

LINKAGES TO MODEL INPUT AND OUTPUT

As mentioned above, input data are sometimes needed in post-processing raw output. For example, to derive a time series of shortages for a location, the output delivery decisions (flows) must be compared to the input target delivery. In the case of CALVIN, this is derived from the monthly input paired penalty data for that location. As such, data objects and network element

objects (links and nodes) from the model input interface application will have to be shared with the post-processor and appropriate linkages made between these two applications.

Likewise, “alternatives” or model runs will be managed outside the Post Processor Application by a “Model Data Manager” (see Appendix D). Linkages and sharing of objects between the Model Data Manager and Post Processor will also be required to access model input files, raw output files, and store and retrieve derived output data and portfolio files.

REFERENCES

Coad, Peter (1990) *Object-oriented Analysis*. Edward Yourdon Press, Englewood Cliffs, N.J.