Open Water Information Architecture

Standard Operating Procedures

April 3, 2019

DRAFT

Prepared by…

John J. Helly, UCSD, SDSC
with the Water Supply and Balances Work Team and
AB 1755 Partner Agency Team

Prepared for…
California Department of Water Resources

Contents

List of Figures

List of Tables

Listings

Chapter 1

The Open Water Information Architecture (OWIA)

An *information architecture* is a means of *mobilizing information to satisfy a set of objectives*. This means that an information architecture has a purpose and a focus. It may be as broad as to provide an archive for the preservation of all published materials, such as the university and public library systems, or it may be as narrow as providing decision-support for the water resource management of California. The OWIA addresses the water resource management problem with proven methods, conventional to information sciences community, using modern cyberinfrastructure and computing methods.

This paper discusses the OWIA both from an objectives perspective, from the top-down, as well as a bottom-up requirements baseline as a framework to provide high quality, digital information products for water resource management in perpetuity. The reader is encouraged to think of what ensues in this paper as being about data, not computers. This is important because whatever underlying cyberinfrastructure is in use at any particular time to implement the OWIA, the cyberinfrastructure changes rapidly with respect to the specification of the information products which change relatively slowly. The OWIA must be robust to technology changes if it is to be long-lived and sustainable.

By translating the objectives, currently realized as Stakeholder Use-cases, into functional and technical requirements, in the form of a System Requirements Document (SRD), with an accompanying set of standard operating procedures (SOPs), we provide a means to control *cost, schedule, technical and operational risk*. At the same time, providing project managers (a) a basis for measuring the performance of system(s) built to satisfy the objectives and to (b) compare the relative merit of alternative designs, a (c) rational path for technology updates over time, and a (d) framework for controlling the long-term operations and maintenance costs.

The *Open Water Information Architecture (OWIA)* does these things for the community of stakeholders with an interest in *water resources and open-data*. The OWIA establishes a federation of peer data systems sharing standards and conventions to govern cyberinfrastructure, data and operations. It is a system of systems allowing organizations and individuals to leverage existing investments in staff and data systems while providing a means for interoperability and cooperation. Some of the benefits of the OWIA approach are (i) greater access to standardized data in (ii) an *open-system* providing for (iii) growth in data holdings, (iv) flexible integration of changing technology and innovation, (v) responsiveness to emergent stakeholder objectives with (vi) an elimination of stranded assets.

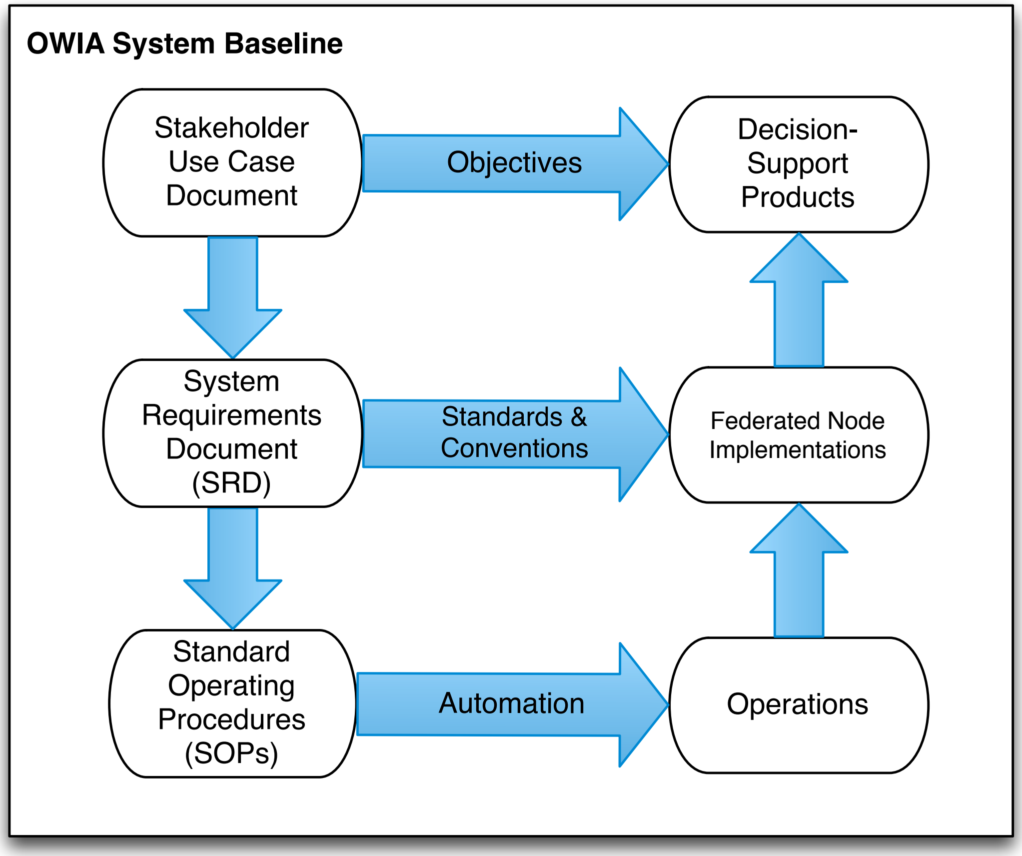

Figure 1.1: Technical baseline documents and their relationship to OWIA data node implementations.

1.1 Concept of Operations

This document contains the functional and technical requirements for the *Open Water Information Architecture (OWIA)* and is called the *OWIA System Requirements Document (SRD)*. It has within it an Appendix:**??** Standards and Conventions that contains narrative explanations that are referred to within individual requirements where appropriate. This is done because the requirements are meant to be terse, declarative, testable statements that are not overloaded with narrative exposition. There are two companion documents to the SRD: (1) the subordinate document *OWIA Standard Operating Procedures (SOPs)* and the (2) parent document *California Council for Science and Technology (CCST) Stakeholder Use Case* document.

The SOPs are compliant with the requirements specified here yet written at a more detailed level of abstraction with examples of programming code or sometimes pseudo-code to exemplify the implementation details important to developers as well as precisely documenting the processing steps (i.e., procedures) used to operate on data. It is meant to be analogous to an *OWIA Programmer's Guide* and, as the OWIA implementation proceeds, there will be open-source code repositories with *minimal working examples (MWE)* for use in improvements and innovations to current procedures and applications implementing those procedures.

Each of these documents is intended for a technical audience although it is hoped that they are comprehensible to a motivated non-technical reader. There is a glossary in the back of the SRD to aid in navigating the technical language and as an effort to disambiguate some of the terms for which there may be competing and inconsistent definitions. In addition to these two, there is a third document that contains the stakeholder

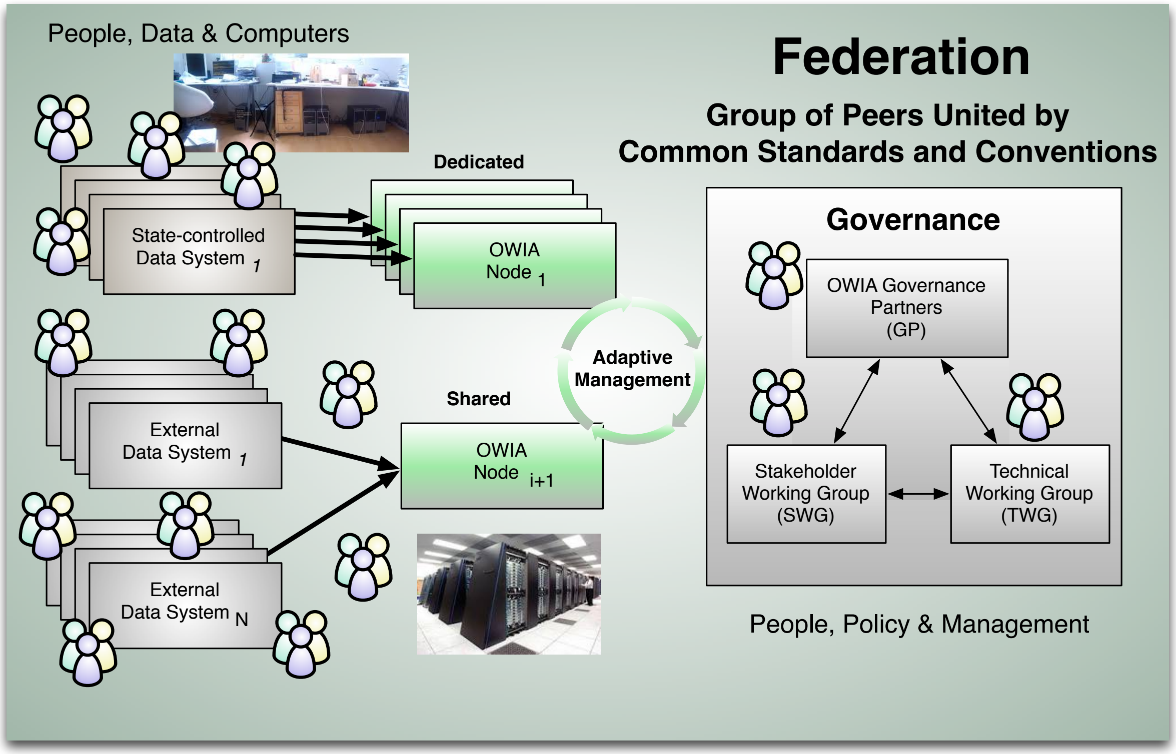

Figure 1.2: Illustration of the OWIA federation *concept-of-operations* with a triumvirate governance structure of general partners (GP) supported by interacting with a stakeholder working group (SWG) and a technical working group (TWG). The federation is comprised of dedicated OWIA system implementations to enable individual data providers to independently integrate the OWIA into their existing operations. Shared OWIA system implementations provide the flexibility for the harvesting non-compliant data sources without insisting that the data producers be OWIA-compliant.

use cases used to develop the stakeholder objectives from each use case. These objectives are being used to define and constrain the requirements contained in the SRD and the procedures for satisfying them defined in the SOPs.

The SRD and SOPs are designed to provide a foundation for a community-based *OWIA* development of a federated set of cyberinfrastructure resources (i.e., computers, networks, data, metadata, and standards and conventions) that are interoperable and highly-automated to minimize labor as well as idiosyncratic anomalies. We therefore refer to them as the *baseline documents* (Figure 1.1). The objective of these baseline documents is to establish a framework for sustainable water resource management and to formalize that framework to a degree exemplified by other systems of standard methods such as those found in [6].

The federated nature of the OWIA extends to its (1) human governance structure as well as its (2) cyber-infrastructure (cf. Section **??** and Figure 1.2). Therefore we speak of the OWIA federation as including both these aspects and will differentiate the two parts contextually when using the term. The *open* aspect means open-access, open-source and open-architecture: encouraging innovation and automation while precluding the siloing and stove-piping that occurs when proprietary software and systems pose restrictive technology dependencies and requirements. The planning horizon is open-ended although intended to provide for a near-term operational system with an initial operating capability (IOC) within 1-2 years evolving to a final operating capability (FOC) over five (5) years that is operationally sustainable while responsive to technol-

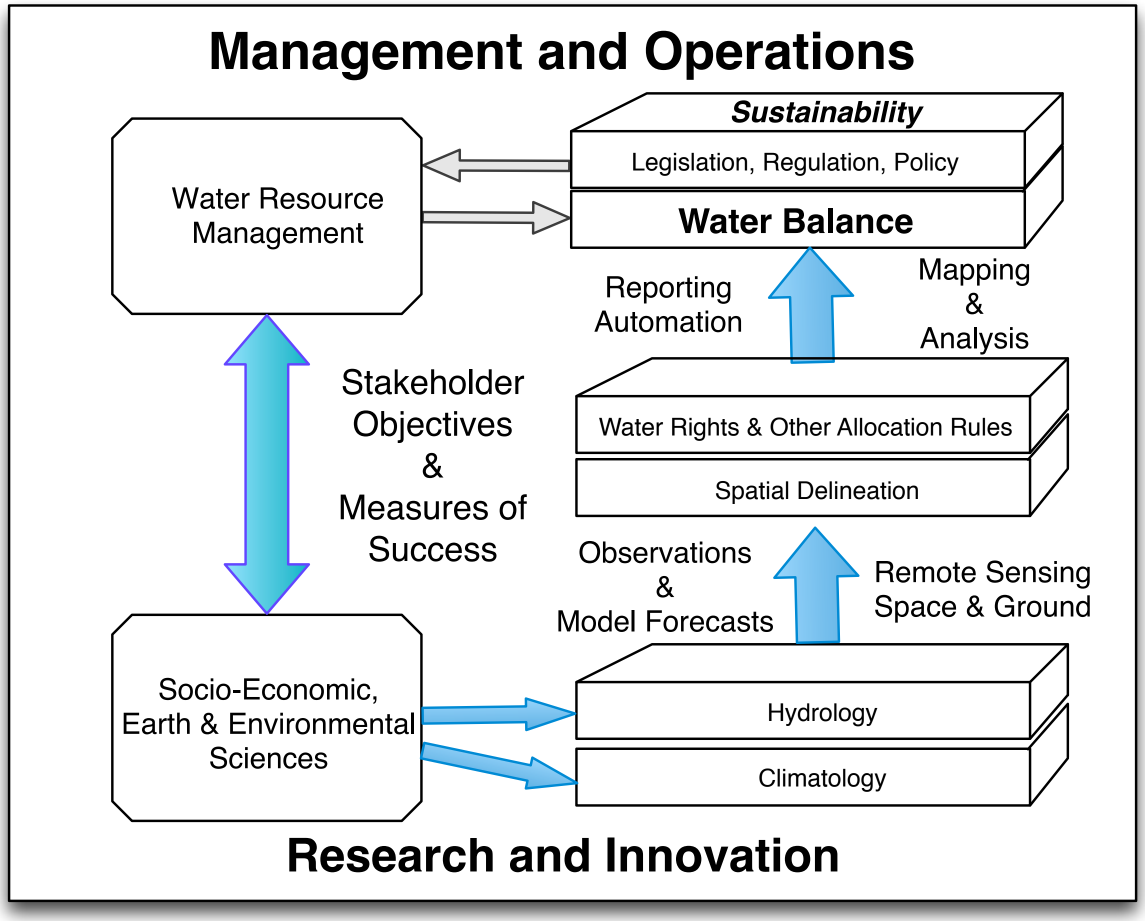

Figure 1.3: Open Water Data Information Architecture (OWIA) framework.

ogy innovation and risk minimization (i.e, cost, schedule, technical and operational) over its lifetime.

The approach is to follow standard system engineering practices [21] that: (1) define stakeholder objectives and, from these, (2) enumerate functional requirements in terms of functional components and major interfaces both of which are implementation-independent, and (3) enumerate technical requirements which specify fundamental technical features such as network transfer rates, storage capacities, reliability, maintainability and availability (RMA), interface dependencies and contingencies and similar quantitative or qualitative requirements at a level of specificity (or abstraction) that is more detailed than the functional requirements on which they are based. It is also designed to present an initial evaluation of some of the obvious design trade-studies to explicate and focus on the key risk areas related to technical, schedule, cost and operational risks.

This is an interative and recursive, hierarchical design approach (Figure **??**) which prioritizes *Stakeholder Objectives, Functional Requirements, and Technical Requirements* respectively and cross-correlates them to each other via a *traceability matrices* (Section **??**) to ensure that there are no *widows or orphans* in the sense that there are no unsupported Objectives or Functional Requirements (i.e., widows) as well as no lower-level design features that are not specified in the Functional Requirements (i.e., orphans). As a development methodology, the system engineering method used here is sometimes contrasted with the agile development method. Every methodology has pros and cons and the reason we use this approach for the OWIA is because we already know a great deal about what is needed to improve access-to and reuse-of the collective set of water resource data and the OWIA focus is on the data content. This is not primarily a process of discovery and prototyping of software applications. For a broader discussion of the pros and cons of alternative software development approaches, the reader is encouraged to consider the discussions provided in [18] and [21].

Finally, some historical perspective is helpful. This document is meant to integrate the thinking on water resource information broadly and digital data about water resources specifically. The OWIA concept developed independently of the AB1755 legislation [1][19] that is currently, as of this writing, driving many efforts across the State of California to comply with its mandates and schedule. Fortuitously, the development of the OWIA and the activation of AB1755-related efforts overlap strongly such that AB1755 requirements are a subset of the broader OWIA requirements. The implementation of the OWIA will satisfy the requirements of AB1755 and support the Sustainable Groundwater Management Act (SGMA) in such a way that we can treat AB1755 as an OWIA use-case as described in Appendix **??**. The OWIA concept is a reflection and integration of a wide range of on-going efforts especially those in the UC WATER Security and Sustainability Research Initiative and CITRIS [7], California Council on Science and Technology (CCST), the Center for Western Weather and Water Extremes (CW3E)[25], the San Diego Supercomputer Center (SDSC) [3, 14, 10, 16, 13, 26, 23, 12, 20, 4, 8, 24, 15, 5, 2, 11, 17] and the UC Santa Barbara Bren School. We expect to grow this community to include private California universities, national laboratories and private sector partners as we go.

1.2 Overview of Standard Operating Procedure (SOP) Processing Workflow

This document is the Open Water Information Architecture (OWIA) Standard Operating Procedures (SOPs) document. It describes the procedures developed for the quality control and publication of data related to water resources but it is not limited to that application alone and reflects a more general approach that has been developed and tested over forty year[9, 10, 15, 14, 13, 12]. This report is based both on the mandates of AB1755 and the Open Water Information Architecture (OWIA) System Requirements Document (SRD)[22].

This document is intended as something of a Progammer's Guide to provide examples of the processing steps and coding examples for use in producing OWIA Level 1 data from non-OWIA Level 0 data. The definitiions of these levels can be found in [22].

Figure 1.4 illustrates the workflow associated with the OWIA quality control and data publication procedures. Table 1.1 summarizes the standard operating procedures in terms of the major purpose of each step and the typical types of output resulting from each step.

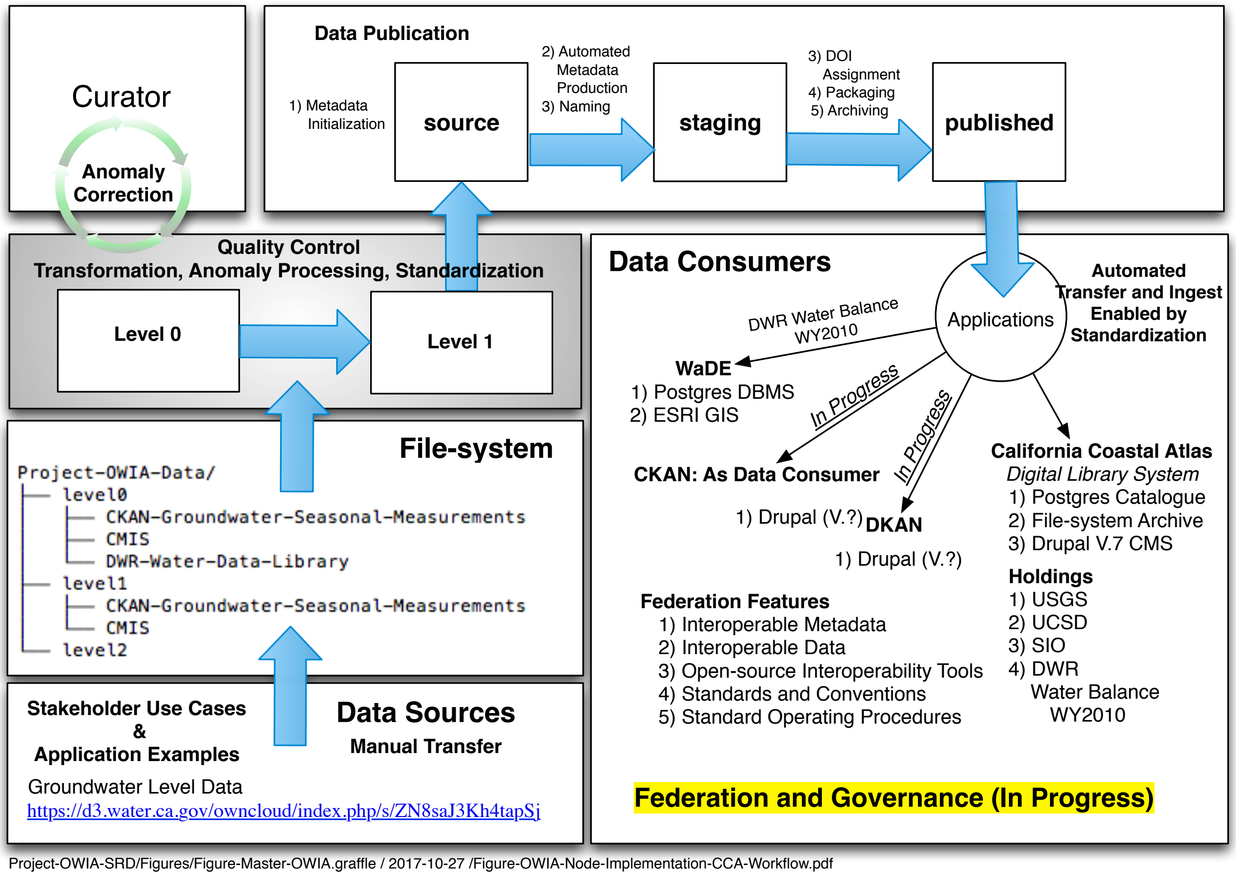

Figure 1.4: Operational workflow for quality control and data publication. Groundwater data is the application example in this figure.

222 ## 1.3 Related Procedural Examples from Software Development

223 FFMPEG

Table 1.1: Overview of standard operating procedures (SOPs) and categorization into quality control (QC), data publication (DP) and analysis (AN) procedures.

Procedure Type	Name	Purpose	Results
Quality Control	QC-1000	Transform to interoperable format and filetype. Verify accuracy of Level 0 to Level 1 Interoperability Transformation for Water Balance Data.	Run QC-1000.R and then manually inspect the results to cross-check the initial Level 1 output data against its Level 0 source for accuracy.
	QC-2000	Perform standardization with range-checking, outlier and anomaly detection. Verify compliance with controlled vocabulary, units of measure, geospatial projection.	QC-2000.R, manual cross-checking.
	QC-3000	Conduct integration verification if producing Level 2 datasets.	QC-3000.R, manual cross-checking.
Data Publication	DP-1000	(1) Prepare metadata and data package for publication. (2) Obtain digital object identifier and finalize processing. (3) Populate archive and catalogue with metadata and data respectively.	Various methods per Technical Working Group.
Data Distribution	API-1000	(1) Application programming interfaces to downstream use-cases.	Various methods per Technical Working Group.
Analysis	AN-1000	Produce the figures (1,2,...,n) and tables (1,2,...,n) necessary.	AN-1000.R

Chapter 2

Standard Operating Procedures (SOPs)

2.1 Water Balance Automation

Water balance is approached by (1) collecting the data entry spreadsheets from each regional office, (2) converting them to *.csv files according to a set of rules, (3) processing them into a controlled vocabulary and parameterization, (4) computing a set of equations based on the controlled vocabulary and parameterization, and then (5) summarizing the results at the DAUCO, HR, PA and ST levels of aggregation after apply adjustments at each spatial scale to account for water re-use at a given scale.

2.1.1 Contributors

These SOPs were developed jointly by Brad Arnold, Glenn Bergquist, Dona Calder, Abby Carevic, Tito Cervantes, Dong Chen, James Common, Matt Correa, Gary Darling, Siran Erysian, Steve Ewert, Robert Fastenau, Todd Flackus, Francisco Guzman, Jason Harbaugh, Scott Hayes, John Helly, Todd Hillaire, Salma Kibrya, Jennifer Kofoid, Kelly Lawler, Michael McGinnis, Salomon Miranda, Lew Moeller, Chris Montoya, Mohammed Mostafavi, Morteza Orang, Toni Pezzetti, Lida Pirjaberi, Mark Rivera, Jessica Salinas Brown, Michael Serna, Gholam Shakouri, Paul Shipman, Jeff Smith, Evelyn Tipton, Lauren Wacker, Paul Wells, Muffet Wilkerson, and Courtney Wilson.

2.1.2 Governing Equations

The general form of the governing equations are linear sums (Eq.2.1). However, there are two sets of equations: two for water use and two for water supply, that are applied sequentially as shown in Figure 2.4. The water use equations (Listings 4.1, 4.2) computes quantities at the DAUCO-level for parameters that are computed from the Level 1 data for each of the sectors: (1) agriculture, (2) urban, (3) managed wetlands, (4) required in-stream flows, (5) wild and scenic rivers. The second set of equations (Listings 4.3, 4.4) computes adjustments in the return flows at the three other spatial scales: state (ST), hydrologic region (HR), and planning areas (PA).

$$y\_ij = \sum_{k=1}^{m} x\_ijk \tag{2.1}$$

where:

y_{ij} = derived parameter (cf. Listings 4.1, 4.2, 4.3, 4.4)
i = sector
j = index into the set of DAUCOs
k = input parameter set index for i^{th} sector
m_i = cardinality of the i^{th} sector input parameter set

2.1.3 Computer Configuration

The software tools used in executing this procedure can vary depending on which operating system you are using and what your preferred manner of working is. Here is a reference set that has been used successfully.

(1) rsync: transfer files between computer systems.
(2) ssh: login to remote computers.
(3) cygwin: provide a Linux-like environment on Windows operating systems.
(4) Qgis: provide GIS capabilities.
(5) LaTex: provide scriptable document preparation.
(6) R: statistical computing software to manipulate data, generate figures, tables, and statistical analyses.

2.1.4 Preparing the Directory Layout

The directory structure should be organized as shown in Figure 2.1. The important points to notice are the relative nature of the directories with respect to the PROJECT_HOME directory, the notion of DATA_INPUT_*, DOCUMENT_*, and specific output directories such as WADE_* for particular output products.

```
options("digits"=4)
options(scipen=0)
#
# ================================================================
# Set directories
# ================================================================
setwd("/Users/hellyj/Archive-local/Project-OWIA-WaterBalance-src/R")
#
PROJECT_HOME             = '/Users/hellyj/Archive-local'
SOURCE_HOME              = paste(PROJECT_HOME,'/Project-OWIA-WaterBalance-src/R', sep='')
SOURCE_FILE              = paste(SOURCE_HOME,'/DWR-QAQC-Verification-300-Modular.R', sep='')
DATA_INPUT_HOME          = paste(PROJECT_HOME,'/Project-OWIA-WaterBalance-Data-Local', sep='')
DATA_OUTPUT_HOME         = DATA_INPUT_HOME
DOCUMENT_HOME            = paste(PROJECT_HOME,'/Project-OWIA-WaterBalance-Documentation/Report-2009',sep='')
#
DATA_INPUT_SUPPLY        = paste(DATA_INPUT_HOME,'/level2/2010/DWR-2010-Master-Supply.csv',sep='')
DATA_INPUT_DEMAND        = paste(DATA_INPUT_HOME,'/level2/2010/DWR-2010-Master-Demand.csv',sep='')
DATA_INPUT_ENVIROMENTAL  = paste(DATA_INPUT_HOME,'/level1/2010/State-Level-Environmental-Data-Merged.csv', sep='')
#
DOCUMENT_FIGURES         = paste(DOCUMENT_HOME,'/Figures/',sep='')
DOCUMENT_TABLES          = paste(DOCUMENT_HOME,'/Tables',sep='')
#
WADE_OUTPUT_SUPPLY       = paste(DATA_OUTPUT_HOME,'/2010/WADE-DWR-2010-Master-Supply-Tabulated.csv',sep='')
WADE_OUTPUT_USE          = paste(DATA_OUTPUT_HOME,'/2010/WADE-DWR-2010-Master-Use-Tabulated.csv',sep='')
#
```

Figure 2.1: Example directory organization. Note that the organization has been factored so that the paths are relative to the PROJECT_HOME which would typically change from computer to computer.

```
MFS-Desktop:~/Archive-local/Project-OWIA-WaterBalance-Data-Local>tree
.
├── level0
│   └── 2009
│       ├── 2009_Central_Coast_NCRO_04222014.xlsm
│       ├── 2009_Central_Coast_SCRO_5-20-14.xlsm
│       ├── 2009_Central_Coast_SRO.xlsm
│       ├── 2009_ColoradoRiver_SRO_1of2.xlsm
│       ├── 2009_ColoradoRiver_SRO_2of2_rev_4-22-13.xlsm
│       ├── 2009_North_Coast_NCRO_04092014.xlsm
│       ├── 2009_North_Coast_NRO.xlsm
│       ├── 2009_North_Coast_PA101_NRO.xlsx
│       ├── 2009_North_Coast_PA102_NRO.xlsx
│       ├── 2009_North_Coast_PA103_NRO.xlsx
│       ├── 2009_North_Coast_PA104_NRO.xlsx
│       ├── 2009_North_Lahontan_NCRO.xlsm
│       ├── 2009_North_Lahontan_NRO.xlsm
│       ├── 2009_Sacramento_River_NCRO_1of2.xlsm
│       ├── 2009_Sacramento_River_NCRO_2of2.xlsm
│       ├── 2009_Sacramento_River_NRO_1of2.xlsm
│       ├── 2009_Sacramento_River_NRO_2of2.xlsm
│       ├── 2009_San_Francisco_Bay_NCRO_09102014.xlsm
│       ├── 2009_San_Joaquin_NCRO_04222014.xlsm
│       ├── 2009_San_Joaquin_NCRO_to_SCRO_04222014.xlsm
│       ├── 2009_San_Joaquin_SCRO_1of2_5-20-14.xlsm
│       ├── 2009_San_Joaquin_SCRO_2of2.xlsm
│       ├── 2009_SouthLahontan_SRO_1of3.xlsm
│       ├── 2009_SouthLahontan_SRO_2of3.xlsm
│       ├── 2009_SouthLahontan_SRO_3of3.xlsm
│       ├── 2009_South_Coast_SRO_rev_4-22-13.xlsm
│       ├── 2009_Tulare_Lake_SCRO_1of3_posted_8-8-12.xlsm
│       ├── 2009_Tulare_Lake_SCRO_2of3_posted.xlsm
│       ├── 2009_Tulare_Lake_SCRO_3of3_posted.xlsm
│       └── ~$2009_North_Coast_NRO.xlsm
├── level1
│   └── 2009
│       └── 2009_North_Coast_NRO.csv
└── level2
    └── 2009

6 directories, 31 files
MFS-Desktop:~/Archive-local/Project-OWIA-WaterBalance-Data-Local>
```

Figure 2.2: Tree diagram of data directory corresponding to the directory layout in Figure 2.1

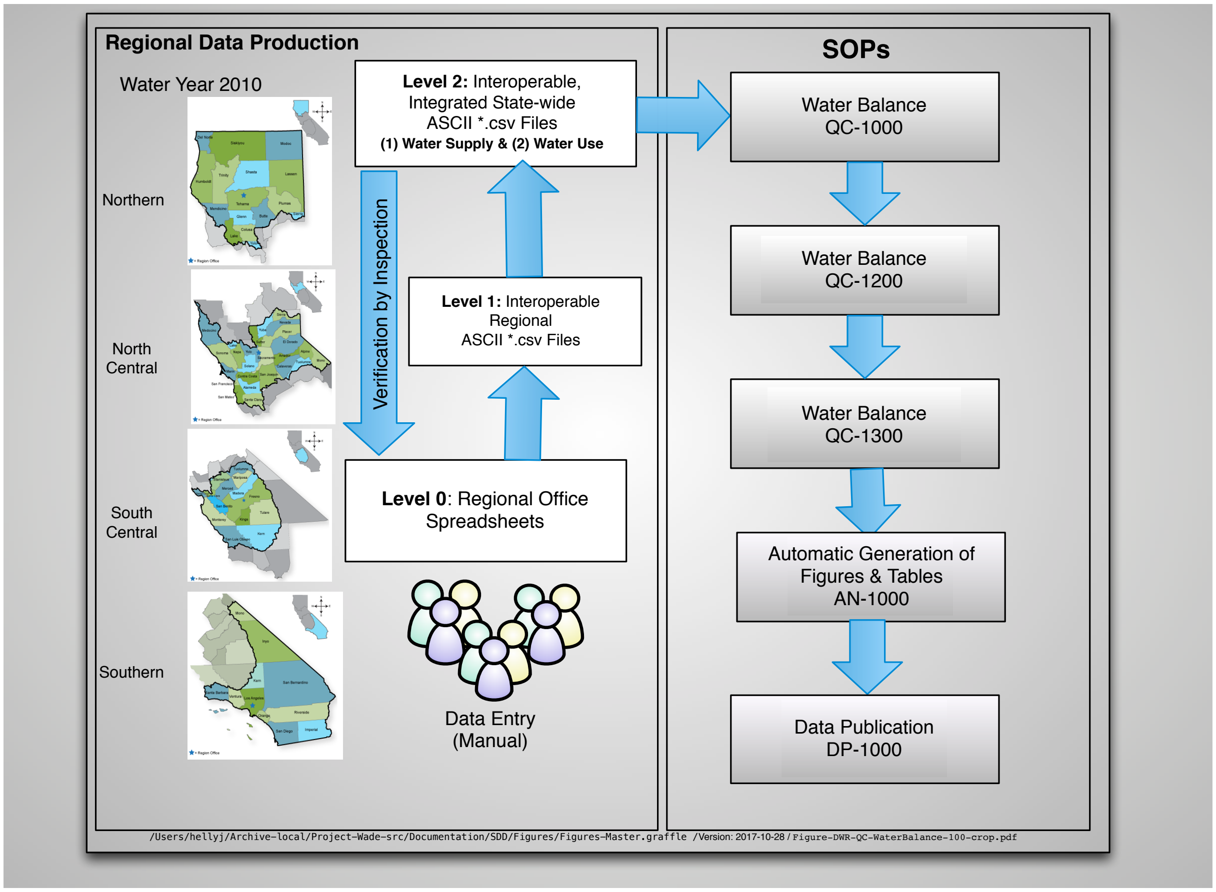

Figure 2.3: Quality control workflow for water balance automation. For this problem, regional data were integrated into a Level 2 (multiple Level 1 datasets) before quality control was applied to enable anomaly detection across the state-level dataset.

2.1.5 GR-1000: Procedure for Extracting Hydrologic Region, Planning Area and DAUCO Data with Georeferencing from DWR Shape Files

1. Load shape file into Qgis as vector layer and export as csv file.
2. Under Vector menu, use Geometry Tools to computer a new layer with polygon centroids in it.
3. Open the Attribute Table and then the Field Calculator. Select Geometry and add fields for $y and $x. Set field attributes to decimal and 15.3 precision since we are working in EPSG:3310, state plane coordinates and these are eastings and northings in meters.
4. Export the attribute table as *.csv and edit it in a spreadsheet.
   (a) Remove rows for islands.
   (b) Remove columns with comma-separated content (i.e., FIPS, Counties).
   (c) Save it with *-Cleaned.csv in the filename using Windows comma-separated format.
5. Run it through the reproject.bash script to generate Longitude and Latitude values and recombine with the rest of the Attribute table using the spreadsheet.

2.1.6 QC-1000: Procedure for Transforming Spreadsheets to *Comma-separated Value* (*.csv) Files

1. Open level 0 spreadsheet in spreadsheet program (e.g, Excel, Open-Office, Libre-Office)

2. Save as level 1 spreadsheet in *.csv format.
3. Close spreadsheet and re-open using *.csv file (removes formatting).
4. *Perform frequent saves.*
5. Trim off right-hand columns.
6. Trim off top rows down to the DAU XXXXX headings.
7. Trim off bottom rows after Water Portfolio rows.
8. Convert the column with alpha-numeric labels to be all text.
9. Delete any remaining blank columns.
10. Add **CategoryA** column and populate with the sector labels (e.g., Agriculture, Urban).
11. Add **CategoryB** and **CategoryC** labels to column headings.
12. Remove any redundant or blank rows.
13. Edit the DAU XXXXX column headings to be DAUXXXXX (no space).
14. Edit *Wild & Scenic* to be *Wild and Scenic*.
15. Format all numeric cells to be floating point numbers without commas.
16. Verify the file contents using an ASCII editor.
17. Rows that have *Totals* will be systematically removed using R script.

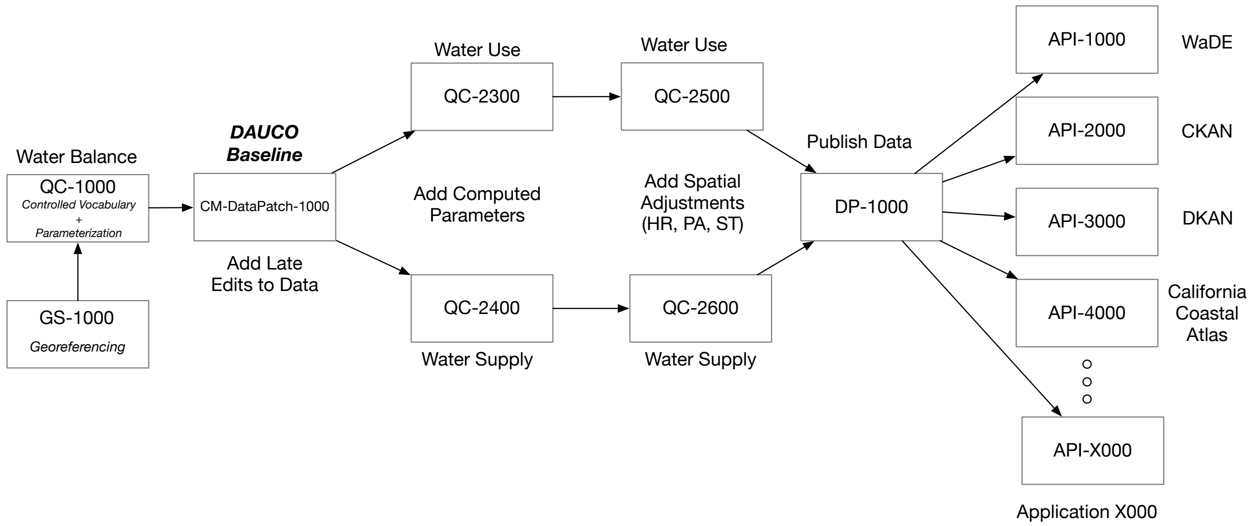

Figure 2.4: Overview of geospatial metadata extraction (GS), quality control (QC), configuration management and change control (CM), data publication (DP) and application programming interfaces (API) for water balance data.

Table 2.1: SOPs applied to water balance data with verification products described.

SOP	Processing Sequence	Verification Products
QC-1000	(1) Obtain Level 0 (L0) data and establish project directory structure using a file-naming convention that uniquely identifies the source of the data, when it is obtained and what it is. (2) Convert input Level 0 (L0) data into an intermediate Level 1 (L1) interoperable format. This may be done programmatically or by manual editing. (3) Merge L0 data with geospatial metadata from State GIS database. Need authoritative, standard source reference here. (4) Apply controlled vocabulary for aggregation categories and parameter names and units. (5) Verify transformation from L0 to L1 did not change values of parameters by visual inspection. (6) Perform range check, anomaly detection and characterization table. (7) Standardize nomenclature using controlled vocabulary. (8) Write out L1 data using filename following this format: `CA-DWR-WaterBalance-QC-1000-2011-2015-Standardized.csv`.	(i) Cross-check tabulation, scatterplots, histograms. (ii) Verify transformation from L0 to L1 did not change values of parameters. (iii) Perform range check and anomaly detection action item list. (iv) Tabulation of category vocabulary and parameter names.
QC-2300	(1) Obtain Level 0 (L0) data and establish project directory structure using a file-naming convention that uniquely identifies the source of the data, when it is obtained and what it is.	(i) Cross-check tabulation, scatterplots, histograms. (ii) Establish new reports if desired based on the new standardization to controlled vocabulary.
QC-2500	(1) Obtain Level 0 (L0) data and establish project directory structure using a file-naming convention that uniquely identifies the source of the data, when it is obtained and what it is.	(i) Cross-check tabulation, scatterplots, histograms. (ii) Establish new reports if desired based on the new standardization to controlled vocabulary.

QC-2400	(1) Obtain Level 0 (L0) data and establish project directory structure using a file-naming convention that uniquely identifies the source of the data, when it is obtained and what it is.	(i) Cross-check tabulation, scatterplots, histograms. (ii) Establish new reports if desired based on the new standardization to controlled vocabulary.
QC-2600	(1) Obtain Level 0 (L0) data and establish project directory structure using a file-naming convention that uniquely identifies the source of the data, when it is obtained and what it is.	(i) Cross-check tabulation, scatterplots, histograms. (ii) Establish new reports if desired based on the new standardization to controlled vocabulary.
DP-1000	(1) Copy data to the data publication platform source directory. (2) Initialize the metadata generation configuration file. (3) Run the Digital Library System (4) Run the EZID digital object identifier generator. (5) Edit metadata and filenaming to include the newly produced DOI. (6) Archive the re-packaged data in the published directory.	(i) Verify DOI against EZID catalogue. (ii) Verify access to published data and metadata in the archive.
AN-1000	(1) Run scripts for table and figure generation.	(i) See *WaDE Node for California System Description Document* for verification product examples.

2.2 Water Budget

2.3 Water Quality

2.4 Environment

Chapter 3

Numerical Methods

3.1 Analysis of Uncertainty

Guidance for the analysis of uncertainty is gleaned from [**?**, **?**] and [6].

3.2 Accuracy and Precision

3.2.1 Significant Digits

There are three rules on determining how many significant figures there are in a number [**?**]:

1. Non-zero digits are always significant.
2. Any zeros between two significant digits are significant.
3. A final zero or trailing zeros, *in the decimal portion only*, are significant.

All numbers are based upon measurements except for a very few that are defined and all measurements are uncertain, we must only use those numbers that are meaningful.

3.3 Propagation of Uncertainty in Calculations

$$In = A \pm \alpha \tag{3.1}$$

$$Out = B \pm \beta \tag{3.2}$$

$$Net = (A - B) \pm \sqrt{\alpha^2 \pm (\alpha\beta) + \beta^2} \tag{3.3}$$

3.4 Uncertainty in Estimates from Numerical Models

3.4.1 Verification and Validation

3.4.1.1 Irreproducible Results Across Computing Platforms

3.4.2 Statistics from Ensembles

Chapter 4

Testbed Description

4.1 Introduction

The AB1755 testbed is meant to be an inter-organizational demonstration project to provide a set of interacting data nodes compliant with the OWIA standards and conventions specified in the SRD. The interactions will be those necessary to implement the IOC use-cases specified in Table 4.1.

4.2 Concept of Operations

The concept of operations for the testbed is depicted in Figure 1.2. Illustration of the OWIA federation concept-of-operations with a triumvirate governance structure of general partners (GP) supported by interacting with a stakeholder working group (SWG) and a technical working group (TWG). The federation is comprised of dedicated OWIA system implementations to enable individual data providers to independently integrate the OWIA into their existing operations. Shared OWIA system implementations provide the flexibility for the harvesting non-compliant data sources without insisting that the data producers be OWIA-compliant.

4.3 Project Management Plan

The project management plan (PMP) implements the project charter listed below. The project lead is Gary Darling. He is supported by the OWIA Technical Working Group and the IOC Data Node Operators.

4.3.1 Project Charter

The project charter is a DWR management tool with fields populated by the information contained below.

4.3.1.1 Objective

The objective of the testbed is to demonstrate the ability of OWIA-compliant data nodes to realize the mandates of AB1755. The demonstration will be to instantiate a sufficient set of OWIA data nodes to implement the IOC use-cases and maintain the SRD and SOP documents to reflect the results.

4.3.1.2 Scope

(1) In Scope:
  (1) SOPs, SRD
  (2) IOC Use-case implementations
  (3) Data node design descriptions (as examples, not prescriptions)
(2) Out of Scope:
  (1) Design of Testbed Data Nodes. This is the responsibility of the data node operators.

4.3.1.3 Deliverables

(1) IOC Use-case implementations
(2) SOP and SRD updates
(3) Data node design descriptions (as examples, not prescriptions)

4.3.1.4 Schedule

Task or Milestone	2018			2019											
	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
IOC															
Kickoff	X														
Preliminary Design Review				X											
Interim Demonstration							X								
Final Design Review										X					
Final Demonstration													X		

Figure 4.1: Preliminary schedule.

4.3.2 Governance Structure

The testbed governance structure is depicted in Figure **??**.

1. General Partners (GP): Project manager acting.
2. Technical Working Group (TWG): As current.
3. Stakeholder Working Group (SWG): Represented by stakeholder use cases document.

4.3.3 Technical Baseline

1. **System Requirements (SRD):** Completed draft of System Requirements Document (SRD) Technical Requirements Chapter.
2. **Stakeholder Use Cases**: two (2) families of Use Cases from the CLEE report (`https://www.law.berkeley.edu/wp-content/uploads/2018/01/DFWD-Use-Cases.pdf`).
   (a) **Input**: Family A: use cases 1, 4, 17 and 20
   (b) **Input**: Family B: use cases 10 and 19
   (c) **Output**: Defined set of figures, tables and analyses required to support these Families.
3. **Node Designs**: One or more candidate *data node*● designs for implementation based on the requirements baseline,

4.3.3.1 Initial Operating Capability (IOC) Definition

4.3.3.2 Full Operating Capability (FOC) Definition

TBD

4.3.4 Measures of Success

The role of measures of success is illustrated in Figure 1.3 and defined for this project in Table 4.1.

Problem Statement			Methodology			Measure of Success
Use-case	Domain	Sectors	Data Node(s)	Input Data Source	SOPs	Output Information Products
Water Balance	California WY2010-2015	All	CCA, Regional Offices, CKAN, WaDE	Regional Office Spreadsheets	QC-1000,2000,3000 series	Water Plan Document Elements + Published Data
Water Budget	San Diego	Urban	CCA, CKAN	DWR UWMP Tables + TBD	QC-1000,2000,3000 series	UWMP Document Elements + Published Data
	Tulare Lake	Agriculture, Environmental	CCA, LBNL, NASA	DWR Appendix A, + TBD	QC-1000,2000,3000 series	Water Budget Example + Published Data
	Central Coast	Agriculture, Environmental	CCA, LBNL, NASA	DWR Appendix A, + TBD	QC-1000,2000,3000 series	Water Budget Example + Published Data
Water Quality	San Joaquin River	Agriculture	CEDEN	CEDEN Data Templates	QC-1000,2000,3000 series	Published data in association with use-case
Water Quality(Use-case 8, Environmental)	Regional Water Board 1, 2, 5	Environmental	CWMW, LBNL	EcoAtlas, CIWQS, CRAM, USACOE	QC-1000,2000,3000 series	Compliance Reports for 401 and 404 programs + published data

Table 4.1: IOC definition regarding problem definition, methodology and measures of success.

4.4 Deliverables Details

4.4.1 Use-case Implementations

4.4.1.1 Water Balance

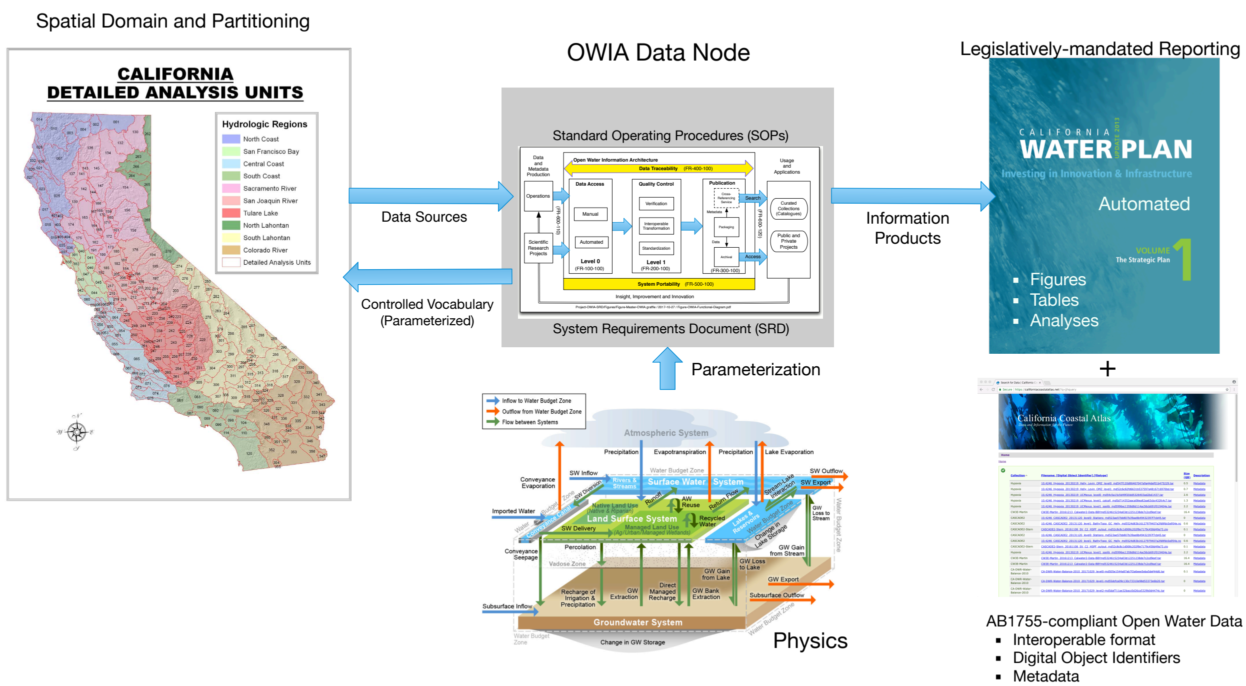

Figure 4.2: Workflow for California Plan WY2010-2015 example.

4.4.1.1.1 WaDE Integration: WY2010-2015

		Year 2011 CategoryE				2012 CategoryE				2013 CategoryE				2014 CategoryE				2015 CategoryE			
		AWU	DEP	NW1	NW2	AWU	DEP	NW1	NW2	AWU	DEP	NW1	NW2	AWU	DEP	NW1	NW2	AWU	DEP	NW1	NW2
		KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt	KAcreFt
HR_NAME	CategoryA	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum
Central Coast	Agriculture	922.0	818.4	818.4	818.4	1012.4	902.3	902.3	902.3	1258.9	1088.2	1088.2	1088.2	1269.6	1033.6	1033.6	1033.6	1088.4	975.2	975.2	975.2
	Instream Flow Requirements	25.4	0.0	0.0	0.0	25.2	0.0	0.0	0.0	24.5	0.0	0.0	0.0	12.0	0.0	0.0	0.0	13.2	0.0	0.0	0.0
	Managed Wetlands	0.4	0.4	0.4	0.4	0.0	0.0	0.0	0.0	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	254.3	141.6	141.7	141.6	276.9	164.0	164.0	164.0	284.2	161.5	162.3	161.5	277.1	171.3	171.3	171.3	244.9	137.5	137.8	137.5
	Wild and Scenic River	154.8	0.0	0.0	0.0	31.3	30.1	30.1	30.1	43.4	0.0	0.0	0.0	13.4	13.2	13.2	13.2	28.7	0.0	0.0	0.0
	All	1356.9	960.4	960.5	960.4	1345.8	1096.4	1096.4	1096.4	1611.4	1250.1	1250.9	1250.1	1572.5	1218.5	1218.5	1218.5	1375.6	1113.1	1113.4	1113.1
Colorado River	Agriculture	3637.4	3150.1	3227.0	3227.0	4217.1	3664.6	3739.2	3739.2	4616.2	3856.5	4052.4	4052.4	4750.0	3963.0	4176.3	4176.3	3816.4	3308.9	3380.7	3380.7
	Instream Flow Requirements	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Managed Wetlands	44.4	44.4	44.4	44.4	44.4	44.4	44.4	44.4	30.3	30.3	30.3	30.3	45.2	45.2	45.2	45.2	44.5	44.5	44.5	44.5
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	476.5	145.7	147.2	147.2	590.9	204.3	206.3	206.3	338.2	203.1	205.0	205.0	302.3	195.1	198.0	198.0	269.1	177.6	179.9	179.9
	Wild and Scenic River	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	All	4158.3	3340.2	3418.6	3418.6	4852.4	3913.3	3989.9	3989.9	4984.7	4089.9	4287.7	4287.7	5097.5	4203.3	4419.5	4419.5	4130.0	3531.0	3605.1	3605.1
North Coast	Agriculture	732.2	524.9	621.0	621.0	794.6	657.8	700.2	700.2	782.3	661.8	686.5	686.5	771.3	651.5	672.6	672.6	729.2	614.3	636.5	636.5
	Instream Flow Requirements	1816.6	1729.6	1729.6	1729.6	1612.5	1506.4	1506.4	1506.4	1450.4	1379.1	1379.1	1379.1	1256.8	1225.3	1225.3	1225.3	1258.9	1222.6	1222.6	1222.6
	Managed Wetlands	186.9	138.9	180.9	180.9	203.9	146.0	172.4	172.4	192.0	124.6	140.9	140.9	131.0	83.7	98.5	98.5	121.4	76.4	96.8	96.8
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	139.1	64.9	64.9	64.9	140.4	65.0	65.2	65.2	148.8	73.2	73.2	73.2	136.7	71.2	71.4	71.4	125.0	65.7	65.7	65.7
	Wild and Scenic River	23250.6	23250.6	23250.6	23250.6	15692.8	15692.8	15692.8	15692.8	12873.2	12873.2	12873.2	12873.2	7824.6	7824.6	7824.6	7824.6	11650.6	11650.6	11650.6	11650.6
	All	26125.4	25708.9	25847.0	25847.0	18444.2	18068.0	18137.0	18137.0	15446.7	15111.9	15152.9	15152.9	10120.4	9856.3	9892.4	9892.4	13885.1	13629.6	13672.2	13672.2
North Lahontan	Agriculture	295.1	252.9	253.3	253.3	331.9	285.4	286.2	286.2	353.6	301.1	301.7	301.7	362.4	306.4	306.9	306.9	292.1	249.3	249.7	249.7
	Instream Flow Requirements	82.9	82.9	82.9	82.9	84.4	84.4	84.4	84.4	84.5	84.5	84.5	84.5	81.1	81.1	81.1	81.1	40.9	40.9	40.9	40.9
	Managed Wetlands	18.8	10.8	12.5	12.5	22.4	15.2	15.7	15.7	22.0	15.0	15.4	15.4	22.5	15.7	15.9	15.9	21.3	14.4	14.9	14.9
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	29.8	9.1	9.1	9.1	31.6	9.5	9.5	9.5	31.4	9.2	9.2	9.2	29.7	8.1	8.1	8.1	25.8	7.6	7.6	7.6
	Wild and Scenic River	470.8	124.9	124.9	124.9	129.3	37.4	37.4	37.4	147.6	39.1	39.1	39.1	95.8	27.4	27.4	27.4	93.1	22.7	22.7	22.7
	All	897.4	480.6	482.7	482.7	599.6	431.9	433.2	433.2	639.1	448.9	449.9	449.9	591.5	438.7	439.4	439.4	473.2	334.9	335.8	335.8
Sacramento River	Agriculture	6594.4	4551.4	5449.3	5449.3	7558.1	5456.8	6241.6	6241.6	7928.2	5745.1	6633.6	6633.6	7166.9	5553.1	6043.2	6043.2	6791.8	5545.6	5852.4	5852.4
	Instream Flow Requirements	5051.9	0.0	2804.6	2804.6	4642.8	0.0	2813.3	2813.3	4568.8	0.0	2804.6	2804.6	4058.8	0.0	2718.4	2718.4	3827.6	0.0	2716.3	2716.3
	Managed Wetlands	578.2	146.4	370.3	370.3	637.0	177.9	401.7	401.7	613.9	193.6	392.4	392.4	641.5	202.6	402.9	402.9	612.6	197.8	390.5	390.5
	Required Delta Outflow	7384.5	7384.5	7384.5	7384.5	5295.9	5295.9	5295.9	5295.9	4498.5	4498.5	4498.5	4498.5	3998.7	3998.7	3998.7	3998.7	3709.8	3709.8	3709.8	3709.8
	Urban	920.8	362.3	694.6	694.6	965.5	353.3	731.1	731.1	1004.0	372.8	762.0	762.0	886.8	309.5	669.7	669.7	784.0	226.9	599.2	599.2
	Wild and Scenic River	5458.8	0.0	514.9	514.9	2734.4	0.0	177.4	177.4	2436.1	0.0	145.5	145.5	1455.6	0.0	72.9	72.9	1642.3	0.0	74.4	74.4
	All	25988.6	12444.6	17218.2	17218.2	21833.7	11283.9	15661.0	15661.0	21049.5	10810.0	15236.6	15236.6	18208.3	10063.9	13905.8	13905.8	17368.1	9680.1	13342.6	13342.6
San Francisco Bay	Agriculture	114.2	100.1	103.3	100.1	154.2	126.6	133.8	126.6	136.2	118.7	122.1	118.8	146.4	130.3	134.5	130.3	151.0	133.0	136.7	133.0
	Instream Flow Requirements	21.8	21.8	21.8	21.8	19.9	19.9	19.9	19.9	12.9	12.9	12.9	12.9	12.9	12.9	12.9	12.9	12.9	12.9	12.9	12.9
	Managed Wetlands	53.3	53.3	53.3	53.3	60.4	60.4	60.4	60.4	58.3	58.3	58.3	58.3	60.6	60.6	60.6	60.6	59.1	59.1	59.1	59.1
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	1137.3	733.2	737.2	737.2	1193.6	787.2	791.2	791.2	1206.4	774.1	778.1	778.1	1169.0	816.5	816.9	816.9	1056.9	757.8	759.6	759.6
	Wild and Scenic River	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	All	1326.6	908.4	915.6	912.4	1428.1	994.1	1005.3	998.1	1413.8	964.0	971.4	968.1	1388.9	1020.3	1024.9	1020.7	1279.9	962.8	968.3	964.6
San Joaquin River	Agriculture	7755.8	6064.0	6668.8	6668.8	8703.7	7263.6	8010.6	8010.6	8502.5	7252.2	7966.0	7966.0	8457.5	7314.5	8060.4	8060.4	10503.6	9076.4	9917.0	9917.0
	Instream Flow Requirements	983.1	0.0	385.2	385.2	596.4	0.0	171.2	171.2	569.2	0.0	167.3	167.3	333.3	0.0	121.6	121.6	374.8	0.0	121.4	121.4
	Managed Wetlands	500.6	219.0	500.2	500.2	529.9	250.2	529.5	529.5	533.8	251.9	529.4	529.4	535.6	261.4	531.5	531.5	546.4	241.3	500.2	500.2
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	647.8	246.1	340.2	340.2	685.1	261.8	395.8	395.8	685.8	267.3	373.9	373.9	654.6	247.2	382.0	382.0	617.4	229.1	310.0	310.0
	Wild and Scenic River	3860.3	0.0	2220.3	2220.3	911.6	0.0	551.5	551.5	1092.1	0.0	684.9	684.9	581.5	0.0	378.3	378.3	528.2	0.0	379.4	379.4
	All	13747.6	6529.1	10114.7	10114.7	11426.7	7775.6	9658.6	9658.6	11383.4	7771.4	9721.5	9721.5	10562.5	7823.1	9473.8	9473.8	12570.4	9546.8	11228.0	11228.0
South Coast	Agriculture	654.9	610.4	610.1	610.4	679.9	634.4	634.9	634.4	985.4	856.3	856.8	856.3	1060.5	920.2	922.0	920.2	690.6	646.4	645.7	646.4
	Instream Flow Requirements	5.7	0.0	0.0	0.0	5.7	0.0	0.0	0.0	5.8	0.0	0.0	0.0	5.8	0.0	0.0	0.0	5.8	0.0	0.0	0.0
	Managed Wetlands	33.0	33.0	33.0	33.0	35.4	35.4	35.4	35.4	31.7	31.7	31.7	31.7	34.0	34.0	34.0	34.0	31.1	31.1	31.1	31.1
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	3530.1	2610.0	2610.3	2610.0	3793.8	2860.0	2859.5	2860.0	3966.5	3015.5	3015.0	3015.5	3992.2	3072.5	3070.7	3072.5	3438.7	2581.5	2582.3	2581.5
	Wild and Scenic River	202.3	0.0	187.7	0.0	27.1	0.0	26.2	0.0	12.3	0.0	11.8	0.0	26.0	0.0	25.6	0.0	12.7	0.0	12.3	0.0
	All	4426.0	3253.4	3441.1	3253.4	4541.9	3529.8	3556.0	3529.8	5001.7	3903.5	3915.3	3903.5	5118.5	4026.7	4052.3	4026.7	4178.9	3259.0	3271.4	3259.0
South Lahontan	Agriculture	332.5	287.3	286.3	287.3	355.7	308.1	307.9	308.1	453.2	351.5	351.4	351.5	444.2	343.3	343.3	343.3	329.9	284.5	284.5	284.5
	Instream Flow Requirements	99.0	80.2	80.2	80.2	68.6	49.8	49.8	49.8	73.6	47.9	47.9	47.9	67.3	49.7	49.7	49.7	62.9	45.5	45.5	45.5
	Managed Wetlands	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	217.8	112.1	113.1	112.1	240.1	125.3	125.3	125.3	227.7	134.4	134.5	134.4	224.6	117.5	117.5	117.5	179.6	93.0	93.0	93.0
	Wild and Scenic River	4.3	0.0	4.3	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.8	0.0	1.5	0.0	1.5	0.0	1.5	0.0	1.5	0.0
	All	653.6	479.6	483.9	479.6	665.3	483.2	483.9	483.2	755.3	533.8	534.6	533.8	737.6	510.5	512.0	510.5	573.9	423.0	424.5	423.0
Tulare Lake	Agriculture	11616.7	9186.6	9186.6	9186.6	12124.1	10341.6	10341.6	10341.6	11659.7	10109.8	10109.9	10109.8	11572.3	10146.3	10145.9	10146.3	11223.3	9897.8	9897.8	9897.8
	Instream Flow Requirements	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Managed Wetlands	84.8	48.6	48.6	48.6	92.8	57.0	57.0	57.0	93.7	61.0	61.0	61.0	95.8	63.9	64.0	63.9	91.8	57.6	57.6	57.6
	Required Delta Outflow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Urban	571.2	194.9	193.5	194.9	599.4	200.8	200.8	200.8	584.5	199.5	199.0	199.5	596.7	203.5	203.0	203.5	468.9	158.1	157.9	158.1
	Wild and Scenic River	3101.1	0.0	0.0	0.0	650.7	0.0	32.8	0.0	524.7	0.0	0.0	0.0	506.4	0.0	0.0	0.0	287.3	0.0	0.0	0.0
	All	15373.8	9430.1	9428.7	9430.1	13468.7	10599.4	10632.2	10599.4	12864.3	10370.3	10369.9	10370.3	12771.2	10413.7	10412.9	10413.7	12071.3	10113.5	10113.3	10113.5

Table 4.2: Information product example for California Plan WY2010-2015 (Working Draft).

4.4.1.2 Water Budget

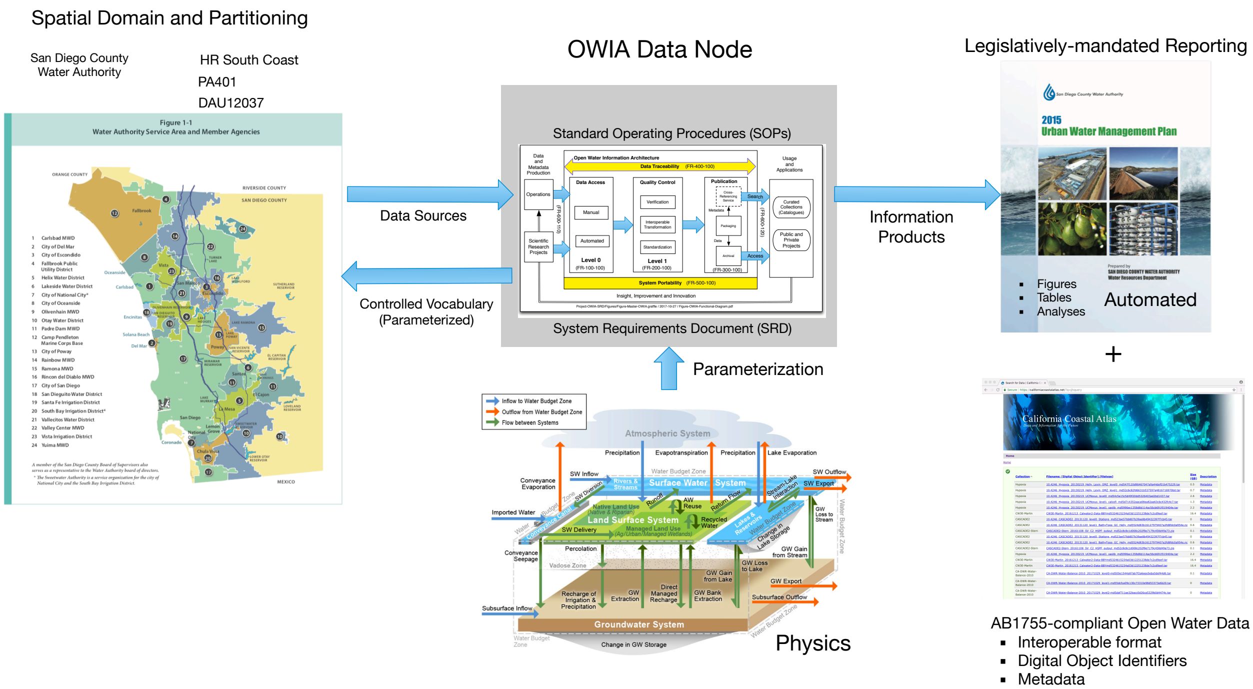

Figure 4.3: Workflow for UWMP use-case with the San Diego County Water Authority example.

4.4.1.2.1 South Coast Hydrologic Region: (PA404, DAU12037), San Diego County Water Authority Urban Water Management Plan (UWMP)

4.4.1.2.2 Tulare Lake

4.4.1.2.3 Central Coast

4.4.1.3 Water Quality

4.4.1.4 Environmental

One or more design alternatives compliant with the *Requirements Baseline* and the *Concept of Operations*.

4.4.2 Data Node Descriptions

An ***OWIA data node*** is the ***set of people and the SRD-compliant cyberinfrastructure*** used to service a specified set of use-cases by ingesting source data, transforming the data by executing ***SOPs*** and resulting in *Output Information Products* such as those of Table 4.1 and Figures 4.2, 4.3. The ***scope of a Data Node*** is bounded by the use-cases that it is responsible for: in whole or in part.

4.4.2.1 CKAN

4.4.2.2 DKAN

4.4.2.3 California Coastal Atlas (CCA)

4.4.3 Document Maintenance

4.4.3.1 SRD

4.4.3.2 SOP

Appendix A

Water Balance SOP Example Implementation: QC-1000.R

```
# =================================================================
# Filename: QC-1000.R
# Author: John Helly (hellyj@ucsd.edu)
# Purpose: Ingest *.csv files from DWR water balance spreadsheets
# =================================================================
rm(list = ls())
library(ggplot2)
library(plyr)
library(reshape2)
library(tables)
#
graphics.off()
# ===========================================================================
# Set directories
# ===========================================================================
PROJECT_ROOT            ='/Users/hellyj/Archive-local'
SRC_ROOT                = paste(PROJECT_ROOT,      '/Project-OWIA-WaterBalance-src/R', sep='')
DATA_INPUT_ROOT         = paste(PROJECT_ROOT,      '/Project-OWIA-WaterBalance-data/level1', sep='')
DATA_OUTPUT_ROOT        = paste(PROJECT_ROOT,      '/Project-OWIA-WaterBalance-data/level2', sep='')
SPATIAL_METADATA        = paste(DATA_INPUT_ROOT,  '/Spatial-Assignments-Georeferencing.csv', sep='')
OUTPUT_STANDARDIZED     = paste(DATA_OUTPUT_ROOT, '/CA-DWR-WaterBalance-QC-1000-2011-2015-Standardized.csv', sep='')
TABLE_OUTPUT_HOME       = paste(PROJECT_ROOT,      '/Project-OWIA-WaterBalance-Documentation/Tables',  sep='')
FIGURE_OUTPUT_HOME      = paste(PROJECT_ROOT,      '/Project-OWIA-WaterBalance-Documentation/Figures', sep='')
CONTROLLED_VOCABULARY   = paste(PROJECT_ROOT,       '/Project-OWIA-WaterBalance-data/level0/CV/CV-100.csv', sep='')
# =================================================================
# Metadata
# =================================================================
COPYRIGHT   = paste("hellyj@ucsd.edu", expression(copyright), Sys.time())
OUTPUT_HOME = FIGURE_OUTPUT_HOME
SOURCE_FILE = paste(SRC_ROOT, '/QC-1000.R', sep='')
```

```
METADATA_01 = paste(SOURCE_FILE,' / ', COPYRIGHT)
# ==========================================================================
# Functions
# ==========================================================================
source(paste(SRC_ROOT,'/functions/f_QC_1000_Reshape.R',                sep=''))
source(paste(SRC_ROOT,'/functions/f_CategoryB.R',                      sep=''))
source(paste(SRC_ROOT,'/functions/f_QC_1000_PlotDelta.R',              sep=''))
source(paste(SRC_ROOT,'/functions/f_QC_1000_Table_100.R',              sep=''))
source(paste(SRC_ROOT,'/functions/f_QC_1000_Table_200.R',              sep=''))
source(paste(SRC_ROOT,'/functions/f_QC_1000_PlotDelta_DAU_PA_YEAR.R', sep=''))
source(paste(SRC_ROOT,'/functions/f_ControlledVocabulary.R',           sep=''))
# ==========================================================================
# Define data sources and reshape the data
# ==========================================================================
source(paste(SRC_ROOT,'/Input-QC-1000-Level1-2015.R', sep=''),echo=FALSE)
source(paste(SRC_ROOT,'/Input-QC-1000-Level1-2014.R', sep=''),echo=FALSE)
source(paste(SRC_ROOT,'/Input-QC-1000-Level1-2013.R', sep=''),echo=FALSE)
source(paste(SRC_ROOT,'/Input-QC-1000-Level1-2012.R', sep=''),echo=FALSE)
source(paste(SRC_ROOT,'/Input-QC-1000-Level1-2011.R', sep=''),echo=FALSE)
source(paste(SRC_ROOT,'/Input-QC-1000-level1-2010.R', sep=''),echo=FALSE)
# =========================================================================
# 1. Concatenate the data by rows (i.e., rbind)
# 2. Populate the HR,PA fields from the reference list of [HR, PA, DAU]
# 3. Georeference DAUs
# 4. Standardize nomenclature
# ==========================================================================
source(paste(SRC_ROOT,'/Input-QC-1000-Rbind-2015.R', sep=''))
source(paste(SRC_ROOT,'/Input-QC-1000-Rbind-2014.R', sep=''))
source(paste(SRC_ROOT,'/Input-QC-1000-Rbind-2013.R', sep=''))
source(paste(SRC_ROOT,'/Input-QC-1000-Rbind-2012.R', sep=''))
source(paste(SRC_ROOT,'/Input-QC-1000-Rbind-2011.R', sep=''))
source(paste(SRC_ROOT,'/Input-QC-1000-Rbind-2010.R', sep=''))
# ==========================================================================
# Controlled Vocabulary standardization
# ==========================================================================
MASTER                  = rbind(MASTER_2010,MASTER_2011,MASTER_2012,MASTER_2013,MASTER_2014,MASTER_2015)
MASTER                  = subset(MASTER, CategoryC != "") # Miscoded values of CategoryC (and CategoryA)
MASTER                  = subset(MASTER, DAU != 'X')
MASTER$KAcreFt          = as.numeric(MASTER$KAcreFt)
MASTER[is.na(MASTER)] = 0 # Set 'NA' to 0
# ===========================================================================
SPATIAL                        = read.table(SPATIAL_METADATA, sep=',', skip=0,header=TRUE, stringsAsFactors=FALSE)
SPATIAL                        = SPATIAL[, (names(SPATIAL) %in% c('DAU','DAU_NAME','DAU_COUNTY',
                                                                   'PA',
                                                                   'HR_CODE','HR_NAME',
                                                                   'Longitude','Latitude'))]
SPATIAL        = SPATIAL[order(SPATIAL$DAU),]
SPATIAL_NODUP = SPATIAL[!duplicated(SPATIAL[,
                  c('DAU','DAU_NAME','PA','HR_CODE','HR_NAME','Longitude','Latitude')
```

```
                    ]),]
Duplicates = tabular(Factor(DAU)~ (n=1), data=SPATIAL_NODUP)
#
# Add missing PA
#
MISSING            = data.frame('DAU20824','Turlock',      '608','6',       'San Joaquin River','-999',     '-999')
names(MISSING)     =          c('DAU',     'DAU_NAME',     'PA', 'HR_CODE', 'HR_NAME',           'Longitude','Latitude')
df_DAU20850        = data.frame('DAU20850','Turlock',      '608','6',       'San Joaquin River','-999',     '-999')
df_DAU20924        = data.frame('DAU20924','Turlock Lake', '608','6',       'San Joaquin River','-999',     '-999')
df_DAU20950        = data.frame('DAU20950','Turlock Lake', '608','6',       'San Joaquin River','-999',     '-999')
names(df_DAU20850) = c('DAU',     'DAU_NAME',     'PA', 'HR_CODE', 'HR_NAME',          'Longitude','Latitude')
names(df_DAU20924) = c('DAU',     'DAU_NAME',     'PA', 'HR_CODE', 'HR_NAME',          'Longitude','Latitude')
names(df_DAU20950) = c('DAU',     'DAU_NAME',     'PA', 'HR_CODE', 'HR_NAME',          'Longitude','Latitude')

MISSING = rbind(MISSING, df_DAU20850, df_DAU20924, df_DAU20950)
SPATIAL_NODUP_608 = rbind(SPATIAL_NODUP, MISSING)
# ============================================================================
MASTER_GC                    = merge(SPATIAL_NODUP_608, MASTER, by = 'DAU' ) # Note to eliminate wrong assignments
#
MASTER_GC[grepl('Scenic',         MASTER_GC$CategoryA),]$CategoryA = 'Wild and Scenic River'
MASTER_GC[grepl('Delta',          MASTER_GC$CategoryA),]$CategoryA = 'Required Delta Outflow'
MASTER_GC[grepl('Water_Supplies', MASTER_GC$CategoryA),]$CategoryA = 'Water Supplies'
MASTER_GC[grepl('Instream',       MASTER_GC$CategoryA),]$CategoryA = 'Instream Flow Requirements'
MASTER_GC[grepl('Wetlands',       MASTER_GC$CategoryA),]$CategoryA = 'Managed Wetlands'
#
MASTER_GC[grepl('Applied Water',  MASTER_GC$CategoryC) &
          grepl('Groundwater'  ,  MASTER_GC$CategoryC),]$CategoryC = 'Applied Water - Groundwater'
MASTER_GC$CategoryC = gsub("_", "-",   MASTER_GC$CategoryC)
MASTER_GC$CategoryC = gsub("::", "-",  MASTER_GC$CategoryC)
MASTER_GC$CategoryC = gsub("Wild &", "Wild and", MASTER_GC$CategoryC)
#
#
# ===========================================================================
# Delete computed rows and non-balance CategoryA rows
# ===========================================================================
A = subset(MASTER_GC,  !grepl('Total',                       CategoryC) &
                       !grepl('Applied Water Use',           CategoryC) &
                       !grepl('Net Water Use',               CategoryC) &
                       !grepl('Depletion',                   CategoryC) &
                       !grepl('Conveyance Applied Water Use',CategoryC) &
                       !grepl('Conveyance Net Water Use',    CategoryC) &
                       !grepl('Conveyance Depletion',        CategoryC) &
                       !grepl('Urban Waste Water Produced',  CategoryA) &
                       !grepl('Water Use Totals',            CategoryA) &
                       !grepl('Portfolio',                   CategoryA) &
                       !grepl('Precipitation',               CategoryC))
# ===========================================================================
# Read Controlled Vocabulary
# ===========================================================================
```

```
CV = read.table(CONTROLLED_VOCABULARY, sep=',', header=TRUE, stringsAsFactors=FALSE)
#
A  = subset(A, CategoryA != '') # Null values in manual conversion effort
A[A$CategoryA=='Urban'                    & A$CategoryB=='15',]$CategoryB='15a' # QC-2500.R corrections from rollups
A[A$CategoryA=='Urban'                    & A$CategoryB=='18',]$CategoryB='18a' # QC-2500.R corrections from rollups
A[A$CategoryA=='Managed Wetlands'         & A$CategoryB=='17',]$CategoryB='17a' # QC-2500.R corrections from rollups
A[A$CategoryA=='Managed Wetlands'         & A$CategoryB=='8',]$CategoryB='8a' # QC-2500.R corrections from rollups
A[A$CategoryA=='Instream Flow Requirements' & A$CategoryB=='3',]$CategoryB='3a' # QC-2500.R corrections from rollups
#
C  = merge(A, CV, by=c('CategoryA','CategoryB'))  # Merge in CategoryD based on CV
 C$CategoryC = C$CategoryC.y

# ======================================================================
# Write the data out for subsequent procedures
# ======================================================================
 write.table(C,OUTPUT_STANDARDIZED,         sep=',',row.names=FALSE, quote=TRUE)
# ======================================================================
# Debugging
# ======================================================================
D = subset(C, DAU=='DAU33936')
 D$CategoryABCD    = paste(D$CategoryA,'/', D$CategoryB,'/', D$CategoryC, '/', D$CategoryD)
 Table_Summary_DAU = tabular(Factor(Year) * (Factor(CategoryABCD)+1) ~
                              (1 + Factor(HR_NAME) *
                               Factor(PA))     * KAcreFt*sum,
                               data=D)
```

Appendix B

Action Items

1: Figure for Regulatory framework and operations (Sara)
2: JH Edit TDD to reflect table changes and provide a revised TDD to Kamyar after TWG review
3: Consider for describing vested interests (i.e. public, private, NGO, quasi-governmental) (Gary)
4: How to implement an SWG (Gary)
5: How to engage other interested parties (Sara mentions MOUs in other national activities) (Gary)
6: David H., JH: Data and metadata (schema,..)
7: Forest, JH: Remote sensing + NASA directions
8: Paul H, JH: Reconcile, unify Water Balance and Water Budget
9: UWMP and AGWMP considered as a joint problem in terms of Water Budget use-case (JH)
10: Regional offices as data nodes? (JH)
11: Common parameterization and controlled Vocab (Sara)
12: Re-visit use-cases in smaller conversation (Charu)

Appendix C

Application to AB1755

C.1 Application-AB1755: OWIA application to AB1755 objectives

Table C.1: Traceability of AB1755 objectives (columns) to OWIA functional requirements (rows).

Identifier	Name	Data Sharing	Documentation	Quality Control	Public Access	Open-source platforms and decision support tools
FR-100-100	Data Acquisition	X				
FR-100-110	*-Manual-	X				
FR-100-120	*-Automated-	X				
FR-200-100	Quality Control-*-	X		X		
FR-200-110	*-Verification-	X		X		
FR-200-120	*-*-Documentation	X	X	X		
FR-200-130	*-*-Reproducibility	X		X		
FR-200-140	*-*-Data Traceability	X		X		
FR-200-150	*-Standardization-	X	X	X		X
FR-200-160	*-*-File-naming Conventions	X	X	X		X
FR-200-170	*-Interoperable Transformation-	X		X		X
FR-200-180	*-*-Separation of Data and Computation	X		X		X
FR-200-190	*-*-Data Interoperability	X	X	X		X
FR-200-200	*-*-Products or Resources	X		X		X
FR-300-100	Publication-*-	X	X		X	
FR-300-110	*-Cross-Referencing-Service-	X	X		X	
FR-300-120	*-*-Assignment of Digital Object Identifiers	X	X		X	
FR-300-130	*-Packaging-	X			X	
FR-300-140	*-*-Compression Methods	X			X	
FR-300-150	*-*-Archive File Formatting	X			X	
FR-300-160	*-Archival-	X			X	
FR-300-170	*-*-Open Access Distribution	X			X	
FR-400-100	Data Traceability-*-	X	X		X	
FR-400-110	*-Metadata Production-	X	X		X	
FR-400-120	*-Intellectual Property Rights Management-	X	X		X	
FR-400-130	*-Public Law Compliance-	X	X		X	
FR-400-140	*-Licensing-	X	X		X	
FR-400-150	*-Liability-	X	X		X	
FR-400-160	*-Searching-	X			X	
FR-400-170	*-*-Cross-referencing System Integration	X			X	
FR-400-180	*-*-Search Engine Optimization	X			X	
FR-400-190	*-Version Control-	X	X			
FR-400-200	*-*-Binary Data	X	X			
FR-400-210	*-*-Non-Binary Data	X	X			
FR-400-220	*-Anomaly Reporting-	X	X			
FR-500-100	System Portability-*-					X
FR-500-110	*-Backup and Restore-					X
FR-500-120	*-Platform Portability-					X
FR-600-100	External Interfaces-*-	X				X
FR-600-110	*-Data and Metadata Acquisition-	X				X
FR-600-120	*-Data and Metadata Distribution-	X				X

Listing 4.1: Water use equations.

```
f_Equations = function(df){
# ==========================================================================
# Water Use by Sector: Translation of TH formulas
# ==========================================================================
EQ = data.frame(ncols=1)
#
EQ$AWUAG =        sum(df[df$CategoryD=='AG1',  ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG2',  ]$KAcreFt)

EQ$NW1AG = EQ$AWUAG - sum(df[df$CategoryD=='AG5',  ]$KAcreFt) -
                      sum(df[df$CategoryD=='AG7',  ]$KAcreFt) -
                      sum(df[df$CategoryD=='AG8',  ]$KAcreFt)

EQ$DEPAG =        sum(df[df$CategoryD=='AG3',  ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG4',  ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG6',  ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG9A', ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG9B', ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG9C', ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG9E', ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG9F', ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG10A',]$KAcreFt) +
                  sum(df[df$CategoryD=='AG10B',]$KAcreFt) +
                  sum(df[df$CategoryD=='AG12', ]$KAcreFt)

EQ$NW2AG = EQ$DEPAG + sum(df[df$CategoryD=='AG11A',]$KAcreFt) +
                      sum(df[df$CategoryD=='AG11B',]$KAcreFt) +
                      sum(df[df$CategoryD=='AG11C',]$KAcreFt) +
                      sum(df[df$CategoryD=='AG11D',]$KAcreFt)

#
EQ$DEPAGC =       sum(df[df$CategoryD=='AG17', ]$KAcreFt) +
                  sum(df[df$CategoryD=='AG18A',]$KAcreFt) +
                  sum(df[df$CategoryD=='AG18B',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG18C',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG18D',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG18E',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG18F',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG19A',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG19B',]$KAcreFt) +
                    sum(df[df$CategoryD=='AG23', ]$KAcreFt)

EQ$NW2AGC=EQ$DEPAGC + sum(df[df$CategoryD=='AG20A',]$KAcreFt) +
                      sum(df[df$CategoryD=='AG20B',]$KAcreFt) +
                        sum(df[df$CategoryD=='AG20C',]$KAcreFt)
EQ$NW1AGC=EQ$NW2AGC

EQ$AWUAGC=EQ$NW1AGC +  sum(df[df$CategoryD=='AG21', ]$KAcreFt) +
                       sum(df[df$CategoryD=='AG22', ]$KAcreFt)
#
# Urban
#
EQ$AWUURB=          sum(df[df$CategoryD=='URB1', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB2', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB3', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB4', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB5', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB6', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB7', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB8', ]$KAcreFt) +
                  sum(df[df$CategoryD=='URB9', ]$KAcreFt)

EQ$NW1URB=EQ$AWUURB - sum(df[df$CategoryD=='URB12', ]$KAcreFt) -
                  sum(df[df$CategoryD=='URB14', ]$KAcreFt) -
                       sum(df[df$CategoryD=='URB15A', ]$KAcreFt) -
                       sum(df[df$CategoryD=='URB15B',]$KAcreFt) +
```

```
                        sum(df[df$CategoryD=='URB15C',]$KAcreFt)

EQ$DEPURB= sum(df[df$CategoryD=='URB10', ]$KAcreFt) +
        sum(df[df$CategoryD=='URB11', ]$KAcreFt) +
        sum(df[df$CategoryD=='URB13', ]$KAcreFt) +
        sum(df[df$CategoryD=='URB16', ]$KAcreFt) +
        sum(df[df$CategoryD=='URB17A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB17B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB17C',]$KAcreFt) +
                sum(df[df$CategoryD=='URB17D',]$KAcreFt) +
        sum(df[df$CategoryD=='URB17E',]$KAcreFt) +
                sum(df[df$CategoryD=='URB17F',]$KAcreFt) +
                sum(df[df$CategoryD=='URB18A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB18B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB20', ]$KAcreFt)

EQ$NW2URB=EQ$DEPURB + sum(df[df$CategoryD=='URB19A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB19B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB19C',]$KAcreFt) +
                sum(df[df$CategoryD=='URB19D',]$KAcreFt)

EQ$AWUURBC=        sum(df[df$CategoryD=='URB25', ]$KAcreFt) +
                sum(df[df$CategoryD=='URB26A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26C',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26D',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26E',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26F',]$KAcreFt) +
                sum(df[df$CategoryD=='URB27A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB27B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB28A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB28B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB28C',]$KAcreFt) +
                sum(df[df$CategoryD=='URB29', ]$KAcreFt) +
                sum(df[df$CategoryD=='URB30', ]$KAcreFt) +
                sum(df[df$CategoryD=='URB31', ]$KAcreFt)

EQ$NW1URBC=EQ$AWUURBC-sum(df[df$CategoryD=='URB29',]$KAcreFt)-
                sum(df[df$CategoryD=='URB30',]$KAcreFt)

EQ$DEPURBC=        sum(df[df$CategoryD=='URB25',]$KAcreFt)   +
                sum(df[df$CategoryD=='URB26A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26C',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26D',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26E',]$KAcreFt) +
                sum(df[df$CategoryD=='URB26F',]$KAcreFt) +
                sum(df[df$CategoryD=='URB27A',]$KAcreFt) +
                sum(df[df$CategoryD=='URB27B',]$KAcreFt) +
                sum(df[df$CategoryD=='URB31',]$KAcreFt)

EQ$NW2URBC=EQ$DEPURBC+ sum(df[df$CategoryD=='URB28A',]$KAcreFt) +
                 sum(df[df$CategoryD=='URB28B',]$KAcreFt) +
                 sum(df[df$CategoryD=='URB28C',]$KAcreFt)
#
# Managed Wetland
#
EQ$AWUMW=          sum(df[df$CategoryD=='MW1',]$KAcreFt)

EQ$NW1MW=EQ$AWUMW -   sum(df[df$CategoryD=='MW3',]$KAcreFt) -
                        sum(df[df$CategoryD=='MW5',]$KAcreFt) -
                        sum(df[df$CategoryD=='MW6',]$KAcreFt)


EQ$DEPMW=          sum(df[df$CategoryD=='MW2', ]$KAcreFt) +
                sum(df[df$CategoryD=='MW4', ]$KAcreFt) +
                sum(df[df$CategoryD=='MW7A',]$KAcreFt) +
                sum(df[df$CategoryD=='MW7B',]$KAcreFt) +
```

```
                  sum(df[df$CategoryD=='MW7C',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW7D',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW7E',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW7F',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW8A',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW8B',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW10',]$KAcreFt)

EQ$NW2MW=EQ$DEPMW +   sum(df[df$CategoryD=='MW9A',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW9B',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW9C',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW9D',]$KAcreFt)

EQ$AWUMWC=          sum(df[df$CategoryD=='MW15', ]$KAcreFt) +
                  sum(df[df$CategoryD=='MW16A',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW16B',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW16C',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW16D',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW16E',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW16F',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW17A',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW17B',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW18A',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW18B',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW18C',]$KAcreFt) +
                  sum(df[df$CategoryD=='MW19', ]$KAcreFt) +
                  sum(df[df$CategoryD=='MW20', ]$KAcreFt) +
                  sum(df[df$CategoryD=='MW21', ]$KAcreFt)

EQ$NW1MWC= EQ$AWUMWC -sum(df[df$CategoryD=='MW19',  ]$KAcreFt) -
                        sum(df[df$CategoryD=='MW20',  ]$KAcreFt)

EQ$NW2MWC= EQ$AWUMWC- sum(df[df$CategoryD=='MW19', ]$KAcreFt) -
                  sum(df[df$CategoryD=='MW20', ]$KAcreFt)

EQ$DEPMWC=  EQ$AWUMWC-sum(df[df$CategoryD=='MW18A',]$KAcreFt) -
                  sum(df[df$CategoryD=='MW18B',]$KAcreFt) -
                  sum(df[df$CategoryD=='MW18C',]$KAcreFt) -
                        sum(df[df$CategoryD=='MW19', ]$KAcreFt) -
                        sum(df[df$CategoryD=='MW20', ]$KAcreFt)
#
# Instream Flow Requirements
#
EQ$AWUIFR=        sum(df[df$CategoryD=='IFR1', ]$KAcreFt)

EQ$NW1IFR=        sum(df[df$CategoryD=='IFR1', ]$KAcreFt) -
                  sum(df[df$CategoryD=='IFR2', ]$KAcreFt)

EQ$NW2IFR=        sum(df[df$CategoryD=='IFR3A',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR3B',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR3C',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR4A',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR4B',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR4C',]$KAcreFt)

EQ$DEPIFR=        sum(df[df$CategoryD=='IFR3A',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR3B',]$KAcreFt) +
                  sum(df[df$CategoryD=='IFR3C',]$KAcreFt)
#
# Wild and Scenic Rivers
#
EQ$AWUWSR=          sum(df[df$CategoryD=='WSR1',]$KAcreFt)

EQ$NW1WSR=          sum(df[df$CategoryD=='WSR1',]$KAcreFt) -
                  sum(df[df$CategoryD=='WSR2',]$KAcreFt)

EQ$NW2WSR=          sum(df[df$CategoryD=='WSR3A',]$KAcreFt) +
                  sum(df[df$CategoryD=='WSR3B',]$KAcreFt) +
```

```
                    sum(df[df$CategoryD=='WSR3C',]$KAcreFt) +
                    sum(df[df$CategoryD=='WSR4A',]$KAcreFt) +
                    sum(df[df$CategoryD=='WSR4B',]$KAcreFt) +
                    sum(df[df$CategoryD=='WSR4C',]$KAcreFt)

EQ$DEPWSR =         sum(df[df$CategoryD=='WSR3A',]$KAcreFt) +
                    sum(df[df$CategoryD=='WSR3B',]$KAcreFt) +
                    sum(df[df$CategoryD=='WSR3C',]$KAcreFt)
#
# Required Delta Outflow
#
EQ$AWURDO = sum(df[df$CategoryD=='RDO1',]$KAcreFt)
EQ$NW1RDO = sum(df[df$CategoryD=='RDO1',]$KAcreFt)
EQ$NW2RDO = sum(df[df$CategoryD=='RDO2',]$KAcreFt)
EQ$DEPRDO = sum(df[df$CategoryD=='RDO2',]$KAcreFt)
# ======================================================================
# Water Supply
# ======================================================================
EQ$SPLAG =          sum(df[df$CategoryD=='SPL1A', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2A1',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2B1',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2C1',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL3A', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL4A', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL5A', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL6A', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL10A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL11A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL12A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL13A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL14A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL15A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL16A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL17A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL18A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL19A',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2D1',]$KAcreFt)

EQ$SPLMW =          sum(df[df$CategoryD=='SPL1B', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2A2',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2B2',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2C2',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL3B', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL4B', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL5B', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL6B', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL10B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL11B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL12B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL13B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL14B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL15B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL16B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL17B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL18B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL19B',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2D2',]$KAcreFt)

EQ$SPLURB =         sum(df[df$CategoryD=='SPL1C', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2A3',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2B3',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2C3',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL3C', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL4C', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL5C', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL6C', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL10C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL11C',]$KAcreFt) +
```

```
                    sum(df[df$CategoryD=='SPL12C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL13C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL14C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL15C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL16C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL17C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL18C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL19C',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2D3',]$KAcreFt)

EQ$SPLIFR =              sum(df[df$CategoryD=='SPL1D', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2A4',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2B4',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2C4',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL3D', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL4D', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL5D', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL6D', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL10D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL11D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL12D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL13D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL14D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL15D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL16D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL17D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL18D',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL19D',]$KAcreFt)

EQ$SPLWS =               sum(df[df$CategoryD=='SPL1E', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2A5',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2B5',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2C6',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL3E', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL4E', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL5E', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL6E', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL10E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL11E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL12E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL13E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL14E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL15E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL16E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL17E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL18E',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL19E',]$KAcreFt)

EQ$SPLRDO =              sum(df[df$CategoryD=='SPL1F', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2A6',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2B6',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL2C6',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL3F', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL4F', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL5F', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL6F', ]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL10F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL11F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL12F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL13F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL14F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL15F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL16F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL17F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL18F',]$KAcreFt) +
                    sum(df[df$CategoryD=='SPL19F',]$KAcreFt)
#
return(EQ)
```

```
}
```

Listing 4.2: Water use adjustments.

```
f_Adjustments = function(df, level){
#
YR_LIST =  unique(df$Year)
    i = 0
    for (YYYY  in YR_LIST){
# ==============================================================================
# Corrections for Hydrologic Regions
# ==============================================================================
#
if (level=='HR'){
    LIST = unique(df$HR_CODE)
for (ITEM in LIST){
        i = i + 1
        A = subset(df, HR_CODE==ITEM & Year==YYYY)
        if (dim(A)[1]==0){ print(paste('Empty df: YYYY=', YYYY, '/ HR_CODE=', ITEM, sep='')); next }
        print(paste('Start: YYYY=', YYYY, '/ HR=', ITEM, sep=''))
#
A[A$CategoryD=='NW1AG',]$KAcreFt   = sum(A[A$CategoryD=='NW1AG',]$KAcreFt)   - sum(A[A$CategoryD=='AG11A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='AG11B', ]$KAcreFt)
A[A$CategoryD=='NW1AGC',]$KAcreFt = sum(A[A$CategoryD=='NW1AGC',]$KAcreFt) - sum(A[A$CategoryD=='AG20A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='AG20B', ]$KAcreFt)
A[A$CategoryD=='NW1IFR',]$KAcreFt = sum(A[A$CategoryD=='NW1IFR',]$KAcreFt) - sum(A[A$CategoryD=='IFR4A', ]$KAcreFt) -
                                                                                         sum(A[A$CategoryD=='IFR4B', ]$KAcreFt)
A[A$CategoryD=='NW1MW',]$KAcreFt   = sum(A[A$CategoryD=='NW1MW',]$KAcreFt)   - sum(A[A$CategoryD=='MW9A',  ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='MW9B',  ]$KAcreFt)
A[A$CategoryD=='NW1MWC',]$KAcreFt = sum(A[A$CategoryD=='NW1MWC',]$KAcreFt) - sum(A[A$CategoryD=='MW18A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='MW18B', ]$KAcreFt)
A[A$CategoryD=='NW1URB',]$KAcreFt = sum(A[A$CategoryD=='NW1URB',]$KAcreFt) - sum(A[A$CategoryD=='URB19A',]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='URB19B',]$KAcreFt)
A[A$CategoryD=='NW1URBC',]$KAcreFt= sum(A[A$CategoryD=='NW1URBC',]$KAcreFt)- sum(A[A$CategoryD=='URB28A',]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='URB28B',]$KAcreFt)
A[A$CategoryD=='NW1WSR',]$KAcreFt = sum(A[A$CategoryD=='NW1WSR',]$KAcreFt) - sum(A[A$CategoryD=='WSR4A', ]$KAcreFt) -
                                                                                         sum(A[A$CategoryD=='WSR4B', ]$KAcreFt)
A[A$CategoryD=='NW2AG',]$KAcreFt   = sum(A[A$CategoryD=='NW2AG',]$KAcreFt)   - sum(A[A$CategoryD=='AG11A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='AG11B', ]$KAcreFt)
A[A$CategoryD=='NW2AGC',]$KAcreFt = sum(A[A$CategoryD=='NW2AGC',]$KAcreFt) - sum(A[A$CategoryD=='AG20A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='AG20B', ]$KAcreFt)
A[A$CategoryD=='NW2IFR',]$KAcreFt = sum(A[A$CategoryD=='NW2IFR',]$KAcreFt) - sum(A[A$CategoryD=='IFR4A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='IFR4B', ]$KAcreFt)
A[A$CategoryD=='NW2MW',]$KAcreFt   = sum(A[A$CategoryD=='NW2MW',]$KAcreFt)   - sum(A[A$CategoryD=='MW9A',  ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='MW9B',  ]$KAcreFt)
A[A$CategoryD=='NW2MWC',]$KAcreFt = sum(A[A$CategoryD=='NW2MWC',]$KAcreFt) - sum(A[A$CategoryD=='MW18A', ]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='MW18B', ]$KAcreFt)
A[A$CategoryD=='NW2URB',]$KAcreFt = sum(A[A$CategoryD=='NW2URB',]$KAcreFt) - sum(A[A$CategoryD=='URB19A',]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='URB19B',]$KAcreFt)
A[A$CategoryD=='NW2URBC',]$KAcreFt= sum(A[A$CategoryD=='NW2URBC',]$KAcreFt)- sum(A[A$CategoryD=='URB28A',]$KAcreFt) -
                                                                               sum(A[A$CategoryD=='URB28B',]$KAcreFt)
```

```
A[A$CategoryD=='NW2WSR',]$KAcreFt = sum(A[A$CategoryD=='NW2WSR',]$KAcreFt) - sum(A[A$CategoryD=='WSR4A', ]$KAcreFt) -
                                                                              sum(A[A$CategoryD=='WSR4B', ]$KAcreFt)
if (i==1){
       MASTER = A
       }
       else
       {
       MASTER = rbind(MASTER, A)
       print(paste('-End: YYYY=', YYYY, '/ i=', i, sep=''))
     }
   }
}
# ============================================================
# End of HR loop
# ============================================================
#
# ========================================================
# PA adjustments below
# ========================================================
if (level=='PA'){
  LIST = unique(df$PA)
  for (ITEM in LIST){
        i = i + 1
        A = subset(df, PA==ITEM & Year==YYYY)
        if (dim(A)[1]==0){ print(paste('Empty df: YYYY=', YYYY, '/ ITEM=', ITEM, sep='')); next }
        print(paste('Start: YYYY=', YYYY, '/ PA =', ITEM, sep=''))
#
# NW1
#
   A[A$CategoryD=='NW1AG',]$KAcreFt    = A[A$CategoryD=='NW1AG',]$KAcreFt    - A[A$CategoryD=='AG11A',]$KAcreFt
   A[A$CategoryD=='NW1AGC',]$KAcreFt   = A[A$CategoryD=='NW1AGC',]$KAcreFt   - A[A$CategoryD=='AG20A',]$KAcreFt
   A[A$CategoryD=='NW1IFR',]$KAcreFt   = A[A$CategoryD=='NW1IFR',]$KAcreFt   - A[A$CategoryD=='IFR4A',]$KAcreFt
   A[A$CategoryD=='NW1MW',]$KAcreFt    = A[A$CategoryD=='NW1MW',]$KAcreFt    - A[A$CategoryD=='MW9A',]$KAcreFt
   A[A$CategoryD=='NW1MWC',]$KAcreFt   = A[A$CategoryD=='NW1MWC',]$KAcreFt   - A[A$CategoryD=='MW18A',]$KAcreFt
   A[A$CategoryD=='NW1URB',]$KAcreFt   = A[A$CategoryD=='NW1URB',]$KAcreFt   - A[A$CategoryD=='URB19A',]$KAcreFt
   A[A$CategoryD=='NW1URBC',]$KAcreFt  = A[A$CategoryD=='NW1URBC',]$KAcreFt  - A[A$CategoryD=='URB28A',]$KAcreFt
   A[A$CategoryD=='NW1WSR',]$KAcreFt   = A[A$CategoryD=='NW1WSR',]$KAcreFt   - A[A$CategoryD=='WSR4A',]$KAcreFt
#
# NW2
#
   A[A$CategoryD=='NW2AG',]$KAcreFt    = A[A$CategoryD=='NW2AG',]$KAcreFt  -
                                         A[A$CategoryD=='AG11A',]$KAcreFt
   A[A$CategoryD=='NW2AGC',]$KAcreFt   = sum(A[A$CategoryD=='NW2AGC',]$KAcreFt) -
                                         sum(A[A$CategoryD=='AG20A',]$KAcreFt)

   A[A$CategoryD=='NW2IFR',]$KAcreFt   = A[A$CategoryD=='NW2IFR',]$KAcreFt   - A[A$CategoryD=='IFR4A',]$KAcreFt
   A[A$CategoryD=='NW2MW',]$KAcreFt    = A[A$CategoryD=='NW2MW', ]$KAcreFt   - A[A$CategoryD=='MW9A',]$KAcreFt
   A[A$CategoryD=='NW2MWC',]$KAcreFt   = A[A$CategoryD=='NW2MWC',]$KAcreFt   - A[A$CategoryD=='MW18A',]$KAcreFt
   A[A$CategoryD=='NW2URB',]$KAcreFt   = A[A$CategoryD=='NW2URB',]$KAcreFt   - A[A$CategoryD=='URB19A',]$KAcreFt
```

```
    A[A$CategoryD=='NW2URBC',]$KAcreFt = A[A$CategoryD=='NW2URBC',]$KAcreFt - A[A$CategoryD=='URB28A',]$KAcreFt
    A[A$CategoryD=='NW2WSR',]$KAcreFt  = A[A$CategoryD=='NW2WSR',]$KAcreFt  - A[A$CategoryD=='WSR4A',]$KAcreFt

# ================================================================
# End of PA loop
# ================================================================
if (i==1){
        MASTER = A
        }
        else
        {
        MASTER = rbind(MASTER, A)
        print(paste('-End: YYYY=', YYYY, '/ i=', i, sep=''))
        }
}
}
# ============================================================
# ST adjustments below
# ============================================================
if (level=='ST'){
         ITEM='CA'
          i = i + 1
     A = subset(df, Year==YYYY)
     if (dim(A)[1]==0){ print(paste('Empty df: YYYY=', YYYY, '/ ITEM =', ITEM, sep='')); next }
     print(paste('Start: YYYY=', YYYY, '/ HR=', ITEM, sep=''))

A[A$CategoryD=='NW1AG',]$KAcreFt = A[A$CategoryD=='NW1AG',]$KAcreFt    - sum(A[A$CategoryD=='AG11A',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='AG11B',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='AG11C',]$KAcreFt)
A[A$CategoryD=='NW1AGC',]$KAcreFt= A[A$CategoryD=='NW1AGC',]$KAcreFt   - sum(A[A$CategoryD=='AG20A',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='AG20B',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='AG20C',]$KAcreFt)
A[A$CategoryD=='NW1IFR',]$KAcreFt = A[A$CategoryD=='NW1IFR',]$KAcreFt - sum(A[A$CategoryD=='IFR4A',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='IFR4B',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='IFR4C',]$KAcreFt)
A[A$CategoryD=='NW1MW',]$KAcreFt = A[A$CategoryD=='NW1MW',]$KAcreFt    - sum(A[A$CategoryD=='MW9A', ]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='MW9B', ]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='MW9C', ]$KAcreFt)
A[A$CategoryD=='NW1MWC',]$KAcreFt = A[A$CategoryD=='NW1MWC',]$KAcreFt - sum(A[A$CategoryD=='MW18A',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='MW18B',]$KAcreFt) -
                                                                                        sum(A[A$CategoryD=='MW18C',]$KAcreFt)
A[A$CategoryD=='NW1URB',]$KAcreFt = A[A$CategoryD=='NW1URB',]$KAcreFt - sum(A[A$CategoryD=='URB19A',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='URB19B',]$KAcreFt) -
                                                                                            sum(A[A$CategoryD=='URB19C',]$KAcreFt)
A[A$CategoryD=='NW1URBC',]$KAcreFt = A[A$CategoryD=='NW1URBC',]$KAcreFt - sum(A[A$CategoryD=='URB28A',]$KAcreFt) -
                                                                          sum(A[A$CategoryD=='URB28B',]$KAcreFt) -
                                                                                            sum(A[A$CategoryD=='URB28C',]$KAcreFt)
A[A$CategoryD=='NW1WSR',]$KAcreFt = A[A$CategoryD=='NW1WSR',]$KAcreFt - sum(A[A$CategoryD=='WSR4A',]$KAcreFt) -
                                                                                            sum(A[A$CategoryD=='WSR4B',]$KAcreFt) -
```

```
                                                                           sum(A[A$CategoryD=='WSR4C',]$KAcreFt)
A[A$CategoryD=='NW2AG',]$KAcreFt = A[A$CategoryD=='NW2AG',]$KAcreFt    - sum(A[A$CategoryD=='AG11A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='AG11B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='AG11C',]$KAcreFt)
A[A$CategoryD=='NW2AGC',]$KAcreFt = A[A$CategoryD=='NW2AGC',]$KAcreFt - sum(A[A$CategoryD=='AG20A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='AG20B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='AG20C',]$KAcreFt)
A[A$CategoryD=='NW2IFR',]$KAcreFt = A[A$CategoryD=='NW2IFR',]$KAcreFt - sum(A[A$CategoryD=='IFR4A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='IFR4B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='IFR4C',]$KAcreFt)
A[A$CategoryD=='NW2MW',]$KAcreFt = A[A$CategoryD=='NW2MW',]$KAcreFt    - sum(A[A$CategoryD=='MW9A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='MW9B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='MW9C',]$KAcreFt)
A[A$CategoryD=='NW2MWC',]$KAcreFt = A[A$CategoryD=='NW2MWC',]$KAcreFt - sum(A[A$CategoryD=='MW18A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='MW18B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='MW18C',]$KAcreFt)
A[A$CategoryD=='NW2URB',]$KAcreFt = A[A$CategoryD=='NW2URB',]$KAcreFt - sum(A[A$CategoryD=='URB19A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='URB19B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='URB19C',]$KAcreFt)
A[A$CategoryD=='NW2URBC',]$KAcreFt = A[A$CategoryD=='NW2URBC',]$KAcreFt - sum(A[A$CategoryD=='URB28A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='URB28B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='URB28C',]$KAcreFt)
A[A$CategoryD=='NW2WSR',]$KAcreFt = A[A$CategoryD=='NW2WSR',]$KAcreFt - sum(A[A$CategoryD=='WSR4A',]$KAcreFt) -
                                                                         sum(A[A$CategoryD=='WSR4B',]$KAcreFt) -
                                                                           sum(A[A$CategoryD=='WSR4C',]$KAcreFt)
# ============================================================
# End of ST loop
# ============================================================
if (i==1){
        MASTER = A
        print(paste('-Init: YYYY=', YYYY, '/ i=', i, sep=''))
        }
        else
        {
        MASTER = rbind(MASTER, A)
        print(paste('-End: YYYY=', YYYY, '/ i=', i, sep=''))
        }
}
}
return(MASTER)
}
```

Listing 4.3: Water supply equations.

```
f_Equations_WaterSupply = function(df){
# ==========================================================================
# Water Supply Equations: Translation of TH formulas at DAUCO-level
# ==========================================================================
EQ = data.frame(ncols=1)
#
# ==========================================================================
# Total Developed Supply
# ==========================================================================
EQ$TDS = sum(df[df$CategoryD=='SPL1A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL1B', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL1C', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL1D', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL1E', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL1F', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2A1', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2A2', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2A3', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2A4', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2A5', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2A6', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2B1', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2B2', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2B3', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2B4', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2B5', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2B6', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2C1', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2C2', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2C3', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2C4', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2C5', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2C6', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2D1', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2D2', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL2D3', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL3A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL3B', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL3C', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL3D', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL3E', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL3F', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL4A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL4B', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL4C', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL4D', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL4E', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL4F', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL5A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL5B', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL5C', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL5D', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL5E', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL5F', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL6A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL6B', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL6C', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL6D', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL6E', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL6F', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL10A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL10B', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL10C', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL10D', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL11A', ]$KAcreFt) +
         sum(df[df$CategoryD=='SPL11B', ]$KAcreFt) +
```

```
          sum(df[df$CategoryD=='SPL11C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL11D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL11E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL11F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL12A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL12B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL12C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL12D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL12E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL12F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL13A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL13B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL13C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL13D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL13E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL13F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL14A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL14B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL14C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL14D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL14E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL14F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL15A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL15B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL15C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL15D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL15E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL15F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL16A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL16B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL16C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL16D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL16E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL16F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL17A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL17B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL17C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL17D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL17E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL17F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL18A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL18B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL18C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL18D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL18E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL18F', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL19A', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL19B', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL19C', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL19D', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL19E', ]$KAcreFt) +
          sum(df[df$CategoryD=='SPL19F', ]$KAcreFt)
# ===============================================================
# Total Return Flow and Reuse
# ===============================================================
EQ$TRFR = sum(df[df$CategoryD=='AG8',    ]$KAcreFt) +
          sum(df[df$CategoryD=='AG21',   ]$KAcreFt) +
          sum(df[df$CategoryD=='URB15A', ]$KAcreFt) +
          sum(df[df$CategoryD=='URB15B', ]$KAcreFt) +
          sum(df[df$CategoryD=='URB15C', ]$KAcreFt) +
          sum(df[df$CategoryD=='URB29',  ]$KAcreFt) +
          sum(df[df$CategoryD=='MW6',    ]$KAcreFt) +
          sum(df[df$CategoryD=='MW19',   ]$KAcreFt) +
          sum(df[df$CategoryD=='IFR2',   ]$KAcreFt) +
          sum(df[df$CategoryD=='WSR2',   ]$KAcreFt)
#
# ===============================================================
# Total Supply and Reuse
```

```
# ==================================================================
EQ$TSR =   EQ$TDS + EQ$TRFR
#
# ==================================================================
# Total Reuse of Deep Percolation
# ==================================================================
EQ$TRDP = sum(df[df$CategoryD=='AG5',   ]$KAcreFt) +
          sum(df[df$CategoryD=='AG7',   ]$KAcreFt) +
          sum(df[df$CategoryD=='AG22',  ]$KAcreFt) +
          sum(df[df$CategoryD=='URB12', ]$KAcreFt) +
          sum(df[df$CategoryD=='URB14', ]$KAcreFt) +
          sum(df[df$CategoryD=='URB30', ]$KAcreFt) +
          sum(df[df$CategoryD=='MW3',   ]$KAcreFt) +
          sum(df[df$CategoryD=='MW5',   ]$KAcreFt) +
          sum(df[df$CategoryD=='MW20',  ]$KAcreFt)
#
# ==================================================================
# Total Net Supply
# ==================================================================
EQ$TNS = EQ$TDS - EQ$TRDP
#
# ==================================================================
# Total Reuse
# ==================================================================
EQ$TR    = EQ$TRFR + EQ$TRDP
#
    return(EQ)
}
```

Listing 4.4: Water supply adjustments.

```
f_Adjustments_WaterSupply = function(df, level){
# ==========================================================================
# Water Supply Adjustments: Translation of TH formulas at DAUCO-level
# ==========================================================================
#
YR_LIST =  unique(df$Year)
    i = 0
    for (YYYY  in YR_LIST){
# ==========================================================================
# HR-level
# ==========================================================================
if (level=='HR'){
    LIST = unique(df$HR)
    for (ITEM in LIST){
        i = i + 1
        A = subset(df, HR==ITEM & Year==YYYY)
        if (dim(A)[1]==0){ print(paste('Empty_df:_YYYY=', YYYY, '/_HR_CODE=', ITEM, sep='')); next }
        print(paste('Start:_YYYY=', YYYY, '/_HR=', ITEM, sep=''))
# TDS
A[A$CategoryD=='SPL23', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt -
                                    sum(A[A$CategoryD=='SPL2A1', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A2', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A3', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A4', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A5', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A6', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B1', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B2', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B3', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B4', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B5', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B6', ]$KAcreFt)
# TRFR
A[A$CategoryD=='SPL24', ]$KAcreFt = A[A$CategoryD=='SPL24',      ]$KAcreFt  +
                                    sum(A[A$CategoryD=='AG11A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG20A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB19A',]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB28A',]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW9A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW18A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='IFR4A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='WSR4A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG11B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG20B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB19B',]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB28B',]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW9B',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW18B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='IFR4B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='WSR4B', ]$KAcreFt)
#
# ================================================================
# Re-calculate the dependent values as per f_Equations_WaterSupply
# ================================================================
#EQ$TSR =  EQ$TDS + EQ$TRFR
A[A$CategoryD=='SPL25', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt +
                                    A[A$CategoryD=='SPL24', ]$KAcreFt
#EQ$TNS = EQ$TDS - EQ$TRDP
A[A$CategoryD=='SPL27', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt -
                                    A[A$CategoryD=='SPL26', ]$KAcreFt
#EQ$TR   = EQ$TRFR + EQ$TRDP
A[A$CategoryD=='SPL28', ]$KAcreFt = A[A$CategoryD=='SPL24', ]$KAcreFt +
                                    A[A$CategoryD=='SPL26', ]$KAcreFt
    if (i==1){
       MASTER = A
       }
```

```
      else
         {
         MASTER = rbind(MASTER, A)
          print(paste('-End: YYYY=', YYYY, '/ i=', i, sep=''))
        }
     }
}
# ==========================================================================
# PA-level
# ==========================================================================
if (level=='PA'){
     LIST = unique(df$PA)
     for (ITEM in LIST){
          i = i + 1
          A = subset(df, PA==ITEM & Year==YYYY)
          if (dim(A)[1]==0){ print(paste('Empty df: YYYY=', YYYY, '/ ITEM=', ITEM, sep='')); next }
          print(paste('Start: YYYY=', YYYY, '/ PA =', ITEM, sep=''))
#
A[A$CategoryD=='SPL23', ]$KAcreFt = A[A$CategoryD=='SPL23',      ]$KAcreFt -
                                    sum(A[A$CategoryD=='SPL2A1', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A2', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A3', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A4', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A5', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A6', ]$KAcreFt)

A[A$CategoryD=='SPL24', ]$KAcreFt = A[A$CategoryD=='SPL24',      ]$KAcreFt  +
                                    sum(A[A$CategoryD=='AG11A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG20A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB19A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB28A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW9A',   ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW18A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='IFR4A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='WSR4A',  ]$KAcreFt)

#
# ==================================================================
# Re-calculate the dependent values as per f_Equations_WaterSupply
# ==================================================================
#EQ$TSR =  EQ$TDS + EQ$TRFR
A[A$CategoryD=='SPL25', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt +
                                    A[A$CategoryD=='SPL24', ]$KAcreFt
#EQ$TNS = EQ$TDS - EQ$TRDP
A[A$CategoryD=='SPL27', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt -
                                    A[A$CategoryD=='SPL26', ]$KAcreFt
#EQ$TR   = EQ$TRFR + EQ$TRDP
A[A$CategoryD=='SPL28', ]$KAcreFt = A[A$CategoryD=='SPL24', ]$KAcreFt +
                                    A[A$CategoryD=='SPL26', ]$KAcreFt

if (i==1){
        MASTER = A
        }
      else
         {
         MASTER = rbind(MASTER, A)
          print(paste('-End: YYYY=', YYYY, '/ i=', i, sep=''))
         }
}
}
# ==========================================================================
# ST-level
# ==========================================================================
if (level=='ST'){
     ITEM='CA'
     i = i + 1
     A = subset(df, Year==YYYY)
     if (dim(A)[1]==0){ print(paste('Empty df: YYYY=', YYYY, '/ ITEM =', ITEM, sep='')); next }
```

```
    print(paste('Start: YYYY=', YYYY, '/ HR=', ITEM, sep=''))
#
A[A$CategoryD=='SPL23', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt -
                                    sum(A[A$CategoryD=='SPL2A1', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A2', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A3', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A4', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A5', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2A6', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B1', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B2', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B3', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B4', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B5', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2B6', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2C1', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2C2', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2C3', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2C4', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2C5', ]$KAcreFt) -
                                    sum(A[A$CategoryD=='SPL2C6', ]$KAcreFt)
A[A$CategoryD=='SPL24', ]$KAcreFt = A[A$CategoryD=='SPL24', ]$KAcreFt +
                                    sum(A[A$CategoryD=='AG11A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG20A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB19A',]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB28A',]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW9A',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW18A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='IFR4A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='WSR4A', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG11B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG20B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB19B',]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB28B',]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW9B',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW18B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='IFR4B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='WSR4B', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG11C', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='AG20C', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB19C',]$KAcreFt) +
                                    sum(A[A$CategoryD=='URB28C',]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW9C',  ]$KAcreFt) +
                                    sum(A[A$CategoryD=='MW18C', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='IFR4C', ]$KAcreFt) +
                                    sum(A[A$CategoryD=='WSR4C', ]$KAcreFt)
                                   #
#
# ================================================================
# Re-calculate the dependent values as per f_Equations_WaterSupply
# ================================================================
#EQ$TSR =  EQ$TDS + EQ$TRFR
A[A$CategoryD=='SPL25', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt +
                                    A[A$CategoryD=='SPL24', ]$KAcreFt
#EQ$TNS = EQ$TDS - EQ$TRDP
A[A$CategoryD=='SPL27', ]$KAcreFt = A[A$CategoryD=='SPL23', ]$KAcreFt -
                                    A[A$CategoryD=='SPL26', ]$KAcreFt
#EQ$TR    = EQ$TRFR + EQ$TRDP
A[A$CategoryD=='SPL28', ]$KAcreFt = A[A$CategoryD=='SPL24', ]$KAcreFt +
                                    A[A$CategoryD=='SPL26', ]$KAcreFt

 if  (i==1){
        MASTER = A
        print(paste('-Init: YYYY=', YYYY, '/ i=', i, sep=''))
        }
    else
        {
        MASTER = rbind(MASTER, A)
```

```
            print(paste('-End: YYYY=', YYYY, '/ i=', i, sep=''))
            }
}
}
return(MASTER)
}
```

Table 4.2: Input-Table-CV-Parameterization.tex

Table	CategoryD	CategoryB	CategoryA	CategoryC
1	AG1	1	Agriculture	Applied Water
2	AG10A	10a	Agriculture	Return Flow to Salt Sink
3	AG10B	10b	Agriculture	Return Flow for Delta Outflow
4	AG11A	11a	Agriculture	Return Flow to Developed Supply (Other DAUCO within PA)
5	AG11B	11b	Agriculture	Return Flow to Developed Supply (Other PA)
6	AG11C	11c	Agriculture	Return Flow to Developed Supply (Other Region)
7	AG11D	11d	Agriculture	Return Flow to Carryover Storage for Next Water Year within DAU
8	AG12	12	Agriculture	Return Flows Evaporation and Evapotranspiration
13	AG17	17	Agriculture	Conveyance Evaporation and ETAW
14	AG18A	18a	Agriculture	Conveyance Return Flow to Oregon
15	AG18B	18b	Agriculture	Conveyance Return Flow to Nevada
16	AG18C	18c	Agriculture	Conveyance Return Flow to Mexico
17	AG18D	18d	Agriculture	Conveyance Deep Percolation to Oregon
18	AG18E	18e	Agriculture	Conveyance Deep Percolation to Nevada
19	AG18F	18f	Agriculture	Conveyance Deep Percolation to Mexico
20	AG19A	19a	Agriculture	Conveyance Return Flows to Salt Sink
21	AG19B	19b	Agriculture	Conveyance Return Flow for Delta Outflow
22	AG2	2	Agriculture	Applied Water - Groundwater Recharge
23	AG20A	20a	Agriculture	Conveyance Return Flow to Developed Supply (Other DAUCO within PA)
24	AG20B	20b	Agriculture	Conveyance Return Flow to Developed Supply (Other PA)
25	AG20C	20c	Agriculture	Conveyance Return Flow to Developed Supply (Other Region)
26	AG21	21	Agriculture	Conveyance Seepage
27	AG22	22	Agriculture	Conveyance Deep Percolation
28	AG23	23	Agriculture	Conveyance Deep Percolation to Salt Sink
33	AG3	3	Agriculture	Evapotranspiration of Applied Water
34	AG4	4	Agriculture	Evaporation and Evapotranspiration of Groundwater Recharge
35	AG5	5	Agriculture	Deep Percolation of Applied Water
36	AG6	6	Agriculture	Deep Percolation of Applied Water to Salt Sink
37	AG7	7	Agriculture	Deep Percolation of Groundwater Recharge
38	AG8	8	Agriculture	Reuse of Return Flows within DAUCO
39	AG9A	9a	Agriculture	Return Flow to Oregon
40	AG9B	9b	Agriculture	Return Flow to Nevada
41	AG9C	9c	Agriculture	Return Flow to Mexico
42	AG9D	9d	Agriculture	Deep Percolation to Oregon
43	AG9E	9e	Agriculture	Deep Percolation to Nevada
44	AG9F	9f	Agriculture	Deep Percolation to Mexico
9	AWUAG	13	Agriculture	Applied Water Use
29	AWUAGC	24	Agriculture	Conveyance Applied Water Use
53	AWUIFR	5	Instream Flow Requirements	Applied Water Use
59	AWUMW	11	Managed Wetlands	Applied Water Use
79	AWUMWC	22	Managed Wetlands	Conveyance Applied Water Use
101	AWURDO	3	Required Delta Outflow	Applied Water Use
129	AWUURB	21	Urban	Applied Water Use
149	AWUURBC	32	Urban	Conveyance Applied Water Use
282	AWUWSR	5	Wild and Scenic River	Applied Water Use
12	DEPAG	16	Agriculture	Depletion
32	DEPAGC	27	Agriculture	Conveyance Depletion
56	DEPIFR	8	Instream Flow Requirements	Depletion
62	DEPMW	14	Managed Wetlands	Depletion
82	DEPMWC	25	Managed Wetlands	Conveyance Depletion
104	DEPRDO	6	Required Delta Outflow	Depletion
132	DEPURB	24	Urban	Depletion
152	DEPURBC	35	Urban	Conveyance Depletion
285	DEPWSR	8	Wild and Scenic River	Depletion
45	IFR1	1	Instream Flow Requirements	Applied Water
46	IFR2	2	Instream Flow Requirements	Reuse of Return Flows within DAUCO
47	IFR3A	3a	Instream Flow Requirements	Return Flow to Salt Sink
48	IFR3B	3b	Instream Flow Requirements	Return Flow to Oregon - Mexico - Nevada
49	IFR3C	3c	Instream Flow Requirements	Return Flow for Delta Outflow
50	IFR4A	4a	Instream Flow Requirements	Return Flow to Developed Supply (Other DAUCO within PA)
51	IFR4B	4b	Instream Flow Requirements	Return Flow to Developed Supply (Other PA)
52	IFR4C	4c	Instream Flow Requirements	Return Flow to Developed Supply (Other Region)
57	MW1	1	Managed Wetlands	Applied Water
58	MW10	10	Managed Wetlands	Return Flows Evaporation and Evapotranspiration

63	MW15	15	Managed Wetlands	Conveyance Evaporation and ETAW
64	MW16A	16a	Managed Wetlands	Conveyance Return Flow to Oregon
65	MW16B	16b	Managed Wetlands	Conveyance Return Flow to Nevada
66	MW16C	16c	Managed Wetlands	Conveyance Return Flow to Mexico
67	MW16D	16d	Managed Wetlands	Conveyance Deep Percolation to Oregon
68	MW16E	16e	Managed Wetlands	Conveyance Deep Percolation to Nevada
69	MW16F	16f	Managed Wetlands	Conveyance Deep Percolation to Mexico
70	MW17A	17a	Managed Wetlands	Conveyance Return Flows to Salt Sink
71	MW17B	17b	Managed Wetlands	Conveyance Return Flow for Delta Outflow
72	MW18A	18a	Managed Wetlands	Conveyance Return Flow to Developed Supply (Other DAUCO within PA)
73	MW18B	18b	Managed Wetlands	Conveyance Return Flow to Developed Supply (Other PA)
74	MW18C	18c	Managed Wetlands	Conveyance Return Flow to Developed Supply (Other Region)
75	MW19	19	Managed Wetlands	Conveyance Seepage
76	MW2	2	Managed Wetlands	Evapotranspiration of Applied Water
77	MW20	20	Managed Wetlands	Conveyance Deep Percolation
78	MW21	21	Managed Wetlands	Conveyance Deep Percolation to Salt Sink
83	MW3	3	Managed Wetlands	Deep Percolation of Applied Water
84	MW4	4	Managed Wetlands	Deep Percolation of Applied Water to Salt Sink
85	MW5	5	Managed Wetlands	Deep Percolation of Groundwater Recharge
86	MW6	6	Managed Wetlands	Reuse of Return Flows within DAUCO
87	MW7A	7a	Managed Wetlands	Return Flow to Oregon
88	MW7B	7b	Managed Wetlands	Return Flow to Nevada
89	MW7C	7c	Managed Wetlands	Return Flow to Mexico
90	MW7D	7d	Managed Wetlands	Deep Percolation to Oregon
91	MW7E	7e	Managed Wetlands	Deep Percolation to Nevada
92	MW7F	7f	Managed Wetlands	Deep Percolation to Mexico
93	MW8A	8a	Managed Wetlands	Return Flow to Salt Sink
94	MW8B	8b	Managed Wetlands	Return Flow for Delta Outflow
95	MW9A	9a	Managed Wetlands	Return Flow to Developed Supply (Other DAUCO within PA)
96	MW9B	9b	Managed Wetlands	Return Flow to Developed Supply (Other PA)
97	MW9C	9c	Managed Wetlands	Return Flow to Developed Supply (Other Region)
98	MW9D	9d	Managed Wetlands	Return Flow to Carryover Storage for Next Water Year within DAU
10	NW1AG	14	Agriculture	Net Water Use (Applied Water - Reuse)
30	NW1AGC	25	Agriculture	Conveyance Net Water Use (Applied Water - Reuse)
54	NW1IFR	6	Instream Flow Requirements	Net Water Use (Applied Water - Reuse)
60	NW1MW	12	Managed Wetlands	Net Water Use (Applied Water - Reuse)
80	NW1MWC	23	Managed Wetlands	Conveyance Net Water Use (Applied Water - Reuse)
102	NW1RDO	4	Required Delta Outflow	Net Water Use (Applied Water - Reuse)
130	NW1URB	22	Urban	Net Water Use (Applied Water - Reuse)
150	NW1URBC	33	Urban	Conveyance Net Water Use (Applied Water - Reuse)
283	NW1WSR	6	Wild and Scenic River	Net Water Use (Applied Water - Reuse)
11	NW2AG	15	Agriculture	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
31	NW2AGC	26	Agriculture	Conveyance Net Water Use (ETAW + Flow/Salt Sink + Outflow)
55	NW2IFR	7	Instream Flow Requirements	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
61	NW2MW	13	Managed Wetlands	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
81	NW2MWC	24	Managed Wetlands	Conveyance Net Water Use (ETAW + Flow/Salt Sink + Outflow)
103	NW2RDO	5	Required Delta Outflow	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
131	NW2URB	23	Urban	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
151	NW2URBC	34	Urban	Conveyance Net Water Use (ETAW + Flow/Salt Sink + Outflow)
284	NW2WSR	7	Wild and Scenic River	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
99	RDO1	1	Required Delta Outflow	Applied Water
100	RDO2	2	Required Delta Outflow	Return Flow for Delta Outflow
159	SPL10A	10a	Water Supplies	Desalination - Urban
160	SPL10B	10b	Water Supplies	Desalination - Instream Flow Requirements
161	SPL10C	10c	Water Supplies	Desalination - Wild and Scenic Flows
162	SPL10D	10d	Water Supplies	Desalination - Required Delta Outflow
163	SPL11A	11a	Water Supplies	Colorado River Deliveries - Agriculture
164	SPL11B	11b	Water Supplies	Colorado River Deliveries - Managed Wetlands
165	SPL11C	11c	Water Supplies	Colorado River Deliveries - Urban
166	SPL11D	11d	Water Supplies	Colorado River Deliveries - Instream Flow Requirements
167	SPL11E	11e	Water Supplies	Colorado River Deliveries - Wild and Scenic Flows
168	SPL11F	11f	Water Supplies	Colorado River Deliveries - Required Delta Outflow
169	SPL12A	12a	Water Supplies	State Water Project Deliveries - Agriculture
170	SPL12B	12b	Water Supplies	State Water Project Deliveries - Managed Wetlands
171	SPL12C	12c	Water Supplies	State Water Project Deliveries - Urban
172	SPL12D	12d	Water Supplies	State Water Project Deliveries - Instream Flow Requirements
173	SPL12E	12e	Water Supplies	State Water Project Deliveries - Wild and Scenic Flows
174	SPL12F	12f	Water Supplies	State Water Project Deliveries - Required Delta Outflow

175	SPL13A	13a	Water Supplies	Central Valley Project - Base Deliveries - Agriculture
176	SPL13B	13b	Water Supplies	Central Valley Project - Base Deliveries - Managed Wetlands
177	SPL13C	13c	Water Supplies	Central Valley Project - Base Deliveries - Urban
178	SPL13D	13d	Water Supplies	Central Valley Project - Base Deliveries - Instream Flow Requirements
179	SPL13E	13e	Water Supplies	Central Valley Project - Base Deliveries - Wild and Scenic Flows
180	SPL13F	13f	Water Supplies	Central Valley Project - Base Deliveries - Required Delta Outflow
181	SPL14A	14a	Water Supplies	Central Valley Project - Project Deliveries - Agriculture
182	SPL14B	14b	Water Supplies	Central Valley Project - Project Deliveries - Managed Wetlands
183	SPL14C	14c	Water Supplies	Central Valley Project - Project Deliveries - Urban
184	SPL14D	14d	Water Supplies	Central Valley Project - Project Deliveries - Instream Flow Requirements
185	SPL14E	14e	Water Supplies	Central Valley Project - Project Deliveries - Wild and Scenic Flows
186	SPL14F	14f	Water Supplies	Central Valley Project - Project Deliveries - Required Delta Outflow
187	SPL15A	15a	Water Supplies	Other Federal Deliveries - Agriculture
188	SPL15B	15b	Water Supplies	Other Federal Deliveries - Managed Wetlands
189	SPL15C	15c	Water Supplies	Other Federal Deliveries - Urban
190	SPL15D	15d	Water Supplies	Other Federal Deliveries - Instream Flow Requirements
191	SPL15E	15e	Water Supplies	Other Federal Deliveries - Wild and Scenic Flows
192	SPL15F	15f	Water Supplies	Other Federal Deliveries - Required Delta Outflow
193	SPL16A	16a	Water Supplies	Ocean Desalination - Agriculture
194	SPL16B	16b	Water Supplies	Ocean Desalination - Managed Wetlands
195	SPL16C	16c	Water Supplies	Ocean Desalination - Urban
196	SPL16D	16d	Water Supplies	Ocean Desalination - Instream Flow Requirements
197	SPL16E	16e	Water Supplies	Ocean Desalination - Wild and Scenic Flows
198	SPL16F	16f	Water Supplies	Ocean Desalination - Required Delta Outflow
199	SPL17A	17a	Water Supplies	Water from Refineries - Agriculture
200	SPL17B	17b	Water Supplies	Water from Refineries - Managed Wetlands
201	SPL17C	17c	Water Supplies	Water from Refineries - Urban
202	SPL17D	17d	Water Supplies	Water from Refineries - Instream Flow Requirements
203	SPL17E	17e	Water Supplies	Water from Refineries - Wild and Scenic Flows
204	SPL17F	17f	Water Supplies	Water from Refineries - Required Delta Outflow
205	SPL18A	18a	Water Supplies	Water Transfers - Regional - Agriculture
206	SPL18B	18b	Water Supplies	Water Transfers - Regional - Managed Wetlands
207	SPL18C	18c	Water Supplies	Water Transfers - Regional - Urban
208	SPL18D	18d	Water Supplies	Water Transfers - Regional - Instream Flow Requirements
209	SPL18E	18e	Water Supplies	Water Transfers - Regional - Wild and Scenic Flows
210	SPL18F	18f	Water Supplies	Water Transfers - Regional - Required Delta Outflow
211	SPL19A	19a	Water Supplies	Inter-basin Water Transfers - Agriculture
212	SPL19B	19b	Water Supplies	Inter-basin Water Transfers - Managed Wetlands
213	SPL19C	19c	Water Supplies	Inter-basin Water Transfers - Urban
214	SPL19D	19d	Water Supplies	Inter-basin Water Transfers - Instream Flow Requirements
215	SPL19E	19e	Water Supplies	Inter-basin Water Transfers - Wild and Scenic Flows
216	SPL19F	19f	Water Supplies	Inter-basin Water Transfers - Required Delta Outflow
217	SPL1A	1a	Water Supplies	Local Supplies - Agriculture
218	SPL1B	1b	Water Supplies	Local Supplies - Managed Wetlands
219	SPL1C	1c	Water Supplies	Local Supplies - Urban
220	SPL1D	1d	Water Supplies	Local Supplies - Instream Flow Requirements
221	SPL1E	1e	Water Supplies	Local Supplies - Wild and Scenic Flows
222	SPL1F	1f	Water Supplies	Local Supplies - Required Delta Outflow
223	SPL23	23	Water Supplies	Total Developed Supply (TDS)
224	SPL24	24	Water Supplies	Total Return Flow and Reuse (TRFR)
225	SPL25	25	Water Supplies	Total Supply and Retuse (TSR)
226	SPL26	26	Water Supplies	Total Reuse of Deep Percolation (TRDP)
227	SPL27	27	Water Supplies	Total Net Supply (TNS)
228	SPL28	28	Water Supplies	Total Reuse (TR)
229	SPL2A1	2a1	Water Supplies	Return Flow from Other DAUCO within PA - Agriculture
230	SPL2A2	2a2	Water Supplies	Return Flow from Other DAUCO within PA - Managed Wetlands
231	SPL2A3	2a3	Water Supplies	Return Flow from Other DAUCO within PA - Urban
232	SPL2A4	2a4	Water Supplies	Return Flow from Other DAUCO within PA - Instream Flow Requirements
233	SPL2A5	2a5	Water Supplies	Return Flow from Other DAUCO within PA - Wild and Scenic Flows
234	SPL2A6	2a6	Water Supplies	Return Flow from Other DAUCO within PA - Required Delta Outflow
235	SPL2B1	2b1	Water Supplies	Return Flow from Other PA - Agriculture
236	SPL2B2	2b2	Water Supplies	Return Flow from Other PA - Managed Wetlands
237	SPL2B3	2b3	Water Supplies	Return Flow from Other PA - Urban
238	SPL2B4	2b4	Water Supplies	Return Flow from Other PA - Instream Flow Requirements
239	SPL2B5	2b5	Water Supplies	Return Flow from Other PA - Wild and Scenic Flows
240	SPL2B6	2b6	Water Supplies	Return Flow from Other PA - Required Delta Outflow
241	SPL2C1	2c1	Water Supplies	Return Flow from Other Region - Agriculture
242	SPL2C2	2c2	Water Supplies	Return Flow from Other Region - Managed Wetlands

243	SPL2C3	2c3	Water Supplies	Return Flow from Other Region - Urban
244	SPL2C4	2c4	Water Supplies	Return Flow from Other Region - Instream Flow Requirements
245	SPL2C5	2c5	Water Supplies	Return Flow from Other Region - Wild and Scenic Flows
246	SPL2C6	2c6	Water Supplies	Return Flow from Other Region - Required Delta Outflow
247	SPL2D1	2d1	Water Supplies	Return Flow to Carryover Storage within DAU from Previous WY - Agriculture
248	SPL2D2	2d2	Water Supplies	Return Flow to Carryover Storage within DAU from Previous WY - Managed Wetlands
249	SPL2D3	2d3	Water Supplies	Return Flow to Carryover Storage within DAU from Previous WY - Urban
250	SPL3A	3a	Water Supplies	Local Imports - Agriculture
251	SPL3B	3b	Water Supplies	Local Imports - Managed Wetlands
252	SPL3C	3c	Water Supplies	Local Imports - Urban
253	SPL3D	3d	Water Supplies	Local Imports - Instream Flow Requirements
254	SPL3E	3e	Water Supplies	Local Imports - Wild and Scenic Flows
255	SPL3F	3f	Water Supplies	Local Imports - Required Delta Outflow
256	SPL4A	4a	Water Supplies	Groundwater Extraction - Unadjudicated - Agriculture
257	SPL4B	4b	Water Supplies	Groundwater Extraction - Unadjudicated - Managed Wetlands
258	SPL4C	4c	Water Supplies	Groundwater Extraction - Unadjudicated - Urban
259	SPL4D	4d	Water Supplies	Groundwater Extraction - Unadjudicated - Instream Flow Requirements
260	SPL4E	4e	Water Supplies	Groundwater Extraction - Unadjudicated - Wild and Scenic Flows
261	SPL4F	4f	Water Supplies	Groundwater Extraction - Unadjudicated - Required Delta Outflow
262	SPL5A	5a	Water Supplies	Groundwater Extraction - Adjudicated - Agriculture
263	SPL5B	5b	Water Supplies	Groundwater Extraction - Adjudicated - Managed Wetlands
264	SPL5C	5c	Water Supplies	Groundwater Extraction - Adjudicated - Urban
265	SPL5D	5d	Water Supplies	Groundwater Extraction - Adjudicated - Instream Flow Requirements
266	SPL5E	5e	Water Supplies	Groundwater Extraction - Adjudicated - Wild and Scenic Flows
267	SPL5F	5f	Water Supplies	Groundwater Extraction - Adjudicated - Required Delta Outflow
268	SPL6A	6a	Water Supplies	Groundwater Extraction - Banked - Agriculture
269	SPL6B	6b	Water Supplies	Groundwater Extraction - Banked - Managed Wetlands
270	SPL6C	6c	Water Supplies	Groundwater Extraction - Banked - Urban
271	SPL6D	6d	Water Supplies	Groundwater Extraction - Banked - Instream Flow Requirements
272	SPL6E	6e	Water Supplies	Groundwater Extraction - Banked - Wild and Scenic Flows
273	SPL6F	6f	Water Supplies	Groundwater Extraction - Banked - Required Delta Outflow
105	URB1	1	Urban	Applied Water - Residential - Single Family Interior
106	URB10	10	Urban	Evapotranspiration of Applied Water
107	URB11	11	Urban	Evaporation and Evapotranspiration of Groundwater Recharge
108	URB12	12	Urban	Deep Percolation of Applied Water
109	URB13	13	Urban	Deep Percolation of Applied Water to Salt Sink
110	URB14	14	Urban	Deep Percolation of Groundwater Recharge
111	URB15A	15a	Urban	Reuse of Return Flows within DAUCO
112	URB15B	15b	Urban	Urban - Wastewater Recycling
113	URB15C	15c	Urban	Urban - Desalination
114	URB16	16	Urban	Evaporation and Evapotranspiration of Wastewater
115	URB17A	17a	Urban	Return Flow to Oregon
116	URB17B	17b	Urban	Return Flow to Nevada
117	URB17C	17c	Urban	Return Flow to Mexico
118	URB17D	17d	Urban	Deep Percolation to Oregon
119	URB17E	17e	Urban	Deep Percolation to Nevada
120	URB17F	17f	Urban	Deep Percolation to Mexico
121	URB18A	18a	Urban	Return Flow to Salt Sink
122	URB18B	18b	Urban	Return Flow for Delta Outflow
123	URB19A	19a	Urban	Return Flow to Developed Supply (Other DAUCO within PA)
124	URB19B	19b	Urban	Return Flow to Developed Supply (Other PA)
125	URB19C	19c	Urban	Return Flow to Developed Supply (Other Region)
126	URB19D	19d	Urban	Return Flow to Carryover Storage for Next Water Year within DAU
127	URB2	2	Urban	Applied Water - Residential - Single Family Exterior
128	URB20	20	Urban	Return Flows Evaporation and Evapotranspiration
133	URB25	25	Urban	Conveyance Evaporation and ETAW
134	URB26A	26a	Urban	Conveyance Return Flow to Oregon
135	URB26B	26b	Urban	Conveyance Return Flow to Nevada
136	URB26C	26c	Urban	Conveyance Return Flow to Mexico
137	URB26D	26d	Urban	Conveyance Deep Percolation to Oregon
138	URB26E	26e	Urban	Conveyance Deep Percolation to Nevada
139	URB26F	26f	Urban	Conveyance Deep Percolation to Mexico
140	URB27A	27a	Urban	Conveyance Return Flows to Salt Sink
141	URB27B	27b	Urban	Conveyance Return Flow for Delta Outflow
142	URB28A	28a	Urban	Conveyance Return Flow to Developed Supply (Other DAUCO within PA)
143	URB28B	28b	Urban	Conveyance Return Flow to Developed Supply (Other PA)
144	URB28C	28c	Urban	Conveyance Return Flow to Developed Supply (Other Region)
145	URB29	29	Urban	Conveyance Seepage

146	URB3	3	Urban	Applied Water - Residential - Multi Family Interior
147	URB30	30	Urban	Conveyance Deep Percolation
148	URB31	31	Urban	Conveyance Deep Percolation to Salt Sink
153	URB4	4	Urban	Applied Water - Residential - Multi Family Exterior
154	URB5	5	Urban	Applied Water - Commercial Use
155	URB6	6	Urban	Applied Water - Industrial Use
156	URB7	7	Urban	Applied Water - Urban Large Landscape
157	URB8	8	Urban	Applied Water - Energy Production
158	URB9	9	Urban	Applied Water - Groundwater
274	WSR1	1	Wild and Scenic River	Applied Water
275	WSR2	2	Wild and Scenic River	Reuse of Return Flows within DAUCO
276	WSR3A	3a	Wild and Scenic River	Return Flow to Salt Sink
277	WSR3B	3b	Wild and Scenic River	Return Flow to Oregon - Mexico - Nevada
278	WSR3C	3c	Wild and Scenic River	Return Flow for Delta Outflow
279	WSR4A	4a	Wild and Scenic River	Return Flow to Developed Supply (Other DAUCO within PA)
280	WSR4B	4b	Wild and Scenic River	Return Flow to Developed Supply (Other PA)
281	WSR4C	4c	Wild and Scenic River	Return Flow to Developed Supply (Other Region)
286	WSR5	5	Wild and Scenic Rivers	Applied Water Use
287	WSR6	6	Wild and Scenic Rivers	Net Water Use (Applied Water - Reuse)
288	WSR7	7	Wild and Scenic Rivers	Net Water Use (ETAW + Flow/Salt Sink + Outflow)
289	WSR8	8	Wild and Scenic Rivers	Depletion

Glossary

federated See Federation 13

federation A federation is a group of data providers and users using jointly agreed-upon standards of operation in a collective fashion to ensure the interoperability of the resources they collectively hold and employ. The term may be used, for example, when describing the interoperation of distinct cyberinfrastructure networks with different internal structures. The term may also be used when human groups agree to collectively manage cyberinfrastructure development and operation using commonly held, and managed, requirements, standards and conventions, and operating procedures to ensure the interoperability of distinct cyberinfrastructure resources (cf. Wikipedia Definition). 13, 65

Federation See federation 65

interoperability The ability of computer systems or software to exchange and make use of data (adapted from the Oxford English Dictionary). 65

procedures An established or official way of doing something (Oxford English Dictionary). 12, 15, 65

Bibliography

[1] California Legislature, *AB-1755 The Open and Transparent Water Data Act. (2015-2016)*, Assembly Bill 1755, California Legislature, 2015.

[2] Dan Cayan and John Helly, *The Wireless Watershed at the Santa Margarita Ecological Reserve*, Southwest Hydrology **September/October** (2003).

[3] B. Chadwick, P.F. P.F. Wang, M. Brand, R. Flick, A. Young, W. O'Reilly, P. Bromirski, W. Crampton, R. Guza, J. Helly, T. Nishikawa, S. Boyce, M. Landon, M. Martinez, I. Canner, and B. Leslie, *A Methodology for Assessing the Impact of Sea Level Rise on Representative Military Installations in the Southwestern United States* `https://pubs.er.usgs.gov/publication/70178493`, Technical Report RC-1703, SPAWAR Systems Center Pacific, 2014.

[4] CODATA-ICSTI Task Group on Data Citation Standards and Practices, *Out of Cite, Out of Mind: The Current State of Practice, Policy, and Technology for the Citation of Data*, Data Science Journal **12** (2013), 1–75.

[5] B. W. Eakins, S.P. Miller, J. Helly, and B. Zelt, *The fully electronic IODP Site Survey Data Bank*, Scientific Drilling **2** (2006), 40–42.

[6] Eugene W. Rice and Roger B. Baird and Andrew D. Eaton and Lenore S. Clesceri (ed.), *Standard Methods for the Examination of Water and Wastewater*, vol. 22nd Edition, American Public Health Association, American Water Works Association, Water Environment Federation, 2012.

[7] Steven Glaser, Martha Conklin, and Roger Bales, *Building an Intelligent Water Information System – American River Prototype*, Technical report, CITRIS, 2013.

[8] Kay Gross, Edie Allen, Caroline Bledsoe, Robert Colwell, Paul Dayton, Megan Dethier, John Helly, Robert Holt, Nancy Morin, William Michener, Steward T. Pickett, and Susan Stafford, *Report of the Committee on the Future of Long-term Ecological Data (FLED)*, Tech. report, Ecological Society of America, Washington, D. C., 1995.

[9] J Helly, *Web-based data management for ecological analysis and synthesis*, Ecological Society of America, vol. Annual Meeting of the Ecological Society of America, Albuquerque, New Mexico, Ecological Society of America, 1997.

[10] J. Helly, T. T. Elvins, D. Sutton, D. Martinez, S Miller, S. Pickett, and A. M. Ellison, *Controlled Publication of Digital Scientific Data*, CACM **45** (2002), no. 5, 97–101.

[11] J. J. Helly, *Visualization of ecological and environmental data*, LTER Network Office, University of New Mexico, Albuquerque, New Mexico, 1998.

[12] John Helly, *New concepts of publication*, Nature **393** (1998).

[13] ———, *Digital libraries in hydrology*, Hydroinformatics: Data Integrative Approaches in Computation, Analysis, and Modeling (Praveen Kumar, Mike Folk, Momcilo Markus, Jay C Alameda, and Peter Bajcsy, eds.), Taylor and Francis, London, England, 2005, p. 552.

[14] John Helly and Maggi Kelly, *Collaborative Management of Natural Resources in San Diego Bay*, Coastal Management **29** (2001), 16.

[15] John Helly, Hubert Staudigel, and Anthony Koppers, *Scalable models of data sharing in Earth sciences*, Geochemistry Geophysics Geosystems **4** (2003).

[16] John J. Helly, *Cyberinfrastructure for data authorship, publication and application interoperability*, no. 1501928, AGU, AGU, December 2012.

[17] John J. Helly and Kevin T. Herbinson, *Visualization of the salinity plume from a coastal ocean water desalination plant*, Water Environment Research **66** (1994), no. 5, 753–758.

[18] ISTQB EXAM CERTIFICATION, *What is Agile model – advantages, disadvantages and when to use it?*, http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it (2018).

[19] J. Gage Marchini, *Connecting the "Drops" of California Water Data: Chapter 506: The Open and Transparent Water Data Act*, The University of the Pacific Law Review (2017), 785–799.

[20] William K. Michener, James W. Brunt, John J. Helly, Thomas B. Kirchner, and Susan G. Stafford, *Nongeospatial metadata for the ecological sciences*, Ecological Applications **7** (1997), no. 1, 330–242.

[21] NASA, *NASA Systems Engineering Handbook*, Tech. Report NASA-SP-2007-6105-Rev-1-Final-31Dec2007, NASA, 2007.

[22] OWIA Technical Working Group, *OWIA System Requirements Document*, Report, DWR, 2017.

[23] Paul F. Uhlir Rapprteur, *For Attribution – Developing Data Attribution and Citation Practices and Standards: Summary of an International Workshop*, Report ISBN 978-0-309-26728-1, National Acadeies Press, 2012.

[24] J. Pundsack, R. Belland, D. Broderson, G.C. Fox, J. Dozier, J. Helly, W. Li, P. Morin, M. Parsons, A. Roberts, C. Tweedie, and C. Yang., *Report on Workshop on Cyberinfrastructure for Polar Sciences. St. Paul, Minnesota. University of Minnesota Polar Geospatial Center.*, Report, National Science Foundation, 2013.

[25] F. M. Ralph, K. A. Prather, D. Cayan, J. R. Spackman, P. DeMott, M. Dettinger, C. Fairall, R. Leung, D. Rosenfeld, S. Rutledge, D. Waliser, A. B. White, J. Cordeira, A. Martin, J. Helly, and J. Intrieri, *CalWater Field Studies Designed to Quantify the Roles of Atmospheric Rivers and Aerosols in Modulating U.S. West Coast Precipitation in a Changing Climate.*, Bulletin of the American Meteorological Society (2015).

[26] Hubert Staudigel, John Helly, Anthony A. P. Koppers, Henry F. Shaw, William F. McDonough, Albrecht W. Hofmann, Charles H. Langmuir, Kerstin Lehnert, Baerbel Sarbas, Louis A. Derry, and Alan Zindler, *Electronic data publication in geochemistry*, Geochem. Geophys. Geosyst. **4** (2003), no. 3.

DEPARTMENT OF WATER RESOURCES • STATE OF CALIFORNIA •